AIAA JOURNAL
Vol. 34, No. 10, October 1996

Implicit Upwind Residual-Distribution Euler and Navier–Stokes Solver on Unstructured Meshes

E. Issman,* G. Degrez,† and H. Deconinck‡
von Kármán Institute for Fluid Dynamics, Rhode-St-Genèse B-1640, Belgium

Implicit iterative solution techniques are considered for application to a compressible Euler and Navier–Stokes solver using upwind residual-distribution schemes on unstructured meshes. Numerical evaluation of the complete Jacobian matrix needed for the linearization process is achieved at low cost, either by finite difference approximation or by Broyden's update. It enables nonlinear solution strategies such as Newton iterative methods where linear systems are solved approximately using an accelerated iterative scheme. The linearized backward Euler scheme is used to integrate the discretized equations in time, together with a simple time-step evolution strategy. Alternatively, when this strategy fails, it is possible to use a fixed-point acceleration method that has proven quite robust. Numerical applications show the efficiency of the iterative strategy for various flow conditions.

I. Introduction

THE construction of a computational fluid dynamics (CFD) algorithm for steady-state Euler or Navier–Stokes equations can be conceptually divided into two stages, i.e., the construction of a space discretization scheme and of an iterative solution strategy, the solution strategy needing to be iterative because of the nonlinear nature of the flow equations and possibly of the space-discretization scheme. By far the most widespread iterative strategy in CFD is time-stepping, i.e., the integration in time of the unsteady equations. The simplest time-stepping strategies are explicit, e.g., forward Euler or Runge–Kutta methods. These strategies present a number of valuable advantages: They are straightforward to set up, require few parameters, and are robust. However, explicit methods are subject to stability constraints that can severely limit their convergence properties for certain classes of problems such as subsonic and viscous flows, making them prohibitively time-consuming, even on modern computers. To obtain superior convergence properties, two main approaches are widely used, i.e., multigrid acceleration of a basic explicit (or implicit) time-stepping scheme or implicit time-stepping. The former approach has been popularized in particular by the work of Jameson,[1] whereas the latter was first applied to the compressible Euler and Navier–Stokes equations by Briley and McDonald.[2] In the latter method, as well as in many methods developed subsequently, the linear system arising from the linearization of the implicit time discretization was simplified to reduce the cost of the linear solver while retaining the favorable stability properties of the unapproximated scheme. As pointed out by Degrez and Issman,[3] this can be viewed as performing a single iteration of a linear iterative (relaxation) scheme for solving the unapproximated linear system. Later, some investigators[4,5] proposed to perform several iterations of a linear iterative scheme to improve the time-stepping convergence history. More recently still, simple linear iterative schemes were substituted by more efficient Krylov subspace linear iterative schemes, in particular the GMRES algorithm developed by Saad and Schultz[6] in the mid-1980s. CFD methods based on such algorithms were first applied to structured grid schemes[7,8] but recently gained popularity for unstructured grid solvers as well.[9,10] Now, in most of these applications, the linear system matrix is not explicitly computed, either because the analytical expressions are unavailable or prohibitively expensive to compute, or because of excessive storage requirements. Instead, the matrix–vector products required by the linear iterative scheme are approximated by a finite difference expression (Fréchet derivative).

The present paper presents the construction of an implicit time-stepping method for an unstructured Euler and Navier–Stokes solver based on the recently developed multidimensional upwind residual-distribution schemes.[11–14] The compactness of the discretization stencil (at most, the Galerkin finite element stencil) allows the computation of all entries of the linear system matrix at low cost, either by a finite difference approximation or using Broyden's update algorithm.[15] Section II briefly recalls the principles of the space-discretization scheme and the basic time-stepping strategy. Section III describes the methods used to compute the residual Jacobian matrix, Sec. IV presents the linear iterative solver and preconditioning techniques, and Sec. V discusses nonlinear update strategies designed to enhance the method's global convergence. Numerical results are presented in Sec. VI for inviscid and viscous flows.

II. Computational Method

A. Space Discretization

The space discretization of the inviscid fluxes is based on the residual-distribution or fluctuation-splitting approach.[11–14] Considering a linear finite element representation of the flow variables on a triangular mesh (two-dimensional problem), the cell residual or fluctuation Φ^T is defined as the flux balance over a triangle, i.e.,

$$\Phi^T = \int_\Omega \left(\frac{\partial \boldsymbol{F}}{\partial x} + \frac{\partial \boldsymbol{G}}{\partial y} \right) \mathrm{d}\Omega = \oint_{\partial\Omega_T} (\boldsymbol{F} n_x + \boldsymbol{G} n_y)\, \mathrm{d}l \qquad (1)$$

where $\boldsymbol{F}$ and $\boldsymbol{G}$ are the inviscid flux vectors in the x and y directions, respectively. The residual-distribution method consists in distributing this residual to the three vertices of the triangle, i.e., the contribution of triangle T to the nodal residual $\boldsymbol{R}_i$ is

$$\boldsymbol{R}_i^T = \beta_i^T \Phi^T \qquad (2)$$

where β_i^T is a distribution matrix. For conservation, the distribution matrices of the three nodes of a triangle must sum to the identity matrix ($\sum_j \beta_j^T = I$). Popular schemes such as Ni's Lax–Wendroff scheme,[16] the first-order finite volume scheme, and the finite element Streamline Upwind/Petrov–Galerkin (SUPG) scheme[9] can be expressed in this general residual-distribution framework.

The essential features of the multidimensional upwind residual distribution schemes are as follows:

1) Decomposition of the cell residual Φ^T in some optimum way, generalizing the characteristic decomposition used in one-dimensional Riemann solvers,

2) Imposition of a set of properties to the distribution schemes that ensure an (almost) second-order accuracy together with a discrete steady-state maximum principle, thus precluding spurious oscillation around discontinuities.

Received May 17, 1995; presented as Paper 95-1653 at the AIAA 12th Computational Fluid Dynamics Conference, San Diego, CA, June 19–22, 1995; revision received Feb. 16, 1996; accepted for publication Feb. 21, 1996.

*Ph.D. Candidate. Student Member AIAA.
†Associate Professor. Member AIAA.
‡Professor. Member AIAA.

1. Cell Residual Decomposition

The purpose of the cell residual decomposition is to uncouple as much as possible the system of Euler equations, allowing the use of scalar distribution schemes on the uncoupled equations. First, thanks to a conservative linearization that is the two-dimensional extension of Roe's linearization,[17] Φ^T can be expressed in quasilinear form:

$$\Phi^T = S_T\left[\boldsymbol{A}\frac{\partial \boldsymbol{U}}{\partial x} + \boldsymbol{B}\frac{\partial \boldsymbol{U}}{\partial y}\right] \tag{3}$$

where S_T is the triangle surface, the flux Jacobian matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are evaluated at the Roe average state, and the gradients are computed as a function of Roe's parameter vector[18] gradient. The quasilinear expression then can be transformed and expressed in terms of any set of transformed variables V, i.e.,

$$\boldsymbol{A}\frac{\partial \boldsymbol{U}}{\partial x} + \boldsymbol{B}\frac{\partial \boldsymbol{U}}{\partial y} = \boldsymbol{T}\left(\underbrace{\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}}_{\tilde{\boldsymbol{A}}}\frac{\partial \boldsymbol{V}}{\partial x} + \underbrace{\boldsymbol{T}^{-1}\boldsymbol{B}\boldsymbol{T}}_{\tilde{\boldsymbol{B}}}\frac{\partial \boldsymbol{V}}{\partial x}\right) \tag{4}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are related by $\partial \boldsymbol{U} = \boldsymbol{T}\partial \boldsymbol{V}$.

A particularly successful choice of variables for steady flow turns out to be the set of steady characteristic variables:

$$\partial \boldsymbol{W} = \left[\beta\frac{\partial p}{\rho a} + M\partial\breve{v} \quad \beta\frac{\partial p}{\rho a} - M\partial\breve{v}, \quad \frac{\partial p}{\rho a} + M\partial\breve{u} = \frac{\partial H - T\partial s}{a}, \quad \partial p - a^2\partial\rho = (\gamma - 1)\rho T\partial s\right]^t \tag{5}$$

where $\breve{u}$ and $\breve{v}$ are the velocity components in the streamline coordinate system and $\beta = \sqrt{|M^2 - 1|}$. With this choice of variables, the cell residual reads simply

$$\Phi^T = S_T\boldsymbol{T}\begin{bmatrix} \boldsymbol{A}\cdot\nabla\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \\ \boldsymbol{u}\cdot\nabla w_3 \\ \boldsymbol{u}\cdot\nabla w_4 \end{bmatrix} \tag{6}$$

with

$$\boldsymbol{A} = \begin{pmatrix} \chi\nu^+ & \chi\nu^- \\ \chi\nu^- & \chi\nu^+ \end{pmatrix}\boldsymbol{e}_s + \begin{pmatrix} (\chi/\beta) & 0 \\ 0 & (\chi/\beta) \end{pmatrix}\boldsymbol{e}_t$$

where $\boldsymbol{e}_s$ and $\boldsymbol{e}_t$ are the unit vectors in the streamwise and transverse directions, respectively, $\nu^{\pm} = (M^2 - 1 \pm \beta^2)/2\beta^2$, and $\chi = \beta/\max(M, 1)$. The expressions for the transformation matrix T, as well as a complete derivation, are given by Paillère.[12] The last two equations always uncouple from the first two, expressing the conservation of entropy and total enthalpy along streamlines in steady inviscid flow. The acoustic subsystem formed by the first two equations diagonalizes for supersonic flow, expressing conservation of acoustic Riemann invariants along Mach lines, whereas it forms a coupled elliptic subsystem for subsonic flow. In supersonic flow, the transformed quasilinear expression of the cell residual being purely diagonal, it is possible to apply a purely scalar distribution scheme, i.e.,

$$\boldsymbol{R}_i^T = S_T\boldsymbol{T}\mathcal{D}_i\begin{bmatrix} \boldsymbol{a}_{11}\cdot\nabla w_1 \\ \boldsymbol{a}_{22}\cdot\nabla w_2 \\ \boldsymbol{u}\cdot\nabla w_3 \\ \boldsymbol{u}\cdot\nabla w_4 \end{bmatrix} \tag{7}$$

where $\mathcal{D}_i$ is the diagonal matrix of scalar distribution coefficients. For subsonic flows, the coupled nature of the acoustic subsystem imposes the use of matrix distribution schemes, e.g., the Lax–Wendroff scheme or system generalizations of the upwind scalar distribution schemes described in Sec.II.A.2 (see, e.g., Ref. 13).

This steady characteristic decomposition, also coined hyperbolic/elliptic splitting by Paillère et al.,[11] is believed to be the optimum decomposition for steady flows. Unfortunately, the transformation becomes ill-conditioned at very low speeds, which leads to severe convergence problems in flows with extended low-velocity regions, such as separated viscous flows. In these cases, alternative decompositions, such as the pseudo-Mach angle decomposition[12,14] based on unsteady characteristic variables, can be used.

2. Scalar Distribution Schemes

For the linear advection equation $\boldsymbol{\lambda}\cdot\nabla u = 0$, the cell residual ϕ^T is simply

$$\phi^T = \sum_j k_j u_j \tag{8}$$

where $k_j = \frac{1}{2}\boldsymbol{\lambda}\cdot\boldsymbol{n}_j$ and $\boldsymbol{n}_j$ is the scaled inward normal of the edge opposed to node j. Optimum distribution schemes are designed by imposing the following properties:

1) Upwind character ($\mathcal{U}$): $\beta_j = 0$ when $k_j \le 0$. This expresses that no fraction of the cell residual is distributed to upstream nodes. Note that, because of the dimension-by-dimension approach, the vertex-based first-order upwind finite volume method does not satisfy this property.

2) Positivity ($\mathcal{P}$): $R_i^T = \beta_i\phi^T = \sum_j c_{ij}u_j$ is such that $c_{ij} \le 0\,\forall\, j \ne i$. This property, called the local extremum diminishing (LED) property by Jameson,[19] ensures that the steady-state solution satisfies a discrete maximum principle, thereby precluding spurious oscillations.

3) Linearity preservation ($\mathcal{LP}$): β_j remain bounded as $\phi^T \to 0$. This property was shown to be equivalent to the absence of cross-wind diffusion, which, for a homogeneous advection problem on a regular mesh, is equivalent to second-order accuracy.[12] This was subsequently verified by numerical experiments.

4) Continuity ($\mathcal{C}$): The distribution coefficients β_j must be continuous functions of the advection angle and of the solution-gradient angle. This requirement was found essential for robustness.

For one-target triangles (triangles such that only one $k_j > 0$—there must be at least one and at most two because $\sum_j k_j = 0$), the optimum distribution satisfying properties 1–3 is to set $\beta_j = 1$ for the single downstream node and $\beta_k = 0$ for $k \ne j$. For two-target triangles, ($k_k < 0, k_i, k_j > 0$), it is not possible to construct a linear scheme that is simultaneously $\mathcal{P}$ and $\mathcal{LP}$, this being a two-dimensional generalization of Godunov's theorem. The N-scheme developed by Roe,[20] defined by

$$R_i^{T,N} = k_i^+\left(\sum_j k_j^-\right)^{-1}\sum_j k_j^-(u_j - u_i) \tag{9}$$

where $k_i^+ = \max(k_i, 0)$ and $k_i^- = \min(k_i, 0)$, is the optimum linear positive scheme in the sense that it is the linear positive scheme with minimum cross diffusion. Being linear and positive, it is thus only first-order accurate. A matrix generalization of this scheme for systems has recently been developed[13] and was used in some of the numerical tests. Nonlinear $\mathcal{P}$ and $\mathcal{LP}$ schemes then can be constructed easily by applying a limiter to the N-scheme, i.e., defining the distribution coefficient as

$$\beta_i = \varphi\left(\frac{R_i^{T,N}}{\phi^T}\right) \tag{10}$$

and imposing the following properties to the limiter function φ:

$$\varphi(r) + \varphi(1 - r) = 1 \qquad \text{consistency}$$

$$\varphi(r) \text{ bounded} \qquad \text{linearity preservation}$$

$$\varphi(r)/r \ge 0 \qquad \text{positivity}$$

All of these properties are satisfied by the min-mod limiter, which defines the positive streamwise invariant (PSI) scheme

$$\beta_i^{\text{PSI}} = \text{min-mod}\left(\frac{R_i^{T,N}}{\phi^T}\right) \tag{11}$$

Note that the limiting is applied on the same compact stencil as the original linear positive scheme, i.e., (almost) second-order accuracy is achieved without enlarging the discretization stencil. This has very favorable consequences for the development of the implicit time-stepping strategy.

3. Space Discretization of the Viscous Terms

The viscous terms are discretized using a central vertex-based finite volume technique[21] equivalent to the Galerkin finite element discretization.

B. Time-Stepping Strategy

Using the space discretization method outlined in the preceding section, the semidiscretization of the unsteady Euler and Navier–Stokes equations reads

$$\Omega_i \frac{\mathrm{d}\boldsymbol{U}_i}{\mathrm{d}t} = -\boldsymbol{R}_i = -\left(\boldsymbol{R}_i^{\mathrm{INV}} + \boldsymbol{R}_i^{\mathrm{VIS}}\right) \tag{12}$$

where Ω_i is the median dual-cell surface. The iterative strategy in the present work is used to solve the steady-state problem by linearized backward Euler time-stepping. Specifically, denoting U and R the vectors of nodal conservative variables and residuals, respectively, and Ω the diagonal matrix of median dual-cell surfaces, the time-stepping algorithm reads

Loop over time: (for $k = 0, 1, \ldots$) until convergence:
Choose time increment $\Delta^k t$,
Compute increment $\Delta^k U$ as the solution of:

$$\underbrace{\left[\frac{\Omega}{\Delta^k t} + J_R(U^k)\right]}_{J_F(U^k)} \Delta^k U = -R(U^k) \tag{13}$$

Update: $U^{k+1} = U^k + \Delta^k U$

where $J_R(U) = [\partial R(U)]/\partial U$ is the Jacobian of the residual $R(U)$, a sparse and nonsymmetric matrix, and J_F denotes the augmented Jacobian $\Omega/\Delta^k t + J_R$.

At each time-step k, the main ingredients of the algorithm can be listed as 1) choosing a time increment $\Delta^k t$, 2) computing a Jacobian matrix $J_R(U^k)$, and 3) solving the linear system (13) and carrying out the update. These issues are examined in turn.

III. Sparse Jacobian Computations

For classical spatial discretization techniques, such as flux–vector-splitting finite volume techniques, analytical expressions for the Jacobian entries can be derived relatively easily. For a computationally complex residual expression, as is the case for the present fluctuation-splitting approach, there is little possibility for computing the Jacobian analytically. To circumvent this difficulty, one can resort, when the linear systems (13) are solved iteratively, to Jacobian-free techniques where only the effect of the Jacobian on some vector quantity is approximated by a finite difference (directional or Fréchet derivative). An alternative technique consists in computing numerically each individual entry of the Jacobian. This latter approach presents several advantages:

1) It is possible to use linear algorithms that require the matrix transpose, such as biconjugate gradient (BiCG) or quasiminimum-residual (QMR).[24]

2) It is possible to derive the preconditioners from this Jacobian matrix, which enhances their quality.

3) The cost of linear iterations is minimal (sparse matrix–vector product).

For many discretizations, this approach is, however, very costly in terms of both CPU time and storage requirements. In Sec. III.A, it is shown that the present discretization technique can be implemented quite economically. Finally, rather than recomputing the full Jacobian at each iteration, it is also possible to compute approximate updates, e.g., using Broyden's algorithm, which is described in Sec. III.A.

A. Finite-Difference Jacobian Computation

One of the essential characteristics of the family of discretization schemes described in Sec. II is that they involve only distance-one neighbors. For scalar problems, each line in the Jacobian matrix has only as many nonzero entries as there are distance-one neighbors plus one (corresponding to the diagonal entry). For systems, the same conclusion holds except that Jacobian entries should be considered as $n \times n$ blocks, where n is the system dimension (4 for the two-dimensional Euler and Navier–Stokes equations). Each of these entries can be computed numerically, using the forward difference formula

$$\left[\frac{\partial \boldsymbol{R}_i(\boldsymbol{U})}{\partial \boldsymbol{U}_j}\right]_m \simeq \frac{\boldsymbol{R}_i(\boldsymbol{U}_j + \varepsilon \mathbf{1}_m) - \boldsymbol{R}_i(\boldsymbol{U})}{\varepsilon} \tag{14}$$

where $\boldsymbol{R}_i(\boldsymbol{U}_j + \varepsilon \mathbf{1}_m)$ represents the nodal residual at node i when the mth component of $\boldsymbol{U}$ at node j is perturbed by a small quantity ε. The previous formula actually gives the mth column of the block $(J_R)_{i,j}$.

This expression can be evaluated most efficiently using an assembly process similar to that used to compute the residual itself, namely loop over the cells of the mesh, compute the cell contribution to all Jacobian entries that it involves, and add it to the corresponding entry. Specifically, the algorithm is

INITIALIZE $R(U) = 0$, $J_R(U) = 0$,
Loop over triangles ($T = 1, 2, \ldots,$ nbr of cells):
  Compute fluctuation and distribute contributions
  to the three nodes ($i = 1, 2, 3$) : $\boldsymbol{R}_i \leftarrow \boldsymbol{R}_i + \boldsymbol{R}_T^i$,
  Loop over the three nodes of the cell ($j = 1, 2, 3$):
    Loop over the four components of $\boldsymbol{U}_j$ ($m = 1, 2, 3, 4$):
      —Perturb the mth component of $\boldsymbol{U}_j \leftarrow \boldsymbol{U}_j + \varepsilon \mathbf{1}_m$,
      —Compute new fluctuation,
      —Distribute the three contributions ($i = 1, 2, 3$):

$$\left[\frac{\partial \boldsymbol{R}_i(\boldsymbol{U})}{\partial \boldsymbol{U}_j}\right]_m \leftarrow \left[\frac{\partial \boldsymbol{R}_i(\boldsymbol{U})}{\partial \boldsymbol{U}_j}\right]_m + \left[\boldsymbol{R}_i^T(\boldsymbol{U}_j + \varepsilon \mathbf{1}_m) - \boldsymbol{R}_i^T(\boldsymbol{U})\right]/\varepsilon$$

where $\boldsymbol{R}_i^T(\boldsymbol{U}_j + \varepsilon \mathbf{1}_m)$ denotes the residual contribution to node i when the mth component of $\boldsymbol{U}$ at node j has been perturbed. It results that all entries in the Jacobian can be computed with a cost equal to that of only $3 \times 4 = 12$ (in three-dimensional, 20) residual evaluations.

A key issue to the numerical computation of the Jacobian as a finite difference approximation is the proper choice of ε, which can be determined here on a component-by-component basis. The question is treated by Dennis and Schnabel,[15] who advocate

$$\varepsilon = \sqrt{\eta} \max[|\boldsymbol{U}_{j,m}|, \mathrm{typ}(\boldsymbol{U}_{j,m})]\mathrm{sign}(\boldsymbol{U}_{j,m}) \tag{15}$$

with $\mathrm{typ}(\boldsymbol{U}_{j,m})$ a typical user-defined order of magnitude for the mth component of U at node j and η a lower bound on the inaccuracy in the residual $R(U)$ evaluation (relative noise). At best, this lower bound is the machine-epsilon of the computer and can be larger if $R(U)$ is computed by a lengthy piece of code.

B. Broyden's Method

Broyden's update method is the multidimensional extension of the secant method used for univariate problems, avoiding the need for computing any derivative. If the kth Newton step is denoted (the time-step has been ignored from the formulation. However, the argumentation that follows still holds, because backward Euler discretization in time amounts to a classical Newton's method where the increment $\Delta^k U$ has been underrelaxed for the update) by

$$J_R(U^k)\Delta^k U = -R(U^k)$$

with $\Delta^k U = U^{k+1} - U^k$, the generalization of the one-dimensional secant condition is that $J_R(U^{k+1})$ satisfies

$$J_R(U^{k+1})\Delta^k U = \Delta^k R \tag{16}$$

where $\Delta^k R = R(U^{k+1}) - R(U^k)$. However, this does not determine $J_R(U^{k+1})$ uniquely in more than one dimension. In Broyden's update approach, $J_R(U^{k+1})$ is chosen by making the least change (see Ref. 15 for proper matrix norms) to $J_R(U^k)$, consistent with condition (16). As such, the method suffers a major drawback because it entails a complete fill-in of the Jacobian matrix, whereas the true Jacobian matrix is sparse. Alternatively, we can look for the solution to the same least-change problem under the additional condition $B \in S(J_R)$ where $S(J_R)$ represents the set of $n \times n$ matrices with the same sparsity pattern as J_R. The resulting update is given by

$$J_R(U^{k+1}) = J_R(U^k) + \mathcal{P}_{S(J_R)}\left\{D^{-1}\left[\Delta^k R - J_R(U^k)\Delta^k U^t\right]\Delta^k U\right\}$$

where $\mathcal{P}_{S(J_R)}$ is the matrix operator that maps any matrix onto the same matrix but is restricted to the sparsity pattern of J_R and D is an $n \times n$ diagonal matrix that accounts for the sparsity structure of the Jacobian matrix:

$$D_{ii} = \delta_i^{k\,t}\delta_i^k \qquad \text{with} \qquad \left(\delta_i^k\right)_j = \begin{cases} 0 & \text{if} \quad (J_R)_{i,j} = 0 \\ (\Delta^k U)_j & \text{otherwise} \end{cases} \tag{17}$$

where δ_i^k is the restriction of $\Delta^k U$ to the discretization stencil at node i and the superscript t denotes the transpose. Broyden's method allows the Jacobian matrix to be updated without having to compute 12 residual evaluations. On the other hand, nonlinear convergence will be, at most, linear, and more iterations will be needed at the nonlinear level.

IV. Solution of the Linear System

A. Iterative Solver

For symmetric positive definite linear systems, the optimum iterative scheme is the well-known conjugate-gradient method, or its twin, the conjugate-residual method. The methods are optimal in the sense that they minimize in some sense the error or the residual but they require only simple (two-term) recurrences. For nonsymmetric systems, it is impossible to achieve these properties simultaneously. Schemes such as GMRES[6] keep the residual minimization property, but they require that all search directions to be stored. Conversely, schemes such as the BiCG[22] or conjugate-gradient squared (CGS)[23] require only simple recurrences, but they give up the minimum residual property. The QMR method[24] and its transpose-free variant (TFQMR)[25] require only simple recurrences and enforce a residual quasiminimization that provides a smoother convergence than the corresponding BiCG and CGS methods. In the present study, the GMRES, QMR, and TFQMR schemes are used.

B. Preconditioner

It is well known that the conjugate-gradient method and, generally, all Krylov subspace methods are not very efficient unless the linear system is preconditioned. In the present study, left preconditioning has been used, i.e., the iterative scheme is used to solve the preconditioned system

$$\tilde{J}_F(U^k)^{-1} J_F(U^k)\Delta^k U = -\tilde{J}_F(U^k)^{-1} R(U^k) \tag{18}$$

where $\tilde{J}_F(U^k)$ is the preconditioner. Right preconditioning also was considered and gave similar results.

Block incomplete lower upper (ILU)0, modified ILU (MILU), and symmetric Gauss–Seidel (SGS) preconditioners have been considered. They are all approximate (incomplete) LU factorizations of the system matrix J_F, i.e., $\tilde{J}_F(U^k) = LP^{-1}U$ with $\text{Diag}(L) = \text{Diag}(U) = P$, where L and U are lower and upper triangular matrices, respectively.

ILU(0): L, P, and U are computed using the exact factorization expressions, except that L and U are limited to the sparsity pattern of J_F.

MILU: Same as ILU, except that L and U are taken directly from J_F.

SGS: Simplest choice corresponding to $L = \text{Lower}(J_F)$, $P = \text{Diag}(J_F)$, and $U = \text{Upper}(J_F)$.

The three preconditioning techniques differ not only by their computational cost but, more important, by their storage requirements. Whereas ILU(0) requires for the preconditioner an additional storage equal to that of the original matrix, MILU requires only the additional storage of a (block) diagonal matrix and SGS requires no additional storage. Details of the sparse-matrix implementation are given in Ref. 26.

V. Nonlinear Iterative Strategies

As mentioned in Sec. II, the iterative strategy to reach a steady solution used in the present work is linearized backward Euler time-stepping. This involves a free parameter, the time-step $\Delta^k t$, that should be selected to optimize the nonlinear convergence. Although the backward Euler scheme is unconditionally stable from the point of view of linear stability theory and offers the best asymptotic rate of convergence for $\Delta^k t \to \infty$, the choice of $\Delta^k t = \infty$, which corresponds to Newton's method, is not universally optimal because of the limited radius of convergence of Newton's method. What is at each time level the optimum time step, i.e., the time step that will minimize the number of iterations to convergence remains, unfortunately, an unanswered question. This fact is all the more embarrassing because the choice of an inadequate time increment can be very detrimental to the overall convergence performance.

In this study, two nonlinear iterative strategies are used, namely, a simple time-step amplification strategy, possibly combined with the linesearch backtracking algorithm,[27] and a fixed-point method acceleration strategy, which are briefly described.

Finally, in general, Broyden's Jacobian update proved less sensitive to the nonlinear iterative strategy than the finite difference Jacobian computation, although, as mentioned above, it yields, at most, a linear convergence rate.

A. Time-Step Amplification Strategy

In this very simple strategy, the time step is controlled in the following way:

1) Pick an initial Courant–Friedrichs–Lewy (CFL) number CFL_0,

2) At each time step, compute the new CFL number as $\text{CFL}^k = \min(C_2\ \text{CFL}^{k-1},\ \text{CFL}_{\text{MAX}})$, where C_2 and CFL_{MAX} are two user-specified constants.

This simple strategy worked correctly for many cases, but not for very difficult cases such as second-order computations of separated viscous flows, with C_2 in the range 1.5–2 and CFL_{MAX} in the range 10^2–10^5, depending on the flow case and the discretization scheme. Its robustness can be enhanced slightly by combining it with the linesearch backtracking algorithm,[27] whose principle is to underrelax the solution update $\Delta^k U$ if it results in a nonlinear residual norm increase. The success of the linesearch backtracking algorithm lies in the fact that the solution update is in a descent direction. This has been proven to be true for the Newton–GMRES scheme[28] and was found experimentally to be true also in many instances for backward Euler–GMRES. When linesearch backtracking is applied, the time-step control strategy described above is slightly modified, i.e., rather than being multiplied by an amplification factor C_2, the time step is multiplied by a reduction factor C_1 (typically 0.5) if the previous step required underrelaxation.

B. Acceleration of Fixed-Point Method

The fixed-point-method acceleration strategy is based on the observation that, in the iterative solution of the linear systems (13), nothing imposes the condition that the time step used in the preconditioner $\tilde{J}_F$ be the same as in the augmented Jacobian J_F. Now, if the linear systems (13) were solved exactly or almost exactly (with a very low linear convergence threshold), changing the time step in the preconditioner would affect only the number of linear iterations. However, because the number of linear iterations is limited and the solution of the linear system at each time step thus is far from being exact, the time step in the preconditioner (which may be viewed as a linear relaxation factor) actually influences the nonlinear convergence history.

The fixed-point-method acceleration strategy corresponds to taking Newton's method as the global iterative strategy [i.e., $\Delta^k t = \infty$ in Eq. (13)] together with a moderate time step $\Delta^k t_{\text{pre}}$ in the preconditioner. This approach, which was already used by Issman and Degrez,[5] is building up the main features of the flow at the very early stages of the convergence process much faster than the classic backward Euler discretization in time. However, if the time step in the preconditioner and/or the number of linear iterations are not allowed to increase, the method never reaches the quadratic convergence properties of Newton's method. The method appears, therefore, as complementary to the time-step amplification strategy because it can be used for the first nonlinear iterations and provide, so doing, a well-featured initial guess for backward Euler.

We provide below an approximate and incomplete interpretation of the scheme, justifying the name we gave to it: the accelerated fixed-point method. At the early stages of implicit methods, the basic technique consisted of simple point- or line-relaxation procedures over the computational domain, immediately followed by a nonlinear update. The relaxation procedure would be based on some

approximation $\tilde{J}_F(U^k)$ of the augmented Jacobian of the residual $J_F(U^k) = 1/\Delta^k t + J_R(U^k)$, and the overall process would look like a fixed-point method, such as

$$U^{k+1} = U^k - \tilde{J}_F^{-1}(U^k)R(U^k)$$

$$= \mathcal{H}(U^k)$$

This formulation is a particular case of the single-step backward Euler time-stepping where only one single iteration is performed at the linear level, with $\tilde{J}_F(U^k)$ as the preconditioning matrix. Then, let us define $\mathcal{G}(U) = U - \mathcal{H}(U)$ and apply Newton–GMRES to solve $\mathcal{G}(U) \equiv 0$:

$$U^{k+1} = U^k + \Delta^k \qquad \text{with} \qquad J_G(U^k)\Delta^k = -G(U^k) \tag{19}$$

If we approximate J_G by $\tilde{J}_F^{-1} J_R$ (which is only true at the nonlinear convergence), we obtain

$$U^{k+1} = U^k + \Delta^k \qquad \text{with} \qquad \tilde{J}_F^{-1} J_R \Delta^k = -\tilde{J}_F^{-1}(U^k)R(U^k)$$

which is nothing other than full Newton iterations to solve $R(U) = 0$, where the system arising at each linearization has been left-preconditioned by $\tilde{J}_F^{-1}$. In practice, the technique indeed adds $1/\Delta^k t$ only in the preconditioning matrix.

Numerical experiments have shown that the accelerated fixed-point method requires CFL numbers of $\mathcal{O}(1)$, $\mathcal{O}(10)$.

VI. Numerical Results

Three test cases have been chosen to demonstrate the performance of the method in different flow configurations. Additional cases are presented in Ref. 26. Convergence histories show the absolute L2 norm of the density residual relative to the number of nonlinear iterations and/or CPU time. All tests were performed on a DEC Alpha AXP 3000/400 workstation. Unless specified otherwise, computations were performed with the following options: time-step amplification strategy, finite difference Jacobian computation, and GMRES linear solver with a constant Krylov subspace dimension of 30 and ILU left-preconditioning. The linear solver was stopped if the normalized linear residual dropped below 10^{-5}. This threshold was generally reached only when the CFL number was small.

A. Subsonic Flow in a Sine-Bump Channel

The inviscid flow in a sine-shaped channel is computed on a mesh made of 2208 cells. Subsonic inlet and outlet boundary conditions are applied, corresponding to a Mach number at the inlet $M_{\text{in}} = 0.5$. The solution is isentropic and irrotational, and Mach isolines are expected to be completely symmetrical. This rather academic test case is considered for the purpose of comparing the various options (Jacobian computation, linear solver) described above. The effect of the preconditioner is also studied in Ref. 26.

Computations are made with the hyperbolic/elliptic splitting model, with SUPG and PSI distribution schemes on the acoustic system and the two decoupled equations for entropy and enthalpy, respectively. Figure 1 shows the grid and Mach-number contours for this configuration. The maximum entropy deviation {computed as $\Sigma = [(p/p_\infty)/(\rho/\rho_\infty)^\gamma] - 1$} is equal to 0.000091, a very low value. Table 1 presents the L_2 norm of the entropy deviation for a sequence of grids of increasing refinement. The results, taken from Ref. 12, confirm the (almost) second-order accuracy of the method.

Convergence histories for $M_{\text{in}} = 0.5$ are depicted in Fig. 2 for three different linear iterative algorithms: GMRES, QMR, and TFQMR. Convergence to 10^{-13} is reached in about 40 iterations in less than 5 CPU minutes. The maximum CFL number was set to 5×10^4. It can be seen that GMRES performs better than QMR and TFQMR, essentially in terms of CPU time. The effect on the convergence of computing the Jacobian numerically or updating it by Broyden's formula also was investigated, both with GMRES at the linear level. Results are shown in Fig. 3. Broyden's update performs as expected: More than twice as many iterations are needed to reach convergence, but it is about as fast in terms of CPU time.

Low-Mach-number computations corresponding to $M_{\text{in}} = 0.1$, 0.01, 0.001 were performed by premultiplying the nodal residual

Table 1 Grid convergence study for the hyperbolic/elliptic scheme

No. of points	(548)	(2208)	(8930)	Order
Entrophy deviation	0.000304	0.000091	0.000029	1.80

Table 2 Summary of results for the subcritical multielement airfoil

$\sum C_L$	$\sum C_D$	$M_{\min}$	$M_{\max}$	$\Sigma_{\min}$	$\Sigma_{\max}$
5.052	0.065	0.002	0.705	−0.0008	0.0005

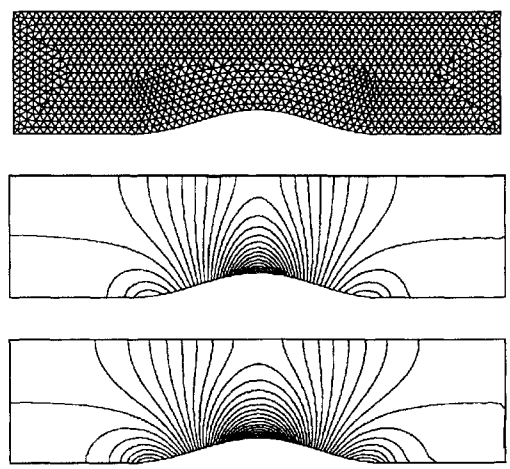


Fig. 1 Test case 1: mesh (top), Mach isolines $M_\infty = 0.5$ (middle), and $M_\infty = 0.001$ (bottom).

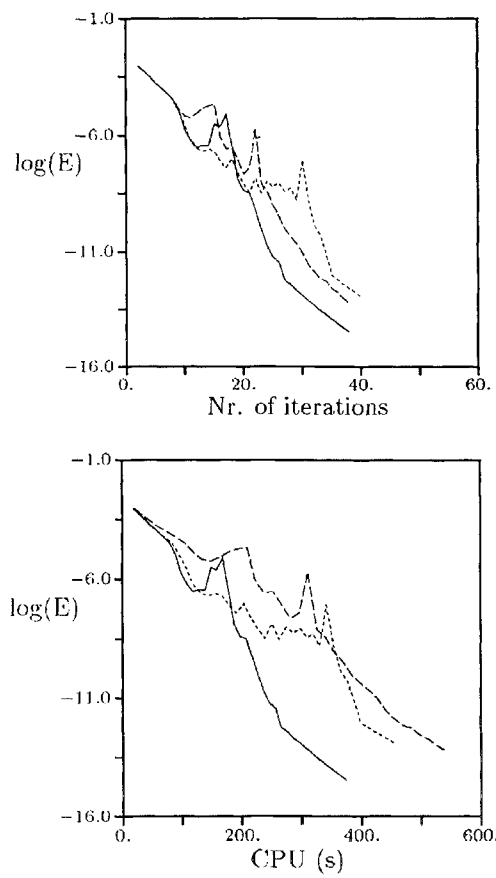


Fig. 2 Test case 1: comparison of ——, GMRES; – – –, QMR; and - - - -, TFQMR.

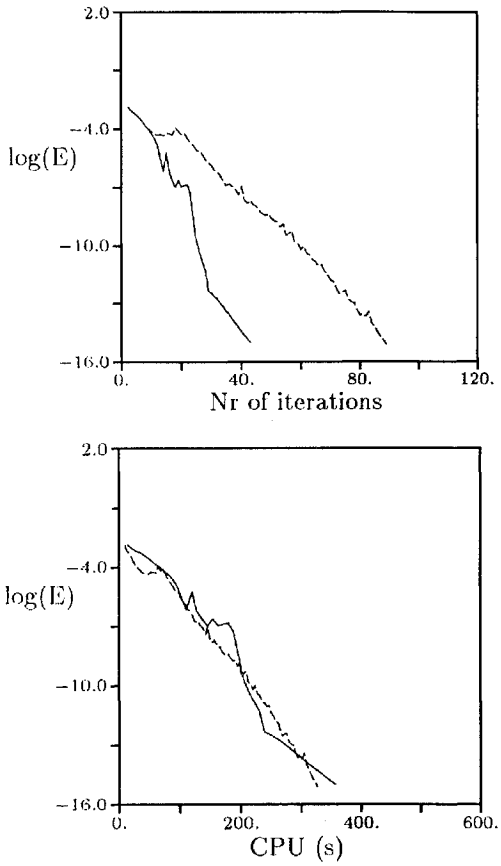


Fig. 3 Test case 1, comparison of Jacobian computed by finite difference (FD) and by Broyden's update (BROYD): ——, FD and – – –, BROYD.

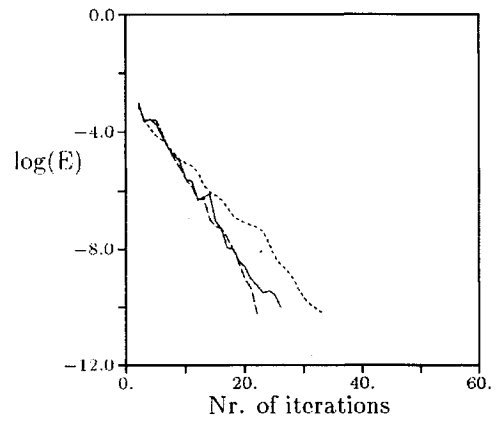


Fig. 4 Test case 1, low-Mach-number convergence histories: ——, M_∞ = 0.1; – – –, M_∞ = 0.01; and - - - -, M_∞ = 0.001.

with the van Leer–Lee–Roe preconditioning matrix[29] and using the accelerated fixed-point method with a starting CFL number of 1 increasing linearly up to 10. Solutions obtained are similar and Mach isoline contours for $M_{in} = 0.001$ are shown in Fig. 1. Convergence histories, shown in Fig. 4, exhibit the same behavior, irrespective of the Mach-number value.

B. Subcritical Flow over a Multielement Airfoil

The flow over a four-element airfoil configuration is computed with freestream condition $M_\infty = 0.2$ and 0-deg angle of attack on a grid made of 10,018 cells. The computation is first carried out with a first-order matrix N-scheme. Almost quadratic convergence is observed, as shown in Fig. 5a, requiring 25 nonlinear iterations to reach machine accuracy in 1711 CPU seconds. Restarting on this converged solution, the second-order-accurate hyperbolic/elliptic model with SUPG/Lax–Wendroff distribution scheme

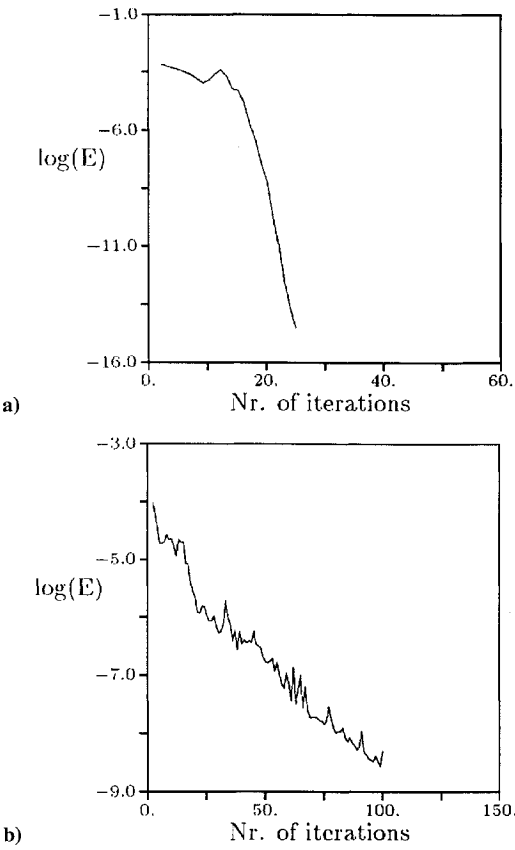


Fig. 5 Test case 2, convergence histories for computations: a) first order and b) second order.

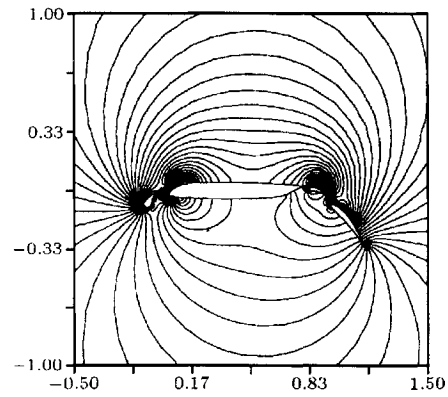
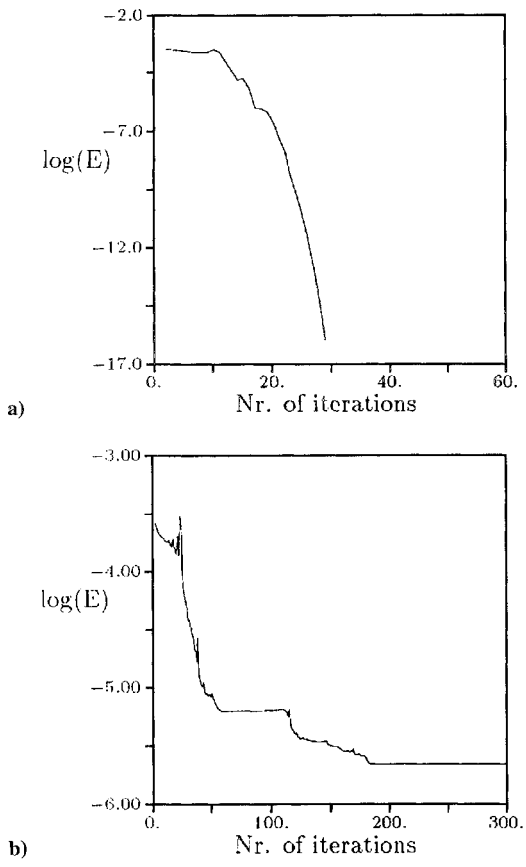


Fig. 6 Test case 2, Mach isolines.

was used with backward Euler and a maximum CFL number equal to 100. Convergence history is depicted in Fig. 5b, corresponding to a CPU elapsed time of 5507 s. Mach-number contours of the final solution are shown in Fig. 6. They indicate the very low numerical entropy generation. Table 2 presents the maximum and minimum values of entropy, Mach number, and the total values of the lift and drag coefficients for this computation.

C. Hypersonic Flow over a 7.5-deg Ramp

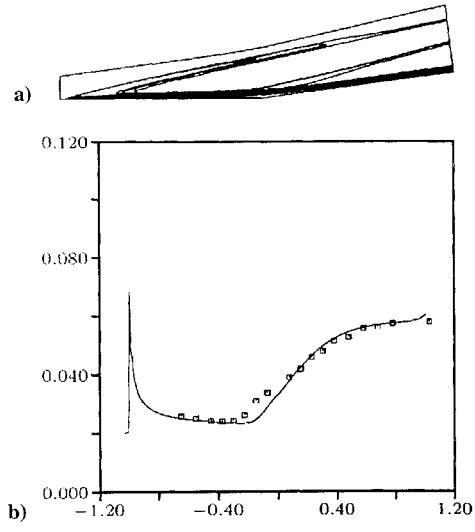
Laminar flow over a 7.5-deg deflected ramp with a freestream Mach number of M_∞ of 5.96 and a Reynolds number based on the hinge line of $Re = 2.77 \times 10^5$ is computed on a structured 93×49 grid with maximum aspect ratio of about a hundred. This particularly demanding test case is first computed with a first-order-accurate matrix N-scheme. Almost-Newton convergence can be achieved with a starting CFL number of 1 increased by a factor $C2 = 1.5$ at each iteration, as shown in Fig. 7a, for a total CPU time of 1700 s. The computation is then redone using the second-order-accurate pseudo-Mach angle decomposition with PSI distribution scheme. For this



a)

b)

Fig. 7 Test case 3, convergence histories for computations: a) first order and b) second order.



a)

b)

Fig. 8 Test case 3, a) Mach isolines and b) pressure distribution: comparison with experimental results (□).

purpose, an accelerated fixed-point method with a CFL of 0.1 could be used to converge the solution starting from uniform flow in about a hundred iterations, as shown in Fig. 7b, requiring less than 3600 CPU seconds. Mach-number contours are shown in Fig. 8a, where the development of the boundary layer and the shock generated by viscous interaction can be observed as well as the interaction of the shock and the boundary layer. The surface pressure distribution is shown in Fig. 8b, together with experimental results by Gautier.[30] The separated bubble is slightly underestimated, indicating the limits of the spatial discretization scheme currently used.

VII. Conclusions

Implicit iterative schemes for an upwind residual-distribution unstructured Euler and Navier–Stokes solver are presented. The first element of an implicit method is the computation of a Jacobian matrix $J_R(U)$. This can be achieved here by a sparse finite difference approximation method, at the low cost of 12 residual evaluations. Essential constituents to make this approach successful are a compact discretization scheme (without prejudice to accuracy) and a flexible sparse matrix technology. Alternatively, the Jacobian matrix can be updated by a sparse Broyden's formula without computing any derivative. Although the latter method is likely to require more iterations at the nonlinear level, it may be as efficient in CPU time as using the real Jacobian. In both cases, preconditioning matrices can be derived easily, possibly at no extra storage cost, and appear to be quite reliable. The most delicate issue was the time-step selection strategy. Although a simple amplification strategy performs well in numerous cases, it can break down in the toughest situations, for which the fixed-point acceleration method represents a valuable alternative.

Numerical results demonstrate the performances of the iterative strategies set forth. Almost-Newton convergence is achieved for first-order and simple second-order computations. For more complex second-order computations, although Newton convergence is not achieved, the implicit iterative scheme still far outperforms explicit time-stepping.

Acknowledgments

E. Issman was financially supported by the Belgian "Fonds pour la formation à la recherche dans l'industrie et l'agriculture." The paper also benefited greatly from the reviewers' comments.

References

[1]Jameson, A., "Solution of the Euler Equations for Two Dimensional Transonic Flow by a Multigrid Method," *Applied Mathematical Computations*, Vol. 13, No. 13, 1983, pp. 327–356.

[2]Briley, W. R., and McDonald, H., "Solution of the Three-Dimensional Compressible Navier–Stokes Equations by an Implicit Technique," *Lecture Notes in Physics*, Vol. 35, 1975, pp. 105–110.

[3]Degrez, G., and Issman, E., "Solving Compressible Flow Problems with Subspace Iteration Methods," *Proceedings of the 3rd International Congress on Industrial and Applied Mathematics* (Hamburg, Germany) (to be published).

[4]Whitfield, D. L., and Taylor, L. K., "Discretized Newton-Relaxation Solution of High Resolution Flux-Difference Split Schemes," *Proceedings of the AIAA 10th Computational Fluid Dynamics Conference*, AIAA, Washington, DC, 1991 (AIAA Paper 91-1539).

[5]Issman, E., and Degrez, G., "Convergence Acceleration of a 2D Euler/Navier–Stokes Solver by Krylov Subspace Methods," *Computational Fluid Dynamics '94*, edited by S. Wagner, E. H. Hirschel, J. Périaux, and R. Piva, Wiley, 1994, pp. 348–355.

[6]Saad, Y., and Schultz, M. H., "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Inear Systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856–869.

[7]Venkatakrishnan, V., "Preconditioned Conjugate Gradient Methods for the Compressible Navier–Stokes Equations," *AIAA Journal*, Vol. 29, No. 7, 1991, pp. 1092–1110.

[8]Orkwis, P., and McRae, S., "A Newton's Method Solver for the Axisymmetric Navier–Stokes Equations," *AIAA Journal*, Vol. 30, No. 6, 1992, pp. 1507–1524.

[9]Johan, Z., Hughes, T. J. R., and Shakib, F., "A Globally Convergent Matrix-Free Algorithm for Implicit Time-Marching Schemes Arising in Finite Element Analysis in Fluids," *Computational Methods in Applied Mechanics and Engineering*, Vol. 87, 1991, pp. 281–304.

[10]Venkatakrishnan, V., and Mavriplis, D. J., "Implicit Solvers for Unstructured Meshes," *Journal of Computational Physics*, Vol. 105, No. 1, 1991, 83–91.

[11]Paillère, H., Deconinck, H., and Roe, P., "Conservative Upwind Residual-Distribution Schemes Based on the Steady Characteristics of the Euler Equations," *Proceedings of the AIAA 12th Computational Fluid Dynamics Conference*, AIAA, Washington, DC, 1995 (AIAA Paper 95-1700).

[12]Paillère, H., "Multidimensional Residual Distribution Schemes for the Euler and Navier–Stokes Equations on Unstructured Grids," Ph.D. Thesis, Faculté des Sciences Appliquées, Univ. Libre de Bruxelles, Belgium, 1995.

[13]Paillère, H., Carette, J.-C., van der Weide, E., Issman, E., Deconinck, H., and Degrez, G., "Implicit Multidimensional Upwind Residual Distribution Schemes on Adaptive Meshes," *Progress and Challenges in CFD Methods and Algorithms* (Seville, Spain), No. 17, AGARD, 1995.

[14]Paillère, H., and Deconinck, H., "A Review of Multidimensional Upwind Residual Distribution Schemes for the Euler Equations," *Computational Fluid Dynamics Review*, edited by K. Oshima and M. Hafez, Wiley, New York, 1995.

[15]Dennis, J. E., and Schnabel, R. D., *Numerical Methods for Unconstrained Optimization and Non-Linear Equations*, Series in Computational Mathematics, Prentice–Hall, Englewood Cliffs, NJ, 1983, Chap. 8.

[16]Ni, R. H., "A Multiple Grid Scheme for Solving the Euler Equations," *AIAA Journal*, Vol. 20, No. 11, 1982, pp. 1565–1571.

[17]Deconinck, H., Roe, P. L., and Struijs, R., "A Multidimensional Generalization of Roe's Flux Difference Splitter for the Euler Equations," *Computers and Fluids Journal*, Vol. 22, 1993, pp. 215–222.

[18]Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors and Difference Schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–352.

[19]Jameson, A., "Analysis and Design of Numerical Schemes for Gas Dynamics 1: Artificial Diffusion, Upwind Biasing, Limiters and their Effect on Accuracy and Multigrid Convergence," *International Journal of Computational Fluid Dynamics*, Vol. 4, No. 3/4, 1995, pp. 171–218.

[20]Roe, P. L., "'Optimum' Upwind Advection on a Triangular Mesh," TR 90-75, Inst. for Computer Applications in Science and Engineering, Hampton, VA, 1990.

[21]Barth, T. J., "Aspects of Unstructured Grids and Finite Volume Solvers for the Euler and Navier–Stokes Equations," Rept. 787, AGARD-FDP-VKI Special Course on "Unstructured Grid Methods for Advection Dominated Flows," 1992, Chap. 6.

[22]Lanczos, C., "Solution of Systems of Linear Equations by Minimized Iterations," *Journal of Research of the National Bureau of Standards*, Vol. 49, 1952, pp. 33–53.

[23]Sonneveld, P., "CGS, a Fast Lanczos-Type Solver for Nonsymmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 10, 1989, pp. 36–52.

[24]Freund, R. W., and Nachtigal, N. M., "QMR: A Quasi-Minimal Residual Method for Non-Hermitian Linear Systems," *Numerical Mathematics*, Vol. 60, 1991, pp. 315–339.

[25]Freund, R. W., "A Transpose-Free Quasi-Minimal Residual Algorithm for Non-Hermitian Linear Systems," *SIAM Journal of Scientific and Statistical Computing*, Vol. 14, 1993, pp. 470–482.

[26]Issman, E., Degrez, G., and Deconinck, H., "Implicit Iterative Methods for a Multidimensional Upwind Euler/Navier–Stokes Solver on Unstructured Meshes," *Proceedings of the AIAA 12th Computational Fluid Dynamics Conference*, AIAA, Washington, DC, 1995 (AIAA Paper 95-1563).

[27]Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipes, the Art of Scientific Computing*, 2nd ed., Cambridge Univ. Press, Cambridge, England, UK, 1992, Chaps. 9 and 10.

[28]Brown, P. N., and Saad, Y., "Hybrid Krylov Methods for Non-Linear Systems of Equations," *SIAM Journal of Scientific and Statistical Computing*, Vol. 11, No. 3, 1990, pp. 450–481.

[29]Van Leer, B., Lee, W. T., and Roe, P. L., "Characteristic Time Stepping or Local Preconditioning of the Euler Equations," AIAA Paper 91-1552, 1991.

[30]Gautier, B. G., "Etude Théorique et Expérimentale des Effets du Refroidissement Pariétal sur l'Interaction onde de Choc—Couche Limite Laminaire en Écoulement Plan Hypersonique," Ph.D. Thesis, Faculté des Sciences Appliquées, Univ. Libre de Bruxelles, Belgium, 1972.