Aachen Institute for Advanced Study in Computational Engineering Science

Preprint: AICES-2009-14

06/July/2009

Convergence Acceleration for Simulation of Steady-State Compressible Flows Using High-Order Schemes

F. Iacono and G. May

RWTH AACHEN UNIVERSITY

AI CES

Financial support from the Deutsche Forschungsgemeinschaft (German Research Association) through grant GSC 111 is gratefully acknowledged.

List of AICES technical reports: http://www.aices.rwth-aachen.de/preprints

Convergence Acceleration for Simulation of Steady-State Compressible Flows Using High-Order Schemes

Francesca Iacono* and Georg May†

AICES, RWTH Aachen, Schinkelstr. 2, 52062 Aachen, Germany

The formulation of strategies for high-order discretization methods for compressible flow simulations is now to a certain extent understood, whereas the development of techniques for efficiently solving the resulting discrete equations has generally been lagging behind. Needs and constraints change greatly from case to case. In order to achieve the best-practice combination of all the ingredients composing the global strategy, target-specific tuning is required. We here take into consideration a Spectral Difference discretization for the Euler equations. We investigate convergence acceleration given by a full multigrid strategy implemented in conjunction with a hybrid multilevel relaxation technique. We also present the idea for a time-implicit relaxation technique, representing an intermediate solution between matrix-explicit and matrix-free techniques.

I. Introduction

Within the Computational Fluid Dynamics (CFD) community low-order methods are favored,[1–5] because of their robustness and reliability. Yet in the last years high-order (third and above) methods have aroused the general interest. In fact, discretization errors are recognized to be one of the most important sources of error in nowadays' simulations.[6] The expectation is that an efficient high-order discretization may achieve high accuracy in a flow showing a wide variety of length scales at reduced cost, by avoiding the use of excessive grid resolution. Obviously, because of the different asymptotic nature of these methods, the cost comparison between methods highly depends on the required levels of accuracy. Long-term aim is to make these methods competitive as alternative solvers to the Navier-Stokes equations on unstructured grids.

While the formulation of discretization strategies for high-order methods is now to an acceptable extent understood,[7–18] the development of techniques for efficiently solving the discrete equations arising from these methods has generally been falling behind. This is due in some measure to the structure of the discrete equations originating from complex discretization strategies. Their nonlinear nature further complicates the scenario, so that no unique solution strategy can be found to be the best for all circumstances. Memory storage constraints, available computational power, complexity both of the domain and of the flow physics, desired order of accuracy are just some of the variables determining from case to case which is the best-practice solution. The tuning necessary to set out best-practice strategies in each situation includes the observation of non-optimal behavior and the quest for improvements.

In this heterogeneous scenario convergence acceleration represents a big challenge. A promising strategy to accelerate convergence has been found to be the combination of multigrid methods with efficient relaxation techniques.[2,19–21] Many different possibilities fall under this same label. Nowadays, geometric multigrid methods - termed often as *h-multigrid* - are routinely used to accelerate the convergence of the Euler and Navier-Stokes equations to a steady state on unstructured grids. The *Full Approximation Storage* (FAS) algorithm introduced in Ref. 22 is suitable for general nonlinear problems, with the same relaxation and interpolation procedures used at all levels.

It is well established that h-multigrid acceleration can drastically reduce the computational costs.[3,23–25] In fact, in order to achieve steady solutions for nonlinear hyperbolic equations, transient error modes can

*Ph.D. candidate, AIAA Student Member

†Junior Research Group Leader, AIAA Member

be eliminated mainly through two mechanisms: by damping, and by expulsion from the computational domain.[1,3] h-Multigrid contributes to both of these aspects. As regards damping, low-frequency errors in the solution are transferred on coarser meshes, where they become high frequency errors, that are more effectively smoothed by traditional relaxation methods. In addition to this, a coarse mesh propagates the error modes faster, accelerating in return the phenomen of expulsion as well. The combination of h-multigrid methods with traditional relaxation techniques is mature for first- and second-order schemes. Grid-independent convergence has been proved for elliptic operators.[22,26] On the one hand, such a rigorous theory does not exist for problems involving a hyperbolic component.[27,28] On the other hand, experience has shown that methods based on nonlinear h-multigrid are extremely effective for the Euler equations.[1,29,30]

Nonlinear *p-Multigrid* is a natural extension of h-multigrid methods to high-order finite element formulations, where systems of equations are solved by recursively iterating on solution approximations of different polynomial order. Even though p-multigrid shows some good properties, it is difficult to see how the critical convective convergence mode could be accelerated by the use of p-multigrid without using h-multigrid as well.[31] Furthermore, the workload decreases only marginally for p-coarsening, which contributes to our view that p-multigrid is not necessarily very efficient.

It is possible to use multigrid with different relaxation schemes on different mesh levels and with different levels of approximation. The approaches proposed in Ref. 32 and Ref. 33 exploit this concept in two complementary manners. Luo et al.[32] developed a p-multigrid method to solve the compressible Euler equations on unstructured grids, using explicit relaxation on the highest level of approximation and implicit time relaxation on lower levels. In this way, they try to reduce the storage requirements and to gain advantage from the mature matrix-free implicit methods derived from the finite volume context. May et al.[33] started in part from similar considerations, ending up with an opposite approach. On the one hand, explicit schemes are considered ineffective for high orders of accuracy. An example of this can be found in Ref. 34. On the other hand, as already pointed out, p-multigrid does not pay much contribution to error convection and expulsion. Moreover, integrated quantities, which are often the desired output in CFD computations, are often converged even at rather high levels of the residuals, and when high-frequency errors still persist. Thus a so-called *Hybrid Multilevel* approach,[33] using implicit relaxation on the highest level of approximation and explicit multigrid on the lower levels, was formulated.

Another aspect that we have considered in this work is the large storage often needed by implicit relaxation techniques with explicit storage of the system matrix. In particular, we attempt the design of a technique, based on partial matrix-storage, which offers a compromise between the large storage needed by matrix-explicit techniques and the high computational cost of matrix-free techniques. It will be revealed that a weak point in this strategy is the lack of a suitable preconditioning, able to exploit the information stored about the global matrix of the system. This outcome strengthens our vision that one of the questions to be next addressed is the quest for suitable preconditioners.

The remainder of this paper is structured in the following way. In the next section, we briefly recall the Euler equations, describing the dynamics of compressible inviscid fluids. Section III presents the spatial semidiscretization procedure, based on the Spectral Difference method. After that, a range of possible relaxation techniques for the resulting system of semidiscretized ordinary differential equations is described in section IV. Therein, multigrid combination with relaxation techniques as well as the new time-implicit approach, intermediate between the matrix-explicit and the matrix-free strategies, are presented. The results of numerical experiments over a smooth bump and around a NACA0012 airfoil are presented in section V, and finally conclusions are drawn in section VI.

II. Euler Equations in Conservation Form

Let Θ be an open subset of $\mathbb{R}^p$, and let $\mathbf{f}_l$, $1 \leq l \leq d$, be d smooth functions from Θ into $\mathbb{R}^p$; the general form of a system of conservation laws in d space variables is

$$\frac{\partial \mathbf{u}}{\partial t} + \sum_{l=1}^{d} \frac{\partial}{\partial x_l} \mathbf{f}_l(\mathbf{u}) = \mathbf{0}, \tag{1}$$

where $\mathbf{x} = (x_1, \ldots, x_d)^T \in \mathbb{R}^d$ is the space variable, $t > 0$ is the time variable, and $\mathbf{u} = (u_1, \ldots, u_p)^T$ is a vector-valued function from $\mathbb{R}^d \times [0, +\infty)$ into Θ. The system (1) is written in divergence form and the functions $\mathbf{f}_l = (f_{l,1}, \ldots, f_{l,p})^T$ are the flux functions.

The *Euler equations* for a compressible inviscid fluid can be written in divergence form as in Eq. (1) with

$$\mathbf{u} = \begin{pmatrix} \rho \\ q_1 \\ q_2 \\ E \end{pmatrix}, \qquad \mathbf{f}_1(\mathbf{u}) = \begin{pmatrix} q_1 \\ p + q_1^2/\rho \\ q_1 q_2/\rho \\ (E+p)q_1/\rho \end{pmatrix}, \qquad \mathbf{f}_2(\mathbf{u}) = \begin{pmatrix} q_2 \\ q_1 q_2/\rho \\ p + q_2^2/\rho \\ (E+p)q_2/\rho \end{pmatrix}, \tag{2}$$

where we took $d = 2$ (hence $p = 4$). In Eq. (2), ρ is the density of the fluid, $q_l = \rho u_l$, $l = 1, 2$, are the components of the momentum, $\mathbf{v} = (u_1, u_2)^T$ the velocity, $E = \rho e$ the total energy, $e = \varepsilon + \dfrac{|\mathbf{v}|^2}{2}$ the specific total energy, ε the specific internal energy, and p the pressure. Thus the equations express the laws of conservation of mass, momentum, and total energy for the fluid. As regards the pressure p we do consider the equation of state for a polytropic ideal gas, given by

$$p = (\gamma - 1)\rho\varepsilon, \tag{3}$$

where $\gamma = 1.4$ is the ratio of specific heats for air.

III. Spatial Semidiscretization

We separate the space discretization procedure from the time-marching procedure. This allows problems of spatial discretization error and artificial dissipation to be studied independently of the problems of time-marching stability and convergence acceleration.

Much of the research effort is nowadays applied to the development of numerical methods for conservation laws with the following properties: locally conservative, high-order accurate in smooth regions of the solution, showing sharp and monotone shock transitions, geometrically flexible, computationally efficient, and simply formulated. A rough classification of the methods in use can be seen in figure 1.

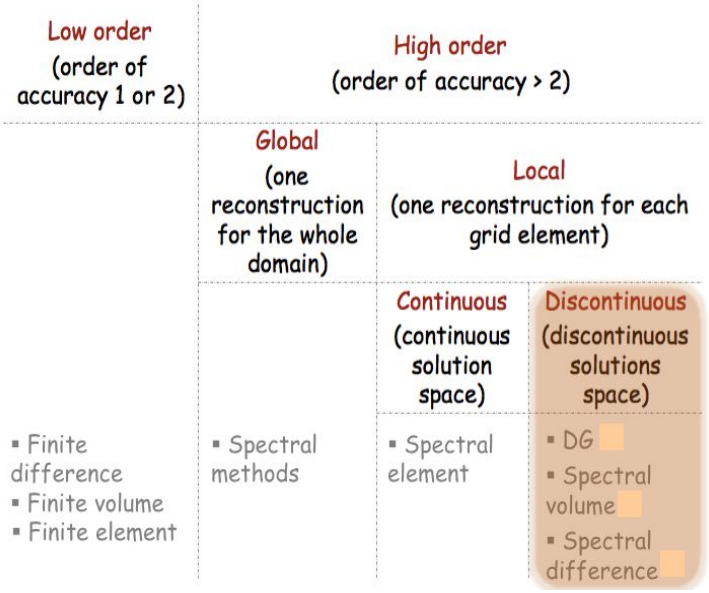


Figure 1. Classification of spatial semidiscretization methods.

Here, however, we focus our attention on high-order local discontinuous semidiscretization methods. The *Spectral Difference* (SD) method for conservation laws on unstructured grids utilizes the concept of discontinuous and high-order local representations to achieve conservation and high accuracy in a manner similar to the Discontinuous Galerkin[7–12] (DG) and Spectral Volume[13–16] (SV) methods, but is based on the finite difference formulation. The SD method has been recently developed by Liu et al.[17,18] and Wang et al.[35] As a matter of fact, the DG, SV, and SD methods are similar in that they share the same solution space, i.e., the space of piecewise-continuous polynomials, and that Riemann solvers are used at the element interfaces to provide solution coupling between the discontinuous elements and appropriate numerical

dissipation necessary to achieve stability. In addition, all of these approaches are locally conservative at the element level, making them suitable for problems with discontinuities. They differ on how the degrees of freedom (DOFs) are chosen, and how they are updated. May[36] showed that the SD method can be obtained under certain assumptions as a quadrature-free DG scheme. The SD method has an important advantage over the DG and SV methods, namely that no integrals have to be evaluated to compute the residuals. Consequently, costly high-order accurate quadrature formulas are avoided.

III.A. Spectral Difference Semidiscretization

Let us consider Eq. (1) and apply the Spectral Difference semidiscretization. Consider a triangulation

$$\mathcal{T}_h = \{T^{(i)} : 1 \leq i \leq N_e\} \text{ on } \Omega \in \mathbb{R}^2 \text{ such that } \overline{\Omega}_h \subseteq \bigcup_{i=1}^{N_e} \overline{T^{(i)}}, \tag{4}$$

N_e being the number of elements in the triangulation. Assume that there exist mappings

$$\Phi^{(i)} : \hat{T} \to T^{(i)}, \hat{\mathbf{x}} \mapsto \mathbf{x}, \quad 1 \leq i \leq N_e, \tag{5}$$

with nonsingular Jacobian $J^{(i)} = D_{\hat{x}}\Phi^{(i)}$, such that each element in the triangulation can be mapped to a reference domain $\hat{T}$ and there is a univocal correspondence between the points $\hat{\mathbf{x}}$ in the reference domain and the points $\mathbf{x}$ in one of the elements $T^{(i)}$ of the triangulation, for all $T^{(i)} \in \mathcal{T}_h$. For simplicity we consider straight-sided triangular elements, although such a restriction is not necessary. (The implementation used in our solver allows for elements with curved edges.)

Let $m+1$ be the desired order of accuracy. For the Spectral Difference scheme we seek solutions $\mathbf{u}_h = (u_{h,1}, \ldots, u_{h,4})^T$, such that for each triangle $\mathbf{u}_h|_{T^{(i)}} \circ \Phi^{(i)} \in \left[\mathcal{P}^m(\hat{T})\right]^4$, where $\mathcal{P}^m(\hat{T}) = \text{span}\left\{\hat{\mathbf{x}}^{\vec{\alpha}} : \hat{\mathbf{x}} \in \hat{T}, \alpha_i \geq 0, |\vec{\alpha}| \leq m\right\}$ is the space of polynomials of total degree m defined on the reference element. Let us introduce N_m being the dimension of the space of polynomials:

$$N_m := \begin{pmatrix} m \\ 2 \end{pmatrix} = \frac{(m+1)(m+2)}{2}. \tag{6}$$

As a basis for $\mathcal{P}^m(\hat{T})$ we consider the multivariate Lagrange interpolation functions L_j, $1 \leq j \leq N_m$, on the reference domain, corresponding to a nodal set $\mathcal{S}_m = \{\hat{\mathbf{x}}_j, 1 \leq j \leq N_m\}$, i.e.,

$$\mathbf{u}_h|_{T^{(i)}} := \sum_{j=1}^{N_m} \mathbf{u}_j^{(i)} L_j(\hat{\mathbf{x}}), \tag{7}$$

where $\mathbf{u}_j^{(i)} = \mathbf{u}(\mathbf{x}_j^{(i)})$, with $\mathbf{x}_j^{(i)} = \Phi^{(i)}(\hat{\mathbf{x}}_j)$ being the j-th solution node in the element $T^{(i)}$.

Furthermore, the nonlinear fluxes in Eq. (1) are projected onto a similar space, such that $(\mathbf{f}_h)_l|_{T^{(i)}} \circ \Phi^{(i)} \in \left[\mathcal{P}^{m+1}(\hat{T})\right]^4$ using a nodal set $\mathcal{Q}_{m+1} = \{\hat{\mathbf{x}}_k, 1 \leq k \leq N_{m+1}\}$ with $\mathring{L}_k$ being the corresponding Lagrange interpolating polynomials:

$$\mathbf{f}_h|_{T^{(i)}} := \sum_{k=1}^{N_{m+1}} \tilde{\mathbf{f}}_k^{(i)} \mathring{L}_k(\hat{\mathbf{x}}). \tag{8}$$

The degrees of freedom are computed for straight-sided elements as

$$\tilde{\mathbf{f}}_k^{(i)} = \begin{cases} (J^{(i)})^{-1}\mathbf{f}\left(\mathbf{u}_k^{(i)}\right), & \hat{\mathbf{x}}_k \in \hat{T}, \\ (J^{(i)})^{-1}\mathbf{f}^{num}, & \hat{\mathbf{x}}_k \in \partial\hat{T}, \end{cases} \tag{9}$$

where $\partial\hat{T}$ is the boundary of the element $\hat{T}$. The coefficients $\mathbf{f}^{num}$ are chosen such that $\mathbf{f}^{num} \cdot \mathbf{n} = \mathbf{h}$, where $\mathbf{h}$ is a standard numerical flux function for the normal component of the flux, and $\mathbf{n}$ is the normal vector to the edge of the element. In our implementation the numerical flux function $\mathbf{h}$ is given by Jameson's CUSP flux,[37] which by definition ensures conservativity of the scheme. The tangential component of the flux can be freely chosen, for instance by averaging the two contributions coming from the two neighbor elements. In

each element $T^{(i)}$, $1 \leq i \leq N_e$, of the triangulation, for each solution node $\mathbf{x}_j^{(i)}$, $1 \leq j \leq N_m$, we obtain the following scheme:

$$\frac{d\mathbf{u}_j^{(i)}}{dt} = -\sum_{k=1}^{N_{m+1}} \nabla_{\hat{\mathbf{x}}} \mathring{L}_k(\hat{\mathbf{x}}_j) \cdot \tilde{\mathbf{f}}_k^{(i)}. \tag{10}$$

We use Hesthaven's interpolation nodes[38] for the nodal set $\mathcal{Q}_{m+1}$. The scheme as described above is independent of the choice of solution nodes,[39] due to the fact, that the differentiated polynomial approximation of the flux function is by construction a polynomial of degree m, and hence represented exactly on any unisolvent set of interpolation nodes.

IV. Relaxation Techniques

After the spatial semi-discretization we can write the system of ordinary differential equations as

$$\frac{d\mathbf{U}}{dt} + \mathbf{R}(\mathbf{U}) = 0 \ , \tag{11}$$

where $\mathbf{U} := \left(\mathbf{u}_1^{(1)}, \ldots, \mathbf{u}_{N_m}^{(N_e)}\right)^T$ is the global vector of all the degrees of freedom, and $\mathbf{R}$ is the nonlinear residual vector of the spatial semidiscretization terms, corresponding to the right-hand side of Eq. (10). Since we consider stationary problems we are actually interested in solving

$$\mathbf{R}(\mathbf{U}) = \mathbf{0}. \tag{12}$$

Despite this, steady solutions to the Euler and Navier-Stokes equations are often obtained by marching the unsteady systems in time until the time-derivative terms have become sufficiently small to ensure the desired degree of steadiness in the solution. Since the objective is simply to reach the steady state and details of the transient solution are irrelevant, the time stepping scheme can be designed solely to maximize the rate of convergence. Hence we solve Eq. (11) by considering t a pseudo-time variable and by marching the field equations to a steady state.

IV.A. Explicit Relaxation Techniques

For the explicit temporal discretization a *Runge-Kutta multistage* technique can be used. We choose in particular *Shu's three-stage* (Shu RK3) scheme:[12]

$$\begin{aligned}
&\mathbf{U}^{(0)} = \mathbf{U}^n, \\
&\mathbf{U}^{(k)} = \sum_{l=1}^{k} \left\{ \alpha_{k,l} \mathbf{U}^{(l-1)} - \Delta\tau \, \beta_{k,l} \mathbf{R}^{(l-1)} \right\}, \quad 1 \leq k \leq 3, \\
&\mathbf{U}^{n+1} = \mathbf{U}^{(3)},
\end{aligned} \tag{13}$$

where the superscript n refers to the external time iterations, the superscript (k) refers to the internal stages of the scheme, and $\Delta\tau$ is the time step. The coefficients arranged in matrix form are:

$$\alpha = \begin{pmatrix} 1 & 0 & 0 \\ \frac{3}{4} & \frac{1}{4} & 0 \\ \frac{1}{3} & 0 & \frac{2}{3} \end{pmatrix}, \qquad \beta = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{4} & 0 \\ 0 & 0 & \frac{2}{3} \end{pmatrix}. \tag{14}$$

We choose this scheme because it preserves the total variation diminishing (TVD) properties of the spatial operator, properties that for the Spectral Difference method with explicit time stepping have been shown[34] using standard limiting methods.

IV.A.1. Combining Explicit Relaxation with Multigrid

For the special case $u_h \in \mathcal{P}^0$, for which the Spectral Difference scheme reduces to a Finite Volume method, one may, following Jameson,[1] use *geometric multigrid* techniques combined with explicit Runge-Kutta methods, under the paradigm of the general FAS methodology.[22]

Assume that the equations have been iterated n steps on a given mesh of characteristic length h - the 'fine' mesh - resulting in an approximation $\mathbf{U}_h^n$, and residual $\mathbf{R}_h(\mathbf{U}_h^n)$. Using a suitable coarser mesh of characteristic length H, and defining appropriate transfer operators I_h^H and $\tilde{I}_h^H$ for the solution and residual respectively, we can restrict the solution and the residual to the coarser mesh:

$$\mathbf{U}_H^0 = I_h^H \mathbf{U}_h^n, \qquad \mathbf{R}_H^0 = \tilde{I}_h^H \mathbf{R}_h^n, \tag{15}$$

where the subscripts h and H to the solution (respectively, residual) remind us on which grid that solution (respectively, residual) lives. The solution can be advanced by one step on a coarse grid by the modified multistage scheme:

$$\begin{aligned} &\mathbf{U}_H^{(0)} = \mathbf{U}_H^0, \\ &\mathbf{U}_H^{(k)} = \sum_{l=1}^{k} \left\{ \alpha_{k,l} \mathbf{U}_H^{(l-1)} + \Delta\tau\, \beta_{k,l} \left[\mathbf{R}_H^{(l-1)} + \mathbf{S}_H \right] \right\}, \qquad 1 \le k \le 3, \\ &\mathbf{U}_H^{+} = \mathbf{U}_H^{(3)}, \end{aligned} \tag{16}$$

where the superscript + indicates the solution updated after the application of the multistage scheme, and

$$\mathbf{S}_H = \tilde{I}_h^H \mathbf{R}_h^n - \mathbf{R}_H^{(0)}. \tag{17}$$

This is formally identical to the original equation, except for an additional source $\mathbf{S}_H$, which allows the same code to be used on the coarse and fine mesh. (See Ref. 1 for details.) After relaxing on the coarse mesh for 3 iterations, the corrected solution on the fine grid is computed as:

$$\mathbf{U}_h^+ = \mathbf{U}_h^n + I_H^h (\mathbf{U}_H^+ - \mathbf{U}_H^0), \tag{18}$$

where I_H^h is usually a polynomial interpolation of a suitable order. Recursive application of this concept allows to extend the method to more than two meshes. In particular, we use W-cycles, i.e., at each level, the coarse grid correction scheme is performed recursively two times.

The combination of multigrid and explicit relaxation techniques has substantially met industrial requirements for inviscid flow calculations. Known shortcomings come from stiffness, relevant only at stagnation points, at shocks and across sonic lines, and directional decoupling, taking place when the flow aligns itself with the computational mesh. These shortcomings become more evident when the approach is applied to viscous problems. In this case also high-aspect-ratio cells are needed inside the boundary layer, hence the stiffness of the system is increased by several orders of magnitude. Moreover, explicit relaxation techniques are conditionally stable[34,39,40] and become more and more ineffective for growing order of accuracy. An example of this can be found in Ref. 34, where the linear stability tables for the combination of Runge-Kutta time stepping together with Spectral Difference and DG discretization applied to the one-dimensional linear advection equation are listed. There, the stability limit decays as m^{-2} for growing order of accuracy m. As a result, one ought to consider implicit relaxation methods as well.

IV.B. Implicit Relaxation Techniques

We can write a *backward Euler discretization* of Eq. (11) as

$$\left[I + \Delta\tau \left(D_{\mathbf{U}} \mathbf{R} \right) \big|_{\mathbf{U}^n} \right] \Delta\mathbf{U}^n = -\Delta\tau\, \mathbf{R}(\mathbf{U}^n), \tag{19}$$

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta\mathbf{U}^n, \tag{20}$$

for $n = 0, 1, , \ldots$, where I is the identity matrix, and $\Delta\tau$ the time step. For $\Delta\tau \to \infty$ one obtains a *Newton* iteration, while finite time steps may be interpreted as *damped Newton* iterations, or in the framework of *pseudo-transient continuation methods*.[41]

Since we consider steady-state calculation, the only limitation on the time step in implicit methods - time accuracy - is removed, allowing us to take extremely large time steps w.r.t. most explicit methods. We hence can set local time steps, how described in Ref. 33. For a fairly large part of transient phase of the solution process, finite CFL numbers have to be used. In particular, at start-up a simple ramping method is introduced to reach a nominal CFL number.[33]

Two main families of implicit relaxation techniques can be individuated, depending on the decision to store the matrix $\mathbb{M}^n := I + \Delta\tau \left(D_{\mathbf{U}}\mathbf{R}\right)\big|_{\mathbf{U}^n}$ or not. The methods are then respectively called matrix-explicit or matrix-free. Both the choices lead to a nested iteration method, consisting mainly of an external loop over the nonlinear Newton corrections and an internal loop computing each Newton correction as solution of the linear system in Eq. (20).

As regards the matrix-free approach, we choose the framework of the so-called *Jacobian-Free Newton-Krylov* (JFNK) methods.[41] In agreement with Ref. 41, we choose (restarted) GMRES[42] as specific Krylov method for the solution of the linear system in Eq. (20). Basing our choice on its favorable convergence properties, we stick to this choice for the matrix-explicit implementation as well.

Implementing the time-implicit relaxation techniques includes tasks such as derivation of the exact Jacobian $\mathbb{J}^n := \left(D_{\mathbf{U}}\mathbf{R}\right)\big|_{\mathbf{U}^n}$, and implementation of different solution methods, including preconditioning techniques. In particular, for the matrix-free methods user-defined routines implementing the matrix-vector product $\mathbb{M}\mathbf{v}$ have to be provided.

IV.B.1. Matrix-Explicit Methods

In the following we will omit the superscript denoting the time iteration. Each 4×4 block of the global matrix can be assembled as

$$\mathbb{M}_{j,q}^{(i,r)} := I_{4\times 4} + \Delta\tau\, \mathbb{J}_{j,q}^{(i,r)}, \tag{21}$$

with

$$\mathbb{J}_{j,q}^{(i,r)} := \frac{\partial \mathbf{R}_j^{(i)}}{\partial \mathbf{u}_q^{(r)}} = \begin{cases} \displaystyle\sum_{l=1}^{2}\sum_{k=1}^{N_{m+1}} d_{j,k,l}\, \mathbb{A}_{k,l}^{(i)}\, l_{k,q}, & \text{if } r = i, \\ \displaystyle\sum_{l=1}^{2}\sum_{k=1}^{N_{m+1}} d_{j,k,l}\, \mathbb{B}_{k,l}^{(i,r)}\, l_{k,q}, & \text{if } T^{(r)} \text{ is a neighbor of } T^{(i)}, \\ \mathbf{0}, & \text{otherwise,} \end{cases} \tag{22}$$

for $1 \le i, r \le N_e$ and $1 \le j, q \le N_m$. $d_{j,k,l}$ is the local differentiation scalar coefficient $d_{j,k,l} = \left.\dfrac{\partial \mathring{L}_k}{\partial \hat{x}_l}\right|_{\hat{\mathbf{x}}_j}$, and the reconstruction coefficients $l_{k,q}$ are given by $l_{k,q} = L_q(\hat{\mathbf{x}}_k)$. The assembly of the Jacobian is relatively straightforward, since the reconstruction and differentiation matrices are available for the residual computation. The 4×4 matrices $\mathbb{A}_{k,l}^{(i)}$ and $\mathbb{B}_{k,l}^{(i,r)}$ are given by the flux Jacobian evaluated at the flux collocation points for interior nodes $\mathbf{x}_k \in T^{(i)}$, and the differentiation of the numerical flux function for nodes $\mathbf{x}_k \in \partial T^{(i)}$:

$$\mathbb{A}_{k,l}^{(i)} := \frac{\partial \tilde{\mathbf{f}}_{k,l}^{(i)}}{\partial \mathbf{u}_k^{(i)}}, \qquad 1 \le l \le d, \quad 1 \le i \le N_e, \tag{23}$$

$$\mathbb{B}_{k,l}^{(i,r)} := \frac{\partial \tilde{\mathbf{f}}_{k,l}^{(i)}}{\partial \mathbf{u}_k^{(r)}}, \qquad 1 \le l \le d, \quad 1 \le i, r \le N_e \text{ with } r \ne i. \tag{24}$$

In order to give an example of the way the global matrix $\mathbb{M}$ looks like, we consider the possible structure

of a matrix coming from a two-dimensional problem:

$$\mathbb{M} = I_{N_e \times N_m \times 4} + \Delta\tau \overbrace{\left(\begin{array}{cccccccccc} \blacksquare & \square & \blacksquare & \square & \blacksquare & \square & \cdots & \square & \square & \square \\ \square & \blacksquare & \blacksquare & \square & \square & \square & \cdots & \blacksquare & \square & \square \\ \square & \square & \blacksquare & \square & \blacksquare & \square & \cdots & \square & \square & \square \\ \square & \square & \square & \blacksquare & \square & \square & \cdots & \square & \square & \square \\ \blacksquare & \square & \square & \square & \blacksquare & \square & \cdots & \square & \square & \blacksquare \\ \blacksquare & \square & \blacksquare & \square & \square & \blacksquare & \cdots & \square & \square & \square \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \square & \square & \blacksquare & \square & \blacksquare & \square & \cdots & \blacksquare & \square & \square \\ \blacksquare & \square & \square & \square & \square & \square & \cdots & \square & \blacksquare & \square \\ \square & \square & \blacksquare & \square & \square & \square & \cdots & \blacksquare & \square & \blacksquare \end{array} \right)}^{N_e} \Bigg\} N_e \tag{25}$$

$\square$ represents a null block and $\blacksquare$ a non-null one. Each block $\blacksquare$ can be 'zoomed' as:

$$\blacksquare^{(i,r)} = \overbrace{\left(\begin{array}{cccc} \frac{\partial \mathbf{R}_1^{(i)}}{\partial \mathbf{u}_1^{(r)}} & \frac{\partial \mathbf{R}_1^{(i)}}{\partial \mathbf{u}_2^{(r)}} & \cdots & \frac{\partial \mathbf{R}_1^{(i)}}{\partial \mathbf{u}_{N_m}^{(r)}} \\ \frac{\partial \mathbf{R}_2^{(i)}}{\partial \mathbf{u}_1^{(r)}} & \frac{\partial \mathbf{R}_2^{(i)}}{\partial \mathbf{u}_2^{(r)}} & \cdots & \frac{\partial \mathbf{R}_2^{(i)}}{\partial \mathbf{u}_{N_m}^{(r)}} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial \mathbf{R}_{N_m}^{(i)}}{\partial \mathbf{u}_1^{(r)}} & \frac{\partial \mathbf{R}_{N_m}^{(i)}}{\partial \mathbf{u}_2^{(r)}} & \cdots & \frac{\partial \mathbf{R}_{N_m}^{(i)}}{\partial \mathbf{u}_{N_m}^{(r)}} \end{array} \right)}^{N_m} \Bigg\} N_m \qquad 1 \leq i, r \leq N_e. \tag{26}$$

In turn, each 4×4 subblock $\dfrac{\partial \mathbf{R}_j^{(i)}}{\partial \mathbf{u}_q^{(r)}}$ is given by the expression in Eq. (22).

In the matrix-explicit approach we store all the non-null blocks of the matrix $\mathbb{M}$, i.e., we store (abstracting from the two-dimensional case)

$$(d+2)^2 \times N_m^2 \times N_e \times (1 + N_n) \tag{27}$$

entries of the matrix, where N_n is the number of neighbors of each element, and the number of solution nodes in each element behaves as

$$N_m = \left(\begin{array}{c} m \\ d \end{array} \right) := \frac{\prod_{l=1}^{d} (m+l)}{d!} \sim m^d. \tag{28}$$

By neighbors we refer to the adjacent elements. $N_n = N_n(i)$ depends on the particular grid and is different for each element $T^{(i)}$. In particular, let us assume that no hanging nodes (respectively, edges) are present in the grid in the two-dimensional (respectively, three-dimensional) case. Thus, we can estimate $0 \leq N_n \leq d+1$ and globally the number of non-null blocks to be stored can be upper-bounded by

$$(d+2)^3 \times N_m^2 \times N_e. \tag{29}$$

This often translates into a (too) large storage requirement, growing as m^4 in two dimensions and as m^6 in three dimensions. This is the reason why, even though the matrix-explicit is computationally fairly efficient, matrix-free techniques have been introduced.

IV.B.2. Matrix-Free Methods

In our implementation a subroutine implementing the matrix-vector product $\mathbb{M}\mathbf{v}$ has to be provided. A first-order Taylor series expansion can be used to approximate the projection of the global matrix onto the Krylov vector $\mathbf{v}$:[33]

$$\mathbb{M}\mathbf{v} \approx \frac{\mathbf{R}(\mathbf{U}+\varepsilon\mathbf{v}) - \mathbf{R}(\mathbf{U})}{\varepsilon}, \tag{30}$$

where ε is the perturbation parameter. Among the different choices summed up in Ref. 41 for the choice of ε, we consider

$$\varepsilon = \sqrt{1 + \|\mathbf{U}\|}\,\varepsilon_{rel}, \tag{31}$$

with $\varepsilon_{rel} = 10^{-6}$.

IV.B.3. Partial-Storage Approach

Starting from the remarks of subsection IV.B.1, we built and implemented a technique, here called *partial-storage approach*, which is an intermediate solution between matrix-explicit and matrix-free techniques. The background idea is storing the building blocks $\mathbb{A}_{k,l}^{(i)}$ and $\mathbb{B}_{k,l}^{(i,r)}$ and assembling the elements of the global matrix $\mathbb{M}$ just in the matrix-vector product routine needed by GMRES in order to build the Krylov vectors. The expected advantages of this intermediate approach were a lighter storage requirement w.r.t. the matrix-explicit approach of section IV.B.1 and a faster convergence in terms of number of iterations needed to converge to the desired accuracy w.r.t. the matrix-free approach of section IV.B.2. The latter would be justified by having available the exact matrix instead of an approximation.

We refer to the $\mathbb{A}_{k,l}^{(i)}$ and $\mathbb{B}_{k,l}^{(i,r)}$ as 'building' blocks, since actually they are not direct component blocks of the final global matrix $\mathbb{M}$, but they are used while building its entries in the way described by Eq. (21) and Eq. (22). If we choose to freeze the matrix for some iterations, say N_f, this means that we compute the partial blocks every N_f iterations and at each iteration we only assemble the blocks of the global matrix $\mathbb{M}$ starting from the partial $\mathbb{A}_{k,l}^{(i)}$ and $\mathbb{B}_{k,l}^{(i,r)}$.

On the one hand, the advantage in spared memory is real: we just need to store

$$d \times (d+2)^2 \times N_{m+1} \times N_e \tag{32}$$

entries, where, analogously to Eq. (28), the number of flux nodes in each element is $N_{m+1} \sim m^d$. Thanks to this approach we can store less floating points, where 'less' means sparing, by a rough estimate, by a factor of up to m^d. This means that the higher the accuracy of the chosen method, the higher the relative spared memory w.r.t. the matrix-explicit approach. On the other hand, this will translate into an augmented computational time, necessary for the repeated assembly of the matrix entries.

IV.B.4. Preconditioning

Matrix-explicit methods using GMRES to solve the linear systems are preconditioned by incomplete LU factorization,[43] denoted $ILU(p)$, where p stands here for the level of additional fill allowed in the incomplete factorization. As regards the matrix-free method, we precondition it, in the framework of the *flexible GMRES*,[44] by using the principle of 'squared preconditioning' introduced in Ref. 33.

As regards the partial-storage approach, the preconditioner will happen to be the key point. In fact, we here chose to adopt the 'squared preconditioning' as well. However, this solution does not derive any benefit from the fact that the exact global matrix can be rebuilt starting from the building subblocks that are stored.

IV.C. Hybrid Multilevel Relaxation Techniques

In the introduction to this paper we have collected some remarks motivating the formulation of the hybrid multilevel approach,[33] where 'hybrid' expresses the combination of implicit and explicit relaxation techniques. In light of the results shown in Ref. 33, we choose this method to be the kernel of our solver.

Here damped Newton iterations combined with GMRES linear solves are used on the highest level of approximation, i.e., the level corresponding to the finest grid and the highest order of accuracy. On lower levels of approximation Runge-Kutta relaxation is combined with geometric h-multigrid between two

consecutive Newton iterations, using polynomials of degree $m = 1$ or $m = 0$. Typically 10 or more W-multigrid cycles are performed between two Newton iterations. Figure 2 illustrates schematically the hybrid approach.

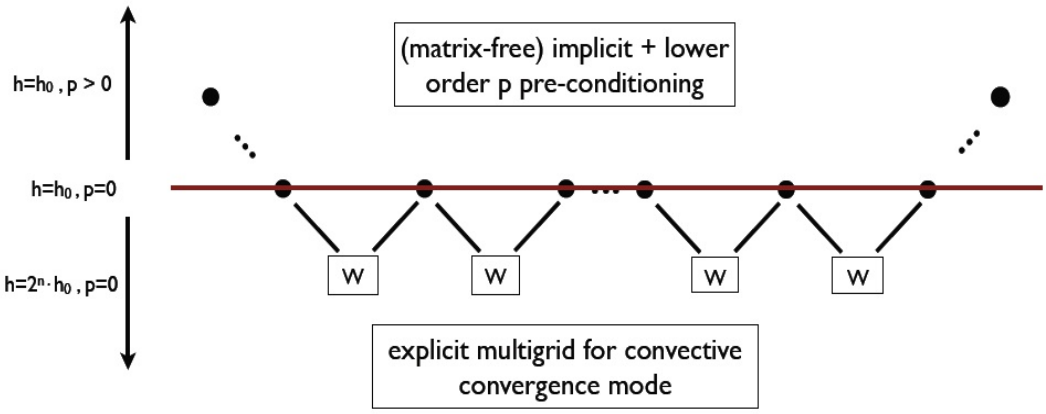


Figure 2. Diagram describing the hybrid multilevel relaxation strategy. The highest (respectively, lowest) level of approximation is on the top (respectively, bottom) of the diagram. The red horizontal line marks the border between geometric h-multigrid (at $p = 0$) and $p > 0$. Computational time can be imagined flowing from the left to the right of the diagram. The blocks marked by the letter W indicate the multiple time-explicit W-multigrid cycles occurring between two implicit damped Newton iterations, represented in the diagram by two black dots in correspondence of the highest level of approximation. Here $p = m$ is the order of the polynomials used in the spatial discretization.

IV.C.1. Full Multigrid

Geometric h-multigrid techniques start with relaxations on the finest grid, then continue on coarser grids to reduce low-frequency error components. In the so-called *Full MultiGrid* (FMG) scheme the whole computation starts on the coarsest grid and then proceeds up till to the finest grid only when sufficiently good approximations to the solution have been achieved. This is repeated till the finest grid is reached, and the obtained approximation is used as initial guess for subsequent multigrid iterations. The background idea is limiting the necessary computational cost: the finer the mesh, the more iterations are needed in order to make full use of the accuracy of the discretization.

We integrated the FMG strategy in the implementation of the hybrid multilevel approach. Figure 3 shows the working principle of FMG. In particular, in our implementation we use W-cycles.

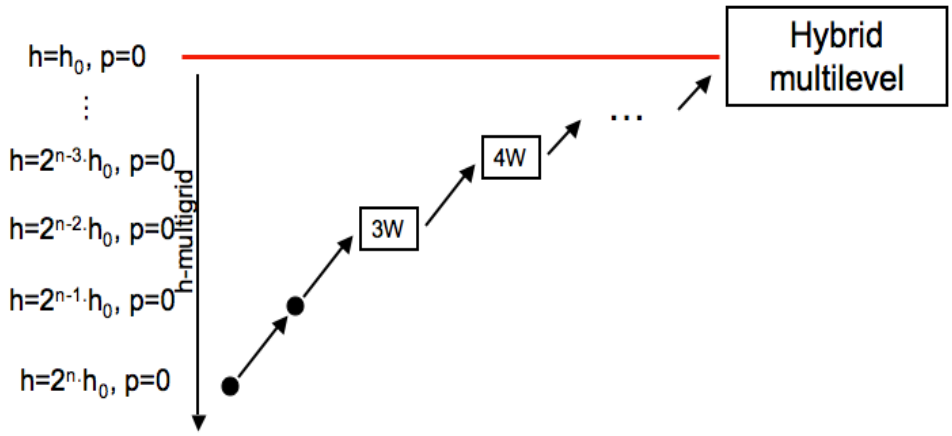


Figure 3. Illustration of full multigrid strategy with W-cycles. The highest (respectively, lowest) level of approximation is on the top (respectively, bottom) of the diagram. The red horizontal line marks the border between geometric h-multigrid (at $p = 0$) and $p > 0$. Computational time can be imagined flowing from the left to the right of the diagram. 'nW' stays for geometric multigrid with W-cycle explicit Runge-Kutta time stepping on n levels. Here $p = m$ is the order of the polynomials used in the spatial discretization.

V. Numerical Results

V.A. Adjusting Parameters of the Solution Algorithm

V.A.1. Solution Accuracy for the Linear System

In the implementation of the time-implicit relaxation we let the library PETSc[45] drive the GMRES algorithm. The default convergence test in PETSc is based on the l_2-norm of the residual. We focused our attention on this aspect after having observed that often too many GMRES iterations are performed without obtaining a real improvement in the global convergence. Hence, we investigate the behavior of the convergence rate w.r.t. the convergence tolerance *rtol* for the decrease of the residual norm relative to the norm of the right-hand side. If we refer to Eq. (19), we have convergence at n-th iteration[a] if

$$\frac{\|\Delta\tau\,\mathbf{R}(\mathbf{U}^n) + \mathbb{M}^n\Delta\mathbf{U}^n\|_2}{\|\Delta\tau\,\mathbf{R}(\mathbf{U}^n)\|_2} \leq rtol. \tag{33}$$

Let us consider inviscid flow over a smooth bump at free-stream Mach number $M_\infty = 0.3$. Computations are performed using the Spectral Difference scheme on a regular triangular mesh consisting of 342 elements. Figure 4 shows computational domain, mesh and contours of constant Mach number. We use the matrix-

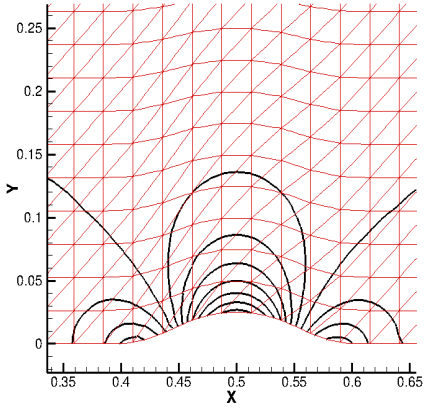


Figure 4. Inviscid flow over a smooth bump at free-stream Mach number $M_\infty = 0.3$. Computational mesh (342 elements) and contours of constant Mach number. Cubic polynomials have been used for the reconstruction of the Mach number.

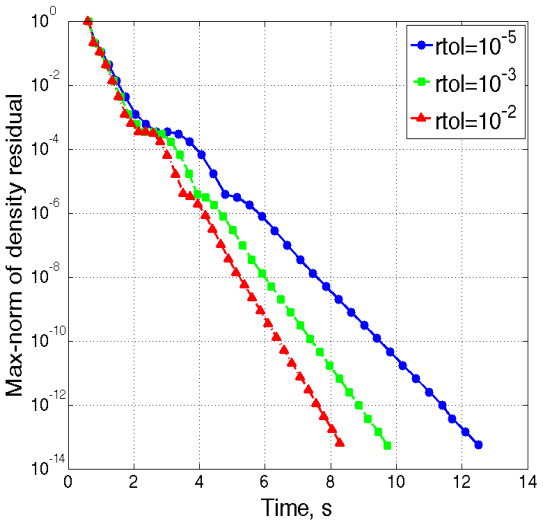
explicit approach of section IV.B.1: the global matrix is frozen for $N_f = 3$ Newton iterations and GMRES(30) is preconditioned by an ILU(2) decomposition. Local polynomial approximations of order $m = 2$, $m = 3$, and $m = 5$ have been used, leading respectively to $8,208$, $13,680$, and $28,728$ degrees of freedom. A comparison of the convergence for various values of *rtol* is shown in figure 5.

This clearly shows that the convergence test for GMRES can be loosened. Both the convergence rate w.r.t. computational time and the convergence rate w.r.t. the global number of GMRES iterations improve. Of course this observation cannot be simply generalized to all the test cases in this way. Nevertheless, this aspect shall be kept in mind as a possible way to accelerate convergence. In particular, we made use of this remark for the runs described in section V.A.2.

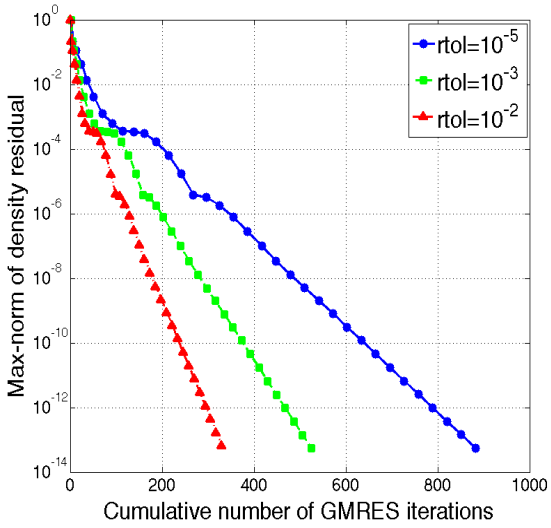
V.A.2. Partial-Storage Approach

We here show some results on the approach proposed in section IV.B.3. Its implementation revealed that in its current form it is not a profitable approach. We consider an inviscid smooth flow around a NACA0012 airfoil, with $M_\infty = 0.3$ and angle of attack $\alpha = 0°$. Airfoil profile, finest mesh and contours of constant Mach number are plotted in figure 6. We choose implicit damped Newton/GMRES method as outlined in section IV.B. In particular, we use GMRES(30) with the hybrid multilevel approach of section IV.C. Polynomials of order $m = 2$ are used on the highest level of approximation. Coherently with Ref. 33, 20 geometric multigrid cycles with explicit Shu RK3 smoothing (see section IV.A) are performed between two consecutive Newton steps, starting with a piecewise linear finite-volume method on the same mesh, followed

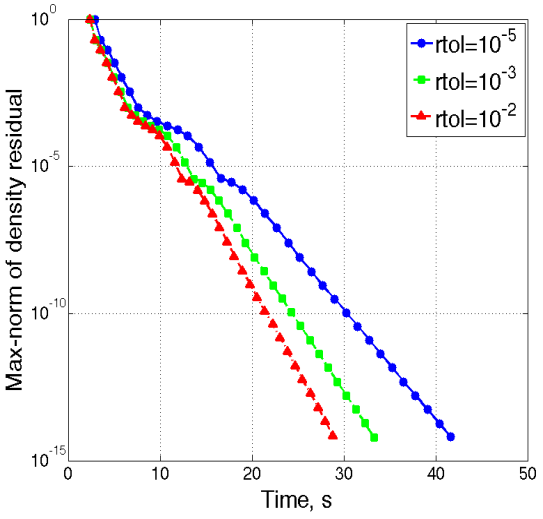
[a]Here a simplified version of the convergence test has been given. See Ref. 45 for the complete test.



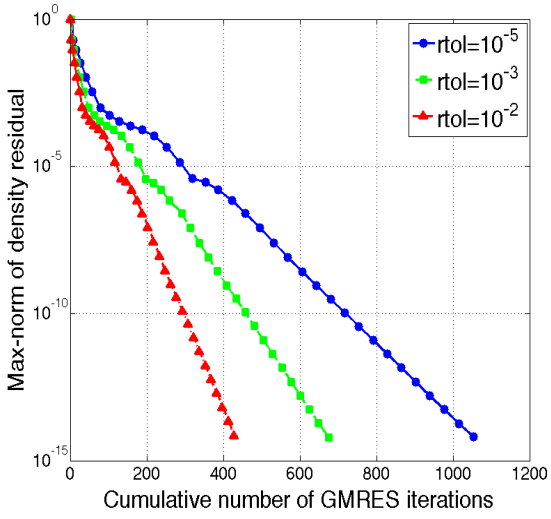
(a) Density residual vs. Time with $m = 2$ ($DOFs = 8,208$).



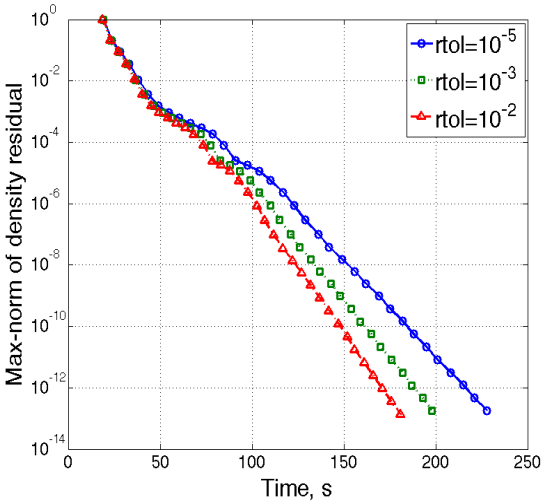
(b) Density residual vs. Cumulative number of GMRES iterations with $m = 2$ ($DOFs = 8,208$).



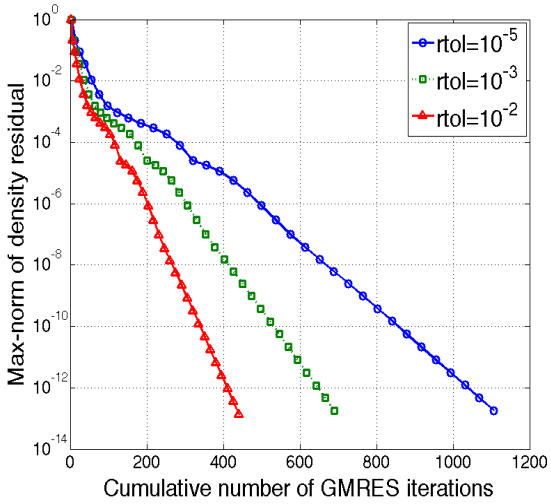
(c) Density residual vs. Time with $m = 3$ ($DOFs = 13,680$).



(d) Density residual vs. Cumulative number of GMRES iterations with $m = 3$ ($DOFs = 13,680$).

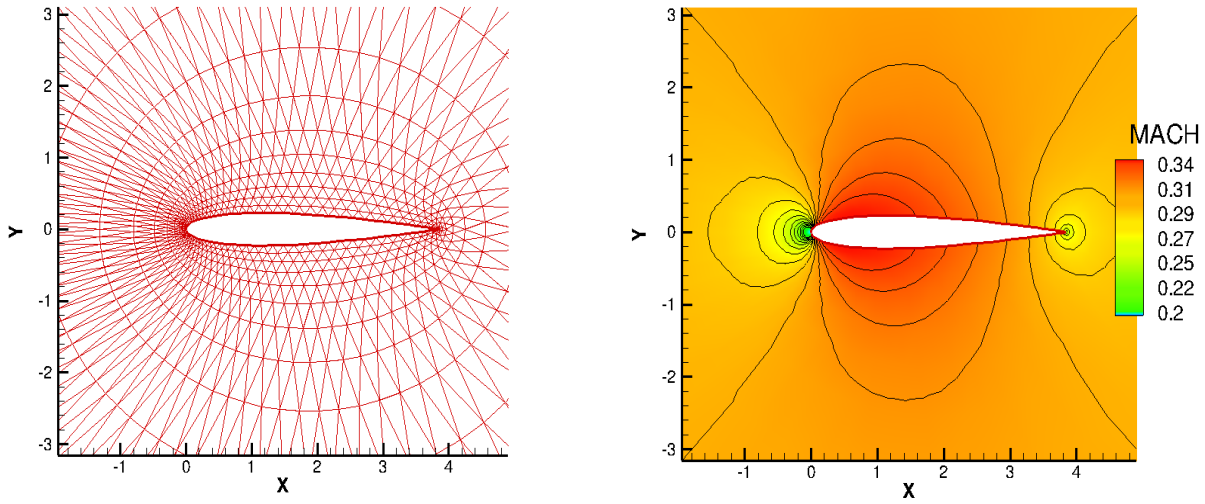


(e) Density residual vs. Time with $m = 5$ ($DOFs = 28,728$).



(f) Density residual vs. Cumulative number of GMRES iterations with $m = 5$ ($DOFs = 28,728$).

Figure 5. Inviscid flow over a smooth bump at $M_\infty = 0.3$. Convergence of the density residual $\|\mathbf{R}_\rho\|_\infty$ for different orders of approximation polynomials m and different relative tolerances $rtol$.



(a) Finest mesh (2,560 elements)

(b) Contours of constant Mach number

Figure 6. Inviscid flow around the NACA0012 profile at $M_\infty = 0.3$, angle of attack $\alpha = 0°$. A piecewise linear reconstruction is here used for the Mach number.

by first-order finite-volume on two coarser meshes. Table 1 summarizes the meshes used for this test case. On the highest level of approximation we compare the matrix-explicit approach of section IV.B.1, the matrix-free approach of section IV.B.2, and the partial-storage approach of section IV.B.3.

Level	Type of relaxation	Number of DOFs	m	Elements	CFL number
4	Implicit	61,440	2	2,560	550
3	Explicit	2,560	1	2,560	6
2	Explicit	640	0	640	6
1	Explicit	160	0	160	6

Table 1. Hybrid multilevel strategy for the NACA0012 test case. Relaxation method, number of degrees of freedom (DOFs), order of the approximating polynomials (m), number of elements in the mesh, and CFL number are given for each level of approximation.

We first consider implicit relaxation without preconditioning, so that we can have a first feedback on the performance of the method. We limit ourselves to 10 Newton iterations. The first aspect to be observed is that linear and nonlinear residuals (i.e., coming from the GMRES and damped netwon iterations, respectively) obtained with the matrix-explicit and the partial-storage approach are the same up to machine accuracy. This is natural, since the two approaches, when not preconditioned, execute the same operations, i.e., assemble the global matrix in the same exact explicit way. The word 'explicit' in the label matrix-explicit approach refers indeed to the fact that the matrix entries are explicitly stored. Figure 7 shows that the repeated assembly of the global matrix starting from the building blocks penalizes the partial-storage approach, with a computational time roughly 3 times larger w.r.t. the matrix-explicit approach. This, however, is counterbalanced by the storage requirement.

We now introduce preconditioning for both the approaches, following the indications of Section IV.B.4. We choose in particular ILU(2) as preconditioner for the matrix-explicit approach. We take into account the matrix-free approach as well. The partial-storage approach would in fact represent an alternative to the matrix-free approach, when storing the global matrix becomes an unfeasible choice. We also diminish the relative tolerance used in the convergence test for the GMRES method from 10^{-5} to 10^{-2} (see section V.A.1). The results are plotted in figure 8. By this comparison we could observe that similar number of GMRES iterations are needed by the partial-storage and the matrix-free approaches to reach the wished accuracy (here we limited it to 0.6×10^{-4}, since computational time would have become too large, hence uninteresting). The partial-storage approach, however, requires a time almost 5 times larger than the matrix-free approach. In other words, one GMRES iteration for the partial-storage approach is almost 4 times costlier in terms of

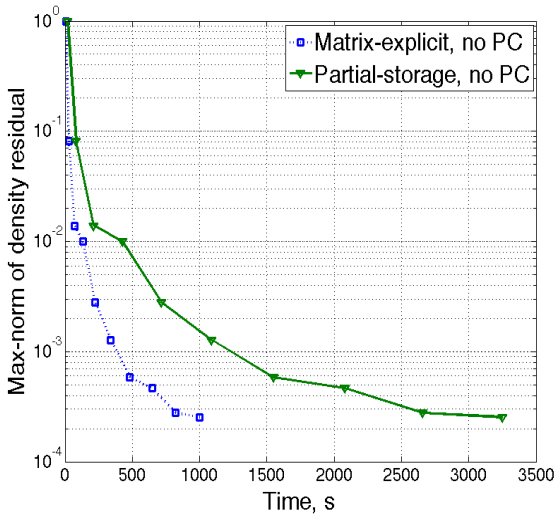
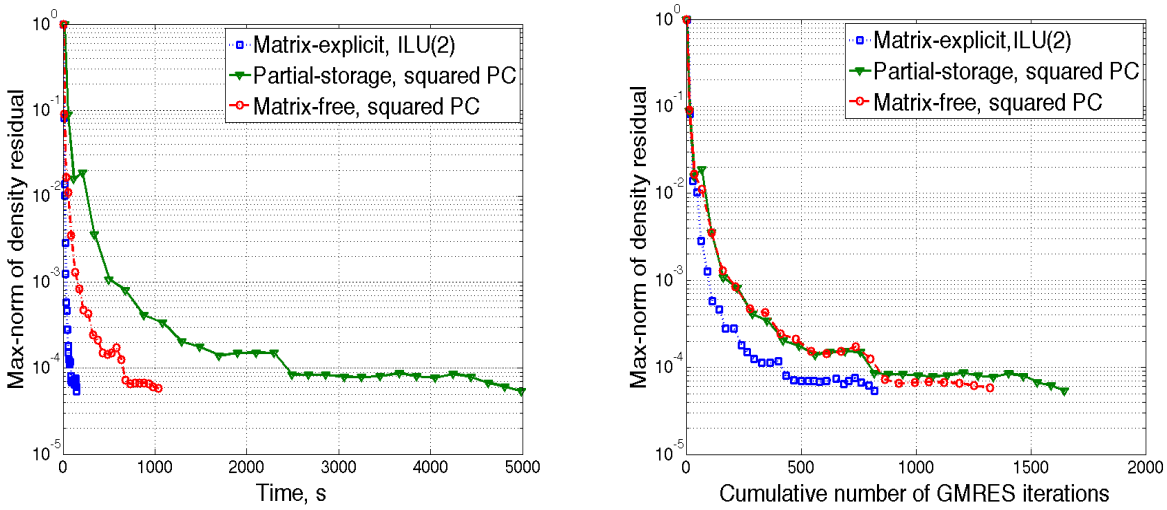


Figure 7. Inviscid flow around the NACA0012 profile at $M_\infty = 0.3$, $\alpha = 0°$. Convergenge of the density residual $\|\mathbf{R}_\rho\|_\infty$. Comparison of matrix-explicit and partial-storage approaches without preconditioning (PC).

time than a GMRES iteration for the matrix-free approach. This, together with the need to store the partial building subblocks, marks the approach in its current form as ineffective to satisfy our expectations.



(a) Density residual vs. Time

(b) Density residual vs. Cumulative number of GMRES iterations

Figure 8. Inviscid flow around the NACA0012 profile at $M_\infty = 0.3$, $\alpha = 0°$. Convergence of the density residual $\|\mathbf{R}_\rho\|_\infty$. Comparison of matrix-explicit, matrix-free, and partial-storage approaches with preconditioning (PC).

It must nevertheless be noticed that, by using the squared preconditioning for the partial-storage approach, we did not take advantage at all of the stored information given by the building subblocks of the global matrix. A careful analysis and understanding of the way in which one could use such information could lead to the achievement of a competitive strategy.

V.B. Convergence Acceleration Using Hybrid Multilevel Methods

The convergence of Newton methods benefits from a good initial approximation of the solution. We use a full multigrid method based on a nonlinear FAS multigrid method relaxation on the volume averages with multistage smoothing, to provide an initial approximation for the hybrid multilevel scheme, see section IV.C.1. Consider inviscid flow around the NACA0012 airfoil at flow conditions $M_\infty = 0.4$ and $\alpha = 5°$. Contours of constant Mach number for this test case are shown in figure 9. Figure 10 shows the convergence history of

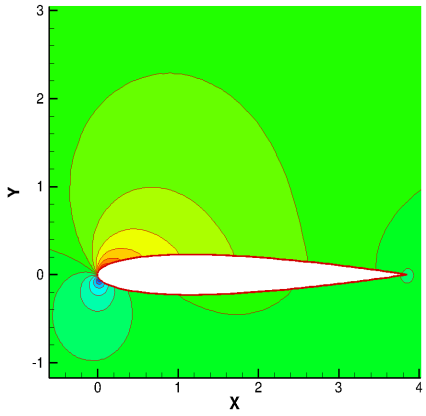
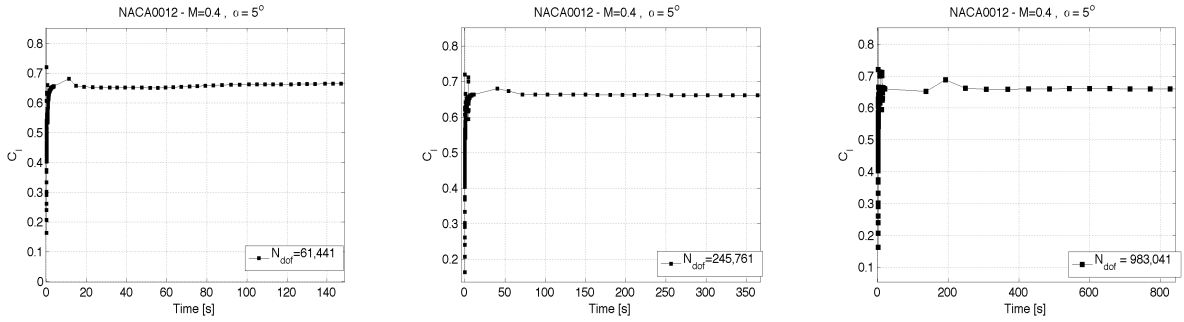


Figure 9. Mach number contours for flow around the NACA0012 profile at $M_\infty = 0.4$, $\alpha = 5^\circ$.



(a) Number of DOFs $= 61,440$ (b) Number of DOFs $= 245,760$ (c) Number of DOFs $= 983,040$

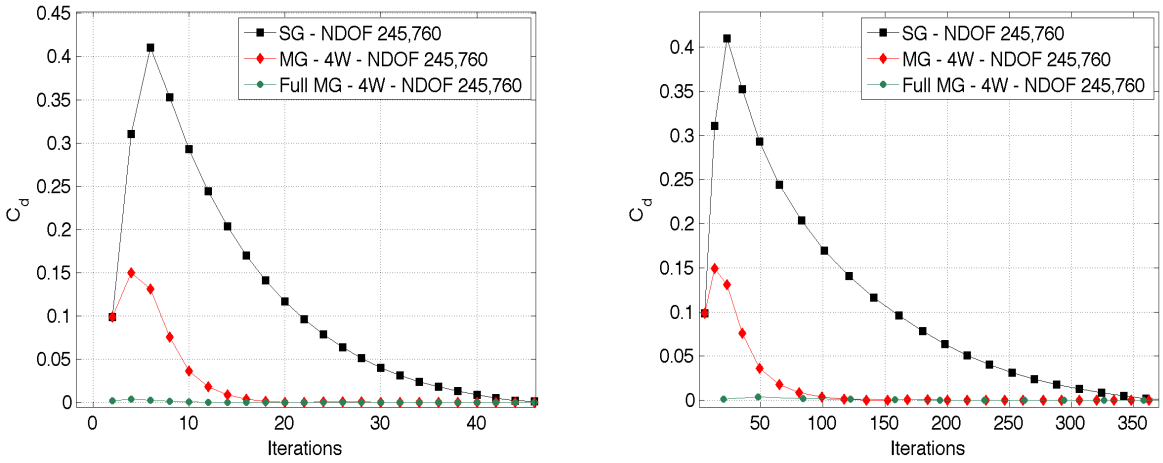
Figure 10. Mesh refinement study. Inviscid Flow around the NACA0012 profile at $M_\infty = 0.4$, $\alpha = 5^\circ$.

Level	N_{dof}	Cells
fine	983,040	40,960
medium	245,760	10,240
coarse	61,440	2,560

Table 2. Meshes used for h-refinement study for hybrid multilevel method.

the lift for three different meshes, as summarized in table 2. Note that the coarsest mesh in this refinement study is identical to the highest levels of approximation used previously with the multilevel approach, as summarized in table 1. The next finer meshes use the previous levels recursively, such that the coarsest mesh is always the same, and the total number of levels used increases by one for each refinement. Approximation polynomials of order $m = 2$ are always used on the finest level of approximation. GMRES(30) was used for linear iterations along with ILU(3) preconditioning. It can be seen in figure 10 that quite rapid convergence is attained.

The same mesh refinement study for inviscid flow around the NACA0012 profile was applied to the flow conditions $M = 0.3$ and $\alpha = 0^\circ$ in Ref. 33, which is repeated here with the additional full multigrid approach included. Convergence is compared between the multilevel method with and without full multigrid, and a straight implicit Newton method for the medium mesh ($N_{dof} = 245,760$) is shown in figure 11. The speed-up is dramatic, while computational overhead incurred for the multilevel method is just 15% per nonlinear iteration, if 20 multistage smoothing cycles are carried out between each Newton update.

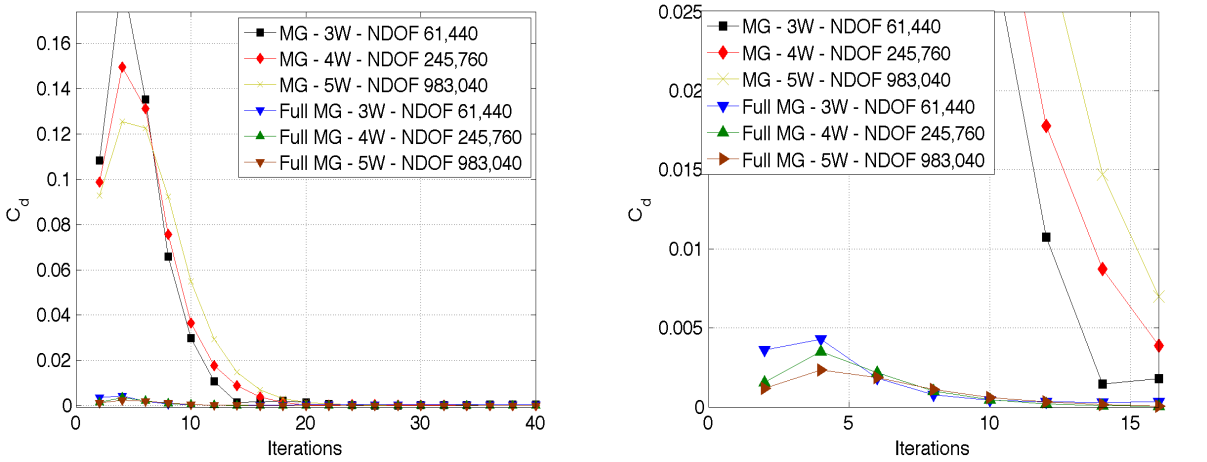


(a) Drag coefficient vs. Newton iterations

(b) Drag coefficient vs. accumulated GMRES iterations

Figure 11. Inviscid Flow around the NACA0012 profile at $M_\infty = 0.3$, $\alpha = 0^\circ$. Comparison of implicit relaxation procedures for the medium mesh ($N_{dof} = 245,760$). SG: single-grid damped Newton method as in Ref. 33. MG: Hybrid Multilevel method as in Ref. 33. Full MG: present full multigrid method.

For the mesh refinement study we focus on the multilevel method with and without full multigrid. See figure 12 for convegence of the drag coefficient. Convergence is nearly mesh-independent for both approaches at finite CFL numbers (nominal $CFL = 550$), with much improved start-up evident for the full multigrid approach.



(a) Drag coefficient vs. Newton iterations

(b) Magnified view of drag coefficient vs. Newton iterations

Figure 12. Inviscid Flow around the NACA0012 profile at $M_\infty = 0.3$, $\alpha = 0^\circ$.

VI. Conclusion and Future Work

In this work we have identified and assessed a best-practice strategy for convergence of high-order methods for steady Euler equations. As demonstrated, the full multigrid strategy based on the hybrid multilevel scheme accelerates the convergence in the transient quite efficiently. Mesh-independence has also nearly been reached.

Much of our attention has here focused on the transient phase. As we already stressed, in the steady-state setting, the design of the time-stepping scheme only aims at maximizing the rate of convergence. In the time-implicit relaxation framework, the main distinction is between exact Newton methods and damped Newton methods. In the first context, Newton's algorithm can be considered as a black box to find the zero of the residual. After the start-up phase the CFL number could be gradually increased up to infinite, with a fully-adaptive Newton implementation. In our experience, the time needed to pass from the initial iterate into the ball of convergence of Newton's method around the solution is often quite large. The combination of finite CFL numbers together with a low tolerance in the convergence test for the solution of the linear system, and cheap multigrid inner cycles between two consecutive damped-Newton iterations, often proves to be a better globalization strategy compared to linear search or trust region methods. Furthemore, in the time needed by Newton's method to reach the ball of quadratic convergence, the damped-Newton strategy has usually already converged to levels of accuracy commonly required by engineering applications.

In fact, finite CFL numbers lead to better-conditioned systems of equations, and this, in turn, affects the efficacy of preconditioning. Nonetheless, it would be opportune to have the option to choose asymptotically-infinite CFL numbers as well. For this, preconditioning, which in our experience is not very effective for our nodal scheme, has to be improved. This is not the only reason why preconditioning needs urgently to be addressed. A good preconditioner is still missing, both for the matrix-free and the partial-storage approaches. The latter exhibits the ability to exploit the particular structure of the global matrix and contemporarily retain information about it. A good preconditioner for this strategy should be able to take advantage of the specific assembly procedure in an analogously clever way. We anticipate this to be a challenging task, since the entries of the matrix to be preconditioned depend in a non-trivial way on the stored data. In particular, the entries are given by a linear combination of the available data.

Acknowledgments

Financial support from the Deutsche Forschungsgemeinschaft (German Research Association) through grant GSC 111, and by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number FA8655-08-1-3060, is gratefully acknowledged. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purpose notwithstanding any copyright notation thereon.

References

[1]Jameson, A., "Solution of the Euler Equations for Two Dimensional Transonic Flow by a Multigrid Method," *Appl. Math. Comput.*, Vol. 13, 1983, pp. 327–356.

[2]Jameson, A. and Yoon, S., "LU Implicit Schemes with Multiple Grids for the Euler Equations," *AIAA 24th Aerospace Sciences Meeting*, No. AIAA Paper 86-0105, AIAA, January 1986.

[3]Pierce, N. A. and Giles, M. B., "Preconditioned Multigrid Methods for Compressible Flow Calculations on Stretched Meshes," *J. Comput. Phys.*, Vol. 136, 1997, pp. 425–445.

[4]Jameson, A. and Caughey, D. A., "How Many Steps Are Required to Solve the Euler Equations of Steady Compressible Flow: in Search of a Fast Solution Algorithm," *15th Computational Fluid Dynamics Conference*, No. AIAA Paper 2001-2673, AIAA, June 2001.

[5]Mavriplis, D. J., "An Assessment of Linear Versus Nonlinear Multigrid Methods for Unstructured Mesh Solvers," *J. Comput. Phys.*, Vol. 175, 2002, pp. 302–325.

[6]Nastase, C. R. and Mavriplis, D. J., "High-Order Discontinuous Galerkin Methods Using an *hp*-Multigrid Approach," *J. Comput. Phys.*, Vol. 213, 2006, pp. 330–357.

[7]Cockburn, B. and Shu, C.-W., "The Runge-Kutta Local Projection P^1-Discontinuous-Galerkin Finite Element Method for Scalar Conservation Laws," *IMA Preprint Series*, Vol. 388, January 1988.

[8]Cockburn, B. and Shu, C.-W., "TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws II: General Framework," *Math. Comput.*, Vol. 52, No. 186, 1989, pp. 411–435.

[9]Cockburn, B., Lin, S.-Y., and Shu, C.-W., "TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws III: One-Dimensional Dystems," *J. Comput. Phys.*, Vol. 84, 1989, pp. 90–113.

[10]Cockburn, B., Hou, S., and Shu, C.-W., "The Runge-Kutta Discontinuous Galerkin Finite Element Method for Conservation Laws IV: the Multidimensional Case," *Math. Comput.*, Vol. 54, No. 190, 1990, pp. 545–581.

[11]Cockburn, B. and Shu, C.-W., "The Runge-Kutta Discontinuous Galerkin Method for Conservation Laws V: Multidimensional Systems," *J. Comput. Phys.*, Vol. 141, 1998, pp. 199–244.

[12]Shu, C.-W. and Osher, S., "Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes," *J. Comput. Phys.*, Vol. 77, 1988, pp. 439–471.

[13]Wang, Z. J., "Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids I: Basic Formulation," *J. Comput. Phys.*, Vol. 178, 2002, pp. 210–251.

[14]Wang, Z. J. and Liu, Y., "Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids II: Extension to Two-Dimensional Scalar Equation," *J. Comput. Phys.*, Vol. 179, 2002, pp. 665–697.

[15]Wang, Z. J., Zhang, L., and Liu, Y., "Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids IV: Extension to Two-Dimensional Systems," *J. Comput. Phys.*, Vol. 194, 2004, pp. 716–741.

[16]Liu, Y., Vinokur, M., and Wang, Z. J., "Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids V: Extension to Three-Dimensional Systems," *J. Comput. Phys.*, Vol. 212, 2006, pp. 454–472.

[17]Liu, Y., Vinokur, M., and Wang, Z. J., "Discontinuous Spectral Difference Method for Conservation Laws on Unstructured Grids," *Computational Fluid Dynamics 2004*, Springer Berlin Heidelberg, 2004, pp. 449–454.

[18]Liu, Y., Vinokur, M., and Wang, Z. J., "Spectral Difference Method for Unstructured Grids I: Basic Formulation," *J. Comput. Phys.*, Vol. 216, No. 2, 2006, pp. 780–801.

[19]Fidkowski, K. J., *A High-Order Discontinuous Galerkin Multigrid Solver for Aerodynamic Applications*, Ph.D. thesis, Massachussettes Institute of Technology, June 2004.

[20]Hackbusch, W. and Probst, T., "Downwind Gauss-Seidel Smoothing for Convection Dominated Pproblems," *Num. Lin. Alg. Appl.*, Vol. 4, 1997, pp. 85–102.

[21]Heys, J. J., Manteuffel, T. A., McCormick, S. F., and Olson, L. N., "Algebraic Multigrid for Higher-Order Finite Elements," *J. Comput. Phys.*, Vol. 204, 2005, pp. 520–532.

[22]Brandt, A., "Multi-Level Adaptive Solutions to Boundary-Value Problems," *Math. Comput.*, Vol. 31, No. 138, 1977, pp. 333–390.

[23]Bank, R. E., Dupont, T. F., and Yserentant, H., "The Hierarchical Basis Multigrid Method," *Numer. Math.*, Vol. 52, 1988, pp. 427–458.

[24]Jameson, A., "Acceleration of Transonic Flow Calculations on Arbitrary Meshes by the Multiple Grid Method," *4th AIAA Computational Fluid Dynamics Conference*, Vol. 1458, 1979, pp. 122–146.

[25]Mulder, W. A., "Multigrid, Alignment and Euler Equations," *Proceedings of the fourth Copper Mountain conference on multigrid methods*, edited by SIAM, 1989, pp. 348–364.

[26]Hackbusch, W., "On the Multi-Grid Method Applied to Difference Equations," *Computing*, Vol. 20, 1978, pp. 291–306.

[27]Mulder, W. A., "A New Multigrid Approach to Convection Problems," *J. Comput. Phys.*, Vol. 83, 1989, pp. 303–323.

[28]Hackbusch, W., *Multi-Grid Methods and Applications*, Springer, 1985.

[29]Jameson, A., "Multigrid Algorithms for Compressible Flow Calculations," *Multigrid Methods II*, edited by W. Hackbusch and U. Trottenberg, Vol. 1228 of *Lect. Notes Math.*, Proceedings of the 2nd European Conference on Multigrid Methods, Springer, 1986, pp. 166–201.

[30]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," *Int. J. Comput. Fluid Dynam.*, Vol. 4, 1995, pp. 171–218.

[31]Helenbrook, B. T., Mavriplis, D. J., and Atkins, H. L., "Analysis of "p"-Multigrid for Continuous and Discontinuous Finite Element Discretizations," *16th AIAA Computational Fluid Dynamics Conference*, No. AIAA Paper 2003-3989, June 2003.

[32]Luo, H., Baum, J. D., and Löhner, R., "A *p*-Multigrid Discontinuous Galerkin Method for the Euler Equations on Unstructured Grids," *J. Comput. Phys.*, Vol. 211, 2006, pp. 767–783.

[33]May, G., Iacono, F., and Jameson, A., "Efficient Algorithms for High-Order Discretizations of the Euler and Navier-Stokes Equations," *47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, No. AIAA Paper 2009-182, January 2009.

[34]May, G., *A Kinetik Scheme for the Navier-Stokes Equations and High-Order Methods for Hyperbolic Conservation Laws*, Ph.D. thesis, Department of aeronautics and astronautics, Stanford University, September 2006.

[35]Wang, Z. J., Liu, Y., May, G., and Jameson, A., "Spectral Difference Method for Unstructured Grids II: Extension to the Euler Equations," *J. Sci. Comput.*, Vol. 32, No. 1, July 2007, pp. 45–71.

[36]May, G., "The Spectral Scheme as Quadrature-Free Discontinuous Galerkin Method," Tech. Rep. AICES-2008-11, Aachen Institute for advanced study in Computational Engineering Science, http://www.aices.rwth-aachen.de/research/preprints, December 2008.

[37]Jameson, A., "Analysis and Design of Numerical Schemes for Gas Dynamics 2: Artificial Diffusion and Discrete Shock Structure," *Int. J. Comp. Fl. Dyn.*, Vol. 5, 1995, pp. 1–38.

[38]Hesthaven, J. S., "From Electrostatics to Almost Optimal Nodal Sets for Polynomial Interpolation in a Simplex," *SIAM J. Numer. Anal.*, Vol. 35, No. 2, 1998, pp. 655–676.

[39]van den Abeele, K., Lacor, C., and Wang, Z. J., "On the Stability and Accuracy of the Spectral Difference Method," *J. Sci. Comput.*, Vol. 37, No. 2, 2008, pp. 162–188.

[40]Cockburn, B. and Shu, C.-W., "The Local Discontinuous Galerkin Method for Time-Dependent Convection-Diffusion Systems," *SIAM J. Numer. Anal.*, Vol. 35, No. 6, December 1998, pp. 2440–2463.

[41]Knoll, D. A. and Keyes, D. E., "Jacobian-Free Newton-Krylov Methods: a Survey of Approaches and Applications," *J. Comput. Phys.*, Vol. 193, 2004, pp. 357–397.

[42]Saad, Y. and Schultz, M. H., "GMRES: a Generalized Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM J. Sci. Stat. Comput.*, Vol. 7, No. 3, 1986, pp. 856–869.

[43]Saad, Y., *Iterative Methods for Sparse Linear Systems*, SIAM, 2nd ed., 2003.

[44]Souläimani, A., Salah, N. B., and Saad, Y., "Enhanced GMRES Acceleration Techniques for Some CFD Problems," *Int. J. Comp. Fl. Dyn.*, Vol. 16, No. 1, 2002, pp. 1–20.

[45]Balay, S., Buschelman, K., Eijkhout, V., Gropp, W., Kaushik, D., Knepley, M., McInnes, L. C., Smith, B., and Zhang, H., *PETSc Users Manual*, Argonne National Laboratory, http://www.mcs.anl.gov/petsc/petsc-as/snapshots/petsc-current/docs/manual.pdf, December 2008.