Multigrid Solution for High-Order Discontinuous Galerkin Discretizations of the Compressible Navier-Stokes Equations

Todd A. Oliver[1], Krzysztof J. Fidkowski[2], and David L. Darmofal[3]

[1] Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139. toliver@mit.edu
[2] kfid@mit.edu
[3] darmofal@mit.edu

1 Introduction

While CFD has achieved significant maturity during the past decades, computational costs are extremely large for aerodynamic simulations of aerospace vehicles. In this applied aerodynamics context, the discretization of the Euler or Navier-Stokes equations is performed almost exclusively by finite volume algorithms. The accuracy of many of these methods is at best second order; i.e. the error decreases as $O(h^2)$ where h is a measure of the grid spacing. Recent studies have shown that this rate may not be adequate for modern engineering applications [4, 8]. The development of a practical higher-order solution method could alleviate this problem by significantly decreasing the computational time required to achieve an acceptable error level.

Numerous reasons exist for why current finite-volume algorithms are not practical at higher-order. A root cause of many of these difficulties lies in the extended stencils that these algorithms employ. These extended stencils lead to high memory requirements due to matrix fill-in and difficulties in achieving stable iterative algorithms [10].

By contrast, finite element formulations introduce higher-order effects compactly within the element. For discontinuous Galerkin (DG) formulations, the element-to-element coupling exists only through the fluxes at the shared boundaries between elements. This limited coupling is an enabling feature which permits the development of an efficient higher-order solver and potentially *significant* improvements in the turn-around time for reliably accurate aerodynamic simulations.

2 Discontinuous Galerkin Discretization

The two-dimensional, compressible Navier-Stokes equations in conservation form are given by:

$$\boldsymbol{u}_t + \nabla \cdot \mathcal{F}_i(\boldsymbol{u}) - \nabla \cdot \mathcal{F}_v(\boldsymbol{u}, \nabla \boldsymbol{u}) = 0, \tag{1}$$

where $\boldsymbol{u} = (\rho, \rho u, \rho v, \rho E)^T$ is the conservative state vector, $\mathcal{F}_i = (\boldsymbol{F}_i^x, \boldsymbol{F}_i^y)$ is the inviscid flux, and $\mathcal{F}_v = \mathcal{A}_v \nabla \boldsymbol{u} = (\boldsymbol{F}_v^x, \boldsymbol{F}_v^y)$ is the viscous flux.

2.1 Inviscid Discretization

Denote by $\mathcal{V}_h^p$ the space of discontinuous vector-valued polynomials of degree p on a subdivision T_h of the domain Ω into elements such that $\Omega = \bigcup_{\kappa \in T_h} \kappa$. The DG discretization of the Euler equations is of the following form: find $\boldsymbol{u}_h \in \mathcal{V}_h^p$ such that $\forall \boldsymbol{v}_h \in \mathcal{V}_h^p$,

$$\sum_{\kappa \in T_h} \Big\{ \int_\kappa \boldsymbol{v}_h^T (\boldsymbol{u}_h)_t \, d\boldsymbol{x} - \int_\kappa \nabla \boldsymbol{v}_h^T \cdot \mathcal{F}_i(\boldsymbol{u}_h) d\boldsymbol{x} + \int_{\partial\kappa \setminus \partial\Omega} {\boldsymbol{v}_h^+}^T \mathcal{H}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \hat{\boldsymbol{n}}) ds + \int_{\partial\kappa \cap \partial\Omega} {\boldsymbol{v}_h^+}^T \mathcal{H}_{\mathrm{bd}}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^b, \hat{\boldsymbol{n}}) ds \Big\} = 0, \tag{2}$$

where $\mathcal{H}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \hat{\boldsymbol{n}})$ and $\mathcal{H}_{\mathrm{bd}}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^b, \hat{\boldsymbol{n}})$ are inviscid, numerical flux functions for interior and boundary edges, respectively. Also, the $()^+$ and $()^-$ notation indicates the trace value taken from the interior and exterior of the element, respectively, and $\hat{\boldsymbol{n}}$ is the outward-pointing normal of the element. For the interior flux function, the Roe-averaged flux function is used [11].

2.2 Viscous Discretization

The viscous terms in Eqn. 1 are discretized using the second form of Bassi and Rebay (BR2) [2]. To write the dicretization in compact form, jump, $[\![\cdot]\!]$, and average, $\{\cdot\}$, operators are defined. For scalar quantities, the operators are given by

$$[\![s]\!] = s^+ \hat{\boldsymbol{n}}^+ + s^- \hat{\boldsymbol{n}}^- \qquad \{s\} = \frac{1}{2}(s^+ + s^-),$$

where $(\cdot)^+$ and $(\cdot)^-$ refer to trace values taken from opposite sides of the face. Thus, $\hat{\boldsymbol{n}}^+ = -\hat{\boldsymbol{n}}^-$. For vector quantities,

$$[\![\boldsymbol{\varphi}]\!] = \boldsymbol{\varphi}^+ \cdot \hat{\boldsymbol{n}}^+ + \boldsymbol{\varphi}^- \cdot \hat{\boldsymbol{n}}^- \qquad \{\boldsymbol{\varphi}\} = \frac{1}{2}(\boldsymbol{\varphi}^+ + \boldsymbol{\varphi}^-).$$

The BR2 discretization is given by the following form: find $\boldsymbol{u}_h \in \mathcal{V}_h^p$ such that $\forall \boldsymbol{v}_h \in \mathcal{V}_h^p$,

$$\mathbb{E} + \sum_{\kappa \in T_h} \int_\kappa \nabla \boldsymbol{v}_h^T \cdot (\mathcal{A}_v \nabla \boldsymbol{u}_h) d\boldsymbol{x} - \int_{\Gamma_i} \Big([\![\boldsymbol{u}_h]\!]^T \cdot \{\mathcal{A}_v^T \nabla \boldsymbol{v}_h\} + [\![\boldsymbol{v}_h]\!]^T \cdot \{\mathcal{A}_v \nabla \boldsymbol{u}_h\} \Big) ds + \int_{\Gamma_i} [\![\boldsymbol{v}_h]\!]^T \cdot \{\boldsymbol{\delta}_f\} ds + \int_{\partial\Omega} (\boldsymbol{u}^b - \boldsymbol{u}_h^+)^T (\mathcal{A}_v^T \nabla \boldsymbol{v}_h)^+ \cdot \hat{\boldsymbol{n}} ds - \int_{\partial\Omega} {\boldsymbol{v}_h^+}^T (\mathcal{A}_v \nabla \boldsymbol{u}_h)^b \cdot \hat{\boldsymbol{n}} ds + \int_{\Gamma_i} {\boldsymbol{v}_h^+}^T \boldsymbol{\delta}_f^b \cdot \hat{\boldsymbol{n}} ds = 0. \tag{3}$$

where Γ_i is the union of interior faces, $\mathbb{E}$ is the Euler discretization defined in Section 2.1, and $\boldsymbol{\delta}_f$ and $\boldsymbol{\delta}_f^b$ are auxiliary variables. The auxiliary variables are given by the following weak statement: find $\boldsymbol{\delta}_f \in [\mathcal{V}_h^p]^2$ such that $\forall \boldsymbol{\psi}_h \in [\mathcal{V}_h^p]^2$,

$$\int_{\kappa^\pm} \boldsymbol{\psi}_h^T \cdot \boldsymbol{\delta}_f^\pm d\boldsymbol{x} = -\frac{1}{2} \int_{\sigma_f} [(\mathcal{A}_v^T \boldsymbol{\psi}_h) \cdot \hat{\boldsymbol{n}}]^{\pm^T} (\boldsymbol{u}_h^\mp - \boldsymbol{u}_h^\pm) ds \tag{4}$$

for interior faces, and

$$\int_{\kappa^+} \boldsymbol{\psi}_h^T \cdot \boldsymbol{\delta}_f^b d\boldsymbol{x} = -\int_{\sigma_f} [(\mathcal{A}_v^T \boldsymbol{\psi}_h) \cdot \hat{\boldsymbol{n}}]^{+^T} (\boldsymbol{u}_h^b - \boldsymbol{u}_h^+) ds \tag{5}$$

for boundary faces, where σ_f denotes the face indexed by f.

The boundary conditions on $\partial\Omega$ are imposed weakly by constructing an exterior boundary state, $\boldsymbol{u}_h^b$, and a normal derivative of the state, $(\nabla \boldsymbol{u} \cdot \hat{\boldsymbol{n}})^b$, on $\partial\Omega$ that are a functions of the interior quantities and boundary condition data. These quantities are then used to construct the inviscid and viscous fluxes at the domain boundaries.

The final discrete form of the DG discretization is constructed by selecting a basis for $\mathcal{V}_h^p$. Specifically, a set of element-wise discontinuous functions $\{\phi_j\}$ is introduced, such that each ϕ_j has local support on only one element. The solution to the DG discretization has the following form,

$$\boldsymbol{u}_h(t, x) = \sum_j \overline{\boldsymbol{u}}_j(t) \phi_j(x).$$

A simple backward Euler discretization in time is used so that the final discrete equations are

$$\mathcal{M} \frac{1}{\Delta t} \left(\overline{\boldsymbol{u}}^{n+1} - \overline{\boldsymbol{u}}^n \right) + \boldsymbol{R}(\overline{\boldsymbol{u}}^{n+1}) = 0, \tag{6}$$

where $\mathcal{M}$ is the mass matrix and $\boldsymbol{R}$ is the residual vector representing discretized viscous and inviscid fluxes. In the following discussion, the overbar notation for the discrete solution vector is dropped.

3 Solution Method

To solve the nonlinear system, $\boldsymbol{R}(\boldsymbol{u}) = 0$, a p-multigrid scheme with a line Jacobi smoother is used. A generic iterative scheme can be written as,

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n - \boldsymbol{P}^{-1} \boldsymbol{R}(\boldsymbol{u}^n), \tag{7}$$

where the preconditioner, $\boldsymbol{P}$, is an approximation to $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}$. Two preconditioners have been considered: an elemental block-Jacobi smoother, in which the unknowns on each element are solved simultaneously, and an elemental line block-Jacobi smoother, in which the unknowns on each line of elements are solved simultaneously. This work focuses on the details of the line smoother and the multigrid solver based on it.

3.1 Line-Implicit Smoother

In strongly convective systems, the transport of information proceeds along characteristic directions. Thus, by solving implicitly on lines of elements connected along these directions, one can alleviate the stiffness associated with strong convection.

Furthermore, for viscous flows, the line solver is an important ingredient in removing the stiffness associated with regions of high grid anisotropy frequently required in viscous layers[1, 9]. In such cases, the lines are formed between elements which exhibit the strongest coupling. In this work, the coupling is defined by the Jacobian of the $p = 0$ discretization of the scalar transport equation,

$$\rho \boldsymbol{u} \cdot \nabla \phi - \mu \nabla^2 \phi = 0$$

where $\rho \boldsymbol{u}$ and μ are taken from the solution at the current iteration. More specfically, the coupling between two elements j and k that share a face is given by

$$\boldsymbol{C}_{j,k} = \max \left(\left| \frac{\partial R_j}{\partial \phi_k} \right|, \left| \frac{\partial R_k}{\partial \phi_j} \right| \right) \tag{8}$$

Using this definition of coupling, lines of elements are formed using the line creation algorithm of Fidkowski and Darmofal [5, 4].

The initial preconditioner for line smoothing consists of subsystems, $\boldsymbol{M}^l$, formed from $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}$, for each line l. $\boldsymbol{M}$ denotes the assembled $\boldsymbol{M}^l$ matrices. To make the system better conditioned, $\boldsymbol{M}$ is augmented by the addition of an unsteady term, yielding the final preconditioner

$$\boldsymbol{P} = \boldsymbol{M} + \frac{1}{\Delta t} \mathcal{M}, \tag{9}$$

The addition of the time term corresponds to solving for a finite time step, Δt, in the unsteady problem. As the solution begins to converge, $\Delta t \to \infty$. Thus, the steady-state solution is obtained.

The inversion of $\boldsymbol{P}$ uses a block-tridiagonal algorithm in which the block diagonal is LU decomposed. As the dominant cost of the line solver (especially for higher-order schemes) is the LU decomposition of the diagonal, the computational cost of the line smoother scales as that of the simpler elemental block-Jacobi. However, the performance of the line smoother is significantly better due to the increased implicitness along strongly coupled directions.

3.2 *p*-Multigrid

p-Multigrid is used in conjunction with the line smoother to improve the performance of the solver. In standard multigrid techniques, solutions on spatially coarser grids are used to correct solutions on the fine grid. In p-multigrid, the idea is similar except that lower order interpolants serve as the "coarse" grids[12, 6].

To solve the nonlinear system in question, the Full Approximation Scheme (FAS) was chosen as the multigrid method. Much of the description that follows is adapted from Briggs[3].

Consider the discretized system of equations given by $\mathbf{R}^p(\mathbf{u}^p) = \mathbf{f}^p$, where $\mathbf{u}^p$ is the discrete solution vector for p^{th} order interpolation, $\mathbf{R}^p(\mathbf{u}^p)$ is the nonlinear system, and $\mathbf{f}^p$ is a source term. Let $\mathbf{v}^p$ be an approximation to the solution vector and define the discrete residual by $\mathbf{r}^p(\mathbf{v}^p) \equiv \mathbf{f}^p - \mathbf{R}^p(\mathbf{v}^p)$. In a basic two-level multigrid method, the exact solution on a coarse level is used to correct the solution on a fine level. This correction scheme is given as follows:

- Restrict the state and residual to the coarse level: $\mathbf{v}_0^{p-1} = \tilde{I}_p^{p-1}\mathbf{v}^p$, $\mathbf{r}^{p-1} = I_p^{p-1}\mathbf{r}^p$.
- Solve the coarse level problem: $\mathbf{R}^{p-1}(\mathbf{v}^{p-1}) = \mathbf{R}^{p-1}(\mathbf{v}_0^{p-1}) + \mathbf{r}^{p-1}$.
- Prolongate the coarse level error and correct the fine level state: $\mathbf{v}^p = \mathbf{v}^p + I_{p-1}^p(\mathbf{v}^{p-1} - \mathbf{v}_0^{p-1})$.

I_p^{p-1} is the residual restriction operator, and I_{p-1}^p is the state prolongation operator. $\tilde{I}_p^{p-1}$ is the state restriction operator and is not necessarily the same as residual restriction. The definitions of these operators are found in [5, 4]. Alternatively, the FAS coarse level equation can be written as

$$\begin{aligned}\mathbf{R}^{p-1}(\mathbf{v}^{p-1}) &= I_p^{p-1}\mathbf{f}^p + \tau_p^{p-1},\\ \tau_p^{p-1} &\equiv \mathbf{R}^{p-1}(\tilde{I}_p^{p-1}\mathbf{v}^p) - I_p^{p-1}\mathbf{R}^p(\mathbf{v}^p).\end{aligned}$$

The first equation differs from the original coarse level equation by the presence of the term τ_p^{p-1}, which improves the correction property of the coarse level. In particular, if the fine level residual is zero, the coarse level correction is zero since $\mathbf{v}^{p-1} = \mathbf{v}_0^{p-1}$.

To make multigrid practical, the basic two level correction scheme is extended to a V-cycle and to full multigrid (FMG). In a V-cycle, a sequence of coarse levels (two or more) is used to correct the solution on the fine level. In FMG, V-cycles on successively finer levels are used to approximate the solution on the finest level. Improved performance and robustness result from starting iterations on the coarsest level rather than the finest. For more details on the multigrid implementation, see [5, 4].

4 Stability Analysis

To determine the stability of the line-implicit relaxation applied to the DG discretization of the Navier-Stokes equations, Fourier analysis is performed on the two-dimensional advection-diffusion problem with periodic boundary conditions. The advection-diffusion problem is given by

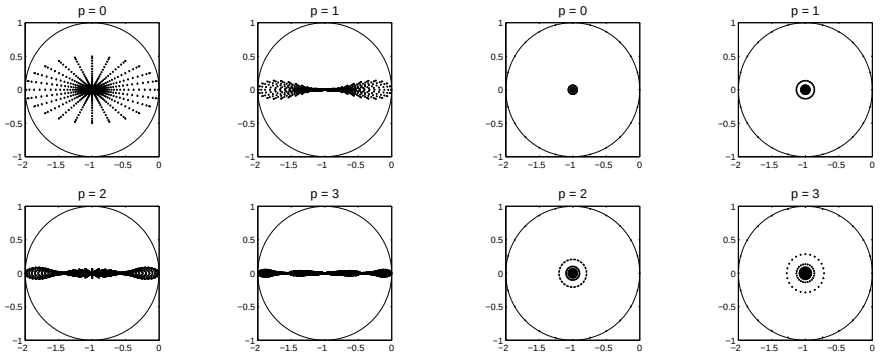
$$au_x + bu_y - \nu(u_{xx} + u_{yy}) = f(x,y) \quad \text{on} \quad [-1,1]\times[-1,1]. \tag{10}$$

Non-dimensionalizing, Eqn. 10 becomes

$$Re(u_{\tilde{x}} + \tan\alpha \, \mathrm{AR} \, u_{\tilde{y}}) - (u_{\tilde{x}\tilde{x}} + \mathrm{AR}^2 u_{\tilde{y}\tilde{y}}) = \frac{\Delta x^2}{\nu} f \quad \text{on} \quad [-1,1]\times[-1,1], \tag{11}$$

where $Re = a\Delta x/\nu$, $\tan\alpha = b/a$, $\mathrm{AR} = \Delta x/\Delta y$, $\tilde{x} = x/\Delta x$, and $\tilde{y} = y/\Delta y$, and Δx and Δy are the grid spacing in the x and y directions, respectively,.

Triangulating the domain $[-1,1]\times[-1,1]$ into $N_x N_y$ rectangular elements and choosing the tensor product of the 1-D Lagrange basis to be the 2-D basis, the discretized form of Eqn. 11 can be written as the system $\mathbf{A}\boldsymbol{u} = \mathbf{f}$. To determine the relaxation footprint, one must compute the eigenvalues of $-\mathbf{P}^{-1}\mathbf{A}$, where $\mathbf{P}$ is the line-implicit preconditioner defined in Section 3.1. These eigenvalues are computed via Fourier stability analysis [4]. While the footprints cannot be computed analytically, numerical results show that, over the wide range of Re, α, AR, and p considered, the eigenvalues are stable. Fig. 1 shows the footprints for $p = 0, 1, 2, 3$ for two cases.

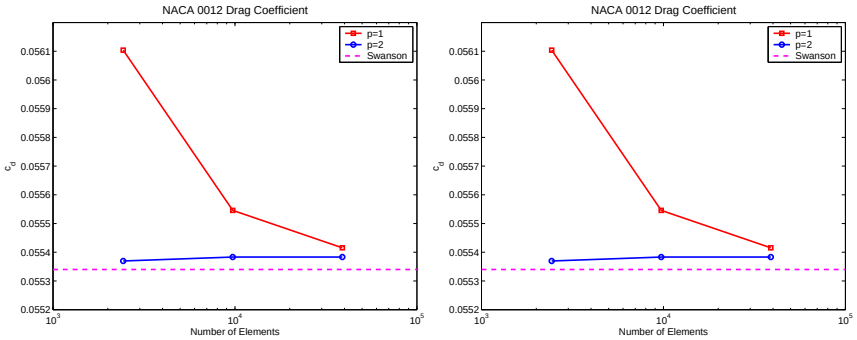


(a) $Re = 10$, $\mathrm{AR} = 100$, $\alpha = 1^o$ (b) $Re = 10^5$, $\mathrm{AR} = 1$, $\alpha = 1^o$

Fig. 1. Line relaxation footprints

5 Results

This section presents results for flow over a NACA 0012 airfoil at $\alpha = 0^o$, $Re = 5000$, $M = 0.5$, with a no slip adiabatic wall. Fig. 2(a) shows convergence of the drag coefficient error with number of elements, and Fig. 2(b) shows the drag coefficient error versus total degrees of freedom (DOF). In both cases, the error is relative to a $p = 3$ solution on *number* elements



(a) Error vs. h (b) Error vs. DOF

Fig. 2. Drag results for NACA 0012, $\alpha = 0^o$, $Re = 5000$, $M = 0.5$

(*number* DOF). Fig. 2(a) shows that optimal order of accuracy is obtained, meaning that the drag error decreases as $O(h^{p+1})$. It also shows the dramatic benefits of high-order solutions in terms of number of elements required to attain a desired accuracy. The $p = 2$ results with only *number* elements are significantly closer to the "exact" solution than the $p = 1$ result on th finest grid. Furthermore, Fig. 2(b) shows the high-order solutions achieve lower error than $p = 1$ with fewer DOF. The $p = 2$ case requires *number* DOF and $p = 3$ requires *number* DOF to achieve the fine grid $p = 1$ error level.

6 Conclusions

A p-multigrid solution method for a high-order DG discretization of the two-dimensional, compressible Navier-Stokes equations has been presented. The line-implicit smoother employed in the p-multigrid solver has been show to be stable for a wide range of flow conditions and interpolation orders, and optimal order of accuracy has been demonstrated for a typical, smooth test case. Furthermore, higher-order solutions have been shown to provide significant gains over $p = 1$ solutions in terms of number of elements and number of degrees of freedom required to achieve a desired error level.

Many oportunities for future work remain. Specifically, no problems with shocks have been considered here. In cases with shocks. limiting is required to stabilize the oscillatory behavior of high-order approximations near discontinuities. A limiter has yet to be implemented. Also, only laminar flows have been considered. A turbulence model must be implemented to allow consideration of a wider class of flows of engineering interest.

References

1. S.R. Allmaras. Analysis of semi-implicit preconditioners for multigrid solution of the 2-d compressible navier-stokes equations, 1995.
2. F. Bassi and S. Rebay. GMRES discontinuous galerkin solution of the compressible Navier-Stokes equations. In Karniadakis Cockburn and Shu, editors, *Discontinuous Galerkin Methods: Theory, Computation and Applications*, pages 197–208. Springer, Berlin, 2000.
3. William Briggs, Van Emden Henson, and Steve F. McCormick. *A Multigrid Tutorial, 2nd Ed.* SIAM, 2000.
4. K. Fidkowski. A high-order discontinuous galerkin multigrid solver for aerodynamic applications. Master's thesis, Massachusetts Institute of Technology, 2004.
5. K. Fidkowski and D. Darmofal. Development of a higher-order solver for aerodynamic applications. *42nd AIAA Aerospace Sciences Meeting.*, AIAA Paper Number 2004-0436, 2004.
6. B.T. Helenbrook, D.J. Mavriplis, and H.A. Atkins. Analysis of p-multigrid for continuous and discontinuous finite element discretizations. AIAA Paper 2003-3989, 2003.
7. David W. Levy, Thomas Zickuhr, John Vassberg, Shreekant Agrawal, Richard A. Wahls, Shahyar Pirzadeh, and Michael J. Hemsch. Data summary from the First AIAA Computational Fluid Dynamics Drag Prediction Workshop. *Journal of Aircraft*, 40(5):875–882, 2003.
8. D.J. Mavriplis. Multigrid strategies for viscous flow solvers on anisotropic unstructured meshes. *Journal of Computational Physics*, 145:141–165, 1998.
9. D.J. Mavriplis. An assessment of linear versus nonlinear multigrid methods for unstructured mesh solvers. *Journal of Computational Physics*, 175:302–325, 2001.
10. P.L. Roe. Approximate riemann solvers, parametric vectors, and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.
11. E. M. Rønquist and A. T. Patera. Spectral element multigrid. i. formulation and numerical results. *J. Sci. Comput.*, 2(4):389–406, 1987.