

Implicit FEM-FCT algorithm for compressible flows

M. Möller¹, D. Kuzmin, and S. Turek

Institute of Applied Mathematics (LSIII), University of Dortmund, Vogelpothsweg 87,
D-44227 Dortmund, Germany, ¹*matthias.moeller@math.uni-dortmund.de*

Summary. The flux-corrected transport (FCT) methodology is generalized to implicit finite element schemes and applied to the Euler equations of gas dynamics. The underlying low-order scheme is constructed by applying scalar artificial viscosity proportional to the spectral radius of the cumulative Roe matrix. All conservative matrix manipulations are performed edge-by-edge which leads to an efficient algorithm for the matrix assembly. The outer defect correction loop is equipped with a block-diagonal preconditioner so as to decouple the discretized Euler equations and solve all equations individually. As an alternative, a strongly coupled solution strategy is investigated in the context of stationary problems which call for large time steps.

1 Introduction

The concepts of flux-corrected transport can be traced back to the celebrated SHASTA scheme proposed by Boris and Book [1] in the early 1970s. Later, their algorithm was superseded by Zalesak's multidimensional limiter [11] and carried over to finite elements by Löhner and his coworkers [7].

In recent publications [3], [4], [5] we presented a generalization of this approach to implicit finite element discretizations. A notable benefit of the new FEM-FCT formulation was the representation of anti-/diffusive terms as sums of skew-symmetric internodal fluxes. Moreover, an iterative limiting strategy was introduced which prevents the limiter from getting overly diffusive for large time steps. In this paper, we concentrate on flux correction for the Euler equations of gas dynamics and discuss various algorithmic aspects pertinent to the treatment of nonlinear hyperbolic systems.

2 FEM-FCT for scalar equations

As a model problem, consider the generic conservation law $\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) = 0$ where $\mathbf{v} = \mathbf{v}(\mathbf{x}, t)$ is a nonuniform velocity field. Let us employ standard Galerkin FEM for the discretization in space and interpolate the convective fluxes in much the same way as the sought solution [2]. After mass lumping, we get an ODE system given by

$$M_L \frac{du}{dt} = Ku \quad \Leftrightarrow \quad m_i \frac{du_i}{dt} = \sum_{j \neq i} k_{ij}(u_j - u_i) + \delta_i u_i, \quad \delta_i = \sum_j k_{ij} \quad (1)$$

where $M_L = \text{diag}\{m_i\}$ denotes the 'lumped' mass matrix and $K = \{k_{ij}\}$ stands for the discrete transport operator. The second expression in (1) corresponds to a single node i where the first term in the right-hand side is engendered by the incompressible part of the transport operator. Provided that $k_{ij} \geq 0, \forall j \neq i$, a system of such form is called *local*

extremum diminishing in the absence of the term $\delta_i u_i$ which vanishes for divergence-free velocity fields and is responsible for the physical growth of local extrema otherwise.

One major ingredient of any FCT algorithm is the nonoscillatory positivity-preserving low-order scheme which can be constructed by applying artificial diffusion $D = \{d_{ij}\}$ so as to render all off-diagonal entries of the linear operator $L = K + D$ nonnegative [3]. Hence, the optimal diffusion coefficients are given by

$$d_{ij} = d_{ji} = \max\{0, -k_{ij}, -k_{ji}\}, \quad d_{ii} = -\sum_{j \neq i} d_{ij}. \quad (2)$$

Since discrete diffusion operators have zero row/column sums [3] the diffusive term can be decomposed into skew-symmetric fluxes $(Du)_i = \sum_{j \neq i} f_{ij}$, where $f_{ij} = d_{ij}(u_j - u_i)$. Thus, the modifications in (2) are conservative. Starting with the Galerkin operator $L := K$, the low-order operator can be constructed by applying artificial diffusion edge-by-edge for each pair of nodes i and j whose basis functions have overlapping support

$$\begin{aligned} l_{ii} &:= l_{ii} - d_{ij}, & l_{ij} &:= l_{ij} + d_{ij}, \\ l_{ji} &:= l_{ji} + d_{ij}, & l_{jj} &:= l_{jj} - d_{ij}. \end{aligned} \quad (3)$$

The *discrete upwinding* technique presented above carries over to multidimensions and yields the least diffusive linear LED scheme. However, **linear** monotonicity-preserving methods can be at most first-order accurate. As a consequence, compensating antidiffusion which constitutes the difference between the discretizations of high and low order

$$P_i = \sum_{j \neq i} f_{ij}, \quad f_{ij} = -\left(m_{ij} \frac{d}{dt} + d_{ij}\right)(u_j - u_i), \quad f_{ji} = -f_{ij} \quad (4)$$

is to be constructed in a **nonlinear** way so as to remove the excessive artificial diffusion.

After an implicit time discretization we obtain a nonlinear algebraic system

$$M_L \frac{u^{n+1} - u^n}{\Delta t} = \theta L u^{n+1} + (1 - \theta) L u^n + P(u^{n+1}, u^n), \quad 0 < \theta \leq 1 \quad (5)$$

which can be solved via the fixed-point defect correction scheme

$$u^{(m+1)} = u^{(m)} + A^{-1} r^{(m)}, \quad u^{(0)} = u^n, \quad m = 0, 1, 2, \dots \quad (6)$$

Here, A is a ‘suitable’ preconditioner and $r^{(m)} = b^{(m+1)} - A u^{(m)}$ is the defect vector for the m -th iteration cycle. The latter incorporates the constant right-hand side stemming from the low-order scheme plus compensating antidiffusion

$$b^{(m+1)} = b^n + \sum_{j \neq i} \alpha_{ij} f_{ij}^{(m)}, \quad \text{where } b^n = [M_L + (1 - \theta) \Delta t L] u^n. \quad (7)$$

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 construction of the solution-dependent correction factors and of the fully discretized antidiffusive fluxes $f_{ij}^{(m)}$ is elucidated in [5]. As an alternative to (7), an iterative limiting strategy was proposed for implicit FEM-FCT schemes operated at large time steps. Roughly speaking, the amount of previously accepted antidiffusion is taken into account so that only the rejected portion of the antidiffusive flux needs to be limited at subsequent defect correction steps.

3 Euler equations

Compressible flows are governed by the Euler equations which represent a system of conservation laws for the mass, momentum and energy of an inviscid fluid. These hyperbolic PDEs are typically written in divergence form

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F} = 0, \quad \text{where} \quad \nabla \cdot \mathbf{F} = \sum_{d=1}^3 \frac{\partial F^d}{\partial x_d}. \quad (8)$$

The vector of conservative variables U and the triple of fluxes $\mathbf{F} = (F^1, F^2, F^3)$ for each direction of the Cartesian coordinate system are defined as follows

$$U = \begin{bmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \otimes \mathbf{v} + p \mathcal{I} \\ \rho H \mathbf{v} \end{bmatrix}. \quad (9)$$

Here, ρ , \mathbf{v} , p , E and $H = E + p/\rho$ stand for the density, velocity, pressure, total energy per unit mass and stagnation enthalpy, respectively. This system is completed by an equation of state $p = (\gamma - 1)\rho(E - |\mathbf{v}|^2/2)$, where $\gamma = c_p/c_v$ denotes the ratio of specific heats for a polytropic gas ($\gamma = 1.4$ for air).

By application of the chain rule, the Euler equations can be written in an equivalent quasi-linear formulation in terms of the Jacobian matrices $\mathbf{A} = (A^1, A^2, A^3)$

$$\frac{\partial U}{\partial t} + \mathbf{A} \cdot \nabla U = 0, \quad \text{where} \quad \mathbf{A} \cdot \nabla U = \sum_{d=1}^3 A^d \frac{\partial U}{\partial x_d}, \quad A^d = \frac{\partial F^d}{\partial U}. \quad (10)$$

4 Galerkin matrix assembly

In what follows, an efficient edge-based assembly technique for the standard Galerkin discretization of the Euler equations is presented. Let us start with the divergence form (8) and interpolate the fluxes using the group finite element formulation [2] which yields an ODE system similar to (1)

$$M_C \frac{d\mathbf{U}}{dt} = K\mathbf{U}. \quad (11)$$

Here, M_C denotes the block-diagonal consistent mass matrix for the coupled system and K is a discrete counterpart of the operator $-\mathbf{A} \cdot \nabla$ for the quasi-linear formulation (10). Let the entries of the mass matrix and the vector of coefficients coming from the discretization of space derivatives be defined as follows

$$m_{ij} = \int_{\Omega} \varphi_i \varphi_j \, d\mathbf{x}, \quad \mathbf{c}_{ij} = \int_{\Omega} \varphi_i \nabla \varphi_j \, d\mathbf{x}. \quad (12)$$

As long as the mesh is fixed, the coefficients \mathbf{c}_{ij} remain constant and thus the operator K can be assembled efficiently without resorting to a costly numerical integration.

Recall that basis functions sum to unity, so that the sum of their derivatives vanishes. Hence, the coefficients \mathbf{c}_{ij} satisfy $\mathbf{c}_{ii} = -\sum_{j \neq i} \mathbf{c}_{ij}$ and the right-hand side of the five coupled equations for node i is given by

$$(KU)_i = - \sum_j \mathbf{c}_{ij} \cdot \mathbf{F}_j = - \sum_{j \neq i} \mathbf{c}_{ij} \cdot (\mathbf{F}_j - \mathbf{F}_i). \quad (13)$$

In his pioneering work on approximate Riemann solvers [9], Roe showed that the differences between the components of \mathbf{F} and \mathbf{U} are related by $\mathbf{F}_j - \mathbf{F}_i = \hat{\mathbf{A}}_{ij}(\mathbf{U}_j - \mathbf{U}_i)$, where the triple of matrices $\hat{\mathbf{A}}_{ij} = (\hat{A}_{ij}^1, \hat{A}_{ij}^2, \hat{A}_{ij}^3)$ corresponds to the Jacobian tensor \mathbf{A} evaluated for the special set of density-averaged variables

$$\hat{\rho}_{ij} = \sqrt{\rho_i \rho_j}, \quad \hat{\mathbf{v}}_{ij} = \frac{\sqrt{\rho_i} \mathbf{v}_i + \sqrt{\rho_j} \mathbf{v}_j}{\sqrt{\rho_i} + \sqrt{\rho_j}}, \quad \hat{H}_{ij} = \frac{\sqrt{\rho_i} H_i + \sqrt{\rho_j} H_j}{\sqrt{\rho_i} + \sqrt{\rho_j}}. \quad (14)$$

This enables us to express the nodal value $(KU)_i$ in terms of the conservative variables

$$(KU)_i = - \sum_{j \neq i} \mathbf{c}_{ij} \cdot \hat{\mathbf{A}}_{ij}(\mathbf{U}_j - \mathbf{U}_i), \quad \text{where} \quad \mathbf{c}_{ij} \cdot \hat{\mathbf{A}}_{ij} = \sum_{d=1}^3 c_{ij}^d \hat{A}_{ij}^d. \quad (15)$$

The dot product can be interpreted as a ‘projection’ of the triple $\hat{\mathbf{A}}_{ij}$ onto the numerical edge \mathbf{ij} . For our purposes, it is expedient to introduce the splitting $\mathbf{c}_{ij} \cdot \hat{\mathbf{A}}_{ij} = -(A_{ij} + B_{ij})$, where the two components of the *cumulative Roe matrix* are defined by [5]

$$A_{ij} = \mathbf{a}_{ij} \cdot \hat{\mathbf{A}}_{ij}, \quad \mathbf{a}_{ij} = -\frac{\mathbf{c}_{ij} - \mathbf{c}_{ji}}{2}, \quad (16)$$

$$B_{ij} = \mathbf{b}_{ij} \cdot \hat{\mathbf{A}}_{ij}, \quad \mathbf{b}_{ij} = -\frac{\mathbf{c}_{ij} + \mathbf{c}_{ji}}{2}. \quad (17)$$

A similar decomposition can be performed for the contribution of the edge \mathbf{ij} to $(KU)_j$

$$\mathbf{c}_{ji} \cdot (\mathbf{F}_j - \mathbf{F}_i) = \mathbf{c}_{ji} \cdot \hat{\mathbf{A}}_{ij}(\mathbf{U}_j - \mathbf{U}_i), \quad \text{where} \quad \mathbf{c}_{ji} \cdot \hat{\mathbf{A}}_{ij} = A_{ij} - B_{ij}. \quad (18)$$

Integration by parts reveals that in the interior of the domain $\mathbf{a}_{ij} = -\mathbf{c}_{ij}$ while $\mathbf{b}_{ij} = 0$ [5], so that only the skew-symmetric part A_{ij} needs to be evaluated for interior edges. The symmetric part B_{ij} only applies to the cumulative Roe matrices for boundary edges. According to (15)–(18), the contribution of the edge \mathbf{ij} to the term KU reads

$$(A_{ij} + B_{ij})(\mathbf{U}_j - \mathbf{U}_i) \quad \longrightarrow \quad (KU)_i, \quad (19)$$

$$(A_{ij} - B_{ij})(\mathbf{U}_j - \mathbf{U}_i) \quad \longrightarrow \quad (KU)_j. \quad (20)$$

Together with the fact that the coefficients \mathbf{c}_{ij} remain constant and thus can be assembled and stored once and for all during the initialization process, (19)–(20) suggest an efficient edge-based algorithm for the matrix assembly. The underlying data structure can be generated from the sparsity pattern of the finite element matrix and contains entries for all pairs of nodes whose basis functions have overlapping supports [5]. In contrast to the scalar case this connectivity exists not only between basis functions for different nodes but also between those for different variables. Hence, each coefficient of the discrete operator is given by a square matrix of dimension equal to the number of variables.

It can be readily inferred from (19)–(20) that the contribution of the numerical edge \mathbf{ij} to the global matrix $K \in \mathbb{R}^{5N \times 5N}$ is given by

$$\begin{aligned} K_{ii} &= -A_{ij} - B_{ij}, & K_{ij} &= A_{ij} + B_{ij}, \\ K_{ji} &= -A_{ij} + B_{ij}, & K_{jj} &= A_{ij} - B_{ij}. \end{aligned} \quad (21)$$

These local Jacobians are evaluated edge-by-edge and their entries K_{ij}^{kl} , $k, l = 1, \dots, 5$ are scattered to the corresponding positions in the 25 blocks $K_{kl} \in \mathbb{R}^{N \times N}$ [5].

5 Artificial viscosity

To a large extent, the ability of a FEM-FCT algorithm to withstand the formation of wiggles depends on the quality of the underlying low-order method. For scalar transport equations we derived the least diffusive positivity-preserving scheme by elimination of negative off-diagonal entries from the discrete transport operator.

In [5] the LED-principle was generalized to hyperbolic systems by rendering all off-diagonal matrix blocks positive semi-definite. If we perform mass lumping and replace the high-order operator K in (11) by the low-order one we obtain the ODE system

$$M_L \frac{dU}{dt} = LU, \quad \text{where} \quad \begin{aligned} L_{ii} &= -A_{ij} - D_{ij}, & L_{ij} &= A_{ij} + D_{ij}, \\ L_{ji} &= -A_{ij} + D_{ij}, & L_{jj} &= A_{ij} - D_{ij}. \end{aligned} \quad (22)$$

The low-order operator L is constructed in much the same way as (21) by applying tensorial artificial viscosity $D_{ij} \in \mathbb{R}^{5 \times 5}$ to the Roe matrices. The global matrix assembly can be adopted from the previous section. The missing symmetric boundary part B_{ij} of the cumulative Roe matrix is incorporated into the raw antidiffusive fluxes

$$F_{ij} = - \left(M_{ij} \frac{d}{dt} + D_{ij} - B_{ij} \right) (U_j - U_i), \quad F_{ji} = -F_{ij}, \quad (23)$$

where $M_{ij} = m_{ij}I$ denotes the local diagonal mass matrix.

The hyperbolicity of the Euler equations implies that any linear combination of the three Jacobian matrices is diagonalizable with real eigenvalues, such that the cumulative Jacobian matrix admits the following factorization

$$A_{ij} = R(\mathbf{a}_{ij})\Lambda(\mathbf{a}_{ij})R(\mathbf{a}_{ij})^{-1}. \quad (24)$$

Let us ‘project’ the density-averaged velocity $\hat{\mathbf{v}}_{ij}$ onto the numerical edge \mathbf{ij} and define the local speed of sound as follows

$$\hat{v}_{ij} = \frac{\mathbf{a}_{ij} \cdot \hat{\mathbf{v}}_{ij}}{|\mathbf{a}_{ij}|}, \quad \hat{c}_{ij} = \sqrt{(\gamma - 1) \left(\hat{H}_{ij} - \frac{|\hat{\mathbf{v}}_{ij}|^2}{2} \right)}. \quad (25)$$

Here, $|\mathbf{a}_{ij}|$ denotes the Euclidean norm of the coefficient vector \mathbf{a}_{ij} . As a consequence, the diagonal matrix of eigenvalues can be readily computed as

$$\Lambda(\mathbf{a}_{ij}) = |\mathbf{a}_{ij}| \text{diag} \{ \hat{v}_{ij} - \hat{c}_{ij}, \hat{v}_{ij}, \hat{v}_{ij}, \hat{v}_{ij}, \hat{v}_{ij} + \hat{c}_{ij} \}. \quad (26)$$

In [5] we gave a detailed description of how to derive a generalization of Roe’s approximate Riemann solver from (24) by elimination of negative eigenvalues.

A much cheaper alternative is to add scalar dissipation proportional to the spectral radius of the Roe matrix $d_{ij} = |\mathbf{a}_{ij}|(|\hat{v}_{ij}| + \hat{c}_{ij})$ [5]. The resulting artificial viscosity operator $D_{ij} = d_{ij}I$, which in fact is the same for all components, needs to be applied only to the five diagonal blocks of the finite element matrix. Numerical examples demonstrate, that in the framework of flux correction the final solution even benefits from this slightly overdifusive low-order scheme because of an improvement in the phase accuracy [4], so that the application of a costly Riemann solver does not pay off.

6 Defect correction

After an implicit time discretization we obtain a nonlinear algebraic system similar to (5) which can also be solved by the defect correction scheme

$$\mathbf{U}^{(m+1)} = \mathbf{U}^{(m)} + [\mathbf{A}^{(m)}]^{-1} \mathbf{R}^{(m)}, \quad \mathbf{R}^{(m)} = \mathbf{B}^{(m+1)} - \mathbf{A}^{(m)} \mathbf{U}^{(m)}. \quad (27)$$

In a practical implementation the ‘inversion’ of A is performed by applying some inner iteration to solve the linear subproblem for the solution increment (an improvement of the residual by 1–2 digits suffices) and update the last iterate thereafter

$$\mathbf{A}^{(m)} \Delta \mathbf{U}^{(m)} = \mathbf{R}^{(m)}, \quad \mathbf{U}^{(m+1)} = \mathbf{U}^{(m)} + \Delta \mathbf{U}^{(m)}, \quad \mathbf{U}^{(0)} = \mathbf{U}^n. \quad (28)$$

The matrix in the left-hand side of this linear system can be replaced by a (block-diagonal) preconditioner $C^{(m)}$, so as to decouple the discretized Euler equations [5]. As an alternative we can apply the (preconditioned) BiCGSTAB method directly to the coupled system (28).

Let us split the low-order operator into its diagonal, subdiagonal and superdiagonal parts $\mathbf{A}^{(m)} = \mathbf{W}^{(m)} + \mathbf{J}^{(m)} + \mathbf{E}^{(m)}$. In [5], a block-Jacobi preconditioner $C^{(m)} = \mathbf{J}^{(m)}$ was suggested for the defect correction scheme, so that only the five diagonal blocks need to be assembled and stored

$$C_{kk}^{(m)} = J_k^{(m)} = M_L - \theta \Delta t L_{kk}^{(m)}, \quad \forall k \quad \text{and} \quad C_{kl}^{(m)} = 0, \quad \forall l \neq k. \quad (29)$$

As a consequence, the linear system (28) resolves into a sequence of scalar subproblems which can be solved separately or at best in parallel. For this purpose an iterative method, e.g. (preconditioned) BiCGSTAB or geometric multigrid, is applied to

$$\begin{aligned} J_k^{(m)} \Delta u_k^{(m)} &= r_k^{(m)}, & k &= 1, \dots, 5 \\ u_k^{(m+1)} &= u_k^{(m)} + \Delta u_k^{(m)}, & u_k^{(0)} &= u_k^n. \end{aligned} \quad (30)$$

However, this segregated solution approach disqualifies for larger time steps since severe convergence problems of the outer iteration can be observed.

Longing for the full potential of the iterative limiter and the unconditional positivity of fully implicit time-stepping we have been testing some coupled solution strategies for (28) by means of a preconditioned BiCGSTAB method. In this case, additional blocks of the low-order operator may need to be assembled and stored or there should be another (direct) way to carry out the matrix-vector multiplications for updating the residual.

Let the preconditioner for the BiCGSTAB solver be given by $\mathbf{W}^{(m)} + \mathbf{J}^{(m)}$, where

$$W_{kl}^{(m)} = M_L - \theta \Delta t L_{kl}^{(m)}, \quad \forall l < k, \quad \text{and} \quad W_{kl}^{(m)} = 0, \quad \forall l \geq k, \quad (31)$$

$$E_{kl}^{(m)} = M_L - \theta \Delta t L_{kl}^{(m)}, \quad \forall l > k, \quad \text{and} \quad E_{kl}^{(m)} = 0, \quad \forall l \leq k. \quad (32)$$

This corresponds to the block-Gauss-Seidel scheme

$$\mathbf{J}^{(m)} \Delta \mathbf{U}^{(m+1)} = \mathbf{R}^{(m)} - \mathbf{W}^{(m)} \Delta \mathbf{U}^{(m+1)} - \mathbf{E}^{(m)} \Delta \mathbf{U}^{(m)}. \quad (33)$$

Swapping the sub- and superdiagonal block-matrices in the equation above gives another variation of this algorithm. The alternating application of both subdiagonal and superdiagonal block-matrices results in a symmetric block-Gauss-Seidel approach. Recall that

$J^{(m)}$ in equation (33) is a block-diagonal matrix for which each block corresponds to a scalar problem similar to (30). They can be solved in much the same way as for the segregated solution approach. The design of ‘optimal’ preconditioners is a nontrivial task which constitutes an important field for future research.

7 Numerical examples

Let us illustrate the potential of the implicit FEM-FCT algorithm by considering a steady two-dimensional supersonic flow over a wedge. Here, the free-stream Mach number is $M = 2.5$ and the deflection angle is $\theta = 15^\circ$. The results presented at the top of Figure 1 are computed on a mesh of 128×128 bilinear elements by the low-order scheme (left) and the implicit FEM-FCT algorithm (right), respectively. It can be readily seen that the shock wave is unacceptably smeared by the low-order method. Nevertheless, both the upstream and downstream Mach numbers are predicted correctly. The iterative flux limiter resolves the shock very precisely within as few as 3–4 elements.

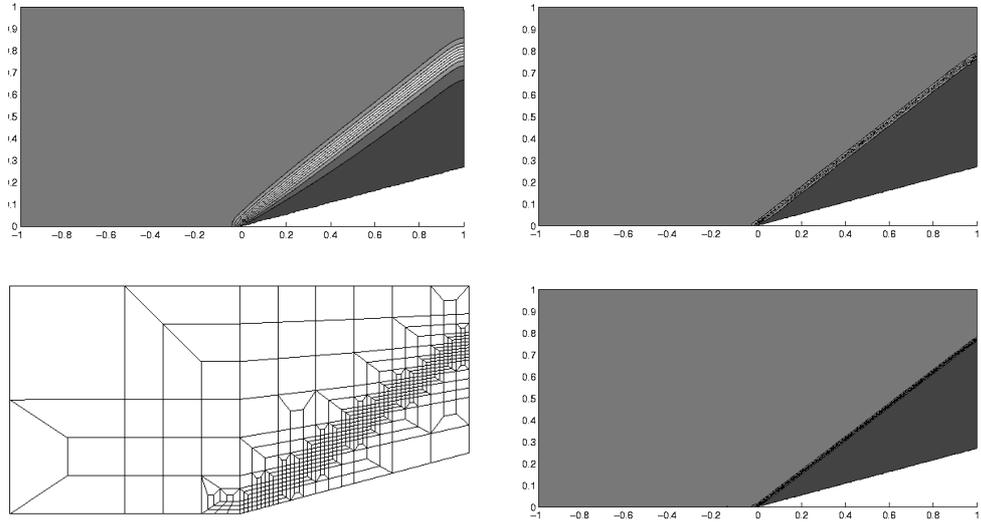


Fig. 1. Compression corner. Oblique shock at $M_1 = 2.5$, $\theta = 15^\circ$.

To demonstrate the potential ability of our discrete FEM-FCT approach to deal with unstructured meshes, we constructed an adaptive coarse grid with the grid points clustered in the vicinity of the shock wave, see Figure 1 (bottom left). After two steps of global refinement this gives the computational mesh of approximately 10,000 vertices. The resulting numerical solution (bottom right) exhibits superb accuracy and remains absolutely free of oscillations.

Numerical results for a variety of standard gas dynamic test cases encompassing both transient and stationary flows are presented in [5]. Moreover, an in-depth investigation of scalar problems can be found in the same publication.

8 Conclusions

To our knowledge, most of the finite element schemes for solving the Euler equations on unstructured grids are explicit and, consequently, subject to an restrictive CFL condition. In this paper, an *implicit* high-resolution finite element scheme for hyperbolic systems was presented making use of the flux-corrected transport paradigm. The underlying low-order operator was constructed by applying scalar artificial viscosity proportional to the spectral radius of the cumulative Roe matrix for each edge of the sparsity graph. An efficient edge-based approach to matrix assembly was proposed. The design of suitable preconditioners for both segregated and coupled solution procedures was addressed. The performance of the new FEM-FCT algorithm was illustrated for a steady supersonic flow without and with adaptive mesh refinement. The development of robust and efficient iterative solvers for implicit FEM-FCT schemes including FAS-FMG multigrid [6],[8] and an analog of the *local MPSC* smoother for the incompressible Navier-Stokes equations [10] will be addressed in forthcoming publications.

References

1. 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.
2. C. A. J. Fletcher, The group finite element formulation. *Comput. Methods Appl. Mech. Engrg.* **37** (1983) 225–243.
3. D. Kuzmin and S. Turek, Flux correction tools for finite elements. *J. Comput. Phys.* **175** (2002) 525–558.
4. D. Kuzmin, M. Möller and S. Turek, Multidimensional FEM-FCT schemes for arbitrary time-stepping. *Int. J. Numer. Meth. Fluids* **42** (2003) 265–295.
5. D. Kuzmin, M. Möller and S. Turek, High-resolution FEM-FCT schemes for multidimensional conservation laws. Technical report No. **231**, University of Dortmund, 2003, Submitted to: *Comput. Methods Appl. Mech. Engrg.*
6. P. W. Hemker and B. Koren, Defect correction and nonlinear multigrid for steady Euler equations. In: W.G. Habashi and M.M. Hafez (ed.). *Computational fluid dynamics techniques*. London: Gordon and Breach Publishers, 1995, 699–718.
7. 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.
8. J.F. Lynn, *Multigrid Solution of the Euler Equations with Local Preconditioning*. PhD thesis, University of Michigan, 1995.
9. P. L. Roe, Approximate Riemann solvers, parameter vectors and difference schemes. *J. Comput. Phys.* **43** (1981) 357–372.
10. S. Turek, *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*, LNCSE **6**, Springer, 1999.
11. S. T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.* **31** (1979) 335–362.