D. Kuzmin
R. Löhner
S. Turek (Eds.)

Flux-Corrected Transport

Scientific Computation

Principles, Algorithms, and Applications

Springer

Scientific Computation

D. Kuzmin R. Löhner S. Turek (Eds.)

Flux-Corrected Transport

Principles, Algorithms, and Applications

With 124 Figures, 25 in color

Springer

Dr. Dmitri Kuzmin
Prof. Dr. Stefan Turek
University of Dortmund
Department of Mathematics
LS III, Vogelpothsweg 87
44227 Dortmund, Germany
kuzmin@math.uni-dortmund.de
ture@featflow.de

Prof. Rainald Löhner
School of Computational Sciences
George Mason University
Chain Bridge Road, 103 4260
Fairfax, Virginia 22030-4444, USA
rlohner@gmu.edu

Library of Congress Control Number: 2004116587

ISBN 3-540-23730-5 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media
springeronline.com

Typesetting: Data prepared by the authors using a Springer TEX macro package
Production: LE-TEX Jelonek, Schmidt & Vöckler GbR, Leipzig
Cover design: *design & production* GmbH, Heidelberg

Printed on acid-free paper 55/3141/YL 5 4 3 2 1 0

To SHASTA, 'a fluid transport algorithm that works'

Participants of the Workshop *30 years of FCT*

From left to right: G. Patnaik, R. Löhner, S. T. Zalesak,
D. L. Book, S. Turek, M. Möller and D. Kuzmin

Preface

In 1973, in the eighth year of its youth, the *Journal of Computational Physics* published the classic Boris and Book paper describing flux-corrected transport (FCT) [1]. Almost all of the monotonicity-preserving and non-oscillatory fluid transport algorithms of today trace their origins, ultimately, to ideas that first appeared in this paper.

Boris and Book's new and far-reaching idea was to locally replace formal truncation error considerations with conservative monotonicity enforcement in those places in the flow where the formal truncation error had lost its meaning, i.e., where the solution was not smooth and where formally high order methods would violate physically-motivated upper and lower bounds on the solution. This is today still the fundamental principle underlying the great bulk of the monotonicity-preserving and non-oscillatory algorithms that have appeared in more recent times. Occasionally this bit of history is lost in some of the more recent literature, in part due to the fact that the paper is now more than 30 years old (and the original publication [2] older still).

In [1], the authors applied this fundamental idea to a specific algorithm they termed SHASTA. They were able to show not only sharp monotone advection of linear discontinuities, but also sharp non-oscillatory gasdynamic shock waves. Included in [1] was a SHASTA calculation of a shock tube problem much more difficult than that used by Sod five years later [3], with nearly monotone results, and with no knowledge of the solution (e.g., Riemann solvers) built in to the algorithm. All of these calculations were the first of their kind with monotonicity-preserving algorithms of greater than first order accuracy. It was also in this paper that the term "flux-limiting" [1, pg. 50] appeared in print for the first time.

In the years following 1973, Boris and Book and colleagues published two more FCT papers in the *Journal of Computational Physics* [4, 5], followed by a chapter in the *Methods in Computational Physics* book series [6] that summarized their work up through 1976. These works refined their ideas, generalizing the algorithms to a larger class of which SHASTA was just one member. Their emphasis was on the continuity equation as a scalar representative of systems

of conservations laws, and upon advective phase error as a primary culprit in the elimination of the errors that remained after non-oscillatory behavior was eliminated via flux limiting. A more recent summary of FCT is given in the book by Oran and Boris [7].

Work on FCT algorithms has also thrived elsewhere in publications far too numerous to reference here. Two notable examples are the extension of FCT to fully multidimensional form by Zalesak in 1979 [8], and the generalization of FCT to finite element discretizations on unstructured grids (e.g., triangles and tetrahedra in two and three dimensions respectively) by Parrott and Christie in 1986 [9]. A general FEM-FCT methodology for the Euler and Navier-Stokes equations of fluid dynamics was introduced by Löhner *et al.* [10]. One of the consequences of this last development has been the ability to perform FCT calculations in extremely complex geometries. An example is the remarkable simulation of the 1993 World Trade Center blast which modeled in detail the garage of the building including all of the parked cars [11].

The response of the scientific computing community to FCT was and still is remarkably strong. Many new publications have emerged during the 1990s and early 2000s. The recent advances include but are not limited to: the use of FCT as an implicit subgrid scale model for *Monotonically Integrated Large Eddy Simulation* (MILES) [12], 'iterative and synchronous flux-correction' [13], monotonicity-preserving 'prelimiting' [14] as well as implicit FEM-FCT schemes based on a generalization of Zalesak's limiter [15]. Clearly the impact of the original paper [1] is still being felt long after its original publication. This impact is seen in an astounding number of citations and application of FCT to virtually every area of science, from aerodynamics and shock physics to atmospheric and ocean constituent transport, magnetohydrodynamics, kinetic and fluid plasma physics, astrophysics, and computational biology.

Most of the contributions compiled in the present volume are based on talks given at the Workshop "High-Resolution Schemes for Convection-Dominated Flows: 30 Years of FCT" which was held in September 2003 at the University of Dortmund, Germany. It was intended to provide a forum for evaluation of the progress made in the development of numerical methods for fluid dynamics during the three decades that elapsed since the birth of FCT. The high caliber of the presented results and many fruitful discussions have made this informal meeting a remarkably successful one. This has led us to unite our efforts and describe the state of the art in this book. The Editors would like to express their sincere gratitude to Prof. Roland Glowinski (University of Houston) for the expert review of the manuscript and thank Prof. Wolf Beiglböck (Springer-Verlag Heidelberg) for the prompt publication.

Dortmund, August 2004 *The authors*

References

1. J. P. Boris and D. L. Book, Flux-corrected transport. I. SHASTA, A fluid transport algorithm that works, *J. Comput. Phys.* **11**, 38 (1973).
2. J. P. Boris "A Fluid Transport Algorithm That Works," Proceedings of the seminar course on computing as a language of physics, 2-20 August 1971, International Centre for Theoretical Physics, Triest, Italy.
3. G. A. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, *J. Comput. Phys.* **27**, 1 (1978).
4. D. L. Book and J. P. Boris, Flux-corrected transport II: Generalizations of the method, *J. Comput. Phys.* **18**, 248 (1975).
5. J. P. Boris and D. L. Book, Flux-corrected transport. III. Minimal error FCT algorithms, *J. Comput. Phys.* **20**, 397 (1976).
6. J. P. Boris and D. L. Book, "Solution of Continuity Equations by the Method of Flux-Corrected Transport," Methods in Computational Physics, Vol. 16, Academic Press, 1976.
7. E. S. Oran and J. P. Boris, Numerical Simulation of Reactive Flow, Elsevier, New York, 1987; 2nd edition: Cambridge University Press, 2001.
8. S. T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, *J. Comput. Phys.* **31**, 335 (1979).
9. A. K. Parrott and M. A. Christie, "FCT applied to the 2-D finite element solution of tracer transport by single phase flow in a porous medium," in Proc ICFD Conf. on Numerical Methods in Fluid Dynamics, Oxford University Press, 609 (1986).
10. R. Löhner, K. Morgan, J. Peraire and M. Vahdati, Finite element flux-corrected transport (FEM-FCT) for the Euler and Navier-Stokes equations. *Int. J. Numer. Meth. Fluids* **7** (1987) 1093–1109.
11. J. D. Baum, H. Luo and R. Löhner, Numerical simulation of the blast in the World Trade Center; *AIAA*-95-0085 (1995).
12. J. P. Boris, F. F. Grinstein, E. S. Oran und R. J. Kolbe, New insights into Large Eddy Simulation, *Fluid Dynamics Research* **10**, Nr. 4-6, 1992, 199-227.
13. C. Schär and P. K. Smolarkiewicz, A synchronous and iterative flux-correction formalism for coupled transport equations. *J. Comput. Phys.* **128** (1996) 101–120.
14. C. R. DeVore, An improved limiter for multidimensional flux-corrected transport. *NASA Technical Report* AD-A360122 (1998).
15. D. Kuzmin and S. Turek, Flux correction tools for finite elements. *J. Comput. Phys.* **175** (2002) 525-558.

Contents

List of Contributors

Joseph D. Baum
Advanced Technology Group
SAIC, McLean, VA 22102, USA
joseph.d.baum@saic.com

David L. Book
Physics Department
Naval Postgraduate School
Monterey, CA 93943, USA
dlbook@nps.navy.mil

Jay P. Boris
NRL Chief Scientist and Director
Code 6400, U.S. Naval Research
Laboratory, Overlook Avenue
Washington, DC 20375, USA
boris@lcp.nrl.navy.mil

Christer Fureby
The Swedish Defense Research
Agency, FOI
Dept. of Weapons and Protection
Warheads and Propulsion
SE-172 90 Stockholm, Sweden
fureby@foi.se

Fernando F. Grinstein
Laboratory for Computational
Physics & Fluid Dynamics
U.S. Naval Research Laboratory
Washington, DC 20375, USA
grinstein@lcp.nrl.navy.mil

John P. Iselin
Dept. of Mechanical Engineering
Bucknell University, Lewisburg
PA 17837, USA

Dmitri Kuzmin
Institute of Applied Mathematics
(LS III), University of Dortmund
Vogelpothsweg 87, D-44227
Dortmund, Germany
kuzmin@math.uni-dortmund.de

Rainald Löhner
School of Computational Sciences
MS 4C7, George Mason University
Fairfax, VA 22030-4444, USA
rlohner@gmu.edu

Matthias Möller
Institute of Applied Mathematics
(LS III), University of Dortmund
Vogelpothsweg 87, D-44227
Dortmund, Germany
matthias.moeller@
math.uni-dortmund.de

Gopal Patnaik
Laboratory for Computational
Physics & Fluid Dynamics
U.S. Naval Research Laboratory
Washington, DC 20375, USA
patnaik@lcp.nrl.navy.mil

Stefan Turek
Institute of Applied Mathematics
(LS III), University of Dortmund
Vogelpothsweg 87, D-44227
Dortmund, Germany
ture@math.uni-dortmund.de

Steven T. Zalesak
Plasma Physics Division
U.S. Naval Research Laboratory
Washington, DC 20375, USA
zalesak@this.nrl.navy.mil

Foreword

Jay P. Boris

NRL Chief Scientist and Director
Code 6400, U.S. Naval Research Laboratory
Overlook Avenue, Washington DC, 20375, USA
boris@lcp.nrl.navy.mil

Flux-Corrected Transport (FCT) was invented more than 32 years ago to improve the quality of numerical convection algorithms. The key word here is quality. Tasked to improve the quality of strong, multidimensional shock computations for an important new research program at NRL, we all knew that the traditional Computational Fluid Dynamics (CFD) approaches were not faring well, even qualitatively, because fluid quantities such as mass density and chemical species number densities would become negative — a physical impossibility. Given the importance of dynamic, high Mach-number shocks in many fields, a number of talented people were working on this problem throughout the world. Flux-Corrected Transport, NRL's qualitatively better solution, broke new ground in 1971. People often ask me how the idea for the nonlinear flux limiter arose and what were the considerations that led to the statement of the underlying principle — "The antidiffusion stage should generate no new maxima or minima in the solution, nor should it accentuate already existing extrema". Here is how it happened.

In the winter of 1968 I was finishing my PhD at the Plasma Physics Laboratory in Princeton when Dr. Keith Roberts, the Director of the Culham Laboratory in England, visited us. Before I knew what had happened, Keith, John Green (my advisor), and Klaus Hain, also visiting from NRL to discuss progress on CFD with Keith, had arranged a post-doctoral appointment at Culham Laboratory for me, to be executed in the coming year under Keith Robert's tutelage. During that year Keith became my mentor and friend and gave me a firm inventive foundation in CFD to go with my plasma physics and astrophysics training. Our work ranged over just about every computational topic imaginable during the year. However, toward year's end, I began preparing to move to Washington D.C. to participate in a large theoretical/computational project that Professor John Dawson of Princeton had arranged with NRL. Thus Keith and I began looking at ways to improve multidimensional CFD — as we had promised Klaus. During some particularly discouraging tests, Keith introduced me to Godunov's 1959 "theorem" — that

second- and higher-order algorithms could not preserve the physical positivity property. It was certainly clear that high-order schemes were not necessarily bringing greater accuracy so physics would have to step in to shore up the failing numerics.

At NRL in September of 1969 everyone was in a rush. Plasma physics, MHD, and CFD simulation techniques were all being pursued simultaneously for the big program. The strong, fluid-dynamic shock problem had become the number one computational roadblock by the fall of 1970 so I was urged to concentrate on the problem full time, finally developing the FCT convection algorithm in the winter. Called SHASTA (Sharp and Smooth Transport Algorithm), this continuity-equation solver was developed over weekends on computers at Princeton and Oak Ridge Laboratory. The results were sufficiently astonishing that both Keith and Klaus urged me to present the new capability at the conference/workshop "Computing as a Language of Physics," held in Trieste, Italy in the summer of 1971. This first paper "A Fluid Transport Algorithm that Works" was published by the organizers in book form through the IAEA.

In 1971 David Book joined the NRL melting pot and began working with me on FCT to extend the continuity equation solver to treat sets of coupled fluid dynamic equations and to generalize the formulation. Our 1973 article in the *Journal of Computational Physics* (submitted in November 1971) extended the Trieste paper to include strong shocks and other fluid flows. This first journal article in the FCT series of three was reprinted as the most cited article up to that time in the 25th Anniversary Issue of the JCP. This article was also acknowledged as the most cited NRL publication at the NRL 75th Anniversary Celebration a year later.

SHASTA was constructed as a layered set of corrections, each layer being added to mitigate the problems introduced by the previous layer. This general approach seldom works, as I have subsequently discovered (repeatedly), but this particular time it did. The starting point was general dissatisfaction with the negative densities and nonphysical wiggles near sharp gradients observed in leapfrog and Lax-Wendroff convection and a deep suspicion of the excess diffusion introduced in first order donor-cell algorithms. I also decided to work on a single, general continuity equation, trusting to earlier work with Keith that a numerical model for any set of fluid equations could be built up with such a building block.

Impressed by the potential of Arbitrary Lagrangian-Eulerian (ALE) algorithms, the first layer of SHASTA is a conservative Lagrangian displacement and compression of linear trapezoids of fluid. It is local, positive, and easy to program but leads to jagged sequences of mismatched trapezoids that quickly spawn most of the problems of a fully Lagrangian approach. The second layer of SHASTA, therefore, was a conservative Eulerian remap of the displaced, distorted trapezoids of fluid back onto the original fixed grid. This introduced a zero-order numerical diffusion whose coefficient, 1/8, was generally worse than donor cell diffusion! As a result, the third layer of SHASTA was an ex-

plicit, linear antidiffusion to subtract the excess diffusion introduced by the remap. Of course this resulted in solutions just as bad as the second-order Lax-Wendroff and leapfrog algorithms.

Rather than resign myself to adding a strong numerical viscosity based on derivatives that would be essentially meaningless near discontinuities and sharp gradients, I began looking in detail at the physics of the local profiles of the density profile being convected — before and after antidiffusion. In each specific case it was clear just how much of the antidiffusive correction flux could be used at each interface and how much had to be thrown away to prevent a particular cell value from being pushed past the monotonicity limits imposed by the neighboring values. However, the expression of this "flux correction," now usually called a flux limiter or slope limiter, was different in every case. Finally recognizing that the sign function of the density difference across a cell interface could be used to collapse the behavior near maxima and near minima into a single formula, the pivotal *max-min* expression emerged after a couple of days of fooling around during a marathon computing session at Oak Ridge in March 1971.

David recognized that the underlying linear convection algorithm could be just about anything; the SHASTA algorithm in the first working FCT code was just a vehicle and maybe not even a particularly good one. We tried variants of Lax-Wendroff and the other simple linear algorithms, resulting in a set of three papers in the *Journal of Computational Physics*, the second with Klaus Hain. Making the zeroth-order diffusion coefficient even larger, 1/6, reduced the phase error in the convection from second order to fourth order with improved retention of profile structure. We found that the initial density profile could be made to emerge unscathed, when the interface velocities are zero, despite the diffusion and antidiffusion stages. David called this "Phoenical" FCT and we use this trick today. David also forced some mathematical rigor into the mix. Stung by a public criticism from Conrad Longmire and Greg Canavan at his first major FCT presentation that the results were faked, he did much experimentation and analysis to understand why FCT was working so well and insisted on a steady stream of peer-reviewed publications. Perhaps the best of these, if not the most cited, was the chapter "Solution of Continuity Equations by the Method of Flux-Corrected Transport" in volume 6 of *Methods in Computational Physics*. David recalls these early days in Chapter 1 below.

Many of NRL's fluid dynamicists made significant contributions in the 1970s. Steve Zalesak invented the multidimensional limiter, one of the main breakthroughs for FCT, and this was modified and extended by Rick DeVore to include MHD flows where $\nabla \cdot B = 0$ is an important consideration. Steve characteristically understates his own contributions in Chapter 2 below. Ed MacDonald showed us that FCT could be added to spectral algorithms, spawning an FFT FCT and a (linearly) reversible FCT algorithm. Elaine Oran and John Gardner contributed significantly to refinement of the time- and direction-split LCPFCT software package described in detail in the first

and second editions of the book *Numerical Simulation of Reactive Flow*. This simple toolbox now has had tens of thousands of copies distributed worldwide. Kailas Kailasanath introduced boundary layer considerations for his ground breaking work on acoustic-vortex interactions and combustion instabilities. Ted Young, Niels Winsor, Sandy Landsberg, and Charles Lind played a major role in vectorizing and then parallelizing FCT, leading directly to NRL's current FAST3D general-geometry CFD models. Rainald Löhner took the lead in carrying FCT into the Finite Element world with a practical general-geometry formulation that also provided a basis for adaptive meshing and moving geometry considerations. He and Joseph Baum consider the status and direction of FCT over the last thirty years in Chapter 5, focussing somewhat on blast and shock problems. Gopal Patnaik first extended FCT to an implicit formulation called BICFCT for slow but fully compressible reacting flows in performing the world's first dynamical, multidimensional flame simulations. A computer manufacturer, Texas Instruments, Inc. even got into the act. They added pairwise vector *min*, *max* and *sign* operations to the pipeline instruction space of their parallel processing Advanced Scientific Computer (ASC) at NRL's request so that the Flux-Corrected Transport in FAST3D would be even faster.

In the chapters following you will also see some of the other things done with these simple ideas in the ensuing three decades. My own focus at the Naval Research Laboratory has been on large-scale urban transport of contaminants. Chapter 4 by Gopal Patnaik, Jay Boris, Fernando Grinstein, and John Iselin considers this very important application, showing how to reduce the numerical dissipation still further for detailed urban and building airflows. 3D turbulence, the source of contaminant dispersion in these urban problems, is treated by Monotone Integrated Large Eddy Simulation (MILES), an LES formulation based implicitly on the monotonicity properties of FCT with stochastic backscatter in a 4th-order phase, time-accurate, finite volume model for detailed building and city aerodynamics. In retrospect, it should not be surprising that an algorithm designed for treating sharp gradients and discontinuities in compressible flow should work equally well for rotational flows. Nevertheless, it has taken fifteen years since its explanation in 1989 for the community to acknowledge the benefits of monotone (in our case FCT) algorithms for the treatment of turbulence. Chapter 3, by Fernando Grinstein and Christer Fureby, considers this originally unintended but greatly appreciated byproduct of FCT.

Chapters 6, 7 and 8, by Dmitri Kuzmin, Matthias Möller, and Stefan Turek complete the book, presenting a unified theoretical foundation for "Algebraic Flux-Correction" from the Finite Element perspective. This approach is intrinsically multidimensional and unifies FCT and TVD variants of monotonicity-preserving algorithms. The authors retrace the original progression of developments, beginning with scalar conservation laws, extending to the hyperbolic compressible Euler equations, and finishing with a methodology for treating incompressible flow problems.

The Conception, Gestation, Birth, and Infancy of FCT

David L. Book

Physics Department
Naval Postgraduate School
Monterey, CA 93943, USA
dlbook@nps.navy.mil

Summary. How Flux-Corrected Transport came to be, as recalled by one of the innovators: recollections of how FCT was developed and of the individuals responsible.

1 Conception

In 1970 there was essentially no reliable way to solve fluid equations numerically. By 1971 there was one. Flux-Corrected Transport (FCT) was the first method developed that yielded physically acceptable results for such equations. The present paper describes how Jay Boris and I developed FCT, with what I hope is an accurate account of our thinking at the time, the path we followed in the course of the development, including the missteps and blind alleys, and the roles of the other individuals who were involved.

Fluid or hydrodynamic equations are partial differential equations dominated by convective motion, that is, equations in which convective derivative terms of the form

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla f$$

play a decisive role, where f is one of the dependent fluid variables (density of mass, momentum, energy, or charge; pressure, entropy, species concentration, etc.), t is time, $\mathbf{v}$ is the flow velocity, $\nabla \equiv \frac{\partial}{\partial \mathbf{r}}$, and $\mathbf{r}$ is position. Examples are the continuity equation for a compressible medium with mass density ρ,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{v} = 0,$$

the Euler (momentum) equation in the presence of a scalar pressure p and a constant gravitational acceleration $\mathbf{g}$, which can be written

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot \rho \mathbf{v}\mathbf{v} + \nabla p + \rho \mathbf{g} = 0,$$

and the Navier–Stokes equation (the Euler equation with the inclusion of viscosity),

$$\rho\frac{\partial \mathbf{v}}{\partial t} + \rho\mathbf{v}\cdot\nabla\mathbf{v} + \nabla p = \mu\nabla^2\mathbf{v}.$$

Such equations are also called convective or hyperbolic, although strictly speaking the Navier–Stokes equation is parabolic in regions where the Laplacian term dominates. In general, the equation expressing the transport of any continuously distributed quantity, together with terms describing sources and losses, is of this form. This category includes all of the familiar conservation laws.

A third of a century ago computer resources were meager by comparison with today's technology, but most of the general computational approaches in use today were known, at least in broad outline, and the first steps had already been taken toward applying them. Finite differences and finite elements (the distinction between them then was somewhat blurred, though now people usually associate these terms with differencing schemes on structured and unstructured grids, respectively), characteristics, quasiparticles, and spectral methods had all been invented, and all of these tools were being applied to the problem of finding computational solutions to fluid equations. Dozens of Fortran codes based on each of these methods, or combinations of several of them, were in existence. They all fell short of what I feel is the goal of any numerical treatment of evolution equations: using limited, i.e., discretized, information about the dependent variables in order to predict the values of those variables at a later time *with the same accuracy or level of confidence.*

The presence of convective derivatives is what makes fluid equations difficult to solve numerically. Because of them the characteristics, the space–time trajectories along which the values of the fluid variables are constant, slope. In order to predict the values at a position $\mathbf{r}$ of the fluid variables at a time t' later than a time t for which their values are known, it is necessary to use information from the points through which the characteristics passed at time t, which are in general different from $\mathbf{r}$. Thus, to predict the values of the fluid variables at a particular point on, e.g., a finite-difference grid may require knowing their values at an earlier time from a location *that was not on the grid.*

In thinking about the implications of sloping characteristics I like to use an analogy with what I call the window problem. Suppose you are in a room with several windows looking out onto a nearby railroad track. When a train comes by each car in succession appears at a given window. Let's say that the train is moving from your left to your right. If you want to know what car is going to appear next at the window in front of you, you can look out the window to the left of it. You don't have to keep a continuous watch; it suffices to glance over at the second window at intervals, which correspond to the discrete timesteps in a finite-difference scheme. Of course, if the windows are too widely spaced, or if you are too close to them, there may be several cars hidden out of sight between the two windows. This is analogous to using

too coarse a mesh in your difference scheme. Likewise, you may miss a car if the intervals between glances are too long, which corresponds to using too long a timestep Δt.

But the information you get is limited in another way, because the window is narrow and you can't see a whole railroad car at one time. In order to make good predictions you have to know something about the kinds of railroad cars that the train is allowed to have. A glimpse may be all you need to know that a car is a boxcar or a tank car or a flat car, but what if it's a car of a type you've never seen before, say, one carrying a crane to lift up wrecks? Or one with unusual dimensions or proportions (Fig. 1)? Your prediction is implicitly based on a bias or prejudice in favor of the kinds of cars you expect to see.

Fig. 1. Pathological railcars

In the same way, a numerical technique must contain a built-in bias about the form solutions can take, because the available information is limited by discretization. No technique can handle all situations. Each one must be tailored to fit a particular class of problems.

The problems Jay and I were interested in were time-dependent fluid problems, especially those involving supersonic flow, and more generally, systems with discontinuities or steep gradients. These include not just shocks (which can occur only when there is supersonic flow), but also contact and tangential discontinuities and abrupt changes in temperature, species concentration, etc. We opted to employ a finite-difference approach because it simplified coding, particularly in multidimensions and at boundaries, and because that was what we were most comfortable with.

The naïve approach to finding a finite-difference approximation to differential equations on a mesh with a uniform spacing Δx is to expand the derivatives in Taylor series:

$$f(x \pm \Delta x) = f(x) \pm \frac{\partial f}{\partial x}\Delta x + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}(\Delta x)^2 \pm \cdots$$

Thus,

$$f(x+\Delta x) - f(x-\Delta x) = 2\frac{\partial f}{\partial x}\Delta x + O(\Delta x)^3$$

or

$$\frac{\partial f(x,t)}{\partial x} = \frac{1}{2\Delta x}\left[f(x+\Delta x,t) - f(x-\Delta x,t)\right] + O(\Delta x)^2,$$

and similarly

$$\frac{\partial f(x,t)}{\partial t} = \frac{1}{2\Delta t}\left[f(x,t+\Delta t) - f(x,t-\Delta t)\right] + O(\Delta t)^2.$$

Hence a straightforward finite-difference approximation to the 1-D passive advection equation

$$\frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x} = 0$$

is

$$\rho(x,t+\Delta t) = \rho(x,t-\Delta t) - \varepsilon\left[\rho(x+\Delta x,t) - \rho(x-\Delta x,t)\right],$$

where $\varepsilon = u\Delta t/\Delta x$ is the Courant number. The resulting finite-difference approximation is of course just second-order leapfrog, a useful scheme that has been widely adopted.

The Taylor-series approach has a number of strengths. It yields difference schemes (such as leapfrog) that are accurate for problems with slowly varying profiles, i.e., those in which discontinuities are absent. Also, it facilitates the analysis of amplitude and phase errors. Thus, assume a sinusoidal density profile

$$\rho_j^0 = \rho_0 \exp(2\pi i j\kappa/N),$$

where $j\Delta x$ is position, κ is the mode number and N is the number of mesh points. If

$$\rho_j^1 = \rho_1 \exp(2\pi i j\kappa/N)$$

is the corresponding profile found after one timestep, then the amplification factor is $A_\kappa = |\rho_1/\rho_0|$ and the relative phase error (the error in the speed with which features are advected numerically, divided by the correct speed) is $R_\kappa = (\kappa\varepsilon)^{-1}\tan^{-1}(\mathrm{Im}\,\rho_1/\mathrm{Re}\,\rho_0) - 1$.

But the Taylor-series expansion approach also has some notable deficiencies. It works only when "order" makes sense, i.e., when the scale of variation is large compared with the mesh spacing, so that the neglect of higher-order terms ("truncation errors") is justified. Consequently, it breaks down at discontinuities, where "dispersive" ripples make their appearance. (The finite-interval Gibbs effect, the analog for discrete Fourier transforms of the well-known Gibbs phenomenon, can also contribute to errors in the vicinity of a discontinuity. I will return to this topic below.)

The key insight (which to the best of my knowledge originated with Jay) is that the Taylor-series approach fails because it does not enforce *positivity*, a property sometimes called monotonicity. For physical reasons some variables

can only take on positive values. Examples are mass and energy density (but not charge or momentum density), temperature, and pressure. Nothing in the Taylor-series approach ensures this. Positivity violations are found to be worst near discontinuities. At discontinuities formal high-order accuracy is less important than maintaining positivity.

This may be an appropriate time to try to spell out what is meant by the term "discontinuity." Shocks, contact discontinuities, and slip lines (tangential discontinuities) would be physically discontinuous in the absence of dissipation. Because dissipation can never be totally absent, however, no physical quantity is ever truly discontinuous, at least in classical physics. There are no discontinuities in nature.

In finite-difference approximations, on the other hand, all changes are discontinuous, but some are more discontinuous than others. A criterion is needed to determine when a discontinuity is "real." This criterion may involve calculating whether the relative or absolute change in a variable exceeds some threshold value, or it may be more complicated. The exact definition, as always, depends on the nature of the problem and one's expectations about the solution being sought.

In the absence of a specific criterion a finite-difference scheme will not be able to distinguish a weak physical discontinuity from a smooth feature or from noise. As I will show, there are several reasons why a profile that should be smoothly varying might develop ripples or "bumps" in a computational treatment. Hence in any scheme susceptible to such errors nonphysical features will be treated just like physical ones. The trick is to avoid, as much as possible, developing them in the first place.

The simplest system of equations that models shocks is that of ideal hydrodynamics. This consists of the continuity and Euler equations and an equation for the energy density

$$E = \frac{1}{2}\rho \mathbf{v}^2 + \frac{p}{\gamma - 1}.$$

The energy equation follows from the adiabatic law (pressure equation) in the form

$$\frac{\partial p}{\partial t} + \mathbf{v} \cdot \nabla p + \gamma p \nabla \cdot \mathbf{v} = 0,$$

where γ is the ratio of specific heats.

These three equations express the conservation of mass, momentum, and energy. That they are conservative is important; if the adiabatic law is used instead of the formally equivalent energy equation, then pressure remains positive but the Rankine–Hugoniot (jump) conditions are violated. If the energy equation is chosen as one of the fundamental evolution equations, then energy is conserved and shocks obey the Rankine–Hugoniot conditions, but pressure is a derived quantity which can become negative. The Rankine–Hugoniot conditions imply that the jump in entropy varies as $(\mathrm{M} - 1)^3$, where M is the Mach number. Using the adiabatic law means that only weak shocks ($\mathrm{M} \to 1$) are calculated correctly. This dilemma confronts all finite-difference schemes.

Physically, viscous dissipation is responsible for creating entropy at a shock front. This is what allows shocks to satisfy the Rankine–Hugoniot conditions. The equations of ideal hydrodynamics in conservative form contain no explicit viscosity terms, but nevertheless admit solutions that satisfy the Rankine–Hugoniot conditions. These so-called weak solutions represent a zero-viscosity limit. Physical shocks in systems with nonzero viscosity differ in having nonzero thickness; that is, the jump takes place over a region of finite extent.

Any numerical scheme must include *some* dissipation if it is going to allow the jump conditions to be satisfied. Before the invention of FCT the most successful finite-difference treatments of supersonic flow, developed at Los Alamos and widely employed elsewhere, introduced an artificial viscosity term, typically in the form of a velocity-dependent coefficient multiplying the second difference of the velocity. This forced the production of entropy at places where the velocity underwent abrupt change and allowed the Rankine–Hugoniot conditions to hold.

The trouble with this approach was that in order to generate enough entropy the coefficient had to be large. This in turn caused the shock to be spread out over several or many mesh spaces. That was fine if the physical viscosity in the problem was large, i.e., if the Reynolds number Re was of order unity. But for realistic problems of high-Reynolds-number flow — say, $Re > 100$ — it was hopelessly inaccurate. To reduce the shock thickness to reasonable values would require being able to determine where shocks were located or were about to form or introducing thousands of grid points in each coordinate direction, which would have been prohibitively expensive. (Remember, this was in 1970.)

There is another downside to using artificial viscosity. The timestep limit in finite-difference problems arises because information can travel no faster than one mesh space per timestep (assuming three-point difference schemes), which means that the Courant number must not exceed unity. Violating this condition in explicit finite-difference treatments of the ideal hydrodynamic equations leads to catastrophic numerical instability. If the spatial mesh in a calculation is refined the timestep must shrink proportionately to stay within safe limits. But when diffusion terms of the form $\mu\nabla^2\mathbf{v}$ dominate, then the timestep scales with the square of the mesh space. Consequently, in methods using artificial viscosity even refining the mesh locally near the shocks becomes much more expensive if the diffusion terms are differenced explicitly. (Implicit differencing has problems of its own.)

A shock wave heats the medium through which it is traveling, which causes the speed of sound to increase. Hence information propagates faster, so that signals in the region behind a shock tend to catch up with the shock. This means that shocks are self-steepening. Other discontinuities, such as contact surfaces, are not. As a result, shear surfaces and interfaces between two different media or between two regions in the same medium with different properties

tend to be smeared out by numerical diffusion and are more difficult to model than shocks.

The passive advection equation models advection and propagates contact discontinuities, but in contrast to the system of hydrodynamic equations it has no self-steepening mechanism. Consequently, it is a more stringent test of numerical fluid-equation solvers. In addition it is simpler to work with than a set of nonlinear equations. Thus, it was natural for us to choose it as our test bench instead of the full set of hydrodynamic equations. We felt that if we could do a good job solving this equation numerically we could solve most fluid problems.

For our basic test problem we chose to propagate a square wave 20 mesh spaces in width across a grid 100 mesh spaces wide (Fig. 2) for 800 timesteps with a constant Courant number $\varepsilon = 0.2$, using periodic boundary conditions in order to allow the profile to reenter the system. As a quantitative figure of merit we used the average absolute value of the error (the L1 norm), abbreviated A.E. This test problem subsequently became a kind of universal standard in the computational physics community.

When various numerical methods are evaluated using this problem one can readily discern their strengths and shortcomings. Those that are inaccurate to zeroth order in terms of Taylor-series expansions in $k\Delta x$, the nondimensionalized wavenumber, fail to track the analytical solution correctly, yielding profiles that move either too fast or too slow. Those that are first-order accurate, such as donor-cell (upwind differencing), yield profiles that move at the right speed and maintain positivity, i.e., $\rho(x) > 0$ everywhere, but become smeared out over an ever-increasing portion of the grid (Fig. 3). In other words, they are highly *diffusive*.

Methods that are second-order accurate, such as leapfrog or Lax–Wendroff, yield profiles that develop multiple ripples (Fig. 4). These arise because the various Fourier harmonics that make up the square wave propagate at different speeds. The long-wavelength components propagate at nearly the right speed,

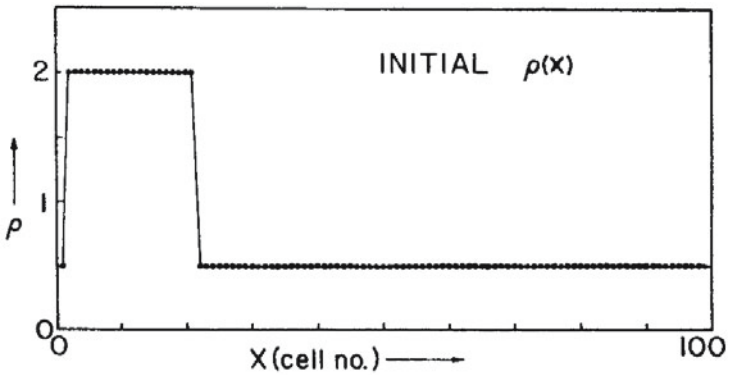


Fig. 2. Initial conditions for the square-wave test

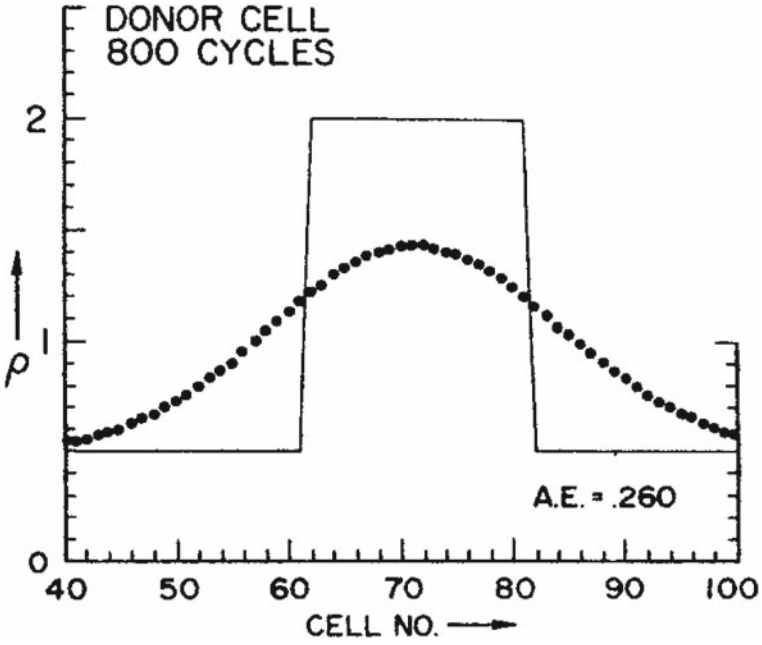


Fig. 3. Square-wave test of Donor Cell (first-order accuracy)

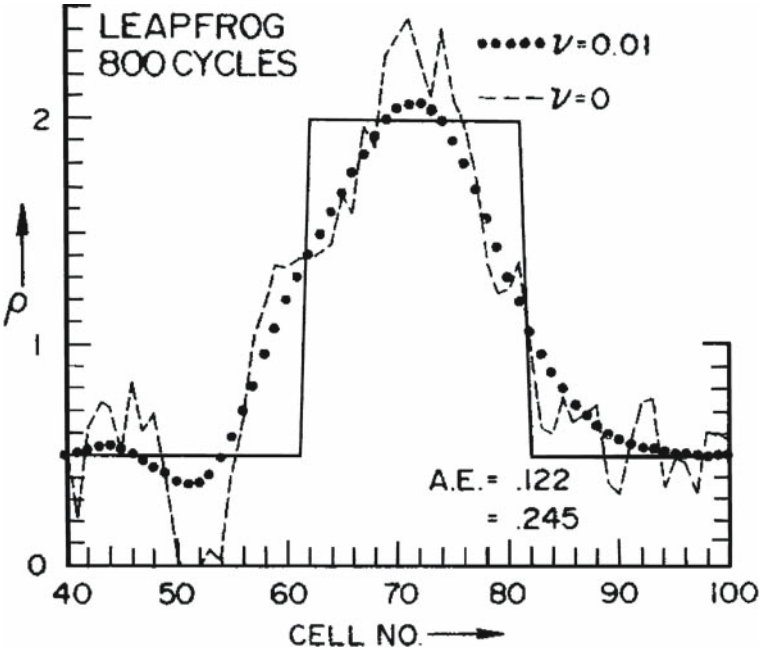


Fig. 4. Square-wave test of Leapfrog (second-order accuracy)

while the short-wavelengths usually lag behind. In other words, the errors are *dispersive*. The ripples grow in amplitude until the profile can become negative in some places. Thus, second-order algorithms do not maintain positivity. Notice that introducing a small amount of smoothing ($\nu = 0.01$) not only eliminates these negative values but also reduces the A.E. The optimum choice of the smoothing coefficient ν is, however, problem-dependent.

It is difficult to say which is worse, diffusive or dispersive errors, Scylla or Charybdis. Going to higher than second order doesn't solve the problem. Every technique that can be expressed in terms of linear finite-difference operations on the dependent variable — including every finite-difference treatment of the passive advection equation and its hydrodynamic kin in existence prior to the invention of FCT — suffers from one or the other failing.

2 Gestation

FCT was the first *nonlinear* finite-difference technique. In my view there were three main steps in our thinking that led to its development: expressing all operations in terms of fluxes, certainly not a new idea at the time; a transport algorithm called SHASTA, which is highly diffusive even in the limit of zero velocity, suggesting the use of "antidiffusion" to cancel out the diffusive errors; and the idea of correcting (limiting) the antidiffusive fluxes in order to maintain positivity (the nonlinear ingredient).

Fluxes are quantities of an extensive variable (e.g., mass, momentum, energy) that pass from one cell or grid point to another. If a finite-difference algorithm can be expressed entirely in terms of fluxes then it is guaranteed to be conservative, because what is removed from one place reappears in another. Thus, advection (transport only) can be approximated using transportative fluxes:

$$\rho_j^T = \rho_j - \varphi_{j+1/2}^T + \varphi_{j-1/2}^T,$$

where

$$\varphi_{j+1/2}^T = \varepsilon_{j+1/2}\rho_{j+1/2},$$

with $\varepsilon_{j+1/2} = u_{j+1/2}\Delta t/\Delta x_{j+1/2}$ and $\rho_{j+1/2}$ defined on the cell boundary, $x_{j+1/2} = \frac{1}{2}(x_j + x_{j+1})$, and $\Delta x_{j+1/2} = x_{j+1} - x_j$.

Similarly, diffusion can be expressed in terms of diffusive fluxes:

$$\rho_j^D = \rho_j + \varphi_{j+1/2}^D - \varphi_{j-1/2}^D,$$

where

$$\varphi_{j+1/2}^D = \nu_{j+1/2}(\rho_{j+1} - \rho_j).$$

In both instances *what is subtracted from one cell is added to its neighbor*, so the total "mass" is conserved.

The second ingredient, SHASTA, was Jay's idea. He devised it by means of a geometric approach. Imagine a finite-difference approximation ρ_j to some continuous variable $\rho(x)$, represented by histograms or rectangles (the broken lines in Fig. 5). Connect the points denoting the values of ρ_j with straight lines to form trapezoids. The area contained in the resulting trapezoids is the same as that contained in the rectangles.

If this profile is then transported across the grid with some velocity $u(x)$, in general each trapezoidal packet of fluid undergoes advection together with compression or expansion. Each mesh point x_j moves to a new location x'_j. In one timestep the two vertical sides x_j and x_{j+1} of a trapezoid thus move in general by different distances, causing it to be deformed as well as translated. The condition $x'_j < x'_{j+1}$, which is necessary to ensure positivity, imposes the limitation $\varepsilon < 0.5$. At the end of each timestep the mass contained in the trapezoid is reassigned to the two rectangles it straddles: the portion to the left of the boundary between two cells is assigned to the left-hand cell and the portion to the right is assigned to the right-hand cell (Fig. 6).

It is easy to see that the process of creating trapezoids and reassigning their mass is highly diffusive. In fact, in the limit when the velocity is identically zero the new value of ρ_j equals the old value plus a second difference with coefficient of 0.125:

$$\rho_j^{n+1} = \rho_j^n + \frac{1}{8}(\rho_{j+1}^n - 2\rho_j^n - \rho_{j-1}^n).$$

Since this algorithm is diffusive, the natural thing to do is to subtract the excess diffusion, or to put it another way, to apply antidiffusion. Antidiffusion is just diffusion with a negative coefficient:

$$\frac{\partial \rho}{\partial t} = -A\frac{\partial^2 \rho}{\partial x^2}, \quad A > 0.$$

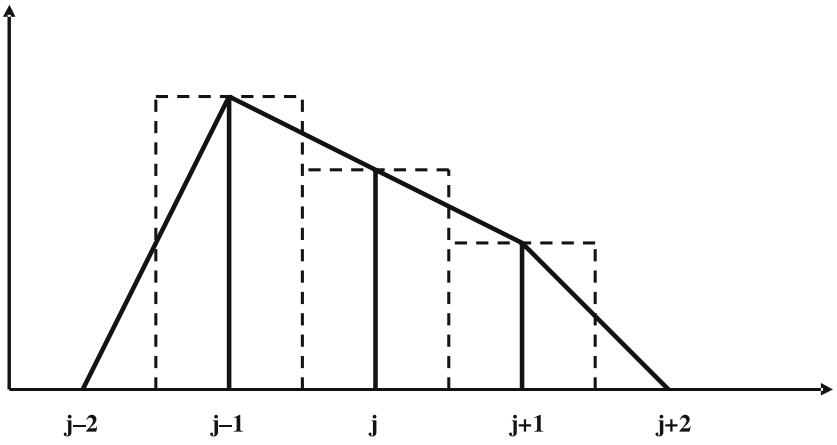


Fig. 5. Finite-difference approximation represented by rectangles and by trapezoids

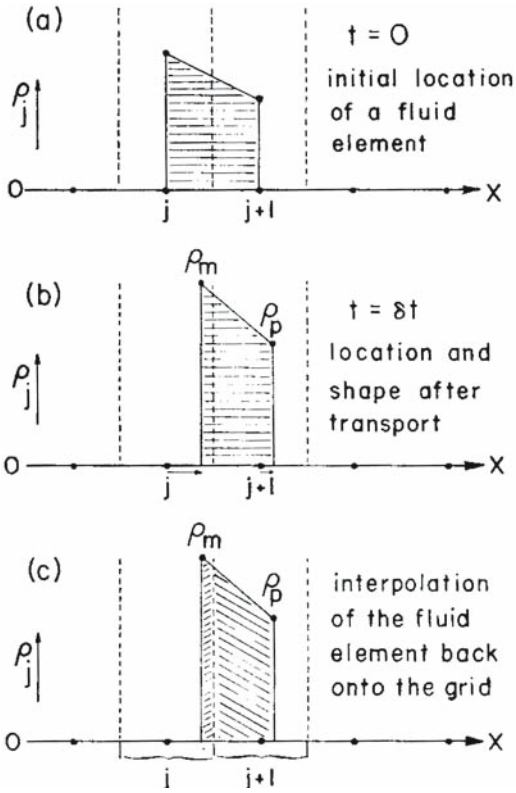


Fig. 6. SHASTA contains a zeroth-order diffusion

Whereas diffusion erodes features, antidiffusion steepens them. Discontinuities become sharper and new extrema can occur. If this process is sufficiently drastic it can violate positivity (Fig. 7).

One way to look at this is to recognize that diffusion converts a second-order algorithm like Lax–Wendroff or leapfrog that has wiggles into one that is first-order like donor cell. Hence taking out the excess diffusion changes it back into a second-order algorithm and restores the wiggles.

Jay recognized that by correcting or *limiting* the antidiffusive fluxes before they are applied he could avoid restoring the wiggles. A flux large enough to push a point j down below its neighbors will create a new minimum that might go negative, so it is necessary to reduce the size of this flux. If the profile already has a minimum at j, then allowing it to be pushed further down is dangerous, so it is necessary to zero any flux that tends to do so. Likewise, it is necessary to avoid enhancing maxima on negative profiles. The two rules can be combined into one: replace the "raw" antidiffusive fluxes with fluxes "corrected" so that *no new extrema can form*

and existing extrema cannot grow. Figure 8 illustrates the four different situations a limiter can face when dealing with the flux between the points j and $j + 1$, assuming the gradient is positive there: (a) no pre-existing

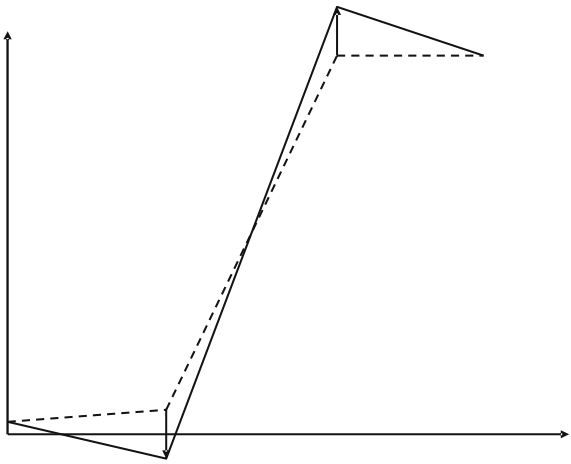


Fig. 7. Antidiffusion can violate positivity

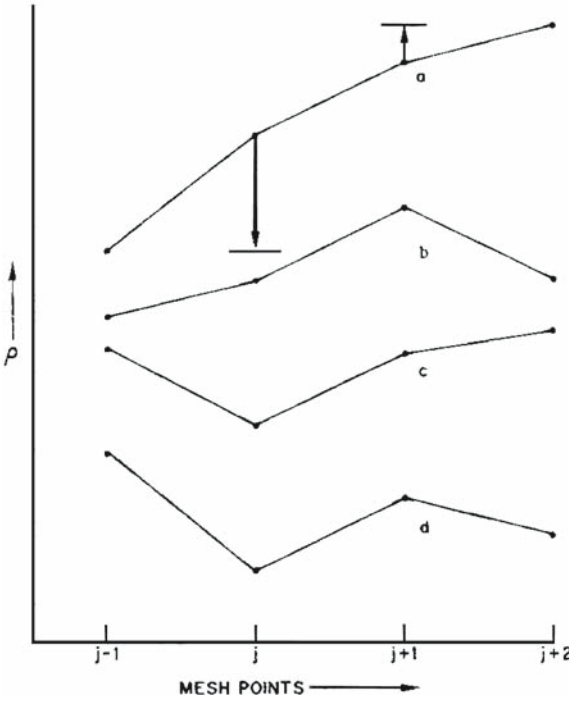


Fig. 8. Possible actions of a "strong" flux limiter on a positive flux

extrema; (b) a maximum present at $j + 1$; (c) a minimum present at j; or (d) both.

This was the first flux limiter that gave satisfactory results. We eventually realized that there are other workable variants, but this one, in which each flux is corrected without reference to others, is arguably the simplest. Because its action sometimes amounts to overkill (in ways I will describe shortly) I called it "strong" flux limiting.

The idea of using a flux limiter was the crucial ingredient in FCT. Some of the credit for it should go to the late Klaus Hain, our colleague at the Naval Research Laboratory. At the time Klaus was also trying to develop an algorithm to solve convective equations. I believe he was the first to recognize that some sort of adjustment or correction in the fluxes was needed, but he had not yet found the right formulation. Jay picked up the idea from him and made it work.

In some ways we were Klaus's competitors more than his collaborators. Klaus was an extremely able numericist, and Jay clearly wanted to be the first to find a successful algorithm. Jay and I worked closely together, but Klaus worked almost entirely alone. I thought it was because Klaus, German-born and about two decades older than us, had trouble speaking and writing English. His wife Gertie, however, assured everyone that he was uncommunicative in German too.

3 Birth

By mid-1971 we had a working one-dimensional flux-corrected version of SHASTA. The name SHASTA supposedly came not from the famous mountain, but from the *nom de guerre* of a topless dancer. (I cannot confirm this from my own experience.) Possibly because he thought this lacking in dignity Jay contrived the acronym "SHarp And Smooth Transport Algorithm," which is how we presented it to the world.

Initially Jay had been unwilling to reveal the secret of FCT to outsiders, but one day he said to me, "Here comes Super-Klaus!" He suspected that Klaus was writing up his work for publication, and that was what changed his mind. We decided that one paper was not enough. Instead there would be a series, which became the celebrated FCT-1, FCT-2, and FCT-3. Jay was the obvious choice as the senior author of the first paper, most of which he drafted, but he must have felt a little guilty about scooping Klaus, because he suggested that the subsequent papers should bear the names of all three of us. Each of us would write the first draft of one of them; Klaus would be lead author of one and I would be lead author of the other. In the event though, Klaus — true to form — never did write anything, and his name appeared only on the second paper.

Actually, the first publications describing FCT appeared not in 1973, but two years earlier. (Hence the workshop for which I prepared the present

paper should properly be called FCT-32, which is a more computational-sounding name than FCT-30 anyway.) I wrote the first journal article, which closely followed the manuscript we were preparing for submission to the Journal of Computational Physics at the time. It appeared in the November issue of NRL Reports of Progress, a house organ read by almost nobody. The only reason I wrote it was because I was then serving on a committee set up in order to make the Reports of Progress more relevant and better known.

But our very first publication was oral and never found its way into print. The application for which FCT was intended was modeling the atmospheric nuclear explosions that would have resulted from the infamous antiballistic missile program. The 1971 Symposium on High-Altitude Nuclear Effects (HANE), held at Stanford in August, was the first exposure of FCT to the computational community at large. The meeting was attended by government and private contractors funded by the Defense Atomic Support Agency, which that year became the Defense Nuclear Agency (DNA), afterward called the Defense Special Weapons Agency, and currently the Defense Threat Reduction Agency.

Several people from NRL were there, presenting results on various aspects of HANE. The attendees from other organizations were of course competing with us for DNA support. We wanted to impress them and our sponsor, but not to tell them too much. I gave the talk on the design of FCT. Jay was very insistent that I not reveal any secrets, and I didn't. I showed the results of the square-wave tests, I described SHASTA, but I didn't explain how the flux limiter worked, nor did I mention an embellishment called a "steepener." A lot of what I said was sheer doubletalk. I have a vivid recollection of a frustrated Greg Canavan from the Air Force Weapons Laboratory standing in the audience, asking question after question, trying to pin me down. Finally he said, "You keep saying, 'Another way to look at it is so-and-so.' Just tell us how it works!"

The meeting was a triumph for FCT and for our group at NRL. Our co-authors on that initial publication, Carl Wagner and Ed McDonald, were among the first users of SHASTA. Carl later worked on controlled fusion in private industry; Ed, who is still at NRL, changed fields and is now widely known for his work modeling sound propagation in the ocean. Jay and I both began applying SHASTA to a variety of problems, as did at least half a dozen of our colleagues in the Plasma Dynamics Branch. Many of these led to plasma and ionospheric physics papers.

At the same time we continued to refine and extend the method while preparing for publication in JCP. Some of the improvements resulted in making the code more efficient. Jay found that the flux limiter, which originally required a nested sequence of IF statements, could be expressed by a one-line formula:

$$\varphi^c_{j+1/2} = \sigma_{j+1/2} \max[0, \min(\sigma_{j+1/2}\Delta_{j-1/2}, |\varphi_{j+1/2}|, \sigma_{j+1/2}\Delta_{j+3/2})].$$

Here

$$\Delta_{j+1/2} = \rho_{j+1}^{TD} - \rho_j^{TD},$$

$$\varphi_{j+1/2} = \mu(\rho_{j+1}^{TD} - \rho_j^{TD}) \quad \text{or} \quad \varphi_{j+1/2} = \mu(\rho_{j+1}^{T} - \rho_j^{T}),$$

where ρ_j^T is the transported density, ρ_j^{TD} is the transported diffused density, and

$$\sigma_{j+1/2} = \text{sign}(\varphi_{j+1/2}).$$

The two versions given here for the raw antidiffusive flux both use the same dimensionless coefficient μ, but the second version, which we called phoenical, has the advantage that when the flow velocity u vanishes $\rho_j^T \to \rho_j$. This permits the profile, which has been smeared out by diffusion, to be restored "like a phoenix," so that the algorithm reduces to the identity operation.

There was one aspect of the early versions of FCT that I felt uneasy about, the use of steepeners. In FCT-1 the antidiffusion coefficient μ was given as "1/8." We wrote "The quotation marks indicate that more exact cancellation of errors can be achieved if one expends a small amount of computational effort by including at least rough approximations to the velocity- and wavenumber-dependent corrections [11]." Footnote 11 explained just what was meant by wavenumber dependence: The antidiffusion coefficient was bigger than the diffusion coefficient by an amount that depended on the size of the discontinuity. Naturally, this yielded very nice square waves. The drawback was that it turned every bump into a square wave!

The steepener was Jay's idea. The rationale for it was that our tests produced profiles that were actually less sharp than shocks calculated with FCT

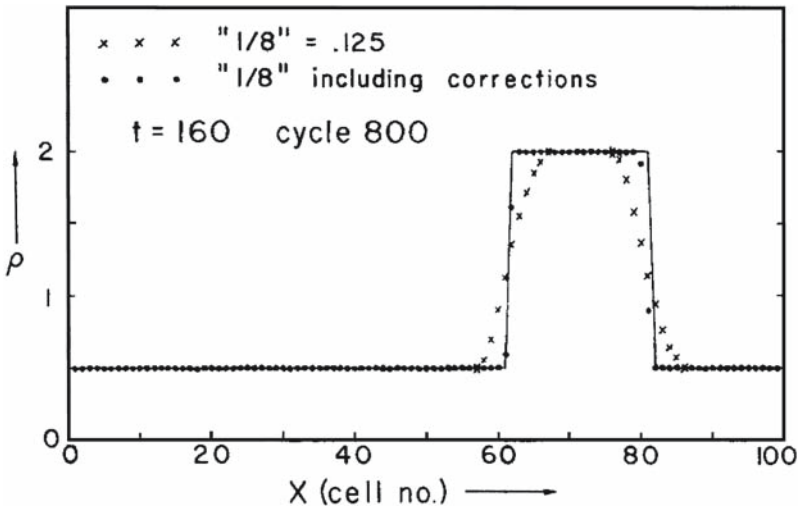


Fig. 9. The effect of introducing an artificial steepener

because, as I mentioned earlier, the passive advection equation has no mechanism for self-steepening. The steepener was supposed to model this mechanism, but to me it seemed an out-and-out kludge, and people who read footnote 11 apparently agreed. In the event steepeners disappeared after FCT-1 and were never mentioned again.

4 Infancy

As we continued to improve and extend SHASTA we gradually realized that what we had was more widely applicable and more general than a mere algorithm. Jay insisted on using the name "flux-corrected transport," but it was several years before the rest of the community distinguished between FCT and SHASTA. (Jay also tried to reserve the term "scheme" for competing algorithms, while referring to FCT as a "method" or "technique," but that invidious distinction, not surprisingly, never caught on. Neither did the term "Flux-Uncorrected Transport," which he used once in a talk.)

My first contribution to helping transform FCT from an algorithm into a method was to derive a formula for the diffusive transport step of SHASTA, i.e., the result of assigning mass to the trapezoids, transporting them, and reassigning it to the mesh. (The original code simply followed the geometric construction.) This formula can be expressed algebraically as

$$\rho_j^{n+1} = \frac{1}{2}Q_-^2(\rho_j^n - \rho_{j-1}^n) + \frac{1}{2}Q_+^2(\rho_{j+1}^n - \rho_j^n) + (Q_- + Q_+)\rho_j^n,$$

where

$$Q_\sigma = \frac{1/2 - \sigma u_j \Delta t/\Delta x}{1 + \sigma(u_{j+\sigma} - u_j)\Delta t/\Delta x}, \quad \sigma = \pm 1.$$

For a uniform velocity field, $u_j = u$, it reduces to

$$\rho_j^{n+1} = \rho_j^n - \frac{\varepsilon}{2}(\rho_{j+1}^n - \rho_{j-1}^n) + \left(\frac{1}{8} + \varepsilon^2\right)(\rho_{j+1}^n - 2\rho_j^n + \rho_{j-1}^n),$$

which is just Lax–Wendroff plus a zeroth-order (in ε) diffusion with coefficient 1/8. So, Jay thought, why not use Lax–Wendroff as the transport algorithm even when the flow field is nonuniform, adding diffusion to it and then applying antidiffusion? This worked just as well as SHASTA.

We tried flux-correcting leapfrog, and that worked fine too. Initially we used a diffusion/antidiffusion coefficient of 0.125 because that was what SHASTA used. It turned out that the limit on the Courant number in order to ensure positivity, $\varepsilon < 0.5$, is the same as for SHASTA, although the geometric interpretation no longer holds. Why is 1/8 the best choice? Wouldn't it be better to use less diffusion sometimes, especially when the flow velocity is very small and less is needed to ensure positivity? I tried running our standard test using different values of the diffusion/antidiffusion coefficient (Fig. 10). It is

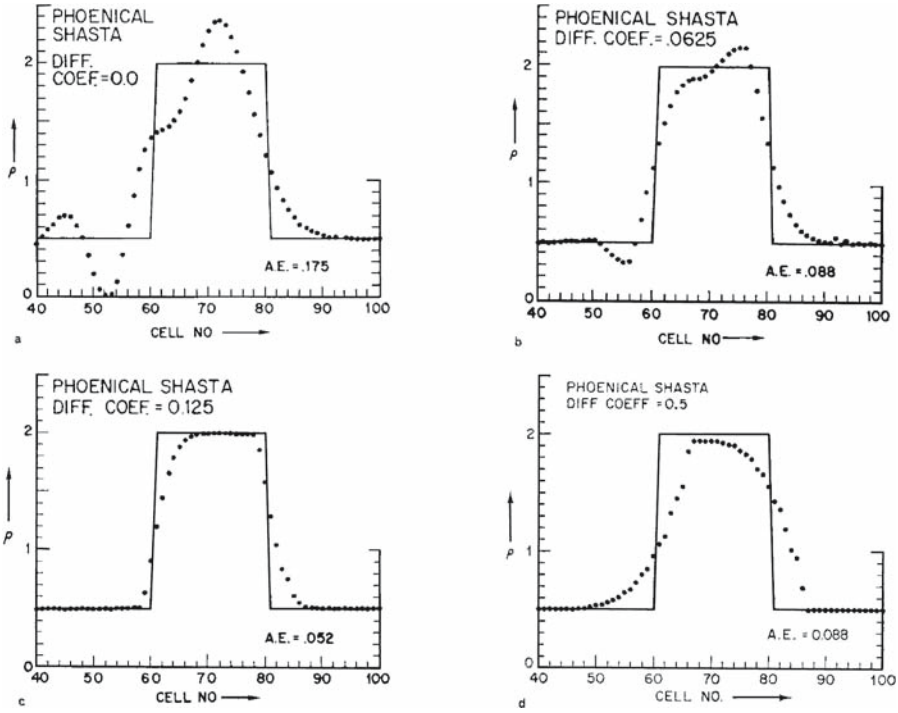


Fig. 10. Result of varying the diffusion/antidiffusion coefficient

evident that the best results come from using 1/8, and that using too much can be as bad as using too little.

We noticed that combining a velocity-dependent antidiffusion with donor cell created a second-order-accurate algorithm formally identical to Lax–Wendroff. If the antidiffusive fluxes are limited with the same prescription that was used in SHASTA the result is a flux-corrected version of donor cell. In this algorithm the diffusion and antidiffusion coefficients vanish when the flow velocity does. In fact, for any value of u flux-corrected donor cell embodies the smallest diffusion coefficient consistent with positivity. But when we tested the new algorithm on square waves we found that the results were inferior to those obtained with flux-corrected Lax–Wendroff or leapfrog. Evidently minimizing the diffusion does not produce the best algorithm.

Jay found the explanation: phase accuracy is more important than amplitude accuracy. This is because the cumulative residual diffusion due to flux limiting results from mashing down the short-wavelength harmonics, which always propagate too slowly or too fast in a finite-difference algorithm. Minimizing the relative phase error works better (because fewer harmonics need mashing) than making the amplification factor as close as possible to unity. In fact, the amplification factor must go to zero for very short wavelengths, or else they wouldn't get mashed. For good results an FCT algorithm should

have phase error that is at least second-order (i.e., proportional to k^2 when expanded in powers of wavenumber).

Another lesson we learned from these experiments was that the diffusion and antidiffusion coefficients can be velocity-dependent, provided that the diffusive and antidiffusive fluxes cancel out. This gave us an extra degree of freedom, an additional knob to turn in fine-tuning algorithms.

While musing on an algorithm called hopscotch I had read about in JCP I dreamed up "reversible FCT," which is basically flux-corrected Crank–Nicolson. It applies half the transport step to the old values and half to the new, together with a diffusion/antidiffusion that is also symmetric between the old and new values:

$$\rho_j^T + \frac{\varepsilon}{4}\left(\rho_{j+1}^T - \rho_{j-1}^T\right) + \nu\left(\rho_{j+1}^T - 2\rho_j^T + \rho_{j-1}^T\right) = \rho_j^0 - \frac{\varepsilon}{4}\left(\rho_{j+1}^0 - \rho_{j-1}^0\right) + \nu\left(\rho_{j+1}^0 - 2\rho_j^0 + \rho_{j-1}^0\right).$$

The transported diffused density ρ_j^{TD} is found by adding $\nu\left(\rho_{j+1}^T - 2\rho_j^T + \rho_{j-1}^T\right)$ to ρ_j^T. Then the raw antidiffusive flux $\varphi_{j+1/2} = \nu\left(\rho_{j+1}^T - \rho_j^T\right)$ is corrected with respect to ρ_j^{TD} and reapplied. This algorithm is second-order for any choice of ν because of symmetry. Setting $\nu = 1/6 + \varepsilon^2/12$ makes the phase error fourth-order. With this choice the square-wave test yielded an A.E. of 0.033, the best we had yet found (Fig. 11).

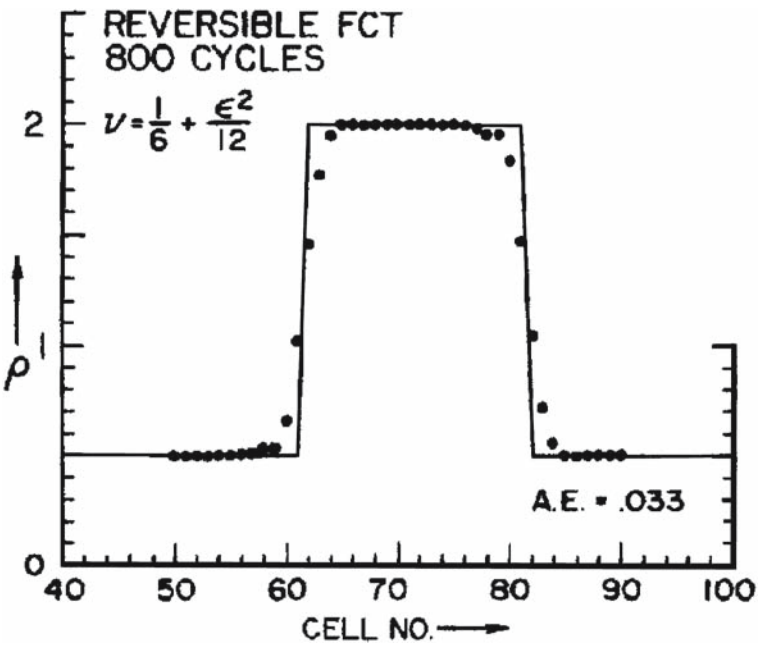


Fig. 11. Reversible FCT

Even without flux correction the underlying transport routine in an FCT algorithm gives better results than conventional algorithms. In a sense it *should*, because conventional algorithms are based on three-point stencils (they involve only the mesh point in question and its two nearest neighbors), while the extra antidiffusion step in FCT introduces information from next-nearest neighbors as well. Can FCT algorithms be made even more accurate by using more complicated stencils?

Thinking about this led me to invent Fourier-transform FCT. Start by Fourier-transforming the density ρ:

$$\rho_j = \sum_{\kappa=1}^{N} \tilde{\rho}_\kappa \exp(2\pi i j \kappa / N).$$

(This is of course implies an N-point stencil, but with fast transforms the computational overhead is acceptable.) Advance each component according to the exact solution of the transformed advection equation:

$$\tilde{\rho}_\kappa(t + \Delta t) = \tilde{\rho}_\kappa(t) \exp(-2\pi i \kappa u \Delta t / \Delta x).$$

Now transform back to x space. The resulting solution has no dissipation and no phase error. On the face of it this algorithm should be error-free, at least for passive advection with a uniform velocity. Indeed, if $\varepsilon = u\Delta t/\Delta x$ is an integer the solution reproduces the analytic solution exactly. There is no need for additional diffusion, antidiffusion, or flux correction.

If $u\Delta t/\Delta x$ is not an integer, however, the solution that results from inverting the Fourier transform has ripples. Jay called this the finite-interval Gibbs effect. These ripples are related to the "window problem," i.e., the impossibility of knowing what goes on between the mesh points. Plotting the inverse of the discrete Fourier transform over the entire interval on which the original discrete function is defined yields a continuous curve. This can be shown to be the smoothest function that passes through the values of the function on the mesh. If the original function contains a sharp discontinuity, this plot not only exhibits the usual Gibbs over- and undershoots at the top and bottom of the jump, but also has wiggles between the mesh points (Fig. 12).

Translating the profile over a fraction of a mesh space exposes these wiggles to view. The Fourier transform thinks a function should behave this way in order to be as smooth as possible, whereas the physics favors one that has as few wiggles as possible. But we know how to fix that: flux-correct it. In other words, add some diffusion, then apply an equal amount of antidiffusion with a flux limiter. Nothing dictates the choice of the coefficient, so we used $\mu = \nu = 1/8$. (Why not? It worked in other algorithms.) The resulting value of 0.022 for the A.E. is the smallest one we ever found (Fig. 13). Thus, at the cost of a higher operation count, Fourier-transform FCT emerged as the optimum FCT algorithm, at least for this test problem.

The three JCP papers focused almost entirely on how FCT worked and on the design of FCT algorithms. In 1976 we contributed an article to the series

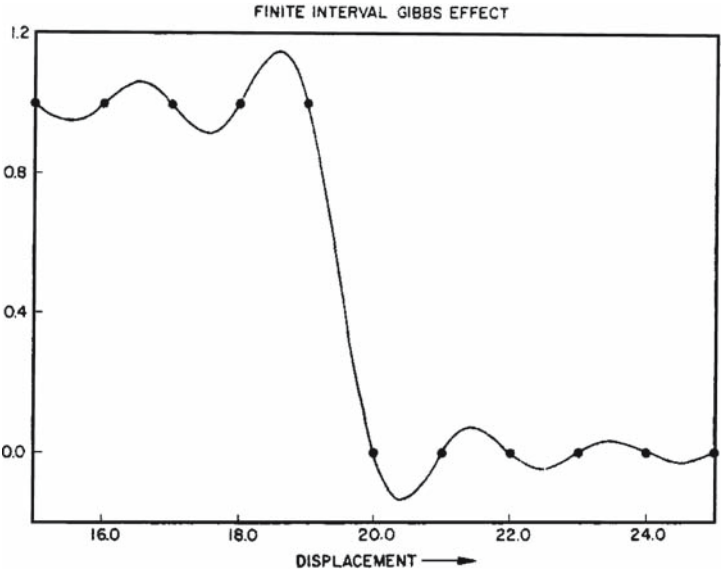


Fig. 12. Continuous function obtained from the discrete Fourier transform of a jump

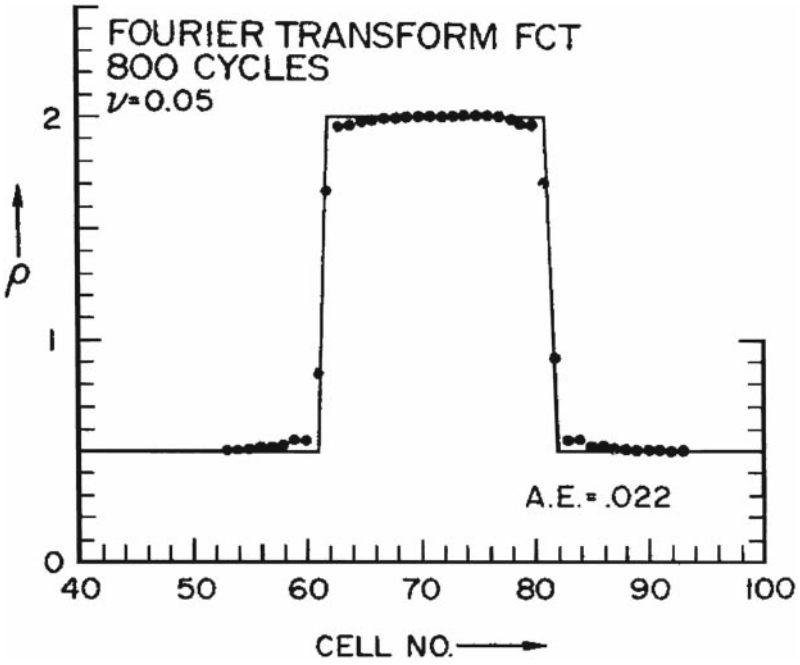


Fig. 13. The optimum FCT algorithm

Methods of Computational Physics. It appeared as a chapter in volume 16, which surveyed numerical techniques for plasma physics problems. This article (which perhaps deserves to be designated FCT-4) reviewed our previously published work, but also discussed some of the codes incorporating FCT and the problems to which we had applied them.

When it came to real applications we found that FCT is not perfect. One source of error is something than can be called residual diffusion, which results when antidiffusion fails to completely cancel diffusion.

The most dramatic manifestation of this is "clipping." An extremum loses a little bit of amplitude each timestep, even if it isn't being advected across a grid, because diffusion squashes the extremum down and strong flux limiting doesn't allow the antidiffusion to push it back up. Ultimately the peak changes into a characteristic flat-topped structure, which we called a plateau. For example, an initially sharp maximum subjected to repeated diffusion and antidiffusion operations gradually flattens out until it forms a plateau three points across (Fig. 14), which is stable.

Figure 14 shows that this flattening is less severe with phoenical or implicit antidiffusion than with the original explicit form of antidiffusion, in which the raw flux is calculated from the diffused profile. The reason is obvious. If T stands for the transport operation, D stands for a three-point centered second difference with coefficient ν, and A represents the same operation with coefficient $\mu = -\nu$, the three versions of FCT can be represented symbolically as

$$\begin{aligned}
&\text{Explicit:} && \rho^1 = (1+A)\rho^{TD} = (1+A)(1+T+D)\rho^0;\\
&\text{Phoenical:} && \rho^1 = [(1+A)(1+T)+D]\rho^0;\\
&\text{Implicit:} && \rho^1 = (1+\ D)^{-1}\rho^{TD} = (1+\ D)^{-1}(1+T+\ D)\rho^0.
\end{aligned}$$

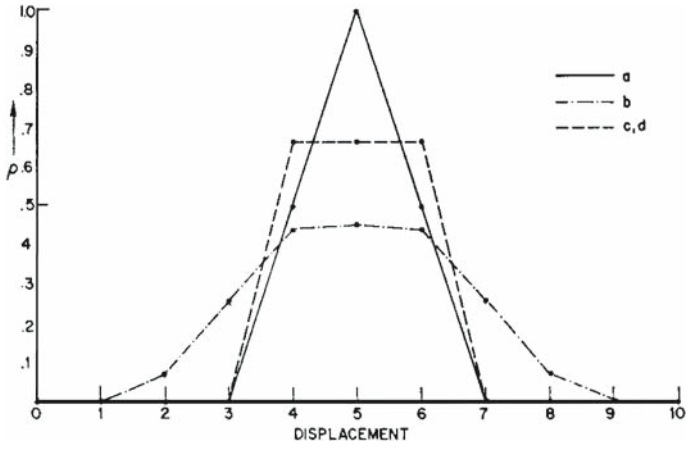


Fig. 14. Result of 20 repetitions to profile (**a**) of diffusion and (**b**) explicit; (**c**) phoenical; (**d**) implicit antidiffusion with strong flux limiting, using coefficients $\mu = \nu = 0.2$

It is clear that in the limit $T \rightarrow 0$ phoenical and implicit FCT reduce to the identity operation, or would except for the action of the flux limiter, but explicit FCT does not. (This is of course why phoenical antidiffusion was invented.)

In an effort to eliminate clipping we tried a number of different flux limiters. One of my early attempts was the one-sided flux limiter. This involved changing the flux limiter so that maxima could grow on positive profiles but minima could not (and vice-versa for negative profiles). The resulting algorithm preserved positivity, but gave rise — unsurprisingly — to one-sided ripples.

Another idea of mine that didn't work out was called "flux-limited diffusion." The innovation here was to apply diffusion only where needed to prevent extrema from growing relative to their original values, rather than put it in everywhere and remove it where it was not needed. The test results, however, were disappointing. Square waves propagated using flux-limited diffusion were badly eroded. The trouble with this approach was that it failed to ensure high phase accuracy in the underlying transport scheme. Ultimately we decided to stick with strong flux limiting. It was easier to live with the symptoms of the disease than with the side effects of the cures.

Another form of residual diffusion is more subtle. If the antidiffusion coefficient is smaller than the diffusion coefficient, some residual diffusion is present even in the absence of flux limiting. The algorithm with fourth-order phase accuracy described in FCT-3 has an amplification factor that just barely — by less than 0.5% — exceeds unity. (We didn't know this until Phil Colella pointed it out years later.) Because of this it was found that codes yielded the best results when the antidiffusion coefficient was reduced slightly by multiplying by a factor called a "mask." (The name arose because the correction was applied using MASK, a Fortran IV bit-manipulating instruction).

An annoying but fairly innocuous departure from realism arises because the flux limiter stops dispersive ripples from growing on the slopes of hills or valleys only when they are about to form new extrema. This leaves flat "terraces" on the slopes, which I like to think of as the ghosts of departed ripples. Improving the phase accuracy of the algorithm helps reduce terracing, but the only real cure is to use a more elaborate form of flux limiting that takes into account second derivatives of the profile.

When we started using FCT for two-dimensional problems a new and much more serious question arose: How could we generalize the flux limiter to multidimensions? Should all the fluxes be corrected in one sweep? In applying antidiffusive fluxes in, say, the x direction, should we worry about extrema only along that axis, or should we look at all points in the neighborhood? The coding was tortuous, and every prescription we tried seemed to fail in some combination of circumstances.

Finally we gave up and decided to use coordinate splitting. That is, on each timestep we treated each coordinate independently, carrying out 1-D transport and 1-D flux limiting first in the x direction and then in y. Symbolically this

can be written, e.g.,

$$\rho^{TD} = (1 + T_x + D_x)(1 + T_y + D_y)\rho^0;$$

$$\rho^1 = (1 + A_x)(1 + A_y)\rho^{TD}.$$

Obviously this introduces spurious terms, e.g., $A_x A_y$. Most of the time they do no great harm because the errors tend to cancel out, but there are some situations where splitting creates unphysical effects. One example is when plateau formation occurs in both coordinate directions at the same time, for example on a hilltop. The result is that contours of constant density (or pressure, etc.) become square. I got very tired of going to meetings where I had to explain why my code generated square fireballs.

5 The Next Generation

Despite all our efforts we never found a way around the problem of creating a multidimensional flux limiter. Eventually I decided that it was insoluble, and I told every colleague who expressed an interest in it not to waste his time. As is well known, the problem turned out not to be insoluble. Likewise, the "pioneering idea of blending high- and low-order discretizations," cited on the FCT-30 web page, was not part of the original concept. It was a later development, due to the same individual who created the multidimensional flux limiter, Steve Zalesak. But that is his story to tell.

References

1. D. L. Book, J. P. Boris, B. E. McDonald, and C. E. Wagner, SHASTA, a transport algorithm that works, Proc. Symposium on High-Altitude Nuclear Effects, Stanford, CA, August 10–12, 1971.
2. D. L. Book and J. P. Boris, A transport algorithm that works, NRL Reports of Progress, Nov. 1971, p.1.
3. J. P. Boris and D. L. Book, Flux-Corrected Transport: I. SHASTA, a fluid transport algorithm that works, *J. Comput. Phys.* **11**, 38–69 (1973) [FCT-1].
4. D. L. Book, J. P. Boris, and K. Hain, Flux-Corrected Transport: II. Generalizations of the Method, *J. Comput. Phys.* **18**, 248–283 (1975) [FCT-2].
5. J. P. Boris and D. L. Book, Flux-Corrected Transport: III. Minimal-error FCT algorithms, *J. Comput. Phys.* **20**, 397–431 (1976) [FCT-3].
6. J. P. Boris and D. L. Book, Solution of Continuity Equations by the Method of Flux-Corrected Transport, in: B. Alder, S. Fernbach, M. Rotenberg, and J. Killeen (Eds.), *Methods in Computational Physics*, vol. 16 (Academic Press, New York, 1976).

The Design of Flux-Corrected Transport (FCT) Algorithms For Structured Grids*

Steven T. Zalesak

Plasma Physics Division
Naval Research Laboratory
Washington, D.C. 20375, USA
zalesak@this.nrl.navy.mil

Summary. A given flux-corrected transport (FCT) algorithm consists of three components: 1) a high order algorithm to which it reduces in smooth parts of the flow; 2) a low order algorithm to which it reduces in parts of the flow devoid of smoothness; and 3) a flux limiter which calculates the weights assigned to the high and low order fluxes in various regions of the flow field. One way of optimizing an FCT algorithm is to optimize each of these three components individually. We present some of the ideas that have been developed over the past 30 years toward this end. These include the use of very high order spatial operators in the design of the high order fluxes, non-clipping flux limiters, the appropriate choice of constraint variables in the critical flux-limiting step, and the implementation of a "failsafe" flux-limiting strategy.

This chapter confines itself to the design of FCT algorithms for structured grids, using a finite volume formalism, for this is the area with which the present author is most familiar. The reader will find excellent material on the design of FCT algorithms for unstructured grids, using both finite volume and finite element formalisms, in the chapters by Professors Löhner, Baum, Kuzmin, Turek, and Möller in the present volume.

1 Introduction: Modern Front-Capturing Methods

We are interested in systems of conservation laws of the form

$$\frac{\partial \mathbf{q}(\mathbf{x},t)}{\partial t} + \nabla \cdot \mathbf{f}(\mathbf{q},\mathbf{x},t) = 0 \tag{1}$$

where $\mathbf{q}(\mathbf{x},t)$ and $\mathbf{f}(\mathbf{q},\mathbf{x},t)$ are vector functions of the independent variables $\mathbf{x}$ and t, which we henceforth refer to as space and time respectively. Examples of such equations include the Navier-Stokes equations, the equations

* This work was supported by the U.S. Department of Energy

of magnetohydrodynamics (MHD), the Vlasov equation, and passively-driven convection.

It is well known that differentiable solutions to (1) may cease to exist after a finite time t_s, even if the initial conditions $\mathbf{q}(\mathbf{x}, 0)$ are smooth. After t_s, only the integral or "weak" form of (1) will have solutions, and these will contain discontinuities in $\mathbf{q}$ and/or one or more of its derivatives. We will term such discontinuities "fronts" for the purpose of this chapter. This situation is addressed by the Lax-Wendroff Theorem, which states that if one's numerical approximation to Eq. (1) is in "flux" or "conservation" form, and the numerical solution converges everywhere but on a set of measure zero to some solution, then that solution is a weak solution to Eq. (1). Thus the great majority of methods designed to treat fronts in the context of Eq. (1) are in conservation form, i.e., a form consisting of numerical fluxes connecting adjacent grid points, these fluxes being used to advance the numerical solution in time.

Using conservation form does not by itself give the desired result however, since one still needs to compute a convergent solution. In general, numerical methods not designed to deal with fronts will not produce the desired convergence in their presence, often producing numerical oscillations that degrade the solution severely. It is precisely this situation that prompted Von Neumann and Richtmyer to add an explicit artificial dissipation term to Eq. (1), the idea being to smooth the fronts to the point where they are resolved on the grid as narrow but smooth features, thereby producing the desired convergence. This is the fundamental idea underlying nearly all conservative numerical methods designed to handle fronts, including the FCT algorithms we address here. (We are excluding, of course, methods that treat fronts as moving internal boundaries, the class of methods known as "front-tracking methods." We are also excluding random choice methods [8, 3, 4].) For the purposes of this chapter, we shall refer to methods which attempt to smooth a front into a narrow but smooth transition as "front-capturing methods."

Over the past 30 years a host of algorithms, known variously as "modern" front-capturing methods or "high resolution methods," have been developed in an attempt to perform calculations more accurately and more efficiently than with the more traditional explicit artificial dissipation approach. The first of these methods was flux-corrected transport (FCT) [1], [2], but there are now a large number of others. What distinguishes the "modern" front-capturing methods from their predecessors is their attempt to constrain the numerical fluxes, grid point by grid point and timestep by timestep, in such a way as to avoid the production of unphysical values in the solution vector $\mathbf{q}$ in and near the fronts, and at the same time treat the regions in space and time in which $\mathbf{q}$ is smooth as accurately as possible. Clearly the success of these methods depends critically on an accurate criterion for determining what constitutes an unphysical value for $\mathbf{q}$, one of the primary topics of this chapter.

One way of stating the design philosophy of these methods, and the one we shall embrace in this chapter, is as follows:

When the numerics fails, substitute the physics.

Clearly the designers of such algorithms must possess a knowledge of the physics being addressed if they are to be successful.

These modern front-capturing methods may be thought of as consisting of three parts:

1. an algorithm to which they reduce in regions of time and space where $\mathbf{q}$ is smooth;
2. an algorithm to which they reduce at fronts; and
3. a mechanism for weighting each of the above algorithms at each grid point and timestep.

Obviously the accuracy of a given modern front-capturing method may depend strongly on the choices made in each of its three parts. In the FCT algorithms we shall consider here, these three parts correspond to the high-order fluxes, the low-order fluxes, and the flux limiter respectively, terms we shall define shortly. Our experience is that FCT algorithms are capable of solving most problems involving fronts with both robustness and accuracy, as long as certain design principles are adhered to. Toward that end, this chapter shall present to the reader a collection of design principles that we have found to be of value in the creation of an FCT algorithm for a given situation. In general, they involve optimizing one's choice of each of the above three components of the algorithm. The reader will not be surprised to learn that a knowledge of the physics problem being addressed is an essential part of the design criteria.

In Section 2, we give a formal definition of FCT, first for the special case of one spatial dimension, and then for multidimensions. In Section 3 we give six design criteria that collectively define what we mean by a "properly designed" FCT algorithm. In Section 4 we give examples of the kind of performance one can expect from a properly designed FCT algorithm, using the scalar linear advection problem. For this problem, the "physics" that must be incorporated into the algorithm is simple and intuitive, and accurate and robust algorithms are easy to construct. In Section 5 we move on to nonlinear systems of equations, using the Euler equations as an example. Here the physics is not trivial as it was in the case of linear advection, and we find that blindly applying the methods that worked well for advection can be disappointing. However, when we transform the problem into a set of variables for which we have a legitimate set of physical constraints that can be imposed, we recover the kind of performance that we saw in the linear advection case. In Section 6 we treat both passively-driven convection and compressible gasdynamics in two space dimensions, and again have to face and solve the question of physically appropriate constraints. Finally in Section 7 we give our conclusions.

2 Flux-Corrected Transport (FCT) Defined

As we mentioned in the previous section, the great majority of methods designed to treat fronts in the context of Eq. (1), are in conservation form, i.e., a form consisting of numerical fluxes connecting adjacent grid points, these fluxes being used to advance the numerical solution in time. In FCT, at every timestep and at every flux point, these fluxes are computed twice, once using an algorithm guaranteed not to generate unphysical values (the "low order fluxes"), and once using an algorithm that is formally of high accuracy in the smooth portions of the solution (the "high order fluxes"). FCT then constructs the net fluxes for the timestep as weighted averages of these two candidate fluxes. The weighting is performed in a manner which ensures that the high order fluxes are used to the greatest extent possible without introducing unphysical values into the solution. The procedure is referred to as "flux-correction" or "flux limiting" for reasons which will become clear shortly.

From the above description, it should be clear that one may easily define an FCT algorithm on any structured or unstructured grid in any number of spatial dimensions, as long as one can define a numerical technique for which the difference between a low order time advancement operator and its higher order counterpart can be written as an array of fluxes between adjacent grid points. Rather than attempt to give a definition at that level of generality, we will give formal definitions for the cases of one spatial dimension, and for two spatial dimensions on a structured grid. From these two examples it should be clear how to construct an FCT algorithm in any number of dimensions, and on any grid, structured or unstructured.

2.1 FCT in One Spatial Dimension

In one spatial dimension, Eq. (1) takes the simpler form

$$\frac{\partial q(x,t)}{\partial t} + \frac{\partial f(q,x,t)}{\partial x} = 0 \tag{2}$$

A simple example of such a system of equations is the system describing one-dimensional ideal inviscid fluid flow, also known as the Euler equations:

$$q = \begin{pmatrix} \rho \\ \rho u \\ \rho E \end{pmatrix}; \quad f = \begin{pmatrix} \rho u \\ \rho u u + P \\ \rho u E + P u \end{pmatrix} \tag{3}$$

where ρ, u, P, and E are the fluid density, velocity, pressure, and specific total energy respectively.

We say that a discrete approximation to Eq. (2) is in conservation or "flux" form when it can be written in the form

$$q_i^{n+1} = q_i^n - \Delta x_i^{-1}[F_{i+(1/2)} - F_{i-(1/2)}] \tag{4}$$

Here q and f are defined on the spatial grid points x_i and temporal grid points t^n, and Δx_i is the cell width associated with cell i. The $F_{i+(1/2)}$ are called numerical fluxes, and are functions of f and q at one or more of the time levels t^n. The functional dependence of F on f and q *defines* the particular discrete approximation.

As mentioned above, FCT constructs the net flux $F_{i+(1/2)}$ point by point and timestep by timestep (nonlinearly) as the weighted average of two fluxes, one produced by a "high order" method and the other by a "low order" method. The formal procedure introduced in [13] is as follows:

1. Compute $F^L_{i+(1/2)}$, the "low order fluxes," using a method guaranteed not to generate unphysical values in the solution for the problem at hand.
2. Compute $F^H_{i+(1/2)}$, the "high order fluxes" using a method chosen to be accurate in smooth regions for the problem at hand.
3. *Define* the "antidiffusive fluxes" [2]

$$A_{i+(1/2)} \equiv F^H_{i+(1/2)} - F^L_{i+(1/2)} \tag{5}$$

4. Compute the time advanced low order ("transported and diffused" [2]) solution:

$$q_i^{td} = q_i^n - \Delta x_i^{-1}[F^L_{i+(1/2)} - F^L_{i-(1/2)}] \tag{6}$$

5. *Limit* the antidiffusive fluxes in a manner such that q_i^{n+1} as computed in step 6 below does not take on nonphysical values:

$$A^C_{i+(1/2)} = C_{i+(1/2)} A_{i+(1/2)}, \quad 0 \le C_{i+(1/2)} \le 1 \tag{7}$$

6. Apply the limited antidiffusive fluxes:

$$q_i^{n+1} = q_i^{td} - \Delta x_i^{-1}[A^C_{i+(1/2)} - A^C_{i-(1/2)}]$$

The critical step in the above is step 5, the flux limiting step. In the absence of step 5 (i.e., $A^C_{i+(1/2)} = A_{i+(1/2)}$), q_i^{n+1} would simply be the time-advanced high order solution.

2.2 Multidimensional Flux-Corrected Transport

Let us see how the procedure above might be implemented in multidimensions. An obvious choice would be to use an operator-splitting technique, splitting along spatial dimensions, when it can be shown that the equations allow such a technique to be used without serious error. Indeed, such a procedure may even be preferable from programming and time-step considerations. However, there are many problems for which such splitting produces

unacceptable numerical results, among which are incompressible or nearly incompressible flow fields. The technique is straightforward and shall not be discussed here. Instead, let us now consider the two-dimensional system of conservation laws

$$q(\mathbf{x},t)_t + f(q,\mathbf{x},t)_x + g(q,\mathbf{x},t)_y = 0 \tag{8}$$

A simple example of such a system of equations, and one we consider later, is the system describing two-dimensional ideal inviscid fluid flow, also known as the two-dimensional Euler equations:

$$q = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}; \quad f = \begin{pmatrix} \rho u \\ \rho uu + P \\ \rho uv \\ \rho uE + Pu \end{pmatrix}; \quad g = \begin{pmatrix} \rho v \\ \rho vu \\ \rho vv + P \\ \rho vE + Pv \end{pmatrix} \tag{9}$$

where ρ, u, v, P, and E are the fluid density, x velocity, y velocity, pressure, and specific total energy respectively.

If we work on a finite volume coordinate-aligned mesh, we can define our two-dimensional FCT algorithm thus:

$$q_{ij}^{n+1} = q_{ij}^n - \Delta V_{ij}^{-1}[F_{i+(1/2),j} - F_{i-(1/2),j} + G_{i,j+(1/2)} - G_{i,j-(1/2)}] \tag{10}$$

where ΔV_{ij} is the volume of cell ij.

Now there are two sets of transportive fluxes F and G, and the FCT algorithm proceeds as before:

1. Compute $F^L_{i+(1/2),j}$ and $G^L_{i,j+(1/2)}$, the "low order fluxes," using a method guaranteed not to generate unphysical values in the solution for the problem at hand.
2. Compute $F^H_{i+(1/2),j}$ and $G^H_{i,j+(1/2)}$, the "high order fluxes," using a method chosen to be accurate in smooth regions for the problem at hand.
3. *Define* the "antidiffusive fluxes" [2]

$$A_{i+(1/2),j} \equiv F^H_{i+(1/2),j} - F^L_{i+(1/2),j}$$

$$A_{i,j+(1/2)} \equiv G^H_{i,j+(1/2)} - G^L_{i,j+(1/2)}$$

4. Compute the time advanced low order ("transported and diffused" [2]) solution:

$$q_{ij}^{td} = q_{ij}^n - \Delta V_{ij}^{-1}[F^L_{i+(1/2),j} - F^L_{i-(1/2),j} + G^L_{i,j+(1/2)} - G^L_{i,j-(1/2)}]$$

5. *Limit* the antidiffusive fluxes in a manner such that q_{ij}^{n+1} as computed in step 6 below does not take on nonphysical values:

$$A^C_{i+(1/2),j} = C_{i+(1/2),j} A_{i+(1/2),j}, \quad 0 \le C_{i+(1/2),j} \le 1$$

$$A^C_{i,j+(1/2)} = C_{i,j+(1/2)} A_{i,j+(1/2)}, \quad 0 \le C_{i,j+(1/2)} \le 1$$

6. Apply the limited antidiffusive fluxes:

$$q_{ij}^{n+1} = q_{ij}^{td} - \Delta V_{ij}^{-1} [A^C_{i+(1/2),j} - A^C_{i-(1/2),j} + A^C_{i,j+(1/2)} - A^C_{i,j-(1/2)}]$$

As can be easily seen, implementation of FCT in multidimensions is straightforward, with the possible exception of Step 5, the flux limiter, which will be addressed in a later section.

3 Design Criteria for FCT Algorithms

Here we give, with only modest detail, six criteria that we believe are necessary for the construction of properly designed (robust but accurate) FCT algorithms. They are:

1. The resolving power of the high order fluxes should be as high as is practical. The term "resolving power" will be defined precisely below.
2. The high order fluxes should have a dissipative component which adapts itself to the resolving power of the nondissipative component.
3. The high order fluxes should be "pre-constrained" with respect to physically appropriate bounds before being input to the flux limiter.
4. The low order flux must be dissipative enough to guarantee that unphysical values in the solution cannot be generated, but should otherwise be as accurate as is practical.
5. The flux limiter should accommodate as flexible a specification of solution bounds as possible, and should utilize constraints that have a strong physical basis.
6. The flux limiter should have a simple fail-safe feature that reduces all fluxes into a grid point to their low order values when the normal flux limiter machinery fails.

We will treat each one of these in turn.

3.1 High Order Fluxes with Very High Resolving Power

The primary result of [14] was that there is a significant advantage in using fluxes derived from very high order, spatially-centered finite difference

operators (fourth order or higher) for the "high order fluxes" in FCT algorithms. That conclusion was based on some analysis showing a strong empirical relationship between order and resolving power (a term which we define below) for centered finite difference operators, some heuristic reasoning as to how FCT works in practice, and several one-dimensional test calculations dominated by linear advection test problems. We review that work in this section.

For analysis, Eq.(2) is usually reduced to a scalar conservation law and linearized:

$$\frac{\partial q}{\partial t} + u\frac{\partial q}{\partial x} = 0 \tag{11}$$

where u is a constant. This is the linear advection equation with advection speed u. Its solution is simply

$$q(x,t) = q(x - ut, 0) \tag{12}$$

That is, the profile is simply translated right or left with velocity u and no change in shape. In Fourier space, this takes the form of each Fourier mode moving with a phase velocity u, with no change in amplitude. Thus all numerical errors associated with a given numerical algorithm can be quantified by a specification of phase velocity error per timestep and amplitude error per timestep as a function of the wavenumber k, and as a function of the discretization step in space and time Δx and Δt respectively.

In [14] we examined a particular algorithm for solving Eq. (11) on a uniform mesh, that of using a leapfrog discretization in time and centered finite differences of arbitrary order in space. This choice allowed us to ignore the amplitude errors entirely, since for the leapfrog discretization these errors vanish for all k and for all Δx and Δt satisfying the Courant condition $\epsilon \equiv |u|\Delta t/\Delta x < 1$. Further noting that the total phase error was the algebraic sum of that induced by the temporal and spatial discretizations separately, as long as both were small, we were able to reduce our algorithm analysis to a single plot, reproduced here in Fig. 1.

The striking aspect of Fig. 1 is the marked effect of the order N of the centered spatial finite difference operator, as well as the timestep Δt, on the resolving power of the algorithm. By "resolving power" we mean, loosely, the ability of an algorithm to maintain low phase errors over a large part of k-space. In more precise terms, we define the resolving power $k_r(E)$ of a given combination of N and Δt to be the largest wavenumber k for which all phase errors are smaller than a pre-specified value E. Thus Fig. 1 tells us that for a given E, we can resolve more and more of k-space if, on a given grid, we simply increase the spatial order N of our centered finite difference operator, decreasing Δt appropriately as we do so. The primary conclusion of [14] was that not only is this statement true for smooth functions, but that it is true in the presence

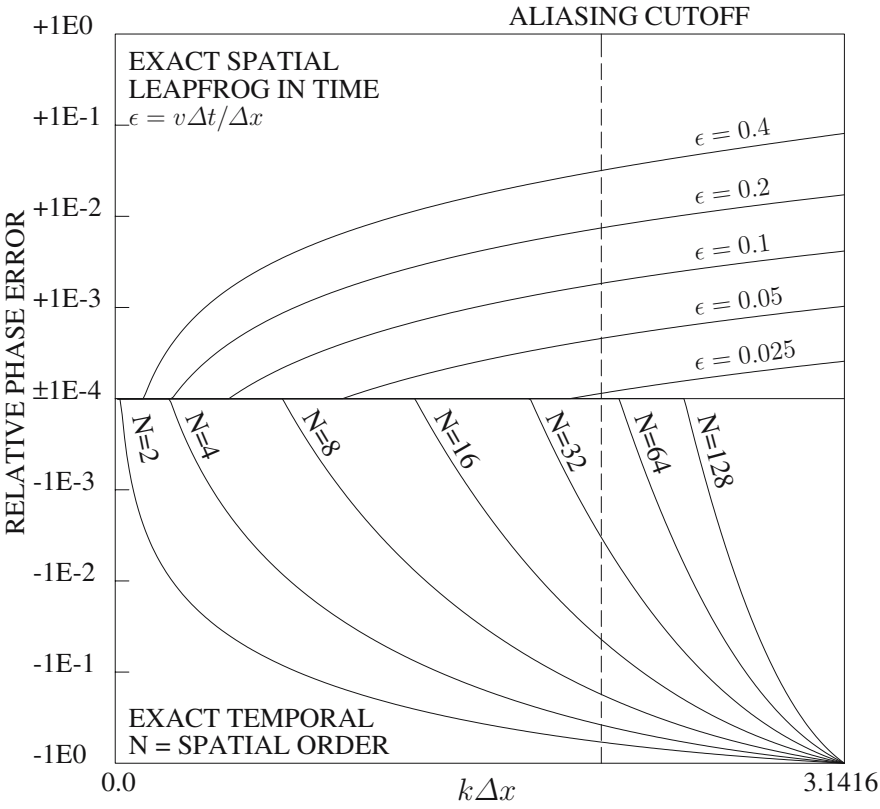


Fig. 1. Plot of relative phase error versus the normalized wavenumber $k\Delta x$ for the leapfrog time-marching scheme and analytic spatial derivatives (top), and for centered finite difference schemes of order N and analytic temporal derivatives (bottom). Figure taken from Ref. [14]

of fronts also, as long as one is treating the fronts with a front-capturing algorithm such as FCT. Nothing since 1981 has dissuaded us from that view, and thus we present it as the first of our FCT design criteria. Computational examples presented later in this chapter will hopefully provide the reader more evidence of its correctness.

3.2 High Order Fluxes with an Adaptive Dissipation Component

Looking at the bottom portion of Fig. 1, we see that the arguments of the last section become less and less convincing as one moves to the extreme right portion of the plot. Phase error in this portion of the plot becomes increasing resistant to reduction by simply increasing N. Indeed, at the Nyquist frequency $k\Delta x = \pi$, the phase error is -100 % regardless of how large we

make N or how small we make Δt. Thus for any given N, there will be some portion of k-space which is resolved poorly. Logic dictates that these Fourier modes be damped rather than carried with an erroneous speed, and that any given Fourier mode be damped nearly entirely by the time it is 180 degrees out of phase with the analytic result. Clearly, from Fig. 1, the functional dependence of this dissipation on k must itself depend on N if we are to achieve this result without damping the modes which are actually being carried accurately.

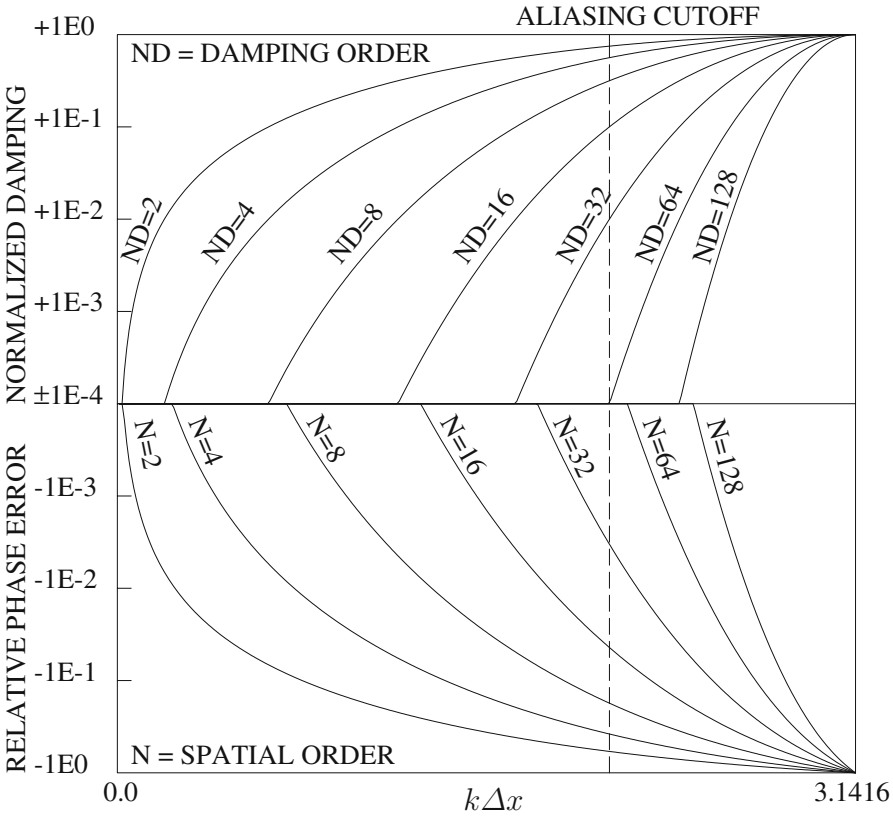


Fig. 2. Top: Plot of the damping induced by a centered finite difference approximation to the N_Dth derivative versus the normalized wavenumber $k\Delta x$, normalized so that the Nyquist mode is completely eliminated. Bottom: Same as bottom of Fig. 1: Plot of relative phase error versus the normalized wavenumber $k\Delta x$ for centered finite difference schemes of order N and analytic temporal derivatives

We have found that one can form such dissipative fluxes from the centered finite difference forms of $d^{N_D}q/dx^{N_D}$ where $N_D \leq N+2$. One then simply adds these fluxes to FCT's "high order" fluxes. Indeed, the early work of Kreiss and Oliger [9] contained calculations of the linear advection of a triangular wave using $N_D = N = 4$ and leapfrog time differencing, with results that, although not oscillation free, were considerably better than without the dissipation. These results are what prompted us to use such forms to construct FCT algorithms, and we have found them to be of sufficient value to include them among our design criteria.

In the top portion of Fig. 2 we plot the damping induced by centered finite difference approximations to $d^{N_D}q/dx^{N_D}$ versus $k\Delta x$, normalized so that the Nyquist mode is completely eliminated. In the bottom portion, we simply reproduce the bottom portion of Fig. 1. Focusing our attention on pairs of lines for which $N = N_D$, we see a rather remarkable match between phase error amplitude and dissipation amplitude as a function of $k\Delta x$. That is, for such pairs, as the relative phase error increases, so does the damping, with approximately the same functional dependence on $k\Delta x$. This is in accordance with our expressed desire to induce damping in proportion to the relative phase error. Thus, for the computational examples in this chapter, we have chosen to use $N_D = N$. An equally good case can be made for the choice $N_D = N + 2$, since this will leave the overall order of the algorithm intact. Indeed we have made that choice ourselves in some contexts. The specific construction of these operators in flux form is given in the appropriate later sections.

3.3 Imposition of Physically-Motivated Constraints on the High Order Fluxes Before the Flux Limiting Step

In general, the numerical algorithms used to construct high order numerical fluxes from the cell-centered values of q assume by necessity a degree of smoothness to q. Thus near fronts it is not unusual, especially for spatial orders higher than 2, to find that these fluxes violate physically-motivated bounds on their values. One could, of course, take the position that the flux limiter itself will ensure that these fluxes are prevented from producing values of q in the next time step that violate appropriate bounds for q, and thus that these unphysical values for the high order fluxes should be allowed to stand when computing the antidiffusive fluxes A. Nonetheless, it is wise to attempt to address this problem at its source, rather than shift the burden to the flux limiter at a later stage, and simply not allow the high order fluxes to take on values that are clearly outside the bounds of possibility. This design principle is not truly new, although not generally expressed as we have above. The prime example is the default behavior of the original Boris-Book flux limiter [2], which the reader will meet shortly. As explained in [13], this flux limiter sets the antidiffusive flux A to zero in virtually all cases where the antidiffusive

flux has the same direction as the gradient of q^{td}, i.e., where A is actually diffusive, and in most cases would not actually cause the adjacent values of q to take on unphysical values. Although it is difficult to make rigorous statements in this context, in the great majority of the cases for which this flux-canceling machinery is active, and for which one can place physically-motivated upper and lower bounds on the value of the flux, the high order flux itself can be shown to be outside those bounds, without resort to arguments about its effect of the subsequent values of q. Another example of constraints imposed on the high order fluxes prior to the primary monotonicity machinery is to be found in the PPM algorithm [5], wherein candidate point values of q at cell interfaces are computed, given its cell averages. Rather than simply calculate these "high order" point values of q in a straightforward way, Colella and Woodward use a multistep algorithm that utilizes MUSCL slope limiting in a way that guarantees that the candidate "high order" interface value of q is bounded by the corresponding cell averages at the adjacent grid points. The motivation and the effects of this "pre-limiting" is similar to that of the Boris-Book limiter.

If one can place rigorous physically-motivated bounds on the high order flux, this step can be quite simple, as well as quite effective. We will see an example of such a situation when we construct a non-clipping flux limiter for advection in the next section. However, in general it is difficult to find such rigorous bounds, for at least two reasons. The first is that the flux $\mathbf{f}$ lives in a space one dimension lower than the corresponding q. For example, in three spatial dimensions in a finite volume context, q represents a volume average over the cell while the fluxes $\mathbf{f}$ are area averages at cell faces. The second is that in general $\mathbf{f}$ is a nonlinear function of q. These two combine to make it quite difficult to reliably place bounds on the high order fluxes. Thus we often use the reasoning implicit in the Boris-Book limiter: if the antidiffusive flux is actually diffusive, then there must be something wrong with the high order flux, and we set that antidiffusive flux to zero before limiting. We are not happy with the lack of rigor associated with that choice, but empirically we have found that this is the correct action to take in most cases.

3.4 Low Order Fluxes That Guarantee That Physical Bounds are not Violated

The prime requirement of FCT's low order flux is that it be guaranteed to produce a solution free of unphysical grid point values. If this cannot be guaranteed, then there is no hope of guaranteeing that property in the final FCT solution, even with failsafe limiting (item 6). One way to satisfy this requirement is to simply incorporate enough numerical dissipation in the algorithm, but overkill here is to be avoided (see below). Another way is to use first-order upwind methods or kinetics-based methods such as the beam scheme of Sanders and Prendergast. When properly chosen, these are probably the best

choice, for their inherent dissipation is usually close to the minimum required to meet the prime requirement. However, be aware that many such schemes using "approximate" Riemann solvers cannot meet the prime requirement. It should be obvious that, within the prime requirement, the low order algorithm should be as accurate as is practical. In particular, any low order flux with more dissipation than is necessary to meet the prime requirement is harmful, putting an extra burden on the flux limiter, which is arguably the weakest link of the algorithm. Thus, for example, one would not use a Lax-Friedrichs flux for an advection problem, since the less diffusive donor cell flux already satisfies the prime requirement. As another example, if we are solving a two-dimensional advection problem, and we must choose between two donor cell algorithms, the first of which allows corner transport, and the second of which does not, we would choose the former as long as it satisfied the prime requirement.

3.5 Flexible Flux Limiters That Utilize Constraints with a Strong Physical Basis

It is useful to consider a flux limiter as being comprised of two components:

1. A physics component which specifies physically-motivated upper and lower bounds on grid point values in the next time step; and
2. An algorithmic machinery component for enforcing the above bounds.

It is clear that the algorithmic machinery component must have sufficient flexibility to accommodate the needs of the physics component. Thus, in our view, a good flux limiter must possess both a robust and accurate physics component and a flexible algorithmic machinery component. There exist several flux limiters which are of extremely simple form, expressible in as little as one line of Fortran. The original flux limiter of Boris and Book, and the ones typically used in algorithms that describe themselves as "TVD" are prime examples. The simplicity of these flux limiters is due to an extremely simple physics component and a very inflexible algorithmic machinery component: they make rather strong assumptions about what constitutes proper upper and lower bounds on the solution at a given grid point at a given timestep, and their algorithmic machinery is capable of accommodating those strong assumptions and little more. These assumptions may be overly restrictive, as they are in the case of the "clipping" phenomenon, or overly loose, as in the case of the "terracing" phenomenon. Thus we strongly encourage the reader to derive the above flux limiters for himself or herself. This exercise will reveal the assumptions implicit in these limiters, and allow an assessment as to their appropriateness for the problem at hand. If they are not appropriate, the reader may wish to consider a more flexible flux limiter, albeit one probably more complex and longer that one line of Fortran. We give an example of a more flexible flux limiter later. In our view the most difficult issue in

the design of flux limiters is the physics component. Most commonly the antidiffusive fluxes are those which update conservative variables directly, and hence the default choice for most FCT algorithms has been to constrain the conservative variables using the default bounds built into the Boris-Book flux limiter. But this can often be a bad choice. Even if we were to circumvent the default bounds by using more flexible algorithmic machinery, it can often be extremely difficult to determine the appropriate bounds for the conserved variables. Using gasdynamics as an example, one would be hard pressed to specify the physically appropriate bounds on mass, momentum, and energy per unit volume at a given grid point at a given timestep, even if he or she were given the complete time history of the computed solution up to that point. We know that the physics allows the formation of new extrema in all three of these quantities. Thus looking at the adjacent grid point values at the previous time step, or even at the values of the time-advanced low order solution (FCT's default), can lead to bounds on the solution that are not physically appropriate. When we discuss the construction of FCT algorithms for gasdynamics in Section 5, we will put forth the hypothesis that much more reliable constraints are to be obtained by performing the flux limiting step in characteristic variables rather than conservative variables.

3.6 Failsafe Flux Limiters

We suppose that this topic comes under the general heading of "dirty laundry," but it cannot be ignored in any objective discussion of front-capturing algorithms. If one is attempting to solve difficult problems, our experience is that, no matter how carefully one tries to design algorithms that are consistent both with numerical analysis and with the physics problem one is attempting to solve, there will be situations in which at least one grid point at at least one time step takes on values that are outside the bounds of physical possibility. For FCT and similar algorithms, these will usually be variables that one is not directly constraining. For example, if one is performing flux limiting on the conserved variables (not recommended here, but done often by many users of FCT), the density by construction can never become negative, but the internal energy can do so, since it is not directly constrained by the limiting process. What should one do in this case? As another example that we ourselves will face later in this chapter in the Woodward-Colella double shock tube problem, even when we limit with respect to what in our view are the most physically appropriate variables, the characteristic variables, we can occasionally generate unphysical grid point values. The characteristic variables are, after all, a linearization, and a linearization can be inaccurate at large jumps. Thus we could generate negative internal energies in that case also (or, in theory, even negative densities!). What action should we take when this happens?

Prudence, if nothing else, would dictate that the algorithm make provisions for such a case. Assuming we do not wish to simply terminate the calculation,

we desire a solution which is as consistent with the design philosophy of the algorithm as possible and, most important of all, is explicitly stated.

For FCT, we believe that there is an obvious solution consistent with FCT's design, and with the numerical analysis goal of being at least first order accurate, and that is the one we choose here: For any such offending grid point, we iteratively drive all the fluxes into or out of that grid point toward their low order values, until the offense is eliminated. Thus it is especially important that FCT's low order scheme be guaranteed to be free of unphysical values! We use an especially simple algorithm here, which we describe in a later section.

4 FCT Algorithms for One Dimensional Linear Advection

We wish to give the reader an idea of the kind of performance one can expect of an FCT algorithm for the simplest of scalar conservation laws, linear advection. That is, we have Eq. (2) with $f = qu$ and u a constant. We use a uniform spatial mesh of cell size Δx. We utilize a method of lines approach, choosing our spatial and temporal discretization independently. All temporal discretizations we shall use (e.g., modified Euler, explicit Runge-Kutta, leapfrog, leapfrog-trapezoidal) involve one or more leapfrog-like substeps of the following form:

$$q_i^{n+1} = q_i^n - \Delta x_i^{-1}[F_{i+(1/2)} - F_{i-(1/2)}] \tag{13}$$

Here t^{n+1} and t^n are substep time levels associated with a particular substep, with associated timestep $\Delta t^{n+1/2}$. The fluxes F are functions of f at one or more of the time levels, not necessarily t^{n+1} and t^n. The timestep $\Delta t^{n+1/2}$ has been absorbed into the definition of the fluxes. This leapfrog-like substep will be used as the fundamental building block for any time discretization we use. Thus we can describe our treatment for all temporal discretizations by describing our treatment of this substep.

Our low order flux for advection is given by the first order upwind scheme:

$$F_{i+(1/2)}^L = \left[\frac{1}{2}(f_{i+1}^n + f_i^n) - \frac{1}{2}|u|(q_{i+1}^n - q_i^n)\right] \Delta t^{n+1/2} \tag{14}$$

The high order fluxes are given by the formulae in the Appendix of [13]. As an example, the fourth order flux is given by:

$$F_{i+(1/2)}^{H4} = \left[\frac{7}{12}(f_{i+1}^a + f_i^a) - \frac{1}{12}(f_{i+2}^a + f_{i-1}^a)\right] \Delta t^{n+1/2} \tag{15}$$

where the time level t^a is meant to denote whatever time level or average of time levels is required by the particular substep of the particular time discretization chosen.

The high order dissipative fluxes of order N_D, which are added to the above high order fluxes, are simply the flux form representation of $\partial^{N_D} q/\partial x^{N_D}$, normalized to damp the Nyquist mode completely in one timestep at a Courant number of unity. As an example, the order 4 dissipative flux is given by:

$$F^{D4}_{i+(1/2)} = -|u| \left[\frac{3}{16}(q^n_{i+1} - q^n_i) - \frac{1}{16}(q^n_{i+2} - q^n_{i-1}) \right] \Delta t^{n+1/2} \tag{16}$$

Thus far we have dealt with only three of our six FCT design criteria, the design of the high and low order fluxes. The other three are the pre-constraint of the high order fluxes, the construction of the flux limiter, and the failsafe limiter. A failsafe limiter is not needed here, since we are directly constraining the only variable of interest. For the moment, we will choose a simple default for the remaining two criteria, the original Boris-Book limiter:

$$A^C_{i+(1/2)} = S \max\left(0, \min(|A_{i+(1/2)}|, S(q^{td}_{i+2} - q^{td}_{i+1})\Delta x, S(q^{td}_i - q^{td}_{i-1})\Delta x)\right)$$
$$\text{where} \quad S \equiv \text{sign}(1, A_{i+(1/2)}). \tag{17}$$

This simple formula implicitly determines our choices for the remaining two design criteria. These choices turn out to be reasonable for this advection problem, at least away from extrema. However, as we will shortly see, we can improve on FCT's performance at extrema by addressing these remaining two criteria explicitly.

All of the tests in this section use the classic explicit fourth order Runge Kutta time discretization, each substep of which is treated in the manner described above.

4.1 Tests of FCT Advection Algorithms on Three Classic Test Problems

In [15] we compared a number of advection algorithms on three test problems chosen from the open literature: the square wave test of Boris and Book [2], the Gaussian of Forester [7], and the semi-ellipse of McDonald [10]. The first test consists of a square wave 20 cells wide to be advected 800 time steps at a Courant number of 0.2. The second test consists of a Gaussian of half width 2 cells to be advected 600 time steps at a Courant number of 0.1. The third and final test consists of a semi-ellipse of radius 15 cells to be advected 600 time steps at a Courant number of 0.1. We will use those same test problems here to demonstrate various aspects of the FCT algorithms we have just described.

In Fig. 3, we examine the effect that we had observed in our earlier work [14]. Running the square wave problem, we vary only the order of the high order flux

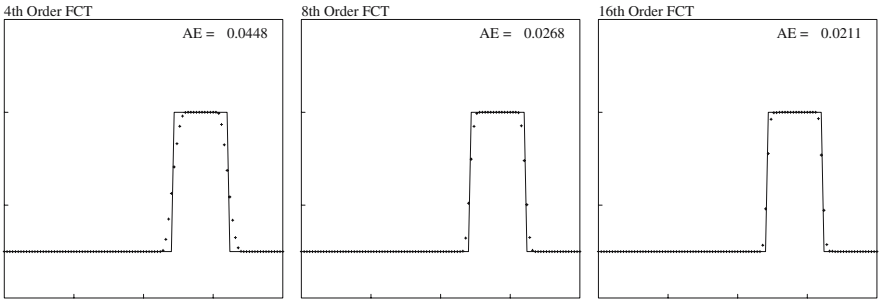


Fig. 3. Results for the Boris-Book square wave using FCT algorithms with high order fluxes of fourth, eighth, and sixteenth order. The analytic solution is shown as a solid line, while the computed solution is shown as discrete data points. The L_1 error is denoted "AE" in the plots, for consistency with the original plots of Boris and Book. Note the marked improvement with resolving power

from fourth to eighth to sixteenth, and see a marked increase in the resolution of the discontinuities. Our interpretation of this effect was, and continues to be, that since the discontinuity is the result of the superposition of a large number of Fourier modes, with precise phase relationships being critical, increasing the resolving power, i.e., the percentage of k-space for which the phase speed is accurate, makes it possible for the flux limiter to introduce less and less dissipation to prevent unphysical values, thus yielding more accurate results.

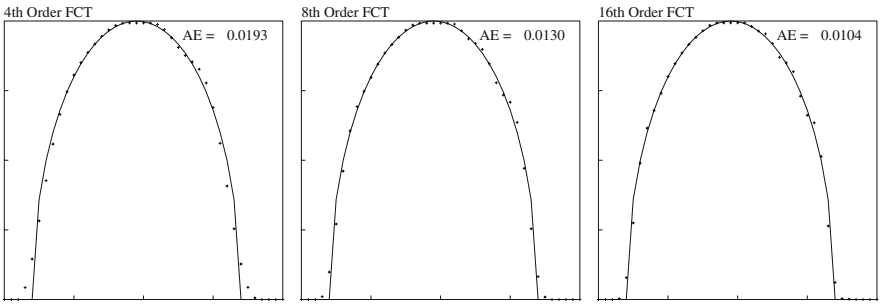


Fig. 4. Results for the semi-ellipse of McDonald using FCT algorithms with high order fluxes of fourth, eighth, and sixteenth order. The analytic solution is shown as a solid line, while the computed solution is shown as discrete data points. The L_1 error is denoted "AE" in the plots. Note the improvement, albeit modest, with increased resolving power. Although mitigated significantly by the high order dissipation, clear hints of the terracing phenomenon are still visible. Compare to Fig. 5

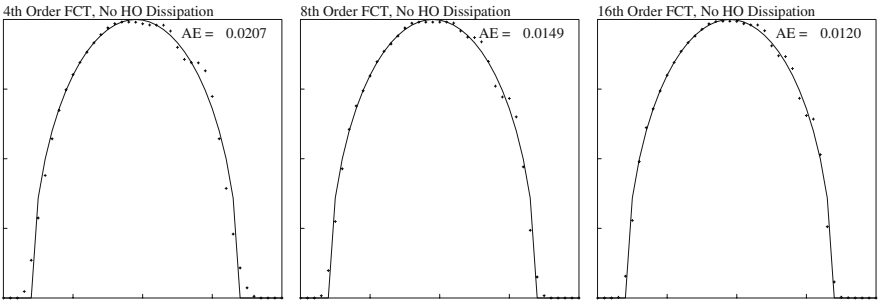


Fig. 5. Same as Figure 4, but with the dissipative component of the high order fluxes removed. Note the "terracing" phenomenon on the right edge of the semi-ellipse, which is the result of dispersive waves being ignored by the flux limiter until they attempt to become extrema. Compare to Fig. 4

In Fig. 4, we show the same sequence of algorithms, but for the semi-ellipse of McDonald. Again we see an increase of performance with resolving power. However, this problem is prone to the "terracing" phenomenon, some hints of which can be seen at the right edge of the semi-ellipse. To show the value of the dissipative component of the high order flux, we show the same problem with the same set of algorithms in Fig. 5, but with the dissipative component eliminated. Although not as dramatic as the effect of increasing the resolving power of the high order flux, the dissipative flux clearly is of value in preventing the occurrence of errors that are not detected by the flux limiter. What is happening here is that dispersive oscillations are being shed by the leading (right) edge of the semi-ellipse. As they propagate into the semi-ellipse they are not detected as oscillations because they are hidden by the large gradient in the right side of the ellipse. As they get closer to the center of the ellipse, they try to take the form of true extrema, at which point they are prevented from doing so by the flux limiter. The damage is already done, however. The effect of the dissipative component in the high order flux is to damp the modes moving with the wrong phase velocity before the fact. Calculations like these, as well as analytic arguments, are the reason we believe that some high order dissipation should be present in most calculations, whether one is using front capturing techniques or not.

The previous set of calculations was an example of one way a flux limiter can fail. In that case, the flux limiter failed to perceive and prevent an error because its definition of an error was the creation of new extrema in q. Note that the algorithmic machinery component of the flux limiter did not fail, but rather its physics component. In this case its "physics" criterion for what constituted an error was too weak. In the next set of calculations, we see an example where exactly the same criterion is too strong, preventing the formation of an extremum when it is physically allowable. In Fig. 6 we show the

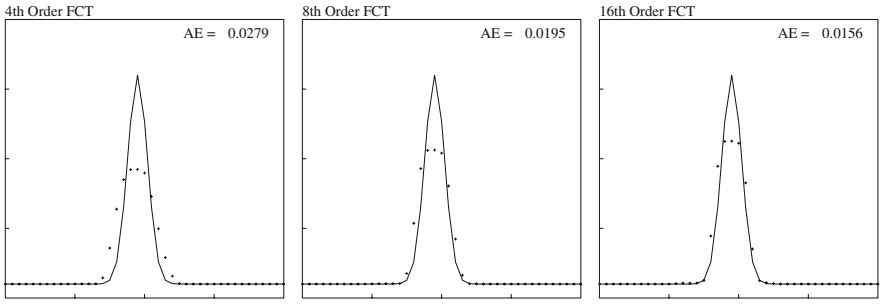


Fig. 6. Results for the Gaussian of Forester using FCT algorithms with high order fluxes of fourth, eighth, and sixteenth order. The analytic solution is shown as a solid line, while the computed solution is shown as discrete data points. The L_1 error is denoted "AE" in the plots. Again we see improvement with resolving power, but the errors are dominated by the "clipping" phenomenon

same sequence of algorithms, but for the Gaussian of Forester. Although we again see the same pattern of increased performance with increased resolving power, we also see the well-known "clipping" problem. Here, as the true peak of the Gaussian passes between grid point centers, the true grid point extrema value should increase and decrease in an oscillatory fashion. However, the flux limiter used here does not allow for that possibility, treating all attempts to accentuate an extremum during a time step as an error to be prevented. The problem can be addressed by using a more flexible limiter and a better estimate of the allowable upper and lower bounds on the solution, as we show in the next subsection.

4.2 An Alternative to the Boris-Book Flux Limiter

In [13], we described a new flux-limiting algorithm for FCT. Although developed primarily to allow the construction of fully multidimensional FCT algorithms, that flux limiter also allowed a much more flexible specification of upper and lower bounds on the solution than did the original Boris-Book limiter Eq. (17). In particular, it allowed the construction of flux limiters which do not clip physical extrema. We describe that algorithm in one spatial dimension in this section, and then use it to construct a non-clipping flux limiter for one dimensional advection. In Section 5 we describe and use the algorithm in two spatial dimensions.

In words, the alternative flux limiter constrains the solution by first computing two independent sets of provisional coefficients $C_{i+(1/2)}$ for each antidiffusive flux, one to enforce the user-supplied upper bounds on the solution, and the other to enforce the user-supplied lower bounds. Both bounds are satisfied simply by choosing the final coefficients to be the minimum of the two provisional coefficients.

The upper bounds constraint is computed by dividing Q_i^+, the maximum allowable net flux into a cell, by P_i^+, the sum of all those fluxes whose effect is to increase the value of q_i. That fraction, bounded by 0 and 1, is provisionally assigned to the $C_{i+(1/2)}$ of each of those fluxes. A similar procedure is undertaken for the lower bounds constraint, and still another provisional value of $C_{i+(1/2)}$ assigned to each of the fluxes whose effect is to decrease the value of q_i. The net $C_{i+(1/2)}$ is simply the minimum of the two temporary values. From the above description it should be clear that this limiter is unambiguously defined for any number of spatial dimensions and for both structured and unstructured meshes, as long as the difference between the low and high order components can be written as fluxes flowing between adjacent cells. In one spatial dimension, the procedure is as follows:

1. Compute, for each grid point i, physically-motivated upper and lower bounds on the solution in the next timestep, q_i^{max} and q_i^{min} respectively. This step is both flexible and critical, requiring intimate knowledge of the science underlying one's equation. It is important that q_i^{td} already satisfy these bounds.
2. For the upper bound, compute P, Q, and their ratio R at each grid point:

$$P_i^+ = \max(A_{i-(1/2)}, 0) - \min(A_{i+(1/2)}, 0) \tag{18}$$
$$Q_i^+ = (q_i^{max} - q_i^{td})\Delta x_i \tag{19}$$
$$R_i^+ = \min(1, Q_i^+/P_i^+),\ P_i^+ > 0, \quad 0 \ \text{otherwise} \tag{20}$$

3. For the lower bound, compute P, Q, and their ratio R at each grid point:

$$P_i^- = \max(A_{i+(1/2)}, 0) - \min(A_{i-(1/2)}, 0) \tag{21}$$
$$Q_i^- = (q_i^{td} - q_i^{min})\Delta x_i \tag{22}$$
$$R_i^- = \min(1, Q_i^-/P_i^-),\ P_i^- > 0, \quad 0 \ \text{otherwise} \tag{23}$$

4. Compute $C_{i+(1/2)}$ by taking a minimum:

$$C_{i+(1/2)} = \begin{cases} \min(R_{i+1}^+, R_i^-) \text{ when } A_{i+(1/2)} > 0, \\ \min(R_i^+, R_{i+1}^-) \text{ when } A_{i+(1/2)} \leq 0. \end{cases} \tag{24}$$

Note that in the above we do not specify the equivalent of Eq. (14) in [13], which, as we explained in the previous section, can be thought of as a method for pre-constraining the high order fluxes prior to the flux limiting step. In the case of linear advection in one dimension, we have a much more robust way of pre-limiting those fluxes, as we will see below.

4.3 A Non-clipping Version of the Alternative Flux Limiter

Let us now specify our non-clipping flux limiter for one-dimensional linear advection. To do so we need to define an algorithm for computing q_i^{min} and q_i^{max}

above. We also need to address our third criterion and specify an algorithm for pre-constraining our high order fluxes. We shall use a similar approach for both. In Fig. 7, we show a technique we shall use to reconstruct extrema between grid points, for use both in specifying q_i^{min} and q_i^{max} and in pre-constraining our high order fluxes. On each interval $[x_i, x_{i+1}]$ we define $q_{i+(1/2)}^{peak}$ to be the value of q at the intersection of the lines formed by connecting the point (x_{i-1}, q_{i-1}) with (x_i, q_i) and the point (x_{i+1}, q_{i+1}) with (x_{i+2}, q_{i+2}). If the x coordinate of this intersection lies between x_i and x_{i+1}, then we consider this $q_{i+(1/2)}^{peak}$ to be a physically legitimate value for q on the interval $[x_i, x_{i+1}]$.

Let us now define the upper and lower bounds for q on the interval $[x_i, x_{i+1}]$ to be

$$q_{i+(1/2)}^{max} = \max(q_i, q_{i+1}, q_{i+(1/2)}^{peak}) \tag{25}$$

$$q_{i+(1/2)}^{min} = \min(q_i, q_{i+1}, q_{i+(1/2)}^{peak}) \tag{26}$$

where all quantities are evaluated at time level n.

We are now in a position to introduce the physics of the problem into the flux limiter. Given that this is an advection problem, and that we choose our

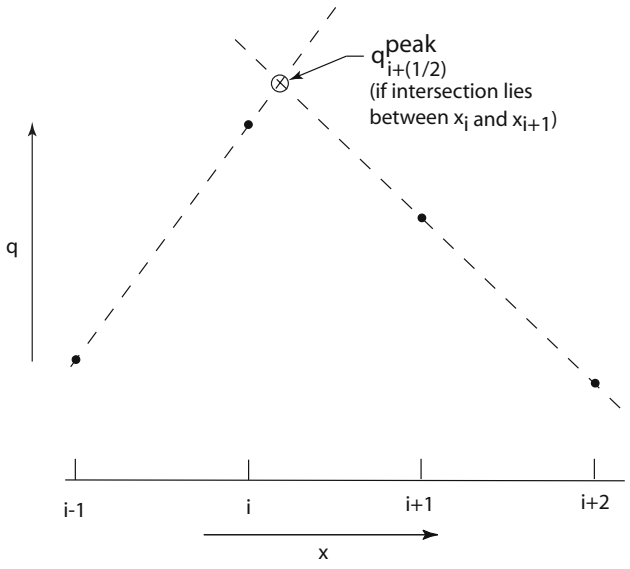


Fig. 7. A possible scheme for extracting information about extrema which exist between grid points at a given point in time. An extremum is assumed to exist between grid points i and $i+1$ if the intersection of the right and left extrapolation of q has an x coordinate between x_i and x_{i+1}. The q coordinate of this intersection in then used both to compute q^{max} and q^{min}, and to pre-constrain the high order flux $F_{i+(1/2)}^H$ (see text)

Courant number to be less than unity, we know that q_i^{n+1} must be bounded by $q_{i-(1/2)}^{min}$ and $q_{i-(1/2)}^{max}$ if $u > 0$, and by $q_{i+(1/2)}^{min}$ and $q_{i+(1/2)}^{max}$ if $u < 0$. Thus we take

$$q_i^{min} = \begin{cases} \min(q_i^{td}, q_{i-(1/2)}^{min}) \text{ when } u > 0, \\ \min(q_i^{td}, q_{i+(1/2)}^{min}) \text{ when } u \leq 0. \end{cases} \tag{27}$$

$$q_i^{max} = \begin{cases} \max(q_i^{td}, q_{i-(1/2)}^{max}) \text{ when } u > 0, \\ \max(q_i^{td}, q_{i+(1/2)}^{max}) \text{ when } u \leq 0. \end{cases} \tag{28}$$

Finally we specify our pre-constraint condition on the high order fluxes. Again we use the physics of the problem. Since this is advection, the physical fluxes $F_{i+(1/2)}$ must be bounded by $uq_{i+(1/2)}^{max}$ and $uq_{i+(1/2)}^{min}$. Thus we define $F_{i+(1/2)}^{max} \equiv \max(uq_{i+(1/2)}^{max}, uq_{i+(1/2)}^{min})$ and $F_{i+(1/2)}^{min} \equiv \min(uq_{i+(1/2)}^{max}, uq_{i+(1/2)}^{min})$, and after computing the unconstrained high order fluxes $F_{i+(1/2)}^{H}$, we constrain them thus:

$$F_{i+(1/2)}^{H} = \min(F_{i+(1/2)}^{max}, \max(F_{i+(1/2)}^{min}, F_{i+(1/2)}^{H})). \tag{29}$$

The results of using the above pre-constraint condition and flux limiter are shown in Fig. 8. For sufficiently high resolving power, the clipping phenomenon has been virtually eliminated. We believe that this demonstrates the advantage of using one's knowledge of the physics of the problem to design FCT and other front-capturing algorithms, rather than accepting their default behavior.

Before leaving this section, let us try to design the "ultimate" high order FCT scheme, and see how it performs on the three test problems we have examined in this section. It was shown by Fornberg that the asymptotic limit of an Nth order finite difference scheme on a periodic domain as N goes to infinity is in fact just the pseudospectral approximation using Fourier modes as basis functions. Thus we will take our high order fluxes to be those which reproduce the pseudospectral discretization. The results of using these pseudospectral fluxes and the non-clipping limiter and pre-constraint algorithm described above are shown in Fig. 9. These are the best results we have been able to produce for these problems using front-capturing algorithms whose high-order components are stable. (Using unstable schemes as the high order component of front-capturing algorithms can produce extremely sharp square waves, but can severely distort the Gaussian and semi-ellipse. Examples are Superbee, Ultrabee, ACM, and the contact detection algorithm in PPM.)

The above examples provide a springboard for the next section, where we address a nonlinear system of equations, and the construction of our FCT algorithms will become more complex. The path we will choose to success will be the same, however: We will incorporate as much knowledge of the physics as possible into the design of the algorithm.

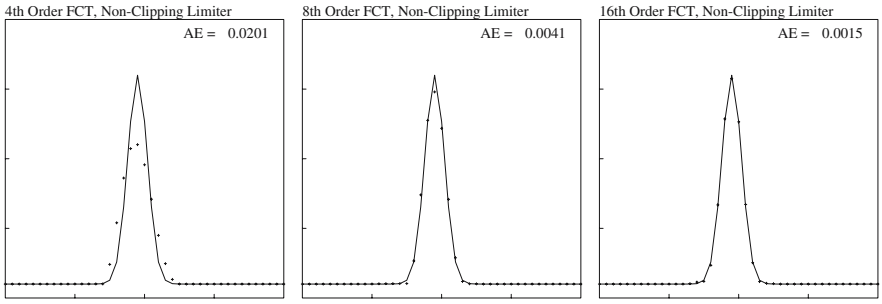


Fig. 8. Same as Figure 6, but using the non-clipping flux limiter and the preconstraint of the high order fluxes described in the text. Note the marked increase of accuracy with increased resolving power. Also note that the clipping can be virtually eliminated as long as one has sufficient resolving power

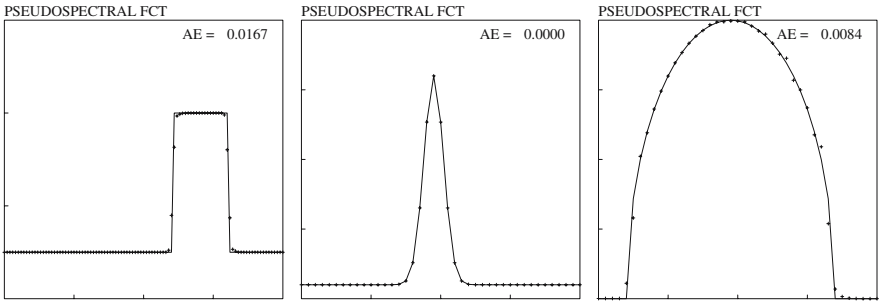


Fig. 9. Performance of a pseudospectral FCT algorithm on all three of the test problems used in this section. We have used the non-clipping flux limiter and the preconstraint of the high order fluxes described in the text. These are the best results we have been able to produce for these problems using front-capturing algorithms whose high-order components are stable

5 FCT Algorithms for One Dimensional Nonlinear Systems of Conservation Laws

We now consider Eq.(2), where f is a fully nonlinear function of q. The example we shall use is that of the Euler equations (3). If we go down our list of FCT design criteria, we find that many of the optimal choices are the same as those for linear advection, or are are obvious generalizations thereof. However, the proper construction of the flux limiter, both with regard to its physics component and with regard to its algorithmic machinery, is less than

obvious. Hence we will focus primarily on the construction of the flux limiter in this section.

We saw in the last section that the results of an FCT calculation can be sensitive to the choice of the flux limiter. But we also learned that modest modifications of the basic FCT machinery allowed us to produce results which are quite good. Thus it is worth looking at both the advection equation and at the advection flux limiters for the purpose of determining how we wish to proceed for systems of hyperbolic conservation laws.

The simplicity of the advection equation allows us to make some rather strong statements about the allowable bounds for q in the next timestep. In particular, we know that for any Courant number less than unity, the value of q_i^{n+1} is bounded by the values of q^n on the interval $[x_{i-1},x_{i+1}]$. This fact allows us to construct reasonably precise upper and lower bounds on the solution. The bounds for q_i^{n+1} used by the Boris and Book flux limiter Eq. (17) are simply the maximum and minimum of $(q_{i-1}^{td}, q_i^{td}, q_{i+1}^{td})$ respectively. While this could certainly be refined, we saw in the last section that this algorithm produces reasonable results if one is willing to tolerate the clipping of extrema. The reason for this, we believe, is that the built-it physics component of the limiter is reasonably close to one physically appropriate to the advection problem, again except near extrema.

5.1 Hyperbolic Systems of Conservation Laws: The Case For Characteristic Variables

Let us now consider systems of hyperbolic conservation laws, using the Euler equations as an example. If we choose to deal completely with the conserved variables q, what sort of statements can we make about upper and lower bounds on q in the next timestep? In contrast to the case for advection, we are at a loss. In fact we know for certain that q_i^{n+1} is not necessarily bounded by q^n anywhere in the vicinity of grid point i. The default route taken by most FCT algorithms is to simply use what worked for advection, and take the upper and lower bounds for q^{n+1} to be the maximum and minimum of $(q_{i-1}^{td}, q_i^{td}, q_{i+1}^{td})$ respectively. While certainly better than the disastrous choice of using the maximum and minimum of $(q_{i-1}^{n}, q_i^{n}, q_{i+1}^{n})$, in contrast to the case for advection, brand new extrema will be a common occurrence. The default choice would allow strong suppression of these new extrema by the combination of the flux limiter and the dissipation in the low order fluxes. Using the non-clipping flux limiter described in the last section would not help either, again because we have no basis by which to determine the magnitude that a new extremum could attain before being declared unphysical. These difficulties are reflected in the difficulty that FCT has, in our experience, in attaining the same kind of clean results for systems of equations that are relatively easy to come by for scalar equations, when using the default strategy

of flux limiting with respect to the conserved variables directly. This is because we have strong and simple statements that we can make about the upper and lower bounds in q in the next time step for the scalar case that we just don't have in the case of systems.

However, there is a set of variables in which a one-dimensional hyperbolic system looks exactly like an advection equation, a set of uncoupled advection equations to be precise, and that is the set of characteristic variables. These variables are not global variables, but rather the result of locally linearizing the equations. Briefly, what we will do in what follows is take the entire flux limiting problem, consisting of the low order solution q^{td} and the set of "antidiffusive fluxes" A, and transform them both into a set of variables in which the same flux limiting problem looks like a set of uncoupled linear advection flux limiting problems. We then limit the fluxes using constraints physically and mathematically appropriate to an advection problem, and then transform the limited fluxes back into conserved variables, where they will be applied to q^{td} to produce the new solution q^{n+1}. This will produce results that in our view are far superior to those produced using the conserved variables directly.

Let us specialize our one-dimensional system of conservation laws Eq. (2) to the case of a flux function f which is solely a function of q (the usual case), and write it in the more compact notation

$$q_t + f(q)_x = 0 \tag{30}$$

where the subscript denotes partial differentiation. We can then further rewrite Eq. (30) in the following form:

$$q_t + A(q)q_x = 0 \tag{31}$$

where A is the $m \times m$ Jacobian matrix $\partial f/\partial q$, and m is the number of conservation laws in Eq. (30). It is not clear that Eq. (31) is any improvement over Eq. (30), since we have lost our explicit conservation form, and in general most entries of A are nonzero. If our goal is to find an appropriate set of constraints on the values of q for the purpose of flux limiting, we apparently have made no progress. But for most hyperbolic systems it is possible to find a new set of variables q' defined by a transformation matrix $T^{-1} = T^{-1}(q)$ such that Eq. (31) takes the form of a series of m decoupled advection equations:

$$T^{-1}q_t + T^{-1}A(q)T \quad T^{-1}q_x = 0 \tag{32}$$

$$q'_t + \Lambda q'_x = 0 \tag{33}$$

where $q' = T^{-1}q$ and $\Lambda = T^{-1}A(q)T$ is a diagonal matrix with diagonal elements λ_j, $1 \leq j \leq m$.

Specifically, we will have m scalar equations of the form

$$\frac{\partial q'_j}{\partial t} + \lambda_j \frac{\partial q'_j}{\partial x} = 0 \tag{34}$$

Our problem has now been reduced to performing the flux limiting step for m independent advection equations, something that we know how to do well, precisely because we have very good information on how to constrain the solution, as we demonstrated in the last section. We note that the characteristic variables q' are not the conserved quantities, and that we wish to construct our FCT algorithm such that the conserved variables q are updated in flux form. Thus it will be important to transform the fluxes themselves between the two spaces, not just the solution vectors.

To construct our characteristic variable (CV) flux limiter, we first look at the basic Boris-Book limiter, Eq. (17)

$$A^C_{i+(1/2)} = S \max\left(0, \min(|A_{i+(1/2)}|, S(q^{td}_{i+2} - q^{td}_{i+1})\Delta x, S(q^{td}_i - q^{td}_{i-1})\Delta x)\right) \tag{35}$$

This one-line formula provides one possible answer to the following question, which we term the "Flux Limiting Problem" (FLP) for advection: Given the time-advanced low order solution q^{td} and perhaps other auxiliary solution vectors, and given a set of antidiffusive fluxes A, what is a set of corrected antidiffusive fluxes A^C that are as close to A as possible, and that will constrain q^{n+1} to lie within the bounds appropriate to the advection problem? The FLP requires at least two inputs q^{td} and A, and asks for one output A^C. A look at Eq.(35), however, will convince the reader that q^{td} itself is not really needed, but rather only its first differences at flux evaluation points $i + (1/2)$. This is not atypical. All flux limiting algorithms of which we are aware have the property of depending only on local variations in q, not on the values of q themselves. This observation makes the construction of a CV limiter particularly simple.

5.2 A Characteristic Variable Implementation of the Boris-Book Flux Limiter

To be concrete here, we present a version of the CV flux limiter using the Boris-Book limiter as a building block. Using other limiters as building blocks may require some modification which will hopefully be obvious to the reader.

Given a hyperbolic system of conservation laws of length m with components j, $1 \le j \le m$ in one spatial dimension, a low-order solution vector q^{td} with components denoted $q^{td(j)}$, $1 \le j \le m$ defined on grid points x_i, and a vector of antidiffusive fluxes A with components denoted $A^{(j)}$, $1 \le j \le m$ defined at flux points $x_{i+(1/2)}$, the following steps define a characteristic variable-based implementation of the Boris-Book flux limiter.

1. Calculate some appropriate average $q^{td(j)}_{i+(1/2)}$, $\forall\, j, i$.
2. From $q^{td}_{i+(1/2)}$ calculate $T^{-1}_{i+(1/2)}$ and $T_{i+(1/2)}$, $\forall\, i$.
3. Set $D_{i+(1/2)} = T^{-1}_{i+(1/2)}(q^{td}_{i+1} - q^{td}_{i})$, $\forall\, i$.
4. Set $B_{i+(1/2)} = T^{-1}_{i+(1/2)} A_{i+(1/2)}$, $\forall\, i$.
5. Set $B^{C(j)}_{i-(1/2)} = S \max\left(0, \min(|B^{(j)}_{i+(1/2)}|, SD^{(j)}_{i+(3/2)}\Delta x, SD^{(j)}_{i-(1/2)}\Delta x)\right)$ $\forall\, j, i$, where $S = \text{sign}(1, B^{(j)}_{i+(1/2)})$.
6. Set $A^{C}_{i+(1/2)} = T_{i+(1/2)} B^{C}_{i+(1/2)}$, $\forall\, i$.

The notation above uses the superscript (j) on quantities only when it is necessary to emphasize that each component of the vector is being manipulated separately. Otherwise when the superscript is not present, a vector operation of length m is assumed.

5.3 Computational Examples: The One Dimensional Euler Equations

The equations of interest are

$$w = \begin{pmatrix} \rho \\ \rho u \\ \rho E \end{pmatrix}; \quad f = \begin{pmatrix} \rho u \\ \rho u u + P \\ \rho u E + P u \end{pmatrix} \tag{36}$$

where ρ, u, P, and E are the fluid density, velocity, pressure, and specific total energy respectively. We will assume an ideal gas equation of state

$$P = (\gamma - 1)\left(\rho E - \frac{1}{2}\rho u^2\right) \tag{37}$$

The matrices T and T^{-1} that we shall need are found by first setting

$$|A - \lambda I| = 0$$

and solving for the eigenvalues λ_j of A. Then for each of these eigenvalues the right and left eigenvectors are found. T is the matrix whose columns are the right eigenvectors of A. T^{-1} is the matrix whose rows are the corresponding left eigenvectors of A. These matrices are well known for this system. They are

$$T = \begin{bmatrix} 1 & 1 & 1 \\ u - c & u & u + c \\ H - uc & \frac{1}{2}u^2 & H + uc \end{bmatrix}$$

$$T^{-1} = \begin{bmatrix} \frac{1}{2}(\frac{\gamma-1}{2}M^2 + \frac{u}{c}) & -\frac{1}{2c} - \frac{(\gamma-1)u}{2c^2} & \frac{\gamma-1}{2c^2} \\ 1 - \frac{\gamma-1}{2}M^2 & \frac{(\gamma-1)u}{c^2} & -\frac{\gamma-1}{c^2} \\ \frac{1}{2}(\frac{\gamma-1}{2}M^2 - \frac{u}{c}) & \frac{1}{2c} - \frac{(\gamma-1)u}{2c^2} & \frac{\gamma-1}{2c^2} \end{bmatrix}$$

where $M^2 \equiv u^2/c^2$, $c^2 = \gamma P/\rho$, and $H = c^2/(\gamma-1) + u^2/2$ is the stagnation enthalpy.

For our low order flux for the Euler equations we choose the Rusanov scheme:

$$F^L_{i+(1/2)} = \left[\frac{1}{2}(f^n_{i+1} + f^n_i) - \frac{1}{4}(Q_i + Q_{i+1})(q^n_{i+1} - q^n_i)\right] \Delta t^{n+1/2} \tag{38}$$

where Q_i is the maximum characteristic speed at i:

$$Q_i = |u_i| + c_i \tag{39}$$

The high order fluxes are as given before for advection. As an example, the fourth order flux is given by:

$$F^{H4}_{i+(1/2)} = \left[\frac{7}{12}(f^a_{i+1} + f^a_i) - \frac{1}{12}(f^a_{i+2} + f^a_{i-1})\right] \Delta t^{n+1/2} \tag{40}$$

where the time level t^a is meant to denote whatever time level or average of time levels is required by the particular substep of the particular time discretization chosen.

The high order dissipative fluxes are modified versions of those used for advection, with the advection speed u replaced by the maximum characteristic speed Q. As an example, the order 4 dissipative flux is given by:

$$F^{D4}_{i+(1/2)} = -\frac{1}{2}(Q_i + Q_{i+1})\left[\frac{3}{16}(q^n_{i+1} - q^n_i) - \frac{1}{16}(q^n_{i+2} - q^n_{i-1})\right] \Delta t^{n+1/2} \tag{41}$$

The flux limiter, when we are not using the CV limiter described above, is again given by the original Boris-Book limiter:

$$A^C_{i+(1/2)} = S \max\left(0, \min(|A_{i+(1/2)}|, S(q^{td}_{i+2} - q^{td}_{i+1})\Delta x, S(q^{td}_i - q^{td}_{i-1})\Delta x)\right)$$
$$\text{where} \quad S \equiv \text{sign}(1, A_{i+(1/2)}). \tag{42}$$

All of the tests in this section use a modified Euler time discretization, each substep of which is treated in the manner described in Section 4.

Our failsafe limiter is the simplest imaginable: If, after flux limiting, either the density or the pressure in a cell is negative, all the fluxes into that cell are set to their low order values, and the grid point values recalculated. Clearly

there is much room for a more precise failsafe mechanism, but this one has proved adequate for the problems presented here.

Now that we have described the algorithms we will be using in this section, we show some results using standard test problems. The first is the shock tube problem due to Sod [11]. The initial conditions consist of a single discontinuity in density (8:1) and pressure (10:1), with both $\gamma = 1.4$ gases at rest. All of our results plot the analytic solution as a solid line, and the computed grid point values as data points, using the temperature field, which we have found to be the field most sensitive to numerical error. In Fig. 10 we show the results of our CV FCT algorithms for $N{=}N_D = 4$, 8, and 16. From left to right in each of the three plots, the reader will recognize the shock wave, the contact discontinuity, and the rarefaction fan associated with this problem. Note the marked increase in the accuracy of the contact discontinuity as the resolving power of the high order fluxes increases, similar to our experiences with advection. For comparison, in Fig. 11 we show the same three calculations, but using the more conventional FCT flux limiter which limits the fluxes based solely on the conserved variables. Note the marked superiority of the characteristic variable-based CV flux limiter.

The next test problem is the double shock tube of Woodward and Colella [12]. This problem involves the complex interaction of very strong waves of all types, and is considerably more difficult than the Sod problem. Here we show the performance of our CV FCT algorithms on three grids, of size 200, 400, and 800 grid points, testing both absolute performance and convergence. In Fig. 12, we show the density field at $t{=}0.2$ using a grid of 200 points, and using our CV FCT algorithms with $N{=}N_D = 4$, 8, and 16. As with the advection tests, and with the Sod test problem, we see increased accuracy with resolving power, but all calculations suffer from lack of resolution.

Figures 13 and 14 show the same calculations with 400 and 800 grid points respectively. We see increased accuracy with resolving power, as well as with grid refinement. Note that the two shock waves are resolved over 1-2 grid points regardless of the resolving power, but that the accuracy (sharpness in this case) of the three contact discontinuities increases markedly with increased resolving power at all refinement levels.

5.4 Using Characteristic Variables in Other FCT Components

Thus far, we have dealt with the use of characteristic variables only in the flux limiter, but there are two other FCT components that could conceivably benefit from their use: the low order fluxes, and the dissipative component of the high order fluxes. The treatment of both is quite similar, so we will discuss them together. Recall that our low order flux for advection was given by the first order upwind method:

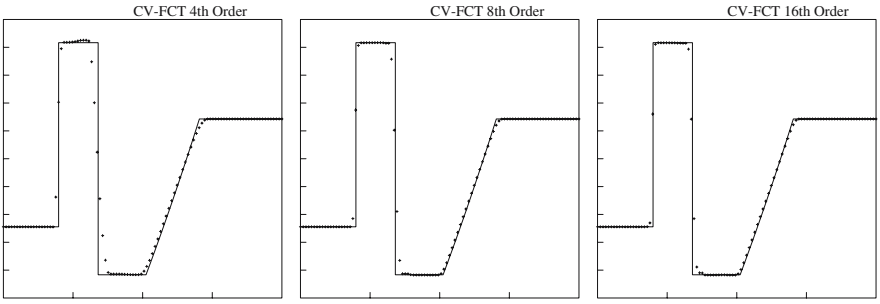


Fig. 10. Results for the temperature field, Sod shock tube problem using CV FCT algorithms with $N{=}N_D = 4$, 8, and 16

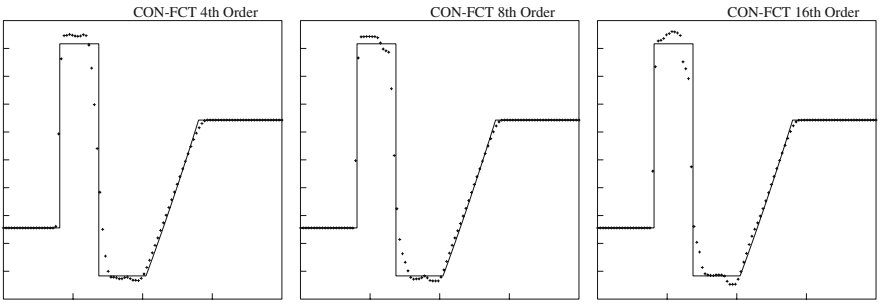


Fig. 11. Same as Fig. 10, but using a flux limiter which limits only with respect to conserved variables

$$F^L_{i+(1/2)} = \left[\frac{1}{2}(f^n_{i+1} + f^n_i) - \frac{1}{2}|u|(q^n_{i+1} - q^n_i)\right] \Delta t^{n+1/2} \tag{43}$$

It can be proven that for any flux of the above form, the coefficient $|u|/2$ used above is the smallest that will guarantee that the flux will maintain the monotonicity of a monotone profile. Thus this flux is in some sense the optimum low order flux for advection.

By contrast, our low order flux for the Euler equations was the Rusanov flux:

$$F^L_{i+(1/2)} = \left[\frac{1}{2}(f^n_{i+1} + f^n_i) - \frac{1}{4}(Q_i + Q_{i+1})(q^n_{i+1} - q^n_i)\right] \Delta t^{n+1/2} \tag{44}$$

where Q_i is the maximum characteristic speed at i:

$$Q_i = |u_i| + c_i \tag{45}$$

This flux is *not* the optimum low order flux for the Euler equations, and is in general considerably more dissipative than necessary to guarantee that

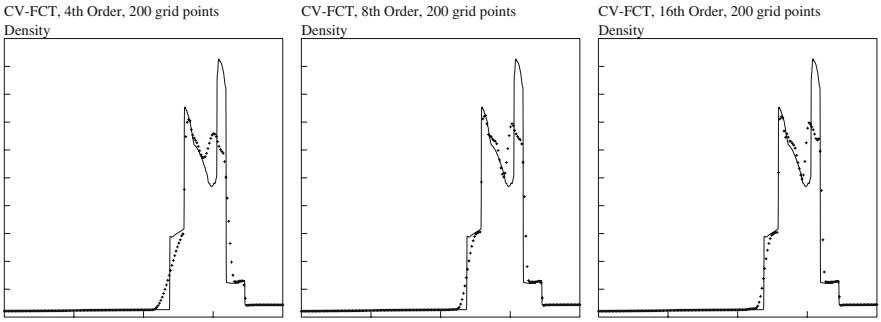


Fig. 12. Results for the density field for the 200-point Woodward-Colella double shock tube problem using CV FCT algorithms with $N{=}N_D = 4$, 8, and 16

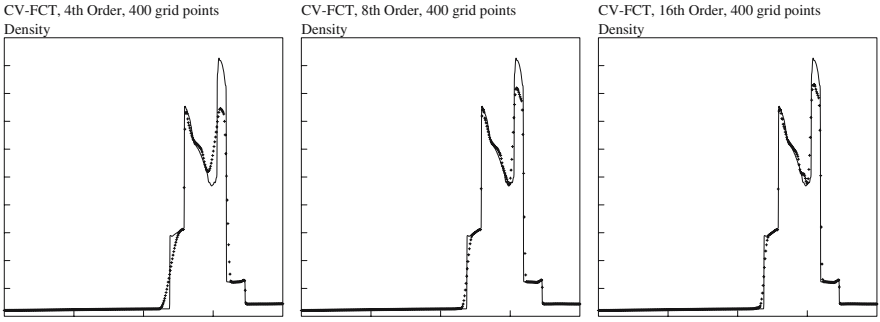


Fig. 13. Same as Fig. 12, but using a grid of 400 points

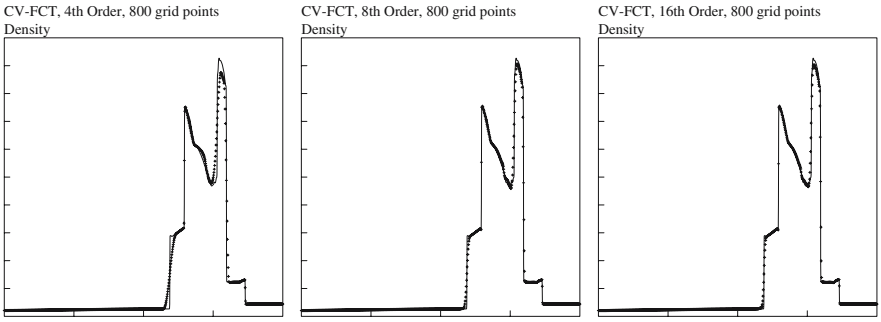


Fig. 14. Same as Fig. 12, but using a grid of 800 points

unphysical solutions cannot be generated. Rather, the Godunov flux is the optimal choice. This flux requires the solution of the full nonlinear Riemann problem at each flux point. However, a good approximation to the Godunov

flux $F^G_{i+(1/2)}$ is obtained by doing exactly what we did to limit fluxes: We transform the entire "low order flux" problem into characteristic variables, where the system is of the form of m uncoupled advection problems, compute first order upwind fluxes in those variables, and then transform the fluxes back to conserved variables:

1. Calculate some appropriate average $q^{n(j)}_{i+(1/2)}$, $\forall\, j, i$.
2. From $q^n_{i+(1/2)}$ calculate $T^{-1}_{i+(1/2)}$ and $T_{i+(1/2)}$, $\forall\, i$.
3. Set $D_{i+(1/2)} = T^{-1}_{i+(1/2)}(q^n_{i+1} - q^n_i)$, $\forall\, i$.
4. Set $D^{(j)}_{i+(1/2)} = -\frac{|\lambda^{(j)}_{i+(1/2)}|}{2} D^{(j)}_{i+(1/2)}$, $\forall\, i, j$.
5. Set $F^L_{i+(1/2)} = \left[\frac{1}{2}(f^n_{i+1} + f^n_i) + T_{i+(1/2)} D_{i+(1/2)}\right] \Delta t^{n+1/2}$.

In principle, this would be a much better choice for our low order flux than the Rusanov flux we have chosen, because it would be less dissipative. However, we cannot forget that the primary property that we want of our low order flux is its guaranteed freedom from unphysical behavior. Anyone familiar with the modern literature on approximate Riemann solvers will recognize the above as one of the popular ways of constructing them. He or she will also know that such approximate solvers are in general devoid of the guarantees that we need. Thus we leave this promising topic for future exploration, and move on to the related topic of optimizing the adaptive dissipation in the high order flux.

Recall that our order 4 dissipative flux for advection was given by:

$$F^{D4}_{i+(1/2)} = -|u| \left[\frac{3}{16}(q^n_{i+1} - q^n_i) - \frac{1}{16}(q^n_{i+2} - q^n_{i-1})\right] \Delta t^{n+1/2} \tag{46}$$

while the corresponding flux for the Euler equations was

$$F^{D4}_{i+(1/2)} = -\frac{1}{2}(Q_i + Q_{i+1}) \left[\frac{3}{16}(q^n_{i+1} - q^n_i) - \frac{1}{16}(q^n_{i+2} - q^n_{i-1})\right] \Delta t^{n+1/2} \tag{47}$$

where Q_i is again the maximum characteristic speed at i:

$$Q_i = |u_i| + c_i \tag{48}$$

Comparing the above pair of equations with the preceding pair, we see that the order 4 dissipative flux for the Euler equations suffers from the same flaw as does the Rusanov flux: in general it will provide more dissipation than is needed. The most extreme example of this is that of very low Mach number flow in which advected waves would be subject to a dissipation proportional to c, while the waves themselves were moving with a velocity of $v \ll c$. A way of addressing this is, again, to use the characteristic variables, dissipating each of the component waves in proportion to its own wave speed.

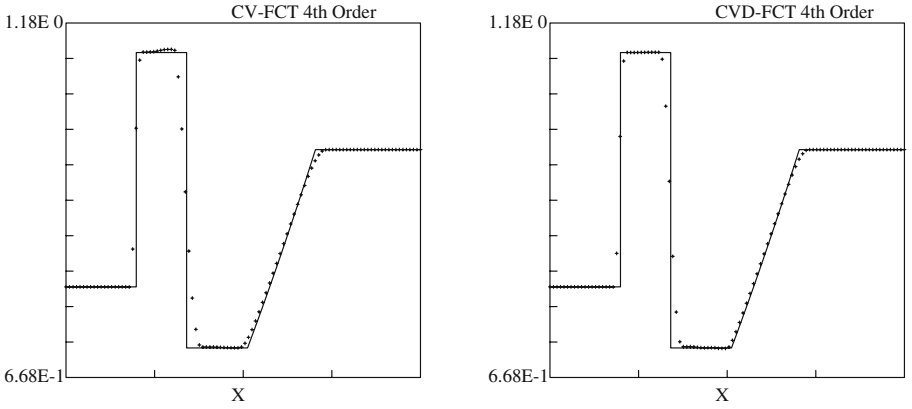


Fig. 15. Sod shock tube problem: Comparison of the $N{=}N_d{=}4$ CV FCT algorithm using the conventional adaptive dissipation given by Eq. (47) (left), and the CV-based adaptive dissipation described in the text (right)

To give a concrete example of the procedure, let us first rewrite Eq. (47):

$$F^{D4}_{i+(1/2)} = \frac{1}{32}(Q_i + Q_{i+1})\left[\Delta q_{i-(1/2)} - 2\Delta q_{i+(1/2)} + \Delta q_{i+(3/2)}\right]\Delta t^{n+1/2}$$

$$\text{where} \quad \Delta q_{i+(1/2)} \equiv q^n_{i+1} - q^n_i. \tag{49}$$

Our CV dissipative flux of order 4 would then be computed as follows:

1. Calculate some appropriate average $q^{n(j)}_{i+(1/2)}$, $\forall\, j, i$.
2. From $q^n_{i+(1/2)}$ calculate $T^{-1}_{i+(1/2)}$ and $T_{i+(1/2)}$, $\forall\, i$.
3. Set $\Delta_{i+(1/2)} = T^{-1}_{i+(1/2)}(q^n_{i+1} - q^n_i)$, $\forall\, i$.
4. Set $D^{(j)}_{i+(1/2)} = \frac{|\lambda^{(j)}_{i+(1/2)}|}{16}(\Delta^{(j)}_{i-(1/2)} - 2\Delta^{(j)}_{i+(1/2)} + \Delta^{(j)}_{i+(3/2)})$, $\forall\, j, i$.
5. Set $F^{D4}_{i+(1/2)} = \left[T_{i+(1/2)}D_{i+(1/2)}\right]\Delta t^{n+1/2}$.

Other adaptive dissipative fluxes can be computed in a similar manner. Rerunning all of our previous CV limiter calculations with this CV adaptive dissipation, we find that only the $N = 4$ calculations display any significant differences. We show only those here. In Fig. 15 we compare two calculations, both using CV limiting, for the Sod shock tube problem. The left panel is the same as that of the left panel in Fig. 10, using the adaptive dissipation given by Eq. (47), while the right panel instead uses the CV adaptive dissipation given by the above algorithm. Note a significant increase in the sharpness of the contact discontinuity, without any other adverse effects. This is, of course, what we hoped we would achieve.

6 Flux-Corrected Transport in Multidimensions

As we have stated before, there is a large class of problems for which an operator splitting strategy, using sequences of one-dimensional time-advancement operators, can be successful. We are assuming here, however, that we are interested in pursuing a more fully multidimensional approach wherein the results are independent, or as independent as possible, of any apparent ordering of one-dimensional operators. Of the three components of an FCT algorithm, only the flux limiter normally presents any difficulty in this regard. Indeed, much of [13] was devoted to defining a fully multidimensional flux limiter, which we present below.

6.1 A Fully Multidimensional Flux Limiter

The alternative flux limiting algorithm presented in Section 4.2 generalizes trivially to any number of spatial dimensions, and in fact to unstructured as well as the structured meshes we consider here. For the sake of completeness we present the algorithm for the structured coordinate-aligned two dimensional mesh referred to in Eq. (10).

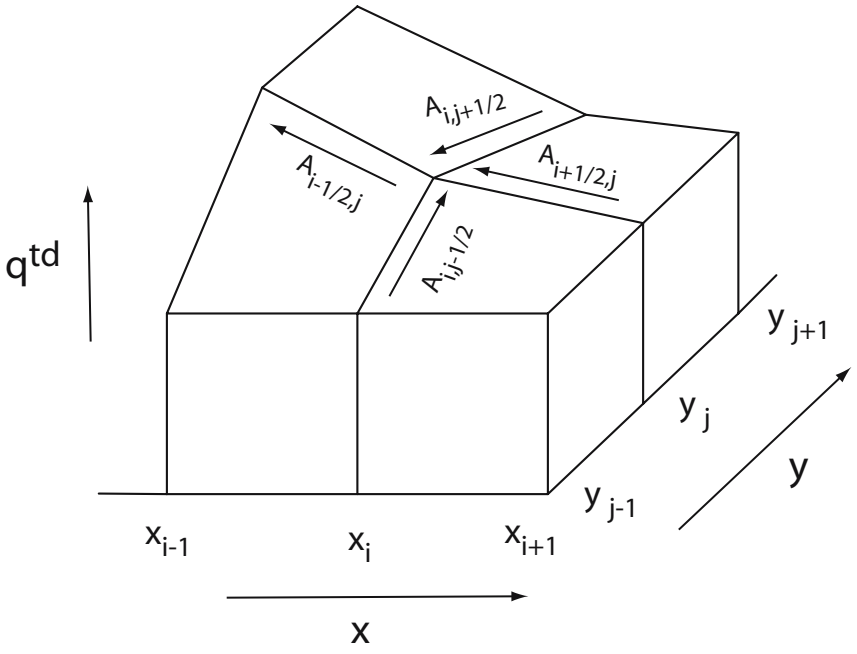


Fig. 16. Schematic of the flux limiting problem in two dimensions

Referring to Fig. 16, we seek to limit the antidiffusive fluxes $A_{i+(1/2),j}$ and $A_{i,j+(1/2)}$ by finding coefficients $C_{i+(1/2),j}$ and $C_{i,j+(1/2)}$ such that

$$A^C_{i+(1/2),j} = C_{i+(1/2),j} A_{i+(1/2),j}, \quad 0 \le C_{i+(1/2),j} \le 1$$

$$A^C_{i,j+(1/2)} = C_{i,j+(1/2)} A_{i,j+(1/2)}, \quad 0 \le C_{i,j+(1/2)} \le 1$$

and such that $A^C_{i+(1/2),j}$, $A^C_{i-(1/2),j}$, $A^C_{i,j+(1/2)}$, and $A^C_{i,j-(1/2)}$ acting in concert shall not cause

$$q^{n+1}_{ij} = q^{td}_{ij} - \Delta V^{-1}_{ij}[A^C_{i+(1/2),j} - A^C_{i-(1/2),j} + A^C_{i,j+(1/2)} - A^C_{i,j-(1/2)}]$$

to exceed some maximum value q^{max}_{ij} or fall below some minimum value q^{min}_{ij}. The procedure is completely analogous to that given in Section 2:

1. Compute, for each grid point ij, physically-motivated upper and lower bounds on the solution in the next timestep, q^{max}_{ij} and q^{min}_{ij} respectively.
2. For the upper bound, compute P, Q, and their ratio R at each grid point:

$$P^+_{ij} = \max(A_{i-(1/2),j}, 0) - \min(A_{i+(1/2),j}, 0) \tag{50}$$

$$+ \max(A_{i,j-(1/2)}, 0) - \min(A_{i,j+(1/2)}, 0) \tag{51}$$

$$Q^+_{ij} = (q^{max}_{ij} - q^{td}_{ij})\Delta V_{ij} \tag{52}$$

$$R^+_{ij} = \min(1, Q^+_{ij}/P^+_{ij}),\ P^+_{ij} > 0, \quad 0 \ \text{otherwise} \tag{53}$$

3. For the lower bound, compute P, Q, and their ratio R at each grid point:

$$P^-_{ij} = \max(A_{i+(1/2),j}, 0) - \min(A_{i-(1/2),j}, 0) \tag{54}$$

$$+ \max(A_{i,j+(1/2)}, 0) - \min(A_{i,j-(1/2)}, 0) \tag{55}$$

$$Q^-_{ij} = (q^{td}_{ij} - q^{min}_{ij})\Delta V_{ij} \tag{56}$$

$$R^-_{ij} = \min(1, Q^-_{ij}/P^-_{ij}),\ P^-_{ij} > 0, \quad 0 \ \text{otherwise} \tag{57}$$

4. Compute $C_{i+(1/2),j}$ and $C_{i,j+(1/2)}$ by taking a minimum:

$$C_{i+(1/2),j} = \begin{cases} \min(R^+_{i+1,j}, R^-_{ij}) \text{ when } A_{i+(1/2),j} > 0, \\ \min(R^+_{ij}, R^-_{i+1,j}) \text{ when } A_{i+(1/2),j} \le 0. \end{cases} \tag{58}$$

$$C_{i,j+(1/2)} = \begin{cases} \min(R^+_{i,j+1}, R^-_{ij}) \text{ when } A_{i,j+(1/2)} > 0, \\ \min(R^+_{ij}, R^-_{i,j+1}) \text{ when } A_{i,j+(1/2)} \le 0. \end{cases} \tag{59}$$

Again note that in the above we do not specify the equivalent of Eq. (14) in [13]. As we have stated, we do not consider that equation to be part of the flux limiter proper, but rather an algorithm for pre-constraining the high order fluxes. Nonetheless, for fully multidimensional problems we have yet to

find anything better, and we use an abbreviated form of that equation to pre-constrain the high order fluxes in the multidimensional advection problems that follow:

$$\begin{aligned} A_{i+(1/2),j} &= 0 \text{ if } A_{i+(1/2),j}(q^{td}_{i+1,j} - q^{td}_{ij}) \le 0 \\ A_{i,j+(1/2)} &= 0 \text{ if } A_{i,j+(1/2)}(q^{td}_{i,j+1} - q^{td}_{ij}) \le 0 \end{aligned} \tag{60}$$

With our fully multidimensional flux limiter in hand, along with our algorithm for pre-constraining the high order fluxes Eq. (60), let us consider two multidimensional problems: passively-driven convection in two dimensions, and compressible gasdynamics in two dimensions.

6.2 FCT Algorithms for Two-Dimensional Passively-Driven Convection

We shall be interested in solving Eq. (8) for the special case where $q(x, y)$ is a scalar and where

$$f = qu \tag{61}$$

$$g = qv \tag{62}$$

Here $u(x, y)$ and $v(x, y)$ are convection velocity components in the x and y directions respectively. They are assumed to be specified either globally or at the very least at cell boundaries. Thus our equation is

$$q_t + (qu)_x + (qv)_y = 0 \tag{63}$$

Our first order of business is to specify high and low order fluxes. Since $u_{i+(1/2),j}$ and $v_{i,j+(1/2)}$ are specified at cell faces, our job reduces to specifying $q_{i+(1/2),j}$ and $q_{i,j+(1/2)}$ at cell faces, and then multiplying them by the appropriate cell face velocity. The low order fluxes, the high order fluxes, and the high order dissipation components are all straightforward generalizations of the fluxes we used in one dimensional linear advection.

For our low order fluxes, we choose a two-dimensional donor cell algorithm:

$$q^L_{i+(1/2),j} = \left\{ \begin{array}{ll} q_{ij} & \text{when } u_{i+(1/2),j} > 0 \\ q_{i+1,j} & \text{when } u_{i+(1/2),j} \le 0 \end{array} \right\} \tag{64}$$

$$q^L_{i,j+(1/2)} = \left\{ \begin{array}{ll} q_{ij} & \text{when } v_{i,j+(1/2)} > 0 \\ q_{i,j+1} & \text{when } v_{i,j+(1/2)} \le 0 \end{array} \right\} \tag{65}$$

$$F^L_{i+(1/2),j} = q^L_{i+(1/2),j} u_{i+(1/2),j} S_{i+(1/2),j} \Delta t^{n+1/2} \tag{66}$$

$$G^L_{i,j+(1/2)} = q^L_{i,j+(1/2)} v_{i,j+(1/2)} S_{i,j+(1/2)} \Delta t^{n+1/2} \tag{67}$$

where $S_{i+(1/2),j}$ and $S_{i,j+(1/2)}$ are the areas of the x and y cell faces respectively. We note that the above "four-flux" donor cell algorithm does not account for corner transport in a single step. While we do not describe them here, variants of the above do allow corner transport and at the same time satisfy the prime requirement of preventing unphysical values of q. Thus these variants are the preferred low order fluxes for this problem, and are the ones we use here.

The high order fluxes are again computed using the formulae in the Appendix of [13]. As an example, the fourth order fluxes are given by:

$$q^{H4}_{i+(1/2),j} = \frac{7}{12}(q^a_{i+1,j} + q^a_{ij}) - \frac{1}{12}(q^a_{i+2,j} + q^a_{i-1,j}) \tag{68}$$

$$q^{H4}_{i,j+(1/2)} = \frac{7}{12}(q^a_{i,j+1} + q^a_{ij}) - \frac{1}{12}(q^a_{i,j+2} + q^a_{i,j-1}) \tag{69}$$

$$F^{H4}_{i+(1/2),j} = q^{H4}_{i+(1/2),j} u_{i+(1/2),j} S_{i+(1/2),j} \Delta t^{n+1/2} \tag{70}$$

$$G^{H4}_{i,j+(1/2)} = q^{H4}_{i,j+(1/2)} v_{i,j+(1/2)} S_{i,j+(1/2)} \Delta t^{n+1/2} \tag{71}$$

where the time level t^a is meant to denote whatever time level or average of time levels is required by the particular substep of the particular time discretization chosen.

The high order dissipative fluxes of order N_D, which are added to the above high order fluxes, again follow very closely to their one-dimensional counterparts. As an example, the order 4 dissipative fluxes are given by:

$$F^{D4}_{i+(1/2),j} = -|u_{i+(1/2),j}| \left[\frac{3}{16}(q^n_{i+1,j} - q^n_{ij}) - \frac{1}{16}(q^n_{i+2,j} - q^n_{i-1,j})\right] S_{i+(1/2),j} \Delta t^{n+1/2} \tag{72}$$

$$F^{D4}_{i,j+(1/2)} = -|v_{i,j+(1/2)}| \left[\frac{3}{16}(q^n_{i,j+1} - q^n_{ij}) - \frac{1}{16}(q^n_{i,j+2} - q^n_{i,j-1})\right] S_{i,j+(1/2)} \Delta t^{n+1/2} \tag{73}$$

The pre-constraint of the high order fluxes is given by Eq. (60). Since we will be limiting directly on the variable q, there is no need for a fail-safe procedure.

For our flux limiter, we choose the multidimensional limiter given above, with q^{max}_{ij} and q^{min}_{ij} specified thus:

$$\begin{aligned} q^+_{ij} &= \max(q^n_{ij}, q^{td}_{ij}) \\ q^{max}_{ij} &= \max(q^+_{i-1,j}, q^+_{i,j}, q^+_{i+1,j}, q^+_{i,j-1}, q^+_{i,j+1}) \\ q^-_{ij} &= \min(q^n_{ij}, q^{td}_{ij}) \\ q^{min}_{ij} &= \min(q^-_{i-1,j}, q^-_{i,j}, q^-_{i+1,j}, q^-_{i,j-1}, q^-_{i,j+1}) \end{aligned} \tag{74}$$

For our test problem we choose the solid body rotation problem given in [13]. We have Eq. (63) with $u = -\Omega(y - y_0)$ and $v = \Omega(x - x_0)$, where Ω is a constant angular velocity, and (x_0, y_0) is the axis of rotation. The computational grid is 100×100 cells, $\Delta x = \Delta y$, with counterclockwise rotation taking place about grid point (50,50). Centered at grid point (50,75) is a cylinder of radius 15 grid points, through which a slot has been cut of width 5 grid points. The time step and rotation speed are such that 1256 time steps will effect one complete revolution of the cylinder about the central point. A perspective view of the initial conditions is shown in Fig. 17. In this and following figures, only the central 50×50 array of grid points around the analytic center of the distribution is shown.

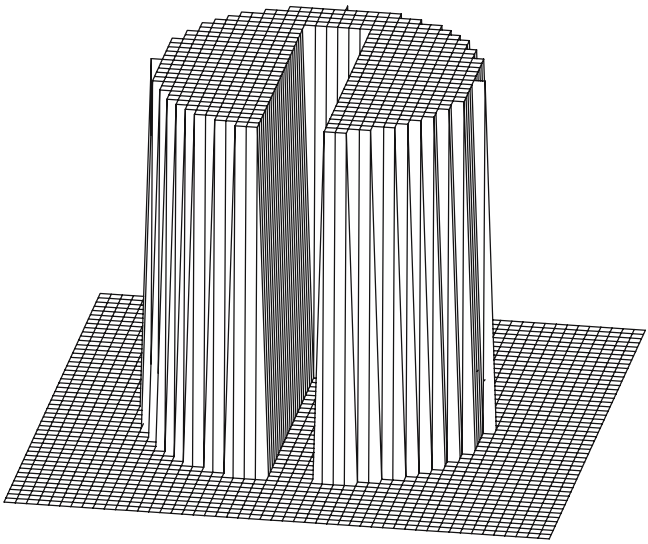


Fig. 17. Initial Condition. Grid points inside the slotted cylinder have $q = 3.0$. All others have $q = 1.0$. Only the central 50×50 array of grid points around the analytic center of the distribution is shown

In Fig. 18 we show the results after one revolution of the cylinder about the axis, using $N = N_D = 4$. We show the profile from four different angles, with the L_1 error denoted by "AE." Overall, the FCT algorithm has performed well. Nowhere on the grid are there values of q outside the bounds of the analytic result, bounds the high order algorithm would have violated in the very first timestep. Yet the numerical diffusion, although certainly present, is far less than that which would have been generated by the low order algorithm. The top of the cylinder has remained flat and free of oscillations, and kept its original value of 3.0. The flat area outside the cylinder has also remained flat and free of oscillations, and kept its original value of 1.0. The L_1 error

is 0.0276. The profile is a bit more diffuse than the fourth order calculation shown in [13]. This can be attributed to the fact that we include a fourth order dissipation term in the high order flux in the present calculation, and did not do so in [13].

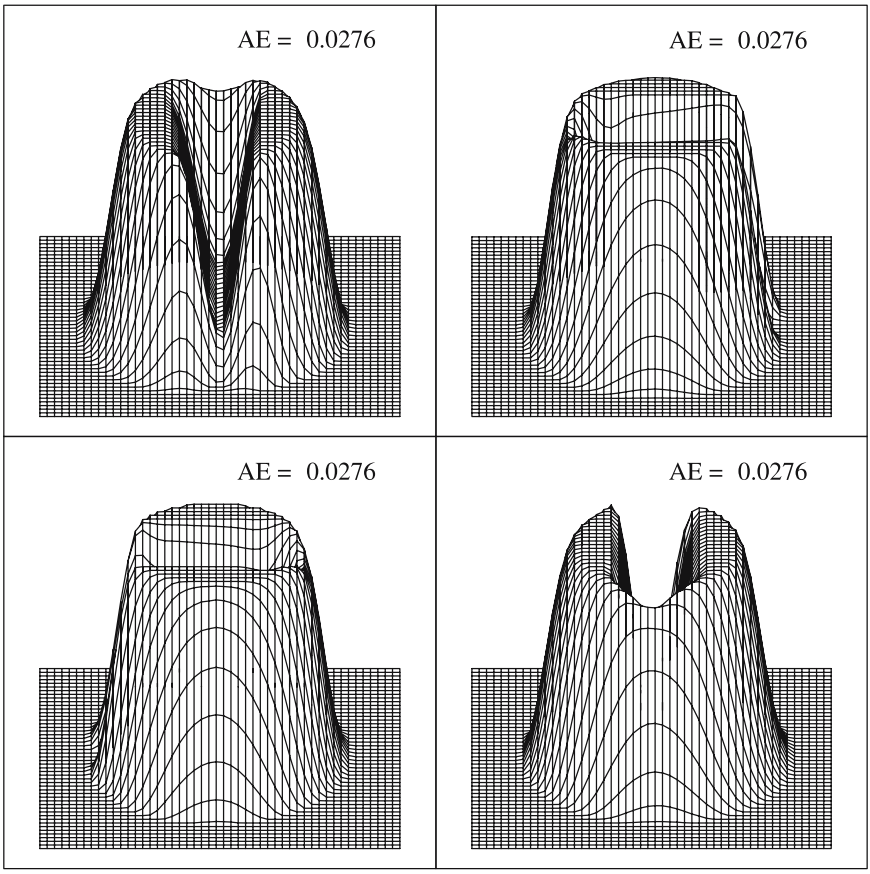


Fig. 18. Results after one revolution with $N = N_D = 4$

In Section 4, we found that by increasing the resolving power of the high order fluxes, we could improve the performance of the corresponding FCT algorithm for one-dimensional advection. Let us see if that pattern plays out in multidimensional advection as well. In Fig. 19 we show the results after one revolution of the cylinder about the axis, using $N = N_D = 8$. The L_1 error is 0.0170. The results are clearly quite a bit better than the $N = N_D = 4$ calculation. There is far less erosion in the slot, and the bridge connecting the two halves of the cylinder has maintained its integrity. In Fig. 20 we show the

results after one revolution of the cylinder about the axis, using $N = N_D = 16$. The L_1 error is 0.0138. Again we see a marked improvement with increasing resolving power in the high order flux.

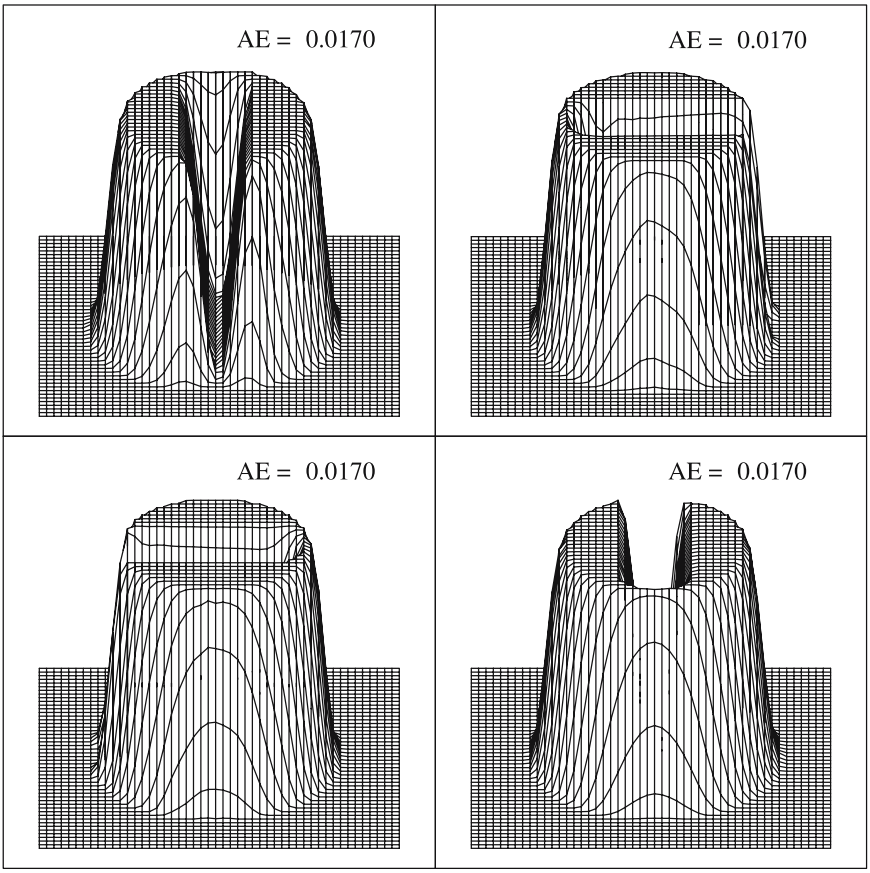


Fig. 19. Results after one revolution with $N = N_D = 8$

A careful look at Fig. 20 will reveal an aspect of the multidimensional limiter used with the bounds given in Eq. (74) that has been noted both in [13] and more recently by DeVore [6]: Even though this combination of limiter and upper and lower bounds does prevent the occurrence of maxima and minima beyond those bounds, this property is not synonymous with the enforcement of monotonicity in any given coordinate direction. Note in particular the breaking of one-dimensional monotonicity along the front upper portion of the cylinder in the lower left panel of Fig. 20. Such breaking of monotonicity is often, but not always, caused by the development of dispersive ripples

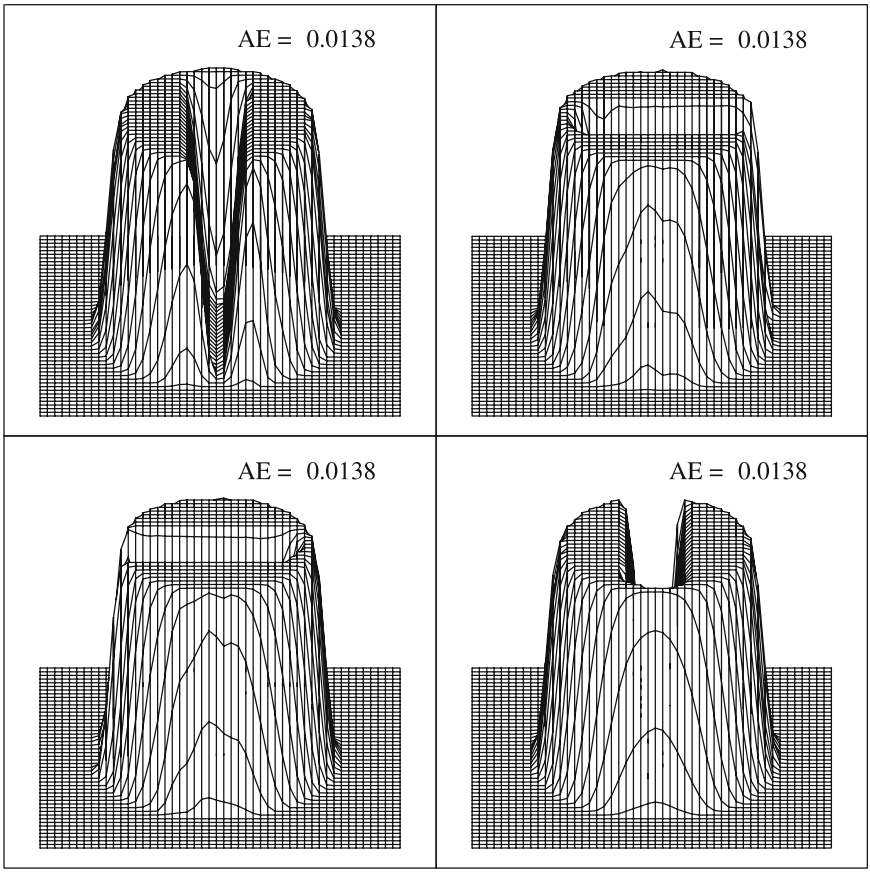


Fig. 20. Results after one revolution with $N = N_D = 16$

due to high order fluxes in one direction which are not seen as errors by the multidimensional limiter due to the presence of a steep gradient in a transverse direction. To address this issue, both [13] and [6] recommended adding a "pre-limiting" step before the multidimensional flux limiter, consisting of a call to the Boris-Book flux limiter for each of the one-dimensional fluxes. That is, prior to the multidimensional flux limiter, $A_{i+(1/2),j}$ is limited with respect to q^{td} in the x-direction, and $A_{i,j+(1/2)}$ is limited with respect to q^{td} in the y-direction, using the Boris-Book limiter. In Fig. 21 we show the results of applying that technique to the $N = N_D = 16$ calculation shown previously in Fig. 20. Although many of the regions of broken monotonicity have been eliminated, the overall solution has been degraded. Significant erosion of the slot and the bridge have taken place, and we now have an L_1 error of 0.0159. This degradation is due primarily to the fact that peaked profiles naturally occur along the outer portions of the cylinder, both initially and as the profile

moves and diffuses slightly. These peaked profiles are subsequently "clipped" by the Boris-Book limiter, giving us worse results than if we had not invoked the pre-limiter at all, at least for this problem.

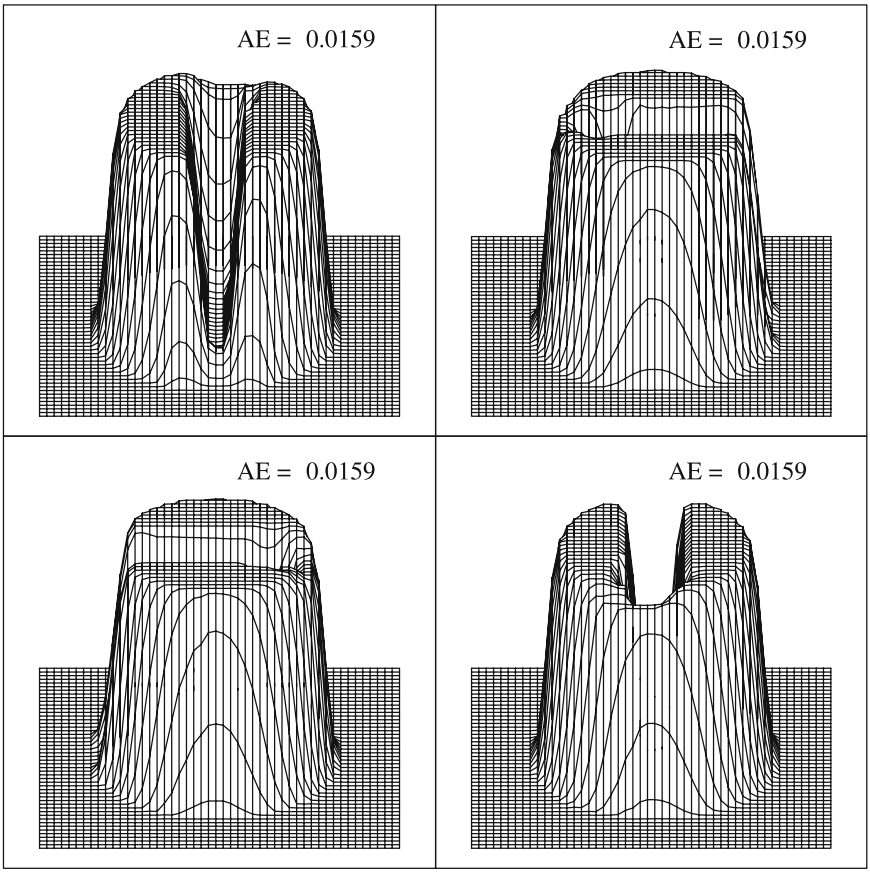


Fig. 21. $N = N_D = 16$ with the Boris-Book pre-limiter

A solution to the above dilemma is to pre-limit using a one-dimensional limiter as above, but to do so using a limiter which does not clip extrema, rather than the Boris-Book limiter. In Fig. 22 we show the results of using a slightly modified version of the non-clipping one-dimensional flux limiter described in Section 4.3 to "pre-limit" the $N = N_D = 16$ calculation shown previously in Fig. 20. We see that not only have many of the regions of broken monotonicity vanished, but the overall solution has improved, with an L_1 error of 0.0137. Thus if pre-limiting is deemed advisable, our recommendation is to use non-clipping limiters rather than the Boris-Book limiter to accomplish that task.

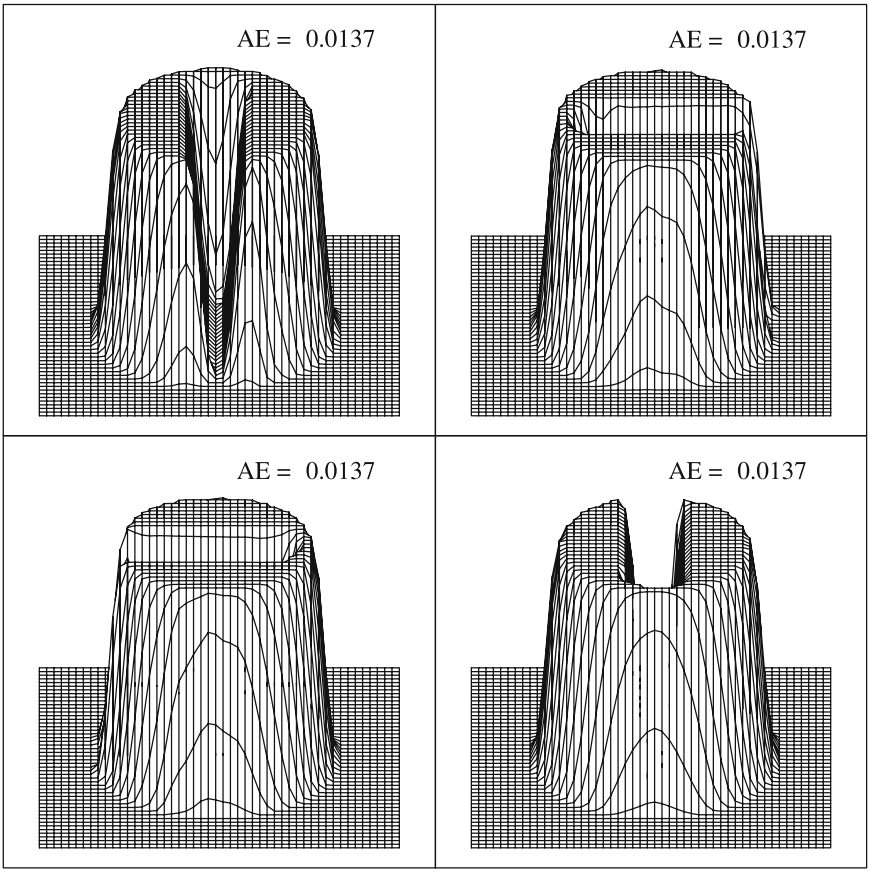


Fig. 22. $N = N_D = 16$ with a non-clipping pre-limiter

6.3 FCT Algorithms for Two-Dimensional Compressible Gasdynamics

We are interested in solving the equations of two-dimensional compressible inviscid fluid flow Eq. (9). Recall that when we studied the corresponding one-dimensional system, we found a distinct advantage to limiting with respect to the characteristic variables rather than the conserved variables. We also found that we could use the Boris-Book limiter with fairly good success, indicating that clipping was not a serious problem, at least when one uses characteristic variables. Here we will try to build on that success.

We immediately face an apparent problem, however. The characteristic variables are only rigorously defined for one spatial dimension, i.e., it is not possible to simultaneously diagonalize both f and g with the same similarity transformation (for gasdynamics). It is clear, then, that if we wish to limit

with respect to characteristic variables, we can only perform flux limiting in one direction at a time. We shall use the characteristic form of the Boris-Book limiter that we developed in Section 5.2 for this task, using it in such a manner as to preserve as much full multidimensionality as possible in the algorithm.

The flux limiter we shall use is exactly as described in Section 5.2, except that we shall require similarity transformations appropriate for the full set of four conserved variables. The ones we actually use here are those appropriate for three-dimensional gasdynamics, with five conserved variables, with the third component of momentum set to zero. The matrices T and T^{-1} in the x direction are given by

$$T = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ u-c & 0 & 0 & u & u+c \\ v & 1 & 0 & v & v \\ w & 0 & 1 & w & w \\ H-uc & v & w & \frac{1}{2}q^2 & H+uc \end{bmatrix}$$

$$T^{-1} = \begin{bmatrix} \frac{1}{2}(\frac{\gamma-1}{2}M^2 + \frac{u}{c}) & -\frac{1}{2c} - \frac{(\gamma-1)u}{2c^2} & -\frac{(\gamma-1)v}{2c^2} & -\frac{(\gamma-1)w}{2c^2} & \frac{\gamma-1}{2c^2} \\ -v & 0 & 1 & 0 & 0 \\ -w & 0 & 0 & 1 & 0 \\ 1-\frac{\gamma-1}{2}M^2 & \frac{(\gamma-1)u}{c^2} & \frac{(\gamma-1)v}{c^2} & \frac{(\gamma-1)w}{c^2} & -\frac{\gamma-1}{c^2} \\ \frac{1}{2}(\frac{\gamma-1}{2}M^2 - \frac{u}{c}) & \frac{1}{2c} - \frac{(\gamma-1)u}{2c^2} & -\frac{(\gamma-1)v}{2c^2} & -\frac{(\gamma-1)w}{2c^2} & \frac{\gamma-1}{2c^2} \end{bmatrix}$$

where $q^2 \equiv u^2 + v^2 + w^2$, $M^2 \equiv q^2/c^2$, H is the stagnation enthalpy, and u, v, and w are the x, y, and z components of velocity respectively. For the y direction, a corresponding set of transformation matrices is used.

The easiest solution is, of course, to simply use directional operator splitting. To demonstrate that such a technique is viable, in Fig. 23 we show a calculation using a directionally split version of the $N = 8$, $N_D = 8$ CV FCT algorithm given here to solve the Mach reflection problem given by Woodward and Colella [12]. The problem consists of a Mach 10 shock reflecting from a 30 degree wedge. The three resolutions used in [12] are shown, corresponding to meshes of 120×30, 240×60, and 480×120 from top to bottom. We invite the reader to compare the results to those obtained elsewhere. In Fig. 24 we show the same calculation, but using the conventional non-CV limiter which limits only on the conserved variables. The morphology of the jet along the bottom wall disagrees both with experimental data and with other published numerical calculations. We conclude that the CV limiter used in Fig. 23 is by far the better choice. We also conclude that, for this particular test problem, directional splitting is satisfactory.

Of course one would prefer not to use directional splitting, since one cannot be sure in advance that the particular physics problem of interest will yield satisfactory results when such splitting is employed. Thus we would prefer not to use full-blown directional splitting, and yet the variables which we desire

to use for flux limiting would seem to require that the limiting step itself be directionally split. Is there a way to satisfy both requirements? Is there some way to be "fully multidimensional" and also use characteristic variables?

One way to define the term "fully multidimensional algorithm" is to demand that the results be independent of any choice of ordering that may be present in an algorithm. Another, perhaps just as satisfactory, is to demand that same independence, except for the flux limiting step itself. We use one variant of each here, which we describe in compact form:

1. Compute all high and low order fluxes fully multidimensionally
2. Either
   - Limit the x-, y-, and z-directed fluxes independently; or
   - Limit the x-, y-, and z-directed fluxes sequentially, updating solution values between steps

The first choice increases the risk that the failsafe limiter will be brought into play, but is truly multidimensional. The second is less likely to generate the need for the failsafe limiter, but is multidimensional only in the second sense above. In Fig. 25, we show the results of the Woodward-Colella Mach reflection problem using a CV limiter and limiting the x- and y-directed fluxes independently. In Fig. 26, we show the results using a CV limiter and limiting the x- and y-directed fluxes sequentially. Both are seen to perform quite well, albeit with results that are virtually indistinguishable from those in Fig. 23. Clearly this test problem, although it is the standard test problem for multidimensional compressible flow, is not one for which one needs a fully multidimensional algorithm to achieve accurate solutions. Nonetheless, we would recommend either of the two multidimensional approaches over the fully split one, as a means of avoiding the splitting errors that may occur when simulating more general flows.

7 Conclusions

We have tried to give the reader a distillation of the design principles for building FCT algorithms, and front-capturing algorithms in general, that we have gleaned from experience over the past several decades. If there is a common thread to all of them it is this: the scientists who use such algorithms must have both input to and knowledge of their design. There may come a day when we no longer hold to this view, when the design of such algorithms can be left to expert numerical analysts alone, but that day has not yet arrived.

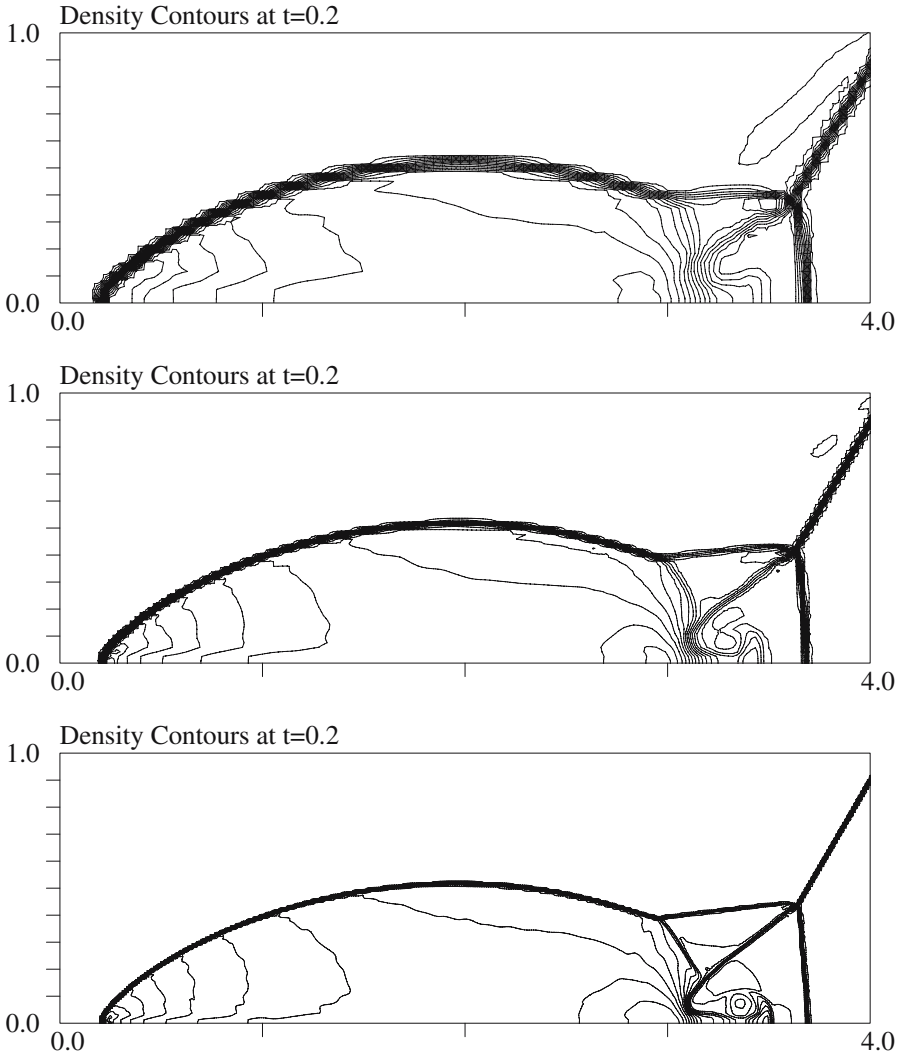


Fig. 23. Isodensity contours for the Woodward-Colella ramp problem at $t = 0.2$ using a CV limiter with $N{=}N_D{=}8$, and directional splitting. From top to bottom, the displayed grids are 120×30, 240×60, and 480×120

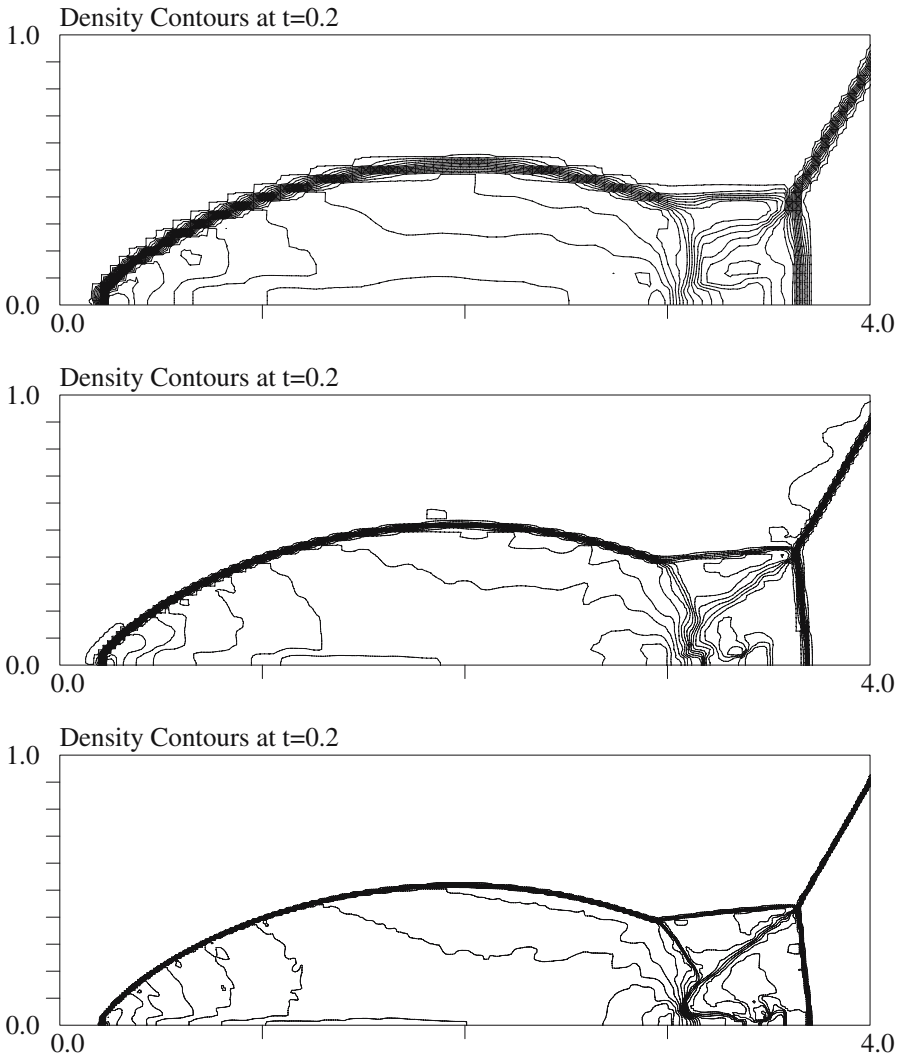


Fig. 24. Same as Fig. 23, but using the conventional flux limiter which limits only on the conserved variables. The morphology of the jet along the bottom wall disagrees both with experimental data and with other published numerical calculations. We conclude that the CV limiter used in Fig. 23 is by far the better choice

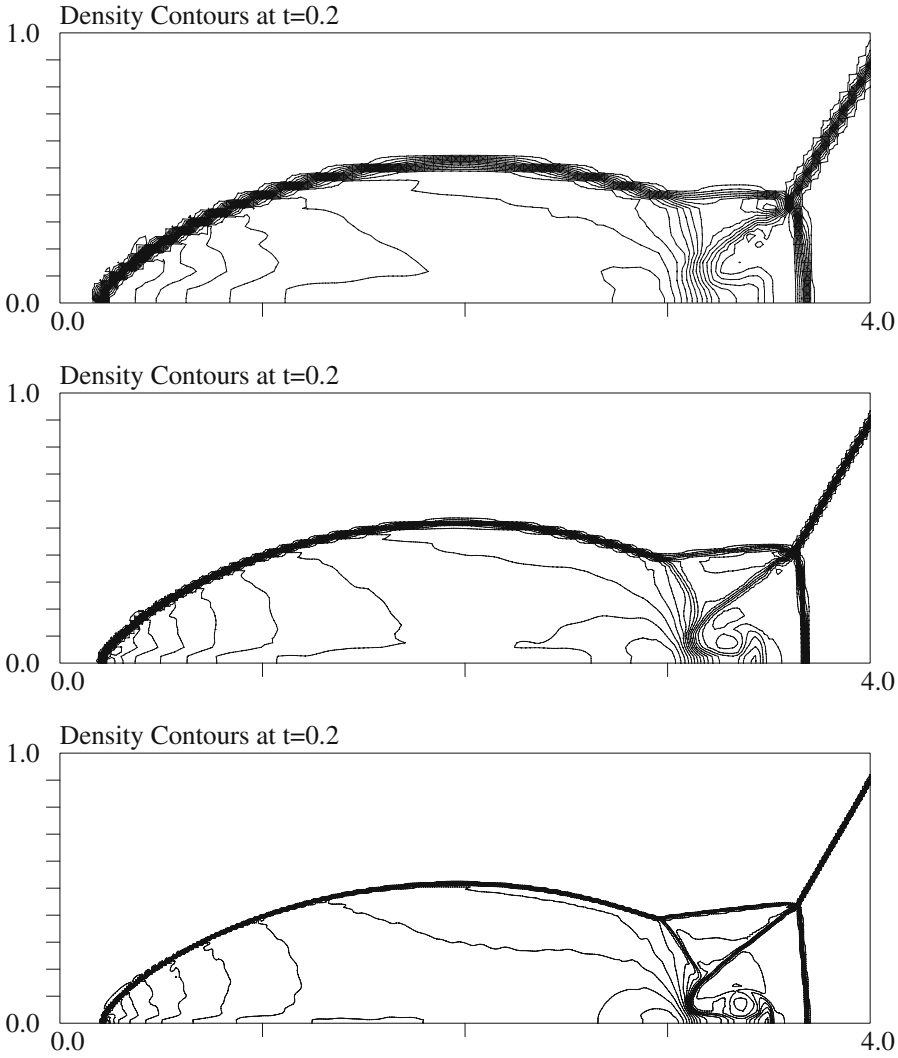


Fig. 25. Same as Fig. 23, but using a CV limiter independently on fully multidimensional antidiffusive fluxes

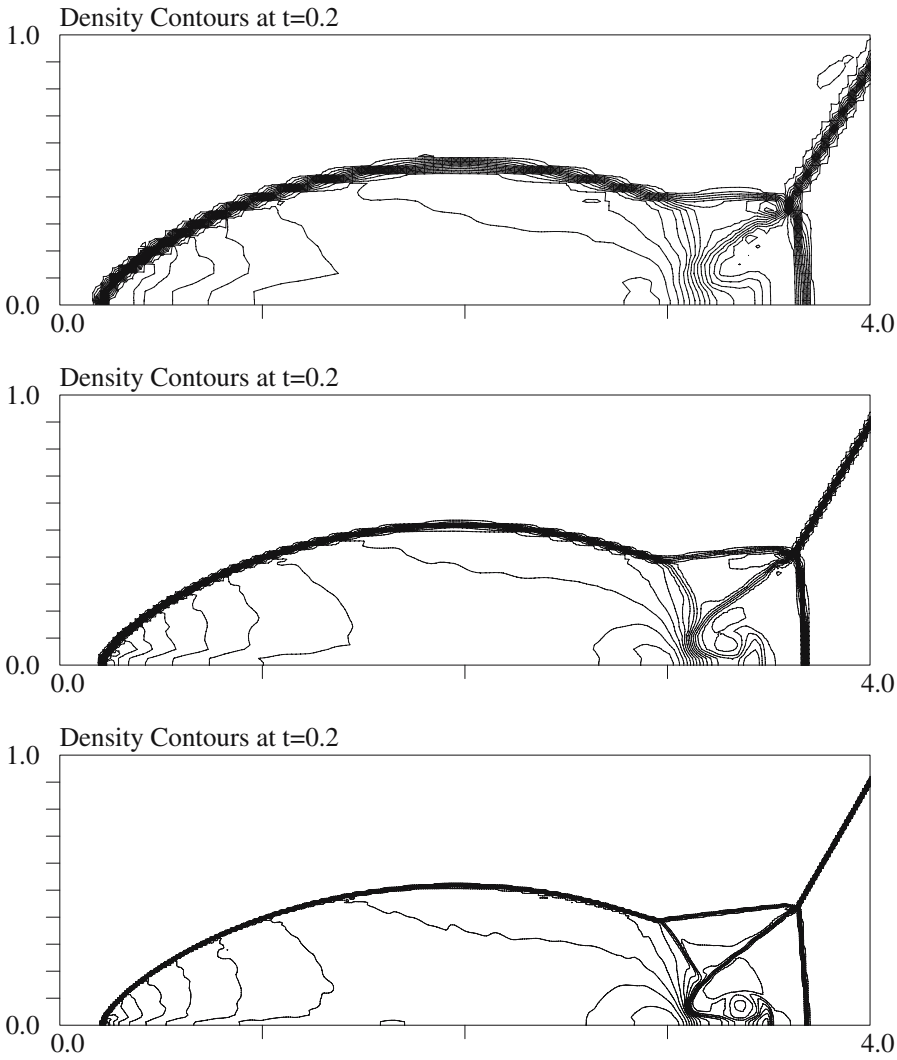


Fig. 26. Same as Fig. 23, but using a CV limiter sequentially on fully multidimensional antidiffusive fluxes

References

1. J. P. Boris. A fluid transport algorithm that works. In *Computing as a Language of Physics*, pages 171–189. International Atomic Energy Commission, 1971.
2. J. P. Boris and D. L. Book. Flux-Corrected Transport I: SHASTA, a fluid-transport algorithm that works. *Journal of Computational Physics*, 11:38–69, 1973.
3. A. J. Chorin. Random choice solution of hyperbolic systems. *Journal of Computational Physics*, 22:517–536, 1976.
4. A. J. Chorin. Random choice methods with application to reacting gas flow. *Journal of Computational Physics*, 25:252–272, 1977.
5. P. Colella and P. R. Woodward. The Piecewise-Parabolic Method (PPM) for gas-dynamical simulations. *Journal of Computational Physics*, 54:174–201, 1984.
6. C. R. DeVore. An Improved Limiter for Multidimensional Flux-Corrected Transport. NRL Memorandum Report 6440-98-8330, Naval Research Laboratory, Washington, DC, 1998.
7. C. K. Forester. Higher order monotonic convective difference schemes. *Journal of Computational Physics*, 23:1–22, 1977.
8. J. Glimm. Solution in the large for nonlinear hyperbolic systems of equations. *Communications on Pure and Applied Mathematics*, 18:697–715, 1955.
9. H.-O. Kreiss and J. Oliger. Comparison of accurate methods for the integration of hyperbolic equations. *Tellus*, 24:199, 1972.
10. B. E. McDonald. Flux-corrected pseudospectral method for scalar hyperbolic conservation laws. *Journal of Computational Physics*, 82:413, 1989.
11. G. A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 27:1–31, 1978.
12. P. R. Woodward and P. Colella. The numerical simulation of two-dimensional flow with strong shocks. *Journal of Computational Physics*, 54:115–173, 1984.
13. S. T. Zalesak. Fully multidimensional Flux-Corrected Transport algorithms for fluids. *Journal of Computational Physics*, 31:335–362, 1979.
14. S. T. Zalesak. Very high order and pseudospectral Flux-Corrected Transport (FCT) algorithms for conservation laws. In R. Vichnevetsky and R. S. Stepleman, editors, *Advances in computer methods for partial differential equations IV*, pages 126–134, Rutgers University, 1981. IMACS.
15. S. T. Zalesak. A preliminary comparison of modern shock-capturing schemes: Linear advection. In R. Vichnevetsky and R. S. Stepleman, editors, *Advances in computer methods for partial differential equations VI*, pages 15–22, Rutgers University, 1987. IMACS.

On Monotonically Integrated Large Eddy Simulation of Turbulent Flows Based on FCT Algorithms

Fernando F. Grinstein[1] and Christer Fureby[2]

[1] Laboratory for Computational Physics & Fluid Dynamics
US Naval Research Laboratory, Washington, DC 20375, USA
grinstein@lcp.nrl.navy.mil
[2] The Swedish Defense Research Agency, FOI
Dept. of Weapons and Protection, Warheads and Propulsion
SE-172 90, Stockholm, Sweden
fureby@foi.se

Summary. Non-classical Large Eddy Simulation (LES) approaches based on using the unfiltered flow equations instead of the filtered ones have been the subject of considerable interest during the last decade. In the Monotonically Integrated LES (MILES) approach, flux-limiting schemes are used to emulate the characteristic turbulent flow features in the high-wavenumber end of the inertial subrange region. Mathematical and physical aspects of implicit SGS modeling using non-linear flux-limiters are addressed using the modified LES-equation formalism. FCT based MILES performance is demonstrated in selected case studies including 1) canonical flows (homogeneous isotropic turbulence and turbulent channel flows), 2) complex free and wall-bounded flows (rectangular jets and flow past a prolate spheroid), 3) very-complex flows at the frontiers of current unsteady flow simulation capabilities (submarine hydrodynamics).

1 Background

High Reynolds (Re) number turbulent flows are of considerable importance in many fields of engineering, geophysics, and astrophysics. Turbulent flows involve multiscale space/time developing flow physics largely governed by large-scale vortical Coherent Structures (CS's). Typical turbulent energy spectra exhibit a large-wavelength portion dependent on the flow features imposed by geometry and boundary conditions, followed by an intermediate inertial subrange — which becomes longer for higher Re and characterizes the virtually inviscid cascade processes, and then by much-faster decaying portion in the dissipation region (e.g., Sect. 5.1.1 below).

Capturing the dynamics of all relevant scales based on the numerical solution of the Navier-Stokes Equations (NSE) constitutes Direct Numerical Simulation (DNS), which is prohibitively expensive for practical flows at moderate-to-high Re. On the other end of computer simulation possibilities, the industrial standard is Reynolds-Averaged Navier-Stokes (RANS) modeling, which involves simulating only the mean flow and modeling the effects of the turbulent scales.

Large Eddy Simulation (LES) is an effective intermediate approach between DNS and RANS, capable of simulating flow features which cannot be handled with RANS such as flow unsteadiness and strong vortex-acoustic couplings. Furthermore, LES provides higher accuracy than RANS at reasonable cost but still typically an order of magnitude more expensive. Desirable modeling choices involve selecting an appropriate discretization of the flow problem at hand, such that the LES cutoff lies within the inertial subrange, and ensuring that a smooth transition can be enforced at the cutoff. The main assumptions of LES are that: (i) transport is largely governed by large-scale unsteady features and that such dominant features of the flow can be resolved, (ii) the less-demanding accounting of the small-scale flow features can be undertaken by using suitable Sub Grid Scale (SGS) models.

In the absence of an accepted universal theory of turbulence to solve the problem of SGS modeling, the development and improvement of such models must include the rational use of empirical information. Several strategies to the problem of SGS modeling are being attempted, see e.g., [1], for a recent survey. After more than thirty years of intense research on LES of turbulent flows based on eddy-viscosity models there is now consensus that such approach is subject to fundamental limitations, [2]. It has been demonstrated, for a number of flows, that the eigenvectors of the SGS stress and rate-of-strain tensors involved in SGS eddy-viscosity models are not parallel, rendering eddy-viscosity models to be inaccurate.

There have been other proposals that do not employ the assumption of co-linearity of SGS stress and rate-of-strain embedded in the eddy-viscosity models, e.g. the scale-similarity model (SSM) [3] and the Approximate Deconvolution Method, (ADM), [4]. Such models may however be numerically unstable, and the more recent efforts have focused on developing mixed models, combining in essence the dissipative eddy-viscosity models with the more accurate but unstable SSM's. The results from such mixed models have been mostly satisfactory but the implementation and computational complexity of these improved combined approaches have limited their popularity. In fact, because of the need to distinctly separate (i.e. resolve) the effects of explicit filtering and SGS reconstruction models from those due to discretization, carrying out such well-resolved LES can typically amount in practice to performing a coarse DNS. As a consequence, it has been argued that the use of hybrid RANS/LES models for realistic whole-domain complex configurations might be unavoidable in the foreseeable future, e.g., [5].

Recognizing the aforementioned difficulties but also motivated by new ideas pioneered at NRL by Boris and collaborators, [6-7], several researchers have abandoned the classical LES formulations and started employing the unfiltered flow equations instead of the filtered ones. Major focus of the new approaches, [8-9], has been on the inviscid inertial-range dynamics and regularization of the under-resolved flow, based on *ab initio* scale separation with additional assumptions for stabilization, or applying monotonicity via nonlinear limiters that implicitly act as a filtering mechanism for the small scales – the original proposal of Boris *et al.* [7]. The latter concept goes back to the 50's to von Neumann and Richtmyer, [10], who used artificial dissipation to stabilize finite-difference simulations of flows involving shocks. This artificial dissipation concept also motivated Smagorinsky [11], in developing his scalar viscosity concept based upon the principles of similarity in the inertial range of 3D isotropic turbulence, [12].

In what follows, we use the modified LES equation formalism to carry out a formal comparative analysis of conventional LES and MILES. The performance of MILES is demonstrated for selected representative case studies including canonical flows, moderately complex free and wall-bounded flows, and extremely complex flows at the frontiers of current flow simulation capabilities. We conclude our presentation by addressing fundamental challenges for further development of the concept of nonlinear Implicit LES (ILES).

2 Conventional LES

For simplicity, we restrict the discussion to incompressible flows described by the Navier-Stokes momentum balance equation,

$$\partial_t(\mathbf{v}) + \nabla \cdot (\mathbf{v} \otimes \mathbf{v}) = -\nabla p + \nabla \cdot \mathbf{S}, \tag{1}$$

in conjunction with the incompressibility (or divergence) constraint $\nabla \cdot \mathbf{v} = 0$, where $\otimes$ denotes the tensorial product, and $\mathbf{S} = 2\nu\mathbf{D}$ and $\mathbf{D} = \frac{1}{2}(\nabla\mathbf{v} + \nabla\mathbf{v}^T)$ are the viscous-stress and strain-rate tensors. The conventional LES procedure, [1], involves three basic ingredients:

(i) low-pass filtering by the convolution

$$\bar{f}(\mathbf{x}, t) = G * f(\mathbf{x}, t) = \int_D G(\mathbf{x} - \mathbf{x}', \Delta) f(\mathbf{x}', t)\, d^3\mathbf{x}',$$

with a prescribed kernel $G = G(\mathbf{x}, \Delta)$ of width Δ,

(ii) finite volume, element or difference discretization,

(iii) *explicit* SGS modeling to close the low-pass filtered equations.

Applying (i) and (ii), using a second order accurate finite volume algorithm, to Equation (1), and rewriting the results in terms of the *modified equations*

approach, i.e., the equation satisfied by the numerical solutions being actually calculated yields, [13-14],

$$\partial_t(\bar{\mathbf{v}}) + \nabla \cdot (\bar{\mathbf{v}} \otimes \bar{\mathbf{v}}) = -\nabla \bar{p} + \nabla \cdot \bar{\mathbf{S}} - \nabla \cdot \mathbf{B} + \mathbf{m}^v + \tau, \tag{2}$$

where,

$$\mathbf{B} = \overline{\mathbf{v} \otimes \mathbf{v}} - \bar{\mathbf{v}} \otimes \bar{\mathbf{v}}, \qquad \mathbf{m}^v = [G*, \nabla](\mathbf{v} \otimes \mathbf{v} + p\mathbf{I} - \mathbf{S}),$$
$$\tau = \nabla \cdot \left[\left[\tfrac{1}{6}\nu\nabla^3\mathbf{v} - \tfrac{1}{8}\nabla^2\mathbf{v} \right] (\mathbf{d} \otimes \mathbf{d}) + \ldots \right] \tag{3}$$

are the SGS stress tensor, commutation error term, and the total (convective, temporal and viscous) truncation error, respectively, $\mathbf{I}$ is the unit tensor, and $\mathbf{d}$ is the topological vector connecting neighboring control volumes (see Fig. 1), and, $[G*, \nabla]f = \overline{\nabla f} - \nabla \bar{f}$. The commutation error term is often lumped together with the SGS force $\nabla \cdot \mathbf{B}$, prior to modeling, and hence a generalized SGS stress tensor $\mathbf{B}$ needs to be prescribed in terms of discretized filtered fields for closure of the new equations — which constitutes (iii) above.

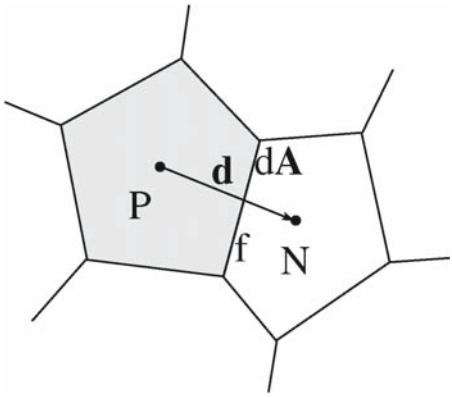


Fig. 1. Grid schematic. P and N denote typical computational cell centers and f an interface; $\mathbf{n}$ denotes a unit vector normal to the interface, and $\mathbf{A}$ its area; $\mathbf{d}$ is the topological vector connecting neighboring cells

Functional modeling consists of the modeling action of the SGS's on the resolved scales. It involves modeling of energetic nature, by which balances of energy are transferred between resolved and subgrid scale ranges, thus accounting for the SGS effects. The energy transfer mechanism from resolved to SGS's is assumed analogous to a Brownian motion superimposed on the large-scale motion. An example of this is the eddy-viscosity approach, in which $\mathbf{B} = -2\nu_k\bar{\mathbf{D}}$ where ν_k is the SGS viscosity — for example, using the Smagorinsky model, [11], or the one equation eddy-viscosity model (OEEVM), [15], its principal drawback is the well-established lack of colinearity between $\mathbf{B}$ and $\bar{\mathbf{D}}$.

Natural improvements to these models use anisotropic counterparts based on tensor forms of the SGS turbulent viscosity, [16]. These more sophisticated closures involve structural modeling, which attempts to model $\mathbf{B}$ without incorporating the interactions between SGS and resolved scales. By relying on actual SGS's in the upper resolved subrange — rather than on those modeled through dissipative eddy viscosity — we can better emulate scatter and backscatter, and the modeling procedures won't require assumptions on local isotropy and inertial range. Potential drawbacks arise, however, because structural models are computationally more expensive and typically not dissipative enough; accordingly, mixed models, combined with an eddy-viscosity model, are often used instead.

3 Implicit LES

A key self-consistency issue required in the conventional LES approach involves separating the computing effects of its three basic elements: filtering, discretization, and reconstruction. Filtering and reconstruction contributions must be resolved, i.e., their effective contributions in Equation (2) must be larger than the total truncation error τ. Also, their upper range of represented (but inaccurate) scales interactions must be addressed — in addition to those between resolved and SGS's. Thus, it is useful to examine $\mathbf{B}$ written in the following way,

$$\mathbf{B} = \overline{\mathbf{v} \otimes \mathbf{v}} - \bar{\mathbf{v}} \otimes \bar{\mathbf{v}} = (\overline{\mathbf{v} \otimes \mathbf{v}} - \overline{\mathbf{v}_P \otimes \mathbf{v}_P}) + (\overline{\mathbf{v}_P \otimes \mathbf{v}_P} - \bar{\mathbf{v}} \otimes \bar{\mathbf{v}}) = \mathbf{B}_1 + \mathbf{B}_2, \quad (4)$$

where $\bar{\mathbf{v}}_P$ denotes the (grid) represented velocity scales, $\mathbf{B}_1$ the interaction between represented and nonrepresented scales – which is not known a priori, and therefore must be modeled – whereas $\mathbf{B}_2$ relates to the interaction between filtered and discretized represented scales, and it can be approximated by prescribing an estimated $\mathbf{v}_P$ in the represented-velocity space (i.e., the solution to the so-called *soft deconvolution problem*), [4]. In this framework, a basic structural SGS model, such as the scale-similarity model, provides $\mathbf{B}_2$, and the eventual need of mixed models results from the recognition that $\mathbf{B}_2$ is not dissipative enough so a secondary regularization through $\mathbf{B}_1$ is needed — i.e., an approximation to $\mathbf{v}$ in physical-velocity space must be prescribed (the *hard deconvolution problem*).

Traditional approaches, motivated by physical considerations on the energy transfer mechanism from *resolved* to SGS's, express $\mathbf{B}_1$ with an appropriately functional model (for example, an eddy-viscosity SGS model), and seek sufficiently high-order discretization and grid resolution to ensure that effects due to τ are sufficiently small. However, we could argue that discretization could implicitly provide $\mathbf{B}_1$ if nonlinear stabilization can be achieved algorithmically via a particular class of numerical algorithms or based on regularizing the discretization of the conservation laws. In fact, Equation (2) suggests that

most schemes can potentially provide built-in or implicit SGS models enforced by the discretization errors τ, provided that their leading order terms are dissipative. We are thus led to the natural question: To what extent can we avoid the (explicit) filtering and modeling phases of LES (i.e., $\mathbf{B}_2 \equiv \mathbf{0}$ and $\mathbf{m}^v \equiv \mathbf{0}$) and focus on the implicit $\mathbf{B}_1$ provided by a suitably chosen discretization scheme?

Not all implicitly implemented SGS models are expected to work: the numerical scheme has to be constructed such that the leading order truncation errors satisfy physically required SGS properties, and hence non-linear discretization procedures will here be required. The analogy to be recalled is that of shock-capturing schemes designed under the requirements of convergence to weak solution while satisfying the entropy condition, [17].

4 Monotonically Integrated LES (MILES)

Nonoscillatory finite-volume (NFV) algorithms can be viewed as relevant for Implicit LES (ILES) of turbulent flows based on *nonlinear* implicit SGS modeling [14, 18], if we propose to focus on two distinct inherent *physical* SGS features to be emulated:

- the anisotropy of high-Re turbulent flows in the high-wave-number end of the inertial subrange region, characterized by very thin filaments of intense vorticity and largely irrelevant internal structure, embedded in a background of weak vorticity, e.g., [19],
- the particular (discrete) nature of laboratory observables (only finite fluid portions transported over finite periods of time can be measured) [18].

We thus require that ILES be based on NFV numerics having a *sharp velocity-gradient capturing capability* operating at the smallest resolved scales. By focusing on the inviscid inertial-range dynamics and on adaptive regularization of the under-resolved flow, ILES thus follows very naturally on the historical precedent of using this kind of schemes for shock capturing — in the sense that requiring emulation (near the cutoff) of the high wavenumber-end features of the inertial subrange region of turbulent flows is analogous to spreading the shock width to the point that it can be resolved by the grid.

Although the history of ILES draws on the development of shock-capturing schemes, the MILES concept — as originally introduced by Boris and his colleagues, [7], and further developed in our previous work, [13-14] — attempts to embody a computational procedure for solving the NSE as accurately as possible by using a particular class of flux-limiting schemes and their associated built-in (or implicit) SGS models. An intriguing MILES feature is the convection discretization that implicitly generates a nonlinear tensor-valued eddy-viscosity, which acts predominantly to stabilize the flow and suppress unphysical oscillations.

MILES draws on the fact that FV methods filter the NSE over nonoverlapping computational cells Ω_P, with the typical dimension $|\mathbf{d}|$ — using a top-hat-shaped kernel, $f_P = \frac{1}{\delta V_P}\int_{\Omega_P} f dV$. In the finite-volume context, discretized equations are obtained from the NSE using Gauss's theorem and by integrating over time with a multistep method parametrized by m, α_i, and β_i,

$$\begin{cases} \frac{\beta_i \Delta t}{\delta V_P}\sum\limits_f \left[F_f^{C,\rho}\right]^{n+i} = 0, \\ \sum\limits_{i=0}^{m}\left(\alpha_i(\mathbf{v})_P^{n+i} + \frac{\beta_i \Delta t}{\delta V_P}\sum\limits_f \left[F_f^{C,\rho}\mathbf{v}_f + \mathbf{F}_f^{D,v}\right]^{n+i} + \beta_i(\nabla p)_P^{n+i}\Delta t\right) = \mathbf{0}, \end{cases} \tag{5}$$

where α, β and m are parameters of the scheme, and $F_f^{C,\rho} = (\mathbf{v}\cdot d\mathbf{A})_f$ and $\mathbf{F}_f^{C,v} = F_f^{C,\rho}\mathbf{v}_f$ are the convective and $\mathbf{F}_f^{D,v} = (\nu\nabla\mathbf{v})_f d\mathbf{A}$ the viscous fluxes. To complete the discretization, all fluxes at face 'f' need to be reconstructed from the dependent variables at adjacent cells. This requires flux interpolation for the convective fluxes and difference approximations for the inner derivatives in the viscous fluxes.

For conventional LES, it is appropriate to use linear (or cubic) interpolation for the convective fluxes and central difference approximations for the inner gradients in the viscous fluxes. This then results in a cell-centered second- or fourth-order accurate scheme. Scheme stability can be enforced not only by conserving momentum, but also kinetic energy, which ensures robustness without numerical dissipation (which compromises accuracy).

Given Equation (5), the methods available for constructing implicit SGS models by means of the leading order truncation errors are generally restricted to nonlinear high-resolution methods for the convective flux $\mathbf{F}_f^{C,v}$ to maintain second-order accuracy in smooth regions of the flow (such high-resolution methods are at least second-order accurate on smooth solutions while giving well-resolved, nonoscillatory discontinuities), [17]. In addition, these schemes are required to provide a leading order truncation error that vanishes as $\mathbf{d} \to \mathbf{0}$ so that it remains consistent with the NSE and the conventional LES model. We focus here on the certain flux-limiting and correcting methods.

To this end, we introduce a flux-limiter Γ that combines a high-order convective flux-function $\mathbf{v}_f^H$ which is well-behaved in smooth flow regions, with a low-order dispersion-free flux-function $\mathbf{v}_f^L$, being well-behaved near sharp gradients, so that the total flux-function becomes $\mathbf{v}_f = \mathbf{v}_f^H - (1-\Gamma)[\mathbf{v}_f^H - \mathbf{v}_f^L]$. Choosing the particular flux limiting scheme also involves specific selections for $\mathbf{v}_f^L$ and $\mathbf{v}_f^H$. In the analysis that follows, $\mathbf{v}_f^H$ and $\mathbf{v}_f^L$ are assumed to be based on linear interpolation, and upwind-biased piecewise constant approximation, respectively, e.g.,

$$\begin{cases} \mathbf{F}_f^{C,v,H} = F_f^{C,\rho}[\ell\mathbf{v}_P + (1-\ell)\mathbf{v}_N - \frac{1}{8}(\mathbf{d}\otimes\mathbf{d})\nabla^2\mathbf{v} + O(|\mathbf{d}|^3)], \\ \mathbf{F}_f^{C,v,L} = F_f^{C,\rho}[\beta^+\mathbf{v}_P + \beta^-\mathbf{v}_N + (\beta^+ - \beta^-)(\nabla\mathbf{v})\mathbf{d} + O(|\mathbf{d}|^2)], \end{cases} \tag{6}$$

where $\beta^{\pm} = \frac{1}{2}(\mathbf{v}_f \cdot d\mathbf{A} \pm |\mathbf{v}_f \cdot d\mathbf{A}|)/|\mathbf{v}_f \cdot d\mathbf{A}|$, and $-\frac{1}{8}(\mathbf{d} \otimes \mathbf{d})\nabla^2\mathbf{v}$ and $(\beta^+ - \beta^-)(\nabla\mathbf{v})\mathbf{d}$ are the leading order truncation errors. The flux limiter Γ is to be formulated as to allow as much as possible of the correction $[\mathbf{v}_f^H - \mathbf{v}_f^L]$ to be included without increasing the variation of the solution — e.g., to comply with the physical principles of causality, monotonicity and positivity, [7], (when applicable) and thus to preserve the properties of the NSE. To see the effects of this particular convection discretization we consider the modified equations corresponding to the semi-discretized equations (5) with the flux-limiting functions in (6) being used for the convective fluxes,

$$\partial_t(\mathbf{v}) + \nabla \cdot (\mathbf{v} \otimes \mathbf{v}) = -\nabla p + \nabla \cdot \mathbf{S} + \nabla \cdot \Big[\mathbf{C}(\nabla\mathbf{v})^T + (\nabla\mathbf{v})\mathbf{C}^T + \chi^2(\nabla\mathbf{v})\mathbf{d} \otimes (\nabla\mathbf{v})\mathbf{d} + [\tfrac{1}{6}\nu\nabla^3\mathbf{v} - \tfrac{1}{8}\nabla^2\mathbf{v}](\mathbf{d} \otimes \mathbf{d}) + \ldots\Big], \tag{7}$$

with $\nabla \cdot \mathbf{v} = 0$, and where $\mathbf{C} = \chi(\mathbf{v} \otimes \mathbf{d})$ and $\chi = \frac{1}{2}(1-\Gamma)(\beta^- - \beta^+)$. In particular, we note that in smooth regions, $\Gamma = 1$ implies that $\chi = 0$ and $\mathbf{C} = \mathbf{0}$, and the leading order truncation error becomes $\tau = \nabla \cdot \big[[\frac{1}{6}\nu\nabla^3\mathbf{v} - \frac{1}{8}\nabla^2\mathbf{v}](\mathbf{d} \otimes \mathbf{d})\big]$. Comparing with the analysis of the momentum equation in the framework of the conventional LES approach (Equation (2)) suggests that the MILES modified equation incorporates additional dissipative and dispersive terms, and we can consistently identify the implicit SGS stress term,

$$\mathbf{B} = \mathbf{C}(\nabla\mathbf{v})^T + (\nabla\mathbf{v})\mathbf{C}^T + \chi^2(\nabla\mathbf{v})\mathbf{d} \otimes (\nabla\mathbf{v})\mathbf{d}. \tag{8}$$

The implicit SGS stress tensor can according to Equation (8) be decomposed into $\mathbf{B}^{(1)} = \mathbf{C}(\nabla\mathbf{v})^T + (\nabla\mathbf{v})\mathbf{C}^T$ and $\mathbf{B}^{(2)} = \chi^2(\nabla\mathbf{v})\mathbf{d} \otimes (\nabla\mathbf{v})\mathbf{d}$, in which the former is a tensor-valued eddy-viscosity model, while the latter is of a form similar to the scale similarity model. The decomposition in Equation (8) can also be interpreted as breaking $\mathbf{B}$ into its slow and rapid varying parts — relative to the time scale of its response to variations in the mean flow, [20]. In MILES, the rapid part that cannot be captured by isotropic models relates to $\mathbf{B}^{(2)}$, while the slow part relates to $\mathbf{B}^{(1)}$. Borue & Orszag, [21], have shown that a $\mathbf{B}^{(2)}$ type term improves the correlations between the exact and modeled SGS stress tensor. A closely-related view further explaining the effectiveness of ILES formulations based on local monotonicity (or sign) preservation concepts has been given by Margolin and Rider [18]; they argued that the leading order truncation error introduced by NFV algorithms represents a physical flow regularization term, providing necessary modifications to the governing equations that arise when the motion of *observables* — finite volumes of fluid convected over finite intervals of time — is considered.

Detailed properties of the implicit SGS model are related to the flux limiter Γ and to the choice of low- and high-order schemes; they also relate as well to other specific features of the scheme — e.g., such as monotonicity, l_1-contraction, local monotonicity preservation, and gridding. We have illustrated above in Equation (8) and discussed elsewhere, [13-14], how some of these properties can directly affect the implicit SGS modeling effectiveness

in the MILES context. MILES performance as a function of flux limiter is discussed further below; dependence on the choice of low order scheme has been examined in Ref. [22].

In what follows we address effects of variations in the flux-limiter Γ. To this end we consider first high-resolution schemes that can be formulated using the ratio of consecutive gradients,

$$r = \frac{\delta \mathbf{v}^n_{P-1/2}}{\delta \mathbf{v}^n_{P+1/2}} = \frac{(\mathbf{v}^n_P - \mathbf{v}^n_{P-1})}{(\mathbf{v}^n_{P+1} - \mathbf{v}^n_P)}.$$

Examples of well-known flux-limiters that fit into this category are:

1. the minmod flux-limiter of Roe, e.g. [23], with

$$\Gamma = \max(0, \min(1,\ r)),$$

2. the van-Leer flux-limiter, e.g. [23], with

$$\Gamma = \frac{r + |r|}{1 + |r|},$$

3. the superbee flux-limiter, Roe, e.g. [23], with

$$\Gamma = \max(0, \max(\min(2r, 1), \min(r, 2))),$$

4. the van-Albada flux-limiter, e.g. [24], with

$$\Gamma = \frac{r + r^2}{1 + r^2},$$

5. the GAMMA flux-limiter, e.g. [25], with

$$\Gamma = \frac{1-k}{k} r \left[\theta(r) - \theta\left(r - \frac{k}{1-k}\right)\right] + \theta\left(r - \frac{k}{1-k}\right),$$

where k is a parameter of the scheme such that $k \in [0, 1]$, and θ is the Heavyside function. Note that when $k = 0.5$ this scheme becomes TVD.

Some of these limiters are compared in Fig. 2 together with the r-independent limiting cases — the first-order upwind (UD, $\Gamma = 0$) and the second-order central difference (CD, $\Gamma = 1$) schemes. In Fig. 2 the TVD constraint

$$\mathrm{TV}(\mathbf{v}^{n+1}) \leq \mathrm{TV}(\mathbf{v}^n), \quad \text{where} \quad T\mathrm{V}(\mathbf{v}^n) = \sum\nolimits_P ||\mathbf{v}^n_{P+1} - \mathbf{v}^n_P||,$$

reformulated as $0 \leq |\Gamma(r), \Gamma(r)/r| \leq 2$, [23], is satisfied in the region bounded by the traces associated with the minmod and superbee limiters, which includes the van-Leer, van-Albada and ($k = 0.5$) GAMMA limiters. The diffusivity decreases as the flux-limiters approach that of the superbee limiter, which results in the least diffusive scheme.

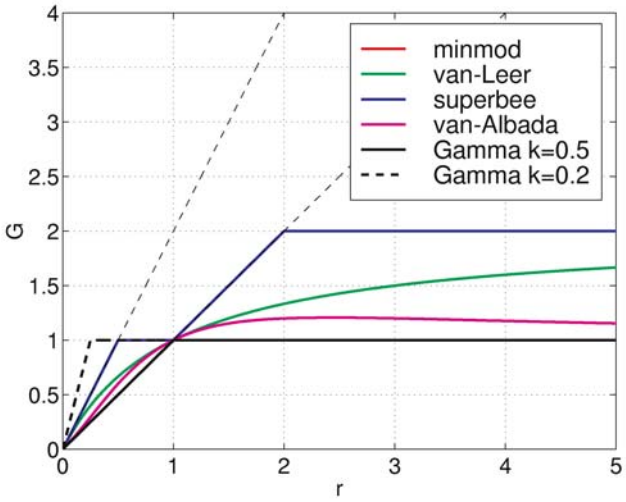


Fig. 2. TVD regions for first and second order accurate TVD schemes together with selected limiters

We can also consider other high-resolution schemes, such as FCT, [26], and PPM, [27], which can also use a similar flux-limiting type formalism based on $\mathbf{v}_f = \mathbf{v}_f^H - (1 - \Gamma)[\mathbf{v}_f^H - \mathbf{v}_f^L]$, but for which the flux limiter cannot simply be formulated in terms of the ratio of consecutive gradients, r. In the present study we have however only included the FCT-limiter, [26]. Some of these schemes (e.g. FCT, PPM) are locally monotonicity-preserving, i.e. given the solution $\mathbf{v}_P^{n+1} = H(\mathbf{v}_{P-k}^n, \mathbf{v}_{P-k+1}^n, \ldots, \mathbf{v}_{P+k}^n)$ then if $\mathbf{v}_P^0 \geq \mathbf{v}_{P+1}^0$ then $\mathbf{v}_P^n \geq \mathbf{v}_{P+1}^n$ for all P and n.

The global performance of MILES as a function of flux-limiter is documented in Fig. 3, showing studies of fully developed turbulent channel flow at a friction velocity based Re-number of $Re_\tau = 590$, compared with DNS results, [27]. The channel flow calculations will be discussed in detail below, and are carried out on 60^3 and 48^3 grids with uniform spacing the streamwise and spanwise directions. Periodic boundary conditions are employed in both streamwise and spanwise directions, together with no-slip conditions in the wall-normal directions. The influence of the flux limiter is comparatively small but has been observed to be sensitive to the wall-normal resolution (cf. [28]). From this comparison it is evident that the van-Leer limiter is too diffusive, producing poor velocity profiles, while both FCT and GAMMA produce velocity profiles that agree well with the reference DNS data. Reported ILES work includes, use of locally-monotonicity-preserving FCT [13-14] and PPM-based studies of homogeneous turbulence [29], locally-sign-preserving MPDATA [18], TVD-based turbulence studies, [30], and studies of channel flows using Godunov's exact Riemann solver, [31]. The following section discusses applications involving FCT-based MILES.

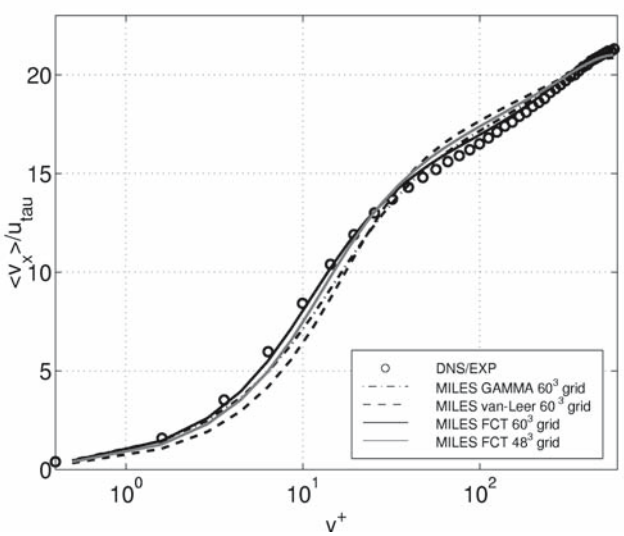


Fig. 3. MILES of $Re_\tau = 590$ channel flow; dependence on choice of flux-limiter

5 MILES Applications

The first category of applications discussed below comprises what we regard to be the traditional Computational Fluid Dynamics (CFD) role. For these cases, laboratory experiments can be carried out exhibiting the effects of flow dynamics and instabilities — typically demonstrating only the end outcome of complex three-dimensional (3D) physical processes. Time-dependent experiments based on numerical simulations with precise control of initial and boundary conditions, are ideally suited to supplement these laboratory studies, providing insights into the underlying flow dynamics and topology. Three groups of such examples are provided: i) canonical flows to demonstrate MILES benchmark studies; ii) mixing layer and jet flows to demonstrate MILES ability to capture complex flow physics; iii) external flows to demonstrate the MILES ability to deal with moderately complex geometries.

5.1 Canonical Cases

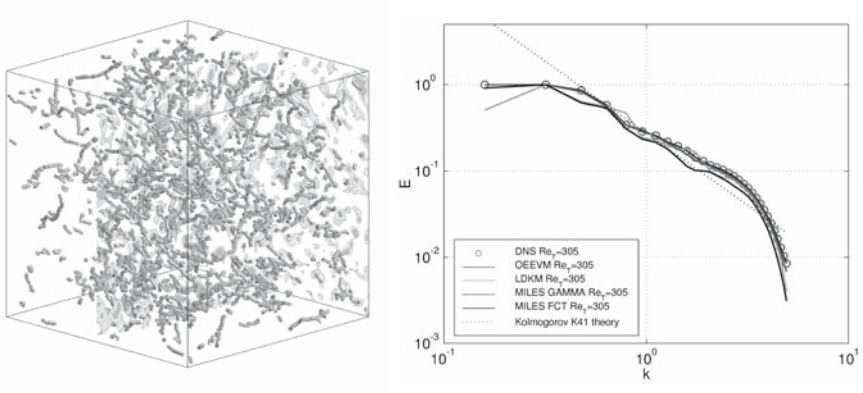
MILES of free shear flows have been extensively benchmarked and/or compared with laboratory flows in various different flow configurations at moderately-to-high Re numbers. Recent studies of forced and decaying homogeneous isotropic turbulence, studies of wakes, subsonic and supersonic mixing layers, jet flows, and channel flows demonstrated that MILES can be successfully used to simulate (and elucidate) the governing features of the unsteady flow dynamics. Selected representative examples are discussed in what follows.

5.1.1 Forced Homogeneous Turbulence

First, we consider forced homogeneous isotropic turbulence for a Taylor Re-number of $Re_T = 96$ at 32^3 and 64^3 resolution, for which DNS data is available, [19]. The body force, $\mathbf{f}$, is here used to create random forcing of the large scales. For this purpose we use the forcing scheme of Eswaran & Pope, [32], to drive the largest flow scales, see [33] for further details. The initial velocity field $\bar{\mathbf{v}} = \bar{\mathbf{v}}(\mathbf{x}, 0)$ is created by superimposing Fourier modes, having a prescribed energy spectrum but random phases.

Figure 4a shows typical visualizations at 64^3 resolution of the second invariant of the velocity gradient, i.e., $Q = 1/2(||\bar{\mathbf{W}}||^2 - ||\bar{\mathbf{D}}||^2)$. The observed vortical structure implies that weak and strong vortices have different topologies while there is no evident structure in the lower intensity regions. The higher intensity regions tend to be organized in slender tubes or elongated filaments as found with other LES studies. The vortical structures predicted by LES are considerably thicker than those obtained by DNS. However, filtering the DNS data results in thicker vortical structures, qualitatively very similar to those obtained by the LES. It is virtually impossible to distinguish between different SGS models by inspecting such visualizations, but by comparing Probability Density Functions (PDF's) of, e.g., the resolved vorticity magnitude $|\bar{\omega}|$ (not shown) we find that MILES and LDKM, [33-34] result in somewhat larger fractions of small-scale vorticity than conventional eddy-viscosity models such as the OEEVM, [15].

Figure 4b presents the time-averaged energy spectra for the $Re_T = 305$ case at 64^3 resolution together with a DNS spectrum, [19], and the theoretical model spectrum of Driscoll & Kennedy, [35]. The energy spectra are found to depend on the effects of the LES models only towards the high wavenumber



(**a**) (**b**)

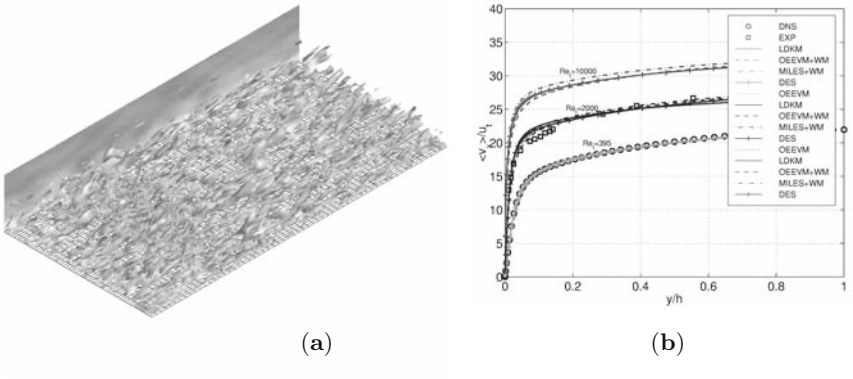
Fig. 4. Forced homogeneous isotropic turbulence: (**a**) iso-surfaces of Q; (**b**) energy spectra

end of the inertial range, and into the viscous subrange. A larger fraction of the turbulence is resolved in the 64^3 case as compared to the 32^3 results, and hence the influence of the SGS model is comparatively smaller in the former case. In general, we find that the spectra from MILES and LDKM are in better agreement with DNS data than, e.g., the OEEVM, at high wavenumbers.

5.1.2 Turbulent Channel Flow

Next we focus on fully developed turbulent channel flow at (bulk) Re-numbers between $Re = 15{,}000$ and $400{,}000$. The channel of length $6h$ and width $3h$ is confined between two parallel plates $2h$ apart, where h is the channel half-width. The flow is driven by a fixed mass flow in the streamwise ($\mathbf{e}_x$) direction. We use no-slip conditions in the cross-stream ($\mathbf{e}_y$) direction and periodic conditions in the ($\mathbf{e}_x$) and spanwise ($\mathbf{e}_z$) directions. The friction velocity is $u_\tau = \sqrt{\tau_w}$, where τ_w is the wall-shear stress. We vary the mass flow to obtain three target friction-velocity-based Re numbers: $Re_\tau \approx 395$, 2000, and 10,000 (the first corresponds to the DNS data, [27], and the second to the laboratory data, [36]). The grid consists of 60^3 cells with uniform spacing in the stream- and spanwise directions, whereas we use geometrical progression in the $\mathbf{e}_y$-direction to appropriately cluster the grid near the walls to resolve the velocity gradients.

Figure 5a shows the main channel flow features in terms of vortex lines, contours of the streamwise velocity component $\bar{v}_x$ and isosurfaces of the second invariant of the velocity gradient Q. By correlating isosurfaces of Q with the velocity fluctuations close to the wall, we found that vortices above the low-speed streaks are often ejected away from the wall, producing hairpin vortices stretched by the ambient shear. Using this mechanism, vorticity pro-



(**a**) (**b**)

Fig. 5. Fully developed channel flow: (**a**) instantaneous visualization in terms of contours of $\bar{v}_x$, vortex lines, and isosurfaces of Q; (**b**) vertical profiles of mean streamwise velocity $\langle \bar{v}_x \rangle$

duced in the viscous region is advected into the boundary layer, making it turbulent. The spanwise resolution appears more important for accurate large-scale-dynamics prediction than the streamwise resolution. The wall-normal resolution is critical for the correct wall-shear stress τ_w prediction, which, in turn, is important for making correct estimates of, for example, the drag.

In Figure 5b, we compare LES mean velocity $\langle \bar{v}_x \rangle$ predictions (integrated over time, x, and z) with DNS data, [27], and experimental data, [36]. For $Re_\tau = 395$, all LES models used show very good agreement with the DNS data. When the flow is well resolved, the SGS model details are of little importance to the resolved flow, because most of the energy (about 98 percent) and structures are resolved on the grid. For $Re_\tau \approx 1800$, we still see good agreement between LES and experimental data, but with somewhat larger scatter in the LES data. This case is well resolved, with about 87 percent of the energy belonging to the resolved scales. We do not have any data to compare with, for $Re_\tau = 10,000$, but we may compare it (asymptotically) with the lower Re_τ number velocity profiles and the log law. The scatter among the LES models is now larger, and we find that the best agreement with the log law is obtained by using Detached Eddy Simulation (DES), [37], and the localized dynamic (LDKM) subgrid turbulence model, [34], followed, in turn, by MILES+WM, OEEVM+WM, [38], and OEEVM, [15], where WM denotes the wall model, [38]. However, for the second order statistical moments, MILES+WM and OEEVM+WM provide better agreement with data.

The eddy-viscosity models are successful because ν_k responds to energy accumulation in the small scales by adjusting the dissipation before it contaminates the resolved scales. MILES performs well because it mimics the resolved flow anisotropies. MILES turns out also to be computationally competitive, with typical work figures of OEEVM = 1.00, MILES = 0.95, OEEVM+WM = 1.05, DES = 1.10, and LDKM = 1.15.

5.2 Free Shear Flows: Global Instabilities and Vorticity Dynamics

Characterizing the local nature of free-shear-flow instabilities and their global nonlinear development in space and time is of fundamental importance for practical shear-flow control. Linear inviscid stability analyses have shown the convectively unstable nature of the spatially evolving subsonic mixing layer with respect to vortical fluctuations. Consequently, except in rare configurations with global-absolute instabilities, we expect environmental disturbances to drive turbulent mixing layers; self-sustained instabilities have generally not been expected. Mechanisms influencing re-initiation of the instabilities and transition to turbulence in free-shear flows are:

- disturbances in the free streams,
- disturbances due to boundary layers, wakes, small recirculation zones, or acoustic environmental disturbances, and
- disturbances fed back from downstream events in the flow.

Isolating these mechanisms is difficult because turbulence in free streams and boundary layers cannot be eliminated. Numerical simulations of spatially evolving shear flows can be essentially eliminated, the first two disturbances and the third can be addressed through careful control of the imposed boundary conditions.

5.2.1 Global Instabilities in Free Flows

An important question is whether a free mixing layer can be globally unstable with the self-excitation upstream induced by pressure disturbances generated via finite-amplitude fluid accelerations downstream. A previous study successfully addressed this question with FCT based MILES of spatially evolving flows, [39]. Numerical simulations of compressible, subsonic, planar shear-flows were used to investigate the role of feedback in the re-initiation of the vortex roll-up. The study dealt with unforced, spatially evolving mixing layers for which the acoustic disturbance propagation can be appropriately resolved; and boundary effects were ensured to be negligible. The simulation shows global self-sustaining instabilities, in which new vortex roll-ups were triggered in the initial shear layer by pressure disturbances originating in the fluid accelerations downstream. This re-initiation mechanism, absent in linear treatments of stability, was demonstrated conclusively and examined as a function of Mach number and free-stream velocity ratio.

Another study demonstrated similar self-excited global instabilities in supersonic, countercurrent jets, based on upstream feedback mechanisms acting on the subsonic outer jet regions (see Fig. 6), [18]. Recognition of these global instabilities provided new insights to explain previously unresolved discrep-

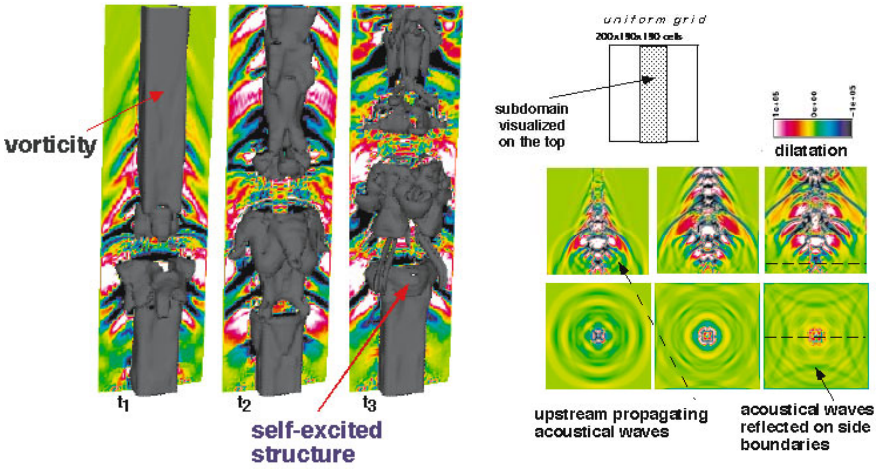


Fig. 6. MILES studies of global instabilities in a countercurrent supersonic cold square jet in terms of instantaneous visualizations, [40]

ancies between laboratory and theoretical studies, suggesting practical approaches to active control of these jets. A key computational capability used in both of these global instability studies was the ability to isolate the generation of propagation of acoustical disturbances correlated with the large-scale vortex dynamics. Relevant features accurately captured with MILES included the quadrupole pattern of acoustic production associated with vortices, the significantly more intense dilatation and pressure fluctuations associated with vortex pairing, as well as the very low fluctuation levels involved (for example, four orders of magnitude smaller than ambient values, [39-40]).

Accurate resolution of the small characteristic fluctuation levels, typically associated with acoustical radiation from flow accelerations, involves major challenges:

- Complex vortex dynamics associated with acoustical production must be captured.
- The numerical algorithm's dispersiveness should be minimized to ensure good modeling of the acoustical propagation properties of the small wavelengths.
- Because of the very small energy of the acoustic field compared to that of the flow field, there is a potential for spurious sound sources due to numerical discretization.

Because of the tensorial nature of its implicit SGS model, and the inherently low numerical diffusion involved, the use of flux limiting in MILES offers an overall effective computational alternative to conventional SGS models in this context. MILES was used to extensively investigate the natural mechanisms of transition to turbulence in rectangular jets evolving from laminar conditions, [41], in compressible (subsonic) jet regimes with aspect ratio $AR = 1$ to 4 and moderately high Re. The studies demonstrated qualitatively different dynamical vorticity geometries characterizing the near jet, involving

- self-deforming and splitting vortex rings,
- interacting ring and braid (rib) vortices, including single ribs aligned with corner regions ($AR \geq 2$), and rib pairs (hairpins) aligned with the corners ($AR = 1$), and,
- a more disorganized flow regime in the far jet downstream, where the rotational fluid volume is occupied by a relatively weak vorticity background with strong, slender tube-like filament vortices filling a small fraction of the domain.

Figure 7a illustrates characteristic axis-switching and bifurcation phenomena from visualizations of laboratory elliptical jets subject to strong excitation at the preferred mode, [42]. We compare it to the carefully developed simulation results (see Figs. 7b and 7c) designed to address unresolved issues in vortex dynamics. Detailed key aspects — namely, reconnection, bridging, and threading (see Fig. 7b) — could not be captured in the laboratory studies and were first demonstrated by the simulations.

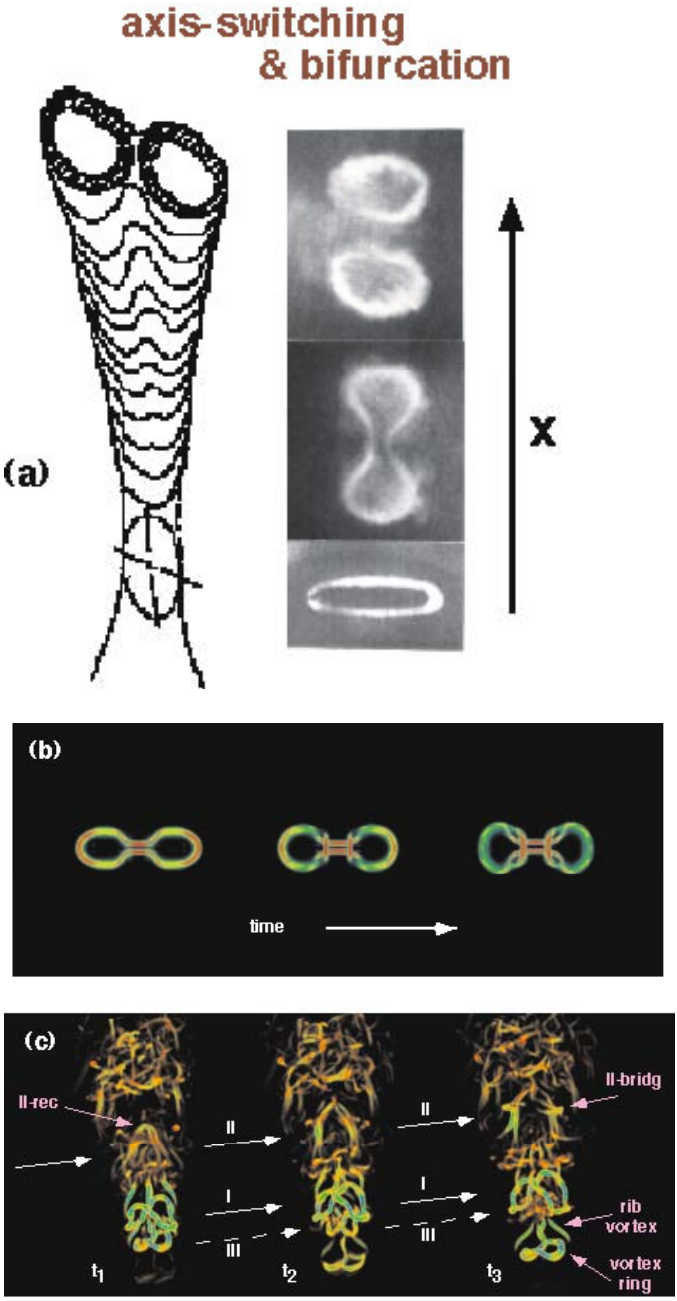


Fig. 7. Vortex dynamics and transition to turbulence in subsonic noncircular jets; (**a**) laboratory studies, [42], (**b**) and (**c**) detailed vortex dynamics elucidated by simulations, [41]

Jet flows develop in different possible ways, depending on

- their particular initial conditions,
- nozzle geometry and modifications introduced at the jet exit,
- the types of unsteady vortex interactions initiated, and
- local transitions from convectively to absolutely unstable flow.

Taking advantage of these flow control possibilities is of interest to improve the mixing of a jet, or plume, with its surroundings in practical applications demanding:

- enhanced combustion between injected fuel and background oxidizer,
- rapid initial mixing and submergence of effluent fluid,
- less intense jet noise radiation,
- reduced infrared plume signature.

For example, the jet entrainment rate — the rate at which fluid from the jet becomes entangled or mixed with that from its surroundings — can be largely determined by the characteristic rib-ring coupling geometry and the vortex-ring axis-switching times (see Fig. 8), [40].

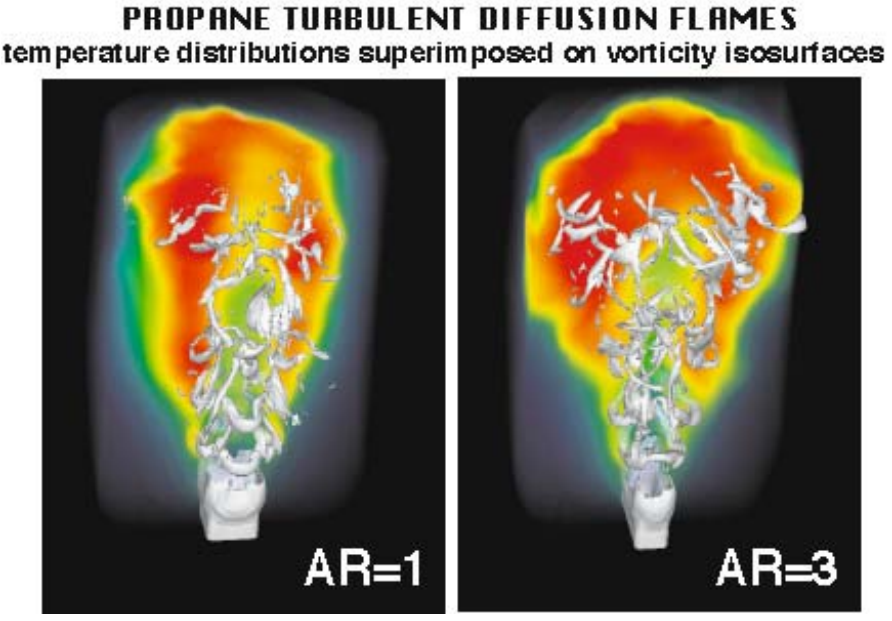


Fig. 8. Visualizations of non-premixed combustion regions as a function of aspect ratio, [41]. Temperature distributions (color) in the back half of the visualized subvolume are superimposed to isosurfaces of the vorticity magnitude (gray)

5.3 Moderately Complex Geometry: Flow over a Prolate Spheroid

Crucial additional issues of LES of inhomogeneous high-Re flows to be addressed relate to boundary condition (supergrid) modeling and overall computational model validation. From the practical point of view, it is of utmost importance to consider how the non-linear combination of all — algorithmic, physics-based, SGS, and supergrid — aspects of the model affect the simulation of complex systems for which detailed DNS-type approaches are not possible and for which only limited experimental data might be available at best.

Despite its simple geometry, the flow around a prolate spheroid at an incidence (see Fig. 6a) contains a rich gallery of complex 3D flow features. These include:

- stagnation flow,
- 3D boundary layers under influence of pressure gradients and streamline curvature
- cross-flow separation, and
- the formation of free vortex sheets producing streamwise vortices.

These features are archetypes of flows over more complicated airborne and underwater vehicles warranting in-depth study. Previously, [42], we studied the flow around a 6:1 prolate spheroid mounted in a wind tunnel with a rectangular cross-section, [44], at $\alpha = 10°$ and $20°$ angles of attack. Based on the freestream velocity v_0 and the body length L, the Re number is $Re_L = 4.2{\cdot}10^6$. The domain is discretized with a block-structured mesh, supported by a double O-shaped block structure. Two meshes are used in order to parameterize effects of the grid on the boundary layer resolution. Mesh A has $0.75{\cdot}10^6$ cells and $y^+ \approx 25$ and mesh B has $1.50{\cdot}10^6$ cells with $y^+ \approx 5$. At the inlet, $\bar{\mathbf{v}} = v_0\mathbf{n}$ and $\partial\bar{p}/\partial n = 0$, where $\mathbf{n}$ is the outward pointing unit normal, and at the outlet $\bar{p} = p_\infty$ and $\partial(\bar{\mathbf{v}} \cdot \mathbf{n})/\partial n = 0$. On the body, no-slip conditions are used.

Figure 9a shows perspective views from the port side of the prolate spheroid at $\alpha = 20°$. The flow is represented by surface streamlines, streamribbons, and contours of the vorticity magnitude $|\bar{\omega}|$ at $x/L = 0.772$, where $\bar{\omega} = \frac{1}{2}\nabla \times \bar{\mathbf{v}}$ is the vorticity. The streamribbons show the complexity of the flow. On the windward side, an attached 3D boundary layer is formed, while on the leeward side, the flow detaches from the hull — because of the circumferentially adverse pressure gradient, and rolls up into a counterrotating pair of longitudinal spiraling vortices on the back of the body. Furthermore, fluid from the windward side is advected across the spheroid, engulfed into the primary vortices and subsequently ejected into the wake.

Figure 9b shows the time-averaged velocity components (U, V, W) at $x/L = 0.772$ and at $\varphi = 90°$. The velocity components are presented in the body-surface coordinate system, [44]. For V and W, we see good agreement between predictions and measurements for all models — with DES providing the least accurate comparison. We obtained the best agreements

with OEEVM and MILES with a wall-model, [38], on grid A (OEEVM+WM and MILES+WM). Concerning U, we found significant differences as a function of the various models and grid resolutions. We found best agreements for MILES+WM and OEEVM+WM, whereas the LDKM and DES predictions show larger deviations from the experimental data. The LDKM appears to require better resolution than what we have provided because it underpredicts the boundary layer thickness. The results from MILES+WM and OEEVM+WM appear virtually unaffected by resolution, which is expected because the wall model is designed to take care of the errors introduced by poor resolution in the boundary layer. Also interesting is that the effects of changing the angle of attack α — very important when studying, for example, maneuvering — are very well reproduced in the simulations.

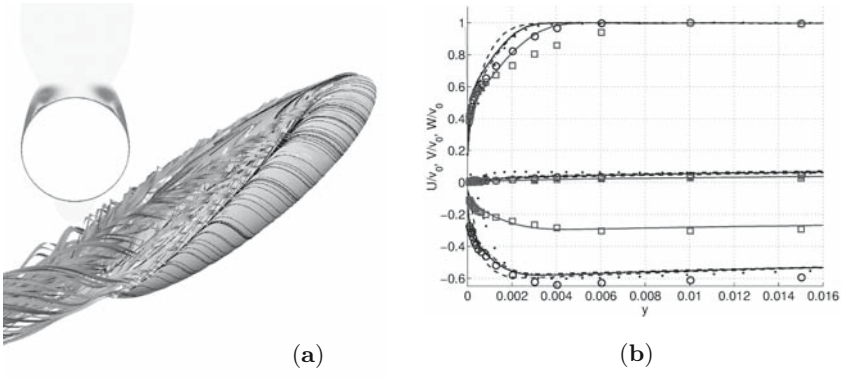


Fig. 9. Flow around a prolate spheroid: (**a**) perspective view and contours of the vorticity magnitude $|\omega|$ at $x/L = 0.772$; (**b**) velocity comparison at $x/L = 0.772$ and $\varphi = 60^\circ$ between experimental (O and □) and predicted data at $\alpha = 10^\circ$ (black) and $\alpha = 20^\circ$ (gray). (—) OEEVM+WM on grid A, (- - -) MILES+WM on grid A, (-.-.) LDKM on grid A, (····) LDKM on grid B and ($\cdot\ \cdot\ \cdot$) DES on grid A

6 Challenging New Role of Simulations

For the studies of submarine hydrodynamics and flows in urban areas discussed separately in this volume [22], it is unlikely that we will ever have a deterministic predictive framework based on computational fluid dynamics. This is due to the inherent difficulty in modeling and validating all the relevant physical sub-processes and acquiring all the necessary and relevant boundary condition information. On the other hand, these cases are representative of very fundamental ones for which whole-domain scalable laboratory studies are impossible or very difficult, but for which it is also crucial to develop predictability.

5.4.1 Submarine Hydrodynamics

The flow around a submarine is extremely complicated and characterized by very high Re, $O(10^9)$. Full-scale experiments are complicated and very expensive and are of limited value due to the difficult measurement settings. RANS of full-scale submarine hydrodynamics are barely within reach, whereas LES is currently out of reach due to the wide range of scales present. For model-scale situations ($Re \approx 10^7$), it might be possible to conduct LES and DES, [45]. In particular, if we're interested in vortex dynamics, flow noise, and the coupling between the propeller dynamics and the flow around the hull, LES and DES are our only alternatives for the foreseeable future.

As Figure 7a shows, each appendage generates a wake and several vortex systems. A horseshoe-vortex pair is formed in the junction between the hull and the appendage, whereas a tip-vortex pair is formed at the tip of the appendage. Additional vortex systems can be formed, e.g., on the side of the sail towards the trailing edge or in the boundary layer of the tapered sections of the hull. These vortex systems can interact with each other and with the (unsteady) boundary layer to form a very complex flow entering the propeller, thus causing vibrations and noise. In addition, the ocean water is usually stratified, with density variations caused by differences in temperature and salinity between ocean currents, or between the surface water and deeper water. Stratification influences the turbulence and the large flow structures in the wake, typically resulting in horizontally flattened flow structures (so-called pancake vortices), which would not occur in nonstratified waters.

The case discussed here is the fully appended DARPA Suboff configuration, [46], constructed from analytical surfaces, and shown in Fig. 10. Experimental data, using hot-film techniques, are provided at $Re = 12 \cdot 10^6$ based on the overall hull length L, the freestream velocity u_0 and ν, [47]. The total measurement uncertainty in the velocity data — i.e., the geometrical mean of the bias and precision errors, is estimated to be about 2.5% of u_0. The computational domain consists of the submarine model mounted in a cylinder having the same hydraulic diameter as the wind tunnel used in the scale model experiments. The cylinder extends one hull-length upstream and two hull-lengths downstream, thus being $4L$ in overall length. For the hull an O-O topology is used, while for the sail and stern appendages C-O topologies are used and care is taken to ensure that the cell spacings and aspect ratios are suitable for capturing the boundary layers along the hull.

Typically, about 20 cells are contained within the thickness of the boundary layer on the parallel midsection of the hull, having a typical wall distance for the first cell $y^+ \approx 8$. Two grids of about $3{\cdot}10^6$ and $6 \cdot 10^6$ nodes were used. At the inlet boundary, $\bar{\mathbf{v}} = u_0\mathbf{n}$ and $(\nabla\bar{p} \cdot \mathbf{n}) = 0$, at the outlet $\bar{p} = p_0$ and $(\nabla\bar{\mathbf{v}}) \cdot \mathbf{n} = \mathbf{0}$, whereas free-slip conditions are used at the wind-tunnel walls, and no-slip conditions are used on the hull. All LES are initiated with quiescent conditions and the unsteady flow in LES is allowed to evolve naturally (i.e., without any external forcing).

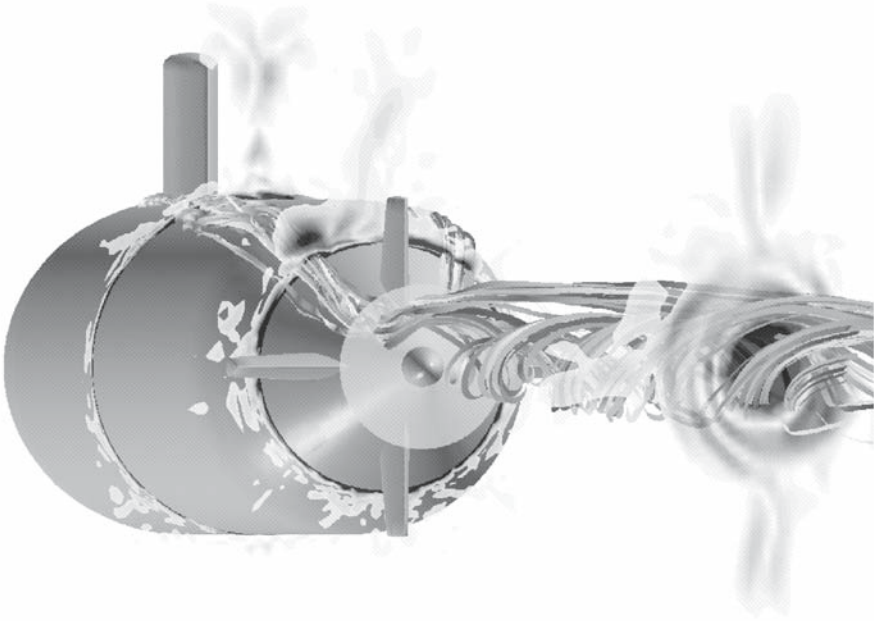
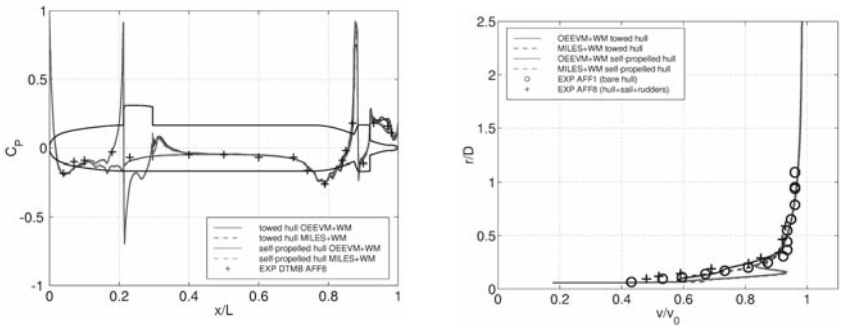


Fig. 10. Submarine hydrodynamics: the main flow features represented by stream-ribbons and contour plots of the vorticity magnitude in three cross-sections

In Figures 11a and 11b we show typical comparisons between predictions of towed and self-propelled cases and experimental data, [47], of the distribution of the time-averaged static pressure coefficient $C_P = 2(\langle\bar{p}\rangle - p_0)/u_0^2$ along the meridian line of hull and of the circumferentially averaged velocity in the propeller plane. Very good agreement between the measurement data and the computations is observed along the entire hull section for the towed case. Virtually no differences in the C_P distribution can be observed between the towed and the self-propelled cases — with the exception of the far-end of the tapered section of the stern, nor do we see significant differences between the MILES+WM and LDKM predictions. Concerning the velocity distributions, the differences are attributed to the presence of the propeller (or rather the actuator-disc used to model the effects of the propeller), and show the effects of the axial pressure gradient, as implicitly imposed by the propeller causing a suction effect along the stern part of the hull. Based on the secondary velocity vector field (not shown) the location of the horseshoe-vortex pair is estimated in the case of the towed case from predictions (measurements) to be at $r/R \approx 0.41$ (0.38) and $\varphi \approx \pm 23^\circ$ ($\pm 22^\circ$), respectively.



(**a**) (**b**)

Fig. 11. Submarine hydrodynamics: comparison of the mean pressure and mean velocity: (**a**) along the meridian line of the hull; (**b**) in the propeller plane

7 Outlook

In the absence of an accepted universal theory of turbulence, the development and improvement of SGS models are unavoidably pragmatic and based on the rational use of empirical information. Classical approaches have included many proposals ranging from inherently limited eddy-viscosity formulations to more sophisticated and accurate mixed models. The main drawback of mixed models relates to their computational complexity, and ultimately, to the fact that well-resolved (discretization-independent) LES is prohibitively expensive for the practical flows of interest at moderate-to-high *Re*. This has recently led many researchers to abandon the classical LES formulations, shifting their focus directly to the SGS modeling implicitly provided by nonlinear stabilization achieved algorithmically, through the use of a particular class of numerical schemes, or based on regularization of the discretization of conservation laws.

In MILES, the effects of SGS physics on the resolved scales are incorporated in the functional reconstruction of the convective fluxes using locally-monotonic methods. Analysis based on the modified equations shows that MILES, based on a particular class of flux-limiting schemes, provides an implicitly implemented anisotropic SGS model dependent on the specifics of the particular numerical scheme — i.e., on the flux limiter Γ, on the choice of low- and high-order schemes, and on the gridding.

MILES seeks to emulate the flow features in the high wave number end of the inertial subrange of turbulent flows — characterized by thin filaments of intense vorticity embedded in a background of weak vorticity. We propose that emulation of the latter feature be the requirement for the more general concept of nonlinear implicit SGS modeling in the context of finite-volume formulations. In ILES, the functional reconstruction of the convective flux functions is carried out via high-resolution nonlinear numerical schemes incorporating a sharp velocity-gradient capturing capability operating at the smallest resolved

scales. By focusing on the inviscid inertial-range dynamics and on regularization of the underresolved flow, ILES follows up very naturally on the historical precedent of using this kind of numerical schemes for shock-capturing. Challenges for ILES include constructing a common appropriate mathematical and physical framework for its analysis and development, further understanding the connections between implicit SGS model and numerical schemes, and, in particular, how to address building physics into the numerical scheme to improve global ILES performance, i.e., on the implicitly-implemented SGS dissipation & backscatter features. Moreover, additional (explicit) SGS modeling might be needed to address inherently small-scale physical phenomena such as scalar mixing and combustion — which are actually outside the realm of any LES approach: how do we exploit the implicit SGS modeling provided by the numerics, to build efficient 'mixed' (explicit/implicit) SGS models?

8 Acknowledgements

This work was completed while one of us (FFG) was Orson Anderson Distinguished Visiting Scholar at IGPP, Los Alamos National Laboratory, on Sabbatical leave from the US Naval Research Laboratory. Support from the Los Alamos National Laboratory base research program and from the Office of Naval Research through the Naval Research Laboratory 6.1 Computational Physics task area are also greatly appreciated.

References

1. Sagaut P.; 2002, "Large Eddy Simulation for Incompressible Flows", Springer, New York.
2. Liu S., Meneveau C. and Katz J.; 1994, "On the Properties of Similarity Subgridscale Models as Deduced from Measurements in a Turbulent Jet", J. Fluid Mech., **275**, p 83.
3. Bardina J.; 1983, "Improved Turbulence Models based on Large Eddy Simulation of Homogeneous Incompressible Turbulent Flows", Ph.D. Thesis, Stanford University.
4. Adams N.A. and Stolz S.; 2001, "Deconvolution Methods for Subgrid-Scale Approximation in LES", in *Modern Simulation Strategies for Turbulent Flows,* B.J. Geurts, ed., Edwards, Philadelphia, p.21.
5. Spalart P.R., Jou, W.H., Strelets M. and Allmaras S.R.; 1997, "Comments on the Feasibility of LES for Wings, and on Hybrid RANS/LES Approach", in *Advances in DNS/LES*, First AFOSR International Conference in DNS/LES (Greyden Press, Columbus).
6. Boris J.P.; 1989, "On Large Eddy Simulation Using Subgrid Turbulence Models", in *Whither Turbulence? Turbulence at the Crossroads,* J.L. Lumley Ed., Springer, New York, p 344.
7. Boris J.P., Grinstein F.F., Oran E.S. and Kolbe R.J.; 1992, "New insights into Large Eddy Simulations", Fluid Dynamics Research, **10**, p 199.

8. Alternative LES and Hybrid RANS/LES; 2002, edited by F.F. Grinstein and G.E. Karniadakis, J. Fluids Engineering, **124**, pp. 821-942.
9. Special Issue of Internat. J. Numer. Methods in Fluids, **39**, (2002), ed. by D. Drikakis, pp. 763-864.
10. von Neumann J. and Richtmyer R.D.; 1950, "A Method for the Numerical Calculation of Hydrodynamic Shocks", J. Applied Physics, **21**, p 232.
11. Smagorinsky J.; 1983, "The Beginnings of Numerical Weather Prediction and General Circulation Modeling: Early Recollections", Advances in Geophysics, **25**, p 3.
12. Smagorinsky, J.; 1963, "General Circulation Experiments with the Primitive Equations. I. The Basic Experiment", Monthly Weather Rev., **101**, p 99.
13. Fureby C. and Grinstein F.F.; 1999, "Monotonically Integrated Large Eddy Simulation of Free Shear Flows", AIAA. J., **37**, 544.
14. Fureby C. and Grinstein F.F.; 2002, "Large Eddy Simulation of High Reynolds Number Free and Wall Bounded Flows", J. Comp. Phys. **181**, p 68.
15. Schumann U.; 1975, "Subgrid Scale Model for Finite Difference Simulation of Turbulent Flows in Plane Channels and Annuli", ‚J. Comput. Phys., **18**, p. 376.
16. Carati D., Winckelmans G.S. & Jeanmart H.; 1999, "Exact Expansions for Filtered Scales Modeling with a wide class of LES Filters", in *Direct and Large Eddy Simulation III,* Eds. Voke, Sandham & Kleiser, p 213.
17. Godunov S.K.; 1999, "Reminiscences About Difference Schemes", J. Comp. Phys., **153**, pp. 6-25.
18. Margolin L.G. and Rider W.J.; 2002, "A Rationale for Implicit Turbulence Modeling", J. Numer. Methods in Fluids, **39**. p 821.
19. Jimenez J., Wray A., Saffman P., and Rogallo R.; 1993, "The Structure of Intense Vorticity in Isotropic Turbulence", J. Fluid Mech., **255**, p 65.
20. Shao L., Sarkar S. and Pantano C.; 1999, "On the Relationship Between the Mean Flow and Subgrid Stresses in Large Eddy Simulation of Turbulent Shear Flows", Phys. Fluids, **11**, p 1229.
21. Borue V. and Orszag S.A.; 1998, "Local Energy Flux and Subgrid-Scale Statistics in Three Dimensional Turbulence", J. Fluid Mech., **366**, p 1.
22. Patnaik G., Boris J.P., Grinstein F.F., and Iselin J.P.; 2004, "Large Scale Urban Simulations with the MILES Approach", Chapter 4 in this volume.
23. Hirsch C.; 1999, "Numerical Computation of Internal and External Flows", J. Wiley and Sones.
24. Albada, G.D., van Leer, B., van Roberts, W.W.; 1982, "A Comparative Study of Computational Methods in Cosmic Gas Dynamics", Astron. Astrophys. , **108**, p 76.
25. Jasak H., Weller H.G., and Gosman A.D.; 1999, "High Resolution NVD Differencing Scheme for Arbitrarily Unstructured Meshes", Int. J. Numer. Meth. Fluids, **31**, p 431.
26. Boris J.P. & Book D.L.; 1973, "Flux Corrected Transport I, SHASTA, a Fluid Transport Algorithm that Works", J. Comp. Phys. **11**, p 38.
27. Colella P, & Woodward P.; 1984, "The Piecewise Parabolic Method (PPM) for Gas Dynamic Simulations", J. Comp. Phys., **54**, p 174.
28. Moser R.D., Kim J. & Mansour N.N.; 1999, "Direct Numerical Simulation of Turbulent Channel Flow up to $Re_\tau = 590$", Phys. Fluids, **11**, p 943.
29. Porter D.H., Pouquet A. and Woodward P.R.; 1994, "Kolmogorov-like Spectra in Decaying Three-Dimensional Supersonic Flows", Phys. Fluids, **6**, p 2133.

30. Garnier E., Mossi M., Sagaut P., Comte P., and Deville M.; 2000, "On the Use of Shock-Capturing Schemes for Large Eddy Simulation", J. Comp. Phys., **153**, p. 273.
31. Okong'o N., Knight D.D., and Zhou G.; 2000, "Large Eddy Simulations Using an Unstructured Grid Compressible Navier-Stokes Algorithm", Int. J. Comp. Fluid Dyn., **13**, p. 303.
32. Eswaran V. and Pope S.B.; 1988, "An Examination of Forcing in Direct Numerical Simulation of Turbulence", Computers and Fluids, **16**, p 257.
33. Fureby C., Tabor G., Weller H. and Gosman D.; 1997, "A Comparative Study of Sub Grid Scale Models in Isotropic Homogeneous Turbulence", Phys. Fluids, **9**, p 1416.
34. Kim W.-W. and Menon S.; 1999, "A new Incompressible Solver for Large-Eddy Simulations", Int. J. Num. Fluid Mech., **31**, p 983.
35. Driscoll R.J. and Kennedy L.A.; 1983, "A Model for the Turbulent Energy Spectrum", Phys. Fluids, **26**, p 1228.
36. Wei T. and Willmarth W.W.; 1989, "Reynolds Number Effects on the Structure of a Turbulent Channel Flow", J. Fluid Mech., **204**, p 57.
37. Nikitin N.V., Nicoud F., Wasistho B., Squires K. D. and Spalart P.R.; 2000, "An Approach to Wall Modelling in Large Eddy Simulations", Phys. Fluids, **12**, p 1629.
38. Fureby C., Alin N., Wikström N., Menon S., Persson L., and Svanstedt N.; 2003, "On Large Eddy Simulations of High Re-number Wall Bounded Flows", AIAA Paper 2003-0066, To appear in the AIAA J., Feb issue.
39. Grinstein F.F. Oran E.S. and Boris J.P.; 1991, "Pressure Field, Feedback and Global Instabilities of Subsonic Spatially Developing Mixing Layers," Physics of Fluids, **3**, 10, pp. 2401–2409.
40. Grinstein F.F. and DeVore C.R.; 2002, "On Global Instabilities in Countercurrent Jets," Physics of Fluids, **14**, no. 3, pp. 1095–1100.
41. Grinstein F.F.; 2001, "Vortex Dynamics and Entrainment in Regular Free Jets," J. Fluid Mechanics, **437**, pp. 69–101.
42. Hussain F. and Husain H.S.; 1989, "Elliptic Jets. Part I. Characteristics of Unexcited and Excited Jets," J. Fluid Mechanics, **208**, pp. 257–320.
43. Wikström N. et al.; 2003, "Large Eddy Simulation of the Flow Past an Inclined Prolate Spheroid," Proc. 3rd Int'l Symp.Turbulence and Shear Flow Phenomena, Sendai, Japan.
44. Wetzel T.G., Simpson R.L. and Chesnakas C.J.; 1998, "Measurement of Three-Dimensional Crossflow Separation," J. Am. Inst. Aeronautics and Astronautics, **36**, pp. 557–564.
45. Alin N., Svennberg U. and Fureby C., 2003, "Large Eddy Simulation of Flows Past Simplified Submarine Hulls," Proc. 8th Int'l Conf. Numerical Ship Hydrodynamics, Busan, Korea, pp. 208-222.
46. Groves N.C., Huang T.T., and Chang M.S.; 1989, "Geometric Characteristics of DARPA SUBOFF Models", Report DTRC/SHD-1298-01, David Taylor Research Ctr., 1989.
47. Huang T.T., et al.; 1992, "Measurements of Flows over an Axisymmetric Body with Various Appendages (DARPA SUBOFF Experiments)," Proc. 19th Symp. Naval Hydrodynamics, Seoul, Korea.

Large Scale Urban Simulations with FCT

Gopal Patnaik[1], Jay P. Boris[1], Fernando F. Grinstein[1] and John P. Iselin[2]

[1] LCP&FD, Naval Research Laboratory
Washington, DC 20375-5344, USA
patnaik@lcp.nrl.navy.mil
[2] Bucknell University, Lewisburg
PA 17837, USA

Summary. Airborne contaminant transport in cities presents challenging new requirements for CFD. The unsteady flow physics is complicated by very complex geometry, multi-phase particle and droplet effects, radiation, latent, and sensible heating effects, and buoyancy effects. Turbulence is one of the most important of these phenomena and yet the overall problem is sufficiently difficult that the turbulence must be included efficiently with an absolute minimum of extra memory and computing time. This paper describes the Monotone Integrated Large Eddy Simulation (MILES) methodology used in NRL's FAST3D-CT simulation model for urban contaminant transport (CT) (see [1] and references therein). We also describe important extensions of the underlying Flux-Corrected Transport (FCT) convection algorithms to further reduce numerical dissipation in narrow channels (streets).

1 Background

Urban airflow accompanied by contaminant transport presents new, extremely challenging modeling requirements. Configurations with very complex geometries and unsteady buoyant flow physics are involved. The widely varying temporal and spatial scales exhaust current modeling capacities. Simulations of dispersion of airborne pollutants in urban scale scenarios must predict both the detailed airflow conditions as well as the associated behavior of the gaseous and multiphase pollutants. Reducing health risks from the accidental or deliberate release of Chemical, Biological, or Radioactive (CBR) agents and pollutants from industrial leaks, spills, and fires motivates this work. Crucial technical issues include transport model specifics, boundary condition modeling, and post-processing of the simulation results for practical use by responders to actual real-time emergencies.

Relevant physical processes to be modeled include resolving complex building vortex shedding and recirculation zones and representing the associated subgrid scale (SGS) turbulent stochastic backscatter. The model must also

incorporate a consistent stratified urban boundary layer with realistic wind fluctuations, solar heating including shadows from buildings and trees, aerodynamic drag and heat losses due to the presence of trees, surface heat sorption variations and turbulent heat transport. Because of the short time spans and large air volumes involved, modeling a pollutant as well mixed globally is typically not appropriate. It is important to capture the effects of unsteady, non-isothermal, buoyant flow conditions on the evolving pollutant concentration distributions. In fairly typical urban scenarios, both particulate and gaseous contaminants behave similarly insofar as transport and dispersion are concerned, so that the contaminant spread can be simulated effectively based on appropriate pollutant tracers with suitable sources and sinks. In other cases the full details of multigroup particle distributions are required.

1.1 Established Approach: Gaussian Plume Models

Contaminant plume prediction technology currently in use throughout the nation is based on Gaussian similarity solutions ("puffs"). This is a class of extended Lagrangian approximations that only really apply for large scales and flat terrain where separated-flow vortex shedding from buildings, cliffs, or mountains is absent. Diffusion is used in plume/puff models to mimic the effects of turbulent dispersion caused by the complex building geometry and wind gusts of comparable and larger size (e.g.,[2-5]). These current aerosol hazard prediction tools for CBR scenarios are relatively fast running models using limited topography, weather and wind data. They give only approximate solutions that ignore the effects of flow encountering 3D structures. The air flowing over and around buildings in urban settings is fully separated. It is characterized by vortex shedding and turbulent fluctuations throughout the fluid volume. In this regime, the usual timesaving approximations such as steady-state flow, potential flow, similarity solutions, and diffusive turbulence models are largely inapplicable. Therefore a clear need exists for high-resolution numerical models that can compute accurately the flow of contaminant gases and the deposition of contaminant droplets and particles within and around real buildings under a variety of dynamic wind and weather conditions.

1.2 Computational Fluid Dynamics Approach

Since fluid dynamic convection is the most important physical process involved in CBR transport and dispersion, the greatest care and effort should be invested in its modeling. The advantages of the Computational Fluid Dynamics (CFD) approach and representation include the ability to quantify complex geometry effects, to predict dynamic nonlinear processes faithfully, and to handle problems reliably in regimes where experiments, and therefore model validations, are impossible or impractical.

1.2.1 Standard CFD Simulations

Some "time-accurate" flow simulations that attempt to capture the urban geometry and fluid dynamic details are a direct application of standard (aerodynamic) CFD methodology to the urban-scale problem. An example is the work at Clark Atlanta University where researchers conduct finite element CFD simulations of the dispersion of a contaminant in the Atlanta, Georgia metropolitan area. The finite element model includes topology and terrain data and a typical mesh contains approximately 200 million nodes and 55 million tetrahedral elements [6]. These are grand-challenge size calculations and are run on 1024 processors of a CRAY T3E. Similar approaches are being used by other research groups (e.g., [7,8]). The chief difficulty with this approach for large regions is that they are quite computer intensive and involve severe overhead associated with mesh generation.

1.2.2 The Large-Eddy Simulation Approach

Capturing the dynamics of all relevant scales of motion, based on the numerical solution of the Navier-Stokes Equations (NSE), constitutes Direct Numerical Simulation (DNS), which is prohibitively expensive for most practical flows at moderate-to-high Reynolds Number (Re). On the other end of the CFD spectrum are the industrial standard methods such as the Reynolds-Averaged Navier-Stokes (RANS) approach, e.g., involving $k - \varepsilon$ models, and other first- and second-order closure methods, which simulate the mean flow and model the effects of all turbulent scales. These are generally unacceptable for urban CT modeling because they are unable to capture unsteady plume dynamics. Large Eddy Simulation (LES) constitutes an effective intermediate approach between DNS and the RANS methods. LES is capable of simulating flow features that cannot be handled with RANS such as significant flow unsteadiness and strong vortex-acoustic couplings, and provides higher accuracy than the industrial methods at reasonable cost.

The main assumptions of Large Eddy Simulation are:

(i) that transport is largely governed by large-scale unsteady convective features that can be resolved,
(ii) that the less-demanding accounting of the small-scale flow features can be undertaken by using suitable subgrid scale (SGS) models.

Because the larger scale unsteady features of the flow are expected to govern the unsteady plume dynamics in urban geometries, the LES approximation has the potential to capture many key features which the RANS methods and the various Gaussian plume methodologies cannot.

2 Monotonically Integrated LES

Traditional LES approaches seek sufficiently high-order discretization and grid resolution to ensure that effects due to numerics are sufficiently small, so that crucial LES turbulence ingredients (filtering and SGS modeling) can be resolved. In the absence of an accepted universal theory of turbulence, the development and improvement of SGS models are unavoidably pragmatic and based on the rational use of empirical information. Classical approaches have included many proposals ranging from, inherently-limited eddy-viscosity formulations, to more sophisticated mixed models combining dissipative eddy-viscosity models with the more accurate but less stable Scale-Similarity Model (SSM), see [9] for a recent survey. The main drawback of mixed models relates to their computational complexity and cost for the practical flows of interest at moderate-to-high Re. The shortcomings of LES methods have led many researchers to abandon the classical LES formulations and shift focus directly to the SGS modeling implicitly provided by nonlinear (monotone) convection algorithms (see, e.g., [10], for a recent survey). The idea that a suitable SGS reconstruction might be implicitly provided by discretization in a particular class of numerical schemes [11] lead to proposing the Monotonically Integrated LES (MILES) approach [12,13]. Later theoretical studies show clearly that certain nonlinear (flux-limiting) algorithms with dissipative leading order terms have appropriate built-in (i.e. "implicit") Sub-Grid Scale (SGS) models [14-16]. Our formal analysis and numerous tests have demonstrated that the MILES implicit tensorial SGS model is appropriate for both free shear flows and wall bounded flows. These are the conditions of most importance for CBR transport in cities.

As discussed further below, the MILES concept can be effectively used as a solid base for CFD-based contaminant transport simulation in urban-scale scenarios, where conventional LES methods are far too expensive and RANS methods are inadequate.

3 MILES for Urban Scale Simulations

The FAST3D-CT three-dimensional flow simulation model [1,17,18] is based on the scalable, low dissipation Flux-Corrected Transport (FCT) convection algorithm [19,20]. FCT is a high-order, monotone, positivity-preserving method for solving generalized continuity equations with source terms. The required monotonicity is achieved by introducing a diffusive flux and later correcting the calculated results with an antidiffusive flux modified by a flux limiter. The specific version of the convection algorithm implemented in FAST3D-CT is documented in [21].

Additional physical processes to be modeled include providing a consistent stratified urban boundary layer and realistic wind fluctuations, solar

heating including shadows from buildings and trees. We must also model aerodynamic drag and heat losses due to the presence of trees, surface absorption variations and turbulent heat transport. Additional features include multi-group droplet and particle distributions with turbulent transport to surfaces as well as gravitational settling, solar chemical degradation, evaporation of airborne droplets, relofting of particles on the ground and ground evaporation of liquids. Incorporating specific models for these processes in the simulation codes is a challenge but can be accomplished with reasonable sophistication. The primary difficulty is the effective calibration and validation of all these physical models since much of the input needed from field measurements of these processes is typically insufficient or even nonexistent. Furthermore, even though the individual models can all be validated to some extent, the larger problem of validating the overall code has to be tackled as well. Some of principally fluid dynamics related issues are elaborated further below.

3.1 Urban Flow Modeling Issues

3.1.1 Atmospheric Boundary Layer Specification

We have to deal with a finite domain and the precise planetary boundary layer characterization upstream of this domain greatly affects the boundary-condition prescription required in the simulations. The weather, time-of-day, cloud cover and humidity all determine if the boundary layer is thermally stable or unstable and thus determine the level and structure of velocity fluctuation. Moreover, the fluctuating winds, present in the real world but usually not known quantitatively except perhaps as a global variance, are known to be important because of sensitivity studies.

In FAST3D-CT the time average of the urban boundary layer is specified analytically with parameters chosen to represent the overall thickness and inflection points characteristic of the topography and buildings upstream of the computational domain. These parameters can be determined self-consistently by computations over a wider domain, since the gross features of the urban boundary layer seem to establish themselves in a kilometer or so, but this increases the cost of simulations considerably.

A deterministic realization of the wind fluctuations is currently being superimposed on the average velocity profiles. This realization is specified as a suitable nonlinear superposition of modes with several wavelengths and amplitudes. Significant research issues remain unresolved in this area, both observationally and computationally. Deterministic [22] and other [23] approaches to formulating turbulent inflow boundary conditions are currently being investigated in this context. The strength of the wind fluctuations, along with solar heating as described just below, are shown to be major determinants of how quickly the contaminant density decreases in time and this in turn is extremely important in emergency applications as it determines overall dosage.

3.1.2 Solar Heating Effects

An accurate ray-tracing algorithm that properly respects the building and tree geometry computes solar heating in FAST3D-CT. The trees and buildings cast shadows depending on the instantaneous angle to the sun. Reducing the solar constant slightly can represent atmospheric absorption above the domain of the simulation and the model will even permit emulating a time-varying cloud cover. The geometry database has a land-use variable defining the ground composition. Our simulations to date identify only two conditions, ground and water, though the model can deal with the differences between grass, dirt, concrete and blacktop given detailed enough land-use data. The simulated interaction of these various effects in actual urban scenarios has been extensively illustrated in [1].

Figure 1 shows that the rate that a contaminant is swept out of a city by the winds can vary by a factor of four or more due to solar heating variations from day to night and due to variations in the relative strength of the wind gusts. The horizontal axis of the figure indicates the relative strength of the gusting fluctuations at the boundaries, from about 20% on the left to about 100% on the right. For each of six different "environmental" conditions, twelve ground-level sources were released, four independent realizations at each of three source locations around the urban geometry. These source locations and the scale lengths of all the wind fluctuations were held fixed

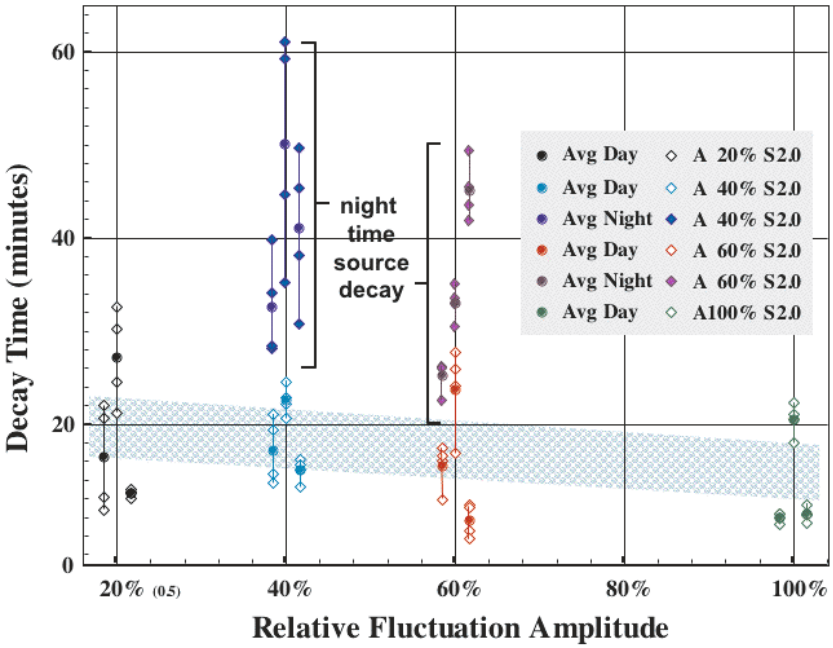


Fig. 1. Effect of fluctuations on trapping of contaminants

for the six different runs. The value of the exponential decay time in minutes is plotted for each source and realization as a diamond-shaped symbol. The figure shows that the decay time is two or three times longer for release at night compared to the day for otherwise identical conditions. The dark blue diamonds (decay times) should be compared with the light blue and the purple diamonds compared with the red. One can also see that the decay times get systematically shorter as the wind fluctuation amplitude is increased from left to right. This is emphasized by the light blue shaded bar through the center of the four daytime data sets.

3.1.3 Tree Effects

Although we can resolve individual trees if they are large enough, their effects (i.e., aerodynamic drag, introduction of velocity fluctuations, and heat losses) are represented through modified forest canopy models [24] including effects due to the presence of foliage. For example, an effective drag-force source term for the momentum equations can be written as $F = -C_d\, a(z)|\mathbf{v}|\mathbf{v}$, where $C_d = 0.15$ is an isotropic drag coefficient, $a(z)$ is a seasonally-adjusted leaf area density, z is the vertical coordinate, and $\mathbf{v}$ is the local velocity. The foliage density is represented in a fractal-like way so that fluctuations will appear even in initially laminar flows through geometrically regular stands of trees.

3.1.4 Turbulent Backscatter

The distribution of subgrid-scale (SGS) viscosity is a distinct feature characterizing the ability of different LES models to capture the underlying unresolved physics, ranging from purely dissipative scalar to tensorial scale-similarity models. An overall positive SGS viscosity implies that energy is transferred from resolvable flow structures towards small, unresolved scales via a cascade process (outscatter). Conversely a negative SGS viscosity implies that energy is overall transferred in the opposite direction by a reverse cascade process, i.e. backscatter – which can become important in complex geometries such as involved in CT when a large amount of turbulent kinetic energy is present in the *unresolved* scales. Even when the systematic, overall cascade corresponds to outscatter, backscatter, both systematic and stochastic, can occur at select wavelengths and for certain nonlinear triads of modes. Modeling how the unresolved features of the flow contribute to the large scales through this (stochastic) backscatter process presents a difficult challenge: how are these effects to be predicted based on the resolvable scale information?

Because of the anisotropic features of the implicit SGS modeling incorporated [15], MILES offers an effective approach for the simulation of inherently inhomogeneous turbulent flows in complex geometries such as involved in CT. It has been demonstrated that this SGS modeling is not purely dissipative, and that some degree of desirable systematic backscatter is actually incorporated implicitly in MILES [15]. Additional (explicit) backscatter effects can

be modeled by taking advantage of the flux-limiter information computed by the FCT convection algorithms.

For each component of the fluid momentum, the unused high order flux is accumulated at each grid point during the direction-split convection stages of the integration, and is measured in terms of its absolute value summed over all three directions for each timestep and suitably normalized by the density, i.e.,

$$\eta = \frac{1}{\langle\rho\rangle} \sum (1 - \Gamma) \ \left|\rho \mathbf{v}_f^H - \rho \mathbf{v}_f^L\right|.$$

The flux limiter Γ is described more fully in section 4 below. This quantity is plotted in Figs. 2 and 3 for several cross-sections from a FAST3D-CT simulation of the airflow over Washington DC.

Figure 2 shows three horizontal planes sliced through the FAST3D-CT grid at constant k indices. These planes are spaced 3 cells apart and indicate the magnitude of the stochastic backscatter velocities at 3 meters (upper right), 21 meters (center) and 39 meters (lower left) off the ground. Red colors indicate an average 10 cm/sec effective subgrid flow speed and faster, yellow indicates about 3 cm/sec and black indicates about 1 cm/sec.

When the FCT algorithm detects structure in the flow that it "knows" cannot be resolved on the grid, only a fraction of the anti-diffusion flux can be applied. The fraction that cannot be used is an implicit estimate of this

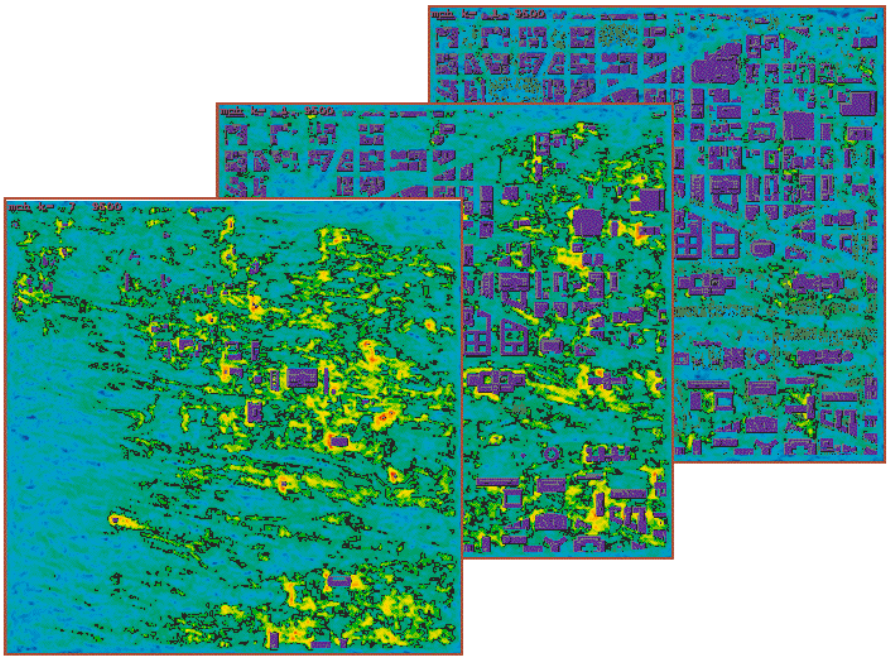


Fig. 2. Magnitude of stochastic backscatter velocities along horizontal planes

unresolved flow and it is coupled on the grid scale to the specifics of the fluid dynamic convection algorithm. This estimate is an inexpensive replacement for the multi-scale filtering schemes used in many of the current LES approaches. Furthermore, since the eddy dissipation has already been included in the (anisotropic) MILES representation with a suitable tensorial form, these unused fluxes are available for other, higher order corrections.

Figure 3 shows eight vertical planes sliced through the FAST3D-CT grid, four north-south planes at constant i indices and four east-west planes at constant j indices. These planes are spaced 25 cells (150 m) apart. The quantity plotted is again the magnitude of the SGS flux η defined above. The blue band adjacent to the upper boundary of the grid is a region where the imposed boundary flow is sufficiently laminar that the stochastic flux is quite small.

Note the effect of building wakes in generating the unused fluxes and also note that the vortex shedding from buildings tends to carry this turbulence about two building heights into the air. This is a nighttime computation so the atmosphere is stabilized due to thermal stratification. Note also that some subgrid activity can be noted away from the buildings due undoubtedly to nonlinear cascade in the atmospheric turbulence being convected with the flow.

FAST3D-CT uses these "stochastic back scatter fluxes" by pseudo-randomly perturbing the resolved flow velocity in each cell by an amount proportional (with a factor of about 0.3) to the unused flux velocities depicted in the Figs. 2 and 3. In some geometries, these additional grid-scale fluctuations break sym-

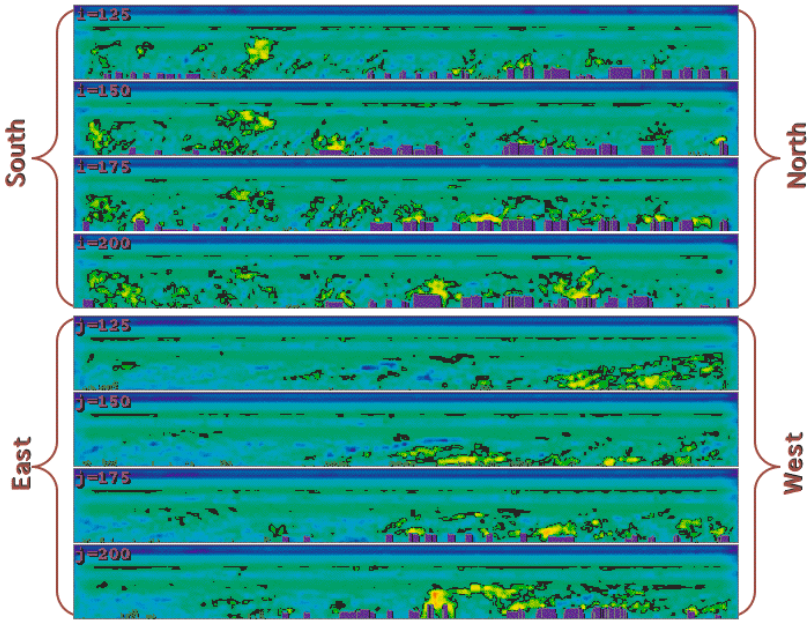


Fig. 3. Magnitude of stochastic backscatter velocities along vertical planes

metries and initiate three-dimensional instabilities via stochastic backscatter that otherwise would have to grow up from computer round-off. They also transport small particles and droplets to material surfaces as a result of unresolved turbulence even though the resolved flow field has a zero velocity normal to the walls. This means that particles and droplets can deposit on a ceiling as well as a floor. Finally, the numerical limiting of the imposed stochastic fluctuations, caused by the nonlinear flux limiter, provides a small additional macroscopic (resolved scale) transport right where the FCT algorithm has detected subgrid structure. Each of these potentially realistic effects requires further theoretical analysis and/or a phenomenological model carefully calibrated by experiment. This work remains to be done.

All MILES, and more generally ILES, methods are quite capable of capturing (at least semi-quantitatively) how much unresolved, small-scale structure is actually present from looking at the evolving resolved-scale solutions. Further, systematic diffusion of the eddy transport type can be automatically left in the flow, but the fluctuating, driving effects of random-phase, unresolved small-scale motions scattering back onto the large scales are missing unless specifically included as a subgrid phenomenology. Therefore it is our continuing interest to employ the unused fluxes with random, or stochastic, multipliers. However, a factor of two increase in the spatial resolution of LES and MILES models will most likely bring more improvement in the accuracy of the well resolved scales than subgrid models will ever provide regardless of the type of LES employed. Work is ongoing to provide satisfying proofs of these statements.

3.1.5 Geometry Specification

An efficient and readily accessible data stream is available to specify the building geometry data to FAST3D-CT. High-resolution (1 m or smaller) vector geometry data in the ESRI ARCVIEW data format is commercially available for most major cities. From these data, building heights are determined on a regular mesh of horizontal locations with relatively high resolution (e.g., 1 m). Similar tables for terrain, vegetation, and other land use variables can be extracted. These tables are interrogated during the mesh generation to determine which cells in the computational domain are filled with building, vegetation, or terrain. This masking process is a very efficient way to convert a simple geometric representation of an urban area to a computational grid.

This grid masking approach is used to indicate which computational cells are excluded from the calculation as well as to determine where suitable wall boundary conditions are to be applied. However, the grid masking approach is too coarse to represent rolling terrain, for which a shaved cell approach is used instead. The terrain surface is represented by varying the location of the lower interface of the bottom cell. Even though this results in a terrain surface that is discontinuous, the jump between adjacent cells is small, and

operational results show that this approach works reasonably well and is far better than the grid masking approach for representing terrain.

A more accurate representation of the geometry is possible with the VCE approach [25] in which the cell volume and interface areas are allowed to vary. This level of detail now begins to approach that of conventional aerodynamics CFD and it remains to be seen if this is necessary.

3.1.6 Wall Boundary Conditions

Appropriate wall boundary conditions must be provided so that the airflow goes around the buildings. It is not possible with the available resolution to correctly model the boundary layer on the surface. Therefore, rough-wall boundary layer models [26] are used for the surface stress, i.e., $\tau = \rho C_D (U_{//})^2$, and for the heat transfer from the wall, $H_o = \rho C_p C_H U_{//}(\Theta - \Theta_o)$, where ρ is the mass density, C_D and C_H are coefficients characterizing the roughness and thermal properties of the walls or ground surface, $U_{//}$ is the tangential velocity at the near-wall (first grid point adjacent to the wall), C_p is the specific heat at constant pressure, and Θ and Θ_o are the potential temperature at the wall, and near-wall, respectively.

4 MILES Implicit SGS model

Historically, flux-limiting (flux-correcting) methods have been of particular interest in the MILES context. A flux-limiter $0 \leq \Gamma \leq 1$ combines a high-order convective flux-function $\mathbf{v}_f^H$ that is well behaved in smooth flow regions, with a low-order dispersion-free flux-function $\mathbf{v}_f^L$ that is well behaved near sharp gradients. Thus the total flux-function with the limiter Γ becomes $\mathbf{v}_f = \mathbf{v}_f^H - (1-\Gamma)[\mathbf{v}_f^H - \mathbf{v}_f^L]$. Properties of the implicit SGS model in MILES are related to the choice of Γ, $\mathbf{v}_f^L$, and $\mathbf{v}_f^H$, as well as to other specific features of the algorithm [15,16]. This is quite similar to choosing/adjusting an (explicit) SGS model in the context of conventional LES.

Because of its inherently less-diffusive nature, prescribing Γ based on local monotonicity constraints is a more attractive choice in developing MILES [14-16]. This is supported by our comparative channel flow studies [16] of the global performance of MILES as a function of flux limiter. For example, the van-Leer TVD limiter (e.g., [27]) was found to be too diffusive as compared to FCT [19] and GAMMA [28] limiters which produce velocity profiles that agree well with the reference DNS data.

4.1 Street Crossings

Another approach to controlling unwanted numerical diffusion is through the appropriate choice of low and high order transport algorithms. In our simulations of urban areas, the typical grid resolution is of the order of 5 to 10

meters. While this resolution is adequate to represent the larger features of the city, many of the smaller features are resolved with only one to two cells. This is true of smaller streets found in cities, which are about 20 m wide. Alleyways are even smaller. These smaller streets may be represented by only one or two cells in our computation, putting a tremendous demand on the numerical convection not to diffuse and retard the flow down these narrow streets.

By using the rough-wall boundary conditions discussed above instead of no-slip boundary conditions, the flow can proceed unhampered down a single street even for streets that are only one cell wide. However, if there is another street intersecting the first, it was found that the flow essentially stagnates at this intersection. The problem only occurs when dealing with streets which are 1–2 cells wide and not with wider streets. After careful inspection, it was determined that the problem arose due to the form of the diffusion term in the low-order solution in the standard FCT algorithm, LCPFCT [21], used in the FAST3D-CT code.

The traditional low-order component of FCT introduces numerical diffusion even when the velocity goes to zero (as in the cross street) [21]. In normal situations, the flux limiter is able to locate an adjacent cell that has not been disturbed by the diffusion in the low-order method and is able to restore the solution to its original undiffused value. However, when the streets are 1–2 cells wide, the region of high velocity is diffused by low-order transport and there are no cells remaining at the higher velocity (Fig. 4). Thus the flux limiter cannot restore the solution in these cells to the original high value.

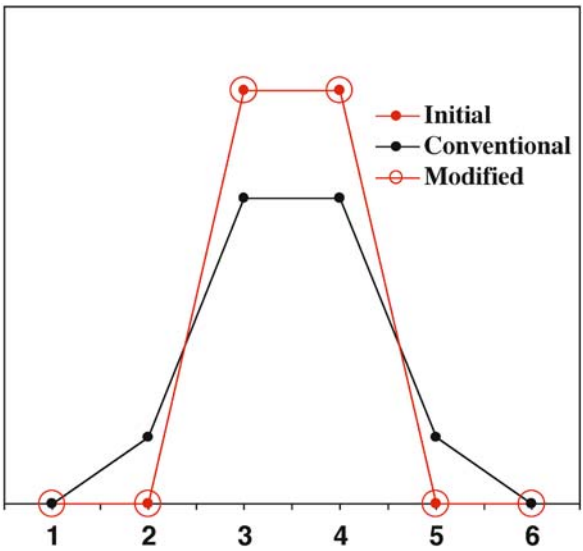


Fig. 4. Advected quantity as function of grid index for both the conventional and modified low-order schemes

A solution to this problem lay in changing the form of the diffusion in the low-order method. In LCPFCT, the algorithmic diffusion coefficient for the low-order scheme is given by $\nu = \frac{1}{6} + \frac{1}{3}\varepsilon^2$, where the Courant number $\varepsilon = |U|\,\Delta t/\Delta x$. Note that ν does not go to zero even when U goes to zero (as in the cross street). The simplest less-diffusive low-order algorithm which ensures monotonicity is the upwind method previously used in the formal MILES analysis (e.g., [16]) for which the diffusion coefficient is given by $\nu_{upwind} = \frac{1}{2}\,|\varepsilon|$, which has the desired form for ν. When the diffusion coefficient in the low-order component of FCT is replaced by ν_{upwind}, the flow no longer stagnates at the intersection of streets (Fig. 4). This variation of the low-order method is only used for the momentum equations. It is not required for the density equation, since the density is almost constant everywhere. With this modification of the low-order method, the global properties of the transport algorithm were altered sufficiently to address this problem peculiar of under-resolved flows in urban areas.

5 Practical Examples

5.1 Gaussian Lagrangian vs. Unsteady 3D Solutions

Gaussian atmospheric transport and dispersion schemes are characterized by some initial direct spreading of the contaminant upwind by the diffusion, regardless of wind speed. The characteristic differences between the three Gaussian similarity solutions in Fig. 5 are similar to the differences between different Gaussian plume/puff models. None of these approximate, idealized solutions has the correct shape, trapping behavior, or plume width when compared to the FAST3D-CT simulation shown in the upper-right panel of the Fig. 5. The contaminant gets trapped in the re-circulation zones behind buildings and continues to spread laterally long after simpler models say the cloud has moved on.

More detailed comparisons using actual "common use" puff/plume models (e.g., [29]) show a range of results depending on how much of the 3D urban boundary layer information from the detailed simulation is incorporated in the Gaussian model. Though building-generated aerodynamic asymmetries cannot be replicated, crosswind spreading and downwind drift can be approximately matched given enough free parameters. However, because the detailed simulations show that the plume expands like an angular sector away from the source, Gaussian models show too rapid a lateral spreading in the vicinity of the source to provide a plume that is approximately the correct width downwind.

5.2 Unsteady 3D Solution — Chicago

The city of Chicago is typical of a large, densely populated metropolitan area in the United States. The streets in the downtown area are laid out in a grid–

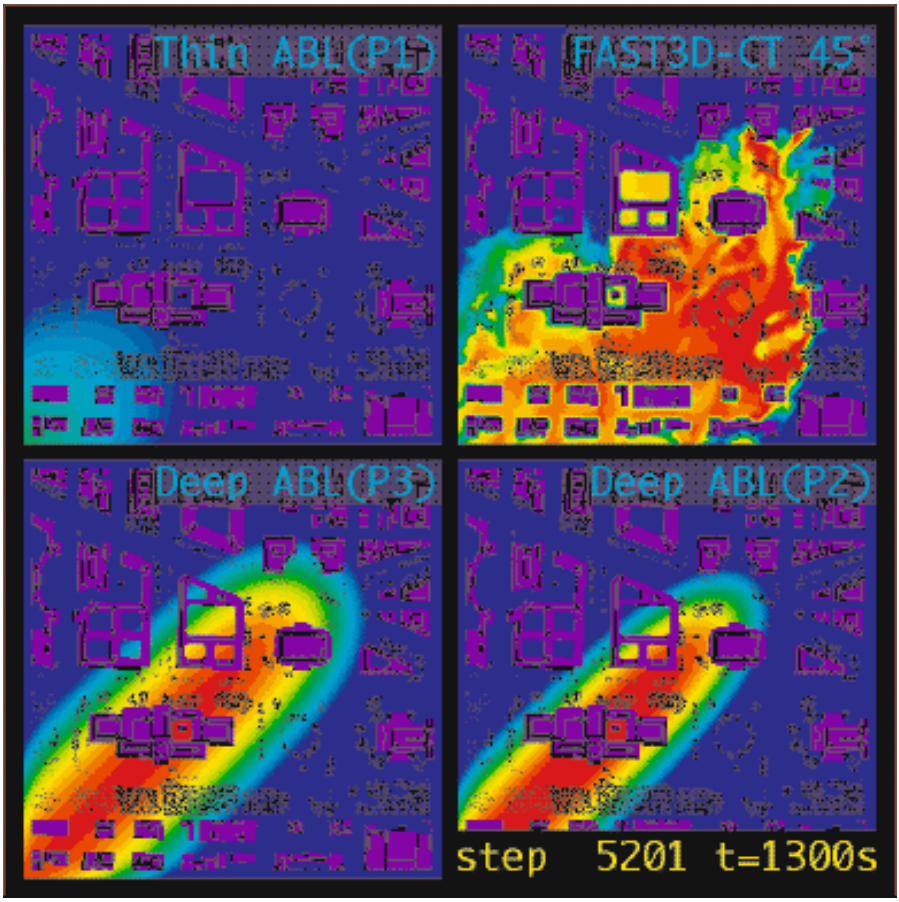


Fig. 5. Comparison of Gaussian Plume and FAST3D-CT simulations

like fashion, and are relatively narrow. The buildings are very tall but with small footprints. For example, the Sears tower is now the tallest building in the U. S.

Figures 6–8 show different views of a contaminant cloud from a FAST3D-CT simulation of downtown Chicago using a $360 \times 360 \times 55$ grid (6 m resolution). A 3 m/s wind off the lake from the east blows contaminant across a portion of the detailed urban geometry data set required for accurate flow simulations. One feature that is very apparent from these figures is that the contaminant is lofted rapidly above the tops of the majority of the buildings. This vertical spreading of the contaminant is solely due to the geometrical effect of the buildings. This behavior has also been observed in other simulations in which the buildings are not as tall.

Placement of the contaminant source can have a very nonlinear effect on the dispersion characteristics. Figures 9 and 10 show results of identical

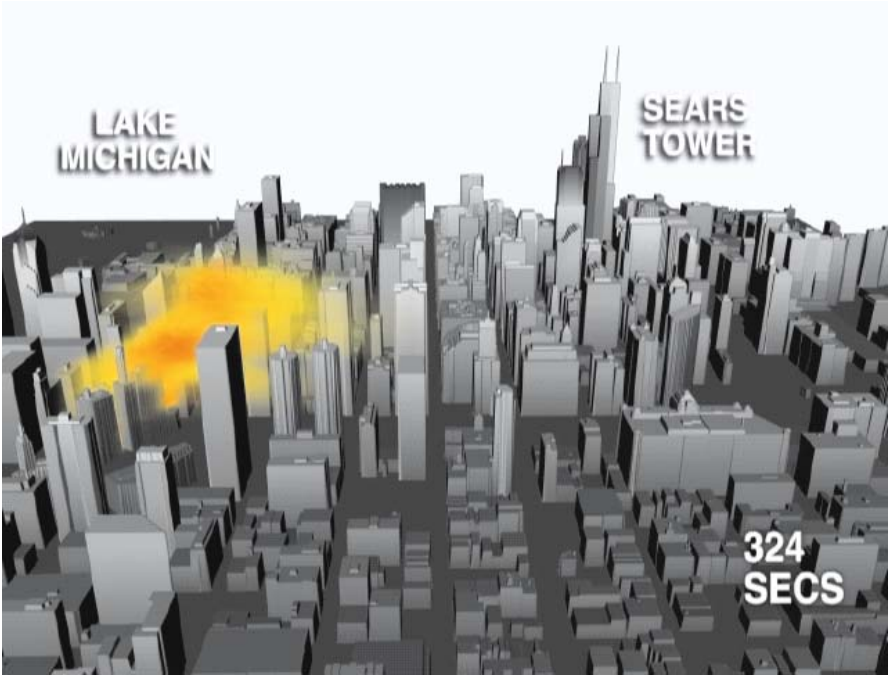


Fig. 6. View of contaminant release looking East toward downtown Chicago

simulations with the exception of the contaminant release locations, which are shown by the red markers. The blue markers show the release location in the other simulation. Although the release locations differed by less than 0.5 km the dispersion characteristics are markedly different. The narrower dispersion pattern in Fig. 9 is likely caused by a channeling effect of the Chicago River where velocities are higher. The wider dispersion pattern in Fig. 10 is likely due to a combination of flow deflection and recirculation of the flow from the building geometry. This behavior may also be dependent on release time. Work is continuing to determine the function dependence on location and release time. However, it is clear that bulk parameterizations of urban surface characteristics will be unable to account for these nonlinear effects.

Additional simulations for Chicago were used to examine the effect of the modified low-order component of FCT as described in Sect. 4.1. Figure 11 shows the contaminant at ground level 9 minutes after release using the standard FCT algorithm LCPFCT [21]. The channeling effect of the Chicago River is quite dominant, though some lateral spreading occurs as well. The calculations were then repeated with the modified low-order method. These results are shown in Fig. 12. It is immediately apparent that the lateral spreading is much larger in this case and that the cloud also propagates

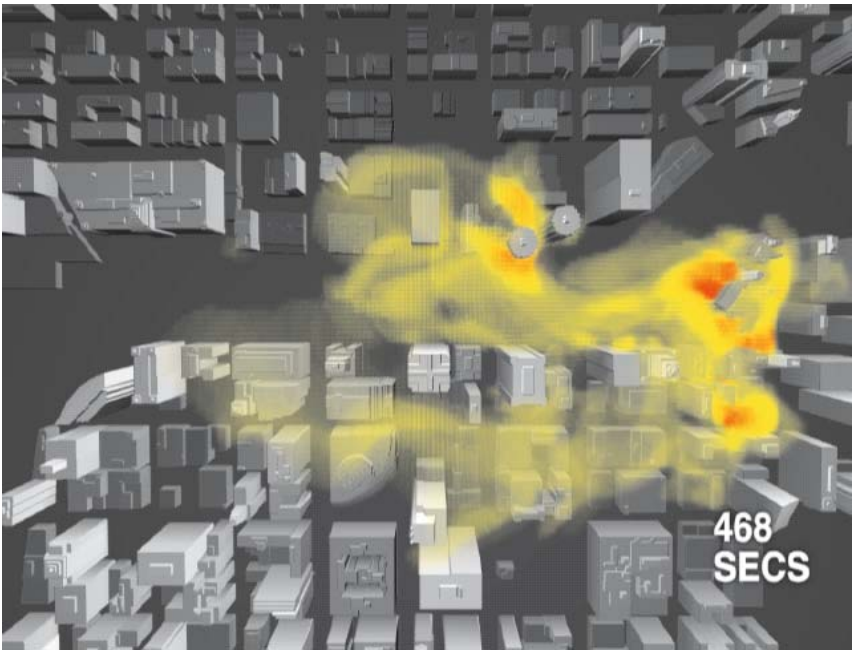


Fig. 7. Overhead view of contaminant concentrations over Chicago River

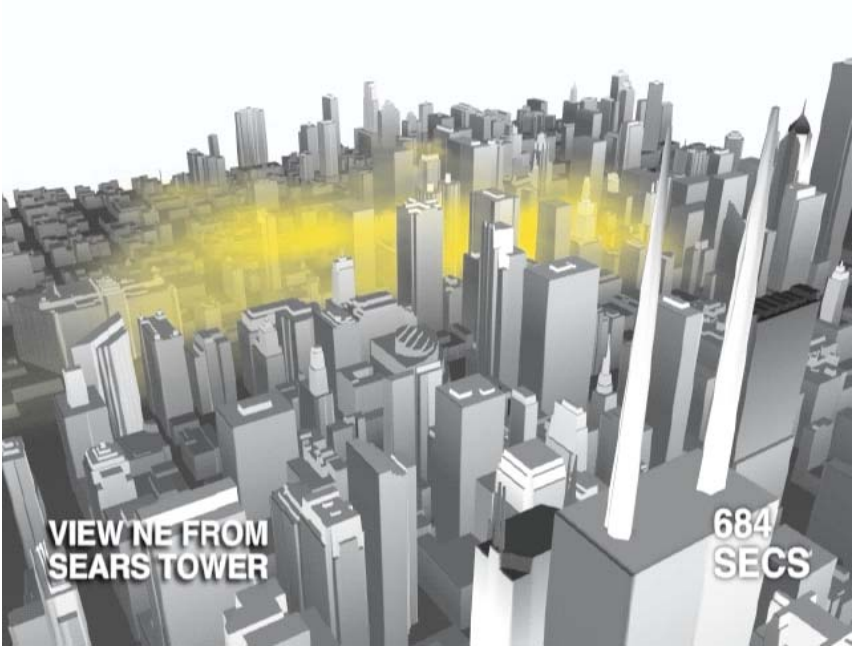


Fig. 8. View of contaminant from the Sears tower

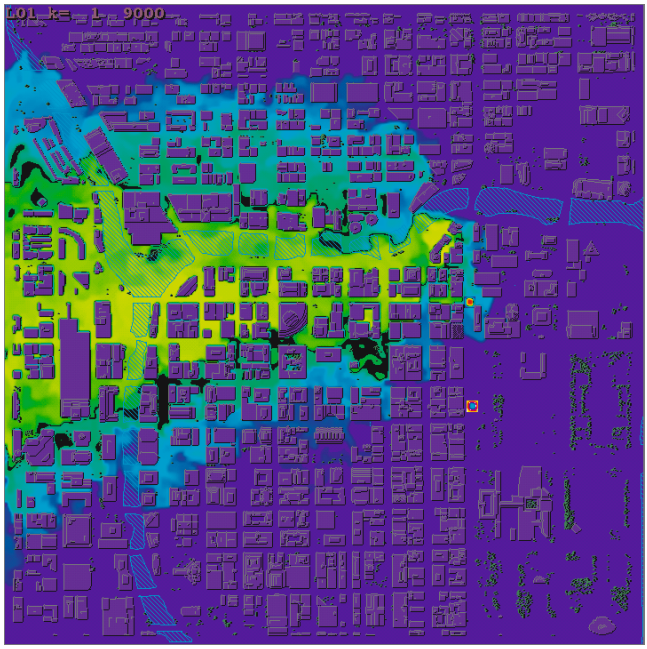


Fig. 9. Contaminant dispersion at ground level for release close to Chicago River

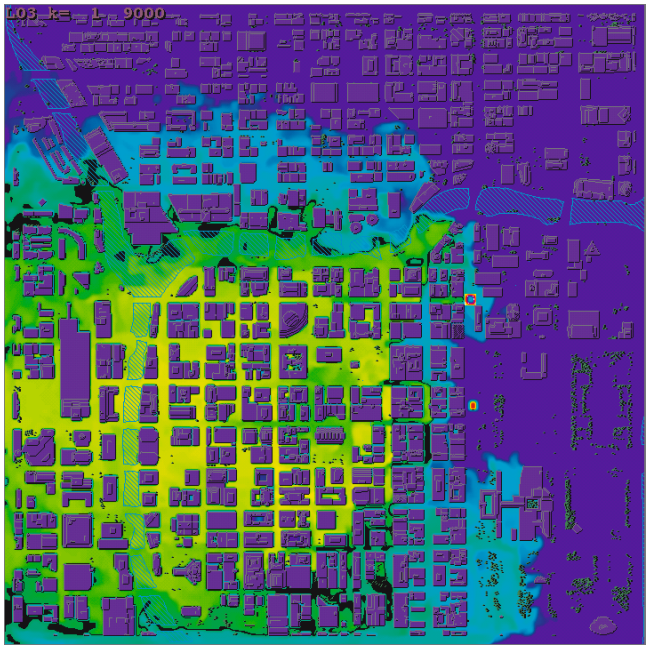


Fig. 10. Contaminant dispersion at ground level for release further from Chicago River

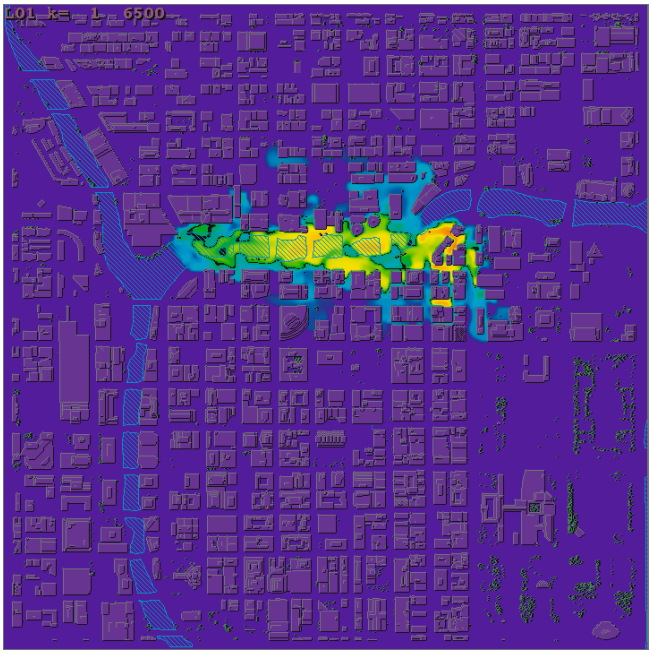


Fig. 11. Contaminant dispersion using standard low-order method

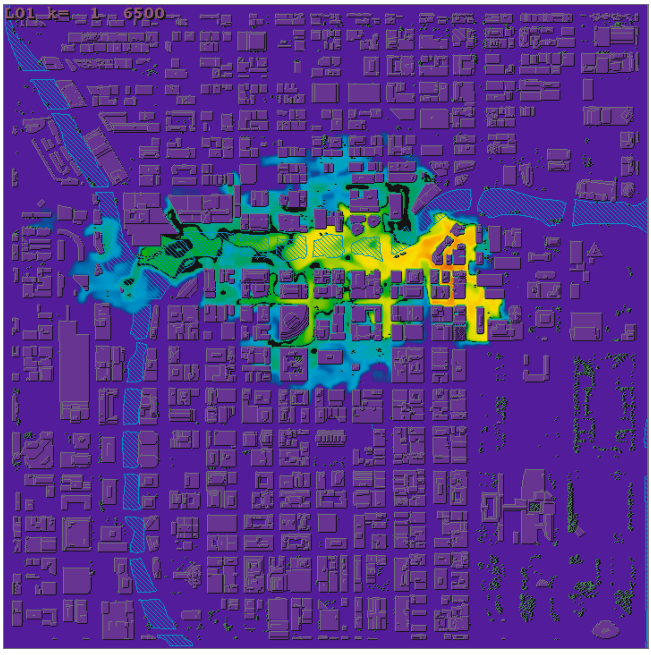


Fig. 12. Contaminant dispersion using modified low-order method

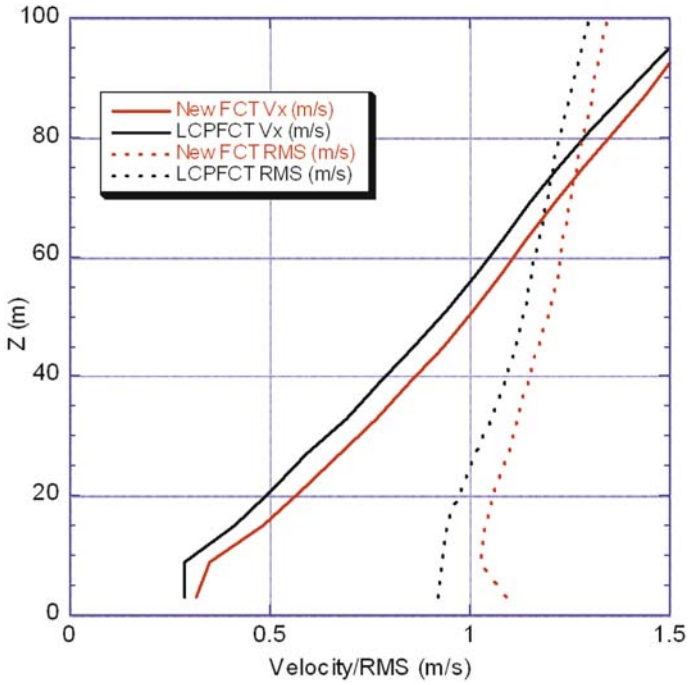


Fig. 13. Comparison of the standard and modified low-order method: velocity and RMS fluctuation profiles

more rapidly downstream. These effects can be attributed to the lowered numerical diffusion in the cross-stream direction and the consequent lowering of numerical diffusion overall. Figure 13 shows the velocity (averaged horizontally over the computational domain) and RMS fluctuation profiles for both LCPFCT and the modified low-order scheme. The modified method has higher values for both velocity and RMS fluctuation. This is consistent with the observation of increased downstream and cross-stream propagation of the contaminant.

While it is expected that the solution with the lower numerical diffusion is preferable, in the absence of experimental measurements, it is not possible to directly assess this improvement.

5.3 Unsteady 3D solution — Baghdad

Baghdad is rather typical of capital cities — it has large, spread-out governmental buildings, parks and monuments. There is no large dense urban core with tall buildings (there are several 20+ story buildings that are spread out). Residential areas are mostly suburban with some high-rise housing. This city structure is quite different from Chicago with its skyscrapers. A limited

amount of high-resolution building data was available from a government-related source; however this data only included the largest buildings and covered a fraction of the area of the city. Large portions, especially residential areas, were not covered. Also, land-use data (trees, water, etc.) were not available in high-resolution form.

The missing data was constructed manually, primarily from commercially available satellite photographs of the city. These photographs had sufficient resolution to discern trees, water, and even types of housing. "Synthetic" buildings were generated to represent areas not covered by the available high-resolution data. Typical building heights and shapes found in suburbs were assigned at random to suburban regions. One of the difficulties not typically found in CFD calculations which proved to be a challenge was to ensure proper geo-referencing of the data, i.e., ensure everything lined up. This is especially severe when working from photographs that do not have a uniform resolution, and may sometimes not have the proper orientation.

5.3.1 In-situ Validation

One of the obvious difficulties that arise for simulations of urban areas is that of validation of results. Experimental data is rarely available, and what little that is available is extremely limited in scope and coverage. The type and extent of data that is available restrict the quality of the validation effort. For Baghdad, no specific field measurements are available. However, just prior to the start of the war in Iraq, large trenches filled with oil were set ablaze in hope that the smoke would obscure targets. The smoke from one such fire provided an opportunity to at least visually "validate" our plume calculations. Figure 14 is a satellite photograph (courtesy DigitalGlobe) of the smoke from a trench fire near the monument to the Unknown Soldier. Figure 15 and 16 show the results from our simulations of the event. Color contours of the tracer gas are shown. The weather conditions for that day were given as "light wind from northwest." The simulations were carried with nominal wind speed of 3 m/s at 340°. An important unknown that had to be estimated is the level of fluctuation in the wind. The simulation depicted in Fig. 15 used a low level of fluctuation, which is consistent with the light steady winds typically found in March in the area. In order to investigate the importance of wind fluctuations, a higher level of fluctuations was simulated as shown in Fig. 16, which had fluctuations four times as high in amplitude as the baseline case (Fig. 15). As expected, the plume does spread slightly further. However, for low wind fluctuations, the spreading is largely controlled by the geometry of the city — an effect which becomes more dominant in dense urban areas. These calculations show that while a good knowledge of the weather is required for accurate predictions, in order to predict a worst-case scenario it is possible to select the appropriate parameters without perfect knowledge of all input conditions.

Fig. 14. Smoke plume from oil fire in Baghdad

6 Concluding Remarks

Physically realistic urban simulations are now possible but still require some compromises due to time, computer, and manpower resource limitations. The necessary trade-offs result in sometimes using simpler models, numerical algorithms, and geometry representations than we would wish. We know that the quality of the spatially and time-varying boundary conditions imposed, that is, the fluctuating winds, require improvement. Detailed time-dependent wind field observations at key locations can be processed suitably to provide initial

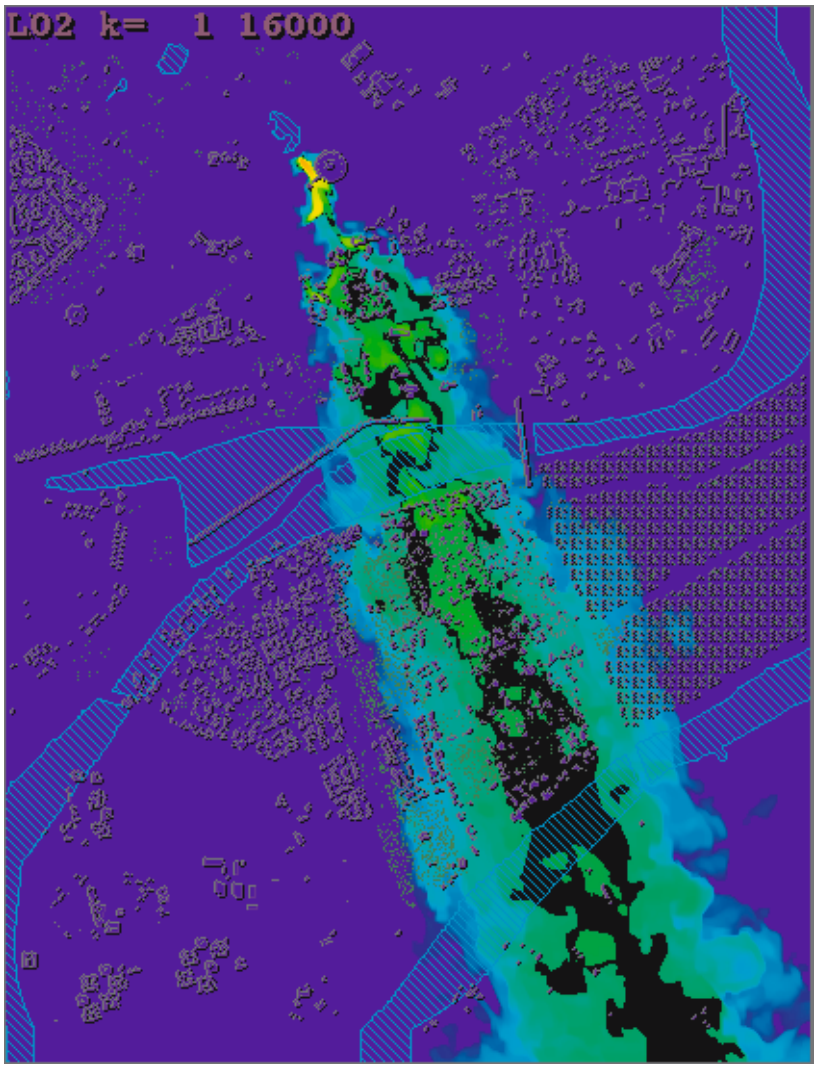


Fig. 15. Simulation of smoke plume. Low wind fluctuations

and boundary conditions and, at the least, can be used for global validation (e.g., [23]).

We believe that the building and large-scale fluid dynamics effects that can be presently captured govern the turbulent dispersion, and expect that the computed predictions will get better in time because the MILES methodology is convergent. However, there is considerable room to improve both the numerical implementation and the understanding of the stochastic backscatter that is included both implicitly and explicitly.

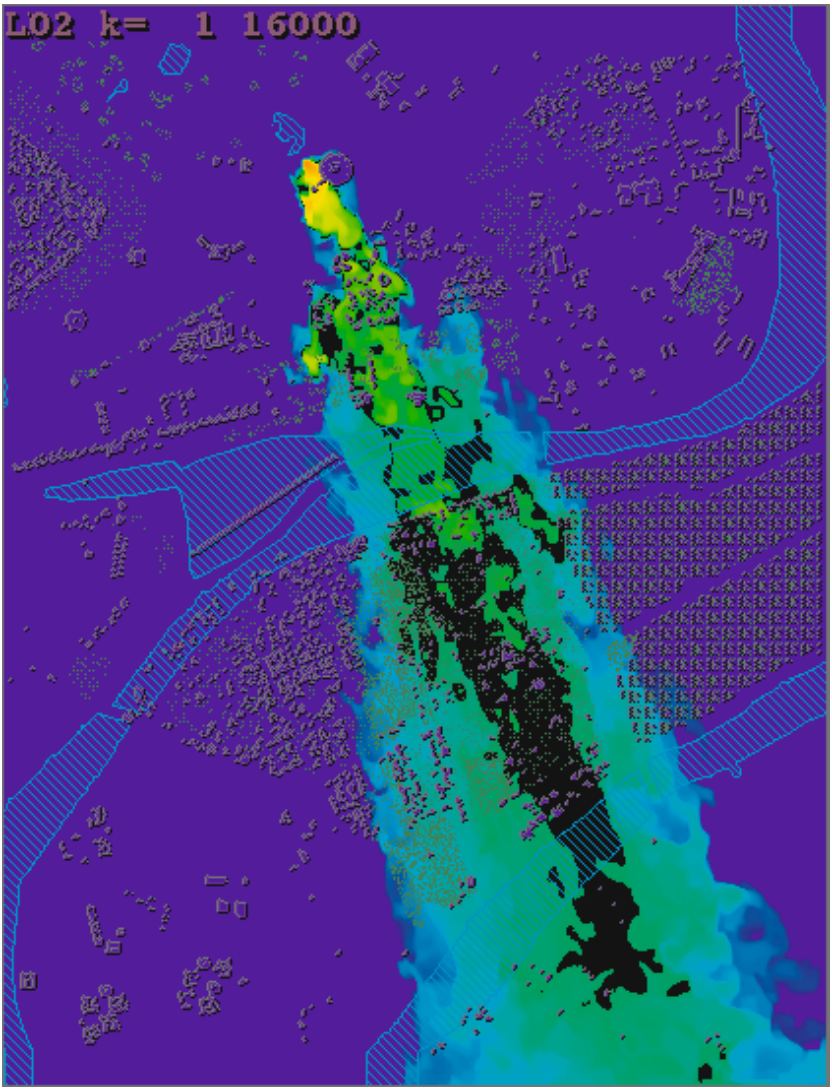


Fig. 16. Simulation of smoke plume. High wind fluctuations

Inherent uncertainties in simulation inputs and model parameters beyond the environmental conditions also lead to errors that need to be further quantified by comparison with high quality reference data. Judicious choice of test problems for calibration of models and numerical algorithms are essential and sensitivity analysis help to determine the most important processes requiring improvement. In spite of inherent uncertainties and model trade-offs it is possible to achieve some degree of predictability.

Testing and calibrating stochastic backscatter algorithms and any theoretical work attending that effort is a difficult task. We would like to suggest a class of test problems for doing this. Figure 7 in Reference [12], shows the convergence of macroscopic entrainment with resolution in a set of MILES simulations performed with FAST3D. This problem was used originally to identify the various numerical effects in MILES and to corroborate the prediction of a minimum in the entrainment at a certain, rather coarse, numerical resolution. The minimum occurs because short wavelengths, that provide some measurable additional entrainment, cannot be resolved near the minimum but the residual numerical diffusion present in high-order FCT (monotone) convection algorithms has become quite small. Therefore increasing resolution actually increases the entrainment. Better LES algorithms, including proposed stochastic backscatter modifications, should reduce the depth of the minimum and push the resolution at the minimum to coarser grids. By revisiting this problem we should be able to extend the quantitative understanding and to calibrate practically the stochastic backscatter model coefficients.

The FAST3D-CT simulation model can be used to simulate sensor and system response to postulated threats, to evaluate and optimize new systems, and conduct sensitivity studies for relevant processes and parameters. Moreover, the simulations can constitute a virtual test range for micro- and nano-scale atmospheric fluid dynamics and aerosol physics, to interpret and support field experiments, and to evaluate, calibrate, and support simpler models.

Figure 7 illustrates the critical dilemma in the CT context: unsteady 3D urban-scenario flow simulations are currently feasible — but they are still expensive and require a degree of expertise to perform. First responders and emergency managers on site for contaminant release threats cannot afford to wait while actual simulations and data post-processing are being carried out. A concept addressing this problem [1,30] carries out 3D unsteady simulations in advance and pre-computes compressed databases for specific urban areas based on suitable (e.g., historical, seasonally adjusted) assumed weather, wind conditions, and distributed test-sources. The relevant information is summarized as *dispersion nomograph*^{TM} data so that it can be readily used through portable devices, in conjunction with urban sensors providing current observational information regarding local contaminant concentrations, wind speed, direction, and relative strength of wind fluctuations.

7 Acknowledgements

The authors wish to thank Bob Doyle and Ted Young for helpful technical discussions and scientific contributions to this effort. Aspects of the work presented here were supported by ONR through NRL, the DoD High Performance Computing Modernization Office, DARPA and MDA.

References

1. J.P. Boris, The Threat of Chemical and Biological Terrorism: Preparing a Response, *Comp. Science and Engineering* **4**, 22–32 (2002)
2. Hazard Prediction and Assessment Capability, available at *http://www.dtra.mil/td/acecenter/td_hpac_fact.html*
3. T. Bauer and M. Wolski, Software User's Manual for the Chemical/Biological Agent Vapor, Liquid, and Solid Tracking (VLSTRACK) Computer Model, Version 3.1, NSWCDD/TR-01/83, April 2001
4. ALOHA Users Manual, 1999. Available for download at: *http://www.epa.gov/ceppo/cameo/pubs/aloha.pdf.* Additional information: *http://response.restoration.noaa.gov/cameo/aloha.html*
5. J.M. Leone, J.S. Nasstrom, D.M. Maddix, D.J. Larsen, and G. Sugiyama, LODI User's Guide, Version 1.0, Lawrence Livermore National Laboratory, Livermore CA (2001)
6. S. Aliabadi and M. Watts, Contaminant Propagation in Battlespace Environments and Urban Areas, AHPCRC Bulletin, Vol 12, No 4 (2002), in *http://www.ahpcrc.org/publications/archives/v12n4/Story3/*
7. S. Chan, FEM3C – An Improved Three-Dimensional Heavy-Gas Dispersion Model: User's Manual, UCRL-MA-116567, Rev. 1, Lawrence Livermore National Laboratory, Livermore CA (1994)
8. F. Camelli and R. Löhner, Assessing Maximum Possible Damage for Release Events, Seventh Annual George Mason University Transport and Dispersion Modeling Workshop, June 2003
9. P. Sagaut, *Large Eddy Simulation for Incompressible Flows*, Springer, New York (2002)
10. F.F. Grinstein & G.E. Karniadakis (Editors), Alternative LES and Hybrid RANS/LES, *J. Fluids Engineering* **124**, 821–942 (2002)
11. J.P. Boris, On Large Eddy Simulation Using Subgrid Turbulence Models, in *Whither Turbulence? Turbulence at the Crossroads*, J.L. Lumley Ed., Springer, New York, p 344 (1989)
12. J.P. Boris, F.F. Grinstein, E.S. Oran, and R.J. Kolbe, New Insights into Large Eddy Simulation, *Fluid Dynamics Research* **10**, 199–228 (1992)
13. E.S. Oran and J.P. Boris, Computing Turbulent Shear Flows — A Convenient Conspiracy, *Computers in Physics* **7**(5), 523–533 (1993)
14. C. Fureby and F.F. Grinstein, Monotonically Integrated Large Eddy Simulation of Free Shear Flows, *AIAA Journal* **37**, 544–556 (1999)
15. C. Fureby and F.F. Grinstein, Large Eddy Simulation of High Reynolds-Number Free & Wall-Bounded Flows, *Journal of Computational Physics* **181**, 68 (2002)
16. F.F. Grinstein and C. Fureby, Recent Progress on MILES for High Reynolds-Number Flows, *Journal of Fluids Engineering* **124**, 848 (2002)
17. B.Z. Cybyk, J.P. Boris, T.R. Young, C.A. Lind, and A.M. Landsberg, A Detailed Contaminant Transport Model for Facility Hazard Assessment in Urban Areas, AIAA Paper 99-3441 (1999)
18. B.Z. Cybyk, J.P. Boris, T.R. Young, M.H. Emery, and S.A. Cheatham, Simulation of Fluid Dynamics Around Complex Urban Geometries, AIAA Paper 2001-0803 (2001)
19. J.P. Boris and D.L. Book, Flux-Corrected Transport I, SHASTA, A Fluid Transport Algorithm that Works, *Journal of Computational Physics* **11**, 8–69 (1973)

20. J.P. Boris, and D.L. Book, Solution of the Continuity Equation by the Method of Flux-Corrected Transport, *Methods in Computational Physics* **16**, 85–129 (1976)
21. J.P. Boris, A.M. Landsberg, E.S. Oran and J.H. Gardner, LCPFCT - A Flux-Corrected Transport Algorithm for Solving Generalized Continuity Equations. U.S. Naval Research Laboratory Memorandum Report NRL/MR/6410-93-7192 (1993)
22. S.D. Mayor, P.R. Spalart, and G.J. Tripoli, Application of a Perturbation Recycling Method in the Large-Eddy Simulation of a Mesoscale Convective Internal Boundary Layer, *J. Atmos. Sci.* **59**, 2385 (2002)
23. J.P. Bonnet, C. Coiffet, J. Delville, Ph. Druault, E. Lamballais, J.F. Largeau, S. Lardeau, and L. Perret, The Generation of Realistic 3D Unsteady Inlet Conditions for LES, AIAA 2003-0065 (2002)
24. M.J. Dwyer, E.G. Patton, and R.H. Shaw, *Boundary-Layer Meteorology*, **84**, 23-43 (1997)
25. A.M. Landsberg, T.R. Young, and J.P. Boris, An Efficient Parallel Method for solving Flows in Complex Three-Dimensional Geometries, AIAA Paper 94-0413 (1994)
26. S. Pal Arya, *Introduction to Micrometeorology*, Academic Press (1988)
27. C. Hirsch, *Numerical Computation of Internal and External Flows*, J. Wiley and Sons (1999)
28. H. Jasak, H.G. Weller, and A.D. Gosman, High Resolution NVD Differencing Scheme for Arbitrarily Unstructured Meshes, *Int. J. Numer. Meth. Fluids* **31**, 431 (1999)
29. J. Pullen, J.P. Boris, T.R. Young, G. Patnaik, and J.P. Iselin, Comparing Studies of Plume Morphology using a Puff Model and an Urban High-Resolution Model, Seventh Annual George Mason University Transport and Dispersion Modeling Workshop, June 2003
30. J.P. Boris, K. Obenschain, G. Patnaik, and T.R. Young, CT-ANALYST$^{\mathrm{TM}}$, Fast and Accurate CBR Emergency Assessment, Proceedings of the 2^{nd} International Conference on Battle Management, Williamsburg, VA, 4–8 November 2002

30 Years of FCT: Status and Directions

Rainald Löhner[1] and Joseph D. Baum[2]

[1] School of Computational Sciences, MS 4C7
George Mason University, Fairfax, VA 22030-4444, USA
rlohner@gmu.edu
[2] Advanced Technology Group
SAIC, McLean, VA 22102, USA
joseph.d.baum@saic.com

Summary. A somewhat historical perspective of the use of FCT for fluid dynamics is given. The particular emphasis is on large-scale blast problems. A comparison with other high-resolution CFD solvers is included to highlight the differences between them, as well as the relative cost. Results from test runs, as well as several relevant production runs are shown. Outstanding issues that deserve further investigation are identified.

1 Introduction

By the early 70's, computers had become sufficiently fast to allow the simulation of unsteady compressible flows with 'complex physics' (nonlinear source terms, arbitrary equations of state, large density / pressure / temperature ranges, etc.). A particularly disturbing observation made time and again was that low-order schemes, although overly dissipative, in many cases gave better results than high-order schemes. High-order schemes tended to have over/ undershoots or ripples in regions of high gradients, which in turn could lead to completely unphysical results (for example, premature ignition for combustion calculations). Ironically, while a tremendous amount of effort was still being devoted to perfecting the 'ultimate linear scheme', Godunov [15] by the end of the 60's had already proven that any such effort would be futile. No linear scheme of order higher than one could give monotonicity preserving results. The resolution of this quandary came with the birth of nonlinear schemes.

Any high-order scheme used to advance the solution either in time or between iterations towards steady-state may be written as

$$u^{n+1} = u^n + \Delta u = u^n + \Delta u^l + (\Delta u^h - \Delta u^l) = u^l + (\Delta u^h - \Delta u^l) \quad . \qquad (1)$$

Here Δu^h and Δu^l denote the increments obtained by the high- and low-order scheme respectively, and u^l is the monotone solution at time $t = t^{n+1}$ of the

low-order scheme. The idea behind any nonlinear scheme such as FCT is to limit the second term on the right-hand side of (1)

$$u^{n+1} = u^l + \lim(\Delta u^h - \Delta u^l) \quad , \tag{2}$$

in such a way that no new over/undershoots are created. Note that even though the original PDE as well as the low- and high-order schemes are linear, the resulting overall scheme is nonlinear, as it depends on the local behaviour of u.

While it became clear that the TVD concept was only enforceable in 1-D (see [16]: 'except in certain trivial cases, any method that is TVD in two space dimensions is at most first-order accurate'; 1-D TVD is being used unchanged to this day on an edge/face basis in 3-D production codes, and TVD has been supplanted by LED as an aim), the FCT concept realized a huge step forward with Zalesak's generalization [52] to schemes of arbitrary order and dimensions. Here was a way to apply FCT to schemes of any order, dimension, and, perhaps most importantly from an application point of view, grid topology. Zalesak's ideas were ported to Finite Element grids by Parrott and Christie [38] for scalar equations. Building on this work, Löhner, Morgan, Peraire and Vahdati [27], [28] added synchronization of limiters for systems of equations, found a steepener for contact discontinuities and added a Lapidus-type artificial viscosity [26] that removed the terracing problem for expansion fans. Further progress in the effective use of vector-machines by renumbering and grouping of elements, fast adaptive refinement with useful error indicators for transient problems [29], the switch from element- to edge-based data structures [34], extension to arbitrary Lagrangian-Eulerian (ALE) frames for problems with moving bodies [5] and mesh embedding [32] led to codes with the inherent geometrical flexibility of unstructured grids that were cost-competitive with structured grid codes. The remarkable combination of faster techniques and more powerful computers led to ever increasing problem size and application scope, as can be seen from Table 1. It is interesting to note that throughout a 15 year period the basic FEM-FCT algorithm for the simulation of transient shock problems has remained essentially unaltered.

Table 1. Increase of Problem Size

Size	Year	Problem	Machine
$> 10^2$	1983	Airfoil	ICL
$> 10^3$	1985	Forebody	CDC-205
$> 10^4$	1986	Train	Cray-XMP
$> 10^5$	1989	Train	Cray-2
$> 10^6$	1991	T-62	Cray-2
$> 10^7$	1994	WTC	Cray-M90
$> 10^8$	1998	Village	SGI-O2000

The increase in compute power also shifted the CFD bottleneck in industry from schemes (80's: CPU time as the competitive differentiator) to grid generation (90's: surface definition to mesh as the competitive differentiator) to process integration (00's: CAD to accurate solution as the competitive differentiator). By the beginning of the 21st century, all CFD codes used in industry were based on some form of unstructured grid. At the same time FCT was placed on a sound theoretical basis by Kuzmin et al. [21], [22], [23], [24], who also extended FCT to fully implicit time-marching and iterative limiting.

2 Basic Principles of FCT

Consider the system of conservation laws

$$\mathbf{u}_{,t} + \mathbf{F}^i_{,i} = \mathbf{S} \quad , \tag{3}$$

where $\mathbf{u}, \mathbf{F}, \mathbf{S}$ denote the unknowns, fluxes and source-terms. Any finite volume or finite element discretization will yield a discrete system of the form:

$$\mathbf{M}^{ij}\hat{u}^j_{,t} = R^i_s + C^{ij}\mathcal{F}_{ij} \quad . \tag{4}$$

Here, $\mathbf{M}, \hat{u}, R_s, C^{ij}, \mathcal{F}^{ij}$ denote the mass-matrix, vector of unknowns, right-hand side due to sources, edge-coefficients for fluxes and edge-fluxes respectively. Let us consider first the traditional TVD approach. For the standard Galerkin approximation we have

$$\mathcal{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j \quad , \tag{5}$$

i.e. an equal weighting of fluxes at the end-point of an edge. This (high-order) combination of fluxes is known to lead to an unstable discretization, and must be augmented by stabilizing terms to achieve a stable, low-order scheme. In what follows, we enumerate the most commonly used options in order to compare them to FCT. We start with those schemes that limit before evaluating fluxes in order to contrast them to FCT, where the limiting is performed after evaluating fluxes.

2.1 Limiting Before Flux Evaluation

If we assume that the flow variables are constant in the vicinity of the edge endpoints i, j, a discontinuity will occur at the edge midpoint. The evolution in time of this local flowfield was first obtained analytically by Riemann [43], and consists of a shock, a contact discontinuity and an expansion wave. More importantly, the flux at the discontinuity remains constant in time. One can therefore replace the average flux of the Galerkin approximation by this so-called Riemann flux. This stable scheme, which uses the flux obtained from

an exact Riemann solver, was first proposed by Godunov [15]. The flux is given by

$$\mathcal{F}_{ij} = 2\mathbf{f}(\mathbf{u}_{ij}^R) \quad , \tag{6}$$

where $\mathbf{u}_{ij}^R$ is the local exact solution of the Riemann problem to the Euler equations, expressed as

$$\mathbf{u}_{lr}^R = Rie(\mathbf{u}_l, \mathbf{u}_r), \tag{7}$$

where

$$\mathbf{u}_r = \mathbf{u}_i, \quad \mathbf{u}_l = \mathbf{u}_j \quad . \tag{8}$$

This scheme represents what one may call the 'ultimate first order scheme'. All waves are taken into account, and the basic underlying physics are well reproduced. In order to achieve a higher order scheme, the amount of inherent dissipation must be reduced. This implies reducing the magnitude of the difference $\mathbf{u}_i - \mathbf{u}_j$ by 'guessing' a smaller difference of the unknowns at the location where the Riemann flux is evaluated (i.e. the middle of the edge). The assumption is made that the function behaves smoothly in the vicinity of the edge. This allows the construction or 'reconstruction' of alternate values for the unknowns at the middle of the edge. The additional information required to achieve a scheme of higher order via these improved values at the middle of the edge can be obtained in a variety of ways:

- Through continuation and interpolation from neighbouring elements (Billey [7]);
- Via extension along the most aligned edge [49]; or
- By evaluation of gradients [50], [33], [34].

The last option is the one most commonly used, but carries a considerable computational overhead: 15 gradients for the unknowns in 3-D can account for a large percentage of CPU time.
The inescapable fact stated in Godunov's theorem that no linear scheme of order higher than one is free of oscillations implies that with these higher order extensions, some form of limiting will be required. For a review of these, see [48]. It is important to note that this form of limiting is done **before flux evaluation**, and that, strictly speaking, it should be performed with characteristic variables. A typical Godunov-based scheme therefore has four main cost components:

- Solution of the exact Riemann problem;
- Gradient-based reconstruction of higher order approximations to the left and right states;
- Forward/backward transformation from conservative to characteristic variables; and
- Limiting.

In the sequel, we will enumerate possible simplifications to each of these cost components, thereby deriving a whole spectrum of commonly used schemes. The solution of the (nonlinear) Riemann problem requires an iterative procedure which is expensive. Therefore, a considerable amount of effort has been devoted to obtain faster 'approximate Riemann solvers' that still retain as much of the physics as the basic Riemann problem [25], [44], [37]. A widely used solver of this class is the one derived by Roe [44], which may be written as:

$$\mathcal{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j - |\mathbf{A}^{ij}|(\mathbf{u}_i - \mathbf{u}_j), \tag{9}$$

where $|\mathbf{A}^{ij}|$ denotes the standard Roe matrix evaluated in the direction d^{ij}. Note that, as before, reducing the magnitude of the difference $\mathbf{u}_i - \mathbf{u}_j$ via reconstruction and limiting leads to schemes of higher order.
A further possible simplification can be made by replacing the Roe matrix by its spectral radius. This leads to a numerical flux function of the form

$$\mathcal{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j - \mid \lambda^{ij} \mid (\mathbf{u}_i - \mathbf{u}_j) \quad , \tag{10}$$

where

$$\mid \lambda^{ij} \mid = \mid v_{ij}^k \cdot S_k^{ij} \mid + c^{ij} \quad , \tag{11}$$

and v_{ij}^k and c^{ij} denote edge values, computed as nodal averages, of the fluid velocity and speed of sound respectively, and S_k^{ij} is the unit normal vector associated with the edge (i.e. in 3-D the unit normal of the finite volume surface associated with the edge). This can be considered as a centered difference scheme plus a second order dissipation operator, leading to a first order, monotone scheme. As before, a higher order scheme can be obtained by a better approximation to the 'right' and 'left' states of the 'Riemann problem'. Given that for smooth problems through the use of limiters the second order dissipation $|\mathbf{u}_i - \mathbf{u}_j|$ reverts to fourth order dissipation [19], [31], and that limiting requires a considerable number of operations, the next possible simplification is to replace the limiting procedure by a pressure sensor function. A scheme of this type may be written as

$$\mathcal{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j - \mid \lambda^{ij} \mid \left[\mathbf{u}_i - \mathbf{u}_j + \frac{\beta}{2}\mathbf{l}_{ji} \cdot (\nabla \mathbf{u}_i + \nabla \mathbf{u}_j)\right] \quad , \tag{12}$$

where $0 < \beta < 1$ denotes a pressure sensor function of the form [41]

$$\beta = 1 - \frac{p_i - p_j + 0.5\ \mathbf{l}_{ji} \cdot (\nabla p_i + \nabla p_j)}{|p_i - p_j| + |0.5\ \mathbf{l}_{ji} \cdot (\nabla p_i + \nabla p_j)|} \quad , \tag{13}$$

and $\mathbf{l}_{ji} = \mathbf{x}_j - \mathbf{x}_i$. For $\beta = 0, 1$, second and fourth order damping operators are obtained respectively. Several forms are possible for the sensor function β [36]. Although this discretization of the Euler fluxes looks like a blend of second and fourth order dissipation, it has no adjustable parameters. The scalar dissipation operator presented above still requires the evaluation of gradients.

This can be quite costly for Euler simulations: for a typical multistage scheme, more than 40% of the CPU-time is spent in gradient-operations, even if a new dissipation operator is only required at every other stage. The reason lies in the very large number of gradients required: 15 for the unknowns in 3-D, and an additional 3 for the pressure. An alternative would be to simplify the combination of second and fourth order damping operators by writing out explicitly these operators:

$$d_2 = \lambda^{ij}(1-\beta)\left[\mathbf{u}_i - \mathbf{u}_j\right] \quad , \quad d_4 = \lambda^{ij}\beta\left[\mathbf{u}_i - \mathbf{u}_j + \frac{\mathbf{l}_{ji}}{2}\cdot(\nabla\mathbf{u}_i + \nabla\mathbf{u}_j)\right] \quad . \tag{14}$$

Performing a Taylor series expansion in the direction of the edge, we have

$$\mathbf{u}_i - \mathbf{u}_j + \frac{\mathbf{l}_{ji}}{2}\cdot(\nabla\mathbf{u}_i + \nabla\mathbf{u}_j) \approx \frac{\mathbf{l}_{ji}^2}{4}\left[\left.\frac{\partial^2\mathbf{u}}{\partial l^2}\right|_j - \left.\frac{\partial^2\mathbf{u}}{\partial l^2}\right|_i\right] \quad . \tag{15}$$

This suggests the following simplification, which neglects the off-diagonal terms of the tensor of second derivatives:

$$\frac{\mathbf{l}_{ji}^2}{4}\left[\left.\frac{\partial^2\mathbf{u}}{\partial l^2}\right|_j - \left.\frac{\partial^2\mathbf{u}}{\partial l^2}\right|_i\right] \approx \frac{\mathbf{l}_{ji}^2}{4}\left[\nabla^2\mathbf{u}_j - \nabla^2\mathbf{u}_i\right] \quad , \tag{16}$$

and leads to the familiar blend of second and fourth order damping operators [20], [35]

$$\mathcal{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j - \mid\lambda^{ij}\mid(1-\beta)\left[\mathbf{u}_i - \mathbf{u}_j\right] - \mid\lambda^{ij}\mid\beta\frac{\mathbf{l}_{ji}^2}{4}\left[\nabla^2\mathbf{u}_j - \nabla^2\mathbf{u}_i\right] \quad . \tag{17}$$

Lax-Wendroff/Taylor-Galerkin

The essential feature of Lax-Wendroff/ Taylor-Galerkin schemes is the combination of time and space discretizations, leading to second order accuracy in both time and space. An edge-based two-step Taylor-Galerkin scheme can readily be obtained by setting the numerical flux to

$$\mathcal{F}_{ij} = 2\mathbf{f}(\mathbf{u}_{ij}^{n+\frac{1}{2}}) \quad , \tag{18}$$

where

$$\mathbf{u}_{ij}^{n+\frac{1}{2}} = \frac{1}{2}(\mathbf{u}_i + \mathbf{u}_j) - \frac{\Delta t}{2}\left.\frac{\partial\mathbf{f}^j}{\partial x_j}\right|_{ij} \quad , \tag{19}$$

and $\left.\frac{\partial\mathbf{f}^j}{\partial x_j}\right|_{ij}$ is computed on each edge and given by either

$$\left.\frac{\partial\mathbf{f}^j}{\partial x_j}\right|_{ij} \approx \frac{\mathbf{l}_{ij}}{\mathbf{l}_{ij}^2}\cdot(\mathbf{F}^i - \mathbf{F}^j) \quad \text{or} \quad \left.\frac{\partial\mathbf{f}^j}{\partial x_j}\right|_{ij} \approx \frac{\mathbf{D}_{ij}}{\mathbf{D}_{ij}^2}\cdot(\mathbf{F}^i - \mathbf{F}^j) \quad , \tag{20}$$

where $\mathbf{D}_{ij}$ denotes the edge-coefficients for the advective terms obtained from the Galerkin approximation. The major advantage of this scheme lies in its speed, since there is no requirement of gradient computations, as well as limiting procedures for smooth flows. An explicit numerical dissipation (e.g. in the form of a Lapidus viscosity [26]) is needed to model flows with discontinuities. Taylor-Galerkin schemes by themselves are of little practical use for problems with strong shocks or other discontinuities. However, they provide high order schemes with the best cost/performance ratio for the flux-corrected transport schemes presented below.

2.2 Limiting After Flux Evaluation

Limiting after flux evaluation is the key idea inherent to all FCT schemes. If we focus on high order schemes of the Lax-Wendroff/Taylor-Galerkin family, the high order increment may be written as

$$\mathbf{M}_l \Delta \mathbf{u}^h = \mathbf{r} + (\mathbf{M}_l - \mathbf{M}_c)\Delta \mathbf{u}^h \quad . \tag{21}$$

Here $\mathbf{M}_l$ denotes the diagonal, lumped mass-matrix, and $\mathbf{M}_c$ the consistent finite element mass-matrix. The low order scheme is simply given by

$$\mathbf{M}_l \Delta \mathbf{u}^l = \mathbf{r} + c_d(\mathbf{M}_c - \mathbf{M}_l)\mathbf{u}^n \quad , \tag{22}$$

i.e. lumped mass-matrix plus sufficient diffusion to keep the solution monotonic. Subtracting these two equations yields the antidiffusive edge contributions

$$(\Delta \mathbf{u}^h - \Delta \mathbf{u}^l) = \mathbf{M}_l^{-1}(\mathbf{M}_l - \mathbf{M}_c)(c_d \mathbf{u}^n + \Delta \mathbf{u}^h) \quad . \tag{23}$$

Note that no physical fluxes appear in the antidiffusive edge contributions. This may also be interpreted as: advance the physical fluxes with extra diffusion, thus assuring transport, conservation, etc. Thereafter, perform the antidiffusive step to enhance the solution as much as possible without violating monotonicity principles. The simplicity of the antidiffusive edge contributions for this class of scheme makes it both fast and very general, and has been one of the main reasons why this scheme has served the CFD community for more than 15 years without major alterations.

Let us treat in more detail limiting after flux evaluation. If we consider an isolated point surrounded by elements, the task of the limiting procedure is to insure that the increments or decrements due to the antidiffusive contributions do not exceed a prescribed tolerance (see Figure 1).

In the most general case, the contributions to a point will be a mix of positive and negative contributions. Given that the antidiffusive element/edge contributions (*AEC*'s) will be limited, i.e. multiplied by a number $0 \leq C_{el} \leq 1$, it may happen that after limiting all positive or negative contributions vanish.

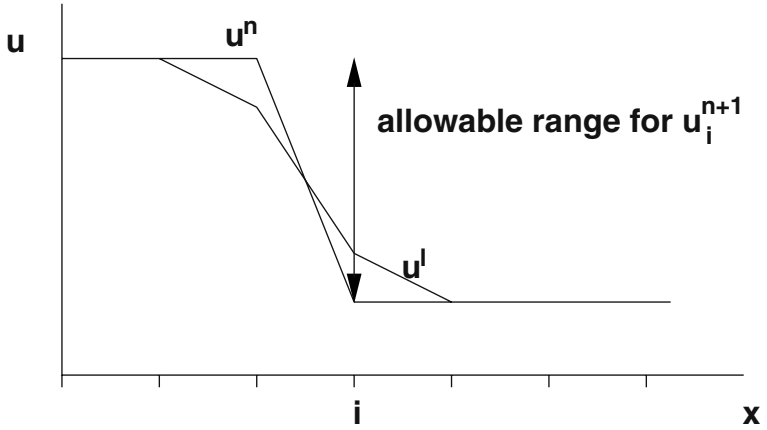


Fig. 1. Limiting procedure

The largest increment (decrement) will occur when only the positive (negative) contributions are considered. For this reason, we must consider what happens if only positive or only negative contributions are added to a point. The comparison of the allowable increments and decrements with these all-positive and all-negative contributions then yields the maximum allowable percentage of the *AEC*'s that may be added or subtracted to a point. On the other hand, an element/edge may contribute to a number of nodes, and in order to maintain strict conservation, the limiting must be performed for all the element/edge node contributions in the same way. Therefore, a comparison for all the nodes of an element/edge is performed, and the smallest of the evaluated percentages that applies is retained.

Defining the following quantities:

$P_i^{\pm}$: the sum of all positive (negative) element/edge contributions to node i

$$P_i^{\pm} = \sum_{el} \left\{ \begin{matrix} \max \\ \min \end{matrix} \right\} (0, AEC_{el}) \quad ; \tag{24}$$

$Q_i^{\pm}$: the maximum (minimum) increment node i is allowed to achieve

$$Q_i^{\pm} = u_i^{\begin{matrix} \max \\ \min \end{matrix}} - u_i^l \quad ; \tag{25}$$

the ratio of positive and negative contributions that ensure monotonicity is given by

$$R^{\pm} := \begin{cases} \min(1, Q^{\pm}/P^{\pm}) & P^+ > 0 > P^- \\ 0 \quad \text{otherwise} \end{cases} \quad . \tag{26}$$

For the elements/edges, the final value taken is the most conservative:

$$C_{el} = \min(element/edge\ nodes) \begin{cases} R^+ & \text{if } EC > 0 \\ R^- & \text{if } EC < 0 \end{cases} . \tag{27}$$

The allowed value $u_i^{\max\atop\min}$ is taken between each point and its nearest neighbours. For element-based schemes, it may be obtained in three steps as follows:

a) Maximum (minimum) nodal unknowns of u^n and u^l:

$$u_i^* = \begin{Bmatrix} \max \\ \min \end{Bmatrix} (u_i^l, u_i^n) \ ; \tag{28}$$

b) Maximum (minimum) nodal value of element/edge:

$$u_e^* = \begin{Bmatrix} \max \\ \min \end{Bmatrix} (u_A^*, u_B^*, ..., u_C^*) \ ; \tag{29}$$

c) Maximum (minimum) unknowns of all elements/edges surrounding node i:

$$u_i^{\max\atop\min} = \begin{Bmatrix} \max \\ \min \end{Bmatrix} (u_1^*, u_2^*, ..., u_m^*) \ . \tag{30}$$

A number of variations are possible for $u_i^{\max\atop\min}$. For example, the so-called 'clipping limiter' is obtained by setting $u_i^* = \{^{\max}_{\min}\} u_i^l$ in a), i.e., by not looking back to the solution at the previous timestep or iteration, but simply comparing nearest neighbour values at the new timestep or iteration for the low-order scheme. As remarked before, the limiting is based solely on the unknowns u, not on a ratio of differences as in most TVD schemes. Table 2 summarizes the main ingredients of high-resolution schemes, indirectly comparing the cost of most current flow solvers.

Table 2. Ingredients of CFD Solvers

Solver	Riemann	Gradient	Char.Transf.	Limiting
Classic Godunov	Yes	Yes	Yes	Yes
Consvar Godunov	Yes	Yes	No	Yes
Consvar Roe	Approx	Yes	No	Yes
Scal. Dissip.	No	Yes	No	Yes
Scal. Edge 2/4	No	Yes	No	No
Scal. Lapl 2/4	No	No	No	No
Taylor-Galerkin	No	No	No	No
TG-FCT	No	No	No	Yes

2.3 Test Cases

We include two simple test cases that demonstrate the performance of the edge-based FEM-FCT Euler solver. In both cases, FCT was run with limiter synchronization on the density and energy.

a) **Shock Tube**: This is a classic example, which was used repeatedly to compare different Euler solvers [47]. Initially, a membrane separates two fluid states given by $\rho_1 = 1.0, \mathbf{v}_1 = 0.0, p_1 = 1.0$ and $\rho_2 = 0.1, \mathbf{v}_2 = 0.0, p_2 = 0.1$. The membrane ruptures, giving rise to a shock, a contact discontinuity and a rarefaction wave. Figure 2a shows the surface mesh and the surface contours of the density. A line-cut through the 3-D mesh is compared to the exact solution in Figure 2b. Note that the number of points appearing here corresponds to faces of tetrahedra being cut, i.e., is 2-3 times the number of actual points. One can see that the shock and contact discontinuities are captured over 2 or 4 elements respectively.

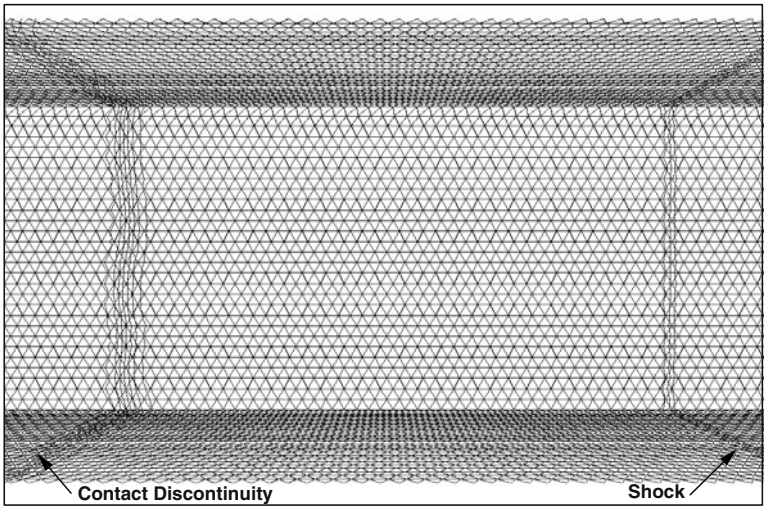


Fig. 2a. Shock tube: surface mesh and density

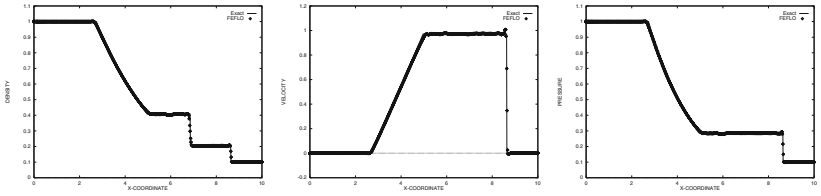


Fig. 2b. Shock tube: comparison to exact values

b) **Shock Diffraction Over a Wall**: The second example shown is typical of some of the large-scale blast simulations carried out with FCT schemes over the last decade [1], [45], [2], [3], [4], [30], and is taken from [42]. The outline of the domain is shown in Figure 3a. Surface pressures for an axisymmetric and 3-D run, together with comparison to photographs from experiments is given in Figures 3b,c. The comparison to experimental results at different stations is shown in Figures 3d. As one can see, FCT schemes yield excellent results for this class of problems.

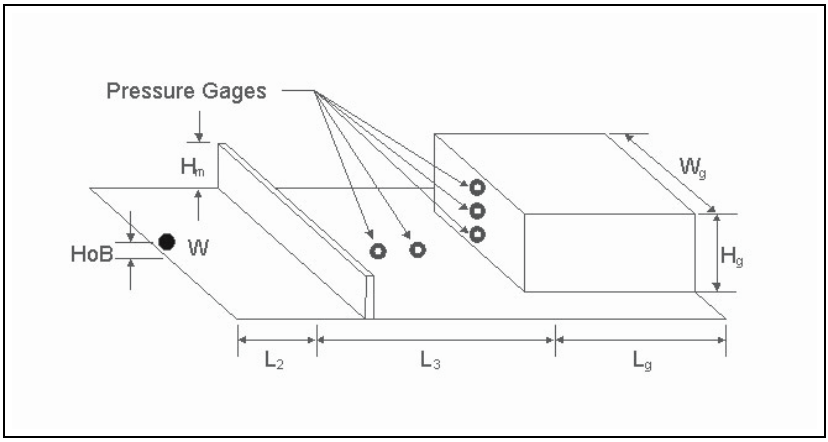


Fig. 3a. Shock diffraction over a wall

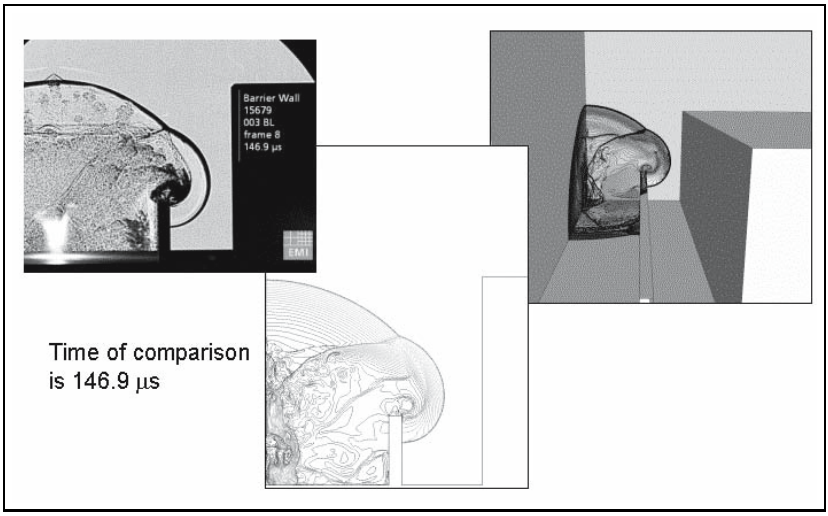


Fig. 3b. Results at time $t = 146\mu sec$

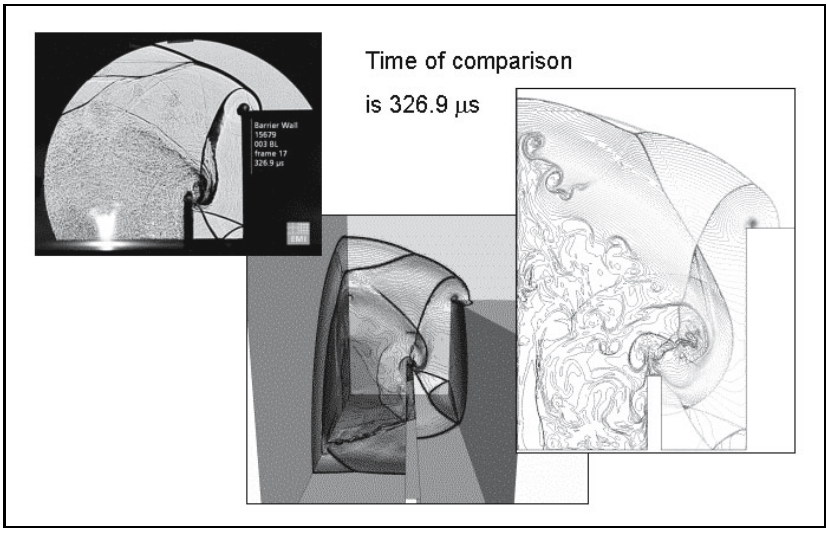


Fig. 3c. Results at time $t = 327 \mu sec$

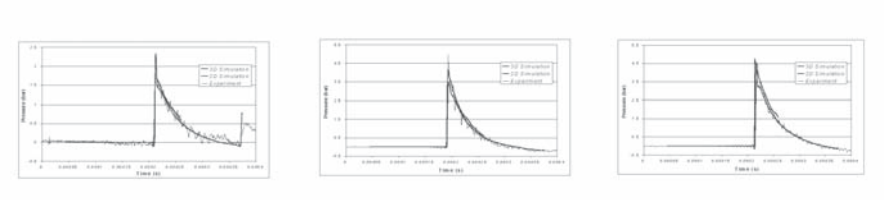


Fig. 3d. Comparison to experimental values

For more verification runs, see [3], [12], [27], [28], [30].

3 Some Landmark Runs

In this section we list a few memorable runs that were conducted with FEM-FCT over the years.

a) **T-62**: This simulation considers the impact of a strong shockwave on a main battlefield tank. The particular tank happens to be a T-62, of which there seem to be an abundance in the Middle East. The aim of the run was to gauge the possible effects of strong shockwaves on such tanks. This run was the first to exceed 2 Mtets, and was conducted on a CRAY-2. It was also the first run of this magnitude to make extensive use of adaptive refinement. Even with an initial mesh grading that exhibited smaller elements close to the vehicle, the mesh was adaptively refined (and coarsened) to 2 levels every 5 timesteps. Figures 4a,b show the surface discretization and pressure at different times. The passage of the shockwave over the vehicle is readily visible. For more information, see [1].

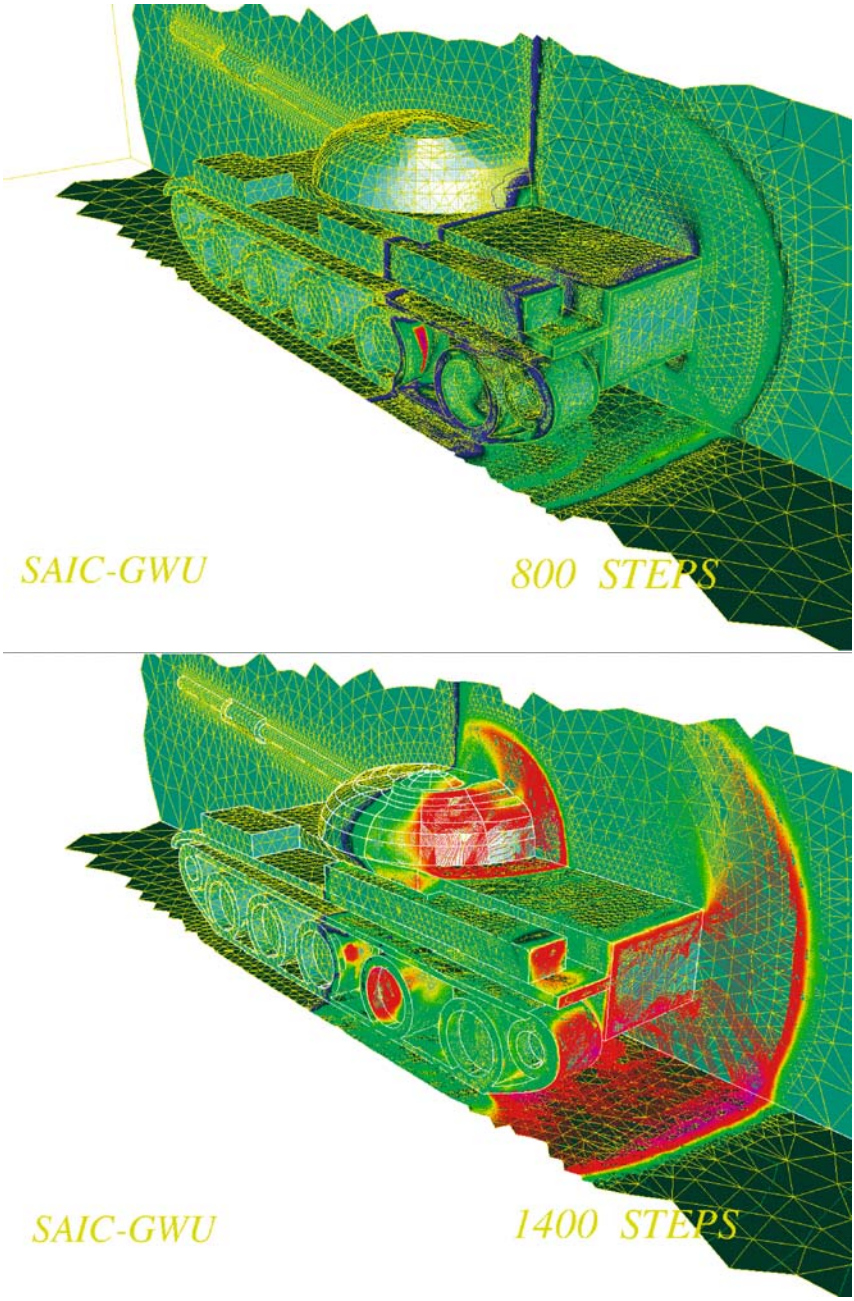


Fig. 4a,b. Shock interaction with T-62

b) **World Trade Center**: The World Trade Center has always been a symbol of global capitalism, and has therefore been a target for terrorist attacks almost since its inception. In 1993 a powerful car bomb was detonated in one of the lower parking levels, causing loss of life and extensive damage. In order to aid in the forensic studies that followed, a simulation was conducted. Data was assembled from a variety of sources: drawings, blueprints, videos, CAD-data. The distribution of cars (more than 400 of them) was set from a statistical distribution of 10 typical models. This run was the first to exceed 20 Mtets, and was conducted on a CRAY-2M. Figure 5 clearly shows the shock wave at approximately $T = 96$ *msec*. For more information, see [4].

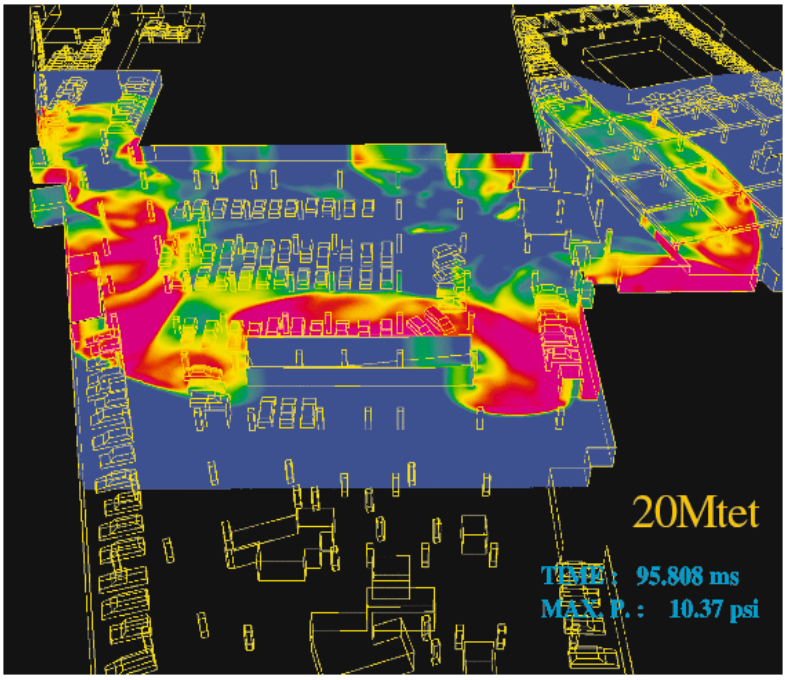


Fig. 5. Shock propagation in World Trade Center

c) **Truck**: This simulation considers the interaction of a strong shockwave with a typical command and control center truck. The aim of the run was to demonstrate the feasibility of conducting fully coupled fluid-structure interaction calculations for this class of problems. The mesh was allowed to move close to the vehicle, and the FEM-FCT algorithm was cast in an Arbitrary Lagrangian-Eulerian (ALE) frame of reference. Approximately 10 global remeshings and countless local remeshings were required to accommodate the severe deformation of the structure. The response may be seen in Figure 6. For more information, see [6].

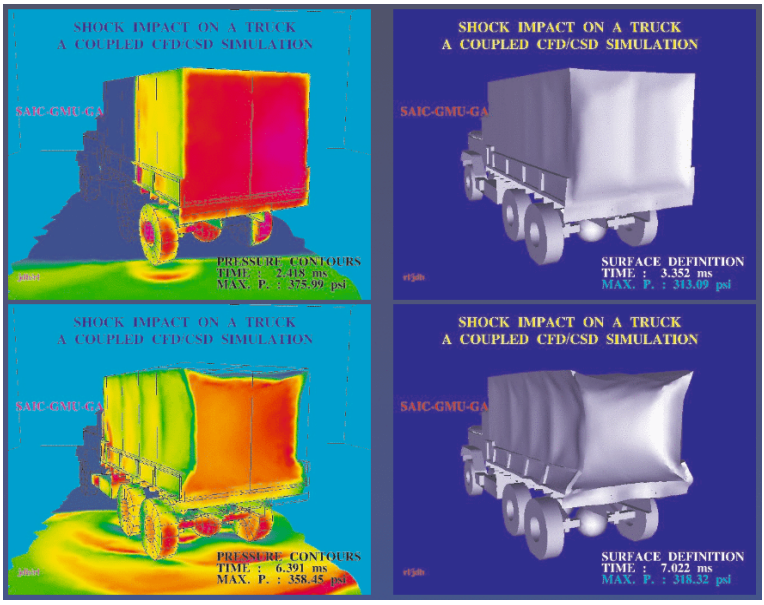


Fig. 6a,b. Shock interaction with a truck

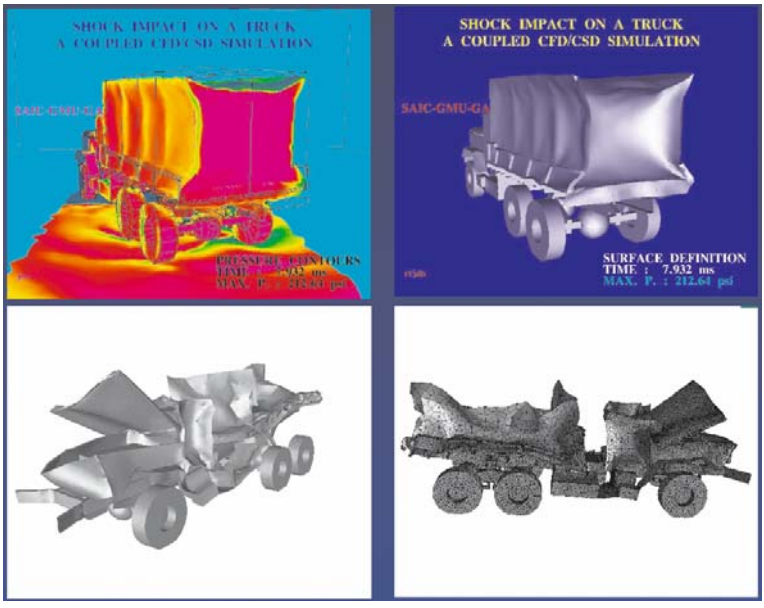


Fig. 6c,d. Shock interaction with a truck

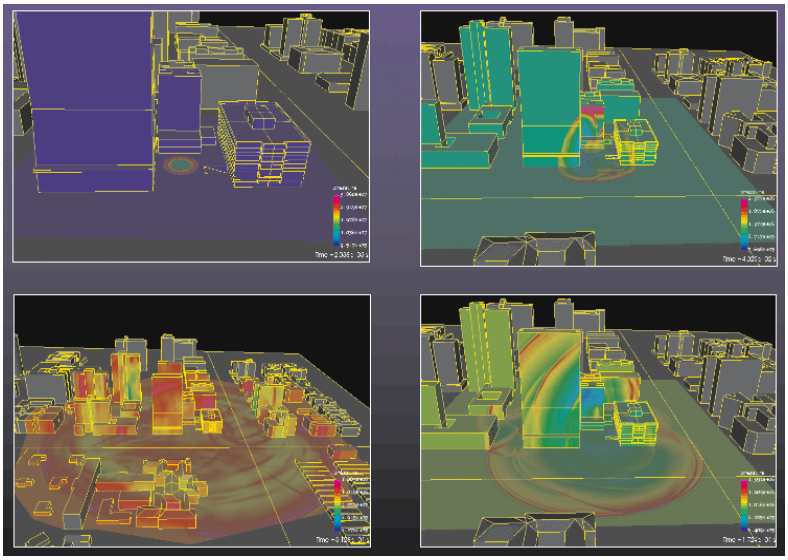


Fig. 7a. Blast in city

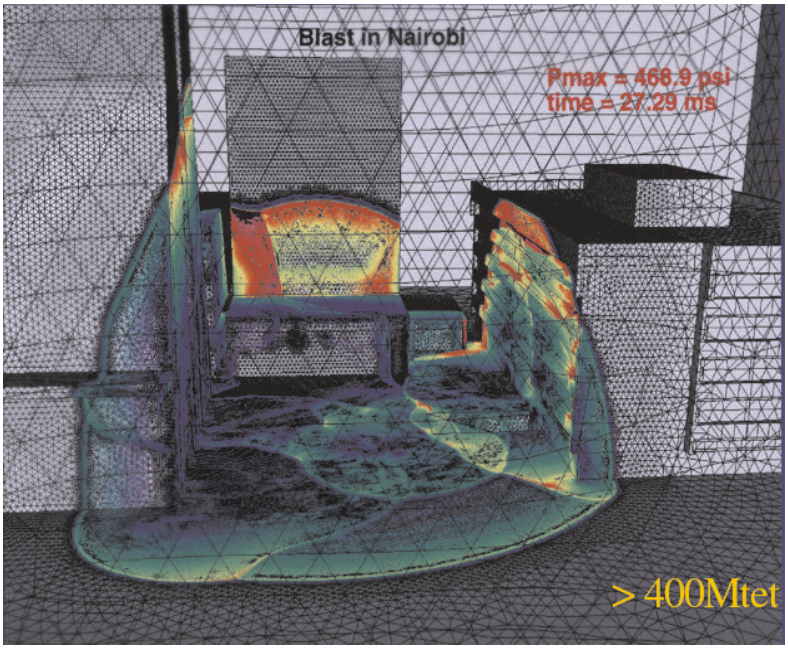


Fig. 7b. Blast in city (detail showing adaptive refinement)

d) **Blast in City**: This simulation considers a strong explosion in a typical city setting. The particular location chosen corresponds to Nairobi, where a powerful bomb was detonated close to the American embassy in 1998. This run was the first to reach 500 Mtets in a fully adaptive, transient setting, and was carried out on a multiprocessor SGI O2000 machine. The propagation of the blast wave can be discerned from Figure 7a. The adaptive refinement of the mesh can be seen in Figure 7b.

d) **Generic Weapon Fragmentation**: This simulation considers the detonation and fragmentation of a generic weapon. It is performed using a JWL model for the fluid and FEM-FCT as the basic flow solver, and a large-deformation structural dynamics code for the casing. The flow mesh is not moving, i.e. the structure inside the flowfield is treated using the embedded adaptive approach [32]. The CSD domain was modeled with approximately 66 Khex elements corresponding to 1,555 fragments whose mass distribution matches statistically the mass distribution encountered in experiments. The structural elements were assumed to fail once the average strain in an element exceeded 60%.

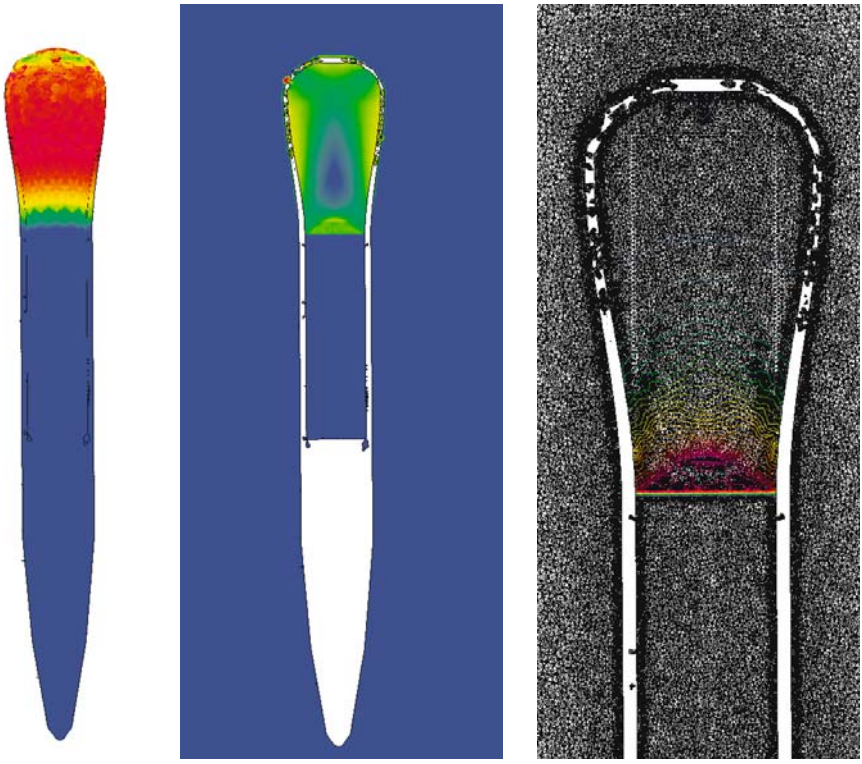


Fig. 8a-c. CSD/flow velocity/mesh at 68 ms

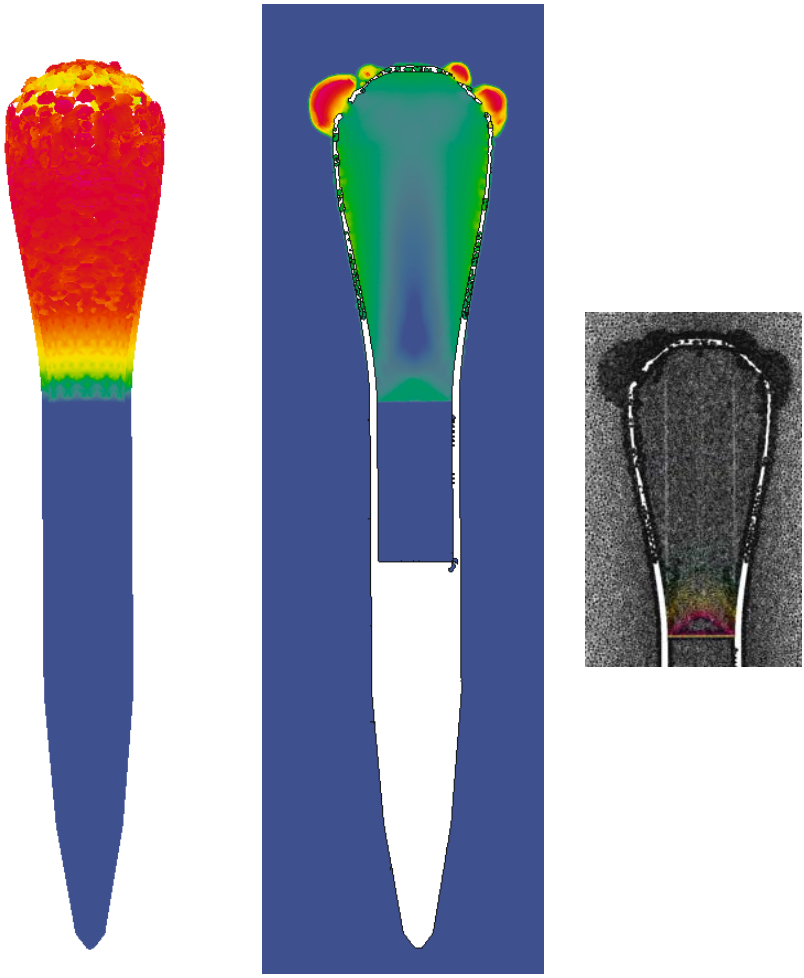


Fig. 8d-f. CSD/flow velocity/mesh at 102 ms

The CFD mesh was refined to 3 levels in the vicinity of the solid surface. Additionally, the mesh was refined based on the modified interpolation error indicator proposed in [29], using the density as indicator variable. Adaptive refinement was invoked every 5 timesteps during the coupled CFD/CSD run. The CFD mesh started with 39 Mtet, and ended with 72 Mtet. Figures 8a-f show the structure as well as the pressure contours in a cut plane at two times during the run. The detonation wave is clearly visible, as well as the thinning of the structural walls and the subsequent fragmentation.

4 Outstanding Issues

In this section, we list some of the the outstanding issues in CFD solvers based on FCT.

4.1 Steepening

The antidiffusive step in FCT is designed to steepen the low-order solution obtained at the new timestep or iteration. In some cases, particularly for high-order schemes of order greater than two, the antidiffusive step can flatten the profile of the solution even further, or lead to an increase of wiggles and noise. This is the case even though the solution remains within its allowed limits. A situation where this is the case can be seen from Figure 9.

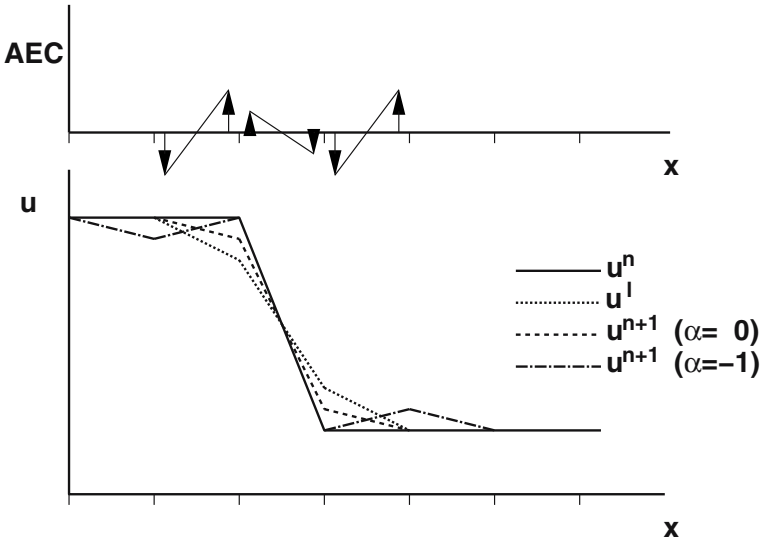


Fig. 9. Steepener for FCT

This type of behaviour can be avoided if the antidiffusive flux is either set to zero or reversed. A simple way to decide when to reverse the antidiffusive fluxes is to compute the scalar product of the low-order solution at the new timestep or iteration and the antidiffusive element/edge contributions:

$$\nabla u^l \cdot AEC < 0 \Rightarrow AEC = -\alpha \cdot AEC \quad , \tag{31}$$

with $0 < \alpha < 1$. Values of α greater than unity lead to steepening. This can be beneficial in some cases, but is highly dangerous, as it can lead to unphysical solutions (e.g. spurious contact discontinuities) in complex applications. Although this type of steepening works well in practice, a more formal analysis/treatment would be beneficial.

4.2 Limiting for Systems of Equations

The results available in the literature [9], [8], [10], [52], [38], [27], [28], [53] indicate that with FCT results of excellent quality can be obtained for a single PDE, e.g. the scalar advection equation. However, in the attempt to extend the limiting process to systems of PDEs no immediately obvious or natural limiting procedure becomes apparent. Obviously, for 1-D problems one could advect each simple wave system separately, and then assemble the solution at the new time step. However, for multidimensional problems such a splitting is not possible, as the acoustic waves are circular in nature. Finite-Difference FCT codes used for production runs [13], [14] have so far limited each equation separately, invoking operator-splitting arguments. This approach does not always give very good results, as may be seen from [47] comparison of schemes for the Riemann problem, and has been a point of continuing criticism by those who prefer to use the more costly Riemann-solver-based, essentially one-dimensional TVD-schemes [25], [44], [37], [18], [48], [11], [51]. An attractive alternative is to introduce 'system character' for the limiter by combining the limiters for all equations of the system. Many variations are possible and can be implemented, giving different performance for different problems. Some of the possibilities are listed here, with comments where empirical experience is available.

a) **Independent treatment of each equation as in operator-split FCT**: This is the least diffusive method, tending to produce an excessive amount of ripples in the non-conserved quantities (and ultimately also in the conserved quantities).

b) **Use of the same limiter (C_{el}) for all equations**: This produces much better results, seemingly because the phase errors for all equations are 'synchronized'. This was also observed by Harten and Zwaas [17] and Zhmakin and Fursenko [54] for a class of schemes very similar to FCT. We mention the following possibilities:

   - Use of a certain variable as 'indicator variable' (e.g. density, pressure, entropy).
   - Use of the minimum of the limiters obtained for the density and the energy ($C_{el} = \min(C_{el}(\rho), C_{el}(\rho e))$) : this produces acceptable results, although some undershoots for very strong shocks are present. This option is currently the preferred choice for strongly unsteady flows characterized by propagating and/or interacting shock waves.
   - Use of the minimum of the limiters obtained for the density and the pressure ($C_{el} = \min(C_{el}(\rho), C_{el}(p))$) : this again produces acceptable results, particularly for steady-state problems.

Limiting Any Set of Quantities

A general algorithm to limit any set of quantities may be formulated as follows:

- Define a new set of (non-conservative) variables $\mathbf{u}'$;
- Transform : $\mathbf{u}, \mathbf{u}^l \rightarrow \mathbf{u}', \mathbf{u}'^l$ and see how much $\mathbf{u}'$ can change at each point $\Rightarrow \Delta\mathbf{u}'|_{\min}^{\max}$;
- Define a mean value of $\mathbf{u}^l$ in the elements and evaluate:

$$\Delta\mathbf{u}' = \mathbf{A}(\overline{\mathbf{u}}^l) \cdot \Delta\mathbf{u} \quad , \tag{32}$$

- Limit the transformed increments $\Delta\mathbf{u}' \Rightarrow C'_{el} \Rightarrow \Delta\mathbf{u}'' = \lim(\Delta\mathbf{u}')$;
- Transform back the variables and add:

$$\Delta\mathbf{u}^* = \mathbf{A}^{-1}(\overline{\mathbf{u}}^l) \cdot \Delta\mathbf{u}'' = \mathbf{A}^{-1}(\overline{\mathbf{u}}^l) \cdot C'_{el} \cdot \mathbf{A}(\overline{\mathbf{u}}^l) \cdot \Delta\mathbf{u} \quad . \tag{33}$$

4.3 Terracing

Terracing typically occurs when smooth profiles are transported over many gridpoints. Without incurring a loss of monotonicity, the profiles become ragged, exhibiting terraces (hence the name). In most cases, terracing occurs due to antidiffusive fluxes that are too strong, as in the case discussed above for steepeners. A usual way to suppress steepening is by adding a small amount of background dissipation, typically in the form of a fourth-order damping or via Lapidus smoothing [26]. While these empirical fixes work well, it would be highly desirable to have a formal understanding of this phenomenon.

5 Conclusions and Outlook

FCT algorithms have proven to be an invaluable ingredient of many production codes over the last three decades. They have been used in a large variety of fields, such as fluid dynamics, plasma physics, petroleum engineering and electromagnetics. The present paper has given a somewhat historical perspective for fluid dynamics, with particular emphasis on large-scale blast problems. A comparison with other high-resolution CFD solvers has been included to highlight the differences between them, as well as the relative cost. Results from test runs, as well as several relevant production runs have been shown. Finally, several outstanding issues that deserve further investigation have been identified.

While many other CFD techniques have appeared in the last two decades, FCT, due to its very favourable cost/accuracy ratio, has been able to survive almost unchanged. The increase in compute power may, perhaps, shift the emphasis to more costly, refined schemes. However, for the truly large-scale problems at present there exist few alternatives to FCT.

6 Acknowledgements

It is our great pleasure to acknowledge the input and stimulus provided by the many colleagues with whom we had the opportunity to work over the years. In particular the teams at GMU, Swansea (Wales, UK), NRL-LCP&FD, SAIC (ATG) and NASA (LARC, GSFC). We would also like to take the opportunity to thank Cray Research, IBM and SGI for many free hours on their machines over the years. The work compiled here would not have been possible without the steady support received from such organizations as the Defense Nuclear Agency, the Defense Threat Reduction Agency and the Air Force Office of Scientific Research. This support is gratefully acknowledged.

References

1. Baum, J.D. and R. Löhner - Numerical Simulation of Shock Interaction with a Modern Main Battlefield Tank; AIAA-91-1666 (1991).
2. Baum, J.D., H. Luo and R. Löhner - Numerical Simulation of a Blast Inside a Boeing 747; AIAA-93-3091 (1993).
3. Baum, J.D. and R. Löhner - Numerical Simulation of Shock-Box Interaction Using an Adaptive Finite Element Scheme; AIAA J. **32**, 4, 682-692 (1994).
4. Baum, J.D., H. Luo and R. Löhner - Numerical Simulation of Blast in the World Trade Center; AIAA-95-0085 (1995).
5. Baum, J.D., H. Luo and R. Löhner - Validation of a New ALE, Adaptive Unstructured Moving Body Methodology for Multi-Store Ejection Simulations; AIAA-95-1792 (1995).
6. Baum, J.D., H. Luo, R. Löhner, C. Yang, D. Pelessone and C. Charman - A Coupled Fluid/Structure Modeling of Shock Interaction with a Truck; AIAA-96-0795 (1996).
7. Billey, V., J. Périaux, P. Perrier, B. Stoufflet - 2-D and 3-D Euler Computations with Finite Element Methods in Aerodynamic; International Conference on Hypersonic Problems, Saint-Etienne, Jan. 13-17 (1986).
8. Book, D.L., J.P. Boris and K. Hain - Flux-corrected Transport. II. Generalizations of the Method; J. Comp. Phys. **18**, 248-283 (1975).
9. Boris, J.P. and D.L. Book - Flux-corrected Transport. I. SHASTA, a Transport Algorithm that works; J. Comp. Phys. **11**, 38-69 (1973).
10. Boris, J.P. and D.L. Book - Flux-corrected Transport. III. Minimal-Error FCT Algorithms; J. Comp. Phys. **20**, 397-431 (1976).
11. Colella, P. - Multidimensional Upwind Methods for Hyperbolic Conservation Laws; LBL-17023, Preprint (1983).
12. deFainchtein, R, S.T. Zalesak, R. Löhner and D.S. Spicer - Finite Element Simulation of a Turbulent MHD System: Comparison to a Pseudo-Spectral Simulation; Comp. Phys. Comm. **86**, 25-39 (1995).
13. Fry, M.A. and D.L. Book - Adaptation of Flux-Corrected Transport Codes for Modelling Dusty Flows; Proc. 14th Int. Symp. on Shock Tubes and Waves (R.D. Archer and B.E. Milton eds.), New South Wales University Press (1983).
14. Fyfe, D.E., J.H. Gardner, M. Picone and M.A. Fry - Fast Three-Dimensional Flux-Corrected Transport Code for Highly Resolved Compressible Flow Calculations; Springer Lecture Notes in Physics 218, 230-234, Springer Verlag (1985).

15. Godunov, S.K. - Finite Difference Method for Numerical Computation of Discontinuous Solutions of the Equations of Fluid Dynamics; Mat. Sb. **47**, 271-306 (1959).
16. Goodman, J.B. and R.J. LeVeque - On the Accuracy of Stable Schemes for 2D Scalar Conservation Laws; Math. Comp. **45**, 15-21 (1985).
17. Harten, A., and G. Zwaas - Self-Adjusting Hybrid Schemes for Shock Computations; J. Comp. Phys. **6**, 568-583 (1972).
18. Harten, A. - High Resolution Schemes for Hyperbolic Conservation Laws; J. Comp. Phys. **49**, 357-393 (1983).
19. Hirsch, C. - Numerical Computation of Internal and External Flow; J. Wiley & Sons (1991).
20. Jameson, A., W. Schmidt and E. Turkel - Numerical Solution of the Euler Equations by Finite Volume Methods using Runge-Kutta Time-Stepping Schemes; AIAA-81-1259 (1981).
21. Kuzmin, D. - Positive Finite Element Schemes Based on the Flux-Corrected Transport Procedure; pp. 887-888 in Computational Fluid and Solid Mechanics, Elsevier (2001).
22. Kuzmin, D. and S. Turek - Explicit and Implicit High-Resolution Finite Element Schemes Based on the Flux-Corrected-Transport Algorithm; pp. 133-143 in Proc. 4th European Conf. Num. Math. and Advanced Appl. (F. Brezzi et al. eds.), Springer-Verlag, (2002)
23. Kuzmin, D. and S. Turek - Flux Correction Tools for Finite Elements; J. Comp. Phys. 175, 525-558 (2002).
24. Kuzmin, D., M. Möller and S. Turek - Multidimensional FEM-FCT Schemes for Arbitrary Time-Stepping; Int. J. Numer. Meth. Fluids 42, 265-295 (2003).
25. van Leer, B. - Towards the Ultimate Conservative Scheme. II. Monotonicity and Conservation Combined in a Second Order Scheme; J. Comp. Phys. **14**, 361-370 (1974).
26. Löhner, R., K. Morgan and J. Peraire - A Simple Extension to Multidimensional Problems of the Artificial Viscosity due to Lapidus; Comm. Appl. Num. Meth. **1**, 141-147 (1985).
27. Löhner, R., K. Morgan, J. Peraire and M. Vahdati - Finite Element Flux-Corrected Transport (FEM-FCT) for the Euler and Navier-Stokes Equations; Int. J. Num. Meth. Fluids **7**, 1093-1109 (1987).
28. Löhner, R., K. Morgan, M. Vahdati, J.P. Boris and D.L. Book - FEM-FCT: Combining Unstructured Grids with High Resolution; Comm. Appl. Num. Meth. **4**, 717-730 (1988).
29. Löhner, R. and J.D. Baum - Adaptive H-Refinement on 3-D Unstructured Grids for Transient Problems; Int. J. Num. Meth. Fluids **14**, 1407-1419 (1992).
30. Löhner, R., Chi Yang, J.D. Baum, H. Luo, D. Pelessone and C. Charman - The Numerical Simulation of Strongly Unsteady Flows With Hundreds of Moving Bodies; Int. J. Num. Meth. Fluids **31**, 113-120 (1999).
31. Löhner, R. - Applied CFD Techniques; J. Wiley & Sons (2001).
32. Löhner, R., J.D. Baum, E.L. Mestreau, D. Sharov, Ch. Charman and D. Pelessone - Adaptive Embedded Unstructured Grid Methods; AIAA-03-1116 (2003).
33. Luo, H., J.D. Baum, R. Löhner and J. Cabello - Adaptive Edge-Based Finite Element Schemes for the Euler and Navier-Stokes Equations; AIAA-93-0336 (1993).
34. Luo, H., J.D. Baum and R. Löhner - Edge-Based Finite Element Scheme for the Euler Equations; AIAA J. **32**, 6, 1183-1190 (1994).

35. Mavriplis, D. - Three-Dimensional Unstructured Multigrid for the Euler Equations; AIAA-91-1549-CP (1991).
36. Mestreau, E., R. Löhner and S. Aita - TGV Tunnel-Entry Simulations Using a Finite Element Code with Automatic Remeshing; AIAA-93-0890 (1993).
37. Osher, S. and F. Solomon - Upwind Difference Schemes for Hyperbolic Systems of Conservation Laws; Math. Comp. **38**, 339-374 (1982).
38. Parrott, A.K. and M.A. Christie - FCT Applied to the 2-D Finite Element Solution of Tracer Transport by Single Phase Flow in a Porous Medium; Proc. ICFD-Conf. Num. Meth. in Fluid Dyn. (K.W. Morton and Baines eds.), Reading, Academic Press (1986).
39. Patnaik, G., R.H. Guirguis, J.P. Boris and E.S. Oran - A Barely Implicit Correction for Flux-Corrected Transport; J. Comp. Phys. (1988).
40. Patnaik, G., K. Kailasanath, K.J. Laskey and E.S. Oran - Detailed Numerical Simulations of Cellular Flames; Proc. 22nd Symposium (Int.) on Combustion, The Combustion Institute, Pittsburgh, (1989).
41. Peraire, J., J. Peiro and K. Morgan - A Three-Dimensional Finite Element Multigrid Solver for the Euler Equations; AIAA-92-0449 (1992).
42. Rice, D.L., M.E. Giltrud, J.D. Baum, H. Luo and E. Mestreau - Experimental and Numerical Investigation of Shock Diffraction About Blast Walls; Proc. 16th Int. Symp. Military Aspects of Blast and Shocks, Keble College, Oxford, UK, September 10-15 (2000).
43. Riemann, G.F.B. - Über die Fortpflanzung ebener Luftwellen von Endlicher Schwingungweite; Abhandlungen der Königlichen Gesellschaft der Wissenschaften zu Göttingen, **8** (1860).
44. Roe, P.L. - Approximate Riemann Solvers, Parameter Vectors and Difference Schemes; J. Comp. Phys. **43**, 357-372 (1981).
45. Sivier, S., E. Loth, J.D. Baum and R. Löhner - Vorticity Produced by Shock Wave Diffraction; Shock Waves **2**, 31-41 (1992).
46. Scannapieco, A.J. and S. L. Ossakow - Nonlinear Equatorial Spread F; Geophys. Res. Lett. **3**, 451 (1976).
47. Sod, G. - A Survey of Several Finite Difference Methods for Systems of Nonlinear Hyperbolic Conservation Laws; J. Comp. Phys. **27**, 1-31 (1978).
48. Sweby, P.K. - High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws; SIAM J. Num. Anal. **21**, 995-1011 (1984).
49. Weatherill, N.P., O. Hassan, M.J. Marchant and D.L. Marcum - Adaptive Inviscid Flow Solutions for Aerospace Geometries on Efficiently Generated Unstructured Tetrahedral Meshes; AIAA-93-3390 (1993).
50. Whitaker, D.L., B. Grossman and R. Löhner - Two-Dimensional Euler Computations on a Triangular Mesh Using an Upwind, Finite-Volume Scheme; AIAA-89-0365 (1989).
51. Woodward, P. and P. Colella - The Numerical Simulation of Two-Dimensional Fluid Flow with Strong Shocks; J. Comp. Phys. **54**, 115-173 (1984).
52. Zalesak, S.T. - Fully Multidimensional Flux-Corrected Transport Algorithm for Fluids; J. Comp. Phys. **31**, 335-362 (1979).
53. Zalesak, S.T. and R. Löhner - Minimizing Numerical Dissipation in Modern Shock Capturing Schemes; pp. 1205-1211 in *Proc. 7th Int. Conf. Finite Elements in Flow Problems* (T.J. Chung and G. Karr eds.), Huntsville, AL (1989).
54. Zhmakin, A.I. and A.A. Fursenko - A Class of Monotonic Shock-Capturing Difference Schemes; NRL Memo. Rep. 4567, (1981).

Algebraic Flux Correction I. Scalar Conservation Laws

Dmitri Kuzmin[1] and Matthias Möller[2]

[1] Institute of Applied Mathematics (LS III), University of Dortmund
Vogelpothsweg 87, D-44227, Dortmund, Germany
kuzmin@math.uni-dortmund.de
[2] matthias.moeller@math.uni-dortmund.de

Summary. An algebraic approach to the design of multidimensional high-resolution schemes is introduced and elucidated in the finite element context. A centered space discretization of unstable convective terms is rendered local extremum diminishing by a conservative elimination of negative off-diagonal coefficients from the discrete transport operator. This modification leads to an upwind-biased low-order scheme which is nonoscillatory but overly diffusive. In order to reduce the incurred error, a limited amount of compensating antidiffusion is added in regions where the solution is sufficiently smooth. Two closely related flux correction strategies are presented. The first one is based on a multidimensional generalization of total variation diminishing (TVD) schemes, whereas the second one represents an extension of the FEM-FCT paradigm to implicit time-stepping. Nonlinear algebraic systems are solved by an iterative defect correction scheme preconditioned by the low-order evolution operator which enjoys the M-matrix property. The diffusive and antidiffusive terms are represented as a sum of antisymmetric internodal fluxes which are constructed edge-by-edge and inserted into the global defect vector. The new methodology is applied to scalar transport equations discretized in space by the Galerkin method. Its performance is illustrated by numerical examples for 2D benchmark problems.

1 Introduction

Over the past three decades that elapsed since the birth of the FCT algorithm, numerous clones and alternative high-resolution schemes based on flux/slope limiters have been proposed in the literature. Nevertheless, the development of reliable discretization techniques for convection-dominated flows remains one of the main challenges in Computational Fluid Dynamics. As a matter of fact, a serious disadvantage of many existing numerical schemes is the lack of generality. Most of them are only suitable for structured or simplex meshes, specific space discretization (finite differences/volumes, discontinuous or linear finite element approximations) and/or explicit time-stepping.

In particular, the use of high-resolution finite element schemes is still rather uncommon in spite of their enormous potential demonstrated by Löhner and his collaborators [33],[35],[36]. The failure of the FEM to be as successful in CFD as in structural mechanics is largely due to the fact that many limiting techniques are essentially one-dimensional and do not carry over to unstructured meshes. In a series of recent publications, we derived a new family of implicit FEM-FCT schemes using a mass-conserving modification of matrices that result from the standard Galerkin discretization [22],[23],[25],[41]. In fact, this approach to the design of nonoscillatory positivity-preserving schemes can be characterized as *Algebraic Flux Correction* which is applicable to discrete transport operators of any origin. All the necessary information is provided by the magnitude, sign, and position of nonzero matrix coefficients.

Furthermore, the AFC methodology forms the basis for a fully multidimensional generalization of Harten's *total variation diminishing* schemes [27]. A node-oriented flux limiter of TVD type is constructed so as to control the ratio of upstream and downstream edge contributions which are associated with the positive and negative off-diagonal coefficients of the high-order transport operator, respectively [27]. This limiting strategy resembles Zalesak's FCT algorithm [54] but its derivation is based on different premises. In addition, the proposed FEM-TVD schemes are upwind-biased, i.e., the raw antidiffusive fluxes are multiplied by the correction factor for the upwind node rather than by the minimum of those for both nodes.

In contrast to flux limiters of TVD type, flux-corrected transport methods operate at the fully discrete level. As a result, the correction factors estimated by the limiter depend on the time step, so that it is impossible to reap the benefits of the fully implicit time-stepping without sacrificing some accuracy. In order to prevent an implicit FEM-FCT discretization from becoming increasingly diffusive at large time steps, flux correction can be carried out in an iterative fashion so as to 'recycle' the rejected antidiffusion step-by-step [28]. In this chapter, we derive and compare algebraic TVD and FCT schemes for scalar convection-diffusion problems. Their performance is evaluated numerically. An extension of both AFC techniques to the Euler and Navier-Stokes equations of fluid dynamics is presented in the next two chapters.

2 Finite Element Discretization

Let us introduce the principles of algebraic flux correction in the finite element framework and keep in mind that they are also applicable to finite volume and finite difference discretizations. Consider the time-dependent continuity equation which represents a mass conservation law for a scalar quantity u

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) = 0 \qquad \text{in } \Omega, \tag{1}$$

where $\mathbf{v} = \mathbf{v}(\mathbf{x}, t)$ is a nonuniform velocity field, which is assumed to be known analytically or computed numerically from a momentum equation solved in

a parallel way. The initial data are given by $u(\mathbf{x},0) = u_0(\mathbf{x})$, and boundary conditions are to be prescribed only at the inlet $\Gamma_{\text{in}} = \{\mathbf{x} \in \Gamma : \mathbf{v} \cdot \mathbf{n} < 0\}$, where $\mathbf{n}$ denotes the unit outward normal to the boundary Γ.

The weak form of equation (1) is derived by integrating the weighted residual over the domain Ω and setting the result equal to zero

$$\int_\Omega w \left[\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) \right] \mathrm{d}\mathbf{x} = 0, \qquad \forall w. \tag{2}$$

If the boundary conditions are specified in terms of fluxes rather than actual values of u at the inlet, it is worthwhile to integrate the convective term by parts and substitute the incoming fluxes into the resulting surface integral.

A common practice in finite element methods for conservation laws is to interpolate the convective fluxes in the same way as the numerical solution

$$u_h = \sum_j u_j \varphi_j, \qquad (\mathbf{v}u)_h = \sum_j (\mathbf{v}_j u_j)\, \varphi_j, \tag{3}$$

where φ_j denote the basis functions spanning the finite-dimensional subspace. This kind of approximation was promoted by Fletcher [13] who called it the *group finite element formulation*. It was found to provide a very efficient treatment of nonlinear convective terms and even lead to a small gain of accuracy for the 2D Burgers equation discretized on a uniform grid [13]. In fact, it is even possible to use mixed interpolations. For instance, a bilinear approximation of u could be combined with its nonconforming counterpart [46] for the convective fluxes which call for the use of edge-oriented degrees of freedom.

The substitution of (3) into (2) yields the following semi-discrete problem

$$\sum_j \left[\int_\Omega \varphi_i \varphi_j \, \mathrm{d}\mathbf{x} \right] \frac{\mathrm{d}u_j}{\mathrm{d}t} + \sum_j \left[\int_\Omega \varphi_i \mathbf{v}_j \cdot \nabla \varphi_j \, \mathrm{d}\mathbf{x} \right] u_j = 0. \tag{4}$$

This gives a system of ordinary differential equations for the nodal values of the approximate solution which can be written compactly in matrix form

$$M_C \frac{\mathrm{d}u}{\mathrm{d}t} = Ku, \tag{5}$$

where $M_C = \{m_{ij}\}$ denotes the consistent mass matrix and $K = \{k_{ij}\}$ stands for the discrete transport operator. The matrix entries are given by

$$m_{ij} = \int_\Omega \varphi_i \varphi_j \, \mathrm{d}\mathbf{x}, \qquad k_{ij} = -\mathbf{v}_j \cdot \mathbf{c}_{ij}, \qquad \mathbf{c}_{ij} = \int_\Omega \varphi_i \nabla \varphi_j \, \mathrm{d}\mathbf{x}. \tag{6}$$

For fixed meshes, the coefficients m_{ij} and $\mathbf{c}_{ij}$ remain unchanged throughout the simulation and, consequently, need to be evaluated just once, during the initialization step. This enables us to update the matrix K in a very efficient way by computing its entries k_{ij} from formula (6) without resorting to costly numerical integration. The auxiliary coefficients $\mathbf{c}_{ij}$ correspond to the discretized space derivatives and have zero row sums, i. e., $\sum_j \mathbf{c}_{ij} = 0$ as long as the sum of the basis functions φ_j is equal to one at every point.

3 Conservative Flux Decomposition

It is common knowledge that the Galerkin FEM is globally conservative [15]. Indeed, summing equations (4) over i and taking into account that the basis functions sum to unity, one recovers the integral form of the conservation law. Therefore, the total mass of u in Ω may only change due to the boundary fluxes. At the same time, the finite element discretization of convective terms does not admit a natural decomposition into a sum of numerical fluxes from one node into another. Since most high-resolution schemes operate with such fluxes, their extension to finite elements proved to be a difficult task. Peraire et al. [45] demonstrated that a conservative flux decomposition is feasible for P_1 finite elements (piecewise-linear approximation on a simplex mesh). A similar technique was proposed by Barth [3],[4] who investigated the relationship between finite element and finite volume discretizations. The transition to an edge-based data structure reportedly offers a number of significant advantages as compared to the conventional element-based formulation. Moreover, it paves the way for a straightforward extension of many popular high-resolution schemes (including TVD) to unstructured meshes [33],[37],[42].

In [25] we developed a flux decomposition technique which is applicable to general finite element approximations on arbitrary meshes including quadrilateral and hexahedral ones. Integration by parts in the weak formulation (2) and the fact that the coefficients $\mathbf{c}_{ij}$ have zero row sums make it possible to decompose the contribution of convective terms to interior nodes into a sum of antisymmetric internodal fluxes (see appendix to this chapter)

$$(Ku)_i = -\sum_{j\neq i} g_{ij}, \qquad \text{where} \quad g_{ij} = (\mathbf{v}_i \cdot \mathbf{c}_{ij})u_i - (\mathbf{v}_j \cdot \mathbf{c}_{ji})u_j. \tag{7}$$

A promising approach to the derivation of nonoscillatory finite element methods consists in replacing the centered Galerkin flux g_{ij} by another consistent numerical flux [37]. On the other hand, it is often desirable to use an already existing finite element code based on conventional data structures. Therefore, we adopt a different strategy in the present chapter.

Due to the fact that the Galerkin method is conservative, it suffices to guarantee that all subsequent matrix manipulations to be performed at the discrete level do not violate this property. To this end, we introduce the concept of *discrete diffusion operators* [23]. They are defined as symmetric matrices

$$D = \{d_{ij}\} \qquad \text{such that} \quad d_{ij} = d_{ji} \tag{8}$$

which have zero row and column sums

$$\sum_i d_{ij} = \sum_j d_{ij} = 0. \tag{9}$$

We remark that the matrix D is typically sparse and its nonzero off-diagonal entries d_{ij} may be positive (diffusion) or negative (antidiffusion).

In the finite element framework, some well-known representatives of this important class of matrices are as follows:

- The *discrete Laplacian operator* which results from the discretization of second derivatives after integration by parts in the weak formulation

$$d_{ij} = \int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j \, \mathrm{d}\mathbf{x}.$$

- The *streamline diffusion operator* which provides a stabilization of convective terms by artificial diffusion in the streamline direction

$$d_{ij} = \int_\Omega \mathbf{v} \cdot \nabla\varphi_i \; \mathbf{v} \cdot \nabla\varphi_j \, \mathrm{d}\mathbf{x}.$$

- The *mass diffusion operator* which is given by the difference between the consistent mass matrix and its lumped counterpart (see below)

$$d_{ij} = \int_\Omega \varphi_i(\varphi_j - \delta_{ij}) \, \mathrm{d}\mathbf{x}.$$

Here δ_{ij} is the Kronecker delta which equals 1 if $i = j$ and 0 otherwise.

A discrete diffusion operator D applied to the vector of nodal values u yields

$$(Du)_i = \sum_j d_{ij} u_j = \sum_{j \neq i} d_{ij}(u_j - u_i) \tag{10}$$

due to the zero row sum property. Hence, the contribution of diffusive terms to node i can be decomposed into a sum of numerical fluxes:

$$(Du)_i = \sum_{j \neq i} f_{ij}, \qquad \text{where} \quad f_{ij} = d_{ij}(u_j - u_i). \tag{11}$$

The flux f_{ij} from node j into node i is proportional to the difference between the nodal values, so it leads to a steepening or flattening of solution profiles depending on the sign of the coefficient d_{ij}. Furthermore, the symmetry of the matrix D implies that $f_{ji} = -f_{ij}$ so that there is no net loss or gain of mass. The amount received by node i is subtracted from node j and vice versa.

The antisymmetric diffusive fluxes can be associated with edges of the graph which represents the sparsity pattern of the global stiffness matrix. For linear finite elements, their number equals the number of actual mesh edges, whereas multilinear and high-order FEM approximations allow for interactions of all nodes sharing the same element. As we are about to see, artificial diffusion operators satisfying the above conditions constitute a very useful tool for the design of multidimensional high-resolution schemes.

4 Design Criteria

First of all, we introduce the algebraic constraints which should be imposed on the discrete operators to prevent the formation of spurious undershoots and overshoots in the vicinity of steep gradients. Assume that the semi-discretized transport equation can be cast in the generic form

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_j \sigma_{ij} u_j, \qquad \text{where} \quad \sigma_{ii} = -\sum_{j \neq i} \sigma_{ij}. \tag{12}$$

In particular, this is feasible for our nodal ODE system (5) if M_C is replaced by its diagonal counterpart M_L resulting from the row-sum mass lumping

$$\sigma_{ij} = \frac{k_{ij}}{m_i}, \qquad \text{where} \qquad m_i = \sum_j m_{ij}, \quad M_L = \mathrm{diag}\{m_i\}$$

and the velocity field $\mathbf{v}$ is discretely divergence-free in the sense that

$$(\nabla \cdot \mathbf{v})_i = \frac{1}{m_i} \sum_j \mathbf{v}_j \cdot \mathbf{c}_{ij} = -\sum_j \sigma_{ij} = 0. \tag{13}$$

This approximation corresponds to a recovery of continuous nodal gradients by means of a lumped-mass L_2-projection. For 'compressible' flows, the sum of the coefficients σ_{ij} is nonvanishing. However, this makes no difference for the algebraic flux correction algorithms to be derived below.

Algebraic Constraint I (semi-discrete level)

If the coefficients of the numerical scheme do have zero row sums, then the right-hand side of (12) can be represented in terms of the off-diagonal ones

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} \sigma_{ij}(u_j - u_i). \tag{14}$$

It was shown by Jameson [19], [20], [21] that negative coefficients in the above expression are the 'villains' responsible for the birth and growth of nonphysical oscillations. Indeed, if $\sigma_{ij} \geq 0$, $\forall j \neq i$ then the spatial discretization proves stable in the L_∞-norm due to the fact that

- maxima do not increase: $u_i = \max_j u_j \quad \Rightarrow \quad u_j - u_i \leq 0 \quad \Rightarrow \quad \frac{\mathrm{d}u_i}{\mathrm{d}t} \leq 0,$
- minima do not decrease: $u_i = \min_j u_j \quad \Rightarrow \quad u_j - u_i \geq 0 \quad \Rightarrow \quad \frac{\mathrm{d}u_i}{\mathrm{d}t} \geq 0.$

As a rule, the coefficient matrices are sparse, so that $\sigma_{ij} = 0$ unless i and j are adjacent nodes. Arguing as above, one can show that a *local* maximum cannot increase, and a *local* minimum cannot decrease. Therefore, semi-discrete

schemes of this type are *local extremum diminishing* (LED). For three-point finite difference methods, the LED constraint reduces to Harten's TVD conditions [16]. If the homogeneous Dirichlet boundary conditions are prescribed at both endpoints, the total variation of the (piecewise-linear) approximate solution can be expressed as follows [19]

$$TV(u_h) := \sum_i |u_{i+1} - u_i| = 2\left(\sum \max u - \sum \min u\right) \tag{15}$$

and is obviously nonincreasing as long as the local maxima and minima do not grow. Therefore, one-dimensional LED schemes are necessarily total variation diminishing. At the same time, the positivity of matrix coefficients is easy to verify for arbitrary discretizations on unstructured meshes so that Jameson's LED criterion provides a very handy generalization of the TVD concepts.

Algebraic Constraint II (fully discrete level)

After the time discretization, an additional condition may need to be imposed in order to make sure that the solution values remain nonnegative if this should be the case for physical reasons. In general, a fully discrete scheme is *positivity-preserving* if it can be represented in the form

$$Au^{n+1} = Bu^n, \tag{16}$$

where $B = \{b_{ij}\}$ has no negative entries and $A = \{a_{ij}\}$ is a so-called *M-matrix* defined as a nonsingular discrete operator such that $a_{ij} \leq 0$ for $j \neq i$ and all the coefficients of its inverse are nonnegative. These properties imply that the positivity of the old solution u^n carries over to $u^{n+1} = A^{-1}Bu^n$. Here and below the superscript n denotes the time level.

As a useful byproduct, our algebraic positivity criterion yields a readily computable upper bound for admissible values of the time step $\Delta t = t^{n+1} - t^n$. In particular, the local extremum diminishing ODE system (14) discretized in time by the standard θ-scheme reads

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \theta \sum_{j \neq i} \sigma_{ij}^{n+1}(u_j^{n+1} - u_i^{n+1}) + (1-\theta) \sum_{j \neq i} \sigma_{ij}^n (u_j^n - u_i^n). \tag{17}$$

It is unconditionally positivity-preserving for $\theta = 1$ (backward Euler method) and subject to the following CFL-like condition otherwise [23],[25]

$$1 + \Delta t(1-\theta) \min_i \sigma_{ii}^n \geq 0 \qquad \text{for} \quad 0 \leq \theta < 1. \tag{18}$$

Note that this estimate is based solely on the magnitude of the diagonal coefficients σ_{ii}^n, which makes it a handy tool for adaptive time step control.

5 Algebraic Flux Correction of TVD Type

The basic idea underlying algebraic flux correction is rather simple and can be traced back to the concepts of flux-corrected transport. Roughly speaking, the governing equation is discretized in space by an arbitrary linear high-order method (e.g. central differences or Galerkin FEM) and the resulting matrices are modified *a posteriori* so as to enforce the constraints I and II imposed above. The flow chart of required algebraic manipulations is sketched in Fig. 1. The time step Δt should be chosen so as to satisfy condition (18).

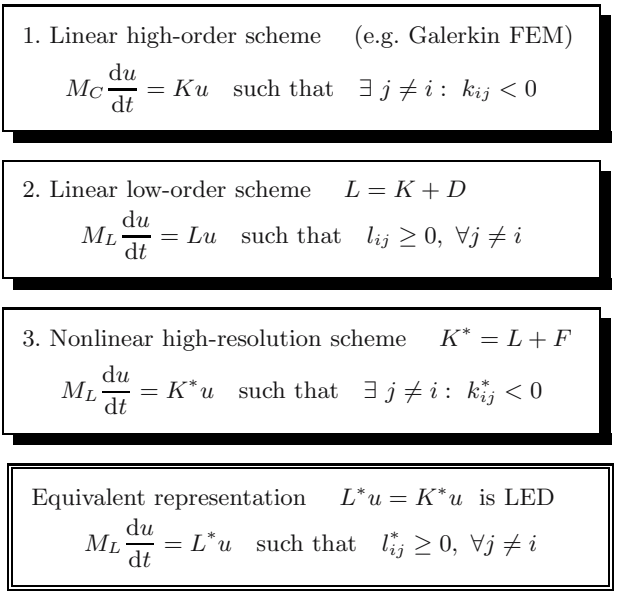


Fig. 1. Roadmap of AFC-TVD manipulations

First, we perform mass lumping and transform the high-order operator K into its nonoscillatory low-order counterpart L by adding a discrete diffusion operator D designed so as to get rid of all negative off-diagonal coefficients. In the next step, excessive artificial diffusion is removed. This is accomplished by applying a limited amount of compensating antidiffusion F which depends on the local solution behavior and improves the accuracy in smooth regions. Both diffusive and antidiffusive terms admit a conservative flux decomposition so that the proposed modifications do not affect the total mass.

It is worth mentioning that the final operator K^* does have some negative off-diagonal coefficients. Nevertheless, the resulting discretization proves local extremum diminishing if (for a given solution vector u) there exists a matrix L^* such that all off-diagonal entries l^*_{ij} are nonnegative and $L^* u = K^* u$. In

the remainder of this section we will dwell on the design of discrete diffusion/antidiffusion operators and introduce flux limiters of TVD type which guarantee the existence of L^* without constructing it explicitly. In much the same way, we will derive a family of implicit FCT schemes using criterion (16) to render the underlying high-order method positivity-preserving.

5.1 Discrete Upwinding

For finite difference and finite volume discretizations, the first-order accurate upwind method yields an operator L which corresponds to the least diffusive linear LED scheme. Up to now, it has been largely unclear how to construct such an optimal low-order discretization in the finite element framework. Streamline-diffusion methods like SUPG are stable but not monotonicity-preserving, whereas other upwind-biased finite element schemes resort to a finite volume approximation of convective terms [1],[2],[52]. At the same time, the LED constraint can be enforced by elimination of negative off-diagonal coefficients from the discrete transport operator. Interestingly enough, this algebraic approach to the design of 'monotone' low-order methods reduces to standard upwinding for the one-dimensional convection equation [22],[23].

As a starting point, we consider a linear high-order discretization, e. g., our semi-discrete problem (5) for the Galerkin method. After mass lumping, each nodal value u_i satisfies an ordinary differential equation of the form

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} k_{ij}(u_j - u_i) + \delta_i u_i, \qquad \text{where} \quad \delta_i = \sum_j k_{ij}. \tag{19}$$

The first term in the right-hand side is associated with the 'incompressible' part of the discrete transport operator K since $\delta_i u_i$ is an approximation of $-u\nabla \cdot \mathbf{v}$ (see above) which vanishes for divergence-free velocity fields and is responsible for a physical growth of local extrema otherwise. For the concomitant low-order scheme to be local extremum diminishing, all off-diagonal coefficients of the linear operator $L = K + D$ must be nonnegative. Hence, the optimal diffusion coefficients are given by

$$d_{ii} = -\sum_{j \neq i} d_{ij}, \qquad d_{ij} = \max\{0, -k_{ij}, -k_{ji}\} = d_{ji}. \tag{20}$$

By construction, $D = \{d_{ij}\}$ is a discrete diffusion operator. It follows that the difference between the resulting scheme and the original one can be represented as a sum of antisymmetric diffusive fluxes $f_{ij}^d = d_{ij}(u_j - u_i)$ between adjacent nodes whose basis functions have overlapping supports. Recall that this is sufficient to guarantee mass conservation at the algebraic level. The above manipulations lead to the desired semi-discrete scheme of low order

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = Lu \quad \text{such that} \quad l_{ij} \geq 0, \ \forall j \neq i. \tag{21}$$

In practice, the elimination of negative off-diagonal entries is performed edge-by-edge without assembling the global matrix D. After the initialization $L := K$, we examine each pair of nonzero off-diagonal coefficients l_{ij} and l_{ji}. If the smaller one is negative, it is set equal to zero and three other entries are modified so as to restore row/column sums:

$$\begin{aligned} l_{ii} &:= l_{ii} - d_{ij}, \quad & l_{ij} &:= l_{ij} + d_{ij}, \\ l_{ji} &:= l_{ji} + d_{ij}, \quad & l_{jj} &:= l_{jj} - d_{ij}. \end{aligned} \tag{22}$$

Without loss of generality, we orient the edges of the sparsity graph so that $l_{ji} \geq l_{ij} = \max\{0, k_{ij}\}$ for the edge $\overrightarrow{ij}$. This orientation convention implies that node i is located 'upwind' and corresponds to the row number of the eliminated negative entry (if any). Furthermore, the nodes can be renumbered so as to transform L into an upper or lower triangular matrix and to design very efficient solvers/smoothers/preconditioners for the resulting linear system.

The 'postprocessing' technique described in this section will be referred to as *discrete upwinding*. Note that the LED constraint is imposed only on the incompressible part of the transport operator. The 'reactive' term $\delta_i u_i$ is not affected by artificial diffusion since $\sum_j l_{ij} = \sum_j (k_{ij} + d_{ij}) = \sum_j k_{ij}$ due to the zero row sum property of D. If the governing equation contains sources and sinks, they may need to be linearized as proposed by Patankar [44] and explained in [25],[41]. Furthermore, physical diffusion can be built into the matrix either before or after discrete upwinding. In the former case, it is automatically detected and the amount of artificial diffusion is reduced accordingly. In our experience, the TVD flux limiters to be presented below should be applied to the convective operator alone. Therefore, it is advisable to incorporate the contribution of physical diffusion into L rather than K.

Example. To elucidate the ins and outs of discrete upwinding in a rather simple setting, consider the one-dimensional counterpart of equation (1)

$$\frac{\partial u}{\partial t} + v \frac{\partial u}{\partial x} = 0, \tag{23}$$

where the velocity v is assumed to be constant and positive. The computational domain $\Omega = (a, b)$ is defined by its two endpoints, and an essential boundary condition of the Dirichlet type is prescribed at the inlet $x = a$.

This hyperbolic equation is discretized in space by the lumped-mass Galerkin method using a piecewise-linear approximation on a uniform mesh of size Δx. The corresponding 2×2 *element matrices* are given by

$$\hat{M}_L = \frac{\Delta x}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \hat{K} = \frac{v}{2} \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}. \tag{24}$$

After the global matrix assembly, the central difference discretization of the convective term is recovered at interior nodes:

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -v \, \frac{u_{i+1} - u_{i-1}}{2 \, \Delta x}. \tag{25}$$

The negative coefficient in the upper-right corner of $\hat{K}$ violates the LED criterion and should be eliminated. To this end, the artificial diffusion operator $\hat{D}$ is designed to be a symmetric matrix with zero row and column sums such that the entry $\hat{l}_{12}$ of the low-order operator $\hat{L} = \hat{K} + \hat{D}$ is equal to zero

$$\hat{D} = \frac{v}{2}\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \quad \Rightarrow \quad \hat{L} = v\begin{bmatrix} 0 & 0 \\ 1 & -1 \end{bmatrix}. \tag{26}$$

The resulting local extremum diminishing scheme is the least diffusive among linear ones. Remarkably, it is equivalent to the standard upwind method

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -v\,\frac{u_i - u_{i-1}}{\Delta x} \tag{27}$$

and proves positivity-preserving under condition (18) which reduces to

$$v\frac{\Delta t}{\Delta x} \leq \frac{1}{1-\theta}, \qquad 0 \leq \theta < 1. \tag{28}$$

Recall that this restriction does not apply to the backward Euler time-stepping which corresponds to 'upwinding in time' and is only first-order accurate.

5.2 Classical TVD Methodology

Let us stay in one dimension for a while in order to review the basic concepts and principles behind Harten's TVD schemes [16],[17]. It is well known that the numerical diffusion inherent to the upwind method is proportional to Δx so that even this 'optimal' LED scheme is only first-order accurate. The quality of the results can be dramatically improved by applying a nonlinear antidiffusive correction $\hat{F}(u)$ to the monotone operator $\hat{L}$. In essence, the final transport operator $\hat{K}^*(u) = \hat{L} + \hat{F}(u)$ is constructed by removing a certain fraction of the artificial diffusion which was added to the high-order discretization to suppress spurious oscillations. Specifically, we consider

$$\hat{F}(u) = -\Phi(r_i)\hat{D} \quad \Rightarrow \quad \hat{K}^*(u) = \hat{K} + [1 - \Phi(r_i)]\hat{D}, \tag{29}$$

where the flux limiter Φ determines the magnitude of admissible antidiffusion in an adaptive fashion. As a rule of thumb, the blending factor $\Phi(r_i)$ should be equal to zero in the vicinity of steep gradients and approach (or even exceed) unity in regions where the solution is sufficiently smooth.

The smoothness sensor r_i is typically defined to be the ratio of consecutive gradients which is to be evaluated at the upwind node

$$r_i = \frac{u_i - u_{i-1}}{u_{i+1} - u_i}. \tag{30}$$

Obviously, this quantity is negative at a local extremum (see Fig. 2), relatively small for smooth data and large if the solution tends to change abruptly.

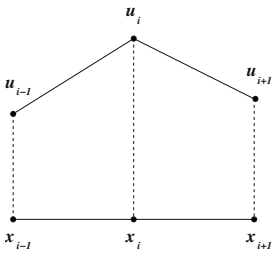


Fig. 2. Three-point stencil in one dimension

The global matrix assembly yields a conservative finite difference scheme

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} + \frac{f_{i+1/2} - f_{i-1/2}}{\Delta x} = 0, \tag{31}$$

where the numerical fluxes $f_{i\pm1/2}$ correspond to a nonlinear combination of first- and second-order approximations to the continuous flux function vu

$$f_{i+1/2} = vu_i + \frac{v}{2}\Phi(r_i)(u_{i+1} - u_i). \tag{32}$$

The resulting semi-discretized equations can be rewritten in the form

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = c_{i-1/2}(u_{i-1} - u_i) + c_{i+1/2}(u_{i+1} - u_i), \tag{33}$$

where the pair of (possibly nonlinear) coefficients $c_{i\pm1/2}$ can be defined in many different ways. In particular, the following representation is feasible

$$c_{i-1/2} = \frac{v}{2\Delta x}\left[2 + \frac{\Phi(r_i)}{r_i} - \Phi(r_{i-1})\right], \qquad c_{i+1/2} = 0. \tag{34}$$

To satisfy the LED criterion and meet the requirements of Harten's theorem [16], the expression in the brackets must be nonnegative. A variety of flux limiters have been proposed in the literature to enforce this condition. It was shown by Jameson [19] that most of them can be interpreted and implemented as *limited average operators* $\mathcal{L}(a,b)$ such that $\Phi(r) = \mathcal{L}(1,r)$. These two-parameter functions are characterized by a number of common properties:

P1. $\mathcal{L}(a,b) = \mathcal{L}(b,a)$.

P2. $\mathcal{L}(ca,cb) = c\mathcal{L}(a,b)$.

P3. $\mathcal{L}(a,a) = a$.

P4. $\mathcal{L}(a,b) = 0 \quad \text{if} \quad ab \le 0$.

In particular, the last one ensures that $\Phi(r) = 0$ if $r \le 0$. Thus, the accuracy of a TVD discretization inevitably degrades to the first order at local extrema.

Another important implication is the symmetry of the flux limiter

$$\Phi(r) = \mathcal{L}(1, r) = r\mathcal{L}(1/r, 1) = r\Phi(1/r) \tag{35}$$

which follows from the properties (P1) and (P2). By virtue of these relations, the *antidiffusive flux* from node $i+1$ into node i is proportional to the limited average of the slopes and has the same effect as a diffusive flux from node $i-1$ provided that the coefficient $\Phi(1/r_i)$ is greater than zero:

$$\Phi(r_i)(u_{i+1} - u_i) = \mathcal{L}(u_{i+1} - u_i, u_i - u_{i-1}) = \Phi(1/r_i)(u_i - u_{i-1}). \tag{36}$$

In other words, the task of the limiter is to guarantee that the antidiffusive flux can be expressed as a 'diffusive' one for which the corresponding off-diagonal coefficient is nonnegative. This is sufficient to satisfy the LED constraint.

Note that the averaging operators $\mathcal{L}$ are applied not to the slope ratio r_i but to its nominator and denominator so that division by zero is ruled out. Some of the standard TVD limiters written in this form are as follows [19]

minmod:	$\mathcal{L}(a,b) = \mathcal{S}(a,b)\ \min\{\lvert a\rvert, \lvert b\rvert\}$
Van Leer:	$\mathcal{L}(a,b) = \mathcal{S}(a,b)\ \frac{2\lvert a\rvert\lvert b\rvert}{\lvert a\rvert+\lvert b\rvert}$
MC:	$\mathcal{L}(a,b) = \mathcal{S}(a,b)\ \min\left\{\frac{\lvert a+b\rvert}{2}, 2\lvert a\rvert, 2\lvert b\rvert\right\}$
superbee:	$\mathcal{L}(a,b) = \mathcal{S}(a,b)\ \max\{\min\{2\lvert a\rvert, \lvert b\rvert\}, \min\{\lvert a\rvert, 2\lvert b\rvert\}\}$

$$\text{where} \quad \mathcal{S}(a,b) = \frac{\operatorname{sign}(a) + \operatorname{sign}(b)}{2} = \begin{cases} 1 & \text{if } a > 0 \ \wedge\ b > 0, \\ -1 & \text{if } a < 0 \ \wedge\ b < 0, \\ 0 & \text{otherwise.} \end{cases}$$

The associated one-parameter limiter functions Φ yield correction factors lying in the range $[0, 2]$, whereby the integer values $0, 1, 2$ correspond to the upwind, central, and downwind approximation of the convective term, respectively.

5.3 Generalized TVD Formulation

Now let us proceed to algebraic flux correction in the multidimensional case. Recall that we discretized the continuity equation in space by the Galerkin method, performed mass lumping and transformed the high-order operator K into a low-order operator L by elimination of negative off-diagonal coefficients. This modification inevitably leads to a global loss of accuracy. According to the Godunov theorem [14], a nonoscillatory high-resolution scheme must be nonlinear even for a linear partial differential equation. On the other hand, the majority of real-life CFD applications are governed by *nonlinear* conservation laws to begin with, so that the computational overhead due to an iterative adjustment of implicit artificial diffusion is not very significant.

Following the algorithm presented for the finite difference TVD schemes, we employ an antidiffusive correction $F(u)$ to reduce the error incurred by discrete upwinding in smooth regions. The modified transport operator $K^*(u)$ for a generalized TVD method exhibits the following structure (cf. Fig. 1)

$$K^*(u) = L + F(u) = K + D + F(u), \tag{37}$$

where both D and $F(u)$ possess the properties of discrete diffusion operators.

In a practical implementation, the contribution of the nonlinear antidiffusive terms to the right-hand side of the final semi-discrete scheme

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = K^* u \tag{38}$$

is assembled edge-by-edge from internodal fluxes. Specifically, we have

$$(Fu)_i = \sum_{j \neq i} f_{ij}^a \qquad \text{such that} \quad f_{ji}^a = -f_{ij}^a, \tag{39}$$

where the antidiffusive flux f_{ij}^a from node j into its **upwind** (in the sense of our orientation convention $l_{ji} \geq l_{ij}$) neighbor i depends on the diffusion coefficient d_{ij} for discrete upwinding and on the entry $l_{ji} = \max\{k_{ji}, k_{ji} - k_{ij}\}$ of the low-order transport operator:

$$f_{ij}^a := \min\{\Phi(r_i) d_{ij}, l_{ji}\}(u_i - u_j). \tag{40}$$

Furthermore, Φ is a standard one-parameter limiter applied to a suitable smoothness indicator r_i (to be specified below). By definition, the downwind node j receives the flux $f_{ji}^a := -f_{ij}^a$ of the same magnitude but with the opposite sign so that mass conservation is guaranteed.

Let us derive a *sufficient* condition for the FEM-TVD scheme (38) to be local extremum diminishing. If $\Phi(r_i) = 0$ or $d_{ij} = 0$, the antidiffusive flux f_{ij}^a vanishes and does not pose any hazard. Therefore, we restrict ourselves to the nontrivial case $f_{ij}^a \neq 0$ which implies that both $\Phi(r_i)$ and d_{ij} are strictly positive. Our objective is to prove the existence of a LED operator L^* which is equivalent to K^* for the given solution u (see the last box in Fig. 1). Clearly, the sensor r_i cannot be chosen arbitrarily. The symmetry property (35) of the limiter Φ makes it possible to represent the antidiffusive flux in the form

$$f_{ij}^a = \Phi(r_i) a_{ij}(u_i - u_j) = \Phi(1/r_i) a_{ij} \Delta u_{ij}, \tag{41}$$

where the (positive) *antidiffusion coefficient* a_{ij} and the *upwind difference* Δu_{ij} are defined as follows

$$a_{ij} := \min\{d_{ij}, l_{ji}/\Phi(r_i)\}, \qquad \Delta u_{ij} := r_i(u_i - u_j). \tag{42}$$

For the numerical solution to be nonoscillatory, the antidiffusive fluxes must behave as diffusive ones, cf. equation (36). The assumption $d_{ij} > 0$ implies

that $k_{ij} < 0$ and $l_{ij} = 0$ for the edge $\overrightarrow{ij}$ which links an upwind node i and a downwind node j. Therefore, the edge contributions to the two components of the modified convective term K^*u in (38) can be written as

$$k^*_{ij}(u_j - u_i) = f^a_{ij}, \qquad k^*_{ji}(u_i - u_j) = l_{ji}(u_i - u_j) - f^a_{ij}. \tag{43}$$

The increment to node j is obviously of diffusive nature and satisfies the LED criterion, since the coefficient $k^*_{ji} = l_{ji} - \Phi(r_i)a_{ij}$ is nonnegative by construction (see the definition of a_{ij}). Furthermore, it follows from relation (41) that the negative off-diagonal entry $k^*_{ij} = -\Phi(r_i)a_{ij}$ of the nonlinear operator K^* is acceptable provided Δu_{ij} admits the following representation

$$\Delta u_{ij} = \sum_{k \neq i} \sigma_{ik}(u_k - u_i), \qquad \text{where} \quad \sigma_{ik} \geq 0, \quad \forall k \neq i. \tag{44}$$

In other words, the limited antidiffusive flux f^a_{ij} from node j into node i should be interpreted as a sum of diffusive fluxes contributed by other neighbors. It remains to devise a multidimensional smoothness indicator r_i and check if the corresponding upwind difference Δu_{ij} satisfies the above condition.

5.4 Slope-Limiter FEM-TVD Algorithm

For classical finite difference TVD schemes, r_i represents the slope ratio (30) at the upwind node, so that $\Delta u_{ij} = u_k - u_i$, where $k = i - 1$ is the second neighbor of node i. However, this natural definition of r_i is no longer possible in multidimensions, whereby each node interacts with more than two neighbors. A geometric approach commonly employed in the literature is to reconstruct a local one-dimensional stencil by insertion of equidistant *dummy nodes* on the continuation of each mesh edge [1],[19],[37],[38]. The difference Δu_{ij} is defined as before using the interpolated or extrapolated solution value at the dummy node k adjacent to the upwind node i. The construction of a three-point stencil for an unstructured triangular mesh is illustrated in Fig. 3.

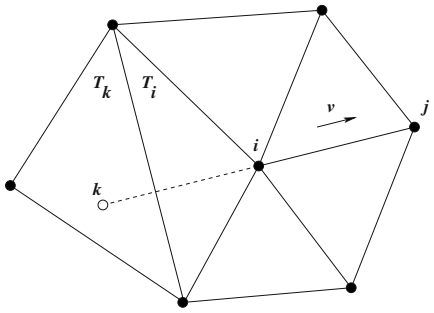


Fig. 3. Three-point stencil in two dimensions

An evaluation of various techniques for the recovery of u_k was performed by Lyra [37]. His comparative study covers the following algorithms:

- interpolation/extrapolation using the upwind triangle T_i containing node i,
- interpolation using the actual triangle T_k containing the dummy node k,
- extrapolation using a least squares reconstruction for the nodal gradient.

Numerical experiments revealed that the solutions depend strongly on the employed strategy. The first option was proved to provide the LED property but failed to produce nonoscillatory results for some aerodynamic applications. The second procedure based on the actual triangle was favored due to the enhanced robustness as compared to the use of the adjacent triangle. However, the resulting discretization is no longer local extremum diminishing and neither is the gradient reconstruction method which corresponds to

$$\Delta u_{ij} = (\mathbf{x}_i - \mathbf{x}_j) \cdot \nabla_h u_i, \tag{45}$$

where $\nabla_h u_i$ stands for a continuous approximation to the solution gradient at the upwind node i recovered by means of a consistent L_2-projection:

$$\nabla_h u_i = \frac{1}{m_i} \sum_{k \neq i} \mathbf{c}_{ik} (u_k - u_i). \tag{46}$$

The involved coefficients $\mathbf{c}_{ik}$ are defined in (6) for the standard Galerkin method. This approach is relatively simple to implement and more efficient than the linear interpolation techniques. However, Lyra [37] reported its performance to be quite poor and emphasized the need for the development of a more robust algorithm for the reconstruction of nodal gradients.

Let us explain why the above choice of the upwind difference may prove unsatisfactory. If any of the scalar products $\mathbf{c}_{ik} \cdot (\mathbf{x}_i - \mathbf{x}_j)$ is negative, then the formula for Δu_{ij} is not of the form (44), so the numerical scheme may fail to satisfy the LED criterion. To rectify this, one can employ a monotone projection operator constructed by resorting to discrete upwinding. Note that $\mathbf{c}_{ki} = -\mathbf{c}_{ik}$ for internal nodes, so that the elimination of negative off-diagonal coefficients leads to the following LED-type reconstruction procedure [26]

$$\Delta u_{ij} = \frac{2}{m_i} \sum_{k \neq i} \max\{0, \mathbf{c}_{ik} \cdot (\mathbf{x}_i - \mathbf{x}_j)\}(u_k - u_i). \tag{47}$$

In one dimension, this kind of extrapolation corresponds to using the upwind gradient and yields $\Delta u_{ij} = u_k - u_i$, where k is the upwind neighbor of i.

By virtue of (42), the upwind difference Δu_{ij} can be converted into the smoothness sensor r_i which provides an estimate of the gradient jump along the edge $\overrightarrow{ij}$. Hence, this geometric approach to the design of nonlinear LED

schemes will be referred to as the *slope-limiter* FEM-TVD algorithm. It may be equipped with any standard limiter Φ. At the same time, the numerical results are rather sensitive to the alignment of the three-point stencil and to the algorithm employed to recover the solution values at the dummy nodes. Moreover, such methods are computationally expensive and may experience severe convergence problems for steady-state applications. This shortcoming was also noticed by Lyra [37] who explained it by the lack of background dissipation and indicated that the convergence rates can be improved to some extent by 'freezing' the antidiffusive terms as the solution approaches the steady state. The main reason for the insufficient robustness seems to be the unidirectional nature of stencil reconstruction and the independent limiting of antidiffusive fluxes associated with the same upwind node.

5.5 Flux-Limiter FEM-TVD Algorithm

Let us abandon the stencil reconstruction technique and design the sensor r_i in a different way. The one-dimensional convection equation (23) discretized in space by the lumped-mass Galerkin FEM or by the central difference method can be written in the form (33) where the two coefficients are given by

$$c_{i-1/2} = \frac{v}{2\Delta x} > 0, \qquad c_{i+1/2} = -\frac{v}{2\Delta x} < 0. \tag{48}$$

Remarkably, the ratio of the upwind and downwind contributions to node i

$$r_i = \frac{c_{i-1/2}(u_{i-1} - u_i)}{c_{i+1/2}(u_{i+1} - u_i)} \tag{49}$$

reduces to the slope ratio r_i defined in (30) as long as the velocity v is constant. Moreover, this interpretation leads to a conceptually different limiting strategy which guarantees the TVD property for variable velocity fields and carries over to multidimensions. As we are about to see, the new algorithm is akin to that proposed by Zalesak [54] in the framework of flux-corrected transport methods, so we adopt his notation to reflect this relationship.

In the multidimensional case, the incompressible part of the original convective term Ku can be decomposed into a sum of edge contributions with negative coefficients and a sum of those with positive coefficients

$$P_i = \sum_{j \neq i} \min\{0, k_{ij}\}(u_j - u_i), \quad Q_i = \sum_{j \neq i} \max\{0, k_{ij}\}(u_j - u_i) \tag{50}$$

which are due to mass transfer from the downstream and upstream directions, respectively. The sum P_i is composed from the *raw antidiffusive fluxes* which offset the error incurred by elimination of negative matrix entries in the course of discrete upwinding. They are responsible for the formation of spurious wiggles and must be securely limited. At the same time, the constituents of the sum Q_i are harmless since they resemble diffusive fluxes and do satisfy

the LED criterion. Thus, it is natural to require that the net antidiffusive flux into node i be a limited average of the original increments P_i and Q_i.

Due to the property (P4) of TVD limiters, it is worthwhile to distinguish between the positive and negative edge contributions to both sums

$$P_i = P_i^+ + P_i^-, \qquad P_i^\pm = \sum_{j\neq i} \min\{0, k_{ij}\} \begin{matrix}\min\\ \max\end{matrix} \{0, u_j - u_i\}, \tag{51}$$

$$Q_i = Q_i^+ + Q_i^-, \qquad Q_i^\pm = \sum_{j\neq i} \max\{0, k_{ij}\} \begin{matrix}\max\\ \min\end{matrix} \{0, u_j - u_i\} \tag{52}$$

and limit the positive and negative antidiffusive fluxes separately. To this end, we pick a standard limiter Φ and compute the *nodal correction factors*

$$R_i^\pm = \Phi(Q_i^\pm / P_i^\pm) \tag{53}$$

which determine the percentage of $P_i^\pm$ that can be retained without violating the LED constraint for row i of the modified transport operator K^*. Clearly, $R_i^\pm$ does not need to be evaluated if the raw antidiffusion $P_i^\pm$ vanishes.

For each edge $\overrightarrow{ij}$ of the sparsity graph, the antidiffusive flux f_{ij}^a from its downwind node j into the upwind node i is constructed as follows:

$$f_{ij}^a := \begin{cases} \min\{R_i^+ d_{ij}, l_{ji}\}(u_i - u_j) & \text{if } u_i \geq u_j, \\ \min\{R_i^- d_{ij}, l_{ji}\}(u_i - u_j) & \text{if } u_i < u_j, \end{cases} \qquad f_{ji}^a := -f_{ij}^a. \tag{54}$$

Importantly, the same correction factor $R_i^\pm$ is applied to all positive/negative antidiffusive fluxes which represent the interactions of node i with its neighbors located downstream in the sense of our orientation convention.

The node-oriented limiting strategy makes it possible to control the combined effect of antidiffusive fluxes *acting in concert* rather than merely the variation of the solution along each edge. Moreover, the new limiter extracts all information from the original matrix K and does not need the coordinates of nodes or other geometric details. The equivalence of (40) and (54) reveals that the underlying smoothness indicator r_i is implicitly defined by

$$r_i = \begin{cases} Q_i^+ / P_i^+ & \text{if } u_i \geq u_j, \\ Q_i^- / P_i^- & \text{if } u_i < u_j. \end{cases} \tag{55}$$

It is easy to verify that $\Delta u_{ij} = r_i(u_i - u_j)$ satisfies condition (44) since all coefficients in the sum of upwind contributions $Q_i^\pm$ are nonnegative and

$$\Delta u_{ij} = \sigma_{ij} Q_i^\pm, \qquad \text{where} \quad \sigma_{ij} = \begin{matrix}\max\\ \min\end{matrix} \{0, u_i - u_j\} / P_i^\pm \geq 0. \tag{56}$$

Thus, our nonlinear semi-discrete scheme (38) proves local extremum diminishing if the antidiffusive fluxes are computed from (54). In the sequel, we will call the new algorithm the *flux-limiter* FEM-TVD method to distinguish it from the one described in the preceding section. In one dimension, both generalizations of TVD schemes reduce to their finite difference prototype.

5.6 Iterative Defect Correction

The algorithm presented so far can be classified as a *method of lines* which starts with an approximation of spatial derivatives and yields a system of coupled ordinary differential equations for the time-dependent nodal values u_i. In principle, the discretization of system (38) in time can be performed by any numerical method for solution of initial value problems. First- or second-order accuracy is sufficient for our purposes, so we can use the standard θ-scheme. Furthermore, we concentrate on implicit time-stepping methods ($0 < \theta \leq 1$) because the implementation of the fully explicit one is straightforward. As a result, we end up with a nonlinear algebraic system of the form

$$M_L \frac{u^{n+1} - u^n}{\Delta t} = \theta K^*(u^{n+1})u^{n+1} + (1-\theta)K^*(u^n)u^n \tag{57}$$

which must be solved iteratively. According to the positivity constraint (18), the time step Δt is subject to a CFL-like condition unless $\theta = 1$.

Successive approximations to the end-of-step solution u^{n+1} can be computed e. g. by the fixed-point defect correction scheme [52]

$$u^{(m+1)} = u^{(m)} + A^{-1} r^{(m)}, \qquad m = 0, 1, 2, \ldots \tag{58}$$

where $r^{(m)}$ denotes the residual for the m-th cycle and A is a 'preconditioner' which should be easy to invert. The iteration process continues until the norm of the defect or that of the relative changes becomes small enough.

In a practical implementation, the 'inversion' of A is also performed by a suitable iterative method for solving the linear subproblem

$$A \Delta u^{(m+1)} = r^{(m)}, \qquad m = 0, 1, 2, \ldots \tag{59}$$

After a certain number of inner iterations, the solution increment $\Delta u^{(m+1)}$ is applied to the last iterate, whereby u^n provides a reasonable initial guess

$$u^{(m+1)} = u^{(m)} + \Delta u^{(m+1)}, \qquad u^{(0)} = u^n. \tag{60}$$

Incidentally, the auxiliary problem (59) does not have to be solved very accurately at each outer iteration. By construction, the evolution operator

$$A = M_L - \theta \Delta t L, \qquad L = K + D \tag{61}$$

for the underlying linear LED scheme (21) enjoys the M-matrix property and constitutes an excellent preconditioner. Furthermore, the diagonal dominance of A can be enhanced by means of an implicit underrelaxation [12]. In fact, iterative defect correction preconditioned by the monotone upwind operator is frequently used to enhance the robustness of CFD solvers. This practice is to be recommended even in the linear case, since an iterative method may fail to converge if applied directly to the ill-conditioned matrix originating from a high-order discretization of the troublesome convective terms.

The defect vector and the constant right-hand side are given by

$$r^{(m)} = b^n - [A - \theta \Delta t F(u^{(m)})]u^{(m)}, \tag{62}$$

$$b^n = M_L u^n + (1-\theta)\Delta t[L + F(u^n)]u^n. \tag{63}$$

Both expressions consist of a low-order contribution augmented by limited antidiffusion of the form (39). The antidiffusive fluxes f_{ij}^a are evaluated edge-by-edge at the corresponding time level and inserted into the global vectors. If they are omitted, we recover the nonoscillatory linear scheme (21) which is overly diffusive. The task of the flux limiter is to determine how much artificial diffusion can be safely removed without violating the LED criterion. We remark that our FEM-TVD algorithm is directly applicable to steady-state problems as well as to time-dependent equations written as stationary boundary value problems in the space-time domain (see below).

5.7 Summary of the FEM-TVD Algorithm

The multidimensional flux limiter of TVD type is easy to implement in a finite element code using the conventional or the edge-based data structure. In fact, the required modifications are limited to the matrix assembly routine which is to be called repeatedly in the outer defect correction loop. Since the origin of the discrete transport operator K is immaterial, finite difference and finite volume discretizations of the form (5) are also admissible. The sequence of 'postprocessing' steps to be performed can be summarized as follows:

In a loop over edges:

1. Retrieve the entries k_{ij} and k_{ji} of the high-order transport operator.
2. Determine the artificial diffusion coefficient d_{ij} from equation (20).
3. Update the four entries of the preconditioner A as required by (22).
4. Adopt the edge orientation $\overrightarrow{ij}$ such that node i is located upwind.
5. Store the order of nodes as well as d_{ij} and l_{ji} for future reference.

In a loop over nodes:

6. Calculate the ratio of upstream/downstream contributions $Q_i^{\pm}$ and $P_i^{\pm}$.
7. Apply a TVD limiter Φ to obtain the nodal correction factors $R_i^{\pm}$.

In a loop over edges:

8. Compute the diffusive flux $f_{ij}^d = d_{ij}(u_j - u_i)$ due to discrete upwinding.
9. Check the sign of $u_i - u_j$ and evaluate the antidiffusive flux f_{ij}^a from (54).
10. Insert the corrected internodal flux $f_{ij} = f_{ij}^d + f_{ij}^a$ into the defect vector.

The performance of the above algorithm will be illustrated by the numerical examples at the end of this chapter. The slope-limiter version can be coded in a similar way using the upwind difference (47) to determine the slope ratio $r_i = \Delta u_{ij}/(u_i - u_j)$ and the corresponding correction factors $\Phi(r_i)$. Note that no loop over nodes is needed in this case. Indeed, the recovery of Δu_{ij} via stencil reconstruction is performed independently for each edge. As an alarming consequence, the contributions of other edges are not taken into account, so that the total antidiffusive flux cannot be properly controlled. However, on regular grids the numerical results are typically quite good [26].

6 Algebraic Flux Correction of FCT Type

In contrast to the algebraic TVD methods presented above, FCT algorithms of this kind are designed at the *fully discrete* level. At the same time, the basic principles of algebraic flux correction remain unchanged. The matrix manipulations to be performed (see Fig. 4) are very similar to those outlined in Fig. 1. In particular, the nonoscillatory low-order scheme is constructed by resorting to discrete upwinding and the accuracy is improved by adding nonlinear antidiffusion $f(u^{n+1}, u^n)$ controlled by the flux limiter. Criterion (16) is invoked to ensure that the positivity of u^n is inherited by the auxiliary solution $\tilde{u}$ to an explicit subproblem and carries over to u^{n+1} [22],[23],[28].

1. Linear high-order scheme (e.g. Galerkin FEM)

$$M_C \frac{u^{n+1} - u^n}{\Delta t} = \theta K u^{n+1} + (1-\theta) K u^n, \quad \exists\, j \neq i:\ k_{ij} < 0$$

2. Linear low-order scheme $L = K + D$

$$M_C \frac{u^{n+1} - u^n}{\Delta t} = \theta L u^{n+1} + (1-\theta) L u^n, \quad l_{ij} \geq 0,\ \forall j \neq i$$

3. Nonlinear high-resolution scheme $f_i = \sum_{j \neq i} f_{ij}^a$

$$M_L \frac{u^{n+1} - u^n}{\Delta t} = \theta L u^{n+1} + (1-\theta) L u^n + f(u^{n+1}, u^n)$$

Equivalent representation $Au^{n+1} = B(\tilde{u})\tilde{u}$, where

$A = M_L - \theta \Delta t L$ is an M-matrix and $b_{ij} \geq 0,\ \forall i, j$

Fig. 4. Roadmap of AFC-FCT manipulations

6.1 High- and Low-Order Schemes

The crux of the generalized FCT methodology is to switch between the underlying high- and low-order discretizations in an adaptive fashion so as to satisfy the algebraic positivity constraint. Consider the difference between the nodal ODE systems (5) and (21) which can be formally written as

$$P(u) = (M_L - M_C)\frac{\mathrm{d}u}{\mathrm{d}t} - Du, \tag{64}$$

where the matrices $M_C - M_L$ and $D = L - K$ represent discrete diffusion operators as defined at the beginning of this chapter. Therefore, the raw antidiffusion $P(u)$ can be decomposed into a sum of internodal fluxes

$$P_i = \sum_{j \neq i} f_{ij}, \qquad f_{ij} = -\left[m_{ij}\frac{\mathrm{d}}{\mathrm{d}t} + d_{ij}\right](u_j - u_i), \quad f_{ji} = -f_{ij}. \tag{65}$$

Note that the semi-discrete flux f_{ij} contains a time derivative multiplied by an entry of the consistent mass matrix M_C. For stationary problems, the increment P_i reduces to the sum of *downwind* edge contributions (50) which are associated with the negative off-diagonal coefficients of the operator K.

After the discretization in time by the θ-scheme, the algebraic systems for the high- and low-order methods are related by the formula [23]

$$M_L\frac{u^{n+1} - u^n}{\Delta t} = \theta L u^{n+1} + (1-\theta)Lu^n + P(u^{n+1}, u^n). \tag{66}$$

The last term in the right-hand side is a fully discrete counterpart of $P(u)$. It represents the amount of compensating antidiffusion that needs to be added to the upwind-like low-order scheme to recover the original high-order one (cf. Fig. 4). It is worth mentioning that the discrete operators K, D and $L = K + D$ may depend on the solution if the velocity field does. To simplify notation, we do not indicate this dependence explicitly.

Iterative defect correction makes it possible to resolve the nonlinearities inherent to the governing equation and/or to the discretization procedure into a sequence of well-behaved linear systems of the form (59). The corresponding defect vector for the $m-$th outer iteration is given by

$$r^{(m)} = b^{(m+1)} - Au^{(m)}. \tag{67}$$

The preconditioner A is defined in (61) whereas the load vector

$$b^{(m+1)} = b^n + P(u^{(m)}, u^n) \tag{68}$$

is composed from the right-hand side for the low-order scheme

$$b^n = [M_L + (1-\theta)\Delta t L]u^n \tag{69}$$

and the sum of fully discretized raw antidiffusive fluxes such that

$$P_i^{(m)} = \sum_{j \neq i} f_{ij}^{(m)}, \qquad f_{ji}^{(m)} = -f_{ij}^{(m)}. \tag{70}$$

It follows from (65) that the flux $f_{ij}^{(m)}$ from node j into node i reads

$$f_{ij}^{(m)} = -m_{ij}[\Delta u_{ij}^{(m)} - \Delta u_{ij}^{n}] - \theta \Delta t d_{ij}^{(m)} \Delta u_{ij}^{(m)} - (1-\theta)\Delta t d_{ij}^{n} \Delta u_{ij}^{n}, \tag{71}$$

where the explicit and implicit antidiffusion is proportional to

$$\Delta u_{ij}^{n} = u_j^n - u_i^n \quad \text{and} \quad \Delta u_{ij}^{(m)} = u_j^{(m)} - u_i^{(m)}, \tag{72}$$

respectively. The implicit part must be updated in each defect correction cycle while the explicit one is to be computed just once per time step.

Substitution of expressions (61) and (67) into (58) reveals that successive approximations to the high-order solution can be computed as follows

$$Au^{(m+1)} = b^{(m+1)}, \qquad m = 0, 1, 2, \ldots \tag{73}$$

By construction, the raw antidiffusive fluxes offset not only the error induced by discrete upwinding but also the numerical diffusion due to mass lumping that could not be removed in the framework of the FEM-TVD methodology. Therefore, FEM-FCT schemes are typically more accurate for strongly time-dependent problems which call for the use of the consistent mass matrix M_C. At the same time, the use of the lumped mass matrix M_L is appropriate for less dynamic ones or those being marched to a steady state. In this case, the first term in the right-hand side of (71) should be omitted.

An extra stabilization of convective terms appears to be necessary for fully explicit schemes [23]. Stabilized finite element methods are typically based on some kind of streamline diffusion which can be added explicitly, incorporated into the test function or emulated by high-order time derivatives in the Taylor series expansion [7],[8],[10]. Recall that streamline diffusion operators are of the form (8)-(9) so that decomposition (11) is feasible. We refer to Löhner *et al.* [35],[36],[33] for a presentation of the explicit FEM-FCT algorithm and restrict ourselves to implicit Galerkin schemes. They enjoy unconditional stability for $\theta \geq 0.5$ and constitute viable high-order methods as long as spurious undershoots and overshoots are precluded by a built-in flux limiter.

6.2 Basic FEM-FCT Algorithm

Due to the fact that the preconditioner A was designed to be an M-matrix, the left-hand side of (73) already satisfies the algebraic constraint (16) and so does the right-hand side if the antidiffusive correction $P(u^{n+1}, u^n)$ is omitted. Clearly, it is desirable to retain as much antidiffusion as possible without generating new extrema and accentuating already existing ones. To this end,

the raw antidiffusive fluxes should be multiplied by appropriate correction factors before they are inserted into the right-hand side. This adjustment should guarantee that the discrete scheme remains positivity-preserving.

A generalized FEM-FCT formulation based on algebraic flux correction was introduced in [22],[23],[25]. The limited antidiffusive fluxes belong into the right-hand side of system (73) which is to be redefined as follows

$$b_i^{(m+1)} = b_i^n + \sum_{j \neq i} \alpha_{ij}^{(m)} f_{ij}^{(m)}, \qquad 0 \leq \alpha_{ij}^{(m)} \leq 1. \tag{74}$$

It is easy to verify that the usual FCT algorithm is recovered for $\theta = 0$. Let us leave the solution-dependent correction factors $\alpha_{ij}^{(m)}$ unspecified for the time being but draw attention to the fact that they are bounded by 1, whereas flux limiters of TVD type are allowed to accept more than 100% of the raw antidiffusive flux. Another notable distinction between the two techniques is that FCT limiters are invariant to the edge orientation. Hence, it is no longer necessary to check which of the two nodes is located upwind (cf. section 6).

As already mentioned above, the antidiffusive fluxes should be limited so as to enforce the AFC constraint (16) making use of an intermediate solution which is supposed to be positivity-preserving. Consider the subproblem

$$M_L \tilde{u}^n = b^n \tag{75}$$

such that $\tilde{u}^n$ corresponds to the explicit low-order solution at the instant $t^{n+1-\theta}$ and reduces to $\tilde{u}^n = u^n$ in the case $\theta = 1$ (backward Euler method). Other time-stepping schemes preserve the positivity of u^n provided that

$$\Delta t \leq \frac{1}{1-\theta} \min_i \left\{ -m_i / l_{ii} |\ l_{ii} < 0 \right\}, \qquad 0 \leq \theta < 1. \tag{76}$$

This readily computable upper bound follows from our CFL-like condition (18) and can serve as the threshold parameter for an adaptive time step control. We remark that the intermediate solution $\tilde{u}^n$ is independent of the iteration counter m and does not change in the course of defect correction.

For the right-hand side $b^{(m+1)}$ to possess the desired representation, the flux limiter should guarantee the existence of a matrix $B = \{b_{ij}\}$ such that

$$b^{(m+1)} = B(\tilde{u}^n)\tilde{u}^n \qquad \text{and} \quad b_{ij} \geq 0, \quad \forall i, j. \tag{77}$$

Under this condition, the resulting scheme proves positivity-preserving since the linear system (73) for each solution update can be cast in the form (16) with $\tilde{u}^n$ in lieu of u^n. It goes without saying that the matrix $B(\tilde{u}^n)$ does not need to be constructed explicitly. We will see shortly that a new interpretation of Zalesak's limiter provides the necessary mechanism for the computation of 'optimal' correction factors for our algebraic FCT schemes.

6.3 Iterative FEM-FCT Algorithm

The main advantage of implicit FEM-FCT schemes is their ability to cope with large time steps. However, the artificial diffusion introduced by discrete upwinding is proportional to the time step, while the amount of acceptable antidiffusion depends solely on the local extrema of the auxiliary solution. Hence, a smaller percentage of the raw antidiffusive flux survives the limiting step as the local Courant number increases. To circumvent this deficiency of our basic FEM-FCT algorithm, we introduce an iterative limiting strategy which prevents the numerical solution from becoming increasingly diffusive at large time steps. A somewhat similar technique was developed by Schär and Smolarkiewicz [47] in the finite difference context but their methodology is inherently explicit so that iterative flux correction does not pay off.

The iterative FEM-FCT procedure [28] differs from the algorithm presented above in that the previously accepted antidiffusion is taken into account and only the rejected portion of the antidiffusive flux needs to be dealt with at subsequent defect correction steps. To this end, the limited antidiffusion is incorporated into the *variable* auxiliary solution $\tilde{u}^{(m)}$ which must be updated along with the right-hand side of (73) at each outer iteration

$$M_L \tilde{u}^{(m)} = b^{(m)}, \qquad b^{(0)} = b^n. \tag{78}$$

Recall that M_L is a diagonal matrix so that no linear system has to be solved and the overhead cost associated with the computation of $\tilde{u}^{(m)}$ is negligible.

Furthermore, the correction factors $\alpha_{ij}^{(m)}$ are based on the local extrema of $\tilde{u}^{(m)}$ rather than $\tilde{u}^n$ and applied to the difference between the raw antidiffusive flux $f_{ij}^{(m)}$ and the cumulative effect of previous corrections $g_{ij}^{(m)}$ which is initialized by zero at the beginning of a new time step

$$\Delta f_{ij}^{(m)} = f_{ij}^{(m)} - g_{ij}^{(m)}, \qquad g_{ij}^{(0)} = 0. \tag{79}$$

The limited flux difference is added to the sum of its predecessors

$$g_{ij}^{(m+1)} = g_{ij}^{(m)} + \alpha_{ij}^{(m)} \Delta f_{ij}^{(m)} \tag{80}$$

and inserted into the right-hand side of the linear system to be solved

$$b_i^{(m+1)} = b_i^{(m)} + \sum_{j \neq i} \alpha_{ij}^{(m)} \Delta f_{ij}^{(m)}. \tag{81}$$

At the first outer iteration, Zalesak's limiter is applied to $\Delta f_{ij}^{(0)} = f_{ij}^{(0)}$ and $\tilde{u}^{(0)} = \tilde{u}^n$ so that the load vector $b^{(1)}$ is identical to that defined in (74). As the iteration process continues, more and more antidiffusion can be built into the auxiliary solution $\tilde{u}^{(m)}$ while the remainder $\Delta f_{ij}^{(m)}$ gradually shrinks. This simplifies the task of the flux limiter and enables it to remove excessive artificial diffusion step-by-step in a positivity-preserving manner.

It is easy to verify by successive substitution that the load vector $b^{(m+1)}$ consists of the low-order contribution b^n and the limited antidiffusion accumulated in the course of iterative defect/flux correction:

$$b_i^{(m+1)} = b_i^n + \sum_{j \neq i} g_{ij}^{(m+1)}, \qquad g_{ij}^{(m+1)} = \sum_{k=0}^{m} \alpha_{ij}^{(k)} \Delta f_{ij}^{(k)}. \tag{82}$$

Moreover, the right-hand side for the high-order Galerkin discretization (68) is recovered if no limiting of $\Delta f_{ij}^{(m)}$ is performed in the $m-$th iteration

$$\alpha_{ij}^{(m)} \equiv 1 \quad \Rightarrow \quad b_i^{(m+1)} = b_i^n + \sum_{j \neq i} (g_{ij}^{(m)} + \Delta f_{ij}^{(m)}) = b_i^n + \sum_{j \neq i} f_{ij}^{(m)}.$$

The task of the flux limiter it to select the correction factors $\alpha_{ij}^{(m)}$ so as to satisfy an analog of (77) and make each solution update $\tilde{u}^{(m+1)} = M_L^{-1} b^{(m+1)}$ positivity-preserving. In the fully explicit case (forward Euler time-stepping, lumped mass matrix) just one iteration is necessary so that $u^{n+1} = \tilde{u}^{(1)}$ is the final solution. For our implicit FEM-FCT schemes, the M-matrix property of the preconditioner A ensures that $u^{(m+1)} \geq 0$ as long as $\tilde{u}^{(m+1)} \geq 0$.

6.4 Zalesak's Limiter

Let us describe Zalesak's algorithm for the computation of correction factors and check if the right-hand side satisfies the following positivity constraint:

$$b_i = m_i \tilde{u}_i + \sum_{j \neq i} \alpha_{ij} f_{ij} \geq 0 \qquad \text{if} \quad \tilde{u}_j \geq 0, \quad \forall j. \tag{83}$$

Note that both (74) and (81) are of this form, so there is no need to distinguish between the basic and iterative limiting strategy in this section.

Varying the correction factors α_{ij} between zero and unity, one can blend the high-order method with the concomitant low-order one. The latter should be used in the vicinity of steep gradients where spurious oscillations are likely to arise. The objective is to control the interplay of antidiffusive fluxes so that they cannot conspire to create or enhance a local extremum [54]. Moreover, all antidiffusive fluxes directed down the gradient of $\tilde{u}$ should be canceled from the outset to prevent the formation of plateaus amidst a steep front

$$f_{ij} := 0 \qquad \text{if} \quad f_{ij}(\tilde{u}_i - \tilde{u}_j) \leq 0. \tag{84}$$

In other words, the antidiffusive flux is not allowed to flatten the auxiliary solution. This optional *prelimiting step* can be traced back to the SHASTA scheme of Boris and Book who designed their limiter so as to reverse (rather than cancel) such 'defective' fluxes [6]. Prelimiting of the form (84) was introduced by Zalesak [54] in relations (14) and (14') of his paper. At the same time,

he argued that the effect of the above amendment is marginal and cosmetic in nature since the vast majority of antidiffusive fluxes entail a steepening of the gradient. Two decades later, the virtues of prelimiting were rediscovered by DeVore [9] who explained its ramifications and demonstrated that it may lead to an appreciable improvement of simulation results. If this step is missing, the FCT limiter appears to be positivity- but not monotonicity-preserving so that solutions may be corrupted by numerical ripples of significant amplitude. This fact was also confirmed by our own numerical experiments [23]. Hence, it is advisable to prelimit the fluxes prior to the computation of α_{ij}.

In the worst case, all antidiffusive fluxes into node i have the same sign. Therefore, it is worthwhile to split the increment P_i as defined in (70) into a sum of positive contributions and a sum of negative ones, cf. (51)

$$P_i = P_i^+ + P_i^-, \qquad P_i^\pm = \sum_{j\neq i} \mathop{\max}_{\min}\{0, f_{ij}\}. \tag{85}$$

The maximum/minimum admissible increment depends on the solution values at the neighboring nodes that share an element/edge with node i

$$Q_i^\pm = \mathop{\max_j}_{\min_j} \Delta u_{ij}^\pm, \qquad \text{where} \quad \Delta u_{ij}^\pm = \mathop{\max}_{\min}\{0, \tilde{u}_j - \tilde{u}_i\}. \tag{86}$$

This corresponds to the following upper/lower bounds for the nodal value

$$\tilde{u}_i^{\max} = \tilde{u}_i + Q_i^+, \qquad \tilde{u}_i^{\min} = \tilde{u}_i + Q_i^-.$$

In order to prevent the formation of a spurious overshoot/undershoot, the positive/negative antidiffusive flux f_{ij} should be multiplied by

$$R_i^\pm = \begin{cases} \min\{1, m_i Q_i^\pm / P_i^\pm\} & \text{if } P_i^\pm \neq 0, \\ 1 & \text{if } P_i^\pm = 0. \end{cases} \tag{87}$$

In our experience, it makes sense to set $R_i^\pm := 1$ at the inlet, where the Dirichlet boundary conditions override the effect of any antidiffusive correction. The same adjustment can/should be performed at outflow boundaries [28],[40].

Recall that a positive flux f_{ij} into node i is always balanced by a negative flux $f_{ji} = -f_{ij}$ into node j and vice versa. Hence, one should check the sign of the flux and apply the minimum of the nodal correction factors

$$\alpha_{ij} = \begin{cases} \min\{R_i^+, R_j^-\} & \text{if } f_{ij} \geq 0, \\ \min\{R_j^+, R_i^-\} & \text{if } f_{ij} < 0, \end{cases} \qquad \alpha_{ji} = \alpha_{ij}. \tag{88}$$

This choice of α_{ij} is safe enough to guarantee that $\tilde{u}_i^{\min} \leq b_i/m_i \leq \tilde{u}_i^{\max}$ so that no enhancement of local extrema takes place. Remarkably, the above algorithm is independent of the underlying discretization and can be implemented as a 'black-box' routine which computes the correction factors for a given auxiliary solution $\tilde{u}$ and an array of raw antidiffusive fluxes f_{ij}.

It remains to prove that the right-hand side b satisfies (83) for a nontrivial parameter constellation such that the antidiffusive correction to node i is nonvanishing. Let k be the number of a neighboring node such that [22],[23]

$$b_i = m_i\tilde{u}_i + c_iQ_i = (m_i - c_i)\tilde{u}_i + c_i\tilde{u}_k, \tag{89}$$

where the auxiliary quantities c_i and Q_i are defined as follows

$$c_i = \frac{\sum_{j\neq i}\alpha_{ij}f_{ij}}{Q_i}, \qquad Q_i = \begin{cases} Q_i^+ & \text{if } \sum_{j\neq i}\alpha_{ij}f_{ij} > 0, \\ Q_i^- & \text{if } \sum_{j\neq i}\alpha_{ij}f_{ij} < 0. \end{cases} \tag{90}$$

Note that division by zero is ruled out since $R_i^{\pm} = 0$ and all positive/negative antidiffusive fluxes into node i are canceled completely in this case:

$$f_{ij} := 0 \qquad \text{if} \quad (Q_i^+ = 0 \ \wedge \ f_{ij} > 0) \ \vee \ (Q_i^- = 0 \ \wedge \ f_{ij} < 0).$$

By definition, the coefficient c_i is nonnegative and it is easy to verify that the following estimate holds by virtue of relations (85)–(88)

$$m_iQ_i^- \leq R_i^-P_i^- \leq \sum_{j\neq i}\alpha_{ij}f_{ij} \leq R_i^+P_i^+ \leq m_iQ_i^+. \tag{91}$$

Thus, the limited antidiffusive fluxes satisfy the double inequality $m_i \geq c_i \geq 0$ and it follows from (89) that b_i is nonnegative for $\tilde{u}_i \geq 0$ and $\tilde{u}_k \geq 0$. Moreover, the positivity of both coefficients in this representation of the right-hand side proves the existence of the matrix $B(\tilde{u})$ in the last box of Fig. 4.

6.5 Clipping and Terracing

Like any other numerical method, the FCT algorithm involves a certain degree of empiricism in the reconstruction of data so that the approximate solutions may exhibit various artifacts such as *clipping* and *terracing* [43]. The former phenomenon refers to a smearing of sharp peaks in the convected profile due to the fact that they cannot be properly resolved on the given mesh and their resurrection is prohibited by the flux limiter. Zalesak [54] managed to alleviate peak clipping by using the old solution u^n along with $\tilde{u}$ in the estimation of the solution bounds. However, this practice is not to be recommended if local extrema decay with time due to physical effects such as compression, expansion and sources/sinks. In this case, the use of information from the previous time step may result in an undershoot/overshoot [23],[25],[43].

Terracing manifests itself in a distortion of smooth profiles and represents 'an integrated, nonlinear effect of residual phase errors' [43] or, loosely speaking, 'the ghosts of departed ripples' [5]. In particular, this problem frequently occurs at outflow boundaries as illustrated in Fig. 5 (left) for the 1D convection equation (23) with $v = 1$ and initial data $u^0 = x$ in $\Omega = [0, 1]$. The linear function, for which flux correction is actually redundant and the

standard Galerkin method would produce excellent results, degenerates into a broken line. The alternating steepening/flattening of the gradient leads to a formation of spurious plateaus. This indicates that FCT algorithms are not *linearity-preserving* and introduce too much antidiffusion in some cases.

Some preliminary speculations regarding the cause and cure of terracing at inflow and outflow boundaries can be found in [23],[28]. In particular, it turns out that a boundary value $\tilde{u}_i$ can be *misinterpreted* as a local extremum [40]. Indeed, it is only compared to the solution values at the neighboring nodes and no information about the solution behavior beyond the (artificial) open boundary is available. The erroneous cancellation of antidiffusive fluxes at the inlet/outlet entails a redistribution of mass in the interior of the domain and eventually leads to the formation of terraces. To illustrate this effect, a heuristic *lever model* was introduced in [40]. Let the piecewise-linear solution be represented by levers of variable length hinged at their midpoints, which correspond to the element mean values, and connected continuously with one another. Pulling down the rightmost lever results in a shearing force which affects the slopes of all components as shown in Fig. 5 (right).

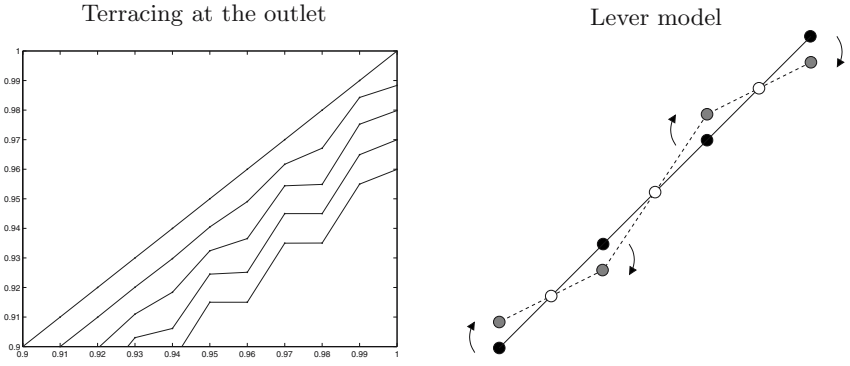


Fig. 5. Pathological behavior of FCT

Interestingly enough, the ripples disappear if we set $R_i^{\pm} := 1$ for nodes belonging to the inflow and outflow boundaries (see above). At the inlet, this adjustment is admissible since the boundary values of the solution are fixed. At the outlet, it amounts to limiting the antidiffusive flux using the correction factor for the *upwind node* as in the case of TVD methods which are largely immune to terracing. The synchronization of nodal correction factors in (88) makes the FCT algorithm invariant to the flow direction and, therefore, vulnerable to incorrect upper/lower bounds for the node located downstream. Hence, the use of an upwind-biased limiting strategy is preferable in the vicinity of open boundaries and local extrema, where terracing is likely to occur as an aftermath of peak clipping. Due to conservation, the clipped mass must be distributed between the neighboring nodes and may easily go astray.

Unfortunately, a complete analysis of clipping and terracing is not available to date and it is not quite clear how to combat these artifacts. An important prerequisite seems to be the use of background diffusion [34]. It is hoped that a further investigation of Zalesak's limiter in the framework of algebraic flux correction and a detailed comparison of FCT with other high-resolution schemes will make it possible to find an effective remedy.

6.6 Summary of the FEM-FCT Algorithm

The new FEM-FCT methodology represents a generalization of the explicit algorithm proposed by Löhner et al. [35]. Obviously, it has a lot in common with the FEM-TVD approach introduced in the first part of this chapter. However, the construction and limiting of antidiffusive fluxes is performed *after* the discretization in time so that the implementation is slightly different. The basic steps and the corresponding parts of the code are as follows:

In the matrix assembly routine:

1. Retrieve the entries k_{ij} and k_{ji} of the high-order transport operator.
2. Determine the artificial diffusion coefficient d_{ij} from equation (20).
3. Update the four entries of the preconditioner A as required by (22).
4. Substitute the explicit diffusive flux into b^n (at the first iteration).
5. Evaluate/increment the raw antidiffusive flux given by (71) or (79).

In the flux correction module:

6. Initialize/update the auxiliary solution $\tilde{u}$ according to (75) or (78).
7. Use Zalesak's limiter to compute the correction factors from (88).
8. Insert the limited antidiffusive fluxes into (74) or (80)–(81).

In the defect correction loop:

9. Solve the linear system (73) or (59)–(60) with $r^{(m)}$ defined in (67).
10. Check convergence and proceed to the next iteration or time step.

As far as the time discretization is concerned, the second-order accurate Crank-Nicolson scheme is to be recommended for transient problems. In this case, the time step must remain relatively small to capture the evolution details and satisfy condition (76). Hence, the basic FEM-FCT algorithm is almost as accurate as the iterative one and certainly more efficient. On the other hand, the solution of steady-state problems by *pseudo-time-stepping* calls for the fully implicit treatment. The backward Euler method, which is only first-order accurate, is also appropriate if a nonuniform distribution of Courant numbers (e.g. due to mesh refinement or a strongly varying velocity field) makes the CFL condition too restrictive. Fully implicit FEM-FCT schemes are unconditionally positivity-preserving and the iterative formulation should be preferred to prevent loss of accuracy at large time steps.

7 Numerical Examples

The examples that follow illustrate the influence of the time discretization and of the flux limiter on the numerical results. An implicit time-stepping is employed in all cases. The performance of the flux-corrected Lax-Wendroff method (explicit, second-order accurate) was studied in [23], where many additional examples for the basic FEM-FCT algorithm can be found. Furthermore, only linear (triangular) and bilinear (quadrilateral) finite elements are considered in this chapter. Three-dimensional simulation results obtained using a (discontinuous) rotated bilinear approximation are presented in [27],[53].

The convergence of one-dimensional FCT and TVD schemes was investigated in [40] and [50], respectively. In particular, the effective order of accuracy p was estimated from the difference between the errors for the solutions computed on two sufficiently fine meshes. It can be shown that [12],[32]

$$p \approx \log_2(E_h/E_{2h}),$$

where E_h is the norm of the error on a uniform mesh with spacing h between the grid points. As reported in [40], the actual convergence rates depend on the discretization procedure and on smoothness of the exact solution. For details, the interested reader is referred to the original publications.

7.1 Solid Body Rotation

Rotation of solid bodies is frequently used to evaluate and compare numerical schemes for convection-dominated problems. A classical example is Zalesak's slotted cylinder test [54] which is intended to assess the ability of the method to cope with steep gradients and reproduce small-scale features. In order to examine the resolution of both smooth and discontinuous profiles, we consider an extended version of this 2D benchmark as proposed by LeVeque [32].

Let a slotted cylinder, a sharp cone and a smooth hump be exposed to the nonuniform velocity field $\mathbf{v} = (0.5-y, x-0.5)$ and undergo a counterclockwise rotation about the center of the unit square $\Omega = (0,1) \times (0,1)$. The initial configuration for this test is depicted in Fig. 6. Each solid body lies within a circle of radius $r_0 = 0.15$ centered at a point with Cartesian coordinates (x_0, y_0). In the rest of the domain, the solution is initialized by zero.

The shapes of the three bodies can be expressed in terms of the normalized distance function for the respective reference point (x_0, y_0)

$$r(x,y) = \frac{1}{r_0}\sqrt{(x-x_0)^2 + (y-y_0)^2}.$$

The center of the slotted cylinder is located at $(x_0, y_0) = (0.5, 0.75)$ and its geometry in the circular region $r(x,y) \le 1$ is given by

$$u(x,y,0) = \begin{cases} 1 & \text{if } |x-x_0| \ge 0.025 \ \vee \ y \ge 0.85, \\ 0 & \text{otherwise.} \end{cases}$$

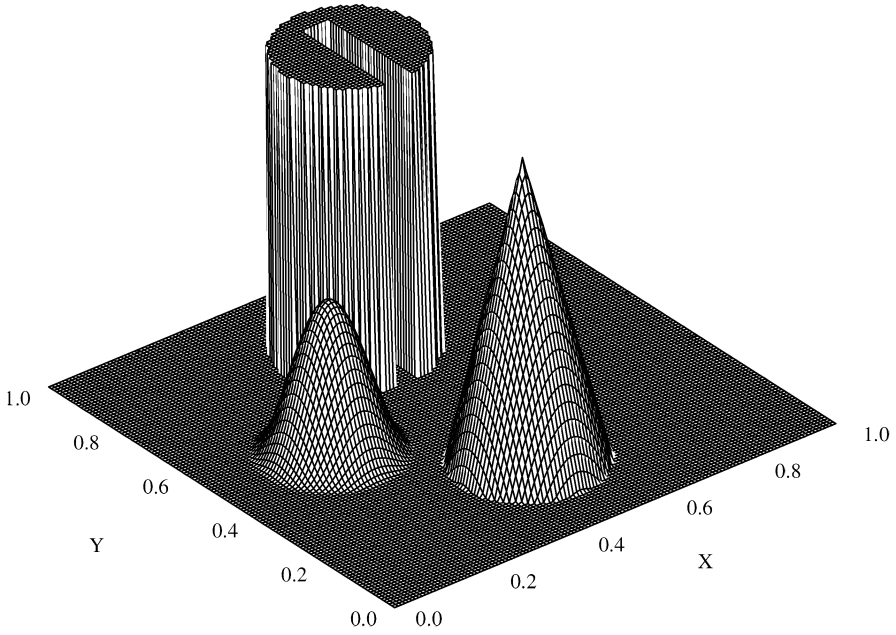


Fig. 6. Initial data and exact solution at $t = 2\pi$

The corresponding analytical expression for the conical body reads

$$u(x, y, 0) = 1 - r(x, y), \qquad (x_0, y_0) = (0.5, 0.25),$$

whereas the shape and location of the hump at $t = 0$ are as follows

$$u(x, y, 0) = 0.25[1 + \cos(\pi \min\{r(x, y), 1\})], \quad (x_0, y_0) = (0.25, 0.5).$$

After one full revolution ($t = 2\pi$) the exact solution to the pure convection equation (1) coincides with the initial data. To expose the deficiencies of linear discretizations, we present the numerical results produced by the standard Galerkin method and its low-order counterpart (discrete upwinding) in Fig. 7. These solutions were computed on a mesh of 128×128 bilinear elements using the second-order accurate Crank-Nicolson time-stepping with $\Delta t = 10^{-3}$. Sure enough, the original high-order scheme reproduces the cone and hump very well but gives rise to spurious wiggles that can be traced to the slotted cylinder. On the other hand, the low-order solution is nonoscillatory but its quality is extremely poor due to the devastating effect of artificial diffusion.

A nonlinear combination of these imperfect linear methods in the framework of algebraic flux correction yields the numerical solutions shown in Fig. 8. The nonphysical oscillations disappear and the resolution of the three bodies is still remarkably crisp. In the case of the FEM-TVD method, the narrow

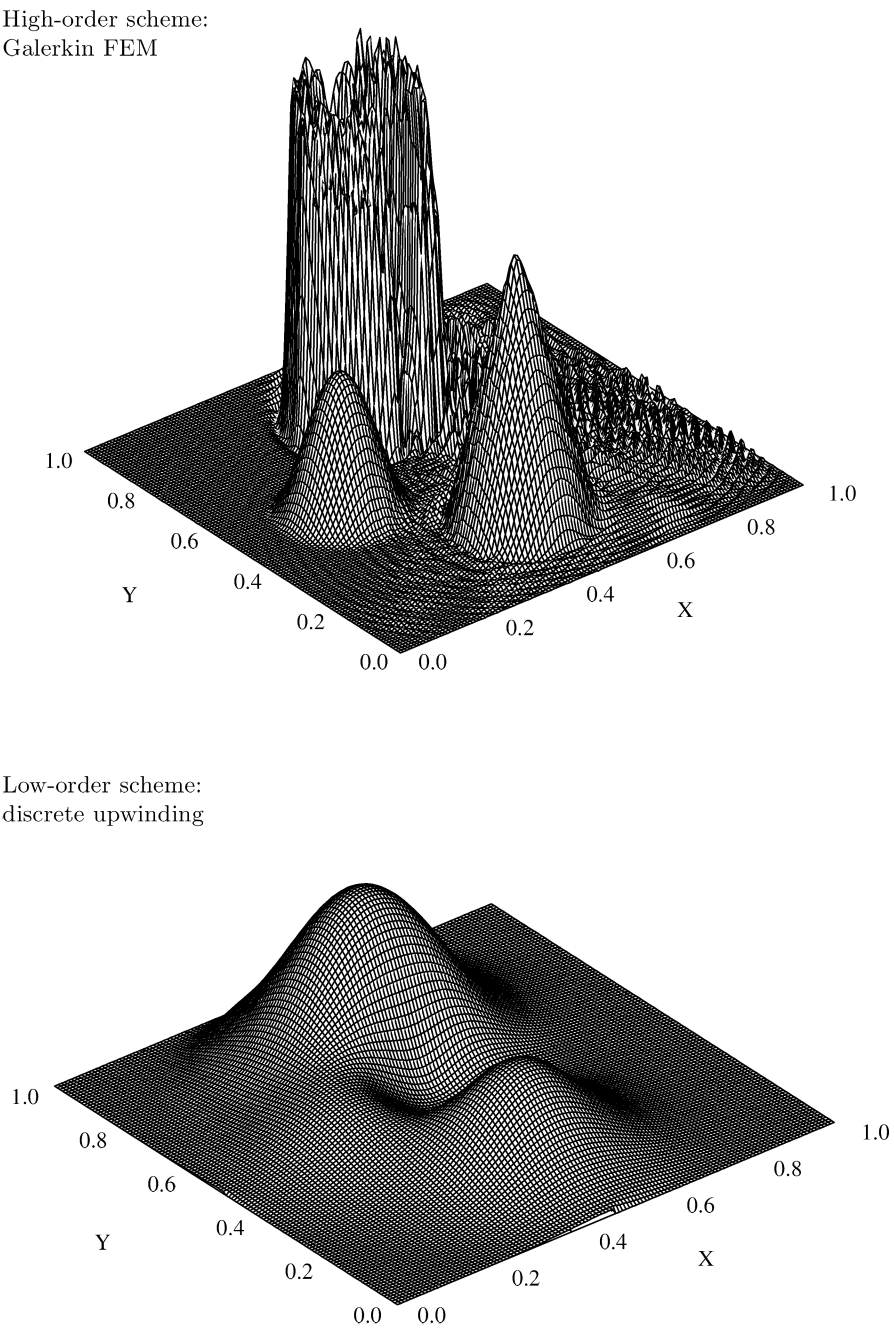


Fig. 7. Solid body rotation: shortcomings of linear methods

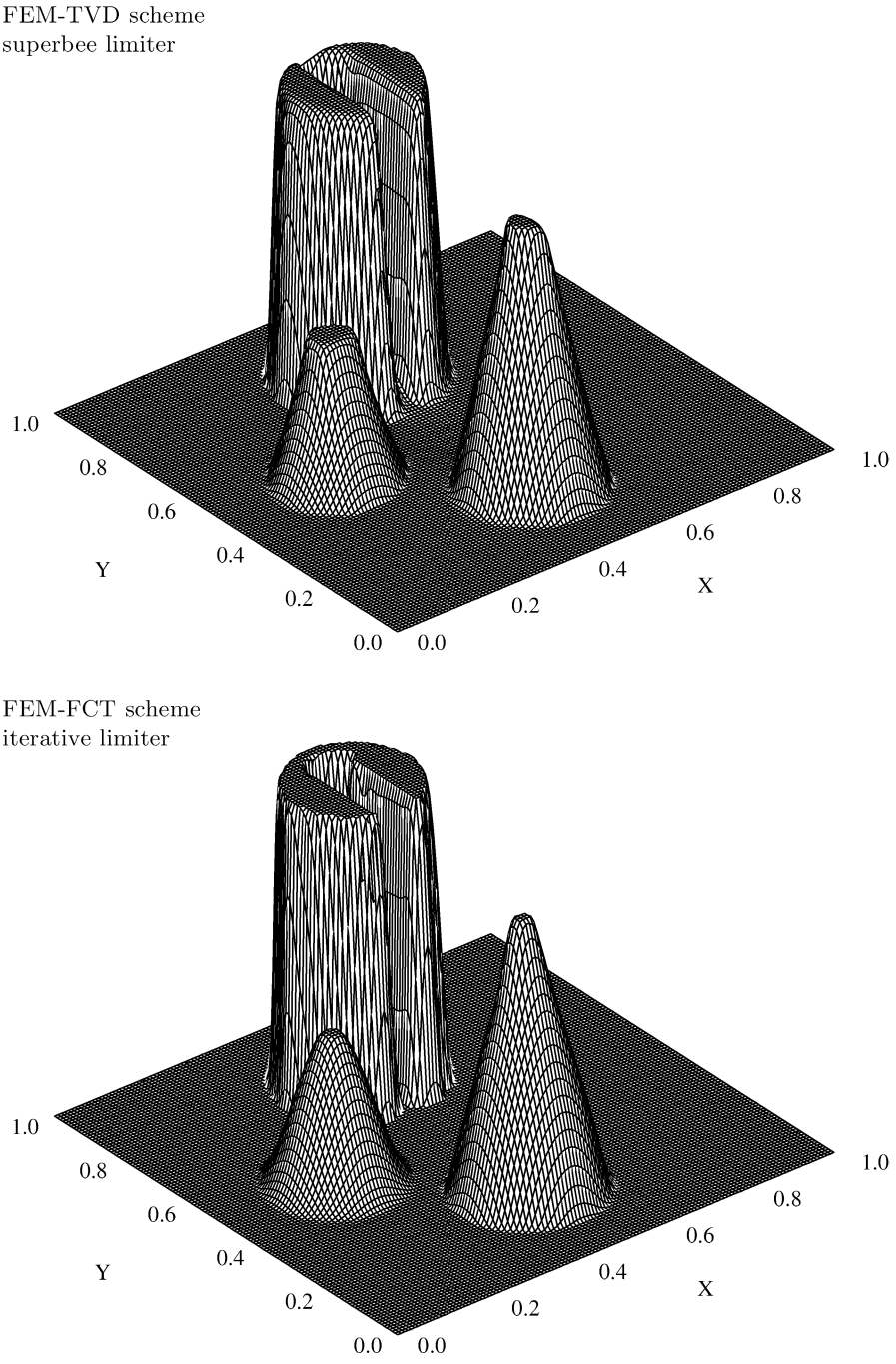


Fig. 8. Solid body rotation on a quadrilateral mesh

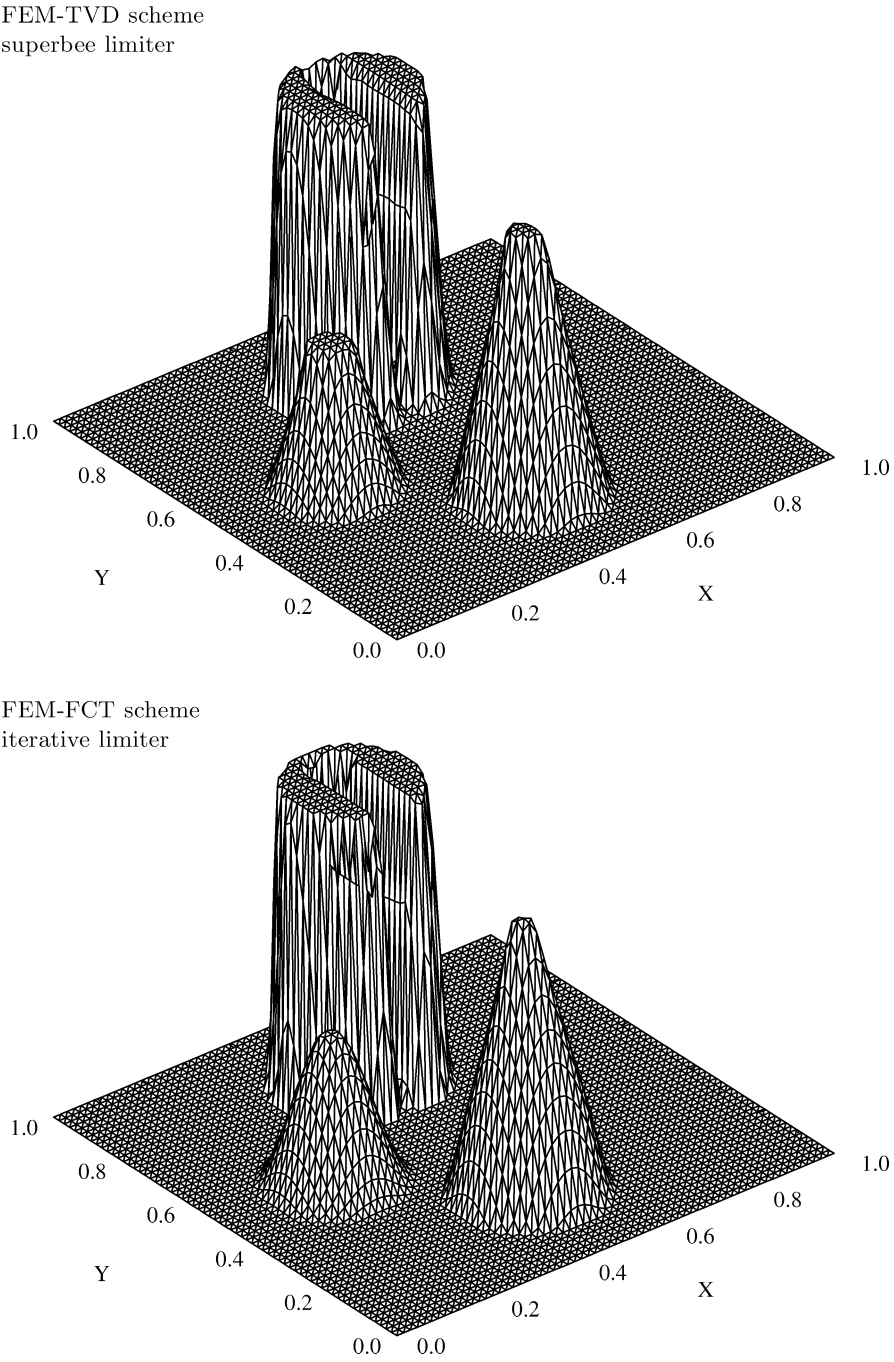


Fig. 9. Solid body rotation on a triangular mesh

bridge of the cylinder is largely preserved but some erosion of the ridges is observed. The overall accuracy is acceptable since the numerical diffusion due to mass lumping is alleviated to some extent by the strongly antidiffusive superbee limiter. At the same time, the excessive antidiffusion entails an artificial steepening of the gradients as well as a gradual flattening of the two peaks, which can be interpreted as a weak form of 'clipping' and 'terracing'. Other TVD limiters are more diffusive (see below) so that the lumping error is aggravated and a pronounced smearing of the solution profiles ensues.

For this strongly time-dependent test problem, the iterative FEM-FCT algorithm performs much better, which can be attributed to the use of the consistent mass matrix. The prelimiting of antidiffusive fluxes was found to be essential. If this optional step is omitted, the ridges of the cylinder are corrupted by harmless but optically disturbing kinks. The (inevitable) peak clipping for the cone does not exceed 10% and the hump is reproduced almost exactly. Similar results are obtained using a piecewise-linear approximation on the mesh constructed from the former one by subdivision of each quadrilateral into two triangles. For visualization purposes, the solutions shown in Fig. 9 were output on a coarser mesh of $64 \times 64 \times 2$ triangular elements.

7.2 Swirling Flow Problem

Another challenging test problem introduced by LeVeque [32] deals with a swirling deformation of the initial data by the incompressible velocity field shown in Fig. 12. The two velocity components are given by

$$v_x = \sin^2(\pi x)\sin(2\pi y), \quad v_y = -\sin^2(\pi y)\sin(2\pi x).$$

The initial condition is a discontinuous function which equals unity within a circular sector of $\pi/2$ radians and zero elsewhere:

$$u(x,y,0) = \begin{cases} 1 & \text{if } (x-1)^2 + (y-1)^2 < 0.64, \\ 0 & \text{otherwise.} \end{cases}$$

The computational results obtained at time $t = 2.5$ using the same parameter settings as in the previous example are shown in Fig. 10–11. In the course of deformation, the mass distribution assumes a complex spiral shape which is nicely resolved by both algebraic flux correction schemes under consideration. The imposed constraints are satisfied and the approximate solutions remain bounded by zero and one regardless of the mesh type. Again, the better accuracy of FEM-FCT as compared to FEM-TVD is due to the dynamic nature of the problem at hand. By construction, the latter algorithm is independent of the time discretization and lends itself to the treatment of steady-state applications. As already mentioned, the optimal degree of implicitness is also problem-dependent. In this example, the Crank-Nicolson time-stepping was selected, since Δt must be chosen impractically small for comparable results to be produced by the fully implicit backward Euler method [23],[25].

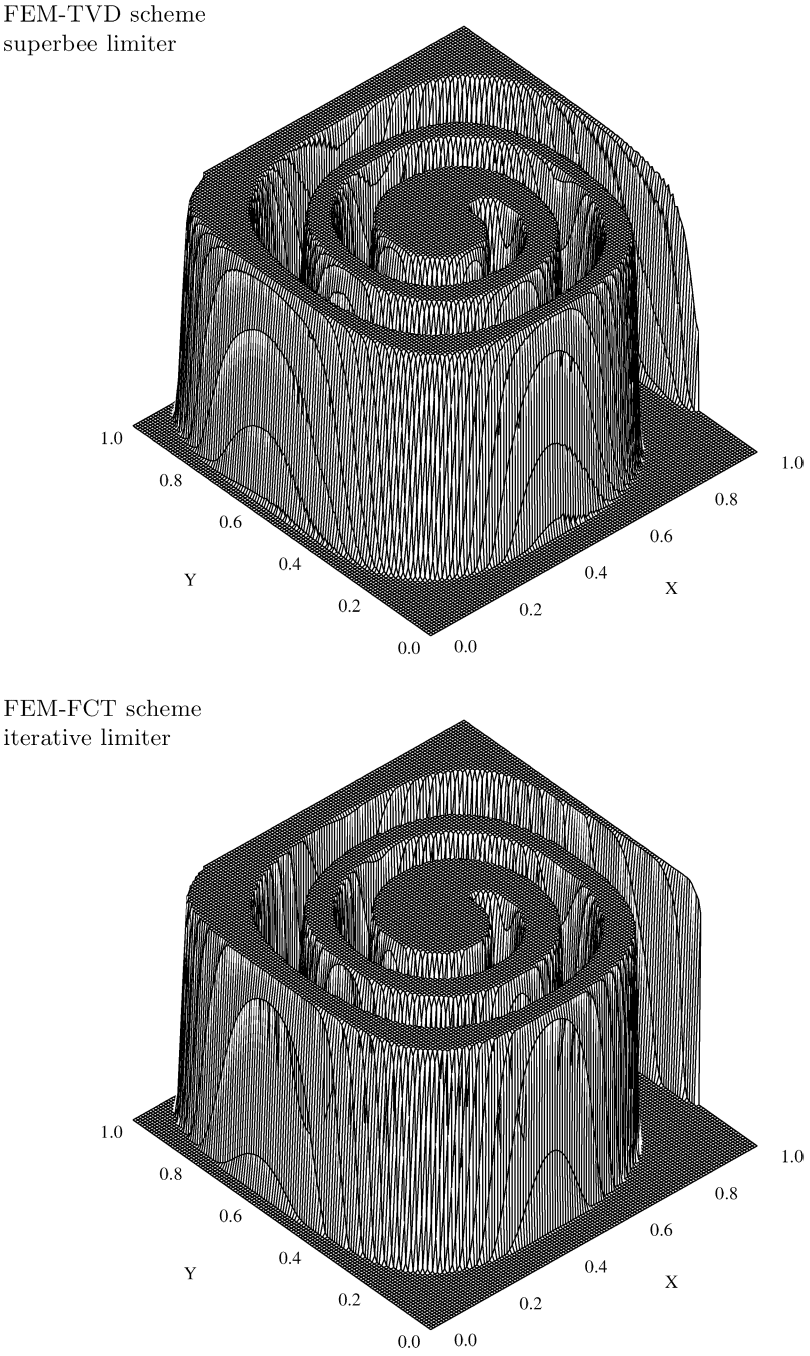


Fig. 10. Swirling deformation on a quadrilateral mesh

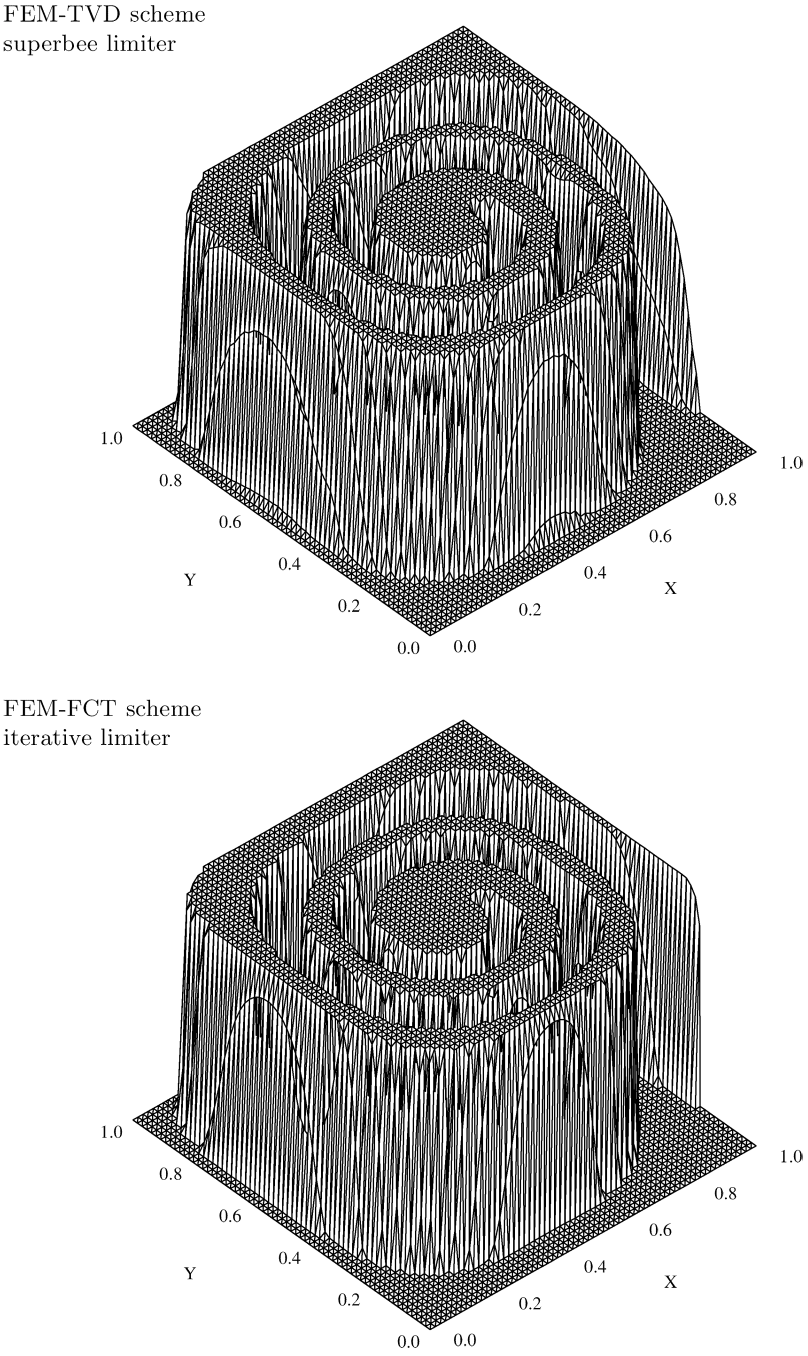


Fig. 11. Swirling deformation on a triangular mesh

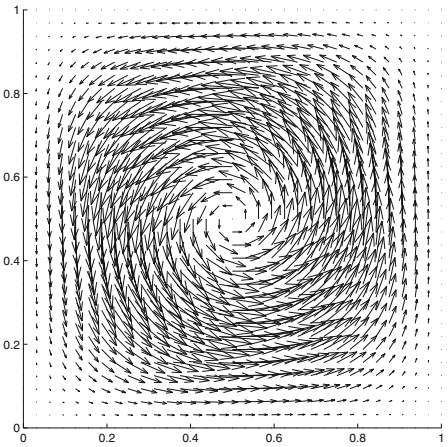


Fig. 12. Velocity field for the swirling flow

7.3 Rotation of a Gaussian Hill

The third test problem devised by Lapin [29] makes it possible to assess the magnitude of artificial diffusion due to the discretization in space and time. This can be accomplished via statistical analysis of numerical solutions to the nonstationary convection-diffusion equation

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u = \epsilon \Delta u \qquad \text{in } \Omega = (-1,1) \times (-1,1), \tag{92}$$

where $\mathbf{v} = (-y, x)$ is the velocity and $\epsilon = 10^{-3}$ is the diffusion coefficient.

The initial condition to be imposed is given by $u(x, y, 0) = \delta(x_0, y_0)$, where δ is the Dirac delta function. In a practical implementation, it is impossible to initialize the solution by a singular function. Instead, the whole mass should be concentrated at a single node of the computational mesh. The integral of the discrete solution over the domain Ω equals the sum of nodal values multiplied by the diagonal entries of the lumped mass matrix

$$\int_\Omega u_h \, \mathrm{d}\mathbf{x} = \int_\Omega \sum_j u_j \varphi_j \, \mathrm{d}\mathbf{x} = \sum_i m_i u_i.$$

The total mass of a delta function equals unity. Hence, one should find node i closest to the peak location (x_0, y_0) and set $u_i^0 = 1/m_i$, $u_j^0 = 0$, $j \neq i$. Alternatively, one can start with the exact solution at a time $t_0 > 0$.

In the rotating Lagrangian reference frame, the convective term vanishes and the resulting diffusion problem can be solved analytically. The solution is a rotating Gaussian hill defined by the normal distribution function

$$u(x, y, t) = \frac{1}{4\pi\epsilon t} \, e^{-\frac{r^2}{4\epsilon t}}, \qquad r^2 = (x - \hat{x})^2 + (y - \hat{y})^2,$$

where $\hat{x}$ and $\hat{y}$ denote the time-dependent peak coordinates

$$\hat{x}(t) = x_0 \cos t - y_0 \sin t, \qquad \hat{y}(t) = -x_0 \sin t + y_0 \cos t.$$

The peaks of the approximate solution may certainly deviate from this destination. Their actual position can be calculated as the mathematical expectation of the center of mass, whereby the probability density u_h is obtained by solving equation (92) by the numerical scheme to be evaluated

$$\hat{x}_h(t) = \int_\Omega x u_h(x, y, t)\, \mathrm{d}\mathbf{x}, \qquad \hat{y}_h(t) = \int_\Omega y u_h(x, y, t)\, \mathrm{d}\mathbf{x}.$$

The quality of approximation depends on the standard deviation

$$\sigma_h^2(t) = \int_\Omega r_h^2 u_h(x, y, t)\, \mathrm{d}\mathbf{x}, \qquad r_h^2 = (x - \hat{x}_h)^2 + (y - \hat{y}_h)^2$$

which quantifies the rate of smearing caused by both physical and numerical diffusion. Due to various discretization errors, σ_h^2 may differ from the exact value $\sigma^2 = 4\epsilon t$. The discrepancy is represented by the relative error

$$\Delta\sigma_{\mathrm{rel}} = \frac{\sigma_h^2 - \sigma^2}{\sigma^2} = \frac{\sigma_h^2}{4\epsilon t} - 1.$$

This statistical quantity provides an excellent estimate of numerical diffusion inherent to the finite element scheme under consideration.

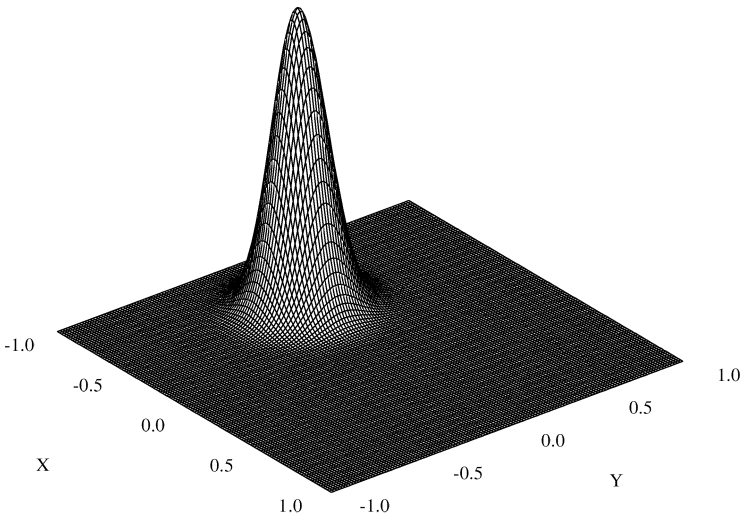


Fig. 13. Snapshot of the Gaussian hill at $t = 2.5\,\pi$

Let us start with the analytical solution corresponding to $x_0 = 0$, $y_0 = 0.5$, and $t_0 = 0.5\pi$. As the Gaussian hill moves around the origin, it is being gradually smeared by diffusion and the height of the peak decreases accordingly. A snapshot taken after one full revolution ($t = 2.5\pi$) is displayed in Fig. 13. In the 'picture norm', the numerical results produced by our FEM-TVD and FEM-FCT schemes on a Cartesian mesh (using the same number of elements and time step as before) are virtually indistinguishable from the exact solution. However, a detailed analysis reveals that the value of the global maximum differs from case to case and so does the numerical variance σ_h.

The knowledge of the exact variance σ enables us to assess the total amount of numerical diffusion for different limiting techniques and to estimate the share of the temporal error. To this end, the values of $\Delta\sigma_{\mathrm{rel}}$ are plotted versus the time step in Fig. 14. If the first-order accurate backward Euler method is employed, the temporal part of the relative variance error dominates at large time steps and decreases linearly as Δt is refined. For the second-order accurate Crank-Nicolson scheme, the magnitude of the (anti-)diffusive error is largely invariant to the time step. This is why the corresponding lines are almost horizontal. In this case, the accuracy of the space discretization is the critical factor so that the choice of the flux limiter plays a key role.

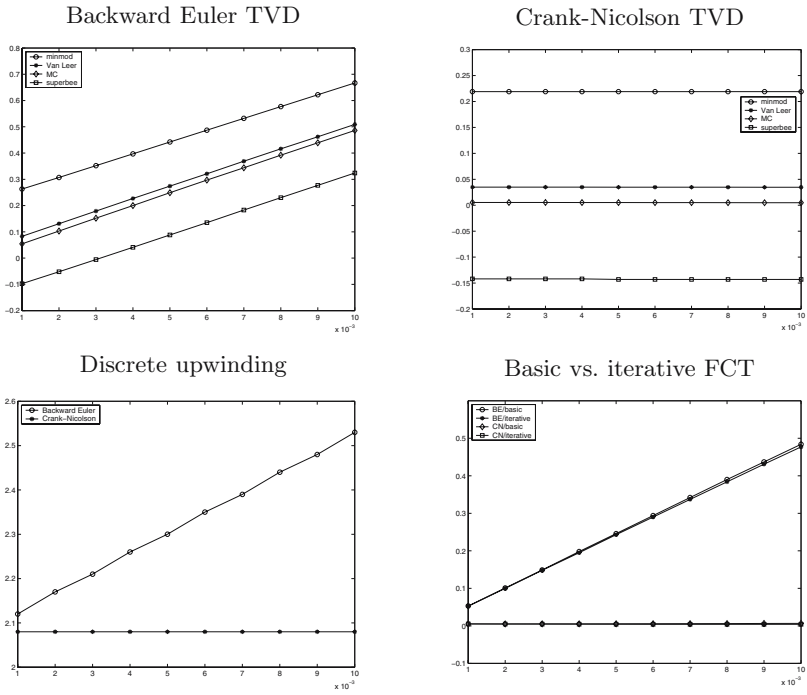


Fig. 14. Gaussian hill: relative variance error vs. the time step

By far the most diffusive solutions are produced by discrete upwinding, whereas the use of algebraic flux correction leads to a dramatic improvement. However, a comparison of the relative variance errors for standard TVD limiters reveals considerable differences in their performance. The most diffusive limiter is minmod followed by Van Leer. The MC limiter outperforms both of them and is more suitable for the treatment of smooth profiles than Roe's superbee limiter. The latter turns out to be slightly underdiffusive so that $\Delta\sigma_{\text{rel}}$ is negative if the spatial discretization error prevails. Although all four limiters qualify for CFD simulations, the 'right' one may be as difficult to select as the free parameter for classical artificial viscosity methods.

As expected, the iterative FCT algorithm is able to accommodate more antidiffusion than the basic limiter but the gain of accuracy is marginal in the range of time steps considered in this example. Note that the curves for the two versions almost meet at $\Delta t = 10^{-2}$ but begin to separate as Δt approaches 10^{-3}. This demonstrates that the Courant number must be 'large' for significant advantages to accrue from the iterative strategy. Its potential can be utilized to the full extent only if the time derivative is relatively small and the temporal accuracy can be sacrificed in favor of efficiency.

7.4 Stationary Convection-Diffusion

Algebraic flux correction schemes can be applied to stationary problems directly (TVD only) or in conjunction with a pseudo-time-stepping technique. In the latter case, the steady-state solution is obtained by marching into the stationary limit of the associated time-dependent problem, whereby the evolution details are immaterial. In essence, the time step represents an artificial parameter for the iterative solver and should be chosen as large as possible to reduce the computational cost. The CFL condition prevents explicit schemes from operating with large time steps, which makes them rather inefficient for such applications. This can be rectified to some extent by resorting to local time-stepping but an implicit time discretization is preferable.

In light of the above, the backward Euler method, which is not to be recommended for transient problems, lends itself to the treatment of steady and creeping flows. Let us apply the fully implicit FCT and TVD schemes to the stationary convection-diffusion equation

$$\mathbf{v} \cdot \nabla u - \epsilon \Delta u = 0 \qquad \text{in } \Omega = (0,1) \times (0,1),$$

where $\mathbf{v} = (\cos 10^{\text{o}}, \sin 10^{\text{o}})$ is the constant velocity and $\epsilon = 10^{-3}$ is the diffusion coefficient. The concomitant boundary conditions read

$$\frac{\partial u}{\partial y}(x,1) = 0, \qquad u(0,y) = \begin{cases} 1 & \text{if } y \geq 0.5, \\ 0 & \text{otherwise,} \end{cases}$$

$$u(x,0) = 0, \qquad u(1,y) = 0.$$

The solution to this singularly perturbed elliptic problem is characterized by the presence of a sharp front next to the line $x = 1$. The boundary layer develops because the solution of the reduced problem ($\epsilon = 0$) does not satisfy the homogeneous Dirichlet boundary condition.

A reasonable initial guess for the desired stationary solution is given by

$$u(x, y, 0) = \begin{cases} 1 - x & \text{if } y \geq 0.5, \\ 0 & \text{otherwise.} \end{cases}$$

It is worthwhile to start with discrete upwinding and use the converged low-order solution as initial data for the nonlinear high-resolution scheme. Even this crude approximation provides a good starting point, so that the extra cost due to the assembly and limiting of antidiffusive fluxes is insignificant.

The numerical solutions depicted in Fig. 15 show that the backward Euler FEM-TVD method is capable of producing nonoscillatory solutions with a sharp resolution of steep fronts and boundary layers. The upper diagram was computed on a grid of 64×64 bilinear elements using the MC limiter. The mesh employed for the lower one consists of as few as 480 elements and is refined in regions where the solution gradients are large. Due to the mesh refinement and a special alignment of the grid lines, the accuracy is comparable to that achieved on the uniform mesh at a much higher computational cost.

The results produced by the FEM-FCT algorithm are presented in Fig. 16. The time step $\Delta t = 1.0$ (Courant number $\nu = 64$) was intentionally chosen to be very large so as to expose the differences between the basic and iterative versions. The former (see the upper diagram) exhibits excessive smearing in the vicinity of the boundary layer, which compromises the benefits offered by the unconditionally stable backward Euler time-stepping. The lower diagram demonstrates that iterative flux correction is free of this drawback.

7.5 Convection in Space-Time

Our last example deals with the pure convection equation (23) discretized by central differences in conjunction with the leapfrog time-stepping

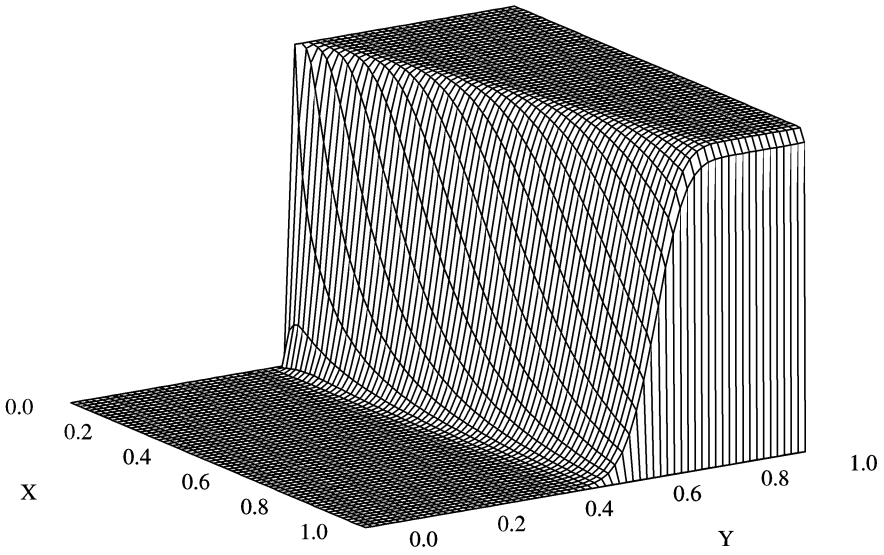
$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} + v \frac{u_{i+1}^n - u_{i+1}^n}{2\Delta x} = 0.$$

At the boundaries of the space-time domain $\Omega = (0, 1) \times (0, 0.5)$ we switch to a (first-order accurate) one-sided finite difference approximation.

Instead of advancing the numerical solution in time step-by-step as usual, let us write all equations in the matrix form $Ku = 0$ and apply our algebraic TVD method to this linear high-order system. In essence, equation (23) is treated as its two-dimensional counterpart (1) with $\mathbf{x} = (x, t)$ and $\mathbf{v} = (1, 1)$. The following initial/boundary conditions are imposed at the 'inlet'

$$u(0, t) = 0, \quad u(x, 0) = \begin{cases} 1 & \text{if } (0.1 \leq x \leq 0.2) \vee (0.3 \leq x \leq 0.4), \\ 0 & \text{otherwise.} \end{cases}$$

64×64 Q_1 elements



20×24 Q_1 elements

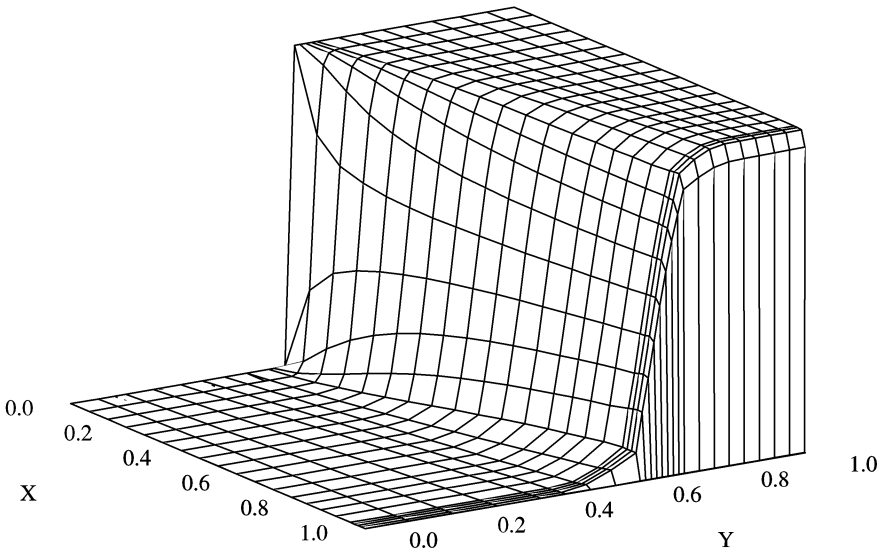


Fig. 15. Stationary solutions produced by the FEM-TVD scheme

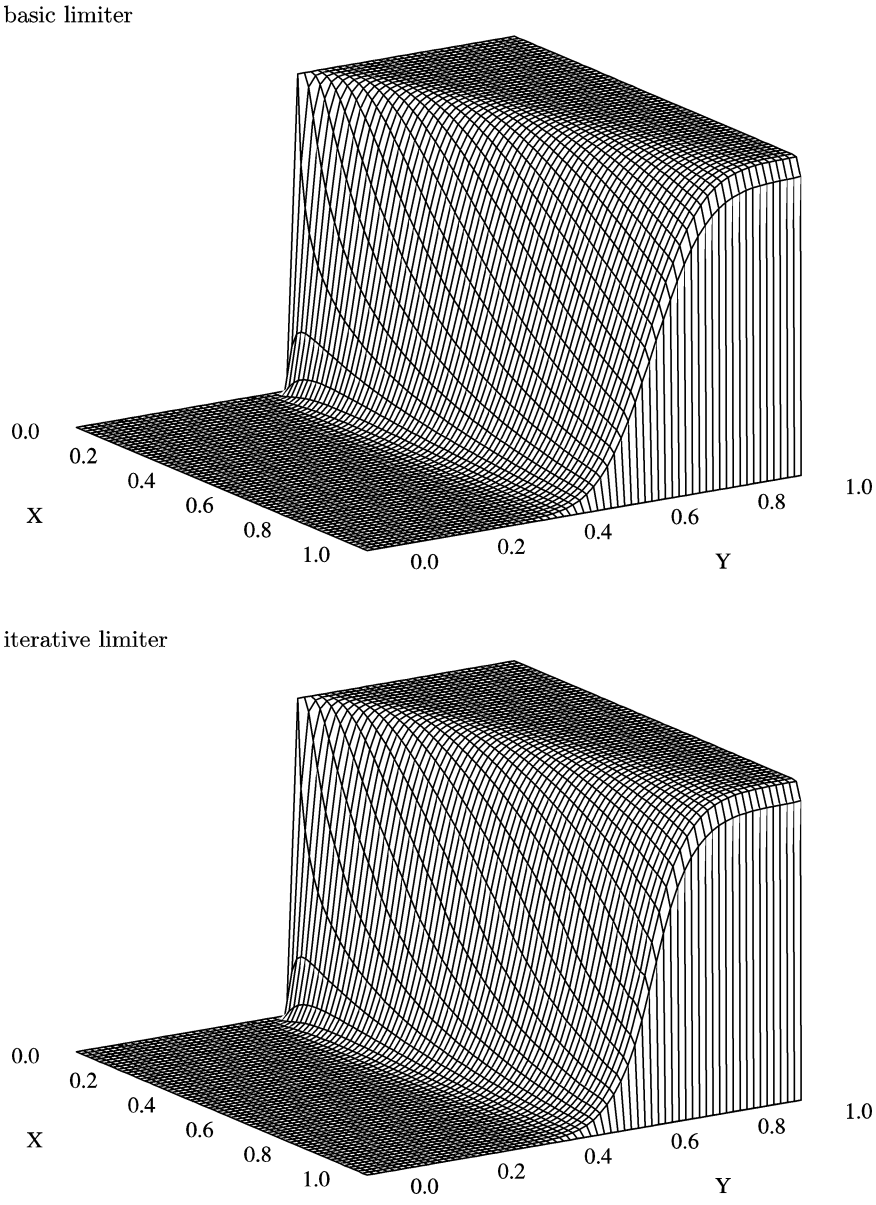


Fig. 16. Stationary solutions produced by the FEM-FCT schemes

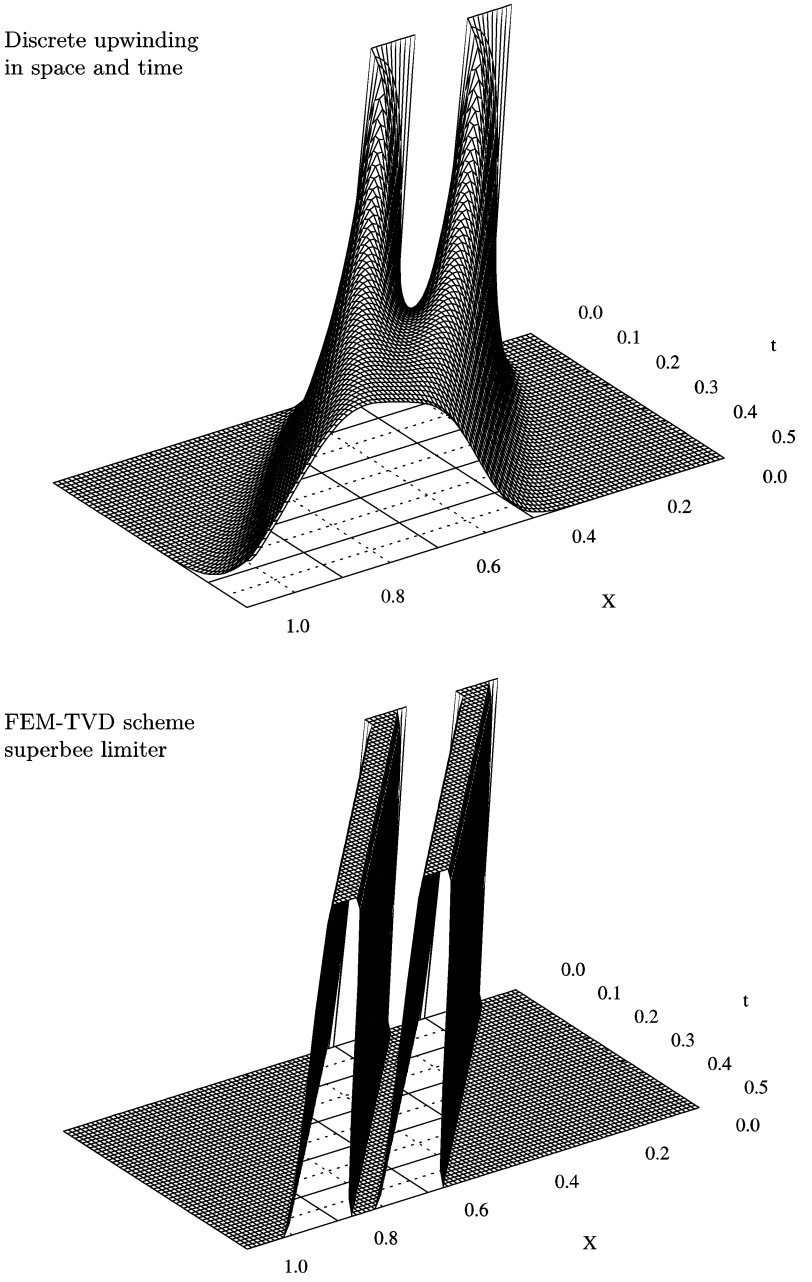


Fig. 17. Convection in the space-time domain $\Omega = (0,1) \times (0,0.5)$

Discrete upwinding applied to the matrix K yields the linear low-order method which corresponds to the upwind difference approximation of spatial derivatives and the backward Euler time-stepping. The computational results obtained on a uniform space-time mesh with $\Delta x = \Delta t = 10^{-2}$ are displayed in Fig. 17 (top). The perspective chosen for visualization is such that the solution of equation (23) at time $t = 0.5$ appears in the foreground. Ideally, it should be a copy of the discontinuous initial data shifted by $vt = 0.5$ along the $x-$axis. However, it can be seen that numerical diffusion rapidly destroys the two rectangular pulses and fills the narrow gap between them.

Using this overly diffusive solution as initial guess for the iterative defect correction scheme, we add compensating antidiffusion controlled by the superbee limiter. The resulting FEM-TVD solution is shown in Fig. 17 (bottom). It preserves the initial profiles very well and is devoid of spurious oscillations in spite of the fact that the Courant number $\nu = v\Delta t/\Delta x = 1$ for this simulation. Thus, it is possible to circumvent the restrictive CFL-like condition (18) and construct nonlinear space-time discretizations of high order.

8 Conclusions and Outlook

A new class of high-resolution finite element schemes was presented. Node-oriented flux limiters of FCT and TVD type were applied at the algebraic level so as to render the underlying Galerkin discretization local extremum diminishing and positivity-preserving. The proposed methodology is very flexible and can be readily integrated into existing CFD software as a modular extension to the matrix assembly routine. Remarkably, algebraic flux correction is applicable to 'arbitrary' discretizations in space and time (explicit and implicit time-stepping, finite elements/differences/volumes, Cartesian and unstructured meshes) and portable to higher dimensions. In fact, the same 'post-processing' routine can be used in 1D, 2D, and 3D implementations.

The generality of our algebraic approach makes it a valuable design tool and suggests many directions for further research. In particular, an extension to higher-order finite elements is feasible but nontrivial. For superlinear approximations, even the mass matrix and the discrete Laplacian operator may have negative off-diagonal coefficients, and it is unclear whether or not they should be eliminated in the course of discrete upwinding. It might be worthwhile to introduce some stabilizing background diffusion so as to reduce phase errors and alleviate terracing. For instance, the fourth-order accurate CNTG scheme [11], which represents a generalization of the method used in the 'reversible' FCT algorithm [5], would be a good candidate for the linear high-order scheme. Furthermore, it would be interesting to investigate how algebraic flux correction performs in the realm of discontinuous Galerkin methods. Last but not least, there is a need for the development of robust and efficient iterative solvers for nonsymmetric algebraic systems that result from an implicit LED discretization of the troublesome convective terms.

A. Galerkin Flux Decomposition

In this appendix, we present the derivation of numerical fluxes for the finite element discretization of the generic conservation law

$$\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{f} = 0 \qquad \text{in } \Omega.$$

The flux function $\mathbf{f}$ may depend on the solution u in a nonlinear way.

Using the divergence theorem to perform integration by parts in the weak form of this equation, we obtain the integral relation

$$\int_\Omega w \frac{\partial u}{\partial t}\,\mathrm{d}\mathbf{x} - \int_\Omega \nabla w \cdot \mathbf{f}\,\mathrm{d}\mathbf{x} + \int_\Gamma w\ \mathbf{f}\cdot\mathbf{n}\,\mathrm{d}s = 0, \qquad \forall w.$$

The approximate solution to this variational problem is sought in the form

$$u_h(\mathbf{x},t) = \sum_j u_j(t)\varphi_j(\mathbf{x}).$$

Consider the group finite element formulation [13] which consists in using the same interpolation for the flux function

$$\mathbf{f}_h(\mathbf{x},t) = \sum_j \mathbf{f}_j(t)\,\varphi_j(\mathbf{x}).$$

Rendering the residual orthogonal to each basis function φ_i of the *ansatz* space gives a system of semi-discretized equations for the nodal values

$$\sum_j \left[\int_\Omega \varphi_i\varphi_j\,\mathrm{d}\mathbf{x}\right]\frac{\mathrm{d}u_j}{\mathrm{d}t} - \sum_j \left[\int_\Omega \nabla\varphi_i\varphi_j\,\mathrm{d}\mathbf{x} - \int_\Gamma \varphi_i\varphi_j\,\mathbf{n}\,\mathrm{d}s\right]\cdot\mathbf{f}_j = 0.$$

The integrals containing the product of basis functions represent entries of the mass matrices for the volume and surface triangulation

$$m_{ij} = \int_\Omega \varphi_i\varphi_j\,\mathrm{d}\mathbf{x}, \qquad \mathbf{s}_{ij} = \int_\Gamma \varphi_i\varphi_j\,\mathbf{n}\,\mathrm{d}s.$$

Furthermore, the volume integrals that result from the discretization of spatial derivatives can be written in the notation of section 2 as follows

$$\left[\int_\Omega \nabla\varphi_i\varphi_j\,\mathrm{d}\mathbf{x}\right]\cdot\mathbf{f}_j = \mathbf{c}_{ji}\cdot\mathbf{f}_j, \qquad \mathbf{c}_{ij} = \int_\Omega \varphi_i\nabla\varphi_j\,\mathrm{d}\mathbf{x}.$$

Recall that the coefficient matrix $\{\mathbf{c}_{ij}\}$ has zero row sums, which enables us to express its diagonal entries in terms of the off-diagonal ones

$$\sum_j \mathbf{c}_{ij} = 0 \quad \Rightarrow \quad \mathbf{c}_{ii} = -\sum_{j\neq i}\mathbf{c}_{ij}.$$

It follows that the ODEs at hand can be cast into the conservation form

$$\sum_j \left[m_{ij} \frac{\mathrm{d}u_j}{\mathrm{d}t} + \mathbf{s}_{ij} \cdot \mathbf{f}_j \right] + \sum_{j \neq i} g_{ij} = 0,$$

where the interior part of the discretized divergence term is assembled from the *Galerkin fluxes* associated with edges of the sparsity graph

$$g_{ij} = \mathbf{c}_{ij} \cdot \mathbf{f}_i - \mathbf{c}_{ji} \cdot \mathbf{f}_j, \qquad g_{ji} = -g_{ij}.$$

These antisymmetric fluxes are responsible for the bilateral mass exchange between two neighboring nodes, whereby no mass is created or destroyed artificially in the interior of the domain. The total amount of u may only change due to the external feed $\mathbf{s}_{ij} \cdot \mathbf{f}_j$ which is equal to zero unless the basis functions for both nodes are nonvanishing on the boundary.

The above flux decomposition makes it possible to extend the wealth of upwinding techniques and slope limiters available for the data structure of Peraire *et al.* [45] to arbitrary Galerkin discretizations. The interested reader is referred to [33],[37],[39] for the mathematical and algorithmic background of such high-resolution finite element schemes that were originally developed for piecewise-linear approximations on triangular meshes.

References

1. P. Arminjon and A. Dervieux, Construction of TVD-like artificial viscosities on 2-dimensional arbitrary FEM grids. *INRIA Research Report* **1111** (1989).
2. K. Baba and M. Tabata, On a conservative upwind finite element scheme for convective diffusion equations. *RAIRO Numerical Analysis* **15** (1981) 3–25.
3. T. J. Barth, Numerical aspects of computing viscous high Reynolds number flows on unstructured meshes. Technical report 91-0721, *AIAA paper*, 1991.
4. T. J. Barth, Aspects of unstructured grids and finite volume solvers for the Euler and Navier-Stokes equations. In *von Karman Institute for Fluid Dynamics Lecture Series* Notes 1994-05, Brussels, 1994.
5. D. L. Book, *The Conception, Gestation, Birth, and Infancy of FCT*. Chapter 1 in this volume.
6. J. P. Boris and D. L. Book, Flux-corrected transport. I. SHASTA, A fluid transport algorithm that works. *J. Comput. Phys.* **11** (1973) 38–69.
7. A. N. Brooks and T. J. R. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.* **32** (1982) 199-259.
8. G. F. Carey and B. N. Jiang, Least-squares finite elements for first-order hyperbolic systems. *Int. J. Numer. Meth. Fluids* **26** (1988) 81–93.
9. C. R. DeVore, An improved limiter for multidimensional flux-corrected transport. *NASA Technical Report* AD-A360122 (1998).
10. J. Donea, L. Quartapelle and V. Selmin, An analysis of time discretization in the finite element solution of hyperbolic problems. *J. Comput. Phys.* **70** (1987) 463–499.
11. J. Donea, V. Selmin and L. Quartapelle, Recent developments of the Taylor-Galerkin method for the numerical solution of hyperbolic problems. *Numerical methods for fluid dynamics III*, Oxford, 171-185 (1988).
12. J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*. Springer, 1996.
13. C. A. J. Fletcher, The group finite element formulation. *Comput. Methods Appl. Mech. Engrg.* **37** (1983) 225-243.
14. S. K. Godunov, Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. *Mat. Sbornik* **47** (1959) 271-306.
15. P. Hansbo, Aspects of conservation in finite element flow computations. *Comput. Methods Appl. Mech. Engrg.* **117** (1994) 423-437.
16. A. Harten, High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.* **49** (1983) 357–393.
17. A. Harten, On a class of high resolution total-variation-stable finite-difference-schemes. *SIAM J. Numer. Anal.* **21** (1984) 1-23.
18. C. Hirsch, *Numerical Computation of Internal and External Flows. Vol. II: Computational Methods for Inviscid and Viscous Flows*. John Wiley & Sons, Chichester, 1990.
19. A. Jameson, Analysis and design of numerical schemes for gas dynamics 1. Artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence. *Int. Journal of CFD* **4** (1995) 171-218.
20. A. Jameson, Computational algorithms for aerodynamic analysis and design. *Appl. Numer. Math.* **13** (1993) 383-422.

21. A. Jameson, Positive schemes and shock modelling for compressible flows. *Int. J. Numer. Meth. Fluids* **20** (1995) 743–776.
22. D. Kuzmin, Positive finite element schemes based on the flux-corrected transport procedure. In: K. J. Bathe (ed.), *Computational Fluid and Solid Mechanics*, Elsevier, 887-888 (2001).
23. D. Kuzmin and S. Turek, Flux correction tools for finite elements. *J. Comput. Phys.* **175** (2002) 525-558.
24. D. Kuzmin and S. Turek, Explicit and implicit high-resolution finite element schemes based on the Flux-Corrected-Transport algorithm. In: F. Brezzi et al. (eds), Proceedings of the 4th European Conference on Numerical Mathematics and Advanced Applications, Springer-Verlag Italy, 2002, 133-143.
25. D. Kuzmin, M. Möller and S. Turek, Multidimensional FEM-FCT schemes for arbitrary time-stepping. *Int. J. Numer. Meth. Fluids* **42** (2003) 265-295.
26. D. Kuzmin and S. Turek, Finite element discretization tools for gas-liquid flows. In: M. Sommerfeld (ed.), *Bubbly Flows: Analysis, Modelling and Calculation*, Springer, 2004, 191-201.
27. D. Kuzmin and S. Turek, High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter. *J. Comput. Phys.* **198** (2004) 131-158.
28. D. Kuzmin, M. Möller and S. Turek, High-resolution FEM-FCT schemes for multidimensional conservation laws. Technical report **231**, University of Dortmund, 2003. To appear in *Comput. Methods Appl. Mech. Engrg.*
29. A. Lapin, University of Stuttgart. Private communication.
30. P. D. Lax, *Systems of Conservation Laws and Mathematical Theory of Shock Waves*. SIAM Publications, Philadelphia, 1973.
31. R. J. LeVeque, *Numerical Methods for Conservation Laws*. Birkhäuser, 1992.
32. R. J. LeVeque, High-resolution conservative algorithms for advection in incompressible flow. *Siam J. Numer. Anal.* **33** (1996) 627–665.
33. R. Löhner, *Applied CFD Techniques: An Introduction Based on Finite Element Methods*. Wiley, 2001.
34. R. Löhner and J. D. Baum, *30 Years of FCT: Status and Directions.* Chapter 5 in this volume.
35. R. Löhner, K. Morgan, J. Peraire and M. Vahdati, Finite element flux-corrected transport (FEM-FCT) for the Euler and Navier-Stokes equations. *Int. J. Numer. Meth. Fluids* **7** (1987) 1093–1109.
36. R. Löhner, K. Morgan, M. Vahdati, J. P. Boris and D. L. Book, FEM-FCT: combining unstructured grids with high resolution. *Commun. Appl. Numer. Methods* **4** (1988) 717–729.
37. P. R. M. Lyra, *Unstructured Grid Adaptive Algorithms for Fluid Dynamics and Heat Conduction*. PhD thesis, University of Wales, Swansea, 1994.
38. P. R. M. Lyra, K. Morgan, J. Peraire and J. Peiro, TVD algorithms for the solution of the compressible Euler equations on unstructured meshes. *Int. J. Numer. Meth. Fluids* **19** (1994) 827–847.
39. P. R. M. Lyra and K. Morgan, A review and comparative study of upwind biased schemes for compressible flow computation. III: Multidimensional extension on unstructured grids. *Arch. Comput. Methods Eng.* **9** (2002) no. 3, 207-256.
40. M. Möller, *Hochauflösende FEM-FCT-Verfahren zur Diskretisierung von konvektionsdominanten Transportproblemen mit Anwendung auf die kompressiblen Eulergleichungen*. Diploma thesis, University of Dortmund, 2003.

41. M. Möller, D. Kuzmin and S. Turek, Implicit flux-corrected transport algorithm for finite element simulation of the compressible Euler equations. In: M. Křížek *et al.* (eds), *Conjugate Gradient Algorithms and Finite Element Methods*, Springer, 2004, 325-354.
42. K. Morgan and J. Peraire, Unstructured grid finite element methods for fluid mechanics. *Reports on Progress in Physics*, **61** (1998), no. 6, 569-638.
43. E. S. Oran and J. P. Boris, *Numerical Simulation of Reactive Flow.* 2nd edition, Cambridge University Press, 2001.
44. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow.* McGraw-Hill, New York, 1980.
45. J. Peraire, M. Vahdati, J. Peiro and K. Morgan, The construction and behaviour of some unstructured grid algorithms for compressible flows. *Numerical Methods for Fluid Dynamics* **IV**, Oxford University Press, 221-239 (1993).
46. R. Rannacher and S. Turek, A simple nonconforming quadrilateral Stokes element. *Numer. Meth. PDEs* **8** (1992), no. 2, 97-111.
47. C. Schär and P. K. Smolarkiewicz, A synchronous and iterative flux-correction formalism for coupled transport equations. *J. Comput. Phys.* **128** (1996) 101–120.
48. V. Selmin, Finite element solution of hyperbolic equations. I. One-dimensional case. *INRIA Research Report* **655** (1987).
49. V. Selmin, Finite element solution of hyperbolic equations. II. Two-dimensional case. *INRIA Research Report* **708** (1987).
50. A. Sokolichin, *Mathematische Modellbildung und numerische Simulation von Gas-Flüssigkeits-Blasenströmungen.* Habilitation thesis, University of Stuttgart, 2004.
51. P. K. Sweby, High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM J. Numer. Anal.* **21** (1984), 995–1011.
52. S. Turek, *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*, LNCSE 6, Springer, 1999.
53. S. Turek and D. Kuzmin, *Algebraic Flux Correction III. Incompressible Flow Problems.* Chapter 8 in this volume.
54. S. T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.* **31** (1979) 335–362.

Algebraic Flux Correction II. Compressible Euler Equations

Dmitri Kuzmin[1] and Matthias Möller[2]

[1] Institute of Applied Mathematics (LS III), University of Dortmund
Vogelpothsweg 87, D-44227, Dortmund, Germany
kuzmin@math.uni-dortmund.de
[2] matthias.moeller@math.uni-dortmund.de

Summary. Algebraic flux correction schemes of TVD and FCT type are extended to systems of hyperbolic conservation laws. The group finite element formulation is employed for the treatment of the compressible Euler equations. An efficient algorithm is proposed for the edge-by-edge matrix assembly. A generalization of Roe's approximate Riemann solver is derived by rendering all off-diagonal matrix blocks positive semi-definite. Another usable low-order method is constructed by adding scalar artificial viscosity proportional to the spectral radius of the cumulative Roe matrix. The limiting of antidiffusive fluxes is performed using a transformation to the characteristic variables or a suitable synchronization of correction factors for the conservative ones. The outer defect correction loop is equipped with a block-diagonal preconditioner so as to decouple the discretized Euler equations and solve them in a segregated fashion. As an alternative, a strongly coupled solution strategy (global BiCGSTAB method with a block-Gauß-Seidel preconditioner) is introduced for applications which call for the use of large time steps. Various algorithmic aspects including the implementation of characteristic boundary conditions are addressed. Simulation results are presented for inviscid flows in a wide range of Mach numbers.

1 Introduction

Unstructured grid finite element methods appear to be particularly attractive for the treatment of aerodynamic applications governed by the compressible Euler equations [25],[27],[34]. However, most of the algorithms currently in use are explicit and, consequently, subject to the CFL condition which becomes very restrictive in the presence of adaptive mesh refinement. For steady-state flows the computational cost can be drastically reduced by using *local time-stepping* to achieve a (more) uniform distribution of Courant numbers in the domain. For transient flows this approach is not feasible since using different time steps at different mesh points may result in a loss of 'mass' and allow shocks to move at wrong speed. Hence, there is a need for the development of truly *implicit* high-resolution finite element schemes which are unconditionally

stable and much more efficient than explicit ones for the above-mentioned class of CFD applications. At the same time, their implementation is more difficult and the actual performance strongly depends on the choice of data structures, configuration of linear solvers, efficiency of the matrix assembly, and other algorithmic details. In this chapter, we extend the implicit FEM-FCT and FEM-TVD schemes to systems of hyperbolic conservation laws, discuss the subtleties of algebraic flux correction for the Euler equations and present some iterative solution strategies for the resulting algebraic systems.

2 Compressible Euler Equations

The Euler equations of gas dynamics represent a system of conservation laws for the mass, momentum, and energy of an inviscid compressible fluid

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \tag{1}$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v}) + \nabla p = 0, \tag{2}$$

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\rho H \mathbf{v}) = 0, \tag{3}$$

where ρ, $\mathbf{v}$, p, E and $H = E + p/\rho$ are the density, velocity, pressure, total energy per unit mass, and stagnation enthalpy, respectively. This first-order PDE system is a simplification of the more realistic *compressible Navier-Stokes equations* which include the effects of viscosity and heat conduction.

The internal energy is assumed to be a known function of pressure and density. For an ideal polytropic gas, we have the following equation of state

$$p = (\gamma - 1)\rho \left(E - \frac{|\mathbf{v}|^2}{2} \right). \tag{4}$$

In this relation, the constant γ is the ratio of specific heats ($\gamma = 1.4$ for air).

Introducing the vector of conservative variables U and the triple of fluxes $\mathbf{F} = (F^1, F^2, F^3)$ for each coordinate direction in the Euclidean space $\mathbb{R}^3$

$$U = \begin{bmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \end{bmatrix}, \qquad \mathbf{F} = \begin{bmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \otimes \mathbf{v} + p\mathcal{I} \\ \rho H \mathbf{v} \end{bmatrix}, \tag{5}$$

where $\mathcal{I}$ stands for the identity tensor, we can represent the (three-dimensional) Euler equations in the standard divergence form as follows

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F} = 0, \qquad \text{where} \quad \nabla \cdot \mathbf{F} = \sum_{d=1}^{3} \frac{\partial F^d}{\partial x_d}. \tag{6}$$

The chain rule yields an equivalent quasi-linear formulation in which the spatial derivatives are applied to the conservative variables rather than fluxes

$$\frac{\partial U}{\partial t} + \mathbf{A} \cdot \nabla U = 0, \qquad \text{where} \quad \mathbf{A} \cdot \nabla U = \sum_{d=1}^{3} A^d \frac{\partial U}{\partial x_d}. \tag{7}$$

Here $\mathbf{A} = (A^1, A^2, A^3)$ denotes the triple of Jacobian matrices such that [27]

$$F^d = A^d U, \qquad A^d = \frac{\partial F^d}{\partial U}, \qquad d = 1, 2, 3. \tag{8}$$

The first relation in this formula means that each flux component is a homogeneous function of the conservative variables [44].

Due to the hyperbolicity of the Euler equations, any linear combination of the three Jacobians is diagonalizable with real eigenvalues. In other words, the inner product of $\mathbf{A}$ with an arbitrary vector $\mathbf{e} = (e_1, e_2, e_3)$ of the 3D space admits the following factorization [17],[22],[44]

$$\mathbf{e} \cdot \mathbf{A} = \sum_{d=1}^{3} e_d A^d = R \Lambda(\mathbf{e}) R^{-1}, \tag{9}$$

where $\Lambda(\mathbf{e})$ is the diagonal matrix of eigenvalues and R is the matrix of right eigenvectors whereas R^{-1} is composed from the left ones. Analytical expressions for these matrices can be found, for instance, in [37]. At the same time, it is impossible to diagonalize the unidirectional Jacobians A^d simultaneously as they do not commute and, therefore, have different sets of eigenvalues [17]. This information on the spectral properties of $\mathbf{A}$ is important for the design of numerical methods and will be utilized in what follows.

3 High-Order Scheme

The first algorithm to be presented is the basic Galerkin scheme to be endowed with our algebraic flux limiters of FCT and TVD type. As already mentioned in the Introduction, *implicit* finite element methods are still rarely used in compressible flow simulations. Therefore, matrix assembly for the Euler equations has received little attention in the literature. In this section, we show how it can be implemented in an efficient way building on Roe's linearization technique [36] for systems of hyperbolic conservation laws.

Let us start with the divergence form (6) of the Euler equations and discretize them in space leaving the time derivative continuous for the time being. As in the scalar case, we adopt the group finite element formulation. That is, both the conservative variables and the flux function are approximated in terms of their nodal values multiplied by the FEM basis functions:

$$U_h(\mathbf{x}, t) = \sum_j \mathrm{U}_j(t) \varphi_j(\mathbf{x}), \qquad \mathbf{F}_h(\mathbf{x}, t) = \sum_j \mathbf{F}_j(t) \varphi_j(\mathbf{x}). \tag{10}$$

The use of these approximations in the weak form of (6) leads to a system of semi-discretized equations for the time-dependent nodal values

$$\sum_j \left[\int_\Omega \varphi_i \varphi_j \, \mathrm{d}\mathbf{x}\right] \frac{\mathrm{d}\mathrm{U}_j}{\mathrm{d}t} + \sum_j \left[\int_\Omega \varphi_i \nabla\varphi_j \, \mathrm{d}\mathbf{x}\right] \cdot \mathbf{F}_j = 0. \tag{11}$$

In order to solve it numerically, we need to eliminate the dependent variables $\mathbf{F}_j$ in favor of the unknowns U_j so as to obtain an ODE system of the form

$$M_C \frac{\mathrm{d}\mathrm{U}}{\mathrm{d}t} = K\mathrm{U}, \tag{12}$$

where M_C is the block-diagonal mass matrix and K is a discrete counterpart of the operator $-\mathbf{A}\cdot\nabla$ for the quasi-linear formulation (7). Of course, it depends on the solution vector U since the governing equations are nonlinear.

Recall that the basis functions φ_j sum to unity, so that the sum of their derivatives vanishes at every point. Therefore, the auxiliary coefficients $\mathbf{c}_{ij}$ as defined in equation (6) of the previous chapter have zero row sums:

$$\mathbf{c}_{ij} = \int_\Omega \varphi_i \nabla\varphi_j \, \mathrm{d}\mathbf{x}, \qquad \mathbf{c}_{ii} = -\sum_{j\neq i} \mathbf{c}_{ij}. \tag{13}$$

It follows from (11) that the right-hand side of the five coupled equations for node i can be expressed as a sum of edge contributions

$$(K\mathrm{U})_i = -\sum_j \mathbf{c}_{ij}\cdot\mathbf{F}_j = -\sum_{j\neq i} \mathbf{c}_{ij}\cdot(\mathbf{F}_j - \mathbf{F}_i). \tag{14}$$

This sort of representation is typical of finite volume/difference schemes, which enables us to borrow many useful ideas originally developed in this context.

In his pioneering work on approximate Riemann solvers [36], Roe showed that the differences between the components of $\mathbf{F}$ and U are related by

$$\mathbf{F}_j - \mathbf{F}_i = \hat{\mathbf{A}}_{ij}(\mathrm{U}_j - \mathrm{U}_i), \tag{15}$$

where the triple of matrices $\hat{\mathbf{A}}_{ij} = (\hat{\mathrm{A}}^1_{ij}, \hat{\mathrm{A}}^2_{ij}, \hat{\mathrm{A}}^3_{ij})$ corresponds to the Jacobian tensor $\mathbf{A}$ evaluated for a special set of *density-averaged* variables

$$\hat{\rho}_{ij} = \sqrt{\rho_i\rho_j}, \quad \hat{\mathbf{v}}_{ij} = \frac{\sqrt{\rho_i}\mathbf{v}_i + \sqrt{\rho_j}\mathbf{v}_j}{\sqrt{\rho_i}+\sqrt{\rho_j}}, \quad \hat{H}_{ij} = \frac{\sqrt{\rho_i}H_i + \sqrt{\rho_j}H_j}{\sqrt{\rho_i}+\sqrt{\rho_j}} \tag{16}$$

which are called the Roe mean values. The speed of sound $\hat{c}_{ij}$ for the intermediate state $(\hat{\rho}_{ij}, \hat{\mathbf{v}}_{ij}, \hat{H}_{ij})$ is determined as follows:

$$\hat{c}_{ij} = \sqrt{(\gamma-1)\left(\hat{H}_{ij} - \frac{|\hat{\mathbf{v}}_{ij}|^2}{2}\right)}. \tag{17}$$

By virtue of this linearization, the right-hand side of (12) can be assembled from individual edge contributions which depend on the difference between the values of the conservative variables/fluxes at nodes i and j

$$(K\mathrm{U})_i \quad \longleftarrow \quad \mathbf{c}_{ij} \cdot (\mathbf{F}_i - \mathbf{F}_j) = \mathbf{c}_{ij} \cdot \hat{\mathbf{A}}_{ij}(\mathrm{U}_i - \mathrm{U}_j), \tag{18}$$

$$(K\mathrm{U})_j \quad \longleftarrow \quad \mathbf{c}_{ji} \cdot (\mathbf{F}_j - \mathbf{F}_i) = \mathbf{c}_{ji} \cdot \hat{\mathbf{A}}_{ij}(\mathrm{U}_j - \mathrm{U}_i). \tag{19}$$

Let us represent the linear combinations of the averaged Jacobians in terms of the *cumulative Roe matrices* A_{ij} and B_{ij} defined by

$$\mathrm{A}_{ij} = \mathbf{a}_{ij} \cdot \hat{\mathbf{A}}_{ij}, \qquad \mathbf{a}_{ij} = \frac{\mathbf{c}_{ij} - \mathbf{c}_{ji}}{2}, \tag{20}$$

$$\mathrm{B}_{ij} = \mathbf{b}_{ij} \cdot \hat{\mathbf{A}}_{ij}, \qquad \mathbf{b}_{ij} = \frac{\mathbf{c}_{ij} + \mathbf{c}_{ji}}{2}. \tag{21}$$

Integration by parts in (13) reveals that the coefficients $\mathbf{c}_{ij}$ and $\mathbf{c}_{ji}$ satisfy

$$\mathbf{c}_{ji} = -\mathbf{c}_{ij} + \int_\Gamma \varphi_i \varphi_j \, \mathbf{n} \, \mathrm{d}s, \tag{22}$$

where $\mathbf{n}$ denotes the unit outward normal to the boundary Γ. We remark that finite element basis functions have compact support. As a rule, the surface integral is equal to zero unless both nodes belong to the boundary.

By substitution of (22) into formulae (20) and (21), we obtain

$$\mathbf{a}_{ij} = \mathbf{c}_{ij} - \frac{1}{2}\mathbf{s}_{ij}, \qquad \mathbf{b}_{ij} = \frac{1}{2}\mathbf{s}_{ij}, \tag{23}$$

where $\mathbf{s}_{ij}$ is an entry of the 'mass matrix' for the surface triangulation

$$\mathbf{s}_{ij} = \int_\Gamma \varphi_i \varphi_j \, \mathbf{n} \, \mathrm{d}s.$$

In the interior of the domain, $\mathbf{s}_{ij}$ vanishes and (23) reduces to

$$\mathbf{a}_{ij} = \mathbf{c}_{ij}, \qquad \mathbf{b}_{ij} = 0.$$

Consequently, just the antisymmetric part A_{ij} of the cumulative Roe matrix is to be evaluated for each interior edge. The symmetric part B_{ij} is only needed at the boundary and can sometimes be neglected (see below).

It is easy to verify that the inner products in (18)–(19) are given by

$$\mathbf{c}_{ij} \cdot \hat{\mathbf{A}}_{ij} = \mathrm{A}_{ij} + \mathrm{B}_{ij}, \qquad \mathbf{c}_{ji} \cdot \hat{\mathbf{A}}_{ij} = -\mathrm{A}_{ij} + \mathrm{B}_{ij}. \tag{24}$$

Hence, the contribution of the edge ij to the right-hand side of (12) reads

$$(K\mathrm{U})_i \quad \longleftarrow \quad (\mathrm{A}_{ij} + \mathrm{B}_{ij})(\mathrm{U}_i - \mathrm{U}_j), \tag{25}$$

$$(K\mathrm{U})_j \quad \longleftarrow \quad (\mathrm{A}_{ij} - \mathrm{B}_{ij})(\mathrm{U}_i - \mathrm{U}_j). \tag{26}$$

This representation leads to a very efficient edge-based algorithm for matrix assembly since there is no need for numerical integration as long as the coefficients $\mathbf{c}_{ij}$ are initialized and stored. The sparsity graph of the global matrix K depends solely on the underlying mesh and on the type of approximation. For systems of equations, the list of edges is the same as in the scalar case. However, there are interactions not only between basis functions for different mesh nodes but also between those for different variables. Due to this intimate coupling of degrees of freedom, each 'coefficient' of the discrete operator turns into a matrix of size equal to the squared number of variables.

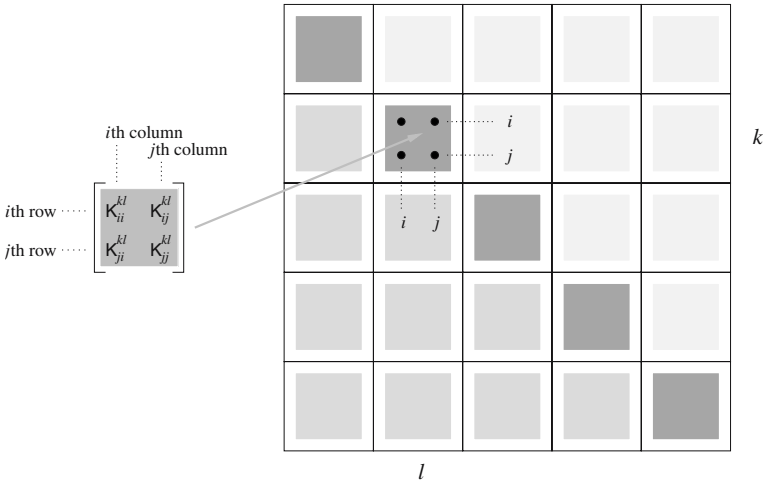


Fig. 1. Matrix assembly for the Euler equations

If the global finite element matrix (or some parts of it) rather than its product with the solution vector U needs to be assembled, this can be accomplished by evaluating the four 5×5 blocks

$$\begin{aligned} \mathrm{K}_{ii} &= \mathrm{A}_{ij} + \mathrm{B}_{ij}, \qquad & \mathrm{K}_{ij} &= -\mathrm{A}_{ij} - \mathrm{B}_{ij}, \\ \mathrm{K}_{ji} &= \mathrm{A}_{ij} - \mathrm{B}_{ij}, \qquad & \mathrm{K}_{jj} &= -\mathrm{A}_{ij} + \mathrm{B}_{ij} \end{aligned} \tag{27}$$

edge-by-edge and scattering their entries to the positions with indices i and j in the corresponding blocks of the operator K. This process is illustrated in Fig. 1. As a rule, it is not necessary to assemble and store the whole global matrix. Instead, it is possible to piece together individual edge contributions of the form (25)–(26) so as to avoid matrix-vector multiplications like $K\mathrm{U}$. Depending on the choice of smoothers/preconditioners for the iterative solver, just a few (if any) global blocks need to be generated explicitly. The block-diagonal or the upper/lower triangular part will suffice for our purposes.

4 Low-Order Scheme

To a large extent, the ability of a high-resolution scheme to withstand the formation of wiggles depends on the quality of the underlying low-order method. Let us construct it at the algebraic level following the strategy presented in the previous chapter. To this end, we perform the row-sum mass lumping and replace the original Galerkin discretization (12) by

$$M_L \frac{\mathrm{d}\mathrm{U}}{\mathrm{d}t} = L\mathrm{U}, \tag{28}$$

where M_L denotes the lumped mass matrix and L is the low-order Jacobian operator. Recall that its scalar counterpart was derived by 'discrete upwinding' which amounts to a conservative elimination of negative off-diagonal entries from the high-order transport operator. For systems of conservation laws, the edge contributions to the global matrix are no longer scalar quantities but matrices themselves. This leads to the following algebraic constraint:

LED principle for systems (semi-discrete level)

Let the system of ordinary differential equations for the values of the conservative variables at node i be represented in the form

$$m_i \frac{\mathrm{d}\mathrm{U}_i}{\mathrm{d}t} = \sum_{j \neq i} \mathrm{L}_{ij}(\mathrm{U}_j - \mathrm{U}_i). \tag{29}$$

If all off-diagonal matrix blocks L_{ij} are *positive semi-definite* (that is, their eigenvalues are nonnegative), then such a discretization is local extremum diminishing for a certain set of local *characteristic variables*. Obviously, this condition is consistent with the LED criterion for scalar conservation laws but in the case of a hyperbolic system it is much less restrictive than the requirement that all off-diagonal coefficients of L be nonnegative.

To construct a nonoscillatory low-order scheme for the compressible Euler equations, we add tensorial artificial viscosity D_{ij} and remove the symmetric part B_{ij} of the cumulative Roe matrix in (27). This gives

$$\begin{aligned} \mathrm{L}_{ii} &= \mathrm{A}_{ij} - \mathrm{D}_{ij}, \qquad & \mathrm{L}_{ij} &= -\mathrm{A}_{ij} + \mathrm{D}_{ij}, \\ \mathrm{L}_{ji} &= \mathrm{A}_{ij} + \mathrm{D}_{ij}, \qquad & \mathrm{L}_{jj} &= -\mathrm{A}_{ij} - \mathrm{D}_{ij}. \end{aligned} \tag{30}$$

These modified edge contributions are built into the blocks of L as explained above. It is worth mentioning that the high-order operator K does not need to be assembled at all. The corresponding raw antidiffusive flux reads

$$\mathrm{F}_{ij} = -\left(\mathrm{M}_{ij}\frac{\mathrm{d}}{\mathrm{d}t} + \mathrm{D}_{ij} + \mathrm{B}_{ij}\right)(\mathrm{U}_j - \mathrm{U}_i), \qquad \mathrm{F}_{ji} = -\mathrm{F}_{ij}. \tag{31}$$

The block $\mathrm{M}_{ij} = m_{ij}\mathrm{I}$, where I denotes the 5×5 identity matrix, is responsible for the error induced by mass lumping. After the time discretization, one obtains an expression similar to (71) of the preceding chapter. It remains to design the matrix D_{ij} so as to enforce the generalized LED constraint.

5 Design of Artificial Viscosities

As pointed out earlier, the system of Euler equations is hyperbolic so that any linear combination of the three Jacobian matrices is diagonalizable with real eigenvalues. In particular, there exists a diagonal matrix Λ_{ij} and a regular matrix R_{ij} of right eigenvectors such that the cumulative Roe matrix A_{ij} for the edge ij admits the following factorization

$$\mathrm{A}_{ij} = |\mathbf{a}_{ij}|\,\mathrm{R}_{ij}\Lambda_{ij}\mathrm{R}_{ij}^{-1}. \tag{32}$$

The scaling factor is given by the Euclidean norm of the coefficient vector

$$|\mathbf{a}_{ij}| = \sqrt{\mathbf{a}_{ij}\cdot\mathbf{a}_{ij}}, \qquad \mathbf{a}_{ij} = (a_{ij}^1, a_{ij}^2, a_{ij}^3)$$

and the diagonal elements of the eigenvalue matrix $\Lambda_{ij} = \mathrm{diag}\,\{\lambda_1,\ldots,\lambda_5\}$ correspond to the characteristic speeds of wave propagation

$$\lambda_1 = \hat{v}_{ij} - \hat{c}_{ij}, \qquad \lambda_2 = \lambda_3 = \lambda_4 = \hat{v}_{ij}, \qquad \lambda_5 = \hat{v}_{ij} + \hat{c}_{ij}. \tag{33}$$

Here $\hat{c}_{ij}$ is the speed of sound for Roe's linearization as defined in (17), while $\hat{v}_{ij}$ is a 'projection' of the density-averaged velocity onto the edge ij

$$\hat{v}_{ij} = \mathbf{e}_{ij}\cdot\hat{\mathbf{v}}_{ij}, \qquad \text{where} \quad \mathbf{e}_{ij} = \frac{\mathbf{a}_{ij}}{|\mathbf{a}_{ij}|}. \tag{34}$$

In the continuous case, the characteristics associated with the multiple eigenvalue $\lambda_2 = \lambda_3 = \lambda_4$ represent the trajectories of fluid particles. In addition, there are two superimposed acoustic waves traveling at speeds $\pm c$ relative to the gas. We remark that the 'upwind' direction is different for different waves since it depends on the sign of the eigenvalues. This is why we abstain from fixing the order of nodes for the edges of the sparsity graph and postpone the discussion of edge orientation for TVD limiters until section 7.2.

The characteristic decomposition (32) enables us to eliminate the negative eigenvalues in the coefficient blocks L_{ij} and L_{ji} of the discrete Jacobian by resorting to an analog of discrete upwinding. The artificial dissipation D_{ij} for the resulting 'flux difference splitting' scheme is defined as follows [19]

$$\mathrm{D}_{ij} = |\mathrm{A}_{ij}| = |\mathbf{a}_{ij}|\,\mathrm{R}_{ij}|\Lambda_{ij}|\mathrm{R}_{ij}^{-1}, \tag{35}$$

where the matrix $|\Lambda_{ij}|$ contains the absolute values of the eigenvalues

$$|\Lambda_{ij}| = \mathrm{diag}\,\{|\lambda_1|,\ldots,|\lambda_5|\}\,. \tag{36}$$

In one dimension, we recover Roe's approximate Riemann solver [36] which is one of the most popular discretization techniques for the Euler equations. For a comprehensive review and a comparative study of such upwind-biased low-order schemes the interested reader is referred to [27],[29].

Instead of dealing with the cumulative Roe matrix, it is possible to employ *directional splitting* and diagonalize the Jacobians one at a time [17],[22]

$$\mathrm{D}_{ij} = \sum_{d=1}^{3} |\mathrm{A}_{ij}^{d}|, \qquad \text{where} \quad \mathrm{A}_{ij}^{d} = a_{ij}^{d} \hat{A}_{ij}^{d}. \tag{37}$$

This decomposition into one-dimensional wave patterns brings about strong numerical diffusion in the crosswind direction. Moreover, the cost of evaluating D_{ij} triples in comparison to (35). On the other hand, the implementation is very simple and flux limiting in terms of characteristic variables is feasible. Many compressible flow solvers are based on this sort of splitting.

A much cheaper alternative to (35) and (37) is to add scalar dissipation proportional to the spectral radius of the Roe matrix [25],[27],[48]

$$\mathrm{D}_{ij} = d_{ij}\mathrm{I}, \qquad \text{where} \quad d_{ij} = |\mathbf{a}_{ij}| \max_{i} |\lambda_i|. \tag{38}$$

Note that it affects just the diagonal blocks of the discrete Jacobian operator and is the same for all variables. This simplifies bookkeeping and reduces the cost of matrix assembly. As long as excessive artificial viscosity is removed in the course of flux correction, the computational effort required for evaluation of D_{ij} from (35) or (37) does not pay off. Surprisingly enough, a slightly overdiffusive low-order method may even be preferable due to the resulting improvement of phase accuracy [19],[48]. Last but not least, scalar dissipation can be used as a preconditioner for the approximate Riemann solver.

Remark. The insertion of artificial diffusion can be combined with a conservative flux decomposition for the term $K\mathrm{U}$, which yields [18],[19]

$$(K\mathrm{U})_i = -\sum_{j \neq i} \mathrm{G}_{ij}, \qquad \mathrm{G}_{ij} = \mathbf{c}_{ij} \cdot \mathbf{F}_i - \mathbf{c}_{ji} \cdot \mathbf{F}_j. \tag{39}$$

Such a representation is particularly useful for finite element codes utilizing an edge-based data structure. Moreover, it facilitates an extension of many one-dimensional discretization schemes to unstructured meshes [27],[31].

The low-order counterpart of G_{ij} can be constructed by adding a diffusive flux depending on the spectral properties of the Roe matrix

$$\mathrm{G}_{ij}^{*} = \mathrm{G}_{ij} - (\mathrm{D}_{ij} + \mathrm{B}_{ij})(\mathrm{U}_j - \mathrm{U}_i). \tag{40}$$

This modification is equivalent to using (30) in lieu of (27) during the assembly of the global Jacobian operator. Hence, the right-hand side of the semi-discrete low-order scheme (29) can be expressed as

$$(L\mathrm{U})_i = -\sum_{j \neq i} \mathrm{G}_{ij}^{*}, \qquad \mathrm{G}_{ji}^{*} = -\mathrm{G}_{ij}^{*} \tag{41}$$

and assembled edge-by-edge from the antisymmetric numerical fluxes G_{ij}^{*} if this is desirable from the viewpoint of implementation.

6 One-Dimensional Case

Before embarking on the design of flux limiters, it is instructive to apply the above low-order method to the one-dimensional Euler equations

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = 0. \tag{42}$$

In this case, the vectors of unknown variables and fluxes reduce to

$$U = \begin{bmatrix} \rho \\ \rho v \\ \rho E \end{bmatrix}, \qquad F = \begin{bmatrix} \rho v \\ \rho v^2 + p \\ \rho H v \end{bmatrix}. \tag{43}$$

The differentiation of F yields the equivalent quasi-linear form

$$\frac{\partial U}{\partial t} + A\frac{\partial U}{\partial x} = 0, \tag{44}$$

where $A = \frac{\partial F}{\partial U}$ is the Jacobian matrix. It is easy to verify that

$$A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{1}{2}(\gamma-3)v^2 & (3-\gamma)v & \gamma - 1 \\ \frac{1}{2}(\gamma-1)v^3 - vH & H - (\gamma-1)v^2 & \gamma v \end{bmatrix}. \tag{45}$$

Furthermore, the characteristic decomposition of A is as follows

$$A = R\Lambda R^{-1}, \qquad \Lambda = \mathrm{diag}\{v - c, v, v + c\}, \tag{46}$$

where v is the fluid velocity and $c = \sqrt{\gamma p/\rho}$ is the local speed of sound for a polytropic gas. The ratio $M = \frac{|v|}{c}$ is called the *Mach number*.

The columns of the matrix R represent the right eigenvectors

$$R = \begin{bmatrix} 1 & 1 & 1 \\ v - c & v & v + c \\ H - vc & \frac{1}{2}v^2 & H + vc \end{bmatrix} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3] \tag{47}$$

and the rows of its inverse R^{-1} correspond to the left ones

$$R^{-1} = \begin{bmatrix} \frac{1}{2}\left(b_1 + \frac{v}{c}\right) & \frac{1}{2}\left(-b_2 v - \frac{1}{c}\right) & \frac{1}{2}b_2 \\ 1 - b_1 & b_2 v & -b_2 \\ \frac{1}{2}\left(b_1 - \frac{v}{c}\right) & \frac{1}{2}\left(-b_2 v + \frac{1}{c}\right) & \frac{1}{2}b_2 \end{bmatrix} = \begin{bmatrix} \mathbf{l}_1 \\ \mathbf{l}_2 \\ \mathbf{l}_3 \end{bmatrix}, \tag{48}$$

where the auxiliary coefficients b_1 and b_2 are given by

$$b_1 = b_2\frac{v^2}{2}, \qquad b_2 = \frac{\gamma - 1}{c^2}.$$

By definition, the eigenvalues and eigenvectors of A satisfy the equations

$$A\mathbf{r}_k = \lambda_k \mathbf{r}_k, \qquad \mathbf{l}_k A = \lambda_k \mathbf{l}_k, \qquad k = 1, 2, 3 \tag{49}$$

which can be written in matrix form as $AR = R\Lambda$ and $R^{-1}A = \Lambda R^{-1}$.

After the space discretization by the low-order scheme, we end up with the following ODE system for the nodal values of the conservative variables

$$m_i \frac{\mathrm{d}\mathrm{U}_i}{\mathrm{d}t} = \sum_{j \neq i} \mathrm{L}_{ij}(\mathrm{U}_j - \mathrm{U}_i), \qquad \text{where} \quad \mathrm{L}_{ij} = \mathrm{K}_{ij} + \mathrm{D}_{ij}. \tag{50}$$

At interior nodes, the high-order element/edge contribution is of the form

$$\mathrm{K}_{ij} = -c_{ij}\hat{A}_{ij} = -\mathrm{A}_{ij}, \qquad j = i \pm 1, \tag{51}$$

where $\hat{A}_{ij}$ is the Jacobian matrix (45) evaluated using the Roe mean values $(\hat{\rho}_{ij}, \hat{v}_{ij}, \hat{H}_{ij})$. On a uniform mesh of linear finite elements, we have

$$m_i = \Delta x, \qquad c_{ij} = \begin{cases} 1/2 & \text{for } \; j = i + 1, \\ -1/2 & \text{for } \; j = i - 1. \end{cases} \tag{52}$$

The viscous dissipation D_{ij} defined in (35) and (37) simplifies to

$$\mathrm{D}_{ij} = \frac{1}{2}|\hat{A}_{ij}|, \qquad \text{where} \quad |\hat{A}_{ij}| = \mathrm{R}_{ij}|\Lambda_{ij}|\mathrm{R}_{ij}^{-1}. \tag{53}$$

Plugging it into formula (40), we obtain the corresponding numerical flux, which is identical to that for Roe's approximate Riemann solver [36]

$$\mathrm{G}_{ij}^{*} = \frac{\mathrm{F}_i + \mathrm{F}_j}{2} - \frac{1}{2}|\hat{A}_{ij}|(\mathrm{U}_j - \mathrm{U}_i), \qquad j = i + 1. \tag{54}$$

The underlying principles and the properties of the resulting first-order scheme are described in many textbooks on gas dynamics [17],[22],[42]. We remark that Roe's method fails to recognize expansion waves and, therefore, may give rise to entropy-violating solutions (rarefaction shocks) in the vicinity of sonic points. A suitable *entropy fix*, for instance, the modification proposed by Harten and Hyman [14], makes it possible to circumvent this problem.

In the case of scalar dissipation (38), the amount of artificial viscosity is proportional to the largest in magnitude eigenvalue of $\hat{A}_{ij}$

$$\mathrm{D}_{ij} = \frac{\lambda_{\max}}{2}\mathrm{I}, \qquad \text{where} \quad \lambda_{\max} = |\hat{v}_{ij}| + \hat{c}_{ij}. \tag{55}$$

Note that the diffusion coefficient $d_{ij} = \frac{\lambda_{\max}}{2}$ equals that for discrete upwinding applied to the fastest wave propagating at the characteristic speed $\lambda_{\max}$. This definition of D_{ij} constitutes a cost-effective alternative to (53) and yields a perfect low-order method for the FCT algorithm [48].

7 Flux Limiting for Systems

Despite the remarkable progress made in the development of high-resolution schemes for scalar conservation laws, their extension to hyperbolic systems remains a challenging open problem. One of the main difficulties that hinder a rigorous generalization of the scalar machinery to the systems arena is the lack of reliable physical and mathematical criteria for the design of flux limiters. Some useful concepts, algorithms, and ideas are presented in this section. For an in-depth coverage of this topic, the reader is referred to [48].

It is tempting to treat the constituents of system (1)–(3) independently using a flux correction algorithm developed for scalar transport equations. Unfortunately, this naïve approach often yields rather disappointing results. In particular, a 'blind' adjustment of the conservative fluxes may turn out to be harmful and give rise to undershoots/overshoots in certain dependent variables, such as pressure, velocity, internal energy, entropy etc. The intricate coupling of the Euler equations makes it very difficult to monitor and control the evolution of all physically relevant quantities simultaneously. Hence, the numerical solutions are strongly influenced not only by the type of the flux limiter but also by the set of variables to which it is applied.

In light of the above, flux limiting for hyperbolic systems is more involved than that for scalar conservation laws. The design criteria should reflect the physical properties of the problem at hand and be enforced in a fail-safe way. To this end, the following strategies have been proposed [25],[48]

- flux limiting in terms of nonconservative (e.g. primitive, characteristic) variables, for which the constraints to be imposed are relevant/important;
- a proper synchronization of the correction factors for individual variables;
- an a-posteriori control and cancellation/tuning of the antidiffusive fluxes into nodes at which nonphysical solution values are detected.

These remedies are problem-dependent and require empirical input as well as a solid understanding of the underlying physics. However, the simulation results are typically rewarding. Following these guidelines, one can perform algebraic flux correction for the Euler equations as explained below.

7.1 Variable Transformations

The first issue to be addressed is the choice of variables. It is essential to use the divergence form (6) of the Euler equations and maintain conservation at the discrete level. This guarantees convergence to a weak solution and prevents shocks from moving at wrong speeds. However, the numerical method should allow for a physical growth/decay of local extrema, so imposing tight solution bounds on the conservative variables is not to be recommended [48]. Instead, viable constraints can be devised and enforced making use of a local transformation of the solution variations and of the raw antidiffusive fluxes to a set of variables that are more amenable to flux correction [25].

In particular, it is not unusual that flux limiting in terms of the primitive variables $(\rho, \mathbf{v}, p)$ produces superior results as compared to the use of the conservative or mixed variables [27],[31]. An even better (albeit costly) alternative is to constrain a set of suitably defined *characteristic variables* for the linearized hyperbolic system. A decisive advantage of this approach is that scalar discretization tools are applicable to the transformed equations and there is no need for an *ad hoc* synchronization of the flux limiter. Moreover, the physical nature of wave propagation is taken into account. Thus, a proper control of the characteristic variables is typically sufficient to guarantee that all other variables are also free of nonphysical oscillations.

For linear hyperbolic systems such that the coefficient matrix A is constant, multiplication of (44) from the left by R^{-1} and use of (46) yields

$$\frac{\partial W}{\partial t} + \Lambda \frac{\partial W}{\partial x} = 0, \qquad \text{where} \quad W := R^{-1}U. \tag{56}$$

Since Λ is a diagonal matrix, this so-called *canonical form* of system (44) consists of three decoupled convection equations

$$\frac{\partial w_k}{\partial t} + \lambda_k \frac{\partial w_k}{\partial x} = 0, \qquad k = 1, 2, 3. \tag{57}$$

It follows that the evolution of the characteristic variables $W = [w_1, w_2, w_3]^T$ is described by three simple waves propagating independently of one another with velocities v and $v \pm c$. The exact solution is constant along the characteristics of (57) which are the straight lines satisfying the ODEs

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \lambda_k, \qquad \text{where} \quad \lambda_k = \text{const}. \tag{58}$$

The solution to the original problem can be recovered as $U = RW$ or

$$U = \sum_{k=1}^{3} w_k \mathbf{r}_k, \qquad w_k(x,t) = w_k(x - \lambda_k t, 0). \tag{59}$$

Hence, the characteristic variables w_k are the coefficients of $\mathbf{r}_k$ in the eigenvector expansion of U which depends only on the initial data [20].

For nonlinear hyperbolic systems, the eigenvalues and eigenvectors of the Jacobian matrix A vary in space and time. Therefore, characteristic variables can only be defined *locally*, for a given solution U. After the spatial discretization, the transformation matrices R and R^{-1} are evaluated for a 'frozen' set of data such as the arithmetic average or the Roe mean values for two nodes. In particular, the tensorial artificial viscosity (53) corresponds to

$$\mathrm{D}_{ij}(\mathrm{U}_j - \mathrm{U}_i) = \frac{1}{2}\mathrm{R}_{ij}|\Lambda_{ij}|\Delta\mathrm{W}_{ij}, \tag{60}$$

where the transformed solution increment $\Delta\mathrm{W}_{ij}$ is given by

$$\Delta\mathrm{W}_{ij} := \mathrm{R}_{ij}^{-1}(\mathrm{U}_j - \mathrm{U}_i). \tag{61}$$

This representation reveals that Roe's approximate Riemann solver can be interpreted as discrete upwinding applied to the decoupled scalar equations for the local characteristic variables. Indeed, the amount of artificial diffusion for field k is proportional to $|\lambda_k|/2$, which is just enough to render the scalar semi-discrete scheme local extremum diminishing (see the 1D example in section 5.1 of the previous chapter). In order to calculate the corresponding correction to the original discretization, the vector of characteristic diffusive fluxes $\mathrm{F}_{ij} = \frac{1}{2}|\Lambda_{ij}|\Delta\mathrm{W}_{ij}$ is converted back to the conservative variables in (60). In practice, both forward and backward variable transformations can/should be performed in a computationally efficient way without generating the eigenvector matrices for each edge explicitly [27],[47]. Arguing as above, one can construct characteristic flux limiters of TVD and FCT type.

7.2 Characteristic TVD Limiter

Characteristic TVD schemes for the Euler equations date back to the 1980s [46],[47]. In the one-dimensional case, their derivation is fairly straightforward, and various *ad hoc* extensions have been used in finite element codes with considerable success [2],[10],[28],[38],[39]. The ongoing quest for a genuinely multidimensional generalization is complicated by the fact that the underlying characteristic decomposition is no longer unique. In multidimensions, the waves can travel in an infinite number of directions, so it is unclear how to transform the linearized hyperbolic system into a set of decoupled convection equations, for which robust numerical techniques are already available. Hence, one needs not only a fully multidimensional transport algorithm but also a multidimensional wave decomposition for the system at hand [40]. A few promising wave models have been proposed for the design of upwind-biased schemes based on the *fluctuation splitting* approach [3] (also referred to as *residual distribution* [7]) but their complexity is still too high.

The freedom of choosing the direction $\mathbf{e}$ in the factorization (9) arbitrarily turns out to be a blessing and a curse at the same time. On the one hand, it provides the necessary flexibility for the development of numerical schemes. On the other hand, the definition of characteristic variables is ambiguous in multidimensions and has a strong influence on the simulation results. Recall that we constructed the artificial viscosity (35) via a diagonalization of the cumulative Roe matrix A_{ij} which was evaluated in the direction of the normalized coefficient vector $\mathbf{e}_{ij}$ defined in (34). As shown by Lyra et al. [28], this approach is also suitable for the design of characteristic FEM-TVD schemes based on unidirectional slope limiting. On the other hand, the use of a single direction $\mathbf{e} = \mathbf{e}_{ij}$ for the definition of local characteristic variables seems to be inappropriate for *node-oriented* flux limiters which operate as follows:

1. Gather upstream/downstream edge contributions to individual nodes.
2. Compute the nodal correction factors on the basis of this information.
3. Check the sign of the antidiffusive fluxes and limit them edge-by-edge.

Obviously, the data to be collected in the first step should correspond to the same set of characteristic variables, whereas the transformation matrices R_{ij} and R_{ij}^{-1} for the cumulative Roe matrix depend not only on the averaged solution values but also on the coefficient vector $\mathbf{e}_{ij}$ for the edge at hand. This has led us to employ the directional splitting (37) and limit the characteristic variables defined by the eigenvectors of $\hat{A}_{ij}^d$ which are independent of the mesh. Of course, this provisional wave model is hardly optimal, so the experienced reader is encouraged to experiment with other options.

Let us construct a characteristic TVD limiter following the strategy proposed by Yee et al. [47] and apply it to the vector of fluxes in each coordinate direction so as to impose the LED constraint on the characteristic variables. Both explicit and implicit time discretizations are feasible. As before, all algebraic modifications to be performed affect just the matrix assembly routine. The limiting process consists of the following algorithmic steps:

1. In a loop over edges, generate the right/left eigenvectors of the unidirectional Roe matrix $\hat{A}_{ij}^d$ $(d = 1, 2, 3)$ which correspond to the eigenvalues

$$\lambda_1^d = \hat{v}_{ij}^d - \hat{c}_{ij}, \qquad \lambda_2^d = \lambda_3^d = \lambda_4^d = \hat{v}_{ij}^d, \qquad \lambda_5^d = \hat{v}_{ij}^d + \hat{c}_{ij}. \tag{62}$$

   Analytical expressions for the eigenvector matrices can be found in [17].

2. Decompose the difference between the nodal values U_j and U_i into its characteristic components via a transformation analogous to (61)

$$\Delta\mathrm{W}_{ij} = [\mathrm{R}_{ij}^d]^{-1}(\mathrm{U}_j - \mathrm{U}_i), \tag{63}$$

   where the rows of $[\mathrm{R}_{ij}^d]^{-1}$ represent the left eigenvectors of $\hat{A}_{ij}^d$.

3. Update the sums $P_i^\pm$ and $Q_i^\pm$ of downstream/upstream edge contributions to node i for the characteristic field number k as follows:

For all edges/fluxes A_{ij}^d	For boundary edges B_{ij}^d
Compute $k_{ij}^a = -a_{ij}^d \lambda_k^d$ and $\delta_a^\pm = \begin{matrix}\max\\ \min\end{matrix} \{0, k_{ij}^a \Delta\mathrm{W}_{ij}^k\}$	Compute $k_{ij}^b = -b_{ij}^d \lambda_k^d$ and $\delta_b^\pm = \begin{matrix}\max\\ \min\end{matrix} \{0, k_{ij}^b \Delta\mathrm{W}_{ij}^k\}$
If $k_{ij}^a < 0$, then augment $P_i^\pm := P_i^\pm + \delta_a^\pm, \;\; Q_j^\pm := Q_j^\pm + \delta_a^\pm$	If $k_{ij}^b < 0$, then augment $P_i^\pm := P_i^\pm + \delta_b^\pm, \;\; P_j^\mp := P_j^\mp - \delta_b^\pm$
If $k_{ij}^a > 0$, then augment $P_j^\pm := P_j^\pm + \delta_a^\pm, \;\; Q_i^\pm := Q_i^\pm + \delta_a^\pm$	If $k_{ij}^b > 0$, then augment $Q_i^\pm := Q_i^\pm + \delta_b^\pm, \;\; Q_j^\mp := Q_j^\mp - \delta_b^\pm$

4. In a loop over nodes, calculate the nodal correction factors for all local characteristic variables in each space direction

$$R_i^\pm = \Phi(Q_i^\pm / P_i^\pm). \tag{64}$$

5. Determine the number of the 'upwind node' *individually* for each scalar wave propagating at the speed λ_k^d relative to the edge ij

$$I = \begin{cases} i & \text{if } k_{ij}^a \le 0, \\ j & \text{if } k_{ij}^a > 0, \end{cases} \tag{65}$$

   where the coefficient k_{ij}^a is the one defined in step 3 (see the left box).

6. Limit the transformed data jumps $\Delta \mathrm{W}_{ij}$ in accordance with the adopted edge orientation for the characteristic variable at hand

$$\Delta\widehat{\mathrm{W}}_{ij}^k = \begin{cases} R_I^+ \Delta \mathrm{W}_{ij}^k & \text{if } \delta_a^k \ge 0, \\ R_I^- \Delta \mathrm{W}_{ij}^k & \text{if } \delta_a^k < 0, \end{cases} \qquad \delta_a^k = k_{ij}^a \Delta \mathrm{W}_{ij}^k. \tag{66}$$

   Note that the value of δ_a^k is invariant to the order of nodes due to the asymmetry of the parameters $k_{ji}^a = -k_{ij}^a$ and $\mathrm{W}_{ji}^k = -\mathrm{W}_{ij}^k$.

7. Add the limited antidiffusive correction to the raw diffusive flux $|\Lambda_{ij}^d| \Delta \mathrm{W}_{ij}$ and transform the result back to the conservative variables

$$\mathrm{F}_{ij}^d = |a_{ij}^d| \mathrm{R}_{ij}^d |\Lambda_{ij}^d| (\Delta \mathrm{W}_{ij} - \Delta\widehat{\mathrm{W}}_{ij}). \tag{67}$$

8. Insert the net (anti-)diffusive flux F_{ij}^d into the right-hand side and/or the defect vector for the modified linear system by setting

$$(K^* \mathrm{U})_i := (K^* \mathrm{U})_i + \mathrm{F}_{ij}^d, \qquad (K^* \mathrm{U})_j := (K^* \mathrm{U})_j - \mathrm{F}_{ij}^d \tag{68}$$

   for $K^* \mathrm{U}$ initialized by $K \mathrm{U}$ for the original Galerkin scheme (11).

The practical implementation of the algorithm may certainly differ from that outlined above and be based on a more involved wave model or used in conjunction with a fractional-step approach such as the ADI method [47].

A few remarks are in order. In the case of scalar convection, the final antidiffusion coefficient for the edge $\overrightarrow{ij}$ was given by $a_{ij} = \min\{R_i^{\pm} d_{ij}, l_{ji}\}$. This extra check is no longer necessary since

$$k_{ji}^a = -k_{ij}^a \quad \Rightarrow \quad l_{ji}^a := k_{ji}^a + d_{ij}^a = 2 d_{ij}^a,$$

where $d_{ij}^a := |a_{ij}^d \lambda_k^d| = |k_{ij}^a|$ is the artificial diffusion coefficient for wave k. The LED property for row j follows from the fact that

$$k_{ji}^* := l_{ji}^a - R_i^{\pm} d_{ij}^a = k_{ji}^a + (1 - R_i^{\pm}) d_{ij}^a = (2 - R_i^{\pm}) d_{ij}^a \ge 0$$

as long as $R_i^{\pm} \le 2$, which is the case for all standard TVD limiters. Furthermore, we found that the symmetric contributions $\delta_b^k = k_{ij}^b \Delta \mathrm{W}_{ij}^k$ of boundary edges can be omitted at steps 5-7 (this amounts to a selective 'prelimiting' of the antidiffusive fluxes) but must be taken into account at step 3.

7.3 Synchronized FCT Limiter

In the FCT community, flux limiting has traditionally been performed on the conservative or primitive variables using a scalar correction factor α_{ij} for the vector F_{ij} of raw antidiffusive fluxes. As already indicated above, this kind of synchronization is mandatory unless the variables under consideration are fully decoupled. Otherwise, spurious ripples may emerge as a consequence of relative phase errors. Löhner *et al.* [24],[25] mentioned the following approaches to the design of a synchronized FCT limiter for the Euler equations:

- evaluation of the correction factors for a single 'indicator variable';
- using the minimum of those obtained for a certain group of variables;
- conservative flux limiting based on local variable transformations.

A general algorithm for constraining an arbitrary set of target quantities is outlined in the recent monograph by Löhner [25]. The following realization of his methodology can be used to carry out algebraic flux correction in terms of any nonconservative variables including the characteristic ones:

1. Initialize the auxiliary arrays by $P_i^{\pm} \equiv 0,\ Q_i^{\pm} \equiv 0,\ R_i^{\pm} \equiv 1.$

2. In a loop over edges, convert the raw antidiffusive fluxes as well as the differences between the nodal values of $\tilde{\mathrm{U}}$ to the new variables

$$\hat{\mathrm{F}}_{ij} = \mathrm{T}_{ij}\mathrm{F}_{ij}, \qquad \Delta\mathrm{W}_{ij} = \mathrm{T}_{ij}(\tilde{\mathrm{U}}_j - \tilde{\mathrm{U}}_i), \tag{69}$$

   where the transformation matrix T_{ij} depends on the choice of variables and is evaluated using an appropriate average of $\tilde{\mathrm{U}}_i$ and $\tilde{\mathrm{U}}_j$ [48].

3. Perform the (optional) prelimiting of the transformed fluxes

$$\hat{\mathrm{F}}_{ij}^k := 0 \qquad \text{if} \quad \hat{\mathrm{F}}_{ij}^k \Delta\mathrm{W}_{ij}^k \geq 0. \tag{70}$$

4. Augment the sums of positive/negative fluxes for each target variable

$$P_i^{\pm} := P_i^{\pm} + \max_{\min}\{0, \hat{\mathrm{F}}_{ij}^k\}, \qquad P_j^{\pm} := P_j^{\pm} + \max_{\min}\{0, -\hat{\mathrm{F}}_{ij}^k\}. \tag{71}$$

5. Update the maximum/minimum admissible increments as follows

$$Q_i^{\pm} := \max_{\min}\left\{Q_i^{\pm}, \Delta\mathrm{W}_{ij}^k\right\}, \qquad Q_j^{\pm} := \max_{\min}\left\{Q_j^{\pm}, -\Delta\mathrm{W}_{ij}^k\right\}. \tag{72}$$

6. In a loop over nodes, calculate the nodal correction factors

$$R_i^{\pm} := \min\{1, m_i Q_i^{\pm}/P_i^{\pm}\} \qquad \text{if} \quad |P_i^{\pm}| > 0. \tag{73}$$

7. In a loop over edges, inspect the sign of the antidiffusive fluxes and compute the final correction factors for the selected control variables

$$\alpha_{ij}^k = \begin{cases} \min\{R_i^+, R_j^-\} & \text{if } \hat{\mathrm{F}}_{ij}^k \geq 0, \\ \min\{R_j^+, R_i^-\} & \text{if } \hat{\mathrm{F}}_{ij}^k < 0. \end{cases} \tag{74}$$

8. Apply the synchronized weight $\hat{\alpha}_{ij} = \min_k \alpha_{ij}^k$, directly to F_{ij} or limit $\hat{\mathrm{F}}_{ij}$ and transform it back to the conservative variables

$$\mathrm{F}_{ij}^* = \mathrm{T}_{ij}^{-1} \hat{\mathrm{F}}_{ij}^*, \qquad \text{where} \quad \hat{\mathrm{F}}_{ij}^* = \hat{\alpha}_{ij} \hat{\mathrm{F}}_{ij}. \tag{75}$$

9. Check the result and set $\mathrm{F}_{ij}^* := 0$ if nonphysical behavior is detected [48].

10. Insert the limited antidiffusive flux F_{ij}^* into the right-hand side B and/or into the defect vector for the fully discrete scheme

$$\mathrm{B}_i := \mathrm{B}_i + \mathrm{F}_{ij}^*, \qquad \mathrm{B}_j := \mathrm{B}_j - \mathrm{F}_{ij}^*. \tag{76}$$

At present, there is still a large degree of empiricism in the construction of synchronized FCT limiters, and their performance is strongly problem-dependent. Reportedly, the use of synchronization on the density and energy is to be recommended for highly dynamic flows, while the density and pressure should be controlled for steady-state applications [24],[25]. At the same time, the minimum of correction factors for all conservative variables is too restrictive. In particular, insignificant fluctuations of the crosswind velocity may cause a complete cancellation of the antidiffusive flux. Therefore, an equal treatment of all velocity components yields poor results, especially if the flow takes place in a predominant direction. Interestingly enough, the iterative limiting strategy introduced in the previous chapter is much less sensitive to the choice of indicator variables and to the synchronization procedure, since the rejected antidiffusion can be reused at subsequent flux/defect correction steps.

Zalesak was the first to explore the potential of characteristic variables in the realm of FCT methods [48]. His characteristic version of the Boris-Book limiter is remarkably robust (due to a 'fail-safe' control mechanism) and yields impressive results for the one-dimensional Euler equations. For an in-depth presentation of the underlying design principles, the reader is referred to [48]. Unfortunately, no fully multidimensional extension is available to date. Since Zalesak's limiter is node-oriented, the use of operator splitting is currently unavoidable for the reasons explained above. In fact, the situation is even more difficult than in the case of TVD algorithms since the raw antidiffusive fluxes are evaluated in terms of U^n and/or U^{n+1}, whereas the transformation matrices for the limiting step depend on the intermediate solution $\tilde{\mathrm{U}}$.

The time-dependent nature of FCT suggests a fractional-step approach to the limiting of characteristic variables in the framework of a 'Cartesian' wave model. A very simple way to decouple the components of $\mathbf{F}$ in system (6) is to invoke the first-order accurate Marchuk-Yanenko splitting

$$\frac{\partial U_d}{\partial t} + \frac{\partial F^d}{\partial x_d} = 0 \qquad \text{in } (t^n, t^{n+1}) \qquad d = 1, 2, 3. \tag{77}$$

The final solution to each subproblem serves as initial data for the next one

$$U_d^n = U_{d-1}^{n+1}, \qquad U_0^{n+1} = U^n, \qquad U^{n+1} = U_3^{n+1}.$$

Due to the splitting of spatial derivatives and of the associated Jacobians, the transformation matrices $\mathrm{T}_{ij} = [\mathrm{R}_{ij}^d]^{-1}$ provide a suitable definition of local characteristic variables for Zalesak's limiter. We remark that no synchronization of the nodal correction factors is necessary in this case. Furthermore, the employed operator splitting does not rule out the use of unstructured meshes, since the limiting procedure remains fully multidimensional. However, the solution of three subproblems per time step and costly variable transformations impose a heavy burden in terms of CPU time. Therefore, the advantages of flux limiting on characteristic variables are rather dubious so far.

8 Iterative Solution Strategies

The nonlinear algebraic systems resulting from an implicit FEM-FCT or FEM-TVD discretization of the Euler equations exhibit the same structure as their scalar counterparts. This enables us to design iterative solvers building on the defect correction technique described previously. Let the successive approximations to the desired solution U^{n+1} be updated as follows

$$\mathrm{U}^{(m+1)} = \mathrm{U}^{(m)} + [A(\mathrm{U}^{(m)})]^{-1}\mathrm{R}^{(m)}, \qquad m = 0, 1, 2, \ldots \tag{78}$$

where $A(\mathrm{U}^{(m)})$ is a suitable preconditioner. A good candidate is the matrix for system (28) discretized in time by the standard $\theta-$scheme

$$A(\mathrm{U}^{(m)}) = M_L + \theta \Delta t L(\mathrm{U}^{(m)})\mathrm{U}^{(m)}, \qquad 0 < \theta \leq 1. \tag{79}$$

It is worthwhile to construct $A(\mathrm{U}^{(m)})$ using scalar dissipation (38) no matter what form of D_{ij} is adopted for the artificial viscosity built into $\mathrm{R}^{(m)}$.

The constant right-hand side for the low-order method is given by

$$\mathrm{B}^n = M_L + (1-\theta)\Delta t L(\mathrm{U}^n)\mathrm{U}^n. \tag{80}$$

It is augmented by limited antidiffusion and incorporated into the defect

$$\mathrm{R}^{(m)} = \mathrm{B}^n + \mathrm{F}(\mathrm{U}^{(m)}, \mathrm{U}^n) - A(\mathrm{U}^{(m)})\mathrm{U}^{(m)} \tag{81}$$

which vanishes for the exact solution of the discrete problem. It is essential to guarantee that $\mathrm{R}^{(m)}$ does approach zero in an actual simulation. This is especially important in the case of implicit TVD-like methods which satisfy the imposed criteria only upon convergence and may produce nonphysical solution values if the number of outer iterations is insufficient.

Since the inversion of the global matrix $A(\mathrm{U}^{(m)})$ is prohibitively expensive, the following two-step implementation of (78) is to be employed:

$$A(\mathrm{U}^{(m)})\Delta\mathrm{U}^{(m+1)} = \mathrm{R}^{(m)}, \qquad m = 0, 1, 2, \ldots \tag{82}$$

$$\mathrm{U}^{(m+1)} = \mathrm{U}^{(m)} + \Delta\mathrm{U}^{(m+1)}, \qquad \mathrm{U}^{(0)} = \mathrm{U}^n. \tag{83}$$

The linear system to be solved at the first step can be written as

$$\begin{bmatrix} A_{11}^{(m)} & A_{12}^{(m)} & A_{13}^{(m)} & A_{14}^{(m)} & A_{15}^{(m)} \\ A_{21}^{(m)} & A_{22}^{(m)} & A_{23}^{(m)} & A_{24}^{(m)} & A_{25}^{(m)} \\ A_{31}^{(m)} & A_{32}^{(m)} & A_{33}^{(m)} & A_{34}^{(m)} & A_{35}^{(m)} \\ A_{41}^{(m)} & A_{42}^{(m)} & A_{43}^{(m)} & A_{44}^{(m)} & A_{45}^{(m)} \\ A_{51}^{(m)} & A_{52}^{(m)} & A_{53}^{(m)} & A_{54}^{(m)} & A_{55}^{(m)} \end{bmatrix} \begin{bmatrix} \Delta \mathrm{U}_1^{(m+1)} \\ \Delta \mathrm{U}_2^{(m+1)} \\ \Delta \mathrm{U}_3^{(m+1)} \\ \Delta \mathrm{U}_4^{(m+1)} \\ \Delta \mathrm{U}_5^{(m+1)} \end{bmatrix} = \begin{bmatrix} \mathrm{R}_1^{(m)} \\ \mathrm{R}_2^{(m)} \\ \mathrm{R}_3^{(m)} \\ \mathrm{R}_4^{(m)} \\ \mathrm{R}_5^{(m)} \end{bmatrix}. \quad (84)$$

Here the row index refers to the vector of unknowns for one conservative variable (density, momentum component or energy) which interacts with the others via the off-diagonal blocks of the preconditioner $A(\mathrm{U}^{(m)})$.

Simultaneous solution of these equations gobbles up a lot of resources in terms of CPU time and memory requirements. Instead, the constituents of the discretized Euler equations can be decoupled and treated separately or, better yet, in parallel. This can be accomplished by switching to a block-diagonal preconditioner. Let us consider the *block-Jacobi method* [5] obtained from (84) by setting $A_{kl}^{(m)} := \delta_{kl} A_{kl}^{(m)}$, where δ_{kl} is the Kronecker delta. In this case, the global update (83) reduces to a sequence of scalar subproblems

$$A_{kk}^{(m)} \Delta \mathrm{U}_k^{(m+1)} = \mathrm{R}_k^{(m)}, \qquad k = 1, \ldots, 5. \quad (85)$$

$$\mathrm{U}_k^{(m+1)} = \mathrm{U}_k^{(m)} + \Delta \mathrm{U}_k^{(m+1)}, \qquad \mathrm{U}_k^{(0)} = \mathrm{U}_k^{n}, \quad (86)$$

whereby a (small) number of inner iterations is performed to compute the solution increment $\Delta \mathrm{U}_k^{(m+1)}$ for each variable. An implicit or explicit underrelaxation [12] may be applied at the first and second step, respectively.

This segregated algorithm is very easy to implement and its performance is satisfactory as long as the time step remains relatively small. However, the nonlinear convergence rates deteriorate and the simulation may even fail completely as the time step increases. In particular, a time-marching technique combined with (85)-(86) may experience severe convergence problems when applied to steady transonic flows. This can be attributed to the fact that the *characteristic condition number* (the ratio of the largest to the smallest wave speed) is very high for Mach numbers approaching zero or unity. As a result, the Euler equations become very stiff and call for the use of local preconditioning [6] and/or a strongly coupled solution strategy.

A much more robust iterative solver can be devised on the basis of a Krylov-subspace or multigrid method for the **global** system (84). To avoid the storage of all matrix blocks and 'invert' one block at a time, it is worthwhile to use a smoother/preconditioner of block-Jacobi or block-Gauß-Seidel type. For an abstract system $A\mathrm{X} = \mathrm{B}$ having the same structure as (84), the latter updates the components of the vector X row-by-row as follows

$$A_{kk} \mathrm{X}_k^{(m+1)} = \mathrm{B}_k - \sum_{l<k} A_{kl} \mathrm{X}_l^{(m+1)} - \sum_{l>k} A_{kl} \mathrm{X}_l^{(m)}. \quad (87)$$

This splitting differs from the block-Jacobi method in that the latest solution values are used for the evaluation of the right-hand side. In either case, only the diagonal blocks of the global matrix need to be 'inverted' approximately in the inner iteration process. However, the block-Gauß-Seidel method is preferable as it yields better convergence rates at no additional cost [5].

A preliminary evaluation of BiCGSTAB and multigrid solvers equipped with this (weakly) coupled relaxation technique indicates that they are far superior to the algorithm (85)-(86) when it comes to simulation of (quasi-) stationary flows [32],[33]. On the other hand, the segregated approach or even a fully explicit time-stepping is appropriate for transient applications since the time step must remain rather small for accuracy reasons. Due to the tradeoff between robustness and efficiency, the choice of the 'best' solution strategy is highly problem-dependent. Therefore, a general-purpose CFD code should consist of several modules optimized for different flow regimes [43].

9 Implementation of Boundary Conditions

The numerical treatment of boundary conditions for the Euler equations is an issue of utmost importance and its influence on the simulation results should not be underestimated. A comprehensive coverage of this topic is available in a number of textbooks [10],[13],[17],[44]. This section is intended to give an insight into some theoretical and numerical aspects pertinent to the implementation of *characteristic boundary conditions* in a finite element code.

For a hyperbolic system that consists of N_v partial differential equations, $N_p \leq N_v$ physical boundary conditions (PBC) plus $N_n = N_v - N_p$ numerical boundary conditions (NBC) are to be prescribed. The former must secure the existence and uniqueness of the exact solution, while the latter are supposed to ensure that various perturbations generated in the interior of the computational domain leave it without being reflected at the boundaries [17]. Hence, a proper combination of PBC and NBC must be imposed by means of some extra matrix/vector manipulations to be described below.

9.1 Physical Boundary Conditions

The number of physical boundary conditions N_p depends on the pattern of wave propagation. For simplicity, consider a local coordinate system such that the unit outward normal is aligned with the $x-$axis. The characteristic variables associated with the direction $\mathbf{n} = (\pm 1, 0, 0)$ are given by [44]

$$W = \left[v_n - \frac{2c}{\gamma - 1}, s, v_2, v_3, v_n + \frac{2c}{\gamma - 1} \right]^T, \tag{88}$$

where $v_n = \mathbf{n} \cdot \mathbf{v}$ stands for the normal velocity, c is the speed of sound and $s = c_v \log(p/\rho^\gamma)$ is the entropy defined up to an additive constant.

The canonical transformation (56) of the linearized Euler equations yields a sequence of decoupled convection equations for the five components of the vector W which are called the *Riemann invariants*. These scalar quantities are transported along the characteristic curves at the respective speeds

$$\lambda_1 = v_n - c, \qquad \lambda_2 = \lambda_3 = \lambda_4 = v_n, \qquad \lambda_5 = v_n + c \tag{89}$$

which represent the eigenvalues of $A_n = \mathbf{n} \cdot \mathbf{A}$. For a pure convection equation, boundary conditions are required only at the inlet. Hence, N_p equals the number of negative eigenvalues which correspond to the incoming characteristics. It can readily be seen that the direction of wave propagation depends not only on the sign of the velocity v_n but also on the local Mach number $M = |v_n|/c$. Therefore, several different situations need to be considered:

Supersonic open boundary $\quad |v_n| > c$

At the inlet $v_n < 0$ and all eigenvalues are negative; the values of all characteristic variables must be specified. At the outlet $v_n > 0$ and all eigenvalues are positive; no physical boundary conditions may be imposed.

Subsonic open boundary $\quad |v_n| < c$

At the inlet $v_n < 0$ so that λ_5 is positive while the other eigenvalues are negative; all Riemann invariants except w_5 must be specified. At the outlet $v_n > 0$ so that the first eigenvalue is negative, whereas the other ones are positive. Hence, just the boundary value of w_1 is to be prescribed.

Solid wall boundary $\quad v_n = 0$

At the solid wall, the normal velocity component must be constrained to zero. Under this *no-penetration* or *free slip* condition, there is no convective flux through the boundary. All eigenvalues except $\lambda_1 = -c$ are nonnegative so that only w_1 must be specified. This Riemann invariant corresponds to an acoustic wave entering the domain at the speed of sound.

In practice, physical boundary conditions are typically imposed on the primitive variables or given in terms of input data like the total enthalpy, entropy, temperature or inclination angle. The problem remains well-posed as long as the number of boundary conditions is correct.

9.2 Numerical Boundary Conditions

The need for numerical boundary conditions stems from the fact that the actual problem to be solved is formulated in terms of the conservative variables rather than Riemann invariants. Therefore, it is impossible to impose the Dirichlet boundary conditions in the usual way. It is common practice to recover the boundary values by changing to the characteristic variables, evaluating the incoming Riemann invariants from the physical boundary conditions and extrapolating the outgoing ones from the interior of the domain [17],[44],[45]. The inverse transformation yields the desired values of the conservative variables which can be used to compute the numerical fluxes for a

finite volume method or treated as Dirichlet boundary conditions for a finite difference scheme. However, the extrapolation of Riemann invariants lacks a rigorous justification and is expensive to perform on unstructured meshes. Since we are not aware of any publication regarding the implementation of characteristic boundary conditions for an implicit finite element method, we deem it appropriate to go into some technical details.

In order to *predict* the solution values at boundary nodes, we modify the preconditioner $A(\mathrm{U}^{(m)}) = \{a_{ij}^{kl}\}$ by picking out the corresponding rows and setting their off-diagonal entries equal to zero. That is, if node i belongs to the boundary, then $a_{ij}^{kl} := 0,\ \forall j \neq i,\ \forall l \neq k$. This enables us to update the components of the vector U_i explicitly prior to solving the linear system (84). To this end, we divide the components of the nodal defect vector $\mathrm{R}_i^{(m)}$ by the diagonal entries of the preconditioner and increment the old iterate

$$\mathrm{U}_i^* = \mathrm{U}_i^{(m)} + \mathrm{A}_{ii}^{-1}\mathrm{R}_i^{(m)}\ , \qquad \mathrm{A}_{ii} = \mathrm{diag}\ \{a_{ii}^{kk}\}. \tag{90}$$

Next, the provisional solution U_i^* is transformed to the local characteristic variables W_i (see appendix). Recall that the number of physical boundary conditions is equal to the number of incoming characteristics. Hence, those components of the vector W_i^* which correspond to negative eigenvalues of the Jacobian $A_n = \mathbf{n} \cdot \mathbf{A}$ can be replaced by their exact values evaluated from the given input data. The outgoing Riemann invariants, which are associated with numerical boundary conditions, remain unchanged. Finally, we convert the corrected vector W_i^{**} back to the conservative variables, assign the result to U_i^{**} and nullify all entries of the defect vector $\mathrm{R}_i^{(m)}$ so that

$$\Delta\mathrm{U}_i^{(m+1)} = \mathrm{A}_{ii}^{-1}\mathrm{R}_i^{(m)} = 0 \qquad \Rightarrow \qquad \mathrm{U}_i^{(m+1)} = \mathrm{U}_i^{**}. \tag{91}$$

The flow chart of variable transformations and manipulations to be performed for the boundary nodes before the solution of system (84) is displayed in Fig. 2. In essence, the *corrected* values U_i^{**} represent standard Dirichlet boundary conditions for the end-of-step solution $\mathrm{U}_i^{(m+1)}$. Note that there is no need for an *ad hoc* extrapolation of data from the interior. Further implementation details are presented in the appendix to this chapter.

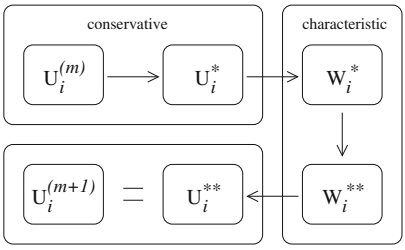


Fig. 2. Variable transformations for boundary nodes

10 Numerical Examples

Below we apply our algebraic flux correction schemes to a few academic test problems that encompass both stationary and time-dependent inviscid flows at different Mach numbers. The computational results presented in this section provide a preliminary validation of the new algorithms and illustrate their characteristic features for an admittedly limited range of applications. At the current stage of development, there are still many aspects that call for further investigation. In particular, it is rather difficult to predict how the flux limiters and iterative solvers would perform on highly unstructured and possibly anisotropic meshes for industrial flows in complex geometries. Some fine-tuning may turn out to be necessary but the numerical examples that follow indicate that (semi-) implicit FEM-FCT and FEM-TVD methods constitute a promising approach to compressible flow simulations.

10.1 Shock Tube Problem

Our first example is the classical shock tube problem which exhibits an interesting flow structure and is widely used in CFD for testing purposes [20],[41]. Its physical prototype is a closed tube filled with inviscid gas which is initially at rest and separated by a membrane into the regions of high and low pressure. The initial conditions for the Riemann problem are given by

$$\begin{bmatrix} \rho_L \\ \mathbf{v}_L \\ p_L \end{bmatrix} = \begin{bmatrix} 1.0 \\ 0.0 \\ 1.0 \end{bmatrix} \quad \text{for } x \in [0, 0.5], \qquad \begin{bmatrix} \rho_R \\ \mathbf{v}_R \\ p_R \end{bmatrix} = \begin{bmatrix} 0.125 \\ 0.0 \\ 0.1 \end{bmatrix} \quad \text{for } x \in (0.5, 1].$$

The rupture of the membrane at $t = 0$ triggers a *normal shock wave* which moves to the right with velocity satisfying the Rankine-Hugoniot conditions. All of the primitive variables are discontinuous across the shock which is followed by a *contact discontinuity*. The latter represents a moving interface between the regions of different densities but constant velocity and pressure. At the same time, a *rarefaction wave* propagates in the opposite direction providing a smooth transition to the original values of the state variables in the left part of the domain. Hence, the flow pattern in the shock tube is characterized by three waves traveling at different speeds [20].

The snapshots shown in Fig. 3 correspond to the time instant $t = 0.231$. The one-dimensional Euler equations were discretized using 100 linear finite elements and the Crank-Nicolson time-stepping with $\Delta t = 10^{-3}$. The analytical solution is designated by the dashed lines in each plot. It was obtained using the technique described by Anderson [1]. The numerical solutions of low order (upper diagrams) are seen to be nonoscillatory and qualitatively correct but their accuracy leaves a lot to be desired. As a matter of fact, Roe's approximate Riemann solver (53) performs slightly better than scalar dissipation (55) but also results in a strong smearing of the moving fronts.

The removal of excessive artificial diffusion in the course of flux correction yields a dramatic improvement of the overall accuracy (see Fig. 3, bottom). The employed FEM-TVD scheme was equipped with the superbee limiter, while the companion FEM-FCT method was implemented following the algorithm proposed by Zalesak [48]. In either case, the low-order preconditioner was based on scalar dissipation and flux limiting was carried out in terms of the characteristic variables. The use of the conservative ones is also feasible provided that the correction factors are properly synchronized. Simulation results for the two-dimensional shock tube problem solved by FCT with synchronization on the density and energy are reported in [19]. If the minimum of correction factors for all conservative variables is adopted, then the basic limiter yields overly diffusive results, whereas the iterative version performs remarkably well [32]. Indeed, a 'cautious' limiting strategy does not degrade the accuracy as long as the rejected antidiffusion can be recycled.

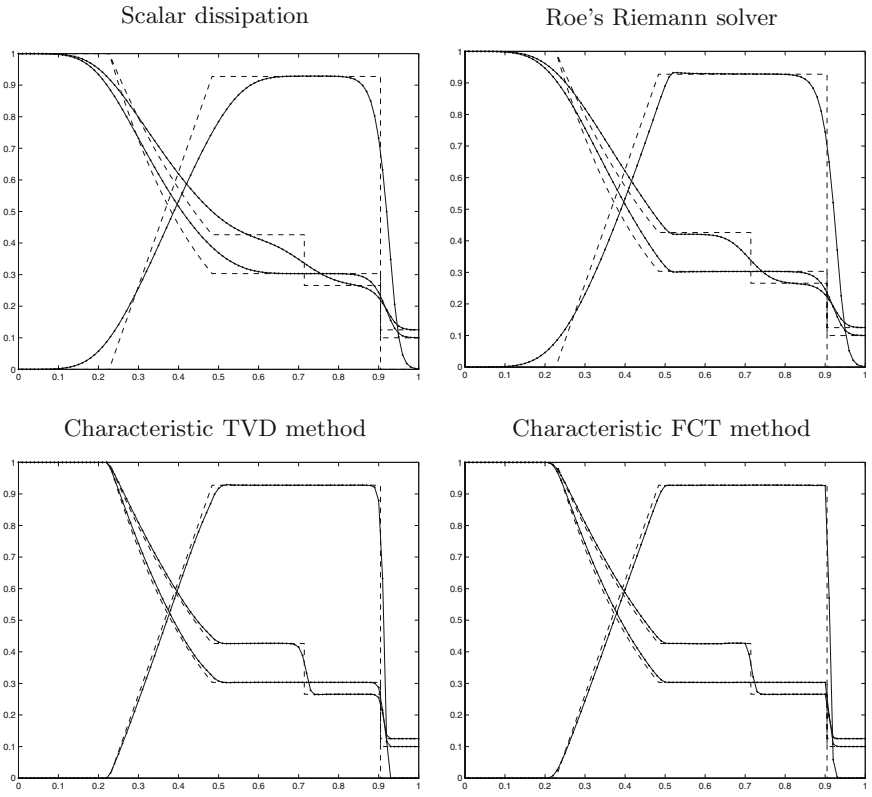


Fig. 3. Shock tube problem: numerical solutions at $t = 0.231$

10.2 Radially Symmetric Riemann Problem

Another transient test problem proposed by LeVeque [21] represents a counterpart of the shock tube problem in polar coordinates. Before an impulsive start, the square domain $\Omega = (-0.5, 0.5) \times (-0.5, 0.5)$ is separated by an imaginary membrane into two subregions: $\Omega_L = \{(x, y) \in \Omega \,|\, r = \sqrt{x^2 + y^2} < 0.13\}$ and $\Omega_R = \Omega \backslash \Omega_L$. The gas is initially at rest, whereby its pressure and density are higher within the circle Ω_L than outside of it:

$$\begin{bmatrix} \rho_L \\ \mathbf{v}_L \\ p_L \end{bmatrix} = \begin{bmatrix} 2.0 \\ 0.0 \\ 15.0 \end{bmatrix} \text{ in } \Omega_L, \qquad \begin{bmatrix} \rho_R \\ \mathbf{v}_R \\ p_R \end{bmatrix} = \begin{bmatrix} 1.0 \\ 0.0 \\ 1.0 \end{bmatrix} \text{ in } \Omega_R.$$

The abrupt removal of the membrane at $t = 0$ gives rise to a radially expanding shock wave which is induced by the pressure difference. The challenge of this benchmark is to capture the moving shock and to make sure that the numerical solution of the Riemann problem remains radially symmetric.

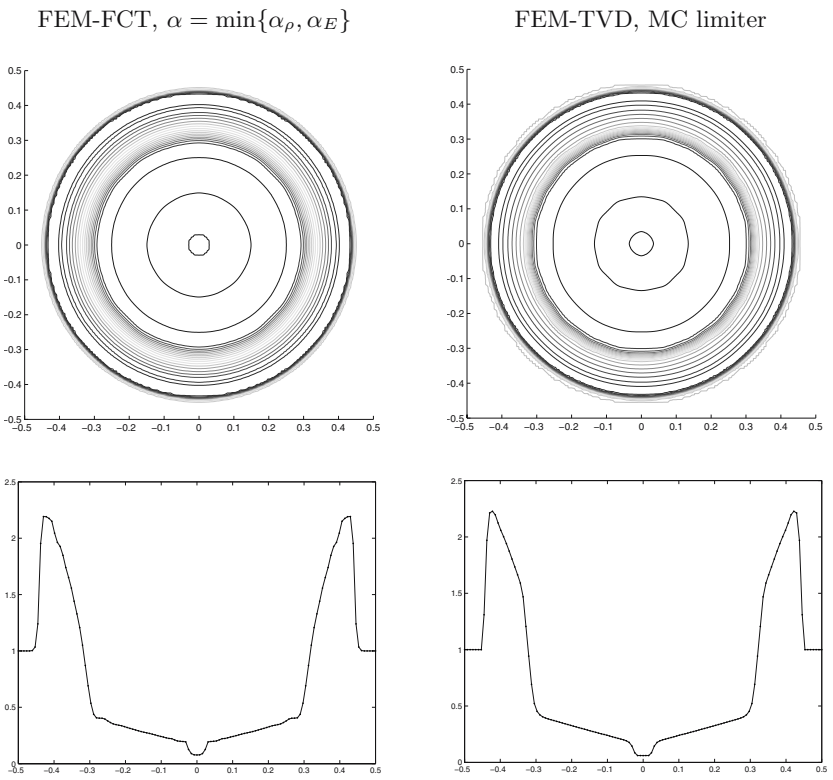


Fig. 4. Radially symmetric density distribution at $t = 0.13$

The simulation results depicted in Fig. 4 were obtained on a uniform mesh of 128×128 bilinear elements by the characteristic FEM-TVD method (left) and by the iterative FEM-FCT algorithm with flux synchronization on the density and energy (right). As before, the discretization in time was performed by the Crank-Nicolson scheme and the constant time step $\Delta t = 10^{-3}$ was employed. The contour lines for the density distribution at $t = 0.13$ (upper diagrams) have the form of concentric circles, so that both high-resolution schemes succeed in preserving the radial symmetry of initial data. The lower diagrams show the density profiles along the $x-$axis. These solutions are in a very good agreement with those computed by the public-domain software package CLAWPACK [23] using the same mesh and time step.

10.3 Compression Corner

To illustrate the potential of the fully implicit backward Euler time-stepping, we consider a steady two-dimensional supersonic flow over a wedge which may imitate e.g. the tip of a projectile. This configuration is sometimes referred to as *compression corner* and constitutes an excellent test problem because it admits an analytical solution in the framework of the oblique shock theory [1]. The originally uniform supersonic flow preserves its freestream characteristics until it reaches the wedge which deflects it upward through an angle θ. Provided that θ is not too large, the change in the flow direction takes place across a shock wave which has the form of a straight line emanating from the tip of the wedge and running oblique to the original flow direction.

Given the inflow Mach number $M_1 = 2.5$ and the deflection angle $\theta = 15^\circ$, one can determine the downstream Mach number $M_2 = 1.87$ and the shock inclination angle $\beta = 36.94^\circ$ as explained, e.g., in [1]. A detailed description of this test case is available in the *CFD Verification and Validation Database* of the NPARC Alliance [35]. The simulation results presented in Fig. 5 were computed on a boundary-fitted mesh of 128×128 bilinear elements and interpolated onto a Cartesian background mesh for visualization purposes. All these solutions were marched to the steady state by the backward Euler method. Although it is unconditionally stable, the magnitude of admissible time steps depends on the robustness of the iterative solver. In particular, the segregated algorithm (85)-(86) cannot be operated at $\Delta t > 10^{-2}$ unless a strong underrelaxation is performed. By contrast, time steps as large as $\Delta t = 1.0$ can be used in conjunction with the coupled solution strategy [32].

The upper diagrams show the distribution of Mach numbers predicted by the low-order method based on scalar dissipation. The upstream and downstream Mach numbers M_1 and M_2 are correct but the transition between them is anything else but discontinuous. The oblique shock is strongly smeared by numerical diffusion and resembles a 'rarefaction wave'. At the same time, this low-order solution provides a reasonable initial guess for the FEM-TVD and FEM-FCT algorithms. Let us examine the moderately diffusive MC limiter

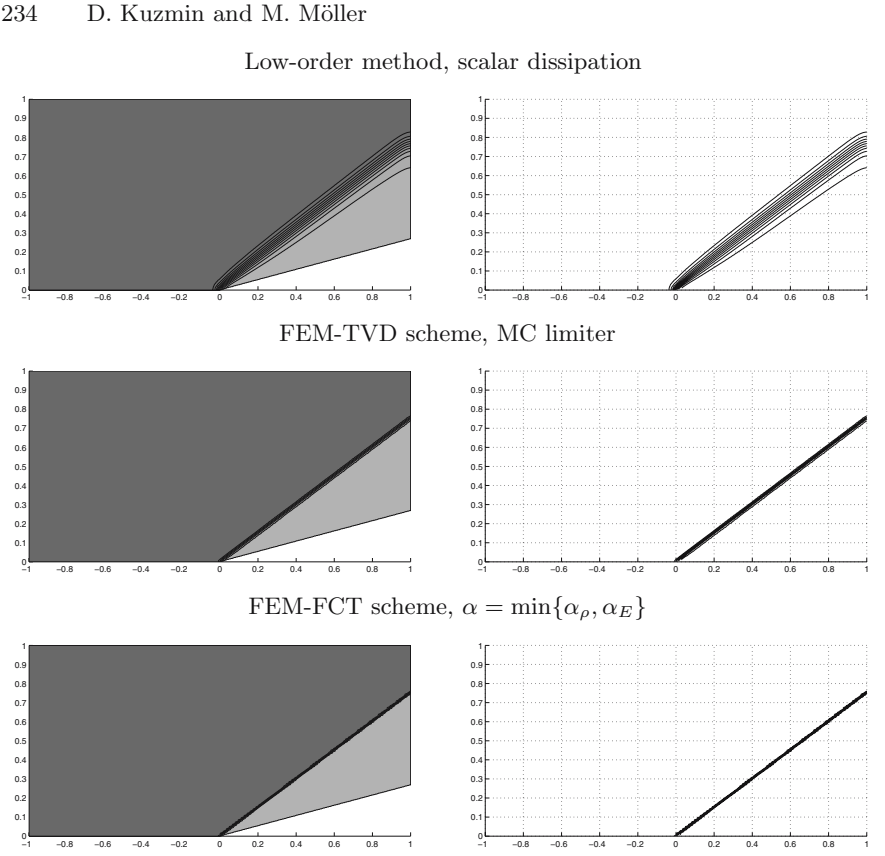


Fig. 5. Compression corner: $M_1 = 2.5$, $\theta = 15^\circ$

and the iterative Zalesak limiter with the ρ-E synchronization of the correction factors. After a sufficient number of outer defect correction steps, a significant improvement is observed in the resolution of the stationary shock (see Fig. 5). In fact, the results produced by both high-resolution schemes are even superior to the reference solution from the NPARC database [35].

10.4 Prandtl-Meyer Expansion

Our next example deals with a steady supersonic flow being deflected downward rather than upward. In this case, the flow behavior is quite different from the one discussed above for the compression corner. As the gas reaches the kink, it starts spreading and the flow characteristics change smoothly across the so-called *Prandtl-Meyer expansion* wave. The streamlines gradually bend downward and eventually become parallel to the lower wall as they leave the rarefaction fan which separates the regions of uniform flow. All flow properties adjust themselves continuously across the rarefaction wave except for

the midpoint at which the wall geometry changes abruptly. The analytical solution to this problem and further details can be found in [1].

In accordance with the NPARC setup for this benchmark [35], we adopt the freestream Mach number $M_1 = 2.5$ and the deflection angle $\theta = 15°$. The resulting expansion fan is composed from an infinite number of iso-Mach lines lying in the angular sector bounded by the angles $\mu_1 = 23.58°$ (upstream) and $\mu_2 = 18.0°$ (downstream) as measured from the ramp portion of the wall. On exit from the rarefaction wave, the analytical value of the Mach number equals $M_2 = 3.24$. Figure 6 displays the low-order solution (top) versus those produced by the FEM-TVD (middle) and FEM-FCT (bottom) schemes. All parameter settings are the same as in the previous example and so is the relative performance of the methods under consideration.

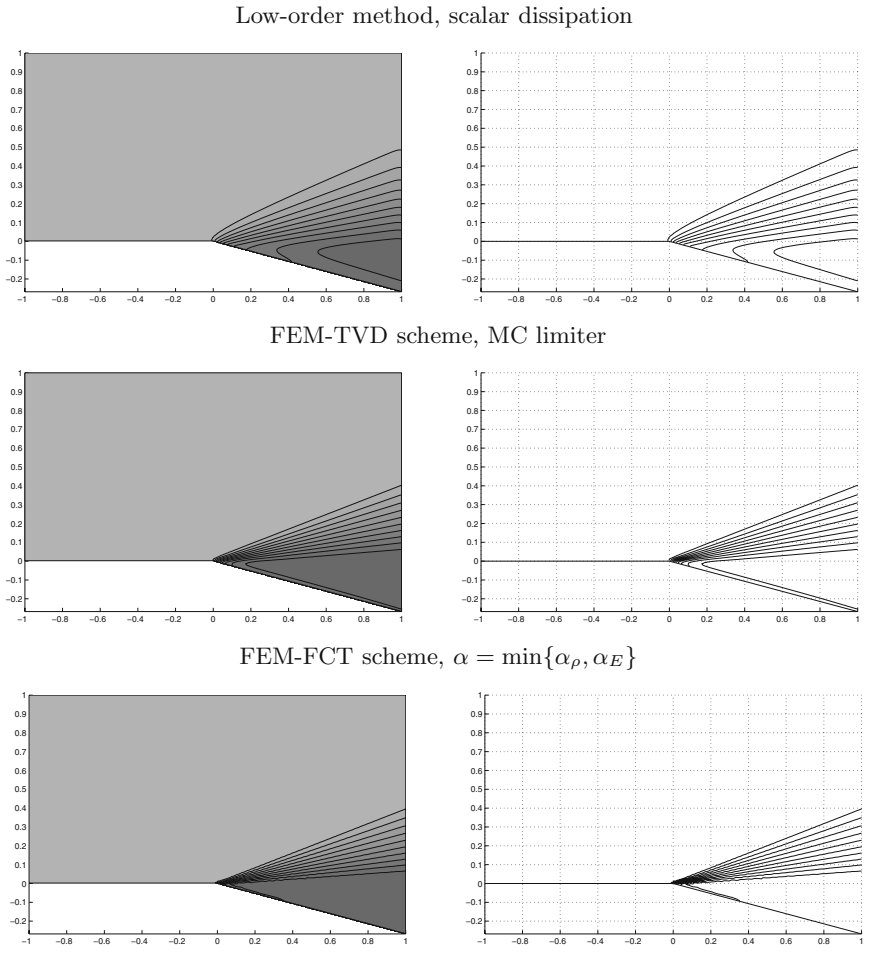


Fig. 6. Prandtl-Meyer expansion: $M_1 = 2.5$, $\theta = 15°$

10.5 Flow Past a Cylinder

In what follows, we stick to the characteristic FEM-TVD algorithm based on the MC limiter and apply it to a number of flow problems in non-Cartesian geometries so as to demonstrate the flexibility of the finite element approach. The first example is a steady supersonic flow past a circular cylinder placed in the center of a rectangular domain [31]. Specifically, the two-dimensional Euler equations are solved in $\Omega = \{(x, y) \in (-2.5, 2.5) \times (-5, 5) \, : \, x^2 + y^2 > 0.25\}$ and the freestream Mach number $M = 3$ is prescribed at the left border.

To generate the hierarchical quad-tree data structures for the geometric multigrid solver, the elements of a manually generated coarse mesh are successively subdivided into four subelements. The body-fitted computational mesh shown in Fig. 7 (left) consists of 15,104 bilinear elements and corresponds to the third refinement level for the underlying coarse grid. Note that local mesh refinement is performed in the wake of the cylinder. The Mach number distribution (right diagram) gives an insight into the flow structure which features

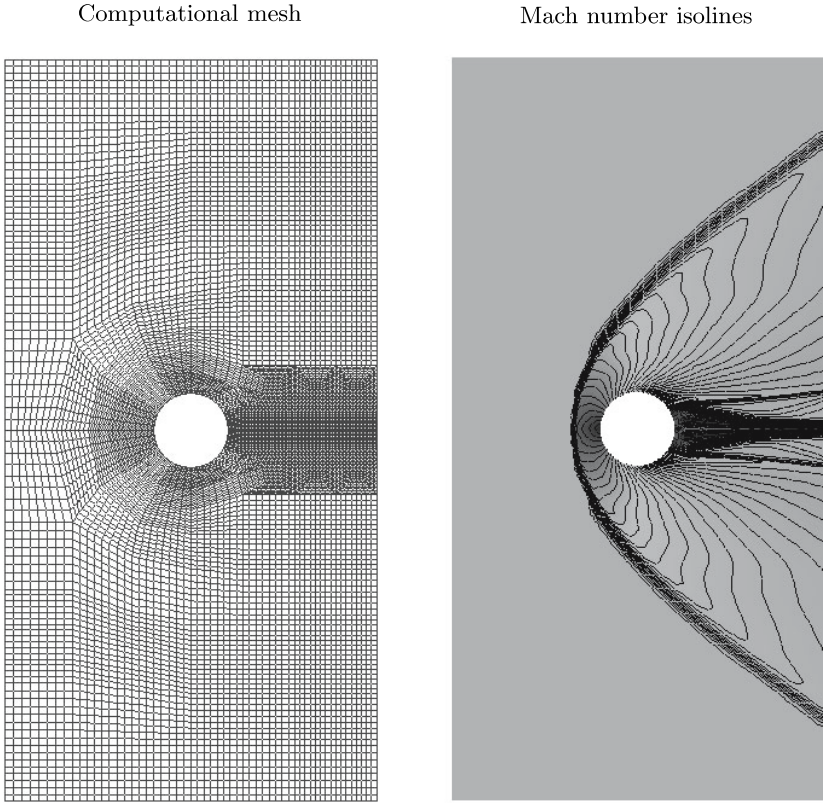


Fig. 7. Steady flow past a cylinder: FEM-TVD scheme, Q_1 elements

a strong bow shock as well as stagnation and rarefaction zones. Contrary to the numerical study published in [27],[31] no stability or convergence problems were encountered during this simulation.

On the other hand, the sensitivity of the computational results to the employed mesh was confirmed by our numerical experiments. The unstructured 'coarse' triangulation displayed in Fig. 8 contains 2,942 elements (1,561 nodes) which amounts to 47,072 elements (23,896 nodes) after two refinement steps. In this case, the mesh size is fairly uniform, except near the internal boundary. The resulting flow pattern agrees well with that presented in Fig. 7 but some differences can be observed in the critical zone downstream of the cylinder, where the numerical solution is highly mesh-dependent [27]. This discrepancy can be attributed to the fact that separated flows admit nonunique solutions in the framework of an inviscid gas model unless additional empirical hypotheses are employed [4]. Therefore, such applications call for the use of the compressible Navier-Stokes equations rather than the Euler equations.

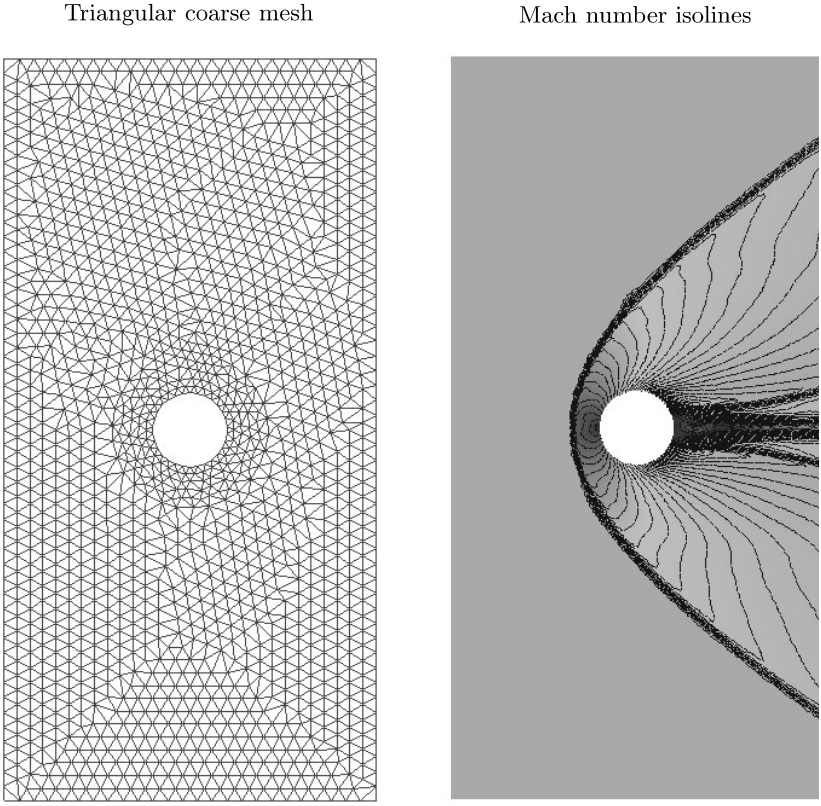


Fig. 8. Steady flow past a cylinder: FEM-TVD scheme, P_1 elements

10.6 Supersonic Scramjet Inlet

Another interesting benchmark configuration was introduced in [8]. A steady supersonic flow enters a narrowing channel at $M = 3$ and impinges on two sharp-cornered internal obstacles, which gives rise to multiple shock wave reflections. The computational domain Ω is delimited by three piecewise-linear boundary components passing through the following set of points [11]

	upper wall						*upper obstacle*				
x_i	0.0	0.4	4.9	12.6	14.25	16.9	4.9	8.9	9.4	14.25	12.6
y_i	3.5	3.5	2.9	2.12	1.92	1.7	1.4	0.5	0.5	1.2	1.4

The 'scramjet inlet' is symmetric so that the coordinates of the corner points for the lower wall/obstacle are given by $(x_i, -y_i)$. As in the previous example, a top-down approach to mesh generation is adopted. The underlying coarse mesh shown in Fig. 9 consists of 1,033 triangles (616 vertices) and the number of elements quadruples after each global refinement step.

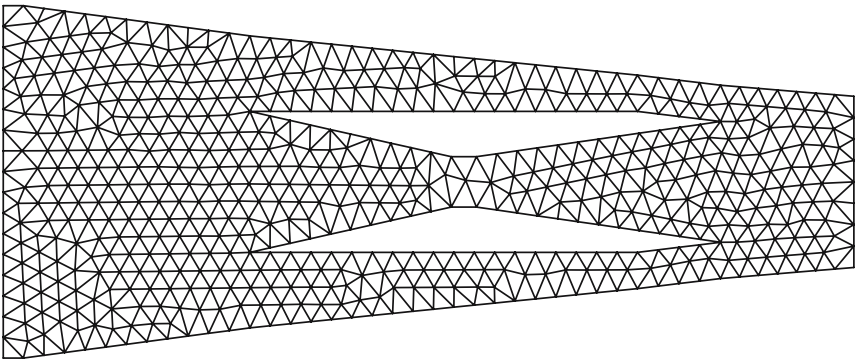


Fig. 9. Scramjet inlet: triangular coarse mesh

The stationary solutions displayed in Fig. 10 were produced by the fully implicit FEM-TVD method via time marching. The upper diagram depicts the Mach number distribution which was computed using 66,112 linear finite elements (33,859 nodes). Four times as many triangles (133,831 nodes) were employed to obtain the lower diagram. It can be seen that strong shocks are reproduced reasonably well but the interplay of weak ones is difficult to capture even on the finer mesh. Like in many other aerodynamic applications, there is a need for **adaptive** mesh refinement which can be readily realized in a finite element code if appropriate error estimators/indicators are available. Interestingly enough, the presented Mach number contours for the global refinement levels 3 and 4 resemble the simulation results of Díaz et al. [8] on the initial mesh and after one adaptation step, respectively.

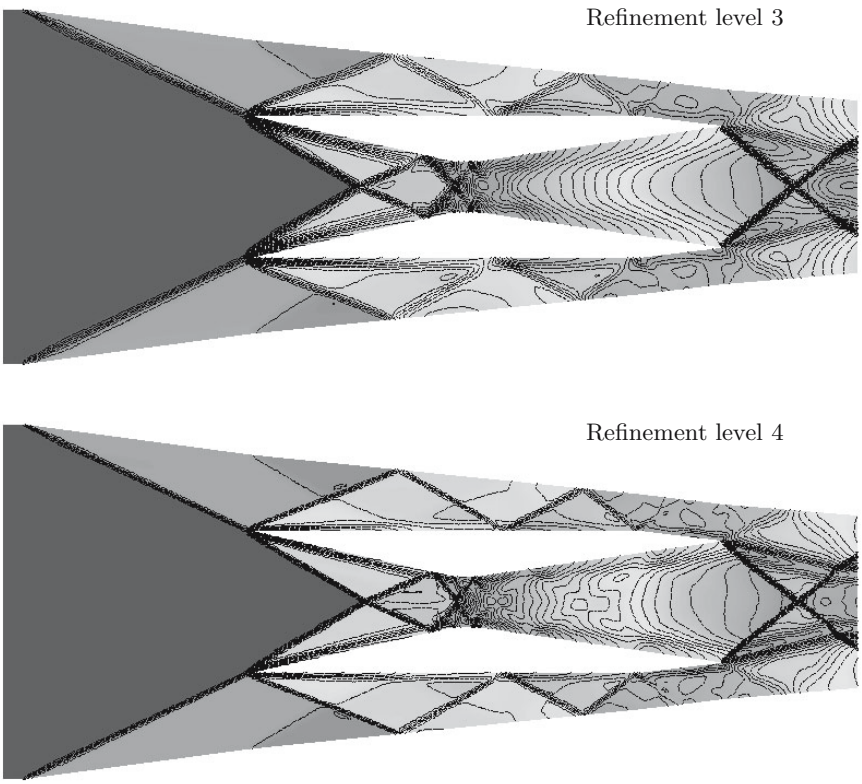


Fig. 10. Scramjet inlet: characteristic FEM-TVD method, MC limiter

10.7 NACA 0012 Airfoil

Finally, let us deal with external flows past a NACA 0012 airfoil at various freestream conditions. The upper and lower surfaces of this symmetric profile are given in the form $y = \pm f(x)$, respectively, where $x \in [0, 1.00893]$ and

$$f(s) = 0.6 \left(0.2969 s^{1/2} - 0.126 s - 0.3516 s^2 + 0.2843 s^3 - 0.1015 s^4 \right).$$

The compressible Euler equations are to be solved in a bounded domain, whose artificial far-field boundary is a circle of radius 10 centered at the tip of the airfoil. The first coarse mesh for this simulation is displayed in Fig. 11. It numbers 5,828 triangular elements and 2,973 vertices. A close-up of the central part shows the mesh details in the vicinity of the internal boundary. As an alternative, we consider a quadrilateral coarse mesh which consists of 3,000 elements and exhibits an unstructured tessellation pattern (see Fig. 12). In either case, the final mesh is constructed from the coarse one by two quadtree refinements, which gives 93,248 triangles or 48,000 quadrilaterals.

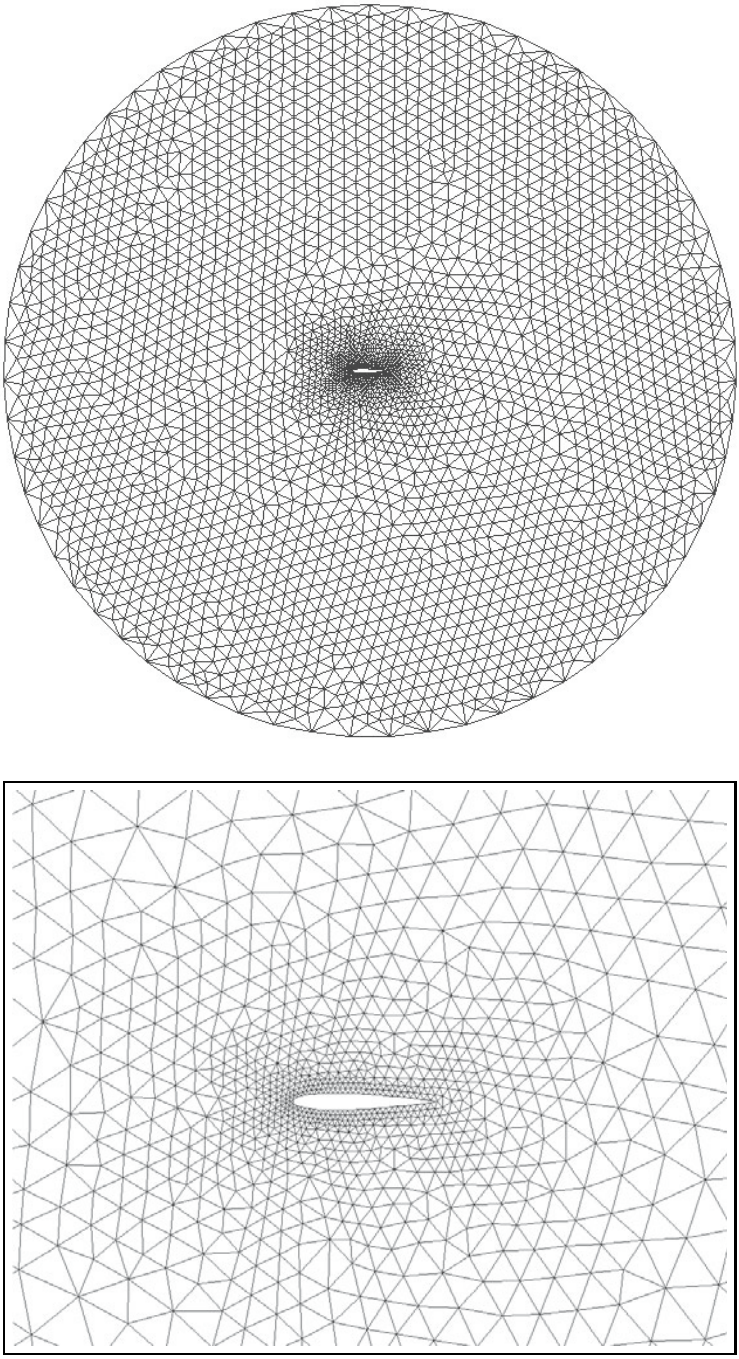


Fig. 11. Triangular coarse mesh for a NACA 0012 airfoil

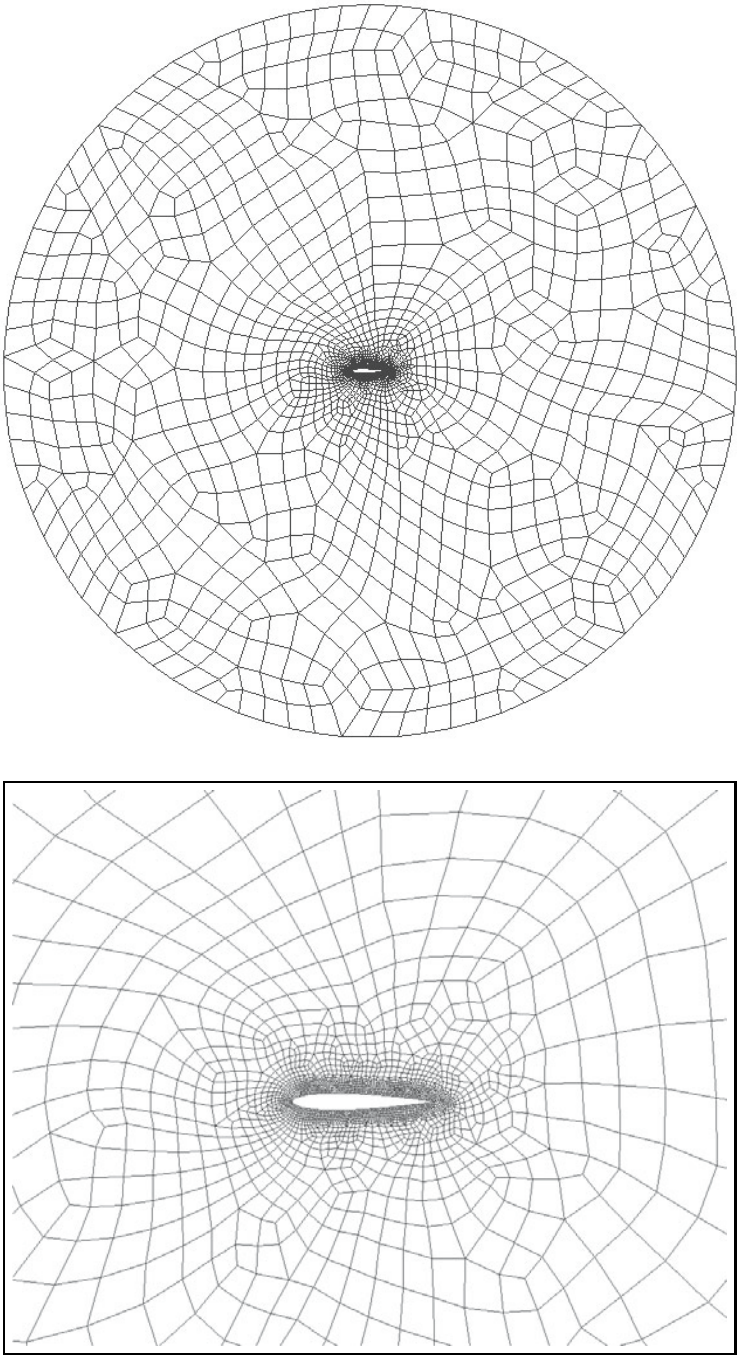


Fig. 12. Quadrilateral coarse mesh for a NACA 0012 airfoil

The density and pressure are initialized by unity and steady-state solutions are sought for the following far-field Mach numbers and angles of attack

Figure 13	$M_\infty = 0.5,\ \alpha = 0°$	subsonic flow
Figure 14	$M_\infty = 0.8,\ \alpha = 1.25°$	transonic flow
Figure 15	$M_\infty = 0.95,\ \alpha = 0°$	transonic flow

A comparison with the simulation results reported in the literature [11],[15] reveals that the numerical solutions presented in Fig. 13-15 are quite accurate, even though no adaptive mesh refinement was performed. Weighted a posteriori error estimates for an adaptive finite element discretization of the Euler equations can be derived as described in [15]. Clearly, even low-order methods yield reasonable results on a suitably refined mesh. The goal of our numerical study was to evaluate the performance of the new algorithm on a nonuniform mesh constructed without using any knowledge of the solution. On the other hand, the advent of self-adaptive FEM-TVD and FEM-FCT schemes would make it possible to exploit the advantages of unstructured grids to the full extent and create a very powerful CFD tool in the long run.

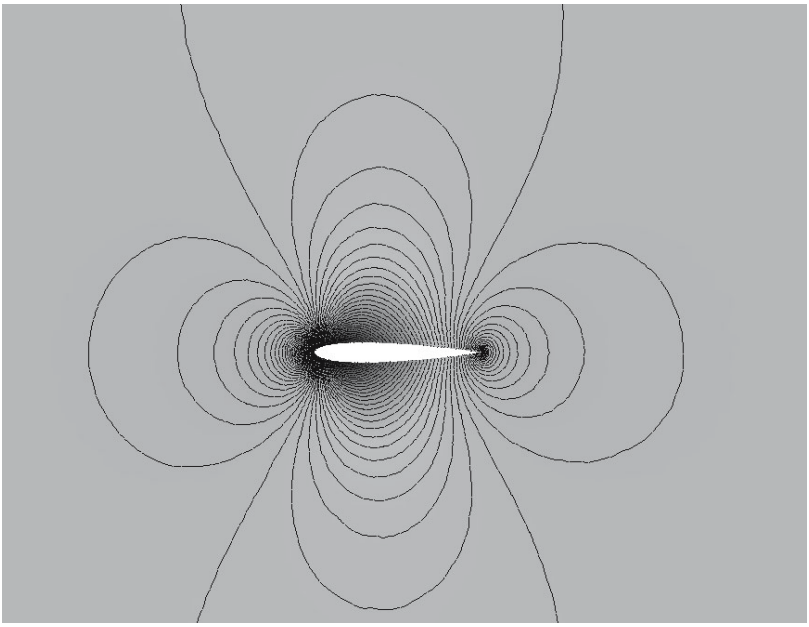
11 Conclusions and Outlook

In the present chapter, we extended the algebraic flux correction paradigm to the compressible Euler equations and proposed a practical algorithm for implementation of the resulting high-resolution schemes in a finite element code. The theory of hyperbolic conservation laws was occasionally revisited to make the exposition as self-contained as possible. At the same time, emphasis was laid on various numerical aspects including but not limited to

- edge-by-edge matrix assembly for implicit Galerkin discretizations;
- construction of approximate Riemann solvers on unstructured grids;
- flux correction, synchronization and the choice of control variables;
- segregated versus coupled solution methods for algebraic systems;
- implementation of characteristic boundary conditions.

Our algebraic FCT and TVD methods were shown to prove their worth in the systems arena. However, there are still many open problems and unexplored areas that call for further research. In particular, characteristic flux limiters would benefit from a fully multidimensional wave model. For implicit time-stepping to be profitable, the combination of linear algebra tools should be adapted to the underlying discretization. A strongly coupled iterative solver can be developed in the framework of the FMG-FAS multigrid procedure which lends itself to the treatment of the stationary Euler equations [16],[26]. An analog of the local MPSC smoother [43] offers a promising alternative to the block-iterative or 'collective' (pointwise-coupled) Gauß-Seidel relaxation. The marriage of algebraic flux correction and high-performance computing techniques for large-scale applications is discussed in the next chapter.

Pressure isolines, FEM-TVD scheme, P_1 elements



Pressure isolines, FEM-TVD scheme, Q_1 elements

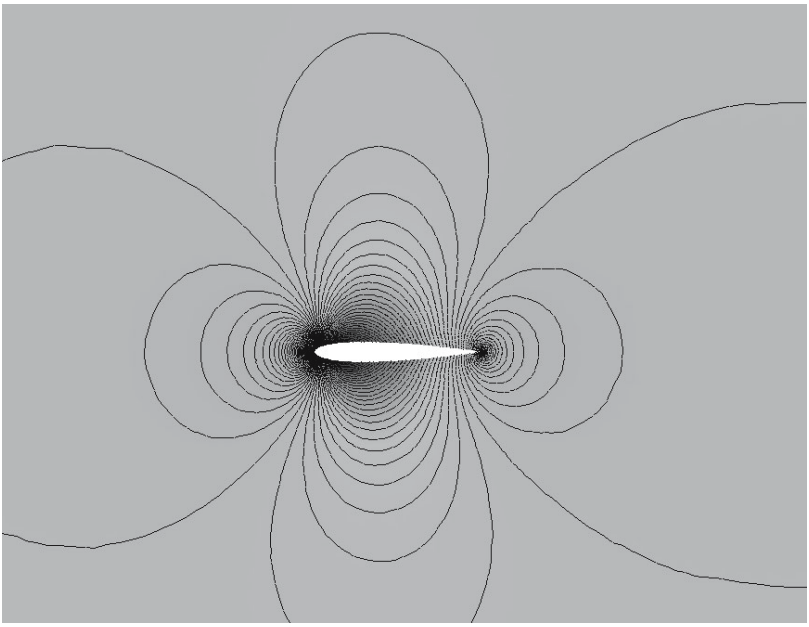
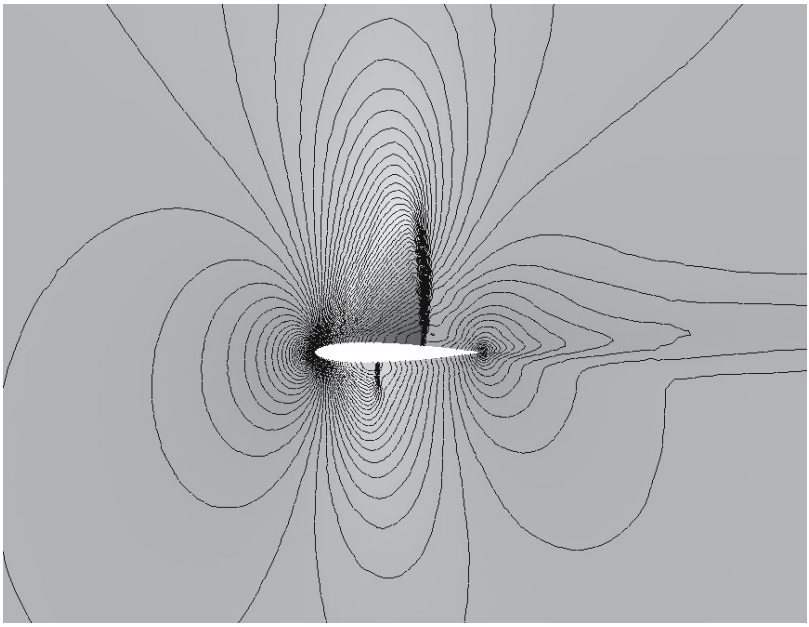


Fig. 13. Flow past a NACA 0012 airfoil at $M_\infty = 0.5,\ \alpha = 0^\circ$

Mach number isolines, FEM-TVD scheme, P_1 elements



Mach number isolines, FEM-TVD scheme, Q_1 elements

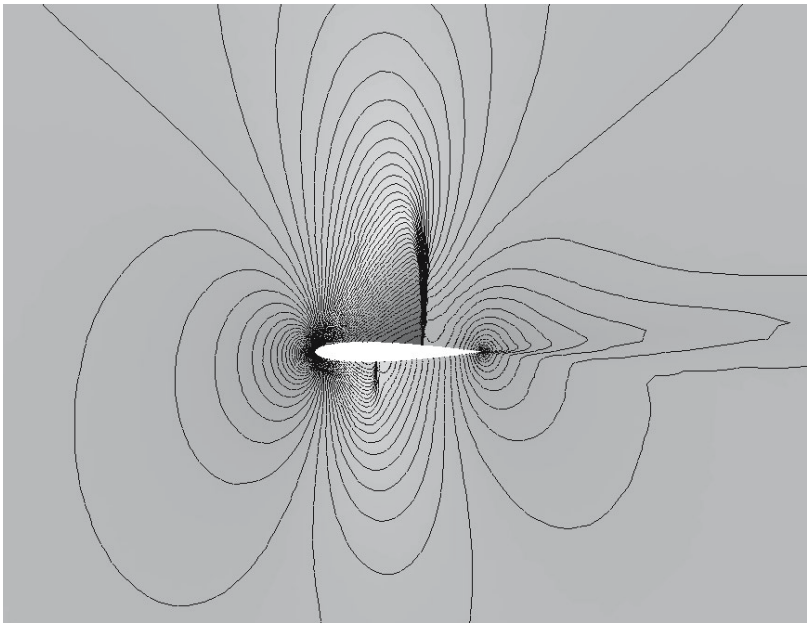
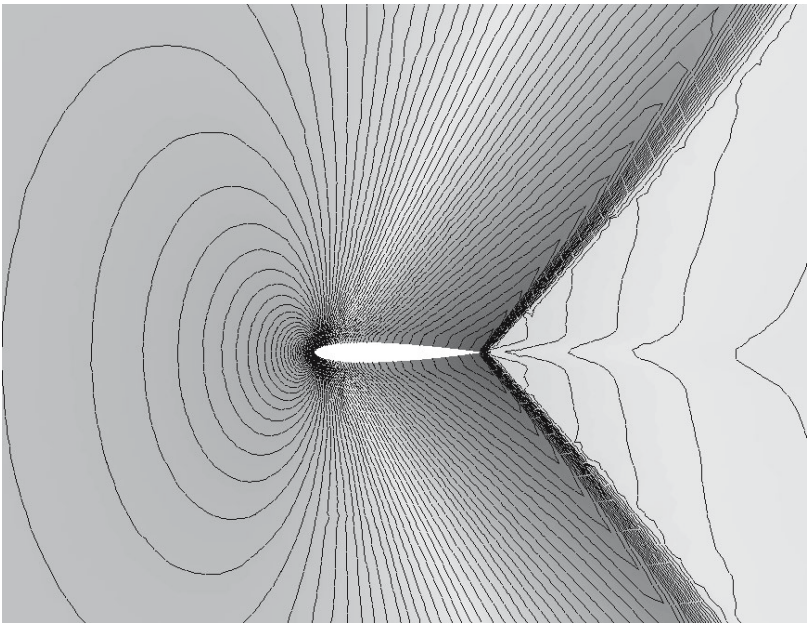


Fig. 14. Flow past a NACA 0012 airfoil at $M_\infty = 0.8,\ \alpha = 1.25^\circ$

Mach number isolines, FEM-TVD scheme, P_1 elements



Mach number isolines, FEM-TVD scheme, Q_1 elements

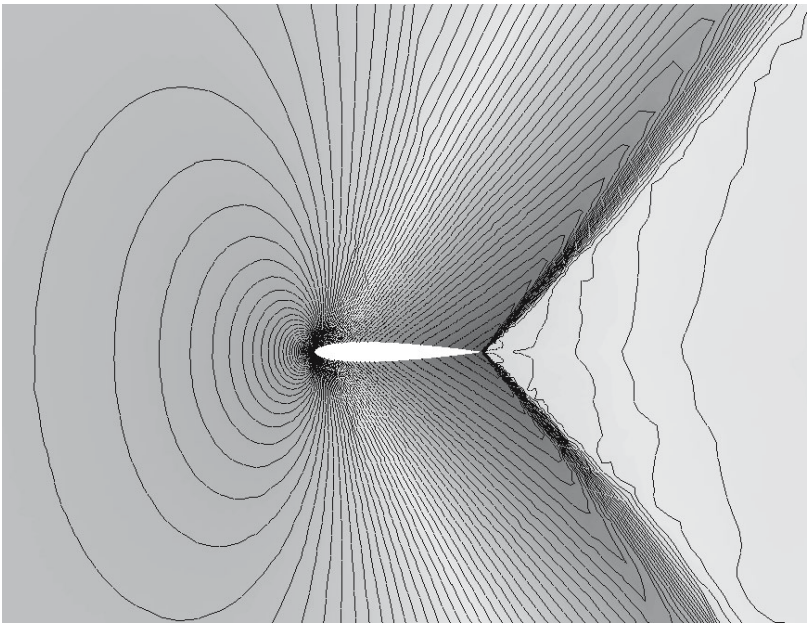


Fig. 15. Flow past a NACA 0012 airfoil at $M_\infty = 0.95,\ \alpha = 0^\circ$

A. Characteristic Boundary Conditions

In this appendix, we elaborate on the implementation of numerical boundary conditions for the Euler equations. The necessary background is presented in section 9. Let us omit the subscript i and denote the predicted values of the density, momentum, and total energy by ρ_*, $(\rho\mathbf{v})_*$ and $(\rho E)_*$, respectively. The velocity $\mathbf{v}_*$, pressure p_*, and speed of sound c_* are given by

$$\mathbf{v}_* = \frac{(\rho\mathbf{v})_*}{\rho_*}, \qquad p_* = (\gamma-1)\left[(\rho E)_* - \rho_*\frac{|\mathbf{v}_*|^2}{2}\right], \qquad c_* = \sqrt{\frac{\gamma p_*}{\rho_*}}.$$

These auxiliary quantities are required for the computation of the provisional Riemann invariants W^* from an analog of (88). The correction to be performed depends on the number and type of physical boundary conditions.

Supersonic open boundary

At a supersonic inlet, the boundary values of all conservative variables can be prescribed without conversion to the Riemann invariants. At a supersonic outlet, no boundary conditions at all are required.

Subsonic inflow boundary

The analytical values of $w_1^{**}, \ldots, w_4^{**}$ are obtained from the physical boundary conditions. The first two Riemann invariants are plugged into the relations

$$w_1^{**} = \mathbf{n}\cdot\mathbf{v}_{**} - \frac{2c_{**}}{\gamma-1}, \qquad w_2^{**} = c_v \log\left(\frac{p_{**}}{\rho_{**}^{\gamma}}\right)$$

for the new boundary values of the normal velocity, speed of sound, pressure, and density. The second pair of parameters yields the tangential velocity

$$\mathbf{v}_\tau = \mathbf{v}_\tau(w_3^{**}, w_4^{**}).$$

The outgoing Riemann invariant retains its predicted value w_5^* so that

$$w_5^{**} = \mathbf{n}\cdot\mathbf{v}_{**} + \frac{2c_{**}}{\gamma-1} = \mathbf{n}\cdot\mathbf{v}_* + \frac{2c_*}{\gamma-1} = w_5^*.$$

This gives a system of five equations for five independent variables. Specifically, the components of the vector U^{**} can be recovered as follows

$$c_{**} = (\gamma-1)\frac{w_5^* - w_1^{**}}{4}, \qquad \rho_{**} = \left[\frac{c_{**}^2}{\gamma}\exp(-w_2^{**}/c_v)\right]^{\frac{1}{\gamma-1}},$$

$$\mathbf{v}_{**} = \mathbf{v}_\tau + \frac{w_1^{**} + w_5^*}{2}\,\mathbf{n}, \qquad (\rho\mathbf{v})_{**} = \rho_{**}\mathbf{v}_{**},$$

$$(\rho E)_{**} = \rho_{**}\left[\frac{c_{**}^2}{\gamma(\gamma-1)} + \frac{|\mathbf{v}_{**}|^2}{2}\right], \qquad \mathrm{U}^{**} = [\rho_{**}, (\rho\mathbf{v})_{**}, (\rho E)_{**}]^T.$$

Subsonic outflow boundary

The incoming Riemann invariant w_1^* is overwritten by the imposed boundary condition $w_1^{**} = \mathbf{n} \cdot \mathbf{v}_{**} - \frac{2c_{**}}{\gamma-1}$. All other characteristic variables retain their boundary values computed at the predictor step. In particular, we have

$$w_2^{**} = c_v \log(p_{**}/\rho_{**}^{\gamma}) = c_v \log(p_*/\rho_*^{\gamma}) = w_2^*,$$

$$w_5^{**} = \mathbf{n} \cdot \mathbf{v}_{**} + \frac{2c_{**}}{\gamma-1} = \mathbf{n} \cdot \mathbf{v}_* + \frac{2c_*}{\gamma-1} = w_5^*$$

and the tangential velocity $\mathbf{v}_\tau$ is defined by w_3^* and w_4^*. This information is sufficient to compute the nodal values of the conservative variables

$$c_{**} = (\gamma-1)\frac{w_5^* - w_1^{**}}{4}, \qquad \rho_{**} = \left[\frac{c_{**}^2}{\gamma}\exp(-w_2^*/c_v)\right]^{\frac{1}{\gamma-1}},$$

$$\mathbf{v}_{**} = \mathbf{v}_* + \left[\frac{w_1^{**} + w_5^*}{2} - \mathbf{n} \cdot \mathbf{v}_*\right]\mathbf{n}, \qquad (\rho\mathbf{v})_{**} = \rho_{**}\mathbf{v}_{**},$$

$$(\rho E)_{**} = \rho_{**}\left[\frac{c_{**}^2}{\gamma(\gamma-1)} + \frac{|\mathbf{v}_{**}|^2}{2}\right], \qquad \mathrm{U}^{**} = [\rho_{**}, (\rho\mathbf{v})_{**}, (\rho E)_{**}]^T.$$

As a rule, the pressure p_{**} rather than the Riemann invariant w_1^{**} is specified at a subsonic outlet. In this case, the transformation formulae read

$$\rho_{**} = \rho_*\left(\frac{p_{**}}{p_*}\right)^{\frac{1}{\gamma}}, \qquad c_{**} = \sqrt{\frac{\gamma p_{**}}{\rho_{**}}},$$

$$\mathbf{v}_{**} = \mathbf{v}_* + \frac{2(c_* - c_{**})}{\gamma-1}\mathbf{n}, \qquad (\rho\mathbf{v})_{**} = \rho_{**}\mathbf{v}_{**},$$

$$(\rho E)_{**} = \frac{p_{**}}{\gamma-1} + \rho_{**}\frac{|\mathbf{v}_{**}|^2}{2}, \qquad \mathrm{U}^{**} = [\rho_{**}, (\rho\mathbf{v})_{**}, (\rho E)_{**}]^T.$$

Solid wall boundary

At an impervious solid wall, the normal velocity v_n must vanish. Taking this physical boundary condition into account, we obtain the expression

$$w_5^{**} = \frac{2c_{**}}{\gamma-1} = \mathbf{n} \cdot \mathbf{v}_* + \frac{2c_*}{\gamma-1} = w_5^*.$$

Furthermore, $v_n = 0$ implies that the entropy and the tangential velocity components do not cross the boundary. Therefore, their original values should not be corrected and the resulting conversion rules are as follows

$$c_{**} = c_* + \frac{\gamma-1}{2}\mathbf{n} \cdot \mathbf{v}_*, \qquad \rho_{**} = \left[\frac{c_{**}^2}{\gamma}\frac{\rho_*^{\gamma}}{p_*}\right]^{\frac{1}{\gamma-1}},$$

$$\mathbf{v}_{**} = \mathbf{v}_* - (\mathbf{n} \cdot \mathbf{v}_*)\mathbf{n}, \qquad (\rho\mathbf{v})_{**} = \rho_{**}\mathbf{v}_{**},$$

$$(\rho E)_{**} = \rho_{**}\left[\frac{c_{**}^2}{\gamma(\gamma-1)} + \frac{|\mathbf{v}_{**}|^2}{2}\right], \qquad \mathrm{U}^{**} = [\rho_{**}, (\rho\mathbf{v})_{**}, (\rho E)_{**}]^T.$$

The second line in this group of formulae corresponds to a projection of the predicted momentum onto the tangent plane. This modification is sufficient to enforce the free slip boundary condition $\mathbf{n} \cdot \mathbf{v}_{**} = 0$. Alternatively, it can be applied to the defect for the momentum equation before computing U^* from (90). In either case, the correction of ρ_* and $(\rho E)_*$ is optional.

If the initial data fail to satisfy the free slip condition, the abrupt change of the normal velocity results in an *impulsive start* which is physically impossible due to inertia. Hence, various numerical difficulties and poor convergence are to be expected. This problem was investigated by Lyra [27],[31] who proposed the following relaxation procedure for the freestream condition

$$\mathbf{n} \cdot \mathbf{v}^{n+1} = (1 - \omega)\mathbf{n} \cdot \mathbf{v}^n, \qquad 0 \leq \omega < 1.$$

In essence, the fluid is allowed to seep through the wall during the startup but the normal velocity is gradually driven to zero as the flow evolves. The relaxation parameter ω determines the rate at which it is adjusted.

Acknowledgements

The authors would like to thank Stefan Turek for fruitful discussions and Jens Acker for assistance with mesh generation.

References

1. J. D. Anderson, Jr., *Modern Compressible Flow*, McGraw-Hill, 1990.
2. P. Arminjon and A. Dervieux, Construction of TVD-like artificial viscosities on 2-dimensional arbitrary FEM grids. *INRIA Research Report* **1111** (1989).
3. J.-C. Carette, H. Deconinck, H. Paillère and P. L. Roe, Multidimensional upwinding: Its relation to finite elements. *Int. J. Numer. Methods Fluids* **20** (1995) no. 8-9, 935-955.
4. G. G. Cherny, *Gas Dynamics* (in Russian). Nauka, Moscow, 1988.
5. R. Codina and M. Cervera, Block-iterative algorithms for nonlinear coupled problems. In: M. Papadrakakis and G. Bugeda (eds), *Advanced Computational Methods in Structural Mechanics*, Chapter 7. Theory and Engineering Applications of Computational Methods, CIMNE, Barcelona, 1996.
6. D. L. Darmofal and B. Van Leer, Local preconditioning: Manipulating mother nature to fool father time. In: D. A. Caughey et al. (eds), *Frontiers of Computational Fluid Dynamics*, Singapore: World Scientific Publishing, 1998, 211-239.
7. H. Deconinck, H. Paillère, R. Struijs and P.L. Roe, Multidimensional upwind schemes based on fluctuation-splitting for systems of conservation laws. *Comput. Mech.* **11** (1993) no. 5-6, 323-340.
8. M. J. Díaz, F. Hecht and B. Mohammadi, New progress in anisotropic grid adaptation for inviscid and viscous flows simulations. In: *Proceedings of the 4th Annual International Meshing Roundtable*. Sandia National Laboratories, 1995.
9. J. Donea and A. Huerta, *Finite Element Methods for Flow Problems*. Wiley, 2003.

10. J. Donea, V. Selmin and L. Quartapelle, Recent developments of the Taylor-Galerkin method for the numerical solution of hyperbolic problems. *Numerical methods for fluid dynamics III*, Oxford, 171-185 (1988).
11. M. Feistauer, J. Felcman and I. Straškraba, *Mathematical and Computational Methods for Compressible Flow*. Clarendon Press, Oxford, 2003.
12. J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*. Springer, 1996.
13. C. A. J. Fletcher, *Computational Techniques for Fluid Dynamics*. Springer, 1988.
14. A. Harten and J. M. Hyman, Self adjusting grid methods for one-dimensional hyperbolic conservation laws. *J. Comput. Phys.* **50** (1983) 235–269.
15. R. Hartmann and P. Houston, Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *J. Comput. Phys.* **183** (2002) 508-532.
16. P. W. Hemker and B. Koren, Defect correction and nonlinear multigrid for steady Euler equations. In: W. G. Habashi and M. M. Hafez (eds), *Computational Fluid Dynamics Techniques.* London: Gordon and Breach Publishers, 1995, 699–718.
17. C. Hirsch, *Numerical Computation of Internal and External Flows. Vol. II: Computational Methods for Inviscid and Viscous Flows*. John Wiley & Sons, Chichester, 1990.
18. D. Kuzmin, M. Möller and S. Turek, Multidimensional FEM-FCT schemes for arbitrary time-stepping. *Int. J. Numer. Meth. Fluids* **42** (2003) 265-295.
19. D. Kuzmin, M. Möller and S. Turek, High-resolution FEM-FCT schemes for multidimensional conservation laws. Technical report **231**, University of Dortmund, 2003. To appear in: *Comput. Methods Appl. Mech. Engrg.*
20. R. J. LeVeque, *Numerical Methods for Conservation Laws*. Birkhäuser, 1992.
21. R. J. LeVeque, Simplified multi-dimensional flux limiting methods. *Numerical Methods for Fluid Dynamics* **IV** (1993) 175–190.
22. R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2003.
23. R. J. LeVeque, CLAWPACK – Conservation LAWs PACKage, available at the URL *http://www.amath.washington.edu/∼claw/*.
24. R. Löhner, K. Morgan, J. Peraire and M. Vahdati, Finite element flux-corrected transport (FEM-FCT) for the Euler and Navier-Stokes equations. *Int. J. Numer. Meth. Fluids* **7** (1987) 1093–1109.
25. R. Löhner, *Applied CFD Techniques: An Introduction Based on Finite Element Methods*. Wiley, 2001.
26. J. F. Lynn, *Multigrid Solution of the Euler Equations with Local Preconditioning.* PhD thesis, University of Michigan, 1995.
27. P. R. M. Lyra, *Unstructured Grid Adaptive Algorithms for Fluid Dynamics and Heat Conduction.* PhD thesis, University of Wales, Swansea, 1994.
28. P. R. M. Lyra, K. Morgan, J. Peraire and J. Peiro, TVD algorithms for the solution of the compressible Euler equations on unstructured meshes. *Int. J. Numer. Meth. Fluids* **19** (1994) 827–847.
29. P. R. M. Lyra and K. Morgan, A review and comparative study of upwind biased schemes for compressible flow computation. I: 1-D first-order schemes. *Arch. Comput. Methods Eng.* **7** (2000) no. 1, 19-55.
30. P. R. M. Lyra and K. Morgan, A review and comparative study of upwind biased schemes for compressible flow computation. II: 1-D higher-order schemes. *Arch. Comput. Methods Eng.* **7** (2000) no. 3, 333-377.

31. P. R. M. Lyra and K. Morgan, A review and comparative study of upwind biased schemes for compressible flow computation. III: Multidimensional extension on unstructured grids. *Arch. Comput. Methods Eng.* **9** (2002) no. 3, 207-256.
32. M. Möller, *Hochauflösende FEM-FCT-Verfahren zur Diskretisierung von konvektionsdominanten Transportproblemen mit Anwendung auf die kompressiblen Eulergleichungen*. Diploma thesis, University of Dortmund, 2003.
33. M. Möller, D. Kuzmin and S. Turek, Implicit FEM-FCT algorithm for compressible flows. To appear in: Proceedings of the *European Conference on Numerical Mathematics and Advanced Applications* (August 18-22, 2003, Prague), to be published by Springer.
34. K. Morgan and J. Peraire, Unstructured grid finite element methods for fluid mechanics. *Reports on Progress in Physics*, **61** (1998), no. 6, 569-638.
35. NPARC Alliance, Computational Fluid Dynamics (CFD) Verification and Validation Web Site: *http://www.grc.nasa.gov/WWW/wind/valid/*.
36. P. L. Roe, Approximate Riemann solvers, parameter vectors and difference schemes. *J. Comput. Phys.* **43** (1981) 357–372.
37. A. Rohde, Eigenvalues and eigenvectors of the Euler equations in general geometries. *AIAA Paper* 2001-2609.
38. V. Selmin, Finite element solution of hyperbolic equations. I. One-dimensional case. *INRIA Research Report* **655** (1987).
39. V. Selmin, Finite element solution of hyperbolic equations. II. Two-dimensional case. *INRIA Research Report* **708** (1987).
40. D. Sidilkover, A genuinely multidimensional upwind scheme and efficient multigrid solver for the compressible Euler equations, *ICASE Report* No. 94-84, 1994.
41. G. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Comput. Phys.* **27** (1978) 1–31.
42. E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer, 1999.
43. S. Turek, *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*, LNCSE 6, Springer, 1999.
44. P. Wesseling, *Principles of Computational Fluid Dynamics*. Springer, 2001.
45. H. C. Yee, Numerical approximations of boundary conditions with applications to inviscid gas dynamics. *NASA report* TM-81265, 1981.
46. H. C. Yee, Construction of explicit and implicit symmetric TVD schemes and their applications. *J. Comput. Phys.* **43** (1987) 151–179.
47. H. C. Yee, R. F. Warming and A. Harten, Implicit Total Variation Diminishing (TVD) schemes for steady-state calculations. *J. Comput. Phys.* **57** (1985) 327–360.
48. S. T. Zalesak, *The Design of Flux-Corrected Transport (FCT) Algorithms For Structured Grids*. Chapter 2 in this volume.

Algebraic Flux Correction III. Incompressible Flow Problems

Stefan Turek[1] and Dmitri Kuzmin[2]

[1] Institute of Applied Mathematics (LS III), University of Dortmund
Vogelpothsweg 87, D-44227, Dortmund, Germany
Stefan.Turek@mathematik.uni-dortmund.de
[2] kuzmin@math.uni-dortmund.de

Summary. Algebraic FEM-FCT and FEM-TVD schemes are integrated into incompressible flow solvers based on the 'Multilevel Pressure Schur Complement' (MPSC) approach. It is shown that algebraic flux correction is feasible for nonconforming (rotated bilinear) finite element approximations on unstructured meshes. Both (approximate) operator-splitting and fully coupled solution strategies are introduced for the discretized Navier-Stokes equations. The need for development of robust and efficient iterative solvers (outer Newton-like schemes, linear multigrid techniques, optimal smoothers/preconditioners) for implicit high-resolution schemes is emphasized. Numerical treatment of extensions (Boussinesq approximation, $k-\varepsilon$ turbulence model) is addressed and pertinent implementation details are given. Simulation results are presented for three-dimensional benchmark problems as well as for prototypical applications including multiphase and granular flows.

1 Introduction

For single-phase Newtonian fluids occupying a domain $\Omega \subset \mathbf{R}^d$ $(d = 2, 3)$ during the time interval $(t_0, t_0+T]$, the incompressible Navier-Stokes equations

$$\begin{aligned} &\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f} \quad , \\ &\nabla \cdot \mathbf{u} = 0 \end{aligned} \tag{1}$$

describe the laminar flow motion which depends on the physical properties of the fluid (viscosity ν) and, possibly, on some external forces $\mathbf{f}$ like buoyancy. The constant density ρ is "hidden" in the pressure $p(x_1, \ldots, x_d, t)$ which adjusts itself instantaneously so as to render the time-dependent velocity field $\mathbf{u}(x_1, \ldots, x_d, t)$ divergence-free. The problem statement is completed by specifying the initial and boundary values for each particular application.

Although these equations seem to have a quite simple structure, they constitute a 'grand challenge' problem for mathematicians, physicists, and engineers alike, and they are (still) object of intensive research activities in the field of *Computational Fluid Dynamics* (CFD). Incompressible flow problems are especially interesting from the viewpoint of applied mathematics and scientific computing, since they embody the whole range of difficulties which typically arise in the numerical treatment of partial differential equations. Therefore, they provide a perfect starting point for the development of reliable numerical algorithms and efficient software for CFD simulations.

Specifically, the problems which scientists and engineers are frequently confronted with, concern the following aspects:

- time-dependent partial differential equations in complex domains
- strongly nonlinear and stiff systems of intricately coupled equations
- convection-dominated transport at high Reynolds numbers ($Re \approx \frac{1}{\nu}$)
- saddle–point problems due to the incompressibility constraint
- local changes of the problem character in space and time

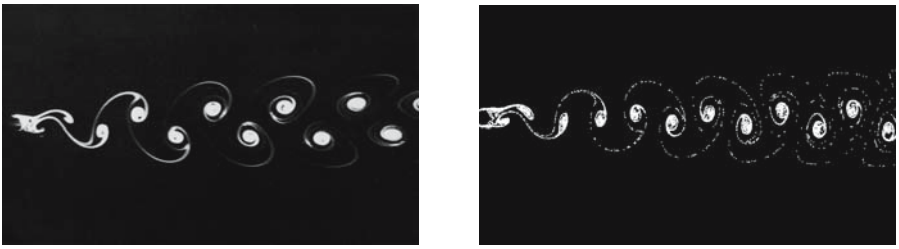


Fig. 1. Experiment (source: Van Dyke's 'Album of Fluid Motion') vs. numerical simulation (source: 'Virtual Album of Fluid Motion') for flow around a cylinder

These peculiarities of our model problem impose stringent requirements on virtually all stages of algorithm design: discretization, solver, and software engineering. In particular, the following difficulties must be reckoned with

- nonlinear systems for millions of unknowns (large but sparse matrices)
- conditional stability (explicit schemes) and/or proper time step control
- anisotropic/unstructured meshes (boundary layers, complex geometries)

Active research aimed at the development of improved numerical methods for the incompressible Navier-Stokes equations has been going on for more than three decades. The number of publications on this topic is enormous (see the book by Gresho et al. [14] for a comprehensive overview). However, in many cases the computational results produced by the available CFD tools are only *qualitatively* correct. A *quantitatively* precise flow prediction for real-life problems requires that the accuracy of discretization schemes be enhanced and/or the solvers become more efficient. This can be easily demonstrated by benchmark computations [40], especially for nonstationary flows.

Moreover, a current trend in CFD is to combine the 'basic' Navier-Stokes equations (2) with more or less sophisticated engineering models from physics and chemistry which describe industrial applications involving turbulence, multiphase flow, nonlinear fluids, combustion/detonation, free and moving boundaries, fluid-structure interaction, weakly compressible effects, etc. These extensions, some which will be discussed in the present chapter, have one thing in common: they require highly accurate *and* robust discretization techniques as well as efficient solution algorithms for generalized Navier–Stokes–like systems. In order to design and implement such powerful numerical methods for real-life problems, many additional aspects need to be taken into account.

The main ingredients of an 'ultimate' CFD code are as follows

- advanced mathematical methods for PDEs ($\rightarrow$ *discretization*)
- efficient solution techniques for algebraic systems ($\rightarrow$ *solver*)
- reliable and hardware-optimized software ($\rightarrow$ *implementation*)

If all of these components were available, the number of unknowns could be significantly reduced (e.g., via adaptivity or a high-order approximation) and, moreover, discrete problems of the same size could be solved more efficiently. Hence, the marriage of optimal numerical methods and fast iterative solvers would make it possible to exploit the potential of modern computers to the full extent and enhance the performance of incompressible flow solvers (improve the MFLOP/s rates) by *orders of magnitude*. This is why these algorithmic aspects play an increasingly important role in contemporary CFD research.

In this contribution, we briefly review the Multilevel Pressure Schur Complement (MPSC) approach to solution of the incompressible Navier-Stokes equations and combine it with a FEM-FCT or FEM-TVD discretization of the (nonlinear) convective terms. We will explain the ramifications of these new (for the FEM community) algebraic high-resolution schemes as applied to incompressible flow problems and discuss a number of computational details regarding the efficient numerical solution of the resulting nonlinear and linear algebraic systems. Furthermore, we will examine different coupling mechanisms between the 'basic' flow model (standard Navier-Stokes equations for the velocity and pressure) and additional scalar or vector-valued transport equations. The primary goal is numerical simulation of high Reynolds number flows which require special stabilization techniques and/or advanced turbulence models in order to capture the relevant physical effects.

On the other hand, we also consider incompressible flows at intermediate and low Reynolds numbers. They may call for a different solution strategy but the evolution of scalar variables (temperatures, concentrations, probability densities, volume fractions, level set functions etc.) is still dominated by transport operators. Moreover, the transported quantities are inherently nonnegative in many cases. Therefore, standard discretization techniques may fail, whereas the positivity-preserving FCT/TVD schemes persevere and yield excellent results as the numerical examples in this chapter will illustrate.

2 Discretization of the Navier-Stokes Equations

Let us discretize the Navier–Stokes equations (2) in time by a standard method for numerical solution of ODEs. For instance, an implicit θ–scheme (backward Euler or Crank-Nicolson) or its three-step counterpart proposed by Glowinski yields a sequence of boundary value problems of the form [47]

Given $\mathbf{u}(t_n)$*, compute* $\mathbf{u} = \mathbf{u}(t_{n+1})$ *and* $p = p(t_{n+1})$ *by solving*

$$\begin{aligned}[I + \theta\Delta t(\mathbf{u}\cdot\nabla - \nu\Delta)]\mathbf{u} + \Delta t\nabla p &= [I - \theta_1\Delta t(\mathbf{u}(t_n)\cdot\nabla - \nu\Delta)]\mathbf{u}(t_n) \\ &+ \theta_2\Delta t\mathbf{f}(t_{n+1}) + \theta_3\Delta t\mathbf{f}(t_n) \end{aligned} \tag{2}$$

subject to the incompressibility constraint $\nabla\cdot\mathbf{u} = 0$.

For the spatial discretization, we choose a finite element approach. However finite volumes, finite differences or spectral methods are possible, too. A finite element model of the Navier–Stokes equations is based on a suitable variational formulation. On the finite mesh $\mathcal{T}_h$ (triangles, quadrilaterals or their analogues in 3D) covering the domain Ω with local mesh size h, one defines polynomial trial functions for velocity and pressure. These spaces H_h and L_h should lead to numerically stable approximations as $h \to 0$, i.e., they should satisfy the so-called *Babuška–Brezzi* (BB) condition [12]

$$\min_{q_h\in L_h}\max_{\mathbf{v}_h\in H_h}\frac{(q_h, \nabla\cdot\mathbf{v}_h)}{\|q_h\|_0\,\|\nabla\mathbf{v}_h\|_0} \geq \gamma > 0 \tag{3}$$

with a mesh–independent constant γ. On the other hand, equal order interpolations for velocity and pressure are also admissible provided that an a priori unstable discretization is stabilized in an appropriate way (see, e.g., [19]).

In what follows, we employ the stable $\tilde{Q}_1/Q_0$ finite element pair (*rotated bilinear/trilinear* shape functions for the velocities, and a piecewise constant pressure approximation). In the two-dimensional case, the nodal values are the mean values of the velocity vector over the element edges, and the mean values of the pressure over the elements (see Fig. 2).

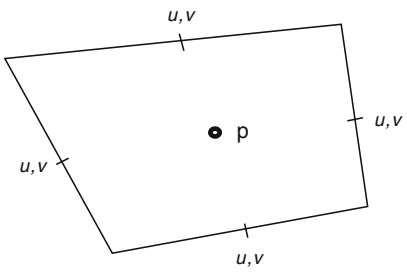


Fig. 2. Nodal points of the nonconforming finite element pair $\tilde{Q}_1/Q_0$

This **nonconforming** finite element is a quadrilateral counterpart of the well–known triangular Stokes element of Crouzeix–Raviart [6] and can easily be defined in three space dimensions. A convergence analysis is given in [39] and computational results are reported in [42] and [43]. An important advantage of this finite element pair is the availability of efficient multigrid solvers which are sufficiently robust in the whole range of Reynolds numbers even on nonuniform and highly anisotropic meshes [41],[47].

Using the notation $\mathbf{u}$ and p also for the coefficient vectors in the representation of the approximate solution, the discrete version of problem (2) may be written as a coupled (nonlinear) algebraic system of the form:

Given $\mathbf{u}^n$ *and* $\mathbf{g}$*, compute* $\mathbf{u} = \mathbf{u}^{n+1}$ *and* $p = p^{n+1}$ *by solving*

$$A\mathbf{u} + \Delta t B p = \mathbf{g} \quad , \quad B^T \mathbf{u} = 0, \qquad \text{where} \tag{4}$$

$$\mathbf{g} = [M - \theta_1 \Delta t N(\mathbf{u}^n)]\mathbf{u}^n + \theta_2 \Delta t \mathbf{f}^{n+1} + \theta_3 \Delta t \mathbf{f}^n \, . \tag{5}$$

Here M is the (consistent or lumped) mass matrix, B is the discrete gradient operator, and $-B^T$ is the associated divergence operator. Furthermore,

$$A\mathbf{u} = [M - \theta \Delta t N(\mathbf{u})]\mathbf{u}, \qquad N(\mathbf{u}) = K(\mathbf{u}) + \nu L, \tag{6}$$

where L is the discrete Laplacian and $K(\mathbf{u})$ is the nonlinear transport operator incorporating a certain amount of artificial diffusion. In the sequel, we assume that it corresponds to a FEM-TVD discretization of the convective term, although algebraic flux correction of FCT type or conventional stabilization techniques (upwinding, streamline diffusion) are also feasible.

The solution of nonlinear algebraic systems like (4) is a rather difficult task and many aspects need to be taken into account:

- **treatment of the nonlinearity:** fully nonlinear solution by Newton–like methods or iterative defect correction, explicit or implicit underrelaxation, monitoring of convergence rates, choice of stopping criteria etc.
- **treatment of the incompressibility:** strongly coupled approach (simultaneous solution for $\mathbf{u}$ and p) vs. segregated algorithms based on operator splitting at the continuous or discrete level (classical projection schemes [3],[56] and pressure correction methods like SIMPLE [8],[34]).
- **complete outer control:** problem-dependent degree of coupling and/or implicitness, optimal choice of linear algebra tools (iterative solvers and underlying smoothers/preconditioners), automatic time step control etc.

This abundance of choices leads to a great variety of incompressible flow solvers which are closely related to one another but exhibit considerable differences in terms of their stability, convergence, and efficiency. The Multilevel Pressure Schur Complement (MPSC) approach outlined below makes it possible to put many existing solution techniques into a common framework and combine their advantages so as to obtain better run-time characteristics. For a detailed presentation and a numerical study of the resulting schemes, the interested reader is referred to the monograph by Turek [47].

3 Pressure Schur Complement Solvers

The fully discretized Navier-Stokes equations (4) as well as the linear subproblems to be solved within the outer iteration loop for a fixed-point defect correction or a Newton-like method admit the following representation

$$\begin{bmatrix} A & \Delta t B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ 0 \end{bmatrix}. \tag{7}$$

This is a saddle point problem in which the pressure acts as the Lagrange multiplier for the incompressibility constraint. In general, we have

$$A = \alpha M + \beta N(\mathbf{u}), \qquad \text{where} \quad \beta = -\theta \Delta t. \tag{8}$$

For time-dependent problems, the parameter α is set equal to unity, whereas the steady-state formulation is recovered for $\alpha := 0$ and $\beta := -1$.

If the operator A is nonsingular, the velocity can be formally expressed as

$$\mathbf{u} = A^{-1}(\mathbf{g} - \Delta t B p) \tag{9}$$

and plugged into the discretized continuity equation

$$B^T \mathbf{u} = 0 \tag{10}$$

which gives a scalar *Schur complement* equation for the pressure

$$B^T A^{-1} B p = \frac{1}{\Delta t} B^T A^{-1} \mathbf{g}. \tag{11}$$

Thus, the coupled system (7) can be handled as follows

1. Solve the Pressure Schur Complement (PSC) equation (11) for p.
2. Substitute p into relation (9) and compute the velocity $\mathbf{u}$.

It is worth mentioning that the matrix A^{-1} is full and should not be assembled explicitly. Instead, an auxiliary problem is to be solved by a direct method or by inner iterations. For instance, the velocity update (9) is equivalent to the solution of the discretized momentum equation $A\mathbf{u} = \mathbf{g} - \Delta t B p$.

Likewise, the matrix $S := B^T A^{-1} B$ is never generated in practice. Doing so would be prohibitively expensive in terms of CPU time and memory requirements. It is instructive to consider a preconditioned Richardson method which yields the following **basic iteration** for the PSC equation

$$p^{(l+1)} = p^{(l)} - C^{-1} \left[S p^{(l)} - \frac{1}{\Delta t} B^T A^{-1} \mathbf{g} \right], \qquad l = 0, \ldots, L-1. \tag{12}$$

Here C is a suitable preconditioner which is supposed to be a reasonable approximation to S but be easier to 'invert' in an iterative way.

The number of PSC cycles L can be fixed or chosen adaptively so as to achieve a prescribed tolerance for the residual. The choice $C := S$ and $L = 1$ is equivalent to the coupled solution of the original saddle-point problem (7). In principle, this challenging task can be accomplished by a properly configured multigrid method. However, the computational cost per iteration is very high and severe problems are sometimes observed on anisotropic grids that contain cells with high aspect ratios. Moreover, the convergence rates do not improve as the time step Δt is refined. Indeed, note that

$$A = M - \theta \Delta t N(\mathbf{u}) \approx M + \mathcal{O}(\Delta t) \tag{13}$$

for sufficiently small time steps. In this case, A can be interpreted as a nonsymmetric (and nonlinear) but well conditioned perturbation of the mass matrix M. On the other hand, the PSC equation (11) reveals that

$$\mathrm{cond}(S) = \mathrm{cond}(B^T[M + \mathcal{O}(\Delta t)]^{-1}B) \approx \mathrm{cond}(L) = \mathcal{O}(h^{-2})\,. \tag{14}$$

It follows that the condition number of the coupled system (7) is bounded from below by $\mathcal{O}(h^{-2})$ regardless of the time step. The nonsymmetric matrix A acting on the velocity components 'improves' for small Δt but, unfortunately, the overall convergence rates of coupled solvers depend also on the elliptic part $B^T A^{-1} B \approx B^T M^{-1} B$ which is (almost) time step invariant.

In light of the above, the coupled solution strategy is inappropriate for numerical simulation of nonstationary flows which are dominated by convection and call for the use of small time steps. Hence, the preconditioner C for the Schur complement operator should be designed so as to take relation (13) into account. Let us consider 'crude' approximations of the form

$$C := B^T \tilde{A}^{-1} B, \tag{15}$$

where the matrix $\tilde{A}$ should be readily 'invertible' but stay close to A at least in the limit $\Delta t \to 0$. Some typical choices are as follows

$$\tilde{A} := \mathrm{diag}(A), \quad \tilde{A} := M_L, \quad \text{and} \quad \tilde{A} := M - \theta \Delta t \nu L.$$

Incompressible flow solvers based on the Richardson iteration (12) with this sort of preconditioning comprise fractional-step projection methods [7],[15], [36],[46], various modifications of the SIMPLE algorithm (for an overview, see Engelman [8] and the literature cited therein) as well as Uzawa-like iterations. They can be classified as *global pressure Schur complement* schemes due to the fact that C is an approximation to the global matrix $S = B^T A^{-1} B$.

On the other hand, coupled solution techniques are to be recommended for the treatment of (quasi-) stationary flows and Navier-Stokes equations combined with RANS turbulence models and/or convection-diffusion equations for other scalar quantities (temperatures, concentrations etc). In this case, it is worthwhile to approximate the pressure Schur complement operator *locally*

via direct inversion of small matrix blocks associated with subdomains Ω_i of the domain Ω or, in general, with a subset of the unknowns to be solved for. The resulting *local pressure Schur complement* techniques correspond to

$$C^{-1} := \sum_i (B^T_{|\Omega_i} A^{-1}_{|\Omega_i} B_{|\Omega_i})^{-1}, \tag{16}$$

whereby the *patches* Ω_i are usually related to the underlying mesh and can consist of single elements or element clusters. The global relaxation scheme is obtained by embedding these "local solvers" into an outer iteration loop of Jacobi or Gauss–Seidel type. This strategy has a lot in common with *domain decomposition* methods, but is more flexible when it comes to the treatment of boundary conditions. A typical representative of local PSC schemes is the Vanka smoother [55] which is widely used in the multigrid community.

Furthermore, it is possible to combine incompressible flow solvers based on global PSC ("operator splitting") and local PSC ("domain decomposition") methods in a general-purpose CFD code. Indeed, all of these seemingly different solution techniques utilize *additive preconditioners* of the form

$$C^{-1} := \sum_i \alpha_i C_i^{-1}.$$

In the next two sections, we briefly discuss the design of such preconditioners and present the resulting basic iteration schemes which can be used as

- preconditioners for Krylov space methods (CG, BiCGSTAB, GMRES)
- *Multilevel pressure Schur complement* (MPSC) smoothers for multigrid

The multigrid approach is usually more efficient, as demonstrated by benchmark computations in [40] (see also the numerical examples below).

4 Global MPSC Approach

The basic idea behind the family of global MPSC schemes is the construction of globally defined additive preconditioners for the Schur complement operator $S = B^T A^{-1} B$. Recall that the matrix A has the following structure

$$A := \alpha M + \beta K(\mathbf{u}) + \gamma L, \tag{17}$$

where $\beta = -\theta \Delta t$ and $\gamma = \nu\beta$. Unfortunately, it is hardly possible to construct a matrix $\tilde{A}$ and a preconditioner $C = B^T \tilde{A}^{-1} B$ that would be a sufficiently good approximation to all three components of A and S, respectively. Therefore, one can start with developing individual preconditioners for the reactive (M), convective (K), and diffusive (L) part. In other words, the original problem can be decomposed into simpler tasks by resorting to operator splitting.

Let the inverse of C be composed from those of 'optimal' preconditioners for the limiting cases of a divergence–free L_2-projection (for small time steps), incompressible Euler equations, and a diffusion-dominated Stokes problem

$$C^{-1} = \alpha' C_M^{-1} + \beta' C_K^{-1} + \gamma' C_L^{-1} \approx S^{-1}, \tag{18}$$

where the user-defined parameters $(\alpha', \beta', \gamma')$ may toggle between (α, β, γ) and zero depending on the flow regime. Furthermore, it is implied that

C_M is an 'optimal' approximation of the reactive part $B^T M^{-1} B$,

C_K is an 'optimal' approximation of the convective part $B^T K^{-1} B$,

C_L is an 'optimal' approximation of the diffusive part $B^T L^{-1} B$.

The meaning of 'optimality' has to be defined more precisely. Ideally, partial preconditioners should be direct solvers with respect to the underlying subproblem. In fact, this may even be true for the fully 'reactive' case $S = B^T M^{-1} B$. However, if these preconditioners are applied as smoothers in a multigrid context and the convergence rates are largely independent of outer parameter settings as well as of the underlying mesh, then this is already sufficient for optimality of the global MPSC solver. Preconditioners C_M, C_K, and C_L satisfying this criterion are introduced and analyzed in [47].

At high Reynolds numbers, the time steps must remain small due to the physical scales of flow motion. Therefore, the lumped mass matrix M_L proves to be a reasonable approximation to the complete operator A. In this case, our basic iteration (12) for the pressure Schur complement equation

$$p^{(l+1)} = p^{(l)} + [B^T M_L^{-1} B]^{-1} \frac{1}{\Delta t} B^T A^{-1} \left[\mathbf{g} - \Delta t B p^{(l)} \right] \tag{19}$$

can be interpreted and implemented as a *discrete projection scheme* such as those proposed in [7],[15],[36]. The main algorithmic steps are as follows [46]

1. Solve the 'viscous Burgers' equation for $\tilde{\mathbf{u}}$

$$A\tilde{\mathbf{u}} = \mathbf{g} - \Delta t B p^{(l)}.$$

2. Solve the discrete 'Pressure-Poisson' problem

$$B^T M_L^{-1} B q = \frac{1}{\Delta t} B^T \tilde{\mathbf{u}}.$$

3. Correct the pressure and the velocity

$$p^{(l+1)} = p^{(l)} + q, \quad \mathbf{u} = \tilde{\mathbf{u}} - \Delta t M_L^{-1} B q.$$

In essence, the right-hand side of the momentum equation is assembled using the old pressure iterate and the intermediate velocity $\tilde{\mathbf{u}}$ is projected onto the subspace of solenoidal functions so as to satisfy the constraint $B^T \mathbf{u} = 0$.

The matrix $B^T M_L^{-1} B$ corresponds to a mixed discretization of the Laplacian operator [15] so that this method is a discrete analogue of the classical projection schemes derived by Chorin ($p^{(0)} = 0$) and Van Kan ($p^{(0)} = p(t_n)$) via operator splitting for the continuous problem [3],[56]. For an in-depth presentation of continuous projection schemes we refer to [35],[36]. Our discrete approach offers a number of important advantages including

- applicability to discontinuous pressure approximations
- consistent treatment (no splitting) of boundary conditions
- alleviation of spurious boundary layers for the pressure
- convergence to the fully coupled solution as l increases
- remarkable efficiency for nonstationary flow problems

On the other hand, discrete projection methods lack the inherent stabilization mechanisms that make their continuous counterparts applicable to equal-order interpolations provided that the time step is not too small [35].

In our experience, it is often sufficient to perform exactly $L = 1$ pressure Schur complement iteration with just one multigrid sweep. Due to the fact that the numerical effort for solving the linear subproblems is insignificant, global MPSC methods are much more efficient than coupled solvers in the high Reynolds number regime. However, they perform so well only for relatively small time steps, so that the more robust local MPSC schemes are to be recommended for low Reynolds number flows.

5 Local MPSC Approach

The local pressure Schur complement approach is tailored to solving 'small' problems so as to exploit the fast cache of modern processors, in contrast to the readily vectorizable global MPSC schemes. As already mentioned above, the basic idea is to subdivide the complete set of unknowns into patches Ω_i and solve the local subproblems **exactly** within an outer block-Gauss-Seidel/Jacobi iteration loop. Typically, every patch (macroelement) for this 'domain decomposition' method consists of one or several neighboring mesh cells and the corresponding local 'stiffness matrix' C_i is given by

$$C_i := \begin{bmatrix} \tilde{A}_{|\Omega_i} & \Delta t B_{|\Omega_i} \\ B^T_{|\Omega_i} & 0 \end{bmatrix}. \tag{20}$$

Its coefficients (and hence the corresponding 'boundary conditions' for the subdomains) are taken from the global matrices, whereby $\tilde{A}$ may represent either the complete velocity matrix A or some approximation of it, for instance, the diagonal part $diag(A)$. The local subproblems at hand are so small that they can be solved directly by Gaussian elimination. This is equivalent to applying the inverse of C_i to a portion of the global defect vector.

The elimination process leads to a fill-in of the matrix, which increases the storage requirements dramatically. Thus, it is advisable to solve the equivalent local pressure Schur complement problem with the compact matrix

$$S_i := B^T_{|\Omega_i} \tilde{A}^{-1}_{|\Omega_i} B_{|\Omega_i} \,. \tag{21}$$

In general, S_i is a full matrix but it is much smaller than C_i, since only the pressure values are solved for. If the patch Ω_i contains just a moderate number of elements, the pressure Schur complement matrix is likely to fit into the processor cache. Having solved the local PSC subproblem, one can recover the corresponding velocity field as described in the previous section.

In any case, the basic iteration for a local MPSC method reads

$$\begin{bmatrix} \mathbf{u}^{(l+1)} \\ p^{(l+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{u}^{(l)} \\ p^{(l)} \end{bmatrix} - \omega^{(l+1)} \sum_{i=1}^{N_p} \begin{bmatrix} \tilde{A}_{|\Omega_i} & \Delta t B_{|\Omega_i} \\ B^T_{|\Omega_i} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \delta\mathbf{u}_i^{(l)} \\ \delta p_i^{(l)} \end{bmatrix}, \tag{22}$$

where N_p denotes the total number of patches, $\omega^{(l+1)}$ is a relaxation parameter, and the global defect vector restricted to a single patch Ω_i is given by

$$\begin{bmatrix} \delta\mathbf{u}_i^{(l)} \\ \delta p_i^{(l)} \end{bmatrix} = \left(\begin{bmatrix} A & \Delta t B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(l)} \\ p^{(l)} \end{bmatrix} - \begin{bmatrix} \mathbf{g} \\ 0 \end{bmatrix} \right)_{|\Omega_i} . \tag{23}$$

In practice, we solve the corresponding auxiliary problem

$$\begin{bmatrix} \tilde{A}_{|\Omega_i} & \Delta t B_{|\Omega_i} \\ B^T_{|\Omega_i} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_i^{(l+1)} \\ q_i^{(l+1)} \end{bmatrix} = \begin{bmatrix} \delta\mathbf{u}_i^{(l)} \\ \delta p_i^{(l)} \end{bmatrix} \tag{24}$$

and compute the new iterates $\mathbf{u}^{(l+1)}_{|\Omega_i}$ and $p^{(l+1)}_{|\Omega_i}$ as follows

$$\begin{bmatrix} \mathbf{u}^{(l+1)}_{|\Omega_i} \\ p^{(l+1)}_{|\Omega_i} \end{bmatrix} = \begin{bmatrix} \mathbf{u}^{(l)}_{|\Omega_i} \\ p^{(l)}_{|\Omega_i} \end{bmatrix} - \omega^{(l+1)} \begin{bmatrix} \mathbf{v}_i^{(l+1)} \\ q_i^{(l+1)} \end{bmatrix} . \tag{25}$$

This two-step relaxation procedure is applied to each patch, so some velocity or pressure components may end up being updated several times. The easiest way to obtain globally defined solution values at subdomain boundaries is to overwrite the contributions of previously processed patches or to calculate an average over all patch contributions to the computational node.

The resulting local MPSC method corresponds to a simple block-Jacobi iteration for the mixed problem (4). Its robustness and efficiency can be easily enhanced by computing the local defect vector (23) using the possibly updated solution values rather than the old iterates $\mathbf{u}^{(l)}_{|\Omega_i}$ and $p^{(l)}_{|\Omega_i}$ for the degrees of freedom shared with other patches. This strategy is known as the block-Gauss-Seidel method. Its performance is superior to that of the block–Jacobi scheme, while the numerical effort is approximately the same (for a sequential code).

It is common knowledge that block-iterative methods of Jacobi and Gauss-Seidel type do a very good job as long as there are no strong mesh anisotropies. However, the convergence rates deteriorate dramatically for irregular triangulations which contain elements with high aspect ratios (for example, stretched cells needed to resolve a boundary layer) and/or large differences between the size of two neighboring elements. The use of ILU techniques alleviates this problem but is impractical for strongly coupled systems of equations. A much better remedy is to combine the mesh elements so as to 'hide' the detrimental anisotropies inside of the patches which are supposed to have approximately the same shape and size. Several adaptive blocking strategies for generation of such isotropic subdomains are described in [41],[47].

The global convergence behavior will be satisfactory because only the local subproblems are ill-conditioned. Moreover, the size of these local problems is usually very small. Thus, the complete inverse of the matrix fits into RAM and sometimes even into the cache so that the use of fast direct solvers is feasible. Consequently, the convergence rates should be independent of grid distortions and approach those for very regular structured meshes. If hardware–optimized routines such as the BLAS libraries are employed, then the solution of small subproblems can be performed very efficiently. Excellent convergence rates and a high overall performance can be achieved if the code is properly tuned and adapted to the processor architecture in each particular case.

6 Multilevel Solution Strategy

Pressure Schur complement schemes constitute viable solution techniques as such but they are particularly useful as smoothers for a *multilevel* algorithm, e.g., a geometric multigrid method. Let us start with explaining the typical implementation of such a solver for an abstract linear system of the form

$$A_N u_N = f_N \,. \tag{26}$$

It is assumed that there exists a hierarchy of levels $k = 1, \ldots, N$ which may be characterized, for instance, by the mesh size h_k. On each of these levels, one needs to assemble the matrix A_k and the right-hand side f_k for the discrete problem. We remark that only f_N is available a priori, while the sequence of residual vectors $\{f_k\}$ for $k < N$ is generated during the multigrid run.

The main ingredients of a (linear) multigrid algorithm are

- matrix–vector multiplication routines for the operators $A_k,\ k \leq N$
- an efficient *smoother* (basic iteration scheme) and a *coarse grid solver*
- *prolongation* I_{k-1}^{k} and *restriction* I_{k}^{k-1} operators for grid transfer

Each k-level iteration $MPSC(k, u_k^0, f_k)$ with initial guess u_k^0 represents a *multigrid cycle* which yields an (approximate) solution of the linear system $A_k u_k = f_k$. On the first level, the number of unknowns is typically so small that the auxiliary problem can be solved directly: $MPSC(1, u_1^0, f_1) = A_1^{-1} f_1$.

For all other levels ($k > 1$), the following algorithm is adopted [47]

Step 1. Presmoothing

Apply m smoothing steps (PSC iterations) to u_k^0 to obtain u_k^m.

Step 2. Coarse grid correction

Calculate f_{k-1} using the restriction operator I_k^{k-1} via

$$f_{k-1} = I_k^{k-1}(f_k - A_k u_k^m)$$

and let u_{k-1}^i $(1 \le i \le p)$ be defined recursively by

$$u_{k-1}^i = MPSC(k-1, u_{k-1}^{i-1}, f_{k-1}), \quad u_{k-1}^0 = 0\,.$$

Step 3. Relaxation and update

Calculate u_k^{m+1} using the prolongation operator I_{k-1}^k via

$$u_k^{m+1} = u_k^m + \alpha_k I_{k-1}^k u_{k-1}^p\,, \tag{27}$$

where the relaxation parameter α_k may be fixed or chosen adaptively so as to minimize the error $u_k^{m+1} - u_k$ in an appropriate norm, for instance, in the discrete energy norm

$$\alpha_k = \frac{(f_k - A_k u_k^m, I_{k-1}^k u_{k-1}^p)_k}{(A_k I_{k-1}^k u_{k-1}^p, I_{k-1}^k u_{k-1}^p)_k}\,.$$

Step 4. Postsmoothing

Apply n smoothing steps (PSC iterations) to u_k^{m+1} to obtain u_k^{m+n+1}.

After sufficiently many cycles on level N, the desired solution u_N of the generic problem (26) is recovered. In the framework of our multilevel pressure Schur complement schemes, there are (at least) two possible scenarios:

Global MPSC approach Solve the discrete problem (26) with

$$A_N := B^T A^{-1} B, \qquad u_N := p, \qquad f_N := \frac{1}{\Delta t} B^T A^{-1} \mathbf{g}.$$

The basic iteration is given by (19) and equivalent to a discrete projection cycle, whereby the velocity field $\mathbf{u}$ is updated in a parallel manner (see above). The bulk of CPU time is spent on matrix-vector multiplications with the Schur complement operator $S = B^T A^{-1} B$ which is needed for smoothing, defect calculation, and adaptive coarse grid correction. Unlike standard multigrid methods for scalar problems, global MPSC schemes involve solutions of

a viscous Burgers equation and a Poisson-like problem in each matrix-vector multiplication step (to avoid matrix inversion). In the case of highly nonstationary flows, the overhead cost is insignificant but it becomes appreciable as the Reynolds number decreases. Nevertheless, numerical tests indicate that the resulting multigrid solvers are optimal in the sense that the convergence rates are excellent and largely independent of mesh anisotropies.

Local MPSC approach Solve the discrete problem (26) with

$$A_N := \begin{bmatrix} A & \Delta t B \\ B^T & 0 \end{bmatrix}, \qquad u_N := \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix}, \qquad f_N := \begin{bmatrix} \mathbf{g} \\ 0 \end{bmatrix}.$$

The basic iteration is given by (22) which corresponds to the block-Gauss-Seidel/Jacobi method. The cost-intensive part is the smoothing step, as in the case of standard multigrid techniques for convection-diffusion equations and Poisson-like problems. Local MPSC schemes lead to very robust solvers for coupled problems. This multilevel solution strategy is to be recommended for incompressible flows at low and intermediate Reynolds numbers.

Further algorithmic details (adaptive coarse grid correction, grid transfer operators, nonlinear iteration techniques, time step control, implementation of boundary conditions) and a description of the high-performance software package FEATFLOW based on MPSC solvers can be found in [47],[48]. Some programming strategies, data structures, and guidelines for the development of a hardware-oriented parallel code are presented in [49],[50],[51].

7 Coupling with Scalar Equations

Both global and local MPSC schemes are readily applicable to the Navier-Stokes equations coupled with various turbulence models and/or scalar conservation laws for temperatures, concentrations, volume fractions, and other scalar variables. In many cases, the quantities of interest must remain strictly nonnegative for physical reasons, and the failure to enforce the positivity constraint for the *numerical* solution may have disastrous consequences. Therefore, a positivity-preserving discretization of convective terms is indispensable for such applications. This prerequisite is clearly satisfied by the algebraic FEM-FCT and FEM-TVD schemes introduced in the previous chapters.

As a representative example of a two-way coupling between (2) and a scalar transport equation, we consider the well-known Boussinesq approximation for natural convection problems. The nondimensional form of the governing equations for a buoyancy-driven incompressible flow reads [4]

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = \nu \Delta \mathbf{u} + T \mathbf{e}_g, \tag{28}$$

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = d \Delta T, \quad \nabla \cdot \mathbf{u} = 0, \tag{29}$$

where $\mathbf{u}$ is the velocity, p is the deviation from hydrostatic pressure and T is the temperature. The unit vector $\mathbf{e}_g$ is directed 'upward' (opposite to the gravitational force) and the nondimensional diffusion coefficients

$$\nu = \sqrt{\frac{Pr}{Ra}}, \qquad d = \sqrt{\frac{1}{Ra\,Pr}}$$

depend on the Rayleigh number Ra and the Prandtl number Pr. Details of this model and parameter settings for the *MIT benchmark problem* (natural convection in a differentially heated enclosure) can be found in [4].

7.1 Finite element discretization

After the discretization in space and time, we obtain a system of nonlinear algebraic equations which can be written in matrix form as follows

$$A_u(\mathbf{u}^{n+1})\mathbf{u}^{n+1} + \Delta t M_T T^{n+1} + \Delta t B p^{n+1} = \mathbf{f}_u, \tag{30}$$

$$A_T(\mathbf{u}^{n+1})T^{n+1} = f_T, \qquad B^T\mathbf{u}^{n+1} = 0. \tag{31}$$

Here and below the superscript $n+1$ refers to the time level, while subscripts identify the origin of discrete operators (u for the momentum equation and T for the heat conduction equation). Furthermore, the matrices A_u and A_T can be decomposed into a reactive, convective, and diffusive part

$$A_u(\mathbf{v}) = \alpha_u M_u + \beta_u K_u(\mathbf{v}) + \gamma_u L_u, \tag{32}$$

$$A_T(\mathbf{v}) = \alpha_T M_T + \beta_T K_T(\mathbf{v}) + \gamma_T L_T. \tag{33}$$

Note that we have the freedom of using different finite element approximations and discretization schemes for the velocity $\mathbf{u}$ and temperature T.

The discrete problem (30)–(31) admits the following representation

$$\begin{bmatrix} A_u(\mathbf{u}^{n+1}) & \Delta t M_T & \Delta t B \\ 0 & A_T(\mathbf{u}^{n+1}) & 0 \\ B^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ T^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_u \\ f_T \\ 0 \end{bmatrix} \tag{34}$$

and can be solved in the framework of a global or local MPSC method.

7.2 Global MPSC algorithm

Nonstationary flow configurations call for the use of operator splitting tools for the coupled system (34). This straightforward approach consists in solving the Navier-Stokes equations for $(\mathbf{u}, p)$ and the energy equation for T in a segregated manner. The decoupled subproblems are embedded into an outer iteration loop and solved sequentially by a global MPSC method (discrete projection) and an algebraic FCT/TVD scheme, respectively. For relatively small time steps, this strategy works very well, and simulation software can be developed in a modular way making use of optimized multigrid solvers. Moreover, it is possible to choose the time step individually for each subproblem.

In the simplest case (just one outer iteration per time step), the sequence of algorithmic steps to be performed is as follows [52]

1. Compute $\tilde{\mathbf{u}}$ from the momentum equation

$$A_u(\tilde{\mathbf{u}})\tilde{\mathbf{u}} = \mathbf{f}_u - \Delta t M_T T^n - \Delta t B p^n.$$

2. Solve the discrete Pressure-Poisson problem

$$B^T M_L^{-1} B q = \frac{1}{\Delta t} B^T \tilde{\mathbf{u}}.$$

3. Correct the pressure and the velocity

$$p^{n+1} = p^n + q, \quad \mathbf{u}^{n+1} = \tilde{\mathbf{u}} - \Delta t M_L^{-1} B q.$$

4. Solve the convection-diffusion equation for T

$$A_T(\mathbf{u}^{n+1})T^{n+1} = f_T.$$

Due to the nonlinearity of the discretized convective terms, iterative defect correction or a Newton-like method must be invoked in steps 1 and 4. This algorithm combined with the nonconforming FEM discretization appears to provide an 'optimal' flow solver for unsteady natural convection problems.

7.3 Local MPSC algorithm

Alternatively, a fully coupled solution of the problem at hand can be obtained following the local MPSC approach. To this end, a multigrid solver is applied to the suitably linearized **coupled** system (34). Each outer iteration for the nonlinearity corresponds to the following solution update [41],[52]

$$\begin{bmatrix} \mathbf{u}^{(l+1)} \\ T^{(l+1)} \\ p^{(l+1} \end{bmatrix} = \begin{bmatrix} \mathbf{u}^{(l)} \\ T^{(l)} \\ p^{(l)} \end{bmatrix} - \omega^{(l+1)} [F(\sigma, l)]^{-1} \begin{bmatrix} \delta\mathbf{u}^{(l)} \\ \delta T^{(l)} \\ \delta p^{(l)} \end{bmatrix}, \tag{35}$$

where the global defect vector is given by the relation

$$\begin{bmatrix} \delta\mathbf{u}^{(l)} \\ \delta T^{(l)} \\ \delta p^{(l)} \end{bmatrix} = \begin{bmatrix} A_u(\mathbf{u}^{(l)}) & \Delta t M_T & \Delta t B \\ 0 & A_T(\mathbf{u}^{(l)}) & 0 \\ B^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(l)} \\ T^{(l)} \\ p^{(l)} \end{bmatrix} - \begin{bmatrix} \mathbf{f}_u \\ f_T \\ 0 \end{bmatrix} \tag{36}$$

and the matrix to be inverted corresponds to the (approximate) Fréchet derivative of the underlying PDE system such that [41],[47]

$$F(\sigma, l) = \begin{bmatrix} A_u(\mathbf{u}^{(l)}) + \sigma R(\mathbf{u}^{(l)}) & \Delta t M_T & \Delta t B \\ \sigma R(T^{(l)}) & A_T(\mathbf{u}^{(l)}) & 0 \\ B^T & 0 & 0 \end{bmatrix}. \tag{37}$$

The nonlinearity of the governing equations gives rise to the 'reactive' contribution R which represents a solution-dependent mass matrix and may cause severe convergence problems. This is why it is multiplied by the adjustable parameter σ. The Newton method is recovered for $\sigma = 1$, while the value $\sigma = 0$ yields the fixed-point defect correction scheme. In either case, the linearized problem is solved by a fully coupled multigrid solver equipped with a local MPSC smoother of 'Vanka' type [41]. As before, the matrix $F(\sigma, l)$ is decomposed into small blocks C_i associated with individual patches Ω_i. The smoothing of the global residual vector is performed patchwise by solving the corresponding local subproblems.

The size of the matrices to be inverted can be further reduced by resorting to the Schur complement approach. For simplicity, consider the case $\sigma = 0$ (extension to $\sigma > 0$ is straightforward). It follows from (30)–(31) that

$$T^{n+1} = A_T^{-1} f_T, \quad \mathbf{u}^{n+1} = A_u^{-1}[\mathbf{f}_u - \Delta t M_T T^{n+1} - \Delta t B p^{n+1}] \tag{38}$$

and the discretized continuity equation can be cast into the form

$$B^T \mathbf{u}^{n+1} = B^T A_u^{-1}[\mathbf{f}_u - \Delta t M_T A_T^{-1} f_T - \Delta t B p^{n+1}] = 0 \tag{39}$$

which corresponds to the pressure Schur complement equation

$$B^T A_u^{-1} B p^{n+1} = B^T A_u^{-1} \left[\frac{1}{\Delta t} \mathbf{f}_u - M_T A_T^{-1} f_T \right]. \tag{40}$$

Thus, highly efficient local preconditioners of the form (21) can be employed instead of C_i. The converged solution p^{n+1} to the scalar subproblem (40) is plugged into (38) to obtain the velocity $\mathbf{u}^{n+1}$ and the temperature T^{n+1}.

The advantages of the seemingly complicated local MPSC strategy are as follows. First of all, steady-state solutions can be obtained without resorting to pseudo-time-stepping. Moreover, the fully coupled treatment of dynamic flows makes it possible to use large time steps without any loss of robustness. On the other hand, the convergence behavior of multigrid solvers for the Newton linearization may turn out to be unsatisfactory and the computational cost per outer iteration is rather high as compared to the global MPSC algorithm. The performance of both solution techniques as applied to the MIT benchmark problem is illustrated by the numerical results reported in [52].

8 Implementation of the $k - \varepsilon$ Model

High-resolution schemes like FCT and TVD play an increasingly important role in simulation of turbulent flows. Flow structures that cannot be resolved on the computational mesh activate the flux limiter which curtails the raw antidiffusion so as to filter out the small-scale fluctuations. Interestingly enough, the residual artificial viscosity provides an excellent subgrid scale model for *monotonically integrated* Large Eddy Simulation (MILES), see [2],[16].

In spite of recent advances in the field of LES and DNS (direct numerical simulation), simpler turbulence models based on Reynolds averaging (RANS) still prevail in CFD software for simulation of industrial processes. In particular, the evolution of the turbulent kinetic energy k and of its dissipation rate ε is governed by two convection-dominated transport equations

$$\frac{\partial k}{\partial t} + \nabla \cdot \left(k\mathbf{u} - \frac{\nu_T}{\sigma_k} \nabla k \right) = P_k - \varepsilon, \tag{41}$$

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot \left(\varepsilon\mathbf{u} - \frac{\nu_T}{\sigma_\varepsilon} \nabla \varepsilon \right) = \frac{\varepsilon}{k}(C_1 P_k - C_2 \varepsilon), \tag{42}$$

where $\mathbf{u}$ denotes the averaged velocity, $\nu_T = C_\mu k^2/\varepsilon$ is the turbulent eddy viscosity and $P_k = \frac{\nu_T}{2}|\nabla\mathbf{u} + \nabla\mathbf{u}^T|^2$ is the production term. For the standard $k-\varepsilon$ model, the default values of the involved parameters are as follows

$$C_\mu = 0.09, \quad C_1 = 1.44, \quad C_2 = 1.92, \quad \sigma_k = 1.0, \quad \sigma_\varepsilon = 1.3.$$

The velocity field $\mathbf{u}$ is obtained from the incompressible Navier-Stokes equations with $\nabla \cdot (\nu + \nu_T)[\nabla\mathbf{u} + (\nabla\mathbf{u})^T]$ instead of $\nu\Delta\mathbf{u}$.

We remark that the transport equations for k and ε are strongly coupled and nonlinear so that their numerical solution is a very challenging task. Moreover, the discretization scheme **must** be positivity-preserving because negative values of the eddy viscosity are totally unacceptable. Unfortunately, implementation details and employed 'tricks' are rarely reported in the literature, so that a novice to this area of CFD research often needs to reinvent the wheel. Therefore, we deem it appropriate to discuss the implementation of a FEM-TVD algorithm for the $k-\varepsilon$ model in some detail.

8.1 Positivity-preserving linearization

The block-iterative algorithm proposed in [27],[28] consists of nested loops so that the coupled PDE system is replaced by a sequence of linear subproblems. The solution-dependent coefficients are 'frozen' during each outer iteration and updated as new values become available. The quasi-linear transport equations can be solved by an implicit FEM-FCT or FEM-TVD scheme but the linearization procedure must be tailored to the need to preserve the positivity of k and ε in a numerical simulation. Due to the presence of sink terms in the right-hand side of both equations, the positivity constraint may be violated even if a high-resolution scheme is employed for the discretization of convective terms. It can be proved that the exact solution to the $k-\varepsilon$ model remains nonnegative for positive initial data [32],[33] and it is essential to guarantee that the numerical scheme will also possess this property.

Let us consider the following representation of the equations at hand [29]

$$\frac{\partial k}{\partial t} + \nabla \cdot (k\mathbf{u} - d_k \nabla k) + \gamma k = P_k, \tag{43}$$

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot (\varepsilon\mathbf{u} - d_\varepsilon \nabla \varepsilon) + C_2 \gamma \varepsilon = C_1 P_k, \tag{44}$$

where the parameter $\gamma = \frac{\varepsilon}{k}$ is proportional to the specific dissipation rate ($\gamma = C_\mu \omega$). The turbulent dispersion coefficients are given by $d_k = \frac{\nu_T}{\sigma_k}$ and $d_\varepsilon = \frac{\nu_T}{\sigma_\varepsilon}$. By definition, the source terms in the right-hand side are nonnegative. Furthermore, the parameters ν_T and γ must also be nonnegative for the solution of the convection-reaction-diffusion equations to be well-behaved [5]. In our numerical algorithm, their values are taken from the previous iteration and their positivity is secured as explained below. This linearization technique was proposed by Lew *et al.* [29] who noticed that the positivity of the lagged coefficients is even more important than that of the transported quantities and can be readily enforced without violating the discrete conservation principle.

Applying an implicit FCT/TVD scheme to the above equations, we obtain two nonlinear algebraic systems which can be written in the generic form

$$A(u^{(l+1)})u^{(l+1)} = B(u^{(l)})u^{(l)} + q^{(k)}, \qquad l = 0, 1, 2, \ldots \tag{45}$$

Here k is the index of the outermost loop in which the velocity $\mathbf{u}$ and the source term P_k are updated. The index l refers to the outer iteration for the $k-\varepsilon$ model, while the index m is reserved for inner flux/defect correction loops. The structure of the matrices A and B is as follows:

$$A(u) = M_L - \theta \Delta t(K^*(u) + T), \tag{46}$$

$$B(u) = M_L + (1-\theta)\Delta t(K^*(u) + T), \tag{47}$$

where $K^*(u)$ is the LED transport operator incorporating nonlinear antidiffusion and T denotes the standard reaction-diffusion operator which is a symmetric positive-definite matrix with nonnegative off-diagonal entries. It is obvious that the discretized production terms $q^{(k)}$ are also nonnegative. Thus, the positivity of $u^{(l)}$ is inherited by the new iterate $u^{(l+1)} = A^{-1}(Bu^{(l)} + q^{(k)})$ provided that $\theta = 1$ (backward Euler) or the time step is sufficiently small.

8.2 Positivity of coefficients

The predicted values $k^{(l+1)}$ and $\varepsilon^{(l+1)}$ are used to recompute the parameter $\gamma^{(l+1)}$ for the next outer iteration (if any). The turbulent eddy viscosity $\nu_T^{(k)}$ is updated in the outermost loop. In the turbulent flow regime $\nu_T \gg \nu$ and the laminar viscosity ν can be neglected. Hence, we set $\nu_{\text{eff}} = \nu_T$, where the eddy viscosity ν_T is bounded from below by ν and from above by the maximum admissible mixing length $l_{\max}$ (e.g. the width of the computational domain). Specifically, we define the limited mixing length l_* as

$$l_* = \begin{cases} \frac{\alpha}{\varepsilon} & \text{if} \quad \varepsilon > \frac{\alpha}{l_{\max}} \\ l_{\max} & \text{otherwise} \end{cases}, \qquad \text{where} \quad \alpha = C_\mu k^{3/2} \tag{48}$$

and use it to update the turbulent eddy viscosity ν_T in the outermost loop:

$$\nu_T = \max\{\nu, l_* \sqrt{k}\} \tag{49}$$

as well as the parameter γ in each outer iteration for the $k-\varepsilon$ model:

$$\gamma = C_\mu \frac{k}{\nu_*}, \qquad \text{where} \quad \nu_* = \max\{\nu, l_* \sqrt{k}\}. \tag{50}$$

In the case of a FEM-TVD method, the positivity proof is only valid for the converged solution to (45) while intermediate solution values may be negative. Since it is impractical to perform many defect correction steps in each outer iteration, it is worthwhile to substitute $k_* = \max\{0, k\}$ for k in formulae (48)–(50) so as to to prevent taking the square root of a negative number. Upon convergence, this safeguard will not make any difference, since k will be nonnegative from the outset. The above representation of ν_T and γ makes it possible to preclude division by zero and obtain bounded coefficients without making any *ad hoc* assumptions and affecting the actual values of k and ε.

8.3 Initial conditions

Another important issue which is seldom addressed in the CFD literature is the initialization of data for the $k-\varepsilon$ model. As a rule, it is rather difficult to devise a reasonable initial guess for a steady-state simulation or proper initial conditions for a dynamic one. The laminar Navier-Stokes equations (2) remain valid until the flow gains enough momentum for the turbulent effects to become pronounced. Therefore, the $k-\varepsilon$ model should be activated at a certain time $t_* > 0$ after the startup.

During the 'laminar' initial phase ($t \le t_*$), a constant effective viscosity ν_0 is prescribed. The values to be assigned to k and ε at $t = t_*$ are uniquely defined by the choice of ν_0 and of the default mixing length $l_0 \in [l_{\min}, l_{\max}]$ where $l_{\min}$ corresponds to the size of the smallest admissible eddies:

$$k_0 = \left(\frac{\nu_0}{l_0}\right)^2, \qquad \varepsilon_0 = C_\mu \frac{k_0^{3/2}}{l_0} \qquad \text{at} \quad t \le t_*. \tag{51}$$

This strategy was adopted because the effective viscosity ν_0 and the mixing length l_0 are somewhat easier to estimate (at least for a CFD practitioner) than k_0 and ε_0. In any case, long-term simulation results are typically not very sensitive to the choice of initial data.

8.4 Boundary conditions

At the inlet Γ_{in}, all velocity components and the values of k and ε are given:

$$\mathbf{u} = \mathbf{g}, \qquad k = c_{bc}|\mathbf{u}|^2, \qquad \varepsilon = C_\mu \frac{k^{3/2}}{l_0} \qquad \text{on} \quad \Gamma_{\text{in}}, \tag{52}$$

where $c_{bc} \in [0.001, 0.01]$ is an empirical constant [5] and $|\mathbf{u}| = \sqrt{\mathbf{u} \cdot \mathbf{u}}$ is the Euclidean norm of the velocity. At the outlet Γ_{out}, the normal gradients of

all scalar variables are required to vanish, and the 'do-nothing' [47] boundary conditions are prescribed:

$$\mathbf{n}\cdot\mathcal{S}(\mathbf{u})=\mathbf{0},\qquad \mathbf{n}\cdot\nabla k=0,\qquad \mathbf{n}\cdot\nabla\varepsilon=0\qquad \text{on}\quad \Gamma_{\text{out}}. \tag{53}$$

Here $\mathcal{S}(\mathbf{u})=-\left(p+\frac{2}{3}k\right)\mathcal{I}+(\nu+\nu_T)[\nabla\mathbf{u}+(\nabla\mathbf{u})^T]$ denotes the effective stress tensor. The numerical treatment of inflow and outflow boundary conditions does not present any difficulty. In the finite element framework, relations (53) imply that the surface integrals resulting from integration by parts vanish and do not need to be assembled.

At an impervious solid wall Γ_w, the normal component of the velocity must vanish, whereas tangential slip is permitted in turbulent flow simulations. The implementation of the no-penetration (free slip) boundary condition

$$\mathbf{n}\cdot\mathbf{u}=0\qquad \text{on}\quad \Gamma_w \tag{54}$$

is nontrivial if the boundary of the computational domain is not aligned with the axes of the Cartesian coordinate system. In this case, condition (54) is imposed on a linear combination of several velocity components whereas their boundary values are unknown. Therefore, standard implementation techniques for Dirichlet boundary conditions based on a modification of the corresponding matrix rows [47] cannot be used.

In order to set the normal velocity component equal to zero, we nullify the off-diagonal entries of the preconditioner $A(\mathbf{u}^{(m)})=\{a_{ij}^{(m)}\}$ in the defect correction loop. This enables us to compute the boundary values of $\mathbf{u}$ explicitly before solving a sequence of linear systems for the velocity components:

$$a_{ij}^{(m)}:=0,\quad \forall j\neq i,\qquad \mathbf{u}_i^{*}:=\mathbf{u}_i^{(m)}+\mathbf{r}_i^{(m)}/a_{ii}^{(m)}\qquad \text{for}\quad \mathbf{x}_i\in\Gamma_w. \tag{55}$$

Next, we project the predicted values $\mathbf{u}_i^{*}$ onto the tangent vector/plane and constrain the corresponding entry of the defect vector $\mathbf{r}_i^{(m)}$ to be zero

$$\mathbf{u}_i^{(m)}:=\mathbf{u}_i^{*}-(\mathbf{n}_i\cdot\mathbf{u}_i^{*})\mathbf{n}_i,\qquad \mathbf{r}_i^{(m)}:=0\qquad \text{for}\quad \mathbf{x}_i\in\Gamma_w. \tag{56}$$

After this manipulation, the corrected values $\mathbf{u}_i^{(m)}$ act as Dirichlet boundary conditions for the solution $\mathbf{u}_i^{(m+1)}$ at the end of the defect correction step.

As an alternative to the implementation technique of predictor-corrector type, the projection can be applied to the residual vector rather than to the nodal values of the velocity:

$$a_{ij}^{(m)}:=0,\quad \forall j\neq i,\qquad \mathbf{r}_i^{(m)}:=\mathbf{r}_i^{(m)}-(\mathbf{n}_i\cdot\mathbf{r}_i^{(m)})\mathbf{n}_i\qquad \text{for}\quad \mathbf{x}_i\in\Gamma_w. \tag{57}$$

For Cartesian geometries, the algebraic manipulations to be performed affect just the normal velocity component. Note that virtually no extra programming effort is required, which is a significant advantage as compared to another feasible implementation based on local coordinate transformations during the element-by-element matrix assembly [9].

8.5 Wall functions

To complete the problem statement, we still need to prescribe the tangential stress as well as the boundary values of k and ε on Γ_w. Note that the equations of the $k-\varepsilon$ model are invalid in the vicinity of the wall where the Reynolds number is rather low and viscous effects are dominant. In order to avoid the need for resolution of strong velocity gradients, *wall functions* can be derived using the boundary layer theory and applied at an internal boundary Γ_δ located at a distance δ from the solid wall Γ_w [31],[32],[33].

In essence, a boundary layer of width δ is removed from the actual computational domain Ω and the equations are solved in the reduced domain Ω_δ subject to the following empirical boundary conditions:

$$\mathbf{n}\cdot\mathcal{D}(\mathbf{u})\cdot\mathbf{t}=-u_\tau^2\frac{\mathbf{u}}{|\mathbf{u}|},\qquad k=\frac{u_\tau^2}{\sqrt{C_\mu}},\qquad \varepsilon=\frac{u_\tau^3}{\kappa\delta}\qquad \text{on}\quad \Gamma_\delta. \tag{58}$$

Here $\mathcal{D}(\mathbf{u})=(\nu+\nu_T)[\nabla\mathbf{u}+(\nabla\mathbf{u})^T]$ is the viscous part of the stress tensor, the unit vector $\mathbf{t}$ refers to the tangential direction, $\kappa=0.41$ is the von Kármán constant and u_τ is the *friction velocity* which is assumed to satisfy

$$g(u_\tau)=|\mathbf{u}|-u_\tau\left(\frac{1}{\kappa}\log y^+ + 5.5\right)=0 \tag{59}$$

in the *logarithmic layer*, where the local Reynolds number $y^+=\frac{u_\tau\delta}{\nu}$ is in the range $20\le y^+\le 100$, and be a linear function of y^+ in the *viscous sublayer*, where $y^+<20$. Note that $\mathbf{u}$ represents the tangential velocity as long as the no-penetration condition (54) is imposed on Γ_δ.

Equation (59) can be solved iteratively, e.g., by Newton's method [31]:

$$u_\tau^{l+1}=u_\tau^l-\frac{g(u_\tau^l)}{g'(u_\tau^l)}=u_\tau^l+\frac{|\mathbf{u}|-u_\tau f(u_\tau^l)}{1/\kappa+f(u_\tau^l)},\qquad l=0,1,2,\ldots \tag{60}$$

where the auxiliary function f is given by

$$f(u_\tau)=\frac{1}{\kappa}\log y_*^+ + 5.5,\qquad y_*^+=\max\left\{20,\frac{u_\tau\delta}{\nu}\right\}.$$

The friction velocity is initialized by the approximation

$$u_\tau^0=\sqrt{\frac{\nu|\mathbf{u}|}{\delta}}$$

and no iterations are performed if it turns out that $y^+=\frac{u_\tau^0\delta}{\nu}<20$. In other words, $u_\tau=u_\tau^0$ in the viscous sublayer. Moreover, we use $y_*^+=\max\{20,y^+\}$ in the Newton iteration to guarantee that the approximate solution belongs to the logarithmic layer and remains bounded for $y^+\to 0$.

The friction velocity u_τ is plugged into (58) to compute the tangential stress, which yields a natural boundary condition for the velocity. Integration by parts in the weak form of the Navier-Stokes equations gives rise to a surface integral over the internal boundary Γ_δ which contains the prescribed traction:

$$\int_{\Gamma_\delta} [\mathbf{n} \cdot \mathcal{D}(\mathbf{u}) \cdot \mathbf{t}] \cdot \mathbf{v} \, ds = - \int_{\Gamma_\delta} u_\tau^2 \frac{\mathbf{u}}{|\mathbf{u}|} \cdot \mathbf{v} \, ds. \tag{61}$$

The free slip condition (54) overrides the normal stress, and Dirichlet boundary conditions for k and ε are imposed in the strong sense. For further details regarding the implementation of wall laws we refer to [31],[32],[33].

8.6 Underrelaxation for outer iterations

Due to the intricate coupling of the governing equations, it is sometimes worthwhile to use a suitable underrelaxation technique in order to prevent the growth of numerical instabilities and secure the convergence of outer iterations. This task can be accomplished by limiting the computed solution increments before applying them to the last iterate:

$$u^{(m+1)} := u^{(m)} + \omega^{(m)}(u^{(m+1)} - u^{(m)}), \qquad \text{where} \quad 0 \le \omega^{(m)} \le 1. \tag{62}$$

The damping factor $\omega^{(m)}$ may be chosen adaptively so as to accelerate convergence and minimize the error in a certain norm [47]. However, fixed values (for example, $\omega = 0.8$) usually suffice for practical purposes. The sort of underrelaxation can be used in all loops (indexed by k, l and m) and applied to selected dependent variables like k, ε or ν_T.

In addition, an *implicit underrelaxation* can be performed in m-loops by increasing the diagonal dominance of the preconditioner [10],[34]

$$a_{ii}^{(m)} := a_{ii}^{(m)} / \alpha^{(m)}, \qquad \text{where} \quad 0 \le \alpha^{(m)} \le 1. \tag{63}$$

Of course, the scaling of the diagonal entries does not affect the converged solution. This strategy proves more robust than an *explicit underrelaxation* of the form (62). On the other hand, no underrelaxation whatsoever is needed for moderate time steps which are typically used in dynamic simulations.

9 Adaptive Time Step Control

A remark is in order regarding the time step selection for implicit schemes. Unlike their explicit counterparts, they are unconditionally stable so that the time step is limited only by accuracy considerations (for nonstationary problems). Thus, it should be chosen adaptively so as to obtain a sufficiently good approximation at the least possible cost. Many adaptive time stepping techniques have been proposed in the literature. Most of them were originally developed in the ODE context and are based on an estimate of the local truncation error which provides a usable indicator for the step size control.

The 'optimal' value of Δt should guarantee that the deviation of a user-defined functional J (pointwise solution values or certain integral quantities like lift and drag) from its exact value does not exceed a given tolerance

$$|J(u) - J(u_{\Delta t})| \approx TOL. \tag{64}$$

Assuming that the error at the time level t^n is equal to zero, a heuristic error indicator can be derived from asymptotic expansions for the numerical values of J computed using two different time steps. For instance, consider

$$\begin{aligned} J(u_{\Delta t}) &= J(u) + \Delta t^2 e(u) + \mathcal{O}(\Delta t)^4, \\ J(u_{m\Delta t}) &= J(u) + m^2 \Delta t^2 e(u) + \mathcal{O}(\Delta t)^4, \end{aligned}$$

where $m > 1$ is an integer number ($m = 2, 3$). The error term $e(v)$ is supposed to be independent of the time step and can be estimated as follows

$$e(u) \approx \frac{J(u_{m\Delta t}) - J(u_{\Delta t})}{(m^2 - 1)\Delta t^2}.$$

For the relative error to approach the prescribed tolerance TOL as required by (64), the new time step Δt_* should be chosen so that

$$|J(u) - J(u_{\Delta t_*})| \approx \left(\frac{\Delta t_*}{\Delta t}\right)^2 \frac{|J(u_{\Delta t}) - J(u_{m\Delta t})|}{m^2 - 1} = TOL.$$

The required adjustment of the time step is given by the formula

$$\Delta t_*^2 = TOL \frac{(m^2 - 1)\Delta t^2}{|J(u_{\Delta t}) - J(u_{m\Delta t})|}.$$

Furthermore, the solution accuracy can be enhanced by resorting to Richardson's extrapolation (see any textbook on numerical methods for ODEs).

The above considerations may lack some mathematical rigor but nevertheless lead to a very good algorithm for automatic time step control [47]

1. Make one large time step of size $m\Delta t$ to compute $u_{m\Delta t}$.
2. Make m small substeps of size Δt to compute $u_{\Delta t}$.
3. Evaluate the relative changes, i. e., $|J(u_{\Delta t}) - J(u_{m\Delta t})|$.
4. Calculate the 'optimal' value Δt_* for the next time step.
5. If $\Delta t_* \ll \Delta t$, reject the solution and go back to step 1.
6. Assign $u := u_{\Delta t}$ or perform Richardson's extrapolation.

Note that the computational cost per time step increases significantly since the solution $u_{m\Delta t}$ may be as expensive to obtain as $u_{\Delta t}$ (due to slow convergence). On the other hand, adaptive time stepping contributes to the robustness of the code and improves its overall efficiency as well as the credibility of simulation results. Further algorithmic details for this approach can be found in [47].

Another simple strategy for adaptive time step control was introduced by Valli et al. [53],[54]. Their *PID controller* is based on the relative changes of a suitable indicator variable (temperature distribution, concentration fields, kinetic energy, eddy viscosity etc.) and can be summarized as follows

1. Compute the relative changes of the chosen indicator variable u

$$e_n = \frac{||u^{n+1} - u^n||}{||u^{n+1}||}.$$

2. If they are too large ($e_n > \delta$), reject u^{n+1} and recompute it using

$$\Delta t_* = \frac{\delta}{e_n} \Delta t_n.$$

3. Adjust the time step **smoothly** so as to approach the prescribed tolerance TOL for the relative changes

$$\Delta t_{n+1} = \left(\frac{e_{n-1}}{e_n}\right)^{k_P} \left(\frac{TOL}{e_n}\right)^{k_I} \left(\frac{e_{n-1}^2}{e_n e_{n-2}}\right)^{k_D} \Delta t_n.$$

4. Limit the growth and reduction of the time step so that

$$\Delta t_{\min} \leq \Delta t_{n+1} \leq \Delta t_{\max}, \qquad m \leq \frac{\Delta t_{n+1}}{\Delta t_n} \leq M.$$

The default values of the PID parameters as proposed by Valli et al. [54] are $k_P = 0.075, k_I = 0.175$ and $k_D = 0.01$. Unlike in the case of adaptive time-stepping techniques based on the local truncation error, there is no need for computing an extra solution with a different time step. Therefore, the cost of the feedback mechanism is negligible. Our own numerical studies [28] confirm that this heuristic control strategy is very robust and efficient.

10 Numerical Examples

Flow around a cylinder. The first incompressible flow problem to be dealt with is the well-known benchmark *Flow around a cylinder* developed in 1995 for the priority research program "Flow simulation on high-performance computers" under the auspices of DFG, the German Research Association [40]. This project was intended to facilitate the evaluation of various numerical algorithms for the incompressible Navier-Stokes equations in the **laminar** flow regime. A quantitative comparison of simulation results is possible on the basis of relevant flow characteristics such as drag and lift coefficients, for which sufficiently accurate reference values are available. Moreover, the efficiency of different solution techniques can be assessed in an objective manner.

Consider the steady incompressible flow around a cylinder with circular cross-section. An in-depth description of the geometrical details and boundary conditions for the 2D/3D case can be found in references [40],[47] which contain all relevant information regarding this benchmark configuration. The flow at $Re = 20$ is actually dominated by diffusion and could be simulated by the standard Galerkin method without any extra stabilization (as far as the discretization is concerned; the iterative solver may require using a stabilized preconditioner). Ironically, it was this 'trivial' steady-state problem that has led us to devise the multidimensional flux limiter of TVD type [24]. Both FCT schemes and slope limiter methods based on stencil reconstruction failed to converge, so the need for a different limiting strategy was apparent.

Furthermore, it is instructive to study the interplay of finite element discretizations for the convective and diffusive terms. As a matter of fact, discrete upwinding can be performed for the cumulative transport operator or just for the convective part. In the case of the nonconforming $\tilde{Q}_1$-elements, the discrete Laplacian operator originating from the Galerkin approximation of viscous terms is a positive-definite matrix but some of its off-diagonal coefficients are negative. Our numerical experiments indicate that it is worthwhile to leave it unchanged and start with a FEM-TVD discretization of the convective term. Physical diffusion can probably be taken into account **after** algebraic flux correction but the sums of upstream and downstream edge contributions $Q^{\pm}$ and $P^{\pm}$ for the node-oriented TVD limiter should be evaluated using the coefficients of the (antisymmetric) convective operator.

To generate hierarchical data structures for the MPSC algorithms implemented in the software package FEATFLOW [48], we introduce a sequence of successively refined quadrilateral meshes. The elements of the coarse mesh shown in Fig. 3 are subdivided into four subelements at each refinement level, and the 2D mesh is extended into the third dimension for a 3D simulation. The two-dimensional results produced by a global MPSC (discrete projection) method with a FEM-TVD discretization of the convective terms are presented in Table 1. The computational mesh for multigrid level NLEV contains NMT midpoints and NEL elements. For the employed $\tilde{Q}_1/Q_0$ finite element pair, NMT represents the number of unknowns for each velocity component, while NEL is the number of degrees of freedom for the pressure. It can be seen that the drag and lift coefficients approach the reference values $C_D \approx 5.5795$ and $C_L \approx 0.01061$ as the mesh is refined. The same outcome can be obtained in the local MPSC framework without resorting to pseudo-time-stepping.

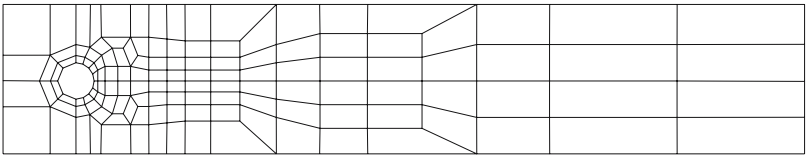


Fig. 3. Coarse mesh (2D) for the DFG benchmark 'Flow around a cylinder'

Originally, stabilization of convective terms in the FEATFLOW package was performed using streamline diffusion or Samarski's upwind scheme, whereby the amount of artificial viscosity depends on the local Reynolds number and on the value of the user-defined parameter UPSAM as explained in [47],[48]. Table 2 illustrates that drag and lift for UPSAM=0.1 and UPSAM=1.0 differ appreciably, especially on coarse meshes. In the former case, both quantities tend to be underestimated, while the latter parameter setting results in an unstable convergence behavior. Note that C_L is negative (!) for NLEV=3.

Since the optimal value of the free parameter is highly problem-dependent, it is impossible to find from *a priori* considerations. In addition, Samarski's hybrid method is only suitable for intermediate and low Reynolds numbers, as it becomes increasingly diffusive and degenerates into the first-order upwind scheme in the limit of inviscid flow. By contrast, the accuracy of our FEM-TVD discretization does not degrade as $Re \to \infty$. However, it does depend on the choice of the flux limiter (MC was employed in the above example) so the method is – arguably – not completely "parameter-free". Moreover, the results are influenced by the type of $\tilde{Q}_1$ basis functions (parametric or non-parametric, with midpoint- or mean-value based degrees of freedom) as well as by the approximations involved in the evaluation of C_D and C_L.

Table 1. Global MPSC method / TVD (MC limiter)

NLEV	NMT	NEL	C_D	C_L
3	4264	2080	5.5504	$0.8708 \cdot 10^{-2}$
4	16848	8320	5.5346	$0.9939 \cdot 10^{-2}$
5	66976	33280	5.5484	$0.1043 \cdot 10^{-1}$
6	267072	133120	5.5616	$0.1056 \cdot 10^{-1}$
7	1066624	532480	5.5707	$0.1054 \cdot 10^{-1}$
8	4263168	2129920	5.5793	$0.1063 \cdot 10^{-1}$

Table 2. Global MPSC method / Samarski's upwind

	UPSAM=0.1		UPSAM=1.0	
NLEV	C_D	C_L	C_D	C_L
3	5.4860	$0.5302 \cdot 10^{-2}$	5.9222	$-0.3475 \cdot 10^{-2}$
4	5.5076	$0.9548 \cdot 10^{-2}$	5.6525	$0.6584 \cdot 10^{-2}$
5	5.5386	$0.1025 \cdot 10^{-1}$	5.5736	$0.1007 \cdot 10^{-1}$
6	5.5581	$0.1044 \cdot 10^{-1}$	5.5658	$0.1048 \cdot 10^{-1}$
7	5.5692	$0.1047 \cdot 10^{-1}$	5.5718	$0.1042 \cdot 10^{-1}$

In Table 3, we present the drag and lift coefficients for a three-dimensional simulation of the flow around the cylinder. The hexahedral mesh for NLEV=4 consists of 49,152 elements, which corresponds to 151,808 unknowns for each velocity component. In order to evaluate the performance of the global MPSC

solver and verify grid convergence, we compare the results to those obtained on a coarser and a finer mesh. All numerical solutions were marched to the steady state by the fully implicit backward Euler method. The discretization of convective terms was performed using (i) finite volume upwinding (UPW), (ii) Samarski's hybrid scheme (SAM), (iii) streamline diffusion stabilization (SD), and (iv) algebraic flux correction (TVD). This numerical study confirms that standard artificial viscosity methods are rather sensitive to the values of the empirical constants, whereas FEM-TVD performs remarkably well. The reference values $C_D \approx 6.1853$ and $C_L \approx 0.95 \cdot 10^{-2}$ for this 3D configuration were calculated in [20] by an isoparametric high-order FEM.

Table 3. Global MPSC method: 3D simulation, $C_D/(C_L \cdot 100)$

NLEV	UPW-1st	SAM-1.0	SD-0.25	SD-0.5	TVD/MC
3	6.08/ 1.01	5.72/ 0.28	5.78/-0.44	5.98/-0.52	5.80/ 0.36
4	6.32/ 1.20	6.07/ 0.62	6.13/ 0.26	6.26/ 0.18	6.14/ 0.46
5	6.30/ 1.20	6.14/ 0.83	6.17/ 0.70	6.23/ 0.64	6.18/ 0.80

Backward facing step. Let us proceed to a three-dimensional test problem which deals with a turbulent flow over a backward facing step at $Re = 44,000$, see [31] for details. Our objective is to validate the implementation of the $k - \varepsilon$ model as described above. As before, the incompressible Navier-Stokes equations are discretized in space using the BB-stable nonconforming $\tilde{Q}_1/Q_0$ finite element pair, while conforming Q_1 (trilinear) elements are employed for the turbulent kinetic energy and its dissipation rate. All convective terms are handled by the fully implicit FEM-TVD method. The velocity-pressure coupling is enforced in the framework of a global MPSC formulation.

Standard wall laws are applied on the boundary except for the inlet and outlet. The stationary distribution of k and ε in the middle cross-section ($z = 0.5$) is displayed in Fig. 4. The variation of the friction coefficient

$$c_f = \frac{2\tau_w}{\rho_\infty u_\infty^2} = \frac{2u_\tau^2}{u_\infty^2} = \frac{2k}{u_\infty^2}\sqrt{C_\mu}$$

along the bottom wall is presented in Fig. 5 (left). The main recirculation length $L \approx 6.8$ is in a good agreement with the numerical results reported in the literature [31]. Moreover, the horizontal velocity component (see Fig. 5, right) assumes positive values at the bottom of the step, which means that the weak secondary vortex is captured as well. The parameter settings for this three-dimensional simulation were as follows

$$\delta = 0.05, \quad c_{bc} = 0.0025, \quad \nu_0 = 10^{-3}, \quad l_0 = 0.02, \quad l_{\max} = 1.0.$$

The computational mesh shown in Fig. 6 contains 57,344 hexahedral cells (pressure unknowns), which corresponds to 178,560 faces (degrees of freedom for each velocity component) and 64,073 vertices (nodes for k and ε).

Distribution of k in the cutplane $z = 0.5$

Distribution of ε in the cutplane $z = 0.5$

Fig. 4. Backward facing step: stationary FEM-TVD solution, $Re = 44,000$

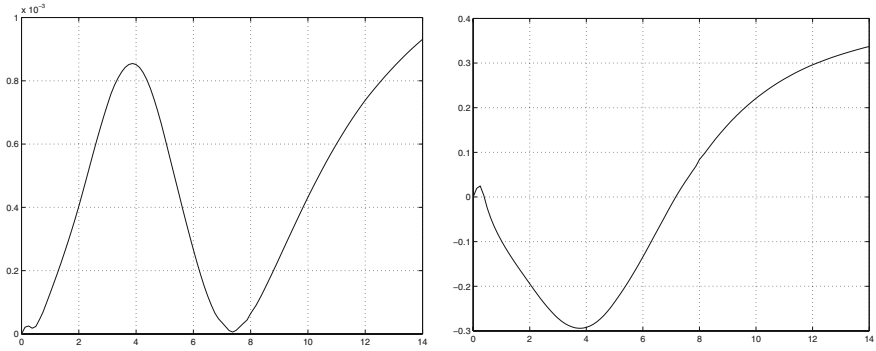


Fig. 5. Distribution of c_f (left) and u_x (right) along the bottom wall

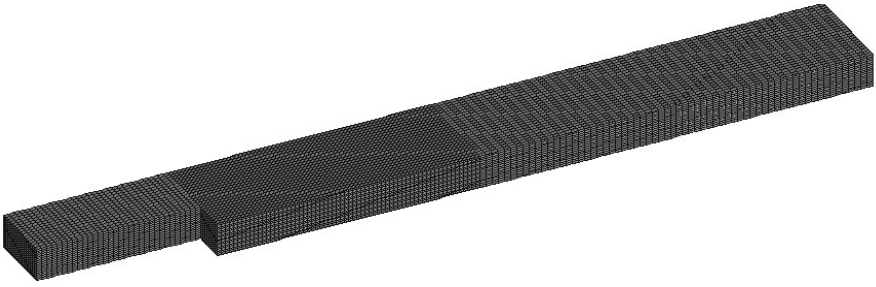


Fig. 6. Hexahedral computational mesh for the 3D simulation

11 Application to More Complex Flow Models

In this section, we discuss several incompressible flow models which call for the use of algebraic flux correction (FCT or TVD techniques). Specifically, let us consider generalized Navier-Stokes problems of the form

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}\right)=\mathbf{f}+\mu\Delta\mathbf{u}-\nabla p \quad , \quad \nabla\cdot\mathbf{u}=0 \tag{65}$$

complemented by additional PDEs which describe physical processes like

1. Heat transfer in complex geometries
   → *Ceramic plate heat exchanger*
2. Multiphase flow with chemical reaction
   → *Gas-liquid reactors*
3. Nonlinear fluids/granular flow
   → *Sand motion in silos*
4. Free and moving boundaries
   → *Level set FEM methods*

Although these typical flow configurations (to be presented below) differ in their complexity and cover a wide range of Reynolds numbers, all of them require an accurate treatment of convective transport. Numerical artifacts such as small-scale oscillations/ripples may cause an abnormal termination of the simulation run due to division by zero, floating point overflow etc. However, in the worst case they can be misinterpreted as physical phenomena and eventually result in making wrong decisions for the design of industrial equipment. Therefore, nonphysical solution behavior should be avoided at any cost, and the use of FEM-FCT/TVD or similar high-resolution schemes is recommendable. Moreover, simulation results must be validated by comparison with experimental data and/or numerical solutions computed on a finer mesh.

11.1 Heat Transfer in 'Plate Heat Exchangers'

The first example deals with the development of optimization tools for a constellation described by 'coupled stacks' with different layers (see Fig. 7). This example is quite typical for a complex flow model in the laminar regime. On the one hand, the Reynolds number is rather small due to slow fluid velocities and very small diameters. On the other hand, an unstructured grid FEM approach is required to resolve the small-scale geometrical details. Moreover, the problem at hand is coupled with additional tracer equations of convection-diffusion type. As a rule, the involved diffusion coefficients are small or equal to zero, so that the problem is transport-dominated. Hence, the key elements of an accurate and efficient solution strategy are an appropriate treatment of convective terms on unstructured meshes as well as an unconditionally stable implicit time discretization (the underlying flow field is quasi-stationary). These critical aspects will be exemplarily illustrated in what follows.

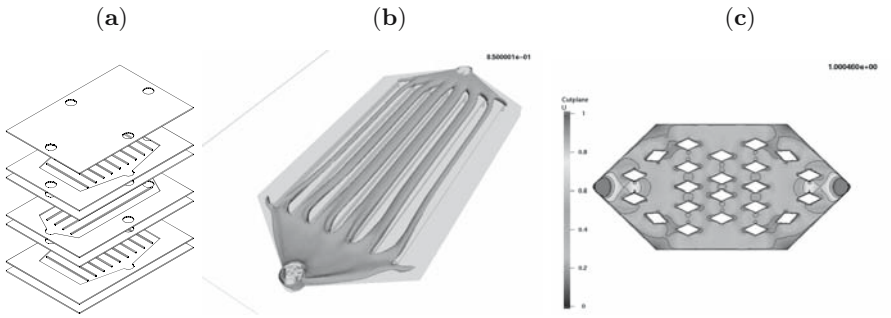


Fig. 7. Plate heat exchanger: (**a**) geometric configuration; (**b**) typical flow pattern; (**c**) velocity field for an 'optimal' distribution of internal objects

The aim of the underlying numerical study in this section is the understanding and improvement of the

- internal flow characteristics
- heat transfer characteristics

in the shown configuration which can be described by a Boussinesq model. Restricted to one stack only, the internal geometry between 'inflow' and 'outflow' holes has to be analyzed and channel-like structures or many internal 'objects' have to be placed to achieve a homogeneous flow field and, correspondingly, a homogeneous distribution of tracer substances in the interior. In order to determine the optimal shape and distribution of internal obstacles, the simulation software to be developed must be capable of resolving all small-scale details. Therefore, the underlying numerical algorithm must be highly accurate and, moreover, sufficiently flexible and robust. Algebraic FCT/TVD schemes belong to the few discretization techniques that do meet these requirements. Some preliminary results based on the incompressible Navier-Stokes equations for velocity and pressure only are presented in Fig. 7.

The following step beyond 'manual optimization' shall be a fully automatic optimization of shape, number, and distribution of the internal objects. Furthermore, the temperature equations are to be solved for each stack as well as for the whole system taking into account heat transfer both in the flow field and between the walls. In addition, chemical reaction models should be included, which gives rise to another set of coupled convection-reaction-diffusion equations. Last but not least, algebraic flux correction is to be employed at the postprocessing step, whereby the 'residence time distribution' inside each stack is measured by solving a pure transport equation for passive tracers. Since the flow field is almost stationary and allows large time steps due to the large viscosity parameters, the nonlinear transport equation has to be treated in an implicit way which is a quite typical requirement for the accurate and efficient treatment for such type of flow problems.

11.2 Bubbly Flow in Gas-Liquid Reactors

Bubble columns and airlift loop reactors are widely used in industry as contacting devices which enable gaseous and liquid species to engage in chemical reactions. The liquid is supplied continuously or in a batch mode and agitated by bubbles fed at the bottom of the reactor. As the bubbles rise, the gaseous component is gradually absorbed into the bulk liquid where it may react with other species. The geometric simplicity of bubble columns makes them rather easy to build, operate and maintain. At the same time, the prevailing flow patterns are very complex and unpredictable, which represents a major bottleneck for the design of industrial units. By insertion of internal parts, bubble columns can be transformed into airlift loop reactors which exhibit a stable circulation pattern with pronounced *riser* and *downcomer* zones (see Fig. 8). Hence, shape optimization appears to be a promising way to improve the reactor performance by adjusting the geometry of the internals.

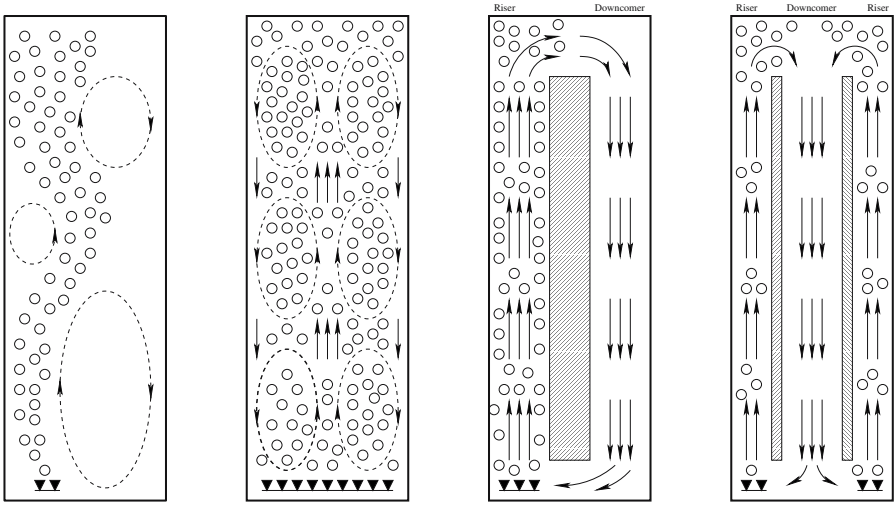


Fig. 8. Bubble columns (left) and airlift loop reactors (right)

In the present chapter, we adopt a simplified two-fluid model which is based on an analog of the Boussinesq approximation (29) for natural convection problems. At moderate gas holdups, the gas-liquid mixture behaves as a weakly compressible fluid which is driven by the bubble-induced buoyancy. Following Sokolichin *et al.* [44],[45] we assume the velocity $\mathbf{u}_L$ of the liquid phase to be divergence-free. The dependence of the effective density $\tilde{\rho}_L$ on the local gas holdup ϵ is taken into account only in the gravity force, which is a common practice for single-phase flows induced by temperature gradients.

This leads to the following generalization of the Navier-Stokes equations

$$\frac{\partial \mathbf{u}_L}{\partial t} + \mathbf{u}_L \cdot \nabla \mathbf{u}_L = -\nabla p_* + \nabla \cdot \left(\nu_T [\nabla \mathbf{u}_L + \nabla \mathbf{u}_L^T]\right) - \epsilon \mathbf{g},$$

$$\nabla \cdot \mathbf{u}_L = 0, \qquad p_* = \frac{p - p_{\text{atm}}}{\rho_L} + \mathbf{g} \cdot \mathbf{e}_g - gh, \tag{66}$$

where the eddy viscosity $\nu_T = C_\mu k^2/\varepsilon$ is a function of the turbulent kinetic energy k and its dissipation rate ε (see above). Recall that the evolution of these quantities is described by two scalar transport equations

$$\frac{\partial k}{\partial t} + \nabla \cdot \left(k\mathbf{u}_L - \frac{\nu_T}{\sigma_k} \nabla k \right) = P_k + S_k - \varepsilon, \tag{67}$$

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot \left(\varepsilon \mathbf{u}_L - \frac{\nu_T}{\sigma_\varepsilon} \nabla \varepsilon \right) = \frac{\varepsilon}{k} (C_1 P_k + C_\varepsilon S_k - C_2 \varepsilon), \tag{68}$$

where the extra source terms are due to the bubble-induced turbulence

$$P_k = \frac{\nu_T}{2} |\nabla \mathbf{u} + \nabla \mathbf{u}^T|^2, \tag{69}$$

$$S_k = -C_k \epsilon \nabla p \cdot \mathbf{u}_{\text{slip}}. \tag{70}$$

The involved slip velocity $\mathbf{u}_{\text{slip}}$ is proportional to the pressure gradient

$$\mathbf{u}_{\text{slip}} = -\frac{\nabla p}{C_W}$$

and the 'drag' coefficient $C_W \approx 5 \cdot 10^4 \, \frac{kg}{m^3 s}$ is determined from empirical correlations for the rise velocity of a single bubble in a stagnant liquid [44].

The gas density ρ_G is related to the common pressure p by the ideal gas law $p = \rho_G \frac{R}{\eta} T$, which enables us to express the local gas holdup ϵ and the interfacial area a_S per unit volume as follows [26],[28]

$$\epsilon = \frac{\tilde{\rho}_G R T}{p \eta}, \qquad a_S = (4\pi n)^{1/3} (3\epsilon)^{2/3}.$$

The *effective* density $\tilde{\rho}_G = \epsilon \rho_G$ and the number density n (number of bubbles per unit volume) satisfy the following continuity equations

$$\frac{\partial \tilde{\rho}_G}{\partial t} + \nabla \cdot (\tilde{\rho}_G \mathbf{u}_G) = -m_{\text{int}}, \tag{71}$$

$$\frac{\partial n}{\partial t} + \nabla \cdot (n \mathbf{u}_G) = 0. \tag{72}$$

The interphase momentum transfer is typically dominated by the drag force and the density of gas is much smaller than that of liquid, so that the inertia and gravity terms in the momentum equation for the gas phase can be

neglected [44],[45]. Under these (quite realistic) simplifying assumptions, the gas phase velocity $\mathbf{u}_G$ can be computed from the algebraic slip relation

$$\mathbf{u}_G = \mathbf{u}_L + \mathbf{u}_{\text{slip}} + \mathbf{u}_{\text{drift}}, \qquad \mathbf{u}_{\text{drift}} = -d_G \frac{\nabla n}{n},$$

where the drift velocity $\mathbf{u}_{\text{drift}}$ is introduced to model the bubble path dispersion by turbulent eddies. It is usually assumed that $d_G = \nu_T/\sigma_G$, where the Schmidt number σ_G equals unity. Substitution into (71)–(72) yields

$$\frac{\partial \tilde{\rho}_G}{\partial t} + \nabla \cdot (\tilde{\rho}_G(\mathbf{u}_L + \mathbf{u}_{\text{slip}}) - \nu_T \nabla \tilde{\rho}_G) = -m_{\text{int}}, \tag{73}$$

$$\frac{\partial n}{\partial t} + \nabla \cdot (n(\mathbf{u}_L + \mathbf{u}_{\text{slip}}) - \nu_T \nabla n) = 0. \tag{74}$$

Note that the contribution of $\mathbf{u}_{\text{drift}}$ gives rise to diffusive terms in both equations and it is implied that $\tilde{\rho}_G \nabla n/n = \nabla \tilde{\rho}_G$. Strictly speaking, this relation is valid only for an (almost) constant bubble mass $m = \tilde{\rho}_G/n$ but can also be used in the framework of 'operator splitting', whereby convection-diffusion and reaction-absorption processes are decoupled from one another.

The sink term m_{int} in equations (71) and (73) is due to the reaction-enhanced mass transfer. It is proportional to the interfacial area a_S and can be modeled in accordance with the standard two-film theory. The effective concentrations of all species in the liquid phase are described by extra convection-reaction-diffusion equations [23],[25],[26],[28]. If the coalescence and breakup of bubbles cannot be neglected, equation (72) should be replaced by a detailed *population balance model* for the bubble size distribution [11]. In any case, we end up with a very large system of convection-dominated PDEs which are strongly coupled and extremely sensitive to nonphysical phenomena that may result from an improper discretization of the convective terms.

In order to implement the above *drift-flux model* in a finite element code, we need to collect all the numerical tools presented so far

- algebraic flux correction schemes for the convective terms
- MPSC solvers for the incompressible Navier-Stokes equations
- block-iterative coupling mechanisms (Boussinesq approximation)
- implementation techniques for the $k-\varepsilon$ turbulence model
- adaptive time-stepping (PID control of the local gas holdup)

The segregated algorithm proposed in [28] consists of nested loops for the intimately coupled subproblems which are solved sequentially using solution values from the previous outer iteration to evaluate the coefficients and source/sink terms. In each time step, the outermost loop is responsible for the coupling of all relevant equation blocks and contains another outer iteration loop for the equations of the $k-\varepsilon$ turbulence model which are closely related to one another and must be solved in a coupled fashion. The buoyancy force in the Navier-Stokes equations is evaluated using the gas holdup from the

previous outer iteration and fixed-point defect correction is employed for all nonlinear convective terms, which gives rise to another sequence of outer iterations. The iterative process is repeated until the residual of the momentum equation and/or the relative changes of all variables become small enough.

Operator splitting tools are employed to separate convection-diffusion and absorption-reaction processes at each time step. First, all scalar quantities are transported without taking the sources/sinks into account. The homogeneous equations are decoupled and can be processed in parallel. An implicit time discretization of Crank-Nicolson or backward Euler type is performed for all equations. The value of the implicitness parameter θ and of the local time step can be selected individually for each subproblem so as to maximize accuracy and/or stability. The communication between the subproblem blocks takes place at the end of the common macro time step Δt_n which is chosen adaptively so as to control the changes of the gas holdup distribution.

The flow chart of algorithmic steps to be performed is as follows [28]

1. Recover the pressure gradient ∇p via L_2-projection.
2. Compute the associated slip velocity $\mathbf{u}_{\text{slip}} = -\frac{\nabla p}{C_W}$.
3. Solve the homogeneous continuity equation for $\tilde{\rho}_G$.
4. Update the number density n according to (74).
5. Convert $\tilde{\rho}_G$ and n into ϵ and a_S; evaluate m_{int}.
6. Solve the transport equations for concentrations.
7. Solve the ODE systems for absorption-reaction.
8. Enter the inner loop for the $k-\varepsilon$ model (67)–(68).
9. Compute the turbulent eddy viscosity $\nu_T = C_\mu \frac{k^2}{\varepsilon}$.
10. Insert ν_T and ϵ into (66) and evaluate the residual.
11. If converged, then proceed to the next time step.
12. Solve the Navier-Stokes equations and go to 1.

The first example deals with the locally aerated bubble column that was investigated in detail by Becker *et al.* [1]. The snapshots of the meandering bubble swarm displayed in Fig. 9 are in a good agreement with experimental data. The evolution of the gas holdup in the middle cross section of a prototypical airlift loop reactor is shown in Fig. 10. Aeration takes place at the bottom of the riser section where both phases flow upward. At the upper surface, the bubbles escape, while the liquid is diverted into the gas-free downcomer so as to form a closed loop. The two-phase flow reaches a steady state within a few seconds after the startup (see the right diagram). Computational results for the reaction-enhanced absorption of CO_2 in a locally aerated bubble column filled with an aqueous solution of NaOH are presented in [25],[28].

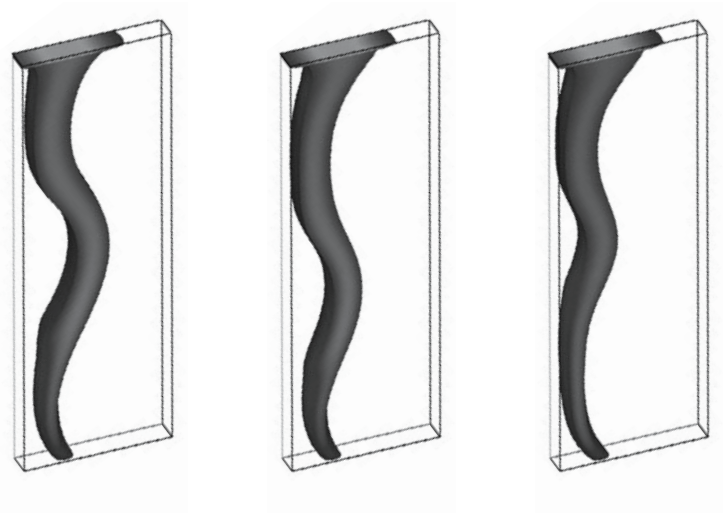


Fig. 9. Gas holdup distribution in a flat bubble column

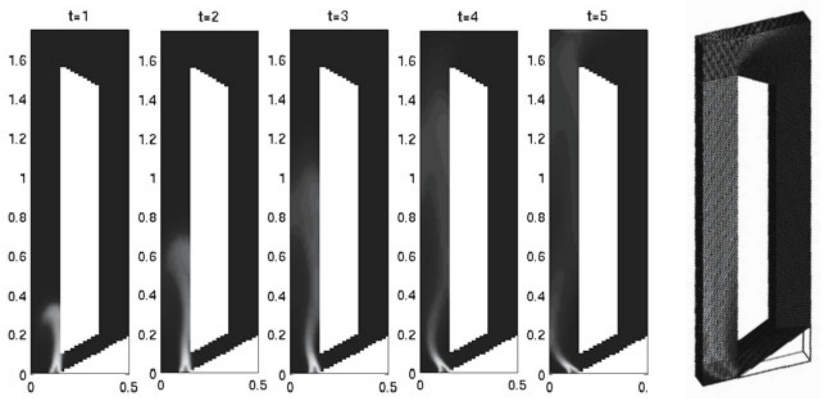


Fig. 10. Gas holdup distribution in an airlift loop reactor

11.3 Nonlinear (Granular) Flow

Another interesting example for the (non-standard) use of TVD techniques for convective operators is the numerical simulation of nonlinear incompressible fluids governed by the Navier-Stokes equations

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \nabla \cdot \mathbf{T} + \nabla p = \mathbf{f} \quad , \quad \nabla \cdot \mathbf{u} = 0 \tag{75}$$

where $\mathbf{T}$ is the stress tensor. Furthermore, the deformation rate tensor $\mathbf{D}$ and the spin tensor $\mathbf{W}$ depend on the velocity gradients as follows

$$\mathbf{D} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T) \quad , \quad \mathbf{W} = \frac{1}{2}(\nabla \mathbf{u} - \nabla \mathbf{u}^T) \tag{76}$$

In the case of Newtonian fluids, the stress tensor in (75) is given by

$$\mathbf{T} = 2\nu\mathbf{D} = \nu(\nabla\mathbf{u} + \nabla\mathbf{u}^T). \tag{77}$$

The most popular representatives of nonlinear fluids are described by *power law* models, whereby the viscosity ν in the constitutive relation (77) depends on $\mathbf{D} = \mathbf{D}(\mathbf{u})$ in a nonlinear way

$$\mathbf{T} = 2\nu(\mathbf{D})\,\mathbf{D} \quad , \quad \nu(\mathbf{D}) = \nu_0(\epsilon_1 + \epsilon_2|\mathbf{D}|)^\alpha \tag{78}$$

with certain parameters ν_0, ϵ_1, ϵ_2 and α. Fluids with $\alpha > 0$ correspond to *shear thickening*, in contrast to the case $\alpha < 0$ (*shear thinning*), both of which lead to numerical challenges. Such *differential type* models — see also *Bingham* or *Reiner-Rivlin* fluids [30],[38] — do not require implicit calculations of $\mathbf{T}$ since the tensor $\mathbf{T}$ can be represented as a (nonlinear) function of $\mathbf{D}(\mathbf{u})$. Only modifications of existing Navier-Stokes solvers have to be performed, without an additional discretization of equations for $\mathbf{T}$.

This is in contrast to the more general class of *rate type* models which couple (nonlinear) evaluations and derivatives of $\mathbf{T}$ with functional evaluations and derivatives of $\mathbf{u}$ and $\mathbf{D}$ in space and time. Examples are *Rivlin-Ericksen* and *second grade* fluids [21],[30] and particularly *Oldroyd* models [13],[38]. Defining the *objective time derivative* of the tensor $\mathbf{T}$ as

$$\frac{\mathcal{D}_a\mathbf{T}}{\mathcal{D}t} = \frac{\partial\mathbf{T}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{T} + \mathbf{T}\mathbf{W} + (\mathbf{T}\mathbf{W})^T - a[\mathbf{T}\mathbf{D} + (\mathbf{T}\mathbf{D})^T]\,, \tag{79}$$

where $-1 \le a \le 1$, a general description for *Oldroyd* models [18],[37] reads

$$\lambda_1\frac{\mathcal{D}_a\mathbf{T}}{\mathcal{D}t} + \mathbf{T} + \gamma(\mathbf{T},\mathbf{D}) = 2\mu\left(\lambda_2\frac{\mathcal{D}_a\mathbf{D}}{\mathcal{D}t} + \mathbf{D}\right). \tag{80}$$

The *relaxation time* λ_1, *retardation time* λ_2 and viscosity parameter μ may additionally depend on $\mathbf{D}$. In variants with $\gamma(\mathbf{T},\mathbf{D}) \neq 0$ (*Oldroyd 8 constants* model, *Larson* model, *Phan-Thien-Tanner* model), there is another nonlinear relationship between $\mathbf{T}$ and $\mathbf{D}$, while in the following we will concentrate on the case $\gamma(\mathbf{T},\mathbf{D}) = 0$ including the so-called *Jeffrey* models, resp., the *Maxwell* fluids. Examples of numerical simulations can be found in papers by Joseph [21], Glowinski [13] and Hron [18]. Consider the following equation for $\mathbf{T}$

$$\mathbf{T} = 2\mu_v\mathbf{D} + \mathbf{S} \quad , \quad \lambda\frac{\mathcal{D}_a\mathbf{S}}{\mathcal{D}t} + \mathbf{S} = 2\mu_e\mathbf{D}, \tag{81}$$

where $\lambda = \lambda_1$, $\mu_v = \mu\frac{\lambda_2}{\lambda_1}$ and $\mu_e = \mu - \mu_v$. Then, we can rewrite the Navier-Stokes model in (75) as

$$\mathrm{Re}\left[\frac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right] + \nabla p - (1-\alpha)\Delta\mathbf{u} - \nabla\cdot\mathbf{S} = \mathbf{f}, \qquad \nabla\cdot\mathbf{u} = 0 \tag{82}$$

$$\mathrm{We}\left[\frac{\partial\mathbf{S}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{S} + \beta_a(\mathbf{S},\mathbf{D})\right] + \mathbf{S} = 2\alpha\mathbf{D}. \tag{83}$$

Given $-1 \leq a \leq 1$, let us introduce the following definitions

$$\beta_a(\mathbf{S}, \mathbf{D}) = \mathbf{SW} + (\mathbf{SW})^T - a[\mathbf{SD} + (\mathbf{SD})^T] \tag{84}$$

$$\alpha = \frac{\mu_e}{\mu_e + \mu_v} \quad , \quad \mathrm{Re} = \frac{L\mathbf{V}\rho}{\mu_e + \mu_v} \quad , \quad \mathrm{We} = \frac{\mathbf{V}\lambda}{L}, \tag{85}$$

where L stands for a characteristic length and $\mathbf{V}$ is the velocity.

As a result, we end up with a coupled Navier-Stokes-like system (82) for the variables $(\mathbf{u}, p)$ with an additional term $\nabla \cdot \mathbf{S}$. Furthermore, there is a nonlinear nonstationary tensor-valued transport-reaction equation in (83) which involves both $\mathbf{S}$ and $\mathbf{D}(\mathbf{u})$. To solve this highly complex system, appropriate discretization techniques in space and time (implicit Euler, Crank-Nicolson or the fractional step method in conjunction with BB-stable FEM approximations) should be employed. In particular, the algebraic TVD techniques are to be recommended for the tensor-valued transport problems since there is an obvious need for a monotone, oscillation-free and highly accurate discretization of the tensor $\mathbf{T}$. Moreover, the above-mentioned MPSC approach to solving discrete saddle point problems and providing a proper coupling between the equations at hand turns out very handy for this kind of applications.

Finally, we mention the hypoplastic model proposed by Kolymbas [22] for the numerical simulation of dry cohesionless granular materials, for instance for the flow of sand in silos. This approach contains components from *rate type* as well as from *differential type* models such that the relation to the previously described *Oldroyd* model becomes evident:

$$\mathrm{Re}\left[\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}\right] = -\nabla p + \nabla \cdot \mathbf{T} + \mathbf{f}, \qquad \nabla \cdot \mathbf{u} = 0 \tag{86}$$

$$\begin{aligned} \frac{\partial \mathbf{T}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{T} = & -[\mathbf{TW} - \mathbf{WT}] \\ & + C_1 \frac{1}{2}(\mathbf{TD} - \mathbf{DT}) + C_2 tr(\mathbf{TD}) \cdot \mathbf{I} \\ & + C_3 \sqrt{tr\mathbf{D}^2}\mathbf{T} + C_4 \frac{\sqrt{tr\mathbf{D}^2}}{tr\mathbf{T}}\mathbf{T}^2 \\ & + \nu(\mathbf{D}) \left[\frac{\partial \mathbf{D}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{D} + \mathbf{DW} - \mathbf{WD}\right]. \end{aligned} \tag{87}$$

Here, $\mathbf{I}$ denotes the unit matrix, $\nu(\mathbf{D})$ a nonlinear (tensorial) function of *power law* type, and C_i are specific material constants. Again, the robust and accurate treatment of the convective terms in the absence of any second order elliptic diffusive term is a very important aspect for numerical simulations so the use of FEM-TVD techniques appears to be especially promising.

11.4 Free Surface/Interface Flows

Our last example is intended to illustrate the need for implicit TVD-type discretizations in the case of (laminar) incompressible flow problems which involve free surfaces, resp., free interfaces. The governing equations read

$$\rho_i \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) - \nabla \cdot (2\mu_i \mathbf{S}) + \nabla p = \rho_i \mathbf{g} \,, \tag{88}$$

$$\nabla \cdot \mathbf{v} = 0 \quad \text{in } \Omega_i \,, \qquad i = 1, 2, \ldots \tag{89}$$

where $\mathbf{S} = \frac{1}{2} \left(\nabla \mathbf{v} + \nabla \mathbf{v}^T \right)$ is the deformation tensor, $\mathbf{g}$ is the gravity and the computational domain $\Omega = \Omega_1 \cup \Omega_2$ with boundary Σ is shared by two immiscible fluids separated by the interface $\Gamma = \bar{\Omega}_1 \cap \bar{\Omega}_2$. The density and (dynamic) viscosity of the i-th fluid are denoted by ρ_i and μ_i, respectively.

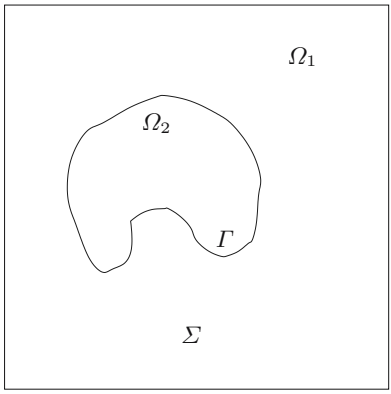


Fig. 11. Free boundary problem

This PDE system is complemented by the boundary condition

$$\mathbf{v} = \mathbf{b} \quad \text{on } \Sigma \,, \tag{90}$$

and the following initial values are prescribed at $t = 0$

$$\mathbf{v}|_{t=0} = \mathbf{v}^0 \quad \text{in } \Omega \,; \qquad \Gamma|_{t=0} = \Gamma^0 \,. \tag{91}$$

Moreover, two extra conditions can be inferred from the conservation of mass and momentum at the free/moving interface. Specifically, we have

1. *Mass balance.* If there is no mass transfer between the two fluids, then the interface Γ moves with the flow in the normal direction at velocity

$$V = \mathbf{v}_1 \cdot \mathbf{n} = \mathbf{v}_2 \cdot \mathbf{n}, \tag{92}$$

where $\mathbf{n}$ is the normal to Γ and $\mathbf{v}_i$ denotes the velocity of the i-th fluid.

2. *Momentum balance.* The interfacial stresses $\mathbf{T}_i = -p\mathbf{I} + 2\mu_i\mathbf{S},\ i = 1, 2$ are related by the following jump condition

$$(\mathbf{T}_1 - \mathbf{T}_2) \cdot \mathbf{n} = \kappa\sigma\mathbf{n}, \tag{93}$$

where σ is the surface tension coefficient and κ is twice the mean curvature of the interface.

Thus, the following relations should hold on the internal boundary Γ

$$[\mathbf{v}]\,|_\Gamma \cdot \mathbf{n} = 0\,, \qquad -\,[-p\mathbf{I} + 2\mu(\mathbf{x})\mathbf{S}]\,|_\Gamma \cdot \mathbf{n} = \kappa\sigma\mathbf{n}\,, \tag{94}$$

where $[\cdot]|_\Gamma = \lim_{\mathbf{x}\in\Omega_2\to\Gamma}(\cdot) - \lim_{\mathbf{x}\in\Omega_1\to\Gamma}(\cdot)$ and it is implied that

$$\rho(\mathbf{x}) = \begin{cases} \rho_1, & \forall\mathbf{x} \in \Omega_1, \\ \rho_2, & \forall\mathbf{x} \in \Omega_2, \end{cases}, \qquad \mu(\mathbf{x}) = \begin{cases} \mu_1, & \forall\mathbf{x} \in \Omega_1, \\ \mu_2, & \forall\mathbf{x} \in \Omega_2. \end{cases}$$

Unlike classical front tracking methods, the *level-set* approach is based on an (implicit) reconstruction of free interfaces via an 'indicator function' φ which is equal to zero directly at the interface. In contrast to the well-known *volume of fluid* (VOF) method [17], which uses a discontinuous indicator function, the level set function φ is smooth and amenable to numerical treatment. In either case, the position of the free interface can be determined by solving a pure transport equation for the corresponding indicator function

$$\frac{\partial\varphi}{\partial t} + \mathbf{v} \cdot \nabla\varphi = 0. \tag{95}$$

In what follows, we adopt the level set formulation and opt to discretize all convective terms by a high-resolution finite element scheme of TVD type.

Fig. 12. Rising bubble simulation by the level-set method

It is common practice to define φ as the 'signed distance' function which is positive in one fluid and negative in the other. By definition, we have

$$|\nabla\varphi| = 1, \qquad \varphi > 0 \quad \text{in } \Omega_1, \quad \varphi < 0 \quad \text{in } \Omega_2.$$

As a rule, the initialization of φ does not present any difficulty but it is not trivial to preserve the 'signed distance' property in the course of a numerical

simulation. Therefore, it is advisable to 'reinitialize' φ once in a while so as to provide an accurate representation of the free interface.

Given the numerical solution φ_{old} of the pure transport problem (95) after a certain number of time steps, we seek a signed distance function φ which has the same zero level set as φ_{old}. It can be recovered as the steady-state solution of another (nonlinear) transport equation which reads

$$\frac{\partial \varphi}{\partial t} = \text{sign}(\varphi_{old})(1 - |\nabla \varphi|). \tag{96}$$

This auxiliary problem for the reinitialization step can be cast in the form

$$\frac{\partial \varphi}{\partial t} + \mathbf{w} \cdot \nabla \varphi = \text{sign}(\varphi_{old}), \qquad \mathbf{w} = \text{sign}(\varphi_{old}) \frac{\nabla \varphi}{|\nabla \varphi|} \tag{97}$$

and solved by a suitable pseudo-time-stepping method. In particular, this can be accomplished in the framework of an implicit TVD-like discretization which is guaranteed to produce nonoscillatory and highly accurate results. Numerical studies indicate that a proper treatment of the level set function φ is crucial to the quality of the interface reconstruction so that the convective terms in equations (95) and (97) should be handled with extreme care.

12 Conclusions

Even for laminar flow models, there is still a strong need for better mathematical approaches as far as the discretization and solution aspects are concerned. The accuracy, flexibility, robustness and, particularly, overall efficiency of many currently available simulation tools leave a lot to be desired. The next laborious step is to develop professional CFD software for grand-challenge industrial problems. However, benchmark computations and other numerical studies demonstrate that one **must** make every effort to hone and refine the 'basic tools' before proceeding to more complex simulations! Otherwise, there might be no chance to succeed in tackling many challenging applications and to achieve not only qualitatively, but also quantitatively accurate results. Hence, the list of topics to be addressed in the near future is as follows:

- **Higher order FEM spaces:** Preliminary investigations performed by our group and others show that polynomial approximations of higher order accuracy might be preferable. Even for the Q_2/P_1 finite element pair (biquadratic velocity, piecewise linear pressure), the efficiency gains can be enormous. However, it is currently not clear how high-order finite elements would behave for nonsmooth solutions which exhibit strong gradients. In particular, an appropriate stabilization of the convective terms is necessary at high Reynolds numbers. Algebraic flux correction of FCT/TVD type seems to be feasible but its extension to the Q_2 elements is not obvious. Another primary goal is the development of hierarchical multigrid solvers

which should ensure that the potential advantages of using higher-order FEM approximations can be realized in practice.

- **Nonlinear solvers:** Since TVD-like discretization techniques are *per se* nonlinear, appropriate Newton-like methods are to be applied if an implicit treatment of the convective terms is adopted, e.g., for the 'tracer transport' in a low Reynolds number flow or at the 'reinitialization step' for the level set function. However, due to the fully discrete and inherently discontinuous nature of algebraic flux correction schemes, the derivation of the (approximate) Jacobian matrices poses a formidable problem which is still largely unsolved and calls for further research.
- **Multigrid solvers:** In a similar vein, the numerical behavior of standard (geometric) multigrid solvers for the resulting linear subproblems has not yet been investigated and requires further in-depth studies.
- **Tensor-valued transport operators:** As described for nonlinear fluids, additional hyperbolic problems with tensor-valued transport operators may need to be dealt with. An extension of our algebraic high-resolution schemes, which were originally developed for scalar quantities, to such applications should be undertaken in the near future.
- **A posteriori error control:** Since the proposed flux correction techniques are applicable to finite element methods, appropriate a posteriori error control mechanisms for Galerkin-type discretizations can be incorporated. To be more specific, residual-based error estimation via dual problems for user-specified quantities in the spirit of Rannacher and his collaborators is to be examined. Its generalization to an unstructured grid FEM equipped with algebraic flux limiters would make it possible to gain 'optimal' control over the local mesh refinement and/or coarsening.

Apart from all these mathematical challenges, the results of recent computational studies (see [49] for a critical discussion) reveal that *memory access, not data processing is costly on modern computers*. This is the major reason why traditional numerical approaches for PDEs and their realization as CFD software have massive problems with achieving a significant percentage of the possible peak performance. On the other hand, the Numerical Linear Algebra tools for the solution of sparse linear systems with millions of unknowns can be significantly improved. This can be accomplished, for instance, by resorting to cache-based implementation techniques and exploiting the local structure of data sets for block-structured grids. Such a hardware-oriented approach may yield an overall speed-up factor of as much as 10 – 1000 even on a single processor. In addition, the use of optimal parallelization strategies may increase the performance of the code by further orders of magnitude.

To realize these ambitious goals, modern numerical methods and programming concepts are to be incorporated into CFD simulation tools. Some of the latest trends are *adaptive meshing* and *a posteriori error control* as well as generalized *multigrid/domain decomposition* solvers. Furthermore, the same

philosophy applies to fully implicit Navier-Stokes solvers of MPSC type which should also be optimized so as to shift the distribution of CPU times away from expensive memory access tasks (assembly of stiffness matrices, defects and residuals, modifications of the mesh) toward more arithmetic-intensive computational work (iterative solution of linear systems by multigrid or Krylov-space methods). It is obvious that the design and implementation of such advanced software concepts is not an easy job. However, the foundations have already been laid in the framework of the ongoing FEAST project (see [50]) aimed at the development of such strategies. Iterative solvers based on the above-mentioned data structures are already available and prove remarkably efficient as compared to conventional implementation techniques.

In fact, current technological trends indicate that the demand for high-performance CFD software will continue to increase. Although the development and optimization of such advanced software products may take a very long time and consume a lot of resources, this investment is certain to yield huge gains in the long run. Indeed, numerical simulation of many 'real life' problems (e.g., turbulent and/or multiphase flows in complex 3D geometries) is still prohibitively expensive. In order to exploit the available computing power to the full extent, software engineering aspects must be accompanied by corresponding efforts in the design of optimal numerical algorithms. Moreover, the peculiarities of modern hardware architecture should be taken into account. Otherwise, many 'nice' mathematical developments may end up being impractical as long as just a small fraction (less than one percent is not unusual) of the potential peak performance can be achieved in practice.

References

1. S. Becker, A. Sokolichin and G. Eigenberger, Gas-liquid flow in bubble columns and loop reactors: Part II. *Chem. Eng. Sci.* **49** (1994) 5747–5762.
2. J. P. Boris, F. F. Grinstein, E. S. Oran und R. J. Kolbe, New insights into Large Eddy Simulation. *Fluid Dynamics Research* **10**, Nr. 4-6, 1992, 199-227.
3. A. J. Chorin, Numerical solution of the Navier–Stokes equations. *Math. Comp.* **22** (1968) 745–762.
4. M. A. Christon, P. M. Gresho, S. B. Sutton, Computational predictability of natural convection flows in enclosures. In: K. J. Bathe (ed) Proc. First MIT Conference on *Computational Fluid and Solid Mechanics*, Elsevier, 2001, 1465-1468, 2001.
5. R. Codina and O. Soto, Finite element implementation of two-equation and algebraic stress turbulence models for steady incompressible flows. *Int. J. Numer. Methods Fluids* **30** (1999) no. 3, 309-333.
6. M. Crouzeix and P. A. Raviart, Conforming and non–conforming finite element methods for solving the stationary Stokes equations. *R.A.I.R.O.* **R–3** (1973) 77–104.
7. J. Donea, S. Giuliani, H. Laval and L. Quartapelle, Finite element solution of the unsteady Navier-Stokes equations by a fractional step method. *Comput. Meth. Appl. Mech. Engrg.* **30** (1982) 53–73.

8. M. S. Engelman, V. Haroutunian and I. Hasbani, Segregated finite element algorithms for the numerical solution of large–scale incompressible flow problems. *Int. J. Numer. Meth. Fluids* **17** (1993) 323–348.
9. M. S. Engelman, R. L. Sani and P. M. Gresho, The implementation of normal and/or tangential boundary conditions in finite element codes for incompressible fluid flow. *Int. J. Numer. Meth. Fluids* **2** (1982) 225–238.
10. J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*. Springer, 1996.
11. C. Fleischer, *Detaillierte Modellierung von Gas-Flüssigkeits-Reaktoren*. Fortschr.-Ber. VDI-Reihe 3, Nr. 691, Düsseldorf: VDI Verlag, 2001.
12. V. Girault and P. A. Raviart, *Finite Element Methods for Navier–Stokes Equations*. Springer Verlag, Berlin–Heidelberg, 1986.
13. R. Glowinski, T. W. Pan, T. I. Hesla, D. D. Joseph, J. Periaux, A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flow. *J. Comput. Phys.* **169** (2001) 363–426.
14. P. M. Gresho, R. L. Sani and M. S. Engelman, *Incompressible Flow and the Finite Element Method: Advection-Diffusion and Isothermal Laminar Flow*. Wiley, 1998.
15. P. M. Gresho, On the theory of semi–implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix, Part 1: Theory, Part 2: Implementation. *Int. J. Numer. Meth. Fluids* **11** (1990) 587–659.
16. F. F. Grinstein and C. Fureby, *On Monotonically Integrated Large Eddy Simulation of Turbulent Flows Based on FCT Algorithms*. Chapter 3 in this volume.
17. C. W. Hirt, A.A. Amsden and J.L. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *J. Comput. Phys.* **14** (1974) 227-253.
18. J. Hron, Numerical simulation of visco-elastic fluids. In: *WDS'97*, Freiburg, 1997.
19. T. J. R. Hughes, L. P. Franca and M. Balestra, A new finite element formulation for computational fluid mechanics: V. Circumventing the Babuska–Brezzi condition: A stable Petrov–Galerkin formulation of the Stokes problem accommodating equal order interpolation. *Comp. Meth. Appl. Mech. Engrg.* **59** (1986) 85–99.
20. V. John, Higher order finite element methods and multigrid solvers in a benchmark problem for the 3D Navier-Stokes equations. *Int. J. Numer. Meth. Fluids* **40** (2002) 775–798.
21. D. D. Joseph, R. Glowinski, J. Feng, T. W. Pan, A three-dimensional computation of the force and torque on an ellipsoid settling slowly through a viscoelastic fluid. *J. Fluid Mech.* **283** (1995) 1-16.
22. D. Kolymbas, An outline of hypoplasticity. *Archive of Applied Mechanics* **61** (1991) 143-151.
23. D. Kuzmin, Numerical simulation of mass transfer and chemical reactions in gas-liquid flows. In: *Proceedings of the 3rd European Conference on Numerical Mathematics and Advanced Applications*, World Scientific, 2000, 237-244.
24. D. Kuzmin and S. Turek, High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter. *J. Comput. Phys.* **198** (2004) 131-158.
25. D. Kuzmin and S. Turek, Efficient numerical techniques for flow simulation in bubble column reactors. In: *Preprints of the 5th German-Japanese Symposium on Bubble Columns*, VDI/GVC, 2000, 99-104.

26. D. Kuzmin and S. Turek, Finite element discretization tools for gas-liquid flows. In: M. Sommerfeld (ed.), *Bubbly Flows: Analysis, Modelling and Calculation*, Springer, 2004, 191-201.
27. D. Kuzmin and S. Turek, Multidimensional FEM-TVD paradigm for convection-dominated flows. Technical report **253**, University of Dortmund, 2004. In: *Proceedings of the IV European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2004)*, Volume II, ISBN 951-39-1869-6.
28. D. Kuzmin and S. Turek, Numerical simulation of turbulent bubbly flows. Technical report **254**, University of Dortmund, 2004. To appear in: *Proceedings of the 3rd International Symposium on Two-Phase Flow Modelling and Experimentation*, Pisa, September 22-24, 2004.
29. A. J. Lew, G. C. Buscaglia and P. M. Carrica, A note on the numerical treatment of the k-epsilon turbulence model. *Int. J. Comput. Fluid Dyn.* **14** (2001) no. 3, 201-209.
30. J. Malek, K. R. Rajagopal, M. Ruzicka, Existence and regularity of solutions and the stability of the rest state for fluids with shear dependent viscosity. *Mathematical Models and Methods in Applied Sciences* **5** (1995) 789-812.
31. G. Medić and B. Mohammadi, NSIKE - an incompressible Navier-Stokes solver for unstructured meshes. *INRIA Research Report* **3644** (1999).
32. B. Mohammadi, A stable algorithm for the k-epsilon model for compressible flows. *INRIA Research Report* **1355** (1990).
33. B. Mohammadi and O. Pironneau, *Analysis of the k-epsilon turbulence model.* Wiley, 1994.
34. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow.* McGraw-Hill, 1980.
35. A. Prohl, *Projection and Quasi-Compressibility Methods for Solving the Incompressible Navier-Stokes Equations.* Advances in Numerical Mathematics. B. G. Teubner, Stuttgart, 1997.
36. L. Quartapelle, *Numerical Solution of the Incompressible Navier-Stokes Equations.* Birkhäuser Verlag, Basel, 1993.
37. A. Quarteroni, M. Tuveri, A. Veneziani, Computational vascular fluid dynamics. *Comp. Vis. Science* **2** (2000) 163-197.
38. K. R. Rajagopal, Mechanics of non-Newtonian fluids. In G. P. Galdi and J. Necas (eds), *Recent Developments in Theoretical Fluid Mechanics.* Pitman Research Notes in Mathematics **291**, Longman, 1993, 129-162.
39. R. Rannacher and S. Turek, A simple nonconforming quadrilateral Stokes element. *Numer. Meth. PDEs* **8** (1992), no. 2, 97-111.
40. M. Schäfer and S. Turek (with support of F. Durst, E. Krause, R. Rannacher), Benchmark computations of laminar flow around cylinder. In E.H. Hirschel (ed.), *Flow Simulation with High-Performance Computers II*, Vol. 52 von *Notes on Numerical Fluid Mechanics*, Vieweg, 1996, 547–566.
41. R. Schmachtel, *Robuste lineare und nichtlineare Lösungsverfahren für die inkompressiblen Navier-Stokes-Gleichungen.* PhD thesis, University of Dortmund, 2003.
42. P. Schreiber, *A new finite element solver for the nonstationary incompressible Navier–Stokes equations in three dimensions.* PhD Thesis, University of Heidelberg, 1996.
43. P. Schreiber and S. Turek, An efficient finite element solver for the nonstationary incompressible Navier–Stokes equations in two and three dimensions. Proc. Workshop *Numerical Methods for the Navier–Stokes Equations*, Heidelberg, Oct. 25–28, 1993, Vieweg.

44. A. Sokolichin, *Mathematische Modellbildung und numerische Simulation von Gas-Flüssigkeits-Blasenströmungen.* Habilitation thesis, University of Stuttgart, 2004.
45. A. Sokolichin, G. Eigenberger and A. Lapin, Simulation of buoyancy driven bubbly flow: established simplifications and open questions. *AIChE Journal* **50** (2004) no. 1, 24–45.
46. S. Turek, On discrete projection methods for the incompressible Navier-Stokes equations: An algorithmical approach. *Comput. Methods Appl. Mech. Engrg.* **143** (1997) 271–288.
47. S. Turek, *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach.* LNCSE 6, Springer, 1999.
48. S. Turek *et al.*, *FEATFLOW: finite element software for the incompressible Navier-Stokes equations.* User manual, University of Dortmund, 2000. Available at *http://www.featflow.de.*
49. S. Turek, C. Becker and S. Kilian, Hardware-oriented numerics and concepts for PDE software. Special Journal Issue for PDE Software, Elsevier, International Conference on Computational Science ICCS'2002, Amsterdam, 2002, FUTURE 1095 (2003), 1-23.
50. S. Turek, C. Becker and S. Kilian, Some concepts of the software package FEAST. In: J. M. Palma, J. Dongarra, V. Hernandes (eds), *VECPAR'98 - Third International Conference for Vector and Parallel Processing*, Lecture Notes in Computer Science, Springer, Berlin, 1999.
51. S. Turek and S. Kilian, An example for parallel ScaRC and its application to the incompressible Navier-Stokes equations. Proc. *ENUMATH'97*, World Science Publ., 1998.
52. S. Turek and R. Schmachtel, Fully coupled and operator-splitting approaches for natural convection. *Int. Numer. Meth. Fluids* **40** (2002) 1109-1119.
53. A. M. P. Valli, G. F. Carey and A. L. G. A. Coutinho, Finite element simulation and control of nonlinear flow and reactive transport. In: *Proceedings of the 10th International Conference on Numerical Methods in Fluids.* Tucson, Arizona, 1998: 450-45.
54. A. M. P. Valli, G. F. Carey and A. L. G. A. Coutinho, Control strategies for timestep selection in simulation of coupled viscous flow and heat transfer. *Commun. Numer. Methods Eng.* **18** (2002), no.2, 131-139.
55. S. P. Vanka, Implicit multigrid solutions of Navier–Stokes equations in primitive variables. *J. Comp. Phys.* **65** (1985) 138–158.
56. J. Van Kan, A second–order accurate pressure–correction scheme for viscous incompressible flow. *SIAM J. Sci. Stat. Comp.* **7** (1986) 870–891.

Index