21st AIAA Computational Fluid Dynamics Conference
June 24-27, 2013, San Diego, CA

AIAA 2013-2870

Design of a Variational Multiscale Method for Turbulent Compressible Flows

Laslo T. Diosady [*]   Scott M. Murman [†]

NASA Ames Research Center, Moffett Field, CA, USA

The ultimate goal of this work is the simulation of separated flow about the three-dimensional FAITH bump to support Reynolds-averaged modeling efforts. Given the relatively high Reynolds number of this experimental configuration, numerical efficiency becomes paramount to enable practical simulations. This work describes the progress in the design of a VMM for high-Reynold-number separated flows, focusing on performance vs. accuracy trade-offs of the numerical schemes.

I. Introduction

Prediction of separation in high Reynolds number turbulent flows is necessary in order to accurately compute forces about even simple geometries. Reynolds-averaged Navier-Stokes (RANS) models, which solve only for the temporally averaged flow field, have generally been unable to accurately predict these flows which feature large scale unsteadiness. Thus, time-accurate simulations are desired which are able to capture as much of the relevant physics as possible. Unfortunately, alternatives to RANS models such as direct numerical simulation (DNS) and large eddy simulation (LES) have resolution requirements which grow rapidly with Reynolds number, making these techniques unfeasible in a practical engineering setting for the foreseeable future. In the interim, high-fidelity time-accurate simulation will be used for detailed investigation of high Reynolds number turbulent flows about basic geometries, such as the Fundamental Aeronautics Investigation of The Hill (FAITH) model.

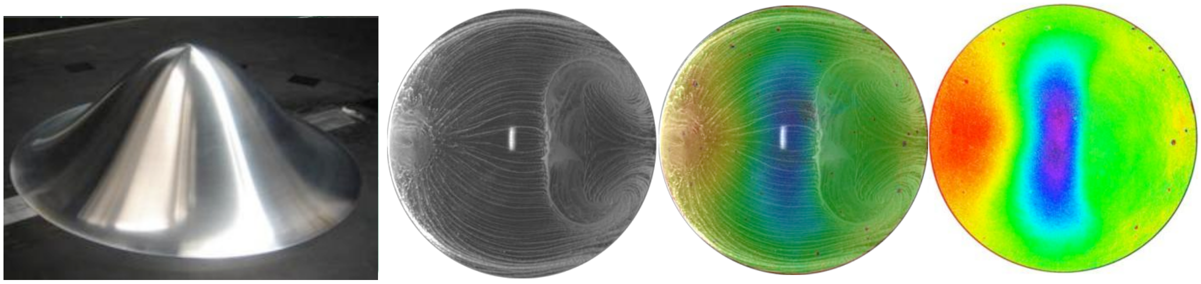
The objective of the FAITH program is to generate an experimental database to assess the ability to predict flow separation location and complex flow behavior in the separated regions. The FAITH program uses a variety of experimental techniques to measure the flow around a simple axisymmetric hill model (depicted in Figure 1).[1] The flow about the FAITH model involves a large turbulent separation region which remains a challenge to model using existing RANS methods. The ultimate goal of our work is to perform a high-fidelity simulation of separated flow about the three-dimensional FAITH bump to support Reynolds-averaged modeling efforts. Given the relatively high Reynolds number of this experimental configuration, numerical efficiency becomes paramount to enable practical simulations.

The Variational Multiscale Method (VMM) is a reformulation of the classical Large-eddy Simulation (LES), where the filtering operation, used to explicitly separate resolved and unresolved scales, is replaced by a finite-element projection operator.[2] As projection and differentiation commute, VMM does not suffer from commutation errors present in classical LES. VMM subgrid-scale models can be limited to the finest computed modes, so that no modeling error is introduced into the resolved coarser modes.[3] Combined with adaptivity, this provides a robust h-p spectral-element framework for viscous simulations which has shown promise for predicting attached wall-bounded flows.[2,4–9]

High fidelity simulation of the FAITH hill model poses a significant numerical challenge. We intend to simulate the wind tunnel test performed at a Reynolds number of 500,000. We believe that we will require spatial resolution in the stream-wise and span wise direction on the order of 20 y^+. Additionally, we need to perform the simulation for at least 10 flow through periods in order to collect sufficient statistical data. As a result we estimate requiring on the order of 10^9 degrees of freedom with 10^9 time steps at CFL 1. Given the enormous spatial and temporal resolution required, numerical efficiency is paramount.

[*] Postdoctoral Fellow, laslo.diosady@nasa.gov

[†] scott.m.murman@nasa.gov



(a) FAITH Model (b) Oil Flow and Pressure Sensitive Paint Visualization

Figure 1. Fundamental Aeronautics Investigation of The Hill (FAITH) model

In this paper we discuss the development of a variational multiscale method for the time-accurate numerical simulation of high Reynolds number wall-bounded turbulent compressible flows. In this work we focus on the numerical infrastructure of the spectral-element method appropriate for high-Reynolds number simulations. Future work will address the closure model.

II. Spatial Discretization

The compressible Navier-Stokes equations are written in conservative form as:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{F}^I - \boldsymbol{F}^V) = 0 \tag{1}$$

The conservative state vector is $\boldsymbol{u} = [\rho, \rho\mathbf{v}, \rho E]$, where ρ is the density, $\mathbf{v}$ the velocity, and E the total energy. The inviscid and viscous fluxes are given, respectively, by:

$$\boldsymbol{F}^I \cdot \boldsymbol{n} = \begin{bmatrix} \rho v_n \\ \rho \mathbf{v} v_n + p\boldsymbol{n} \\ \rho H v_n \end{bmatrix}, \qquad v_n \;=\; \mathbf{v} \cdot \boldsymbol{n}, \qquad \text{and} \qquad \boldsymbol{F}^V = \begin{bmatrix} 0 \\ \boldsymbol{\tau} \\ \mathbf{v} \cdot \boldsymbol{\tau} + \kappa_T \nabla T \end{bmatrix}, \tag{2}$$

where p is the static pressure, $H = E + \frac{p}{\rho}$ is the total enthalpy, $\boldsymbol{\tau}$ is the shear stress tensor, κ_T is the thermal conductivity, $T = p/\rho R$ is the temperature, and R is the gas constant. The pressure is given by:

$$p = (\gamma - 1)\left(\rho E - \tfrac{1}{2}\rho \mathbf{v} \cdot \mathbf{v}\right), \tag{3}$$

where γ is the specific heat ratio. The viscous stress tensor, $\boldsymbol{\tau}$, is given by:

$$\boldsymbol{\tau} = \mu\left(\nabla \mathbf{v} + \nabla \mathbf{v}^T\right) - \lambda(\nabla \cdot \mathbf{v})I \tag{4}$$

where μ is the viscosity, $\lambda = \frac{2}{3}\mu$ is the bulk viscosity and I is the identity matrix.

In this work we use a Galerkin finite-element method, which extends to arbitrary order of accuracy. Higher-order methods show potential for simulations requiring high spatial and temporal resolution, allowing for solutions with fewer degrees of freedom and lower computational cost than traditional second order CFD methods.[10] Higher-order finite element methods are particularly attractive due to the possibility of using local h- and p-adaptation. Galerkin finite-element methods have an increased number of operations, relative to high-order finite-difference methods, due to the increased coupling between degrees of freedom. High-order finite-difference schemes couple degrees of freedom along lines in each coordinate direction, so that the cost of a residual evaluation scales as $N \times d$ for a fixed number of degrees of freedom, where $N = p + 1$ is the order of accuracy and d the spatial dimension. A general Galerkin method couples all degrees of freedom on an element, leading to a cost that scales as N^d. This scaling suggests that in three dimensions, a general Galerkin formulation is penalized in terms of operation count relative to finite-difference methods beyond $p = 2$ or 3. In order to attain similar efficiency within a Galerkin framework, we use tensor product bases and a sum factorization approach, such that differentiation and integration reduce to a sequence of one-dimensional operations. This results in a cost which also scales as $N \times d$.

The domain, Ω, is partitioned into non-overlapping hexahedral elements κ. We define a finite-element space $\mathcal{V}_h$ consisting of piece-wise polynomial functions on each element κ:

$$\mathcal{V}_h = \{\boldsymbol{w}, \boldsymbol{w}|_\kappa \in [\mathcal{P}(\kappa)]^m\} \tag{5}$$

where $m = 5$ is the number of flow equations. Choosing $\mathcal{V}_h$ to be C^0-continuous across elements leads to a continuous Galerkin (CG) discretization, while allowing $\mathcal{V}_h$ to be discontinuous across elements leads to a discontinuous Galerkin (DG) discretization. We have developed a generic finite-element framework using both CG and DG formulations. However, the results presented in this work focus on DG.

We seek a solution $\boldsymbol{u} \in \mathcal{V}_h$ which satisfies the weak form of Eq. (1):

$$\sum_\kappa \left\{ \int_\kappa \left(\boldsymbol{w}\frac{\partial \boldsymbol{u}}{\partial t} - \nabla \boldsymbol{w} \cdot (\hat{\boldsymbol{F}}^I - \hat{\boldsymbol{F}}^V) \right) + \int_{\partial\kappa} \boldsymbol{w}(\hat{\boldsymbol{F}}^I - \hat{\boldsymbol{F}}^V) \cdot \boldsymbol{n} \right\} = 0 \qquad \forall \boldsymbol{w} \in \mathcal{V}_h. \tag{6}$$

Here $\hat{\boldsymbol{F}}^I$ and $\hat{\boldsymbol{F}}^V$ denote numerical fluxes which are functions of the state on both sides of a face shared by two elements, or state and boundary data for a face on the domain boundary. The inviscid flux is computed using the Roe flux,[11] while the viscous flux is computed using the method of Bassi and Rebay.[12] In a CG method, the surface integrals corresponding to faces on the interior of the domain telescope to zero, leaving only surface integral on the boundary of the domain.

Using a tensor product bases on each element the solution $\boldsymbol{u}$ is given by a product of Lagrange polynomials:

$$\boldsymbol{u}(\mathbf{x}(\xi)) = \sum_{i,j,k} \boldsymbol{U}_{ijk}\Phi_{ijk} \qquad \Phi_{ijk} = \phi_i(\xi_1)\phi_j(\xi_2)\phi_k(\xi_3) \tag{7}$$

where $\mathbf{x}(\xi)$ defines a mapping from the reference cube, $\xi \subset [-1,1]^3$, to physical space. ϕ_i is a one-dimensional Lagrange basis defined at Gauss-Legendre (GL) or Gauss-Legendre-Lobatto (GLL) points, while $\boldsymbol{U}_{ijk}$ is the corresponding nodal value of the solution. The integrals in Eq. (6) are evaluated using numerical quadrature. For volume integrals:

$$\int_\kappa \left(\boldsymbol{w}\frac{\partial \boldsymbol{u}}{\partial t} - \nabla \boldsymbol{w} \cdot (\boldsymbol{F}^I - \boldsymbol{F}^V) \right) \;\simeq\; \sum_{p,q,r} \left\{ \left(\boldsymbol{w}\frac{\partial \boldsymbol{u}}{\partial t} - \nabla_\xi \boldsymbol{w} \cdot (\tilde{\boldsymbol{F}}^I - \tilde{\boldsymbol{F}}^V) \right) |J| \right\}_{\xi_p\xi_q\xi_r} w_p w_q w_r \tag{8}$$

where ξ_p, ξ_q, ξ_r are one-dimensional GL or GLL quadrature points, while w_p, w_q and w_r are the associated quadrature weights. J denotes the Jacobian of the mapping from element reference space to physical space, ∇_ξ denotes the gradient with respect to the local coordinate ξ, while $\tilde{\boldsymbol{F}}^I = J^{-1}\boldsymbol{F}^I$ and $\tilde{\boldsymbol{F}}^V = J^{-1}\boldsymbol{F}^V$ are the fluxes mapped to the local element coordinate system. Similarly, surface integrals in Eq. (6) are evaluated as:

$$\int_{\partial\kappa} \boldsymbol{w}(\boldsymbol{F}^I - \boldsymbol{F}^V) \cdot \boldsymbol{n} \;\simeq\; \sum_{p,q} \{\boldsymbol{w}(\boldsymbol{F}^I - \boldsymbol{F}^V) \cdot \boldsymbol{n}\}_{\xi_p\xi_q} w_p w_q \tag{9}$$

Both volume and surface integrals (Eqs. (8) and (9)) are performed as a sequence of three steps described below:

1. Evaluation of the state and gradient at the quadrature points
2. Evaluation of the fluxes at the quadrature points
3. Multiplication of the fluxes with the gradient of the basis functions.

We describe the sum-factorization approach used in steps 1 and 3. The state at a particular quadrature point is given by:

$$\hat{\boldsymbol{U}}_{pqr} \;\equiv\; \boldsymbol{u}(\xi_p, \xi_q, \xi_r) = \sum_i \sum_j \sum_k \boldsymbol{U}_{ijk}\phi_i|_{\xi_p}\phi_j|_{\xi_q}\phi_k|_{\xi_q} \tag{10}$$

The state evaluated at all quadrature points within an element is computed using the sum factorization approach as a sequence of matrix multiplications corresponding to each coordinate direction:[13]

$$\hat{\boldsymbol{U}} \;=\; (B_3 \otimes B_2 \otimes B_1)\boldsymbol{U} = (B_3 \otimes I \otimes I)\{(I \otimes B_2 \otimes I)\{(I \otimes I \otimes B_1)\boldsymbol{U}\}\} \tag{11}$$

where B_1, B_2 and B_3 are $N_q \times N$ matrices where N_q is the number of 1-dimensional quadrature points, while N is the number of 1-dimensional basis functions. The entries of B_1, B_2 and B_3 are:

$$[B_1]_{pi} = \phi_i|_{\xi_p} \qquad [B_2]_{qj} = \phi_j|_{\xi_q} \qquad [B_3]_{rk} = \phi_k|_{\xi_r} \tag{12}$$

The derivative of the state with respect to ξ_1 is computed at the quadrature points as:

$$\frac{\partial \hat{\boldsymbol{U}}}{\partial \xi_1} \quad = \quad (B_3 \otimes B_2 \otimes D_1)\boldsymbol{U} = (B_3 \otimes I \otimes I)\{(I \otimes B_2 \otimes I)\{(I \otimes I \otimes D_1)\boldsymbol{U}\}\} \tag{13}$$

where

$$[D_1]_{pi} = \frac{\partial \phi_i}{\partial \xi}|_{\xi_p} \tag{14}$$

Derivatives in the ξ_2 and ξ_3 directions are computed in a similar manner.

The use of the sum factorization approach leads to an operation count which scales as $N \times d$ for fixed number of degrees of freedom. However, in order to further reduce the cost of our finite-element discretization we use a collocation approach to compute the integrals involved in the residual. When using a collocation approach, the solution points are used as quadrature points reducing the cost of the state and residual evaluations since operations involving the matrices B_1, B_2 and B_3 are eliminated (B_1, B_2 and B_3 reduce to the identity matrix, and hence multiplication by these terms is not necessary). The state is stored at the collocation points and thus is directly available, while the derivatives are computed using terms of the form:

$$\frac{\partial \hat{\boldsymbol{U}}}{\partial \xi_1} \quad = \quad (I \otimes I \otimes D_1)\boldsymbol{U} \tag{15}$$

The third step of the residual evaluation, requiring the weighting of the flux with the gradient of the basis functions, is computed in a similar manner. Namely, we compute terms of the form:

$$\boldsymbol{R}_\kappa \quad = \quad -(I \otimes I \otimes D_1)^T \hat{\boldsymbol{F}}_1 - (I \otimes D_2 \otimes I)^T \hat{\boldsymbol{F}}_2 - (D_3 \otimes I \otimes I)^T \hat{\boldsymbol{F}}_3 \tag{16}$$

where

$$\left[\hat{\boldsymbol{F}}_1\right]_{pqr} \quad = \quad \left(\tilde{\boldsymbol{F}}_1 |J|\right)_{\xi_p \xi_q \xi_r} w_p w_q w_r \tag{17}$$

while $\hat{\boldsymbol{F}}_1$ and $\hat{\boldsymbol{F}}_2$ have similar forms.

With the use of collocation and tensor products we have reduced the amount of computational work to a minimum, however we still have a high operation count per degree of freedom. We seek to minimize this cost through optimization. The efficiency of a particular algorithm is dependent both upon the under lying numerical scheme as well as on an efficient implementation on modern computing architectures. Recently, increases in computational performance have been largely achieved through increased parallelization, both through the use of systems with increasing number of processors, as well as on-processor parallelization through threading and vectorization. The finite-element framework developed here is designed for use with the computing resources at NASA Ames Research Center. In particular, we look to use the Intel Sandy Bridge and Intel Xeon Phi microprocessor architectures as well as their future descendents, which achieve high-performance through single-instruction/multiple-data (SIMD) vector operations. The Sandy Bridge architecture uses the Advanced Vector Extensions (AVX) instruction set with a SIMD register of 256 bits (4 doubles), while the Xeon Phi has a SIMD register of 512 bits (8 doubles) allowing 4 or 8 double operations to be performed in parallel. In order to take full advantage of the available computing resources, it is essential to use the vectorizing capabilities of these micro architectures. In particular, efficient vectorization requires alignment of data and unit-stride memory operations.

We recognize that high-order finite-element methods are well suited to take advantage of modern day computing architectures. Residual evaluations involve a large number of element-wise local operations which can be parallelized over many cores. Additionally, element-wise local operations may be designed in such a way as to align data in a format well suited to vectorization. In order to achieve high computational efficiency we have optimized the numerical kernels involved in the computation of the residual and attempt to reuse these kernels throughout our finite-element software. In particular, we use a set of optimized matrix-matrix multiplication routines for computing the sum-factorization terms in Eqs. (11), (14) and (16), while the evaluation of fluxes at collocation points is performed using data-aligned, unit-stride vector functions.

The use of the sum-factorization approach and optimized routines allow for residual evaluations with a computational cost that is nearly independent of N for $N = [2, 16]$. The CPU time for a residual evaluation, normalized by the number of degrees of freedom, for our DG discretization is shown in Fig. 2. The CPU time is obtained by performing a simulation of the Taylor-Green vortex problem describe in Section **??** on a Sandy Bridge node of Pleiades supercomputer at NASA Ames Research Center. Each Sandy Bridge node consists of 2 eight-core Intel Xeon E5-2670 processors with a clock speed of 2.6Ghz and 2GB per core memory. The simulations were performed with approximately 30000 degrees of freedom per core. Through optimized linear algebra kernels, the increase in operation count with N is offset by more effective use of computing resources. For $N = [4, 16]$, the CPU time for a single residual evaluation is approximately $0.75\mu s/DOF$ which is comparable to the cost for a residual evaluation using the OVERFLOW finite-difference code on the same architecture.

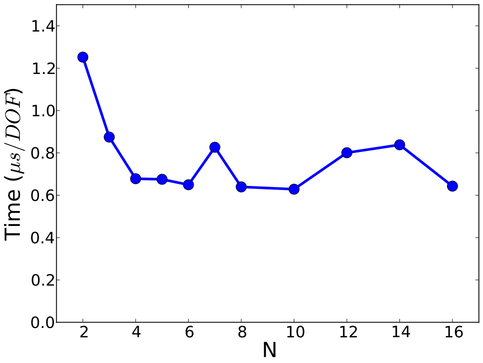


Figure 2. CPU time for Residual Evaluation per Degree of Freedom

III. Temporal Discretization

III.A. Explicit Time-Stepping

Initially we considered an explicit time-stepping scheme using the method of lines. The semi-discrete form of Eq. (6), is written as:

$$\mathbb{M}\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{R}(\boldsymbol{U}) = 0 \tag{18}$$

where $\boldsymbol{U}$ is the vector of spatial degrees of freedom, $\boldsymbol{R}(\boldsymbol{U})$ is the spatial residual and $\mathbb{M}$ is the mass matrix:

$$\mathbb{M}_{ij} = \sum_{\kappa} \int_{\kappa} \Phi_i \Phi_j . \tag{19}$$

Equation (18) is a coupled system of ODEs which is solved numerically using an explicit time-stepping scheme. Each stage of an explicit scheme requires inversion of the mass matrix. In general, a CG discretization has a mass-matrix which is globally coupled, while a DG formulation will have an element-wise block-diagonal mass matrix. In either case, the cost of the storing, factoring and inverting the mass matrix is prohibitive, especially at high-order. However, using our collocation approach, the mass-matrix reduces to an easily invertible diagonal matrix for either CG or DG formulations.

The use of GLL collocation to compute the mass matrix is inexact even in the case of elements with constant Jacobians, as the GLL quadrature with N points is exact only for polynomials up to order $2N - 3$. In a CG formulation a GLL basis ensures that degrees of freedom are associated with edges, faces and nodes, while use of a GL basis necessarily couples all degrees of freedom in the entire domain to ensure continuity. On the other hand in a DG formulation, where the degrees of freedom are associated with elements, the use of a GL collocated basis provides increased accuracy with minimal additional cost.

In order to demonstrate the numerical efficiency of different formulations we compute the isentropic convection of a two-dimensional vortex described in Section V.A at a Mach number of 0.5. Figure 3 shows the L_2 error after 10 flow through periods. We use the DG collocation method with a classical 4th order explicit Runge-Kutta time-stepping scheme. For this problem the temporal error is orders of magnitude lower than the spatial error and so the error is determined by the spatial error alone. As shown in Fig. 3, the use of GL quadrature is more accurate than GLL quadrature for a fixed number of degrees of freedom. The mass lumping with GLL quadrature reduces the order of accuracy of the scheme, consistent with the observation in [14], where the error is shown to act as a filter. As such, the effect of mass lumping is more significant at low order. We note that the reduced accuracy of using the GLL formulation is due solely to the mass-lumping and not to the reduced quadrature in the integration of the spatial part of the residual. This was verified by using more accurate quadrature for the spatial residual term, which gave results identical to those using collocation.

As we are primarily concerned with efficiency, it is necessary to consider any additional cost of using GL quadrature versus GLL quadrature. For a single residual evaluation, the cost of GLL quadrature is slightly less than that using GL quadrature. This is due to evaluation of state or basis functions on element faces, which requires interpolation using GL points, while these values are directly available using GLL points. However, for a viscous simulation interpolation of gradients is required (for either GLL or GL quadrature), and the additional cost of interpolating the state at the same time is minimal (on the order of 1-2 % of a residual evaluation). Thus, in terms of a single residual evaluation we view the use of the GL quadrature in a DG discretization as removing the unnecessary filtering of the highest modes in a colocation scheme, at essentially no additional cost.

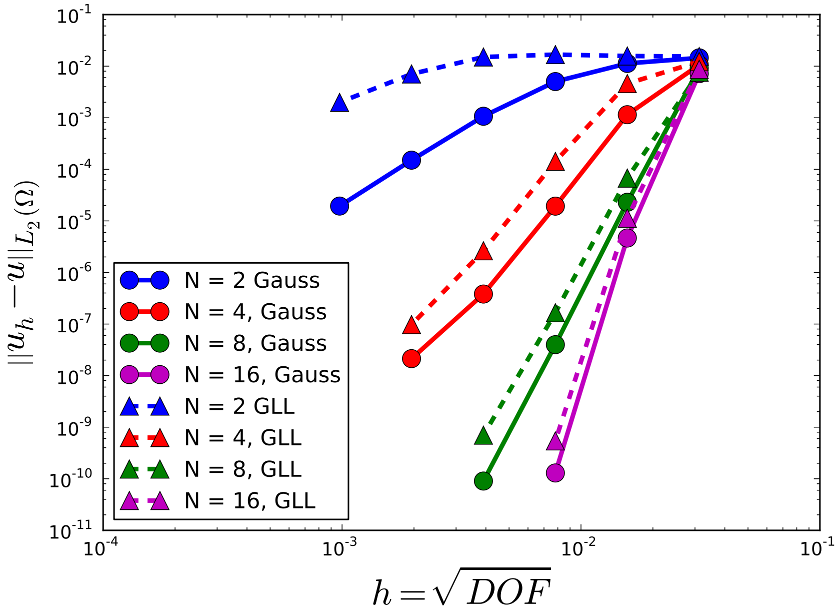


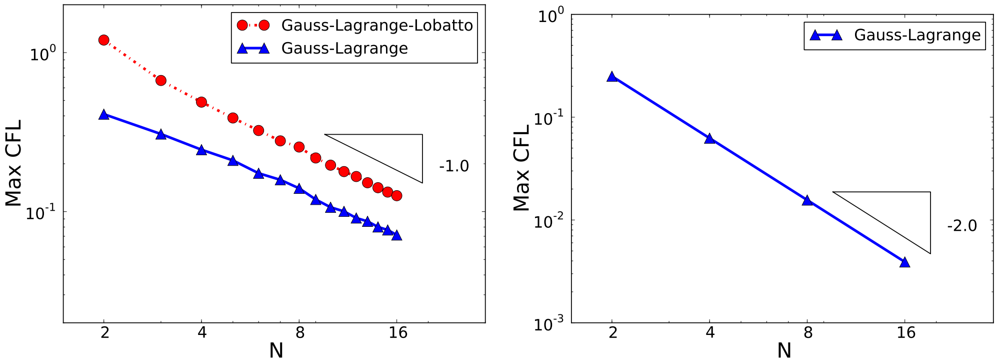
Figure 3. Isentropic Vortex Convection, Error vs. h

III.B. Implicit Time-Stepping

Explicit methods have a maximum time-step which is limited by the CFL condition. For high Reynolds number wall-bounded flows, the stiffness is greatly increased by the resolution required to accurately computed gradients near the wall. While we are interested in computing unsteady flows, and thus our time step is also limited by the desired temporal resolution of the physics of interest, the restriction due to stability may be much more severe, particularly with increasing order of accuracy.

In order to demonstrate the stiffness associated with higher-order methods, we evaluate the maximum stable CFL number using the classical 4th-order explicit Runge-Kutta scheme for two representative flow problems: 1. inviscid isentropic convection of a two-dimensional vortex, and 2. viscous simulation of the Taylor-Green vortex problem. While we are interested in solving viscous flows, the inviscid problem is representative of regions in the flow field far from the wall where viscous effect do not play a significant role. The viscous simulation, performed using a mesh size with cell Reynolds number, $Re_h = O(1)$, is representative of near wall regions of the flow field where fine mesh resolution is required to resolve steep gradients in the flow field.

Figure 4 plots the maximum stable CFL number versus solution order N. We plot the CFL number based on the acoustic speed $CFL = \frac{c\Delta t}{h}$, where Δt is the time step, $h = \sqrt{DOF}$ is the resolution length scale and c the free-stream speed of sound. For the inviscid isentropic convection problem, the maximum allowable time step scales as $1/N$ for a fixed number of degrees of freedom. For the Taylor-Green vortex problem with cell Reynolds number $Re_h \equiv \frac{ch}{\nu} = O(1)$, the maximum stable time step scales as $1/N^2$ for a fixed number of degrees of freedom. The scaling of the maximum allowable time step with N implies an increase in the number of time steps (and hence cost) which scales linearly for inviscid problems and quadratically for viscous problems as we increase the spatial resolution via increasing p.



(a) Inviscid

(b) Viscous

Figure 4. Maximum CFL vs. solution order for the inviscid isentropic vortex convection problem (left) and the viscous Taylor-Green vortex problem at $Re_h = O(1)$ (right).

Implicit methods have the possibility of being more efficient than explicit methods, particularly for higher-order methods. Implicit methods require the solution of a globally coupled nonlinear problem at each iteration. In order for implicit methods to be competitive, the additional cost of solving this globally coupled problem must be offset by a sufficient decrease in the computational work relative to explicit methods. As described in the introduction, the memory required for storing the linearization of the residual is prohibitive for the class of problems we are considering. We choose a scheme with memory requirements similar to an explicit scheme, i.e. we trade the memory for increased operations which can be hidden via optimization and algorithm choice. As such, we consider Jacobian-free Newton-Krylov methods[15] for the solution of the nonlinear problem arising at each iteration,

$$\boldsymbol{R}^*(\boldsymbol{U}^{n+1}) \quad = \quad 0 \tag{20}$$

where $\boldsymbol{R}^*(\boldsymbol{U})$ is the unsteady residual. In a Newton-Krylov method, Eq. (20) is solved using a Newton iteration, where each Newton step requires the solution of a linear system:

$$\frac{\partial \boldsymbol{R}^*}{\partial \boldsymbol{U}} \Delta \boldsymbol{U}^k \quad = \quad -\boldsymbol{R}^*(\boldsymbol{U}^k) \tag{21}$$

The linear system, Eq. (21), is solved using a Krylov method. In this work, we use a preconditioned Generalized Minimal Residual (GMRES) algorithm as our Krylov method.[16] As noted previously, the storage of the Jacobian $\frac{\partial \boldsymbol{R}^*}{\partial \boldsymbol{U}}$ is not feasible. In a Krylov method, the matrix $\frac{\partial \boldsymbol{R}^*}{\partial \boldsymbol{U}}$ is not required; we only require the application of $\frac{\partial \boldsymbol{R}^*}{\partial \boldsymbol{U}}$ to a vector $\boldsymbol{V}$, i.e. we need to compute the Frechet derivative of $\boldsymbol{R}^*$ in the GMRES search direction.

In this work we consider two approaches for computing the Frechet derivative. The first approach, is to approximate the linearization using finite differences:

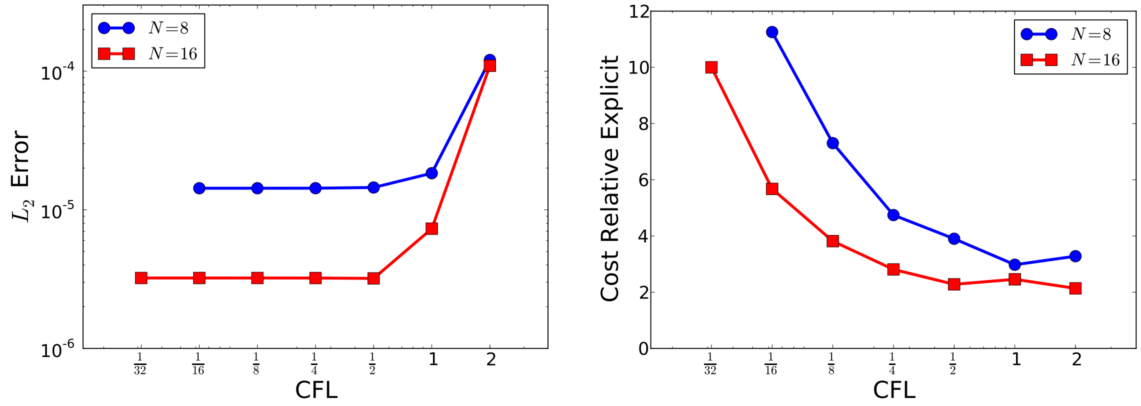
$$\frac{\partial \boldsymbol{R}^*}{\partial \boldsymbol{U}}\boldsymbol{V} \simeq \frac{\boldsymbol{R}^*(\boldsymbol{U}+\epsilon\boldsymbol{V}) - \boldsymbol{R}^*(\boldsymbol{U})}{\epsilon} \tag{22}$$

with suitable choise of step size ϵ.[17] This approach allows the computation of a Frechet derivative for the cost of a single residual evaluation. A particular advantage of this approach is that the approximate Frechet derivative is available once the residual has been computed, and requires no linearization of residual terms. This is particularly advantageous for turbulence models, as hand linearization of various modeling terms is not required. On the other hand, the finite-difference computation of the Frechet derivative is sensitive to the step length ϵ and inaccuracies in the Frechet derivative degrade the convergence of the Newton solver.[15]

An alternative approach is to compute the exact Frechet derivative. This approach is more expensive than the finite-difference approach, however, the additional cost eliminates the dependence on ϵ, and is consistent with the solution of adjoint (dual) problems. In computing the exact Frechet derivative, we are able to reuse the optimized kernels developed for computing the sum factorization, though we require the additional hand linearization of the fluxes. We have found that in our implementation the cost of the exact Frechet derivative is approximately 50% more than a residual evaluation, though some of this cost is offset by improved convergence.

We demonstrate the potential benefits of an implicit method by solving the two test problems described at the beginning of this section using an implicit method. We use a 4th order accurate 5-stage diagonally-implicit Runge-Kutta (DIRK) scheme as our time-stepping scheme. At each stage, the Newton-Krylov algorithm is used to reduce the unsteady residual by 10 orders of magnitude, where at each Newton step the GMRES method is run until the linear residual has been reduced by 6 orders of magnitude or a maximum of 20 Krylov vectors are used. We use the mass-matrix as the preconditioner to GMRES.

We solve the convecting vortex problem at a Mach number of 0.5 using 8th- ($p = 7$) and 16th-order ($p = 15$) spatial discretizations using 64 degrees of freedom in both x- and y- directions.



(a) Error vs. CFL

(b) Cost vs. CFL

Figure 5. Error convergence and residual evaluations using diagonally-implicit Runge-Kutta scheme for the isentropic vortex convection problem using $N = 8$ ($p = 7$) with $h = 1/\sqrt{DOF} = 1/64$.

Figure 5(a) plots the L_2 error after 5 periods versus the CFL number, where $CFL = \frac{c\Delta t}{h}$ and $h = \sqrt{DOF} = 1/64$. The smallest CFL for each solution order corresponds to the maximum stable time step using the classical 4-stage explicit Runge-Kutta method. At this CFL the spatial error dominates the temporal error, as the maximum time step required for stability is much smaller than that needed to resolve the unsteady features of the flow. Using the implicit scheme we are able to use time steps up to an order of magnitude larger than the explicit scheme without increasing the error. For $CFL > 1$, the temporal error introduced by increasing the time step becomes dominant, suggesting an upper bound on the usable time step, due to the problem's physics.

Figure 5 plots the corresponding computational cost relative to the explicit scheme. The cost reported is the number of residual and Frechet derivative evaluations used for the implicit scheme, divided by the number of residual evaluations required for the explicit scheme. Using the same CFL as the explicit scheme, the implicit scheme requires approximately 11 times as many residual evaluations as the comparable explicit method. As the CFL number increases fewer residual/Frechet evaluations are necessary. At $CFL = 1$, we require 3-4 times as many residual evaluations using the explicit method. This implies a CPU time for the implicit method approximately 3-6 times that of the explicit method depending upon whether finite-difference or exact Frechet differentiation is used.

For the inviscid test problem the explicit method is more efficient than the implicit method as the time-step restriction due to stability requirements is not significantly smaller than that required to resolve the unsteady features of the flow. For viscous problems, implicit methods have the potential to be much more efficient than explicit methods due to the increased stiffness. We solve the viscous Taylor-Green vortex problem at $M = 0.1$ and $Re = 16$ using 8th- ($p = 7$) and 16th-order ($p = 15$) spatial discretizations using 64 degrees of freedom in each coordinate direction. Figure 6 plots the cost of the Taylor-Green simulation relative to using an explicit scheme. At $CFL \approx 1$, the implicit method is more efficient than the explicit method, requiring 25-50% of the number of residual evaluations for the explicit scheme.

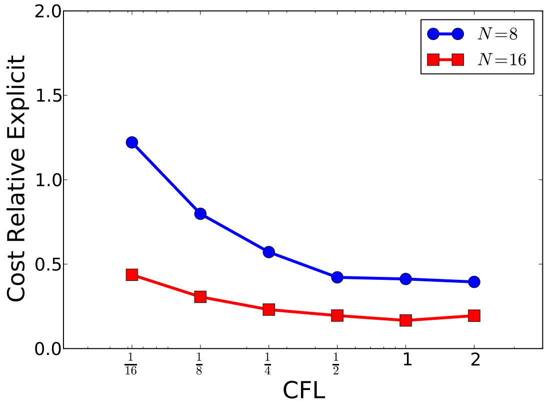


Figure 6. Computational cost of using diagonally-implicit Runge-Kutta method relative to explicit Runge-Kutta method for the Taylor-Green vortex problem ($M = 0.1$, $Re = 16$) using $N = 8$ ($p = 7$) and $N = 16$ ($p = 15$) with $h = 1/\sqrt{DOF} = 1/64$.

III.C. Preconditioning

Preconditioning is necessary in order of obtain good performance in a Newton-Krylov scheme.[15] An effective preconditioner approximates the inverse of the Jacobian matrix and has the effect of clustering the eigenvalues of the preconditioned system, allowing for rapid convergence of the Krylov scheme.[15] We consider preconditioning methods with similar memory requirements as our explicit methods. To this end, we consider an element-wise block-Jacobi preconditioner where the elemental blocks are solved approximately using an alternating-direction-implicit (ADI) scheme.[18] We use a diagonalized-ADI scheme which has previously been used in finite-difference simulations,[19] but not to our knowledge in finite/spectral-element methods.

We derive the diagonalized-ADI scheme by considering a constant coefficient problem:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{A}\boldsymbol{u}) \quad = \quad 0 \tag{23}$$

Applying a discontinuous Galerkin discretization with upwind flux, we have;

$$\sum_{\kappa} \left\{ \int_{\kappa} \left\{ \boldsymbol{w} \frac{\partial \boldsymbol{u}}{\partial t} - \nabla \boldsymbol{w} \cdot \boldsymbol{A}\boldsymbol{u} \right\} + \int_{\partial\kappa} \boldsymbol{w} \widehat{\boldsymbol{A}_n \boldsymbol{u}} \right\} \quad = \quad 0 \tag{24}$$

where $\widehat{\boldsymbol{A}_n \boldsymbol{u}}$ is the upwind flux evaluated as:

$$\widehat{\boldsymbol{A}_n \boldsymbol{u}} \quad = \quad \tfrac{1}{2}\boldsymbol{A}_n(\boldsymbol{u}^+ + \boldsymbol{u}^-) + \tfrac{1}{2}|\boldsymbol{A}_n|(\boldsymbol{u}^+ - \boldsymbol{u}^-) = \boldsymbol{A}_n^+ \boldsymbol{u}^+ + \boldsymbol{A}_n^- \boldsymbol{u}^- \tag{25}$$

where $\boldsymbol{A}_n = \boldsymbol{A} \cdot \boldsymbol{n}$, while $\boldsymbol{u}_+$ and $\boldsymbol{u}_-$ denote, respectively, trace values taken from inside or outside the element κ. Applying a backward Euler time integration scheme, the contribution to Eq. (24) due to $\boldsymbol{w}$ and

$\boldsymbol{u}$ restricted to an element κ given by:

$$\boldsymbol{r}^*_\kappa(\boldsymbol{w}|_\kappa, \boldsymbol{u}|_\kappa) \quad \equiv \quad \int_\kappa \left\{ \boldsymbol{w} \frac{1}{\Delta t} \boldsymbol{u}^{n+1} - \nabla \boldsymbol{w} \cdot \boldsymbol{A} \boldsymbol{u}^{n+1} \right\} + \int_{\partial\kappa} \boldsymbol{w} \boldsymbol{A}^+_n \boldsymbol{u}^{n+1} \tag{26}$$

In Eq. (26), $\boldsymbol{r}^*_\kappa(\boldsymbol{w}|_\kappa, \boldsymbol{u}|_\kappa)$ corresponds to an element-wise block diagonal entry of the Jacobian $\frac{\partial \boldsymbol{R}^*}{\partial \boldsymbol{U}}$. Using a tensor product basis in two-dimensions we write $\boldsymbol{u} = \boldsymbol{U}_{ij} \phi_i(\xi_1)\phi_j(\xi_2)$ and $\boldsymbol{w} = \phi_m(\xi_1)\phi_n(\xi_2)$. Integrating in the reference element, the corresponding discrete form is given by:

$$\begin{aligned}
\frac{\partial \boldsymbol{R}^*_{mn}}{\partial \boldsymbol{U}_{ij}} \quad = \quad & \int_{\xi_1} \int_{\xi_2} \left\{ \phi_m \phi_n \frac{1}{\Delta t} \phi_i \phi_j - \frac{\partial \phi_m}{\partial \xi_1} \phi_n \tilde{\boldsymbol{A}}_1 \phi_i \phi_j - \phi_m \frac{\partial \phi_n}{\partial \xi_2} \tilde{\boldsymbol{A}}_2 \phi_i \phi_j \right\} |J| \\
& + \int_{\xi_2} \left\{ \phi_m \phi_n \tilde{\boldsymbol{A}}^-_1 \phi_i \phi_j |J| \right\}_{\xi_1 = -1} + \int_{\xi_1} \left\{ \phi_m \phi_n \tilde{\boldsymbol{A}}^-_2 \phi_i \phi_j |J| \right\}_{\xi_1 = -1} \\
& + \int_{\xi_2} \left\{ \phi_m \phi_n \tilde{\boldsymbol{A}}^+_1 \phi_i \phi_j |J| \right\}_{\xi_1 = 1} + \int_{\xi_1} \left\{ \phi_m \phi_n \tilde{\boldsymbol{A}}^+_2 \phi_i \phi_j |J| \right\}_{\xi_1 = 1}
\end{aligned} \tag{27}$$

where $\tilde{\boldsymbol{A}} = J^{-1}\boldsymbol{A}$, $\tilde{\boldsymbol{A}}^\pm_1 = \frac{1}{2}(\tilde{\boldsymbol{A}}_1 \pm |\tilde{\boldsymbol{A}}_1|)$ and $\tilde{\boldsymbol{A}}^\pm_2 = \frac{1}{2}(\tilde{\boldsymbol{A}}_2 \pm |\tilde{\boldsymbol{A}}_2|)$. Equation (27) is written in matrix form as:

$$\begin{aligned}
\frac{1}{|J|} \frac{\partial \boldsymbol{R}^*_\kappa}{\partial \boldsymbol{U}_\kappa} \quad = \quad & \left(\mathbb{M}_1 \otimes \frac{I}{\Delta t} \otimes \mathbb{M}_2 \right) - \left(\mathbb{D}^I_1 \otimes \tilde{\boldsymbol{A}}_1 \otimes \mathbb{M}_2 \right) - \left(\mathbb{M}_1 \otimes \tilde{\boldsymbol{A}}_2 \otimes \mathbb{D}^I_2 \right) \\
& + \left(\mathbb{D}^{B-}_1 \otimes \tilde{\boldsymbol{A}}^-_1 \otimes \mathbb{M}_2 \right) + \left(\mathbb{M}_1 \otimes \tilde{\boldsymbol{A}}^-_2 \otimes \mathbb{D}^{B-}_2 \right) + \left(\mathbb{D}^{B+}_1 \otimes \tilde{\boldsymbol{A}}^+_1 \otimes \mathbb{M}_2 \right) + \left(\mathbb{M}_1 \otimes \tilde{\boldsymbol{A}}^+_2 \otimes \mathbb{D}^{B+}_2 \right)
\end{aligned} \tag{28}$$

where

$$\mathbb{M}_1 = \int_{\xi_1} \phi_m \phi_i \qquad \mathbb{D}^I_1 = \int_{\xi_1} \frac{\partial \phi_m}{\partial \xi_1} \phi_i \qquad \mathbb{D}^{B\pm}_1 = \{\phi_m \phi_i\}_{\xi_1 = \pm 1} \tag{29}$$

$$\mathbb{M}_2 = \int_{\xi_2} \phi_n \phi_j \qquad \mathbb{D}^I_2 = \int_{\xi_2} \frac{\partial \phi_n}{\partial \xi_2} \phi_j \qquad \mathbb{D}^{B\pm}_2 = \{\phi_n \phi_j\}_{\xi_2 = \pm 1} \tag{30}$$

In Eq. (28), we have assumed that the Jacobian of the mapping from element reference space to physical space is constant. The non-constant Jacobian case is addressed below. In order to clarify the derivation of our scheme, we define operators $\mathbb{D}_1$ and $\mathbb{D}_2$ which correspond to the action of interior and boundary convection operators in the ξ_1 and ξ_2 directions, respectively, such that we may rewrite Eq. (28) as:

$$\frac{1}{|J|} \frac{\partial \boldsymbol{R}^*_\kappa}{\partial \boldsymbol{U}_\kappa} \quad \equiv \quad \left(\mathbb{M}_1 \otimes \frac{I}{\Delta t} \otimes \mathbb{M}_2 \right) - \left(\mathbb{D}_1 \otimes \tilde{\boldsymbol{A}}_1 \otimes \mathbb{M}_2 \right) - \left(\mathbb{M}_1 \otimes \tilde{\boldsymbol{A}}_2 \otimes \mathbb{D}_2 \right) \tag{31}$$

$$\begin{aligned}
= \quad & \left[\left(\mathbb{M}_1 \otimes \frac{I}{\Delta t} \otimes \mathbb{M}_2 \right) - \left(\mathbb{D}_1 \otimes \tilde{\boldsymbol{A}}_1 \otimes \mathbb{M}_2 \right) \right] \left(\mathbb{M}_1 \otimes \frac{I}{\Delta t} \otimes \mathbb{M}_2 \right)^{-1} \left[\left(\mathbb{M}_1 \otimes \frac{I}{\Delta t} \otimes \mathbb{M}_2 \right) - \left(\mathbb{M}_1 \otimes \tilde{\boldsymbol{A}}_2 \otimes \mathbb{D}_2 \right) \right] \\
& - \left(\mathbb{D}_1 \otimes \tilde{\boldsymbol{A}}_1 \otimes \mathbb{M}_2 \right) \left(\mathbb{M}_1 \otimes \frac{I}{\Delta t} \otimes \mathbb{M}_2 \right)^{-1} \left(\mathbb{M}_1 \otimes \tilde{\boldsymbol{A}}_2 \otimes \mathbb{D}_2 \right)
\end{aligned} \tag{32}$$

Following [18] the second term in Eq. (32) is omitted resulting in the approximate inverse:

$$\begin{aligned}
|J| \left(\frac{\partial \boldsymbol{R}^*_\kappa}{\partial \boldsymbol{U}_\kappa} \right)^{-1} \quad \simeq \quad & \left[\left(\mathbb{M}_1 \otimes \frac{I}{\Delta t} \otimes \mathbb{M}_2 \right) - \left(\mathbb{D}_1 \otimes \tilde{\boldsymbol{A}}_1 \otimes \mathbb{M}_2 \right) \right]^{-1} \left(\mathbb{M}_1 \otimes \frac{I}{\Delta t} \otimes \mathbb{M}_2 \right) \times \\
& \left[\left(\mathbb{M}_1 \otimes \frac{I}{\Delta t} \otimes \mathbb{M}_2 \right) - \left(\mathbb{M}_1 \otimes \tilde{\boldsymbol{A}}_2 \otimes \mathbb{D}_2 \right) \right]^{-1}
\end{aligned} \tag{33}$$

$$\begin{aligned}
= \quad & \left\{ \left[\left(\mathbb{M}_1 \otimes \frac{I}{\Delta t} \right) - \left(\mathbb{D}_1 \otimes \tilde{\boldsymbol{A}}_1 \right) \right] \otimes I \right\}^{-1} \left(I \otimes \frac{I}{\Delta t} \otimes I \right) \times \\
& \left\{ I \otimes \left[\left(\frac{I}{\Delta t} \otimes \mathbb{M}_2 \right) - \left(\tilde{\boldsymbol{A}}_2 \otimes \mathbb{D}_2 \right) \right] \right\}^{-1}
\end{aligned} \tag{34}$$

Here we have factored terms in Eq. (34) to highlight that the inverses correspond to the solution of one-dimensional problems in either ξ_1 or ξ_2 directions. The application of the ADI scheme can be computed using the following sequence of steps:

1. Solve N independent one-dimensional problems in the ξ_1 direction.
2. Scale by the time-step.
3. Solve N independent one-dimensional problems in the ξ_2 direction.

The one-dimensional problems in steps 1 and 3 have m coupled flow equations. Following Pulliam and Chaussee[19] a diagonal form of Eq. (34) is obtained using an eigenvalue decomposition of $\tilde{\boldsymbol{A}}$. We write Eq. (34) as:

$$\begin{aligned}
|J|\left(\frac{\partial \boldsymbol{R}^*_\kappa}{\partial \boldsymbol{U}_\kappa}\right)^{-1} &\simeq \left\{\left[\left(\mathbb{M}_1 \otimes \frac{I}{\Delta t}\right) - \left(\mathbb{D}_1 \otimes \boldsymbol{T}_1^{-1}\boldsymbol{\Lambda}_1\boldsymbol{T}_1\right)\right] \otimes I\right\}^{-1}\left(I \otimes \frac{I}{\Delta t} \otimes I\right) \times \\
&\qquad \left\{I \otimes \left[\left(\frac{I}{\Delta t} \otimes \mathbb{M}_2\right) - \left(\boldsymbol{T}_2^{-1}\boldsymbol{\Lambda}_2\boldsymbol{T}_2 \otimes \mathbb{D}_2\right)\right]\right\}^{-1} \\
&= (I \otimes \boldsymbol{T}_1 \otimes I)\left\{\left[\left(\mathbb{M}_1 \otimes \frac{I}{\Delta t}\right) - (\mathbb{D}_1 \otimes \boldsymbol{\Lambda}_1)\right] \otimes I\right\}^{-1}\left(I \otimes \frac{\boldsymbol{T}_1^{-1}\boldsymbol{T}_2}{\Delta t} \otimes I\right) \times \\
&\qquad \left\{I \otimes \left[\left(\frac{I}{\Delta t} \otimes I\right) - (\boldsymbol{\Lambda}_2 \otimes \mathbb{D}_2)\right]\right\}^{-1}(I \otimes \boldsymbol{T}_2^{-1} \otimes I) \qquad (35)
\end{aligned}$$

where $\boldsymbol{\Lambda}_1$ and $\boldsymbol{\Lambda}_2$ are eigenvalues of $\tilde{\boldsymbol{A}}_1$ and $\tilde{\boldsymbol{A}}_2$ with corresponding eigenvector $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$. The diagonalized ADI scheme, corresponding to the action of the approximate inverse in Eq. (35), is summarized in the following steps:

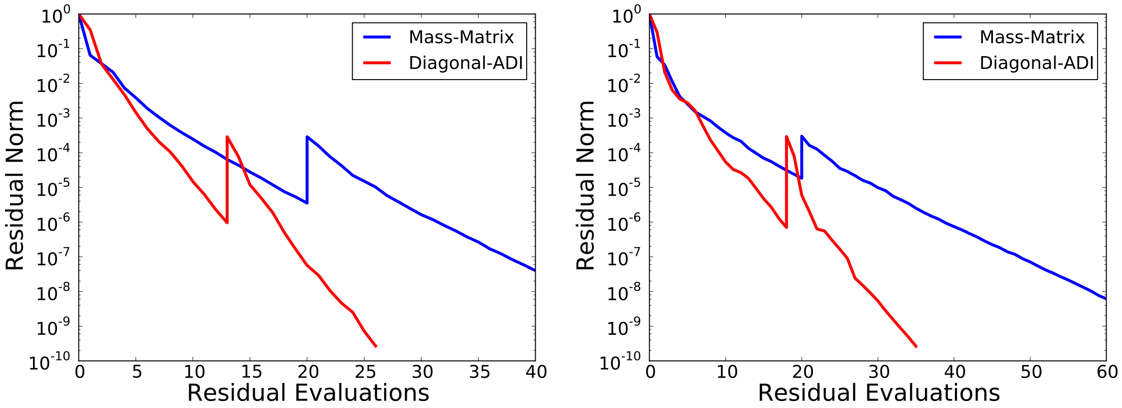
1. Transform to characteristic variables in the ξ_1 direction at each collocation point.
2. Solve $N \times m$ independent scalar one-dimensional advection problems in ξ_1 direction.
3. Transform to characteristic variables in the ξ_2 direction and scale by the time-step.
4. Solve $N \times m$ scalar one-dimensional advection problems in ξ_2 direction.
5. Transform back to original variables

We highlight that in the diagonalized ADI scheme, we solve one-dimensional scalar advection problems, as opposed to a system of equations, which results in a savings without significant loss of convergence rate.

In case of variable coefficients (or variable Jacobian) the linearization, $\frac{\partial \boldsymbol{R}^*}{\partial \boldsymbol{U}}$, can not be expressed simply as the sum of tensor product of matrices. However, we may still apply the steps of the ADI or diagonalized ADI algorithms, recognizing that the one-dimensional problems now have variable coefficients. We note that while for constant coefficient problems the ADI scheme and diagonalized ADI scheme are mathematically equivalent. In the variable coefficient case the diagonalized scheme introduces an additional $O(\Delta t)$ error.[19]

We now apply the diagonalized-ADI scheme for the solution of the isentropic vortex convection problem at a Mach number of 0.5 with $h = \sqrt{DOF} = 1/64$, for 8th- and 16th-order solutions. Figure 7 shows the convergence history for a typical stage of the implicit scheme at $CFL = 1.0$. The linear residual is plotted as a function of the number of residual or Frechet derivative evaluations using both mass-matrix and diagonalized-ADI preconditioning. The diagonalized-ADI scheme reduces the number of residual evaluations required by about a factor of two for this problem. As the cost of applying the diagonalized-ADI scheme is the same as a single residual evaluation, the total CPU time for the diagonalized-ADI scheme is same as using mass-matrix preconditioning. However, we using the diagonalized-ADI scheme requires fewer Krylov vector and hence we are less likely to see stalling of the GMRES algorithm as the complexity of the problem increases.

In order to develop an efficient preconditioner for the viscous dominated case we intend to leverage our finite element framework and pursue spectral/p-multigrid methods. Spectral/p-multigrid methods have been widely used for CG spectral-element discretizations where theoretical and numerical results gives p-independent convergence rates.[20, 21] While no corresponding convergence theory exists for DG discretizations, spectral/p- multigrid approaches have also been widely applied to DG discretizations.[22–25] We have used the approaches previously presented for DG discretizations as the starting point for the development of our spectral/p-multigrid preconditioner, however, at the time of writing we have yet to succeed in developing an efficient spectral/p-multigrid approach for the DG discretization.



(a) $N = 8$ $(p = 7)$

(b) $N = 16$ $(p = 15)$

Figure 7. Residual convergence history for a single Newton-Step using diagonally-implicit Runge-Kutta scheme for the isentropic vortex convection problem with $h = 1/\sqrt{DOF} = 1/64$.

III.D. Space-Time formulation

Finally we consider a fully-space time formulation for our unsteady problem. Space-time formulation has several features which make it especially attractive. First, the use of a space-time formulation enables us to perform adaptation in both spatial and temporal directions, potentially leading to significant reduction in cost. Second, a space-time formulation is a more natural setting for a variational multi-scale formulation. In particular for the use of residual-based variational multiscale methods.

Space-time formulations are generally considered too expensive to be used in practical engineering simulations. Space-time formulations are considered expensive, as they require the solution of a globally coupled system of equations for each time-slab. As we have noted the cost of storing the linearization for a single step of an implicit scheme is prohibitively expensive. Using a space-time formulation this storage cost is scaled by the order of the basis used in the temporal direction. In the previous section we have shown that we can develop an implicit scheme with similar storage requirement as an explicit method using a matrix-free approach. The matrix free approach may be further extended to a space-time formulation such that once again the storage of the Jacobian is not necessary.

We proceed to briefly describe the space-time formulation. Define a time interval (time-slab) $I^n = [t^n, t^{n+1}]$. We define a space-time finite element space $\mathcal{V}_h$ consisting of piece-wise polynomial functions in both space and time on each element κ:

$$\mathcal{V}_h = \{\boldsymbol{w}, \boldsymbol{w}|_\kappa \in [\mathcal{P}(\kappa \times I)]^m\} \tag{36}$$

Once again, if $\mathcal{V}_h$ is C^0 continuous across elements we have a continuous Galerkin formulation in space, otherwise we use a discontinuous Galerkin formulation. We seek a solution $\boldsymbol{u} \in \mathcal{V}_h$ satisfying the weak form of Eq. (1):

$$\begin{aligned}\sum_{\kappa}\Bigg\{\int_I\int_\kappa\left(-\frac{\partial \boldsymbol{w}}{\partial t}\boldsymbol{u}-\nabla\boldsymbol{w}\cdot(\boldsymbol{F}^I-\boldsymbol{F}^V)\right)+\int_I\int_{\partial\kappa}\boldsymbol{w}(\hat{\boldsymbol{F}}^I-\hat{\boldsymbol{F}}^V)\cdot\boldsymbol{n}& \\ +\int_\kappa \boldsymbol{w}(t_-^{n+1})\boldsymbol{u}(t_-^{n+1})-\boldsymbol{w}(t_+^n)\boldsymbol{u}(t_-^n)\Bigg\} &= 0 \qquad \forall\boldsymbol{w}\in\mathcal{V}_h.\end{aligned} \tag{37}$$

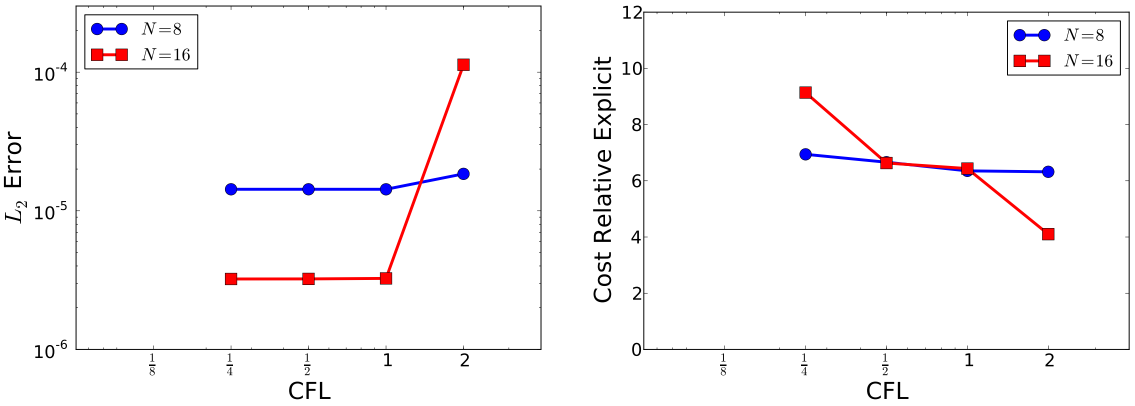
As with our spatial discretization, we use a tensor product basis:

$$\boldsymbol{u}(x(\xi), t(\tau)) \;=\; \sum_{i,j,k,l} U_{ijkl}\Phi_{ijkl} \qquad\qquad \Phi_{ijkl} = \phi_i(\xi_1)\phi_j(\xi_2)\phi_k(\xi_3)\phi_l(\tau) \tag{38}$$

where $t(\tau)$ defines a mapping from the reference interval $\tau \in [-1, 1]$ to the interval $I = [t^n, t^{n+1}]$. The integrals in the space-time domain are computed using collocation and the sum-factorization approach, leading to a residual evaluation cost which scales as $N \times (d + 1)$. Equation (37) gives a system of nonlinear equations of the form $\boldsymbol{R}^*(\boldsymbol{U}) = 0$, which must solved for each time slab. For a given spatial discretization, the size of system which must be solved in each time slab in the space-time formulation is N times larger than that for the implicit scheme as the space-time system involve all temporal collocation point in a space-time slab. We note that this results in an increase relative to the implicit scheme in the memory required for the storage of the Krylov vectors in GMRES. However, we are able to afford this cost since we do not store the Jacobian.

The efficiency of our space-time formulation hinges on the ability of the Newton-Krylov scheme to rapidly converges the nonlinear problem arising at the time slab. As with the implicit time-stepping scheme we anticipate that a spectral-multigrid preconditioner will result in a very efficient formulation. In the space-time formulation we are able to use a spectral multigrid approach in both space and time. At this stage in the development of our algorithm we have yet to successfully implement the space-time spectral multigrid algorithm. However, we demonstrate the potential feasibility of using a space-time approach by using the space-time mass-matrix as preconditioner.

Once again we consider the two simple test cases described earlier. We solve the convecting vortex problem at a Mach number of 0.5 using 8th- ($p = 7$) and 16th-order ($p = 15$) spatial discretizations using 64 degrees of freedom in both x- and y- directions. In the temporal direction, we use 8th- ($p = 7$) and 16th-order ($p = 15$) discretizations, respectively. For each time-slab, the Newton-Krylov algorithm is used to reduce the unsteady residual by 10 orders of magnitude. For each Newton iteration we use the preconditioned GMRES method with 20 Krylov vectors.



(a) Error vs. CFL

(b) Residual Evaluations vs. CFL

Figure 8. Error convergence and residual evaluations using a space-time formulation for the isentropic vortex convection problem using $N = 8$ ($p = 7$) in both space and time with $h = 1/\sqrt{DOF} = 1/64$.

Figure 8 plots the L_2 error after 5 flow through periods versus the space-time CFL number, $CFL = \frac{c\Delta t}{h}$, with $t^{n+1} - t^n = N\Delta t$. As in the implicit case, the spatial error dominates the temporal error at low CFL, while the temporal error becomes significant for CFL ¿ 1.

Figure 8 gives the corresponding cost of the space-time formulation in terms of the number of residual evaluations relative to the explicit scheme. In the accounting in Figure 8 the cost of a space-time residual evaluation is estimated to be N times that of a single spatial residual. We see that using simple mass-matrix preconditioner the space-time formulation has a cost which is about 4-7 times that of the explicit method. For this inviscid test-case, the time-step restriction require to ensure stability for the explicit scheme is not sufficiently restrictive so that benefit may be gained by using an implicit space-time approach.

Next we consider the viscous Taylor-Green problem. Once again, we solve the viscous Taylor-Green vortex problem at $M = 0.1$ and $Re = 16$ using 8th- ($p = 7$) and 16th-order ($p = 15$) spatial discretizations using 64 degrees of freedom in each coordinate direction. We use 8th- ($p = 7$) and 16th-order ($p = 15$) solutions respectively in the temporal direction. Figure 9 plots the cost of the Taylor-Green simulation relative to the explicit scheme. At $CFL \approx 1$, the space-time formulation has a cost which is similar to that of the explicit scheme.

Once again, we emphasize that our space-time formulation has similar memory requirements as an explicit scheme, and using a very simple mass-matrix preconditioning has a reduced computational cost for viscous

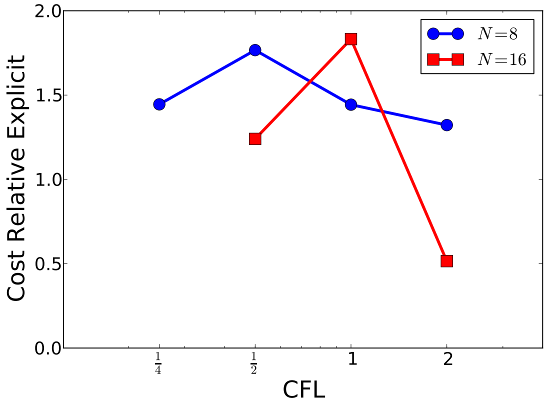


Figure 9. Computational cost of using space-time formulation relative to explicit Runge-Kutta method for the Taylor-Green vortex problem ($M = 0.1$, $Re = 16$) using $N = 8$ ($p = 7$) and $N = 16$ ($p = 15$) with $h = 1/\sqrt{DOF} = 1/64$.

dominated problems. We anticipate that the use of a spectral multigrid approach will further reduce the cost of the space-time approach relative to the explicit scheme.

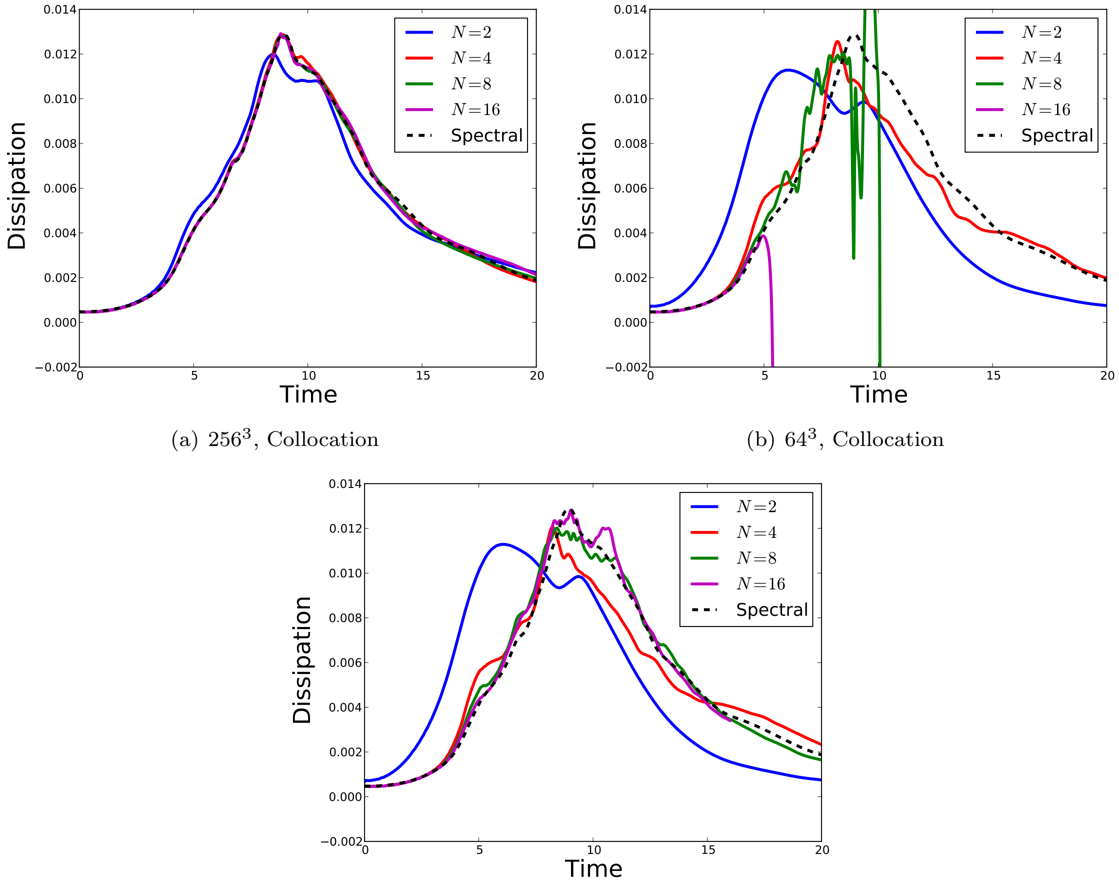
IV. Stabilization

Higher-order numerical methods are attractive for the simulation of unsteady flows due to the low numerical dissipation inherent in these scheme. For the DG discretization numerical stabilization is introduced through the use of upwind fluxes at element boundaries. As the solution order is increased the amount of stabilization introduced is reduced. Unfortunately, with lower numerical dissipation, numerical stability becomes an issue.

In order to reduce the cost of our simulations we have decided to use collocation to compute the volume and surface integrals. The use of numerical quadrature implies inexact projection of the fluxes onto the finite element space. This inexact projection leads to so-called "aliasing" errors. Many different techniques have been proposed for eliminating the instability caused by such aliasing errors such as: filtering, spectrally vanishing viscosity,[26, 27] writing the convective term in a different form[28] or "polynomial dealiasing".[29] In this work we consider "polynomial dealiasing", also known as "over-integration" which attempts to directly address the inaccuracy of the projection by using a better quadrature rule.

For incompressible Navier-Stokes simulations, the nonlinear convection terms are quadratic in the state, implying that for a solution with $N = p+1$ exact projection of the nonlinear terms may be performed using a Gaussian quadrature rule with $3/2N$ points. Other techniques, such as writing the convective terms in skew-symmetric form to ensure discrete conservation of both mass and momentum, also make use of the fact that the incompressible Navier-Stokes equations have a non-linearity which is a quadratic function of the conservative state vector. In the case of the compressible Navier-Stokes equations, the nonlinearity in the inviscid flux is a rational function of the conservative state. Thus, Gaussian quadrature is unable to exactly compute the projection of the Navier-Stokes equations onto the polynomial space. It has been proposed that since the inviscid flux in the compressible Navier-Stokes equations is a cubic function of the primitive variables, using a quadrature rule with $2N$ points is sufficient. This $2N$ rule has been successfully applied to a variety of subsonic turbulent flows.[29, 30]

We demonstrate the aliasing issue by performing a simulation of the Taylor-Green Vortex problem at a $M = 0.1$ and $Re = 1600$. In Figure 10 we plot the kinetic energy dissipation rate $-\frac{dE_k}{dt}$ as a function of time throughout the simulation and compare with DNS data computed using an incompressible spectral code using 512 degrees of freedom in each coordinate direction.[31] For a well resolved simulation using 256 degrees of freedom in each coordinate direction our collocated DG spectral element method is stable throughout the simulation for 2nd-($p = 1$), 4th- ($p = 3$), 8th- ($p = 7$) and 16th-order ($p = 16$) simulations. However, for an under-resolved simulation using only 64 degrees of freedom in each coordinate direction, the 8th- and 16th- order simulations become unstable. Figure 10 plots the kinetic energy dissipation rate for a simulation performed using over-integration with $2N$ points. The use of over-integration stabilizes the 8th- and 16thorder solutions. We note that even at this low resolution the higher-order schemes are more accurate than the low order scheme.



(a) 256^3, Collocation

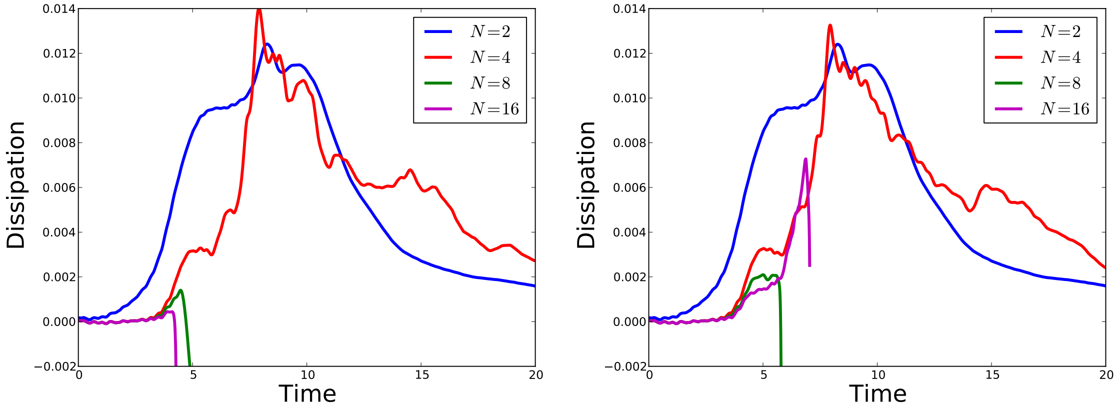
(b) 64^3, Collocation

(c) 64^3 "Over-integration"

Figure 10. Taylor-Green vortex problem at $M = 0.1$, $Re = 1600$ computed using meshes with a) 256^3 degrees of freedom using collocation, b) 64^3 degrees of freedom using collocation, c) 64^3 degrees of freedom using over-integration with $2N$ points

Next we consider the Taylor-Green vortex problem at $Re = 16000$ and $M = 0.2$. Figure **??** plots the kinetic energy dissipation rate using collocation and over-integration. We note that at this higher Mach and Reynolds number, over-integration by a factor of 2 allows for the simulation to remain stable for longer periods of time, however, both 8th- and 16th- order solutions become unstable and additional stabilization may be necessary.

In order to understand the success of polynomial de-aliasing for subsonic compressible flows we consider the errors in integration introduced through the use of quadrature. As an example problem we consider a one-dimensional periodic domain using a single element with $N = 16$. We evaluate the spatial residual using increasing quadrature rules. The residual is computed about a flow with mean Mach number of 0.2 with 20% fluctuations in velocity and pressure. We consider flows with density fluctuations of 0, 5 and 20% about the mean. Figure 12 plots the maximum error in the residual for the energy equation for each mode using a Legendre polynomial basis, computed about 100 random states. For flows with zero density fluctuation, using the $3/2N$ quadrature points allows for exact integration of the lower $1/4$ modes, while the Gaussian quadrature rule using $2N$ or more points is exact. If density perturbations exist in the flow, then exact projection of the flux onto the polynomial space is not possible and an error is introduced for all modes. However, for density fluctuations on the order of 5% the use of $2N$ points is able to reduce the error by nearly 4 orders of magnitude, which appears to be sufficient to ensure stability. As the magnitude of the density fluctuations increases the amount of quadrature required to maintain is similarly expected to increase. So far we have been unable to come up with a satisfactory criteria for how accurate integration is required to

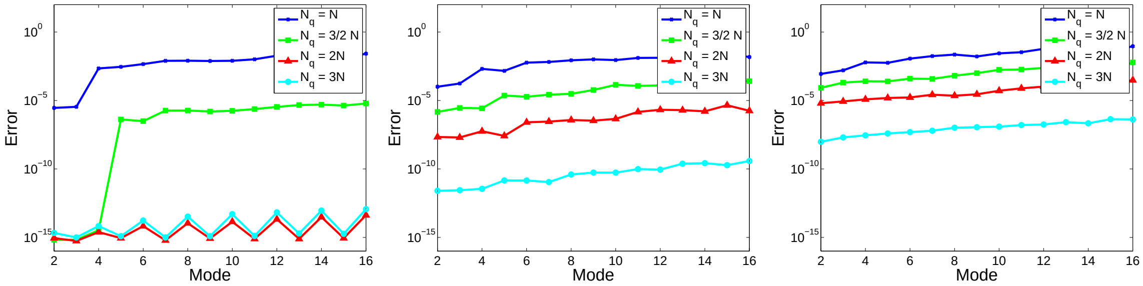


(a) 64^3, Collocation

(b) 64^3 "Over-integration"

Figure 11. Taylor-Green vortex problem at $M = 0.2$, $Re = 16000$ computed using meshes with a) 64^3 degrees of freedom using collocation, b) 64^3 degrees of freedom using over-integration with $2N$ points

maintain stability. Since the computational cost scales with the cube of the factor of over-integration used such a criteria would allow us to most efficiently evaluate the integrals in the residual.



(a) No density variation

(b) 5% density variation

(c) 20% density variation

Figure 12. Errors in evaluation of the residual using numerical quadrature as a function of N with 20% variation in velocity and pressure and density variation of a) 0, b) 5% c) 20%

As noted previously the use of over-integration using $2N$ points increases the cost of a residual evaluation and hence an entire simulation by a factor of 8. When using implicit method we may be able to offset some of this cost by computing Frechet derivatives using lower quadrature rules, as long as the unsteady residual is computed with "over-integration". This corresponds to using an approximate Jacobian in our Newton-Krylov scheme, which may degrade the convergence of the scheme. This is an approach we hope to pursue in the future.

As noted in the introduction, in the variational multiscale framework the numerical stabilization and the subgrid scale model play the same role. Namely, they approximate the effect of the unresolved modes on the resolved modes. We intend to pursue physics-based subgrid scale models using our finite element framework. We note that as we continue down this path the interaction of the numerical stabilization and the subgrid scale modeling is important. In particular, we may wish to consider other types of stabilization such as Streamline-Upwind/Petrov-Galerkin (SUPG),[32,33] Galerkin-Least-Square (GLS)[34] or multiscale[35] methods. These techniques include a stabilization term where the residual of the strong form of the Navier-Stokes equations is weighted with the adjoint of the convective operator in order to add dissipation in the stream-wise direction. When using an explicit scheme such stabilization terms are prohibitively expensive since each stage would involve the solution of a globally couple system. However, in an implicit framework these type of stabilization terms may be more appropriate, especially for turbulent flows.[36]

V. Appendix

V.A. Convection of an Isentropic Vortex

The isentropic vortex convection problem is initialized with a perturbation about a uniform flow given by:

$$\begin{aligned} \delta u &= -U_\infty \beta \frac{y-y_c}{R} e^{-\frac{r^2}{2}} && (39) \\ \delta v &= U_\infty \beta \frac{x-x_c}{R} e^{-\frac{r^2}{2}} && (40) \\ \delta \left(\frac{P}{\rho} \right) &= \frac{1}{2} \frac{\gamma}{\gamma - 1} U_\infty^2 \beta^2 e^{-r^2} && (41) \end{aligned}$$

where U_∞ is the convecting speed of the vortex, β is the vortex strength, R the characteristic radius, and (x_c, y_c) the vortex center. The convection velocity is angled 30° from the x-direction so that the flow does not align with the mesh. The exact solution to this flow is a pure convection of the vortex at speed U_∞.

Taylor Green Vortex

The Taylor-Green vortex flow is simulated using the compressible Navier-Stokes equations at $M_0 = 0.1$. The flow is solved on an isotropic domain which spans $[0, 2\pi L]$ in each coordinate direction. The initial conditions are given by:

$$\begin{aligned} u &= V_0 \sin(x/L) \cos(y/L) \cos(z/L) && (42) \\ v &= -V_0 \cos(x/L) \sin(y/L) \cos(z/L) && (43) \\ w &= 0 && (44) \\ p &= \rho_0 V_0^2 \left[\frac{1}{\gamma M_0^2} + \frac{1}{16} \left(\cos(2x) + \sin(2y) \right) \left(\cos(2z) + 2 \right) \right] && (45) \end{aligned}$$

where u,v and w are the components of the velocity in the x, y and z-directions, p is the pressure and ρ is the density. The flow is initialized to be isothermal ($\frac{p}{\rho} = \frac{p_0}{\rho_0} = RT_0$). The flow is computed at a Reynolds number of $Re = \frac{\rho_0 V_0 L}{\mu} = 1600$, where μ is the viscosity. The Prandtl number is $Pr = 0.71$, while the bulk viscosity is given by the Stokes hypothesis: $\lambda = -\frac{2}{3}\mu$.

References

[1] Bell, J. H., Heineck, J. T., Zilliac, G., and Mehta, R. D., "Surface and Flow Field Measurements on the FAITH Hill Model," AIAA Paper 2012-0704, 2012.

[2] Hughes, T. J. R., Feijoo, G. R., Mazzei, L., and Quincy, J.-B., "The variational multiscale method - a paradigm for computational mechanics," *Comput. Methods Appl. Math.*, Vol. 166, 1998, pp. 3–24.

[3] Collis, S. S., "Monitoring unresolved scales in multiscale turbulence modeling," *Phys. Fluids*, Vol. 13, 2001, pp. 1800–1806.

[4] Hughes, T. J. R., "Multiscale Phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods," *Comput. Methods Appl. Math.*, Vol. 127, 1995, pp. 387–401.

[5] Hughes, T. J., Mazzei, L., and Jansen, K. E., "Large Eddy Simulation and the variational multiscale method," *Computing and Visualization in Science*, Vol. 3, 2000, pp. 47–59.

[6] Koobus, B. and Farhat, C., "A variational multiscale method for the large eddy simulation of compressible turbulent flows on unstructured meshes – application to vortex shedding," *Comput. Methods Appl. Mech. Engrg*, Vol. 193, 2004, pp. 1367–1383.

[7] Farhat, C., Rajasekharan, J., and Koobus, B., "A dynamic variational multiscale method for large eddy simulations on unstructured meshes," *Computer Methods in Applied Mechanics and Engineering*, Vol. 195, No. 1316, 2006, pp. 1667 – 1691.

[8] Oberai, A. and Hughes, T., "The Variational Multiscale Formulation of LES: Channel Flow at $Re_\tau = 590$," AIAA 2002-1056, 2002.

[9] Ramakrishnan, S. and Collis, S. S., "The Local Variational Multi-Scale Method for Turbulence Simulation," Sandia Report SAND2005-2733, 2005.

[10] Wang, Z., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H., Kroll, N., May, G., Persson, P.-O., van Leer, B., and Visbal, M., "High-Order CFD Methods: Current Status and Perspective," *International Journal for Numerical Methods in Fluids*, Vol. 00, 2012, pp. 1–42.

[11] Roe, P. L., "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372.

[12] Bassi, F. and Rebay, S., "GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations," *Discontinuous Galerkin Methods: Theory, Computation and Applications*, edited by K. Cockburn and Shu, Springer, Berlin, 2000, pp. 197–208.

[13] Vos, P., Sherwin, S., and Kirby, R., "From h to p Efficiently: Implementing finite and spectral/hp element discretizations to achieve optimal performance at low and high order approximations." *Journal of Computational Physics*, Vol. 229, No. 13, 2010, pp. 5161–5181.

[14] Gassner, G. and Kopriva, D. A., "A comparison of the dispersion and dissipation errors of Gauss and Gauss-Lobatto discontinuous Galerkin spectral element methods," *SIAM J. Sci. Comput.*, Vol. 33, 2011, pp. 2560–2579.

[15] Knoll, D. A. and Keyes, D. E., "Jacobian-free Newton-Krylov methods: a survey of approaches and applications," *Journal of Computational Physics*, Vol. 193, No. 1, 2004, pp. 357–397.

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

[17] Saad, Y., *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, 1996.

[18] Beam, R. and Warming, R., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," *AIAA Journal*, Vol. 16, No. 4, 1978, pp. 393 – 402.

[19] Pulliam, T. and Chaussee, D., "A Diagonal Form of an Implicit Approximate-Factorization Algorithm," *Journal of Computational Physics*, Vol. 39, 1981, pp. 347–363.

[20] Ronquist, E. M. and Patera, A. T., "Spectral Element Multigrid. I. Formulation and Numerical Results," *Journal of Scientific Computing*, Vol. 2, 1987, pp. 389–406.

[21] Maday, Y. and Munoz, R., "Spectral Element Multigrid. II. Theoretical Justification," *Journal of Scientific Computing*, Vol. 3, 1988, pp. 323–353.

[22] Fidkowski, K. J., Oliver, T. A., Lu, J., and Darmofal, D. L., "p-Multigrid solution of high-order discontiunous Galerkin discretizations of the compressible Navier-Stokes equations," *Journal of Computational Physics*, Vol. 207, No. 1, 2005, pp. 92–113.

[23] Persson, P.-O. and Peraire, J., "An efficient low memory implicit DG algorithm for time dependent problems," AIAA 2006-0113, 2006.

[24] Diosady, L. T. and Darmofal, D. L., "Preconditioning methods for discontinuous Galerkin solutions of the Navier-Stokes equations," *Journal of Computational Physics*, Vol. 228, 2009, pp. 3917–3935.

[25] van der Vegt, J. and Rhebergen, S., "hp-Multigrid as Smoother algorithm for higher order discontinuous Galerkin discretizations of advection dominated flows: Part I. Multilevel Analysis," *Journal of Computational Physics*, Vol. 231, 2012, pp. 7537–7564.

[26] Karamanos, G.-S. and Karniadakis, G., "A Spectral Vanishing Viscosity Method for Large-Eddy Simulations," *Journal of Computational Physics*, Vol. 163, 2000, pp. 22–50.

[27] Kirby, R. and Sherwin, S., "Stabilisation of spectral/hp element methods through spectral vanishing viscosity: Application to fluid mechanics modelling," *Comput. Methods Appl. Mech. Engrg*, Vol. 195, 2006, pp. 3128–3144.

[28] Honein, A. E. and Moin, P., "Higher entropy conservation and numerical stability of compressible turbulence simulations," *Journal of Computational Physics*, Vol. 201, 2004, pp. 531–545.

[29] Kirby, R. M. and Karniadakis, G. E., "De-aliasing on non-uniform grids: algorithms and applications," *Journal of Computational Physics*, Vol. 191, 2003, pp. 249–264.

[30] Gassner, G. and Beck, A. D., "On the accuracy of high-order discretizations for underresolved turbulence simulations," *Theor. Comput. Fluid Dyn.*, Vol. 33, 2011, pp. 2560–2579.

[31] van Ress, W., Leonard, A., Pullin, D., and Koumoutsakos, P., "A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds number," *Journal of Computational Physics*, Vol. 230, 2011, pp. 2794–2805.

[32] Hughes, T. J. R. and Tezduyar, T. E., "Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations," Vol. 45, 1984, pp. 217–284.

[33] Hughes, T. J. R., Franca, L., and Mallet, M., "A new finite element formulation for computational fluid dynamics: I Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics," Vol. 54, 1986, pp. 223–234.

[34] Hughes, T. J. R., Franca, L. P., and Hulbert, G. M., "A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations," Vol. 73, 1989, pp. 173–189.

[35] Franca, L. P., Frey, S. L., and Hughes, T. J. R., "Stabilized finite element methods: I. Application to the advective-diffusive model," Vol. 95, 1992, pp. 253–276.

[36] Bazilevs, Y., Calo, V., Cottrell, J., Hughes, T., Reali, A., and Scovazzi, G., "Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows," *Computer Methods in Applied Mechanics and Engineering*, Vol. 197, No. 14, 2007, pp. 173 – 201.