Subcell finite volume multigrid preconditioning for high-order discontinuous Galerkin methods

Philipp Birken, Gregor J. Gassner and Lea M. Versbach

ABSTRACT
We suggest a new multigrid preconditioning strategy for use in Jacobian-free Newton–Krylov (JFNK) methods for the solution of algebraic equation systems arising from implicit Discontinuous Galerkin (DG) discretisations. To define the new preconditioner, use is made of an auxiliary first-order finite volume discretisation that refines the original DG mesh, but can still be implemented algebraically. As smoother, we consider the pseudo-time iteration W3 with a symmetric Gauss–Seidel-type approximation of the Jacobian. As a proof of concept numerical tests are presented for the one-dimensional Euler equations, demonstrating the potential of the new approach.

1. Introduction
The goal of our research is the construction of efficient Jacobian-free preconditioners for high order Discontinuous Galerkin (DG) discretisations with implicit time integration. One of our main interests is three-dimensional unsteady compressible flow. High-order DG methods (and related methods such as Flux Reconstruction (FR) discretisations) offer great potential for Large Eddy Simulation (LES) of turbulent flows with geometries, such as jet engines. The idea of DG (or FR) is to approximate the solution element-wise using a polynomial, which is allowed to be discontinuous across element interfaces, see Kopriva (2009) and Huynh (2007). Communication and coupling of degrees of freedom (DOF) is only across faces, whereas the element-local computations are very dense. As a result, DG methods are very well suited for domain-decomposition-based parallelisation (see, e.g. Hindenlang et al. 2012; Vincent et al. 2016). The specific variant we consider is the DG Spectral Element Method (DG-SEM), e.g. Kopriva, Woodruff, and Husein (2002). We use a Lagrange-type (nodal) basis with Gauss–Lobatto (GL) quadrature nodes with the collocation of the discrete integration. These choices yield DG operators that satisfy the summation-by-parts (SBP) property (see Gassner 2013), which is the discrete analogue to integration-by-parts. SBP is key to construct methods that are discretely entropy stable and/or kinetic energy preserving.

DG discretisation in space results in a big system of ODEs. Due to geometry features and thin boundary layers that occur in challenging compressible turbulent flow applications as the design of jet engines, aeroplanes and wind turbines, the resulting large system of ODEs is stiff. Implicit time integrators can overcome the deficiency of explicit time integrators with restrictive CFL conditions. However, efficiency can only be restored when the arising large non-linear systems are solved efficiently in terms of CPU time, but also regarding memory consumption. Vincent and Jameson mention that solvers for linear and non-linear equation systems are severely lacking for 3D DG applications as one of four major obstacles that need to be solved if high-order methods are to be widely adopted by, e.g. industry, Vincent and Jameson (2011). Candidates for solvers are FAS-Multigrid (full-approximation multigrid scheme) and preconditioned Jacobian-Free Newton–Krylov methods (JFNK) (Knoll and Keyes 2004) where multigrid can be used as a preconditioner (see Birken and Jameson 2010). The JFNK technology is in general interesting, as the memory use is minimised. Although DG systems have a weakly coupled block structure, the blocks themselves can be large. In particular, the problem for high-order
DG methods is that the number of unknowns per element increases dramatically with increasing polynomial degree and dimension, leading to large dense Jacobian blocks (see Birken et al. 2013; Birken 2012). For a finite volume method, the block size is \(2+d\) with dimension \(d\), whereas for a DG-SEM with \(p\)th degree polynomials, it is \((d+2)(p+1)^d\). For degree 2 in 3D, this is already 135. The favourable memory consumption of the JFNK approach is obsolete if the preconditioner requires the storage of (parts of) the DG system Jacobian.

Hence in this article, we present a novel idea for the construction of a well-performing preconditioner for the JFNK approach, while retaining the low memory use, i.e. a Jacobian-free preconditioner. The main ingredient consists in the construction of a simplified replacement operator. A motivation for this is a previously proved equivalence between a DG-SEM discretisation and a high order FV discretisation, see Fisher et al. (2013). One could, for instance, choose a different polynomial order in the element to generate a replacement operator as in Fidkowski et al. (2005) and Birken et al. (2013). However, we aim to retain the number of DOFs in our replacement operator by introducing subcells in each element, namely \(p+1\) in each spatial direction. On this subcell-element grid, the simplest replacement operator is a first-order finite volume (FV) discretisation. In some sense, we reinterpret the nodal values as input for an FV method. This gives a semi-structured–unstructured approximation, where the elements are unstructured, but inside the element the subcells are structured (Versbach, Birken, and Gassner 2018). We extend the idea of this paper to the Euler equations. In the resulting approximate Jacobian, we now only have \((d+2)(p+1)(2d+1)\) entries (Birken 2012). Furthermore, it allows to use the available knowledge about fast multigrid (MG) methods for FV discretisations on (block-)structured meshes. As a smoother for our FV discretisation, we use a state of the art low memory W3 smoother from Birken, Bull, and Jameson (2018).

A related approach was proposed in Allaneau, Li, and Jameson (2012) for spectral difference (SD) methods, where the replacement operator is also an FV discretisation on a potentially fine grid. However, there the FV grid is not embedded in the high-order grid, but overlayed. Thus, it is necessary to interpolate the solution (and the residual) in-between different grids (with different topologies), which needs interpolation and reconstruction operators similar to Chimera techniques. In contrast, we want to harness in particular the semi-unstructured–structured mesh-topology: our FV discretisation basically lives on the same DOFs as the nodal high-order DG method.

In the remainder of the paper, we first describe our prototype problem, the one-dimensional compressible Euler equations. We then present the DG-SEM and the FV subcell discretisations as well as the multigrid solver. In the last part of the paper, we show numerical experiments to validate the approach and draw our conclusions.

2. One-dimensional Euler equations

As a prototype problem for our novel idea, we consider the one-dimensional compressible Euler equations for a perfect gas

\[
\begin{pmatrix}
\rho \\
m \\
mv + p \\
Hm
\end{pmatrix}_t + \begin{pmatrix}
m \\
E \\
E
\end{pmatrix}_x = 0, \tag{1}
\]

with appropriate initial and boundary conditions. Here \(\rho\) is the density, \(mv\) the momentum, \(p\) the pressure, \(E\) the energy and \(H = E + p/\rho\) the enthalpy. Defining \(U = (\rho, m, \rho E)^T\) and \(f(U) = (mv, mv + p, Hm)^T\), we can write the Euler equations in vector form:

\[
U_t + f(U)_x = 0. \tag{2}
\]

3. Spatial discretisation

3.1. DG-SEM

For the spatial discretisation, we introduce a grid with \(K\) elements \(e^k\) of width \(\Delta x_k, k = 1, \ldots, K\). Each element is transformed to the reference element \([-1, 1]\) by a linear mapping with Jacobian \(J_k = \Delta x_k/2\). The solution is approximated by a nodal polynomial in reference space with degree \(p\) in each element \(e^k\)

\[
U(\xi, t)|_{e^k} \approx U^k(\xi, t) = \sum_{j=1}^{p+1} U_j^k(t) \psi_j(\xi), \tag{3}
\]

where the interpolation nodes are the GL nodes \(\{\xi_j\}_{j=1}^{p+1}\). We use the element mapping to transform the Euler equations into reference space

\[
J_k U_t + f(U)|_{\xi} = 0, \tag{4}
\]
and insert our ansatz (3). Next, we integrate over the reference element, use integration-by-parts for the flux term and replace the fluxes at the element interfaces with so-called numerical flux functions \( f^\ast \) to arrive at

\[
\int_{-1}^{1} J_k U_k^k \psi (\xi) \, d\xi + [f^\ast \psi (\xi)]_{-1}^{1} - \int_{-1}^{1} f(U_k^k) \psi \xi (\xi) \, d\xi = 0. \tag{5}
\]

As a numerical flux function, we choose the Rusanov flux (or local Lax–Friedrich flux)

\[
f^\ast(U^-, U^+) = \frac{f(U^-) + f(U^+)}{2} - \frac{\lambda_{\text{max}}}{2} (U^+ - U^-), \tag{6}
\]

where \( U^-, U^+ \) are the values left and right at an element interface and \( \lambda_{\text{max}} \) is an estimate of the maximum wave speed at the interface.

As stated above, the main idea of the DG-SEM is collocation. We use collocation for our discrete integration, i.e. we replace the integrals in (5) with GL quadrature rules at the same location as our interpolation, which can be interpreted as a collocation of the non-linear fluxes \( f(U) \). With this choice, the DG-SEM operators simplify a lot: we get the diagonal mass matrix \( M_{ij} = \delta_{ij} \omega, i, j = 1, \ldots, p + 1 \), the diagonal boundary matrix \( B = \text{diag}(-1, 0, \ldots, 0, 1) \) and the derivative matrix \( D_{ij} := (\psi_j) \xi (\xi_i), i, j = 1, \ldots, p + 1 \).

Replacing the integrals in (5) and using the definitions of the DG-SEM operators, we arrive at the DG-SEM method in the matrix–vector formulation for one cell \( e^k \)

\[
\dot{U}^k = -\frac{1}{J_k} (M^{-1} B f^\ast - M^{-1} D^T M f). \tag{7}
\]

The elemental residuals are coupled through the numerical flux \( f^\ast \) with DOFs from other elements connected via the interfaces. Collecting the equations for all elements in one big system with unknown \( u \) and applying implicit Euler in time gives

\[
u^{n+1} - u^n - \Delta t G(u^{n+1}) = F(u^{n+1}) = 0, \tag{8}
\]

where \( G(u^{n+1}) \) collects the DG-SEM residuals, i.e. the right-hand side of (7).

### 3.2. Finite volume discretisation

Based on the DG-SEM discretisation, we define an FV discretisation on a subcell mesh with \( p+1 \) cells

![Figure 1. Two DG cells with 6 GL nodes each. An equidistant FV mesh is shown in the right cell.](image)

(see Figure 1). The discretisation on this semi-structured–unstructured grid is used to define the preconditioner and to construct the multigrid method. An FV method is based on approximating the cell averages in a subcell \( i \) in an element \( e^k \)

\[
\frac{1}{\Delta x_i} \int_{x_{i-1/2}}^{x_{i+1/2}} U(x, t^n) \, dx \approx U^{kn}_i \tag{9}
\]

at each time level \( t^n \). For a subcell in the element \( e^k \) with subcell size \( \Delta x_i \), the subcell FV discretisation reads

\[
\dot{U}^k_i + \frac{1}{\Delta x_i} (f^\ast_{i+1/2} - f^\ast_{i-1/2}) = 0, \quad i = 1, \ldots, p + 1; \quad k = 1, \ldots, K. \tag{10}
\]

The numerical flux function \( f^\ast_{i+1/2} \) is again Rusanov flux (6), with the values left and right being not the polynomial values, but the subcell average values instead.

### 4. Preconditioned Jacobian-free Newton-GMRES

The resulting system of nonlinear equations (8) is solved using Newton’s method, written for the equation \( \dot{F}(u) = 0 \):

\[
\text{solve } \frac{\partial F(u)}{\partial u} |_{u = s} = -F(u^{(k)}),
\]

\[
u^{(k+1)} = u^{(k)} + s, \quad k = 0, 1, \ldots
\]

for a given initial guess \( u^{(0)} \). The linear system in (11) is solved using right preconditioned GMRES with a relative tolerance.

In order to not compute the Jacobian in each iteration (11), we replace the matrix–vector products appearing in GMRES by a difference quotient:

\[
\frac{\partial F(u)}{\partial u} \approx \frac{F(u + \epsilon q) - F(u)}{\epsilon}, \quad \text{with } \epsilon = \frac{1}{\|q\|}. \tag{12}
\]
5. Finite volume multigrid preconditioner

The basis for the preconditioner is an agglomeration multigrid method on the finite volume grid. The coarse grid problems are given by applying the FV discretisation on the coarse grid. To restrict fine grid values, they are summed, weighted by the volumes of the respective cells and divided by the total volume. For an equidistant grid in one dimension, the corresponding restriction operator is given by

\[ R_{l-1}^{l} = \frac{1}{2} \begin{pmatrix} 1 & 1 & \ldots & 1 \\ 1 & 1 & \ldots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \ldots & 1 \end{pmatrix}. \]  

(13)

As prolongation we use the injection, where the value in the coarse cell is taken for all corresponding fine cells:

\[ P_{l-1}^{l} = 2R_{l-1}^{l}. \]  

(14)

A generic smoother is an iterative method for the solution of \( A \mathbf{s}_{l} = \mathbf{b}_{l} \) and is given by \( s_{l+1}^{k+1} = M_{S,l}s_{l}^{k} + N_{S,l}^{-1}b_{l} \). The index \( l \) specifies the multigrid level, while \( S \) denotes that this iterative method represents the smoother. It is possible to construct an MG preconditioner based on a V- or a W-cycle, as well as several consecutive cycles. The number of pre- and postsmoothing steps is also flexible. We now write down a V-cycle multigrid algorithm with one step of pre- and postsmoothing, which corresponds to \( \gamma = 1 \) in the following pseudo-code:

- \( s_{l} = M_{S,l}s_{l} + N_{S,l}^{-1}b_{l} \) (presmoothing)
- if \( l > 0 \)
  - \( r_{l-1} = R_{l-1}^{l}(A_{l}s_{l} - b_{l}) \) (restriction)
  - \( v_{l-1} = 0 \)
    - for \( j = 1, \ldots, \gamma : v_{j-1} = MG(v_{j-1}, r_{j-1}) \)
  - \( s_{j} = s_{j} - P_{j}^{l-1}v_{j-1} \) (fine grid correction)
  - \( s_{j} = M_{S,j}s_{j} + N_{S,j}^{-1}b_{j} \) (postsmoothing)

This gives rise to an iterative method of the form \( s_{l+1}^{k+1} = M_{MG,l}s_{l}^{k} + N_{MG,l}^{-1}b_{l} \). In the case of an \( l_{\text{max}} \)-level multigrid cycle with presmoothing on the coarsest level, the preconditioner \( N_{l,MG}^{-1} \approx A^{-1} \) is defined recursively by

\[
\begin{align*}
N_{0,MG}^{-1} &= N_{S,0}^{-1}, \quad \text{and for } l = 1, \ldots, l_{\text{max}}:
N_{l,MG}^{-1} &= M_{S,l}(N_{S,l}^{-1} - P_{l}^{l-1}N_{l-1,MG}^{-1}R_{l-1}^{l}A_{l}N_{S,l}^{-1}) + P_{l}^{l-1}N_{l-1,MG}(R_{l-1}^{l}) + N_{S,l}^{-1}.
\end{align*}
\]

(15)

5.1. Smoothers: pseudo-time iterations

As smoothers for the multigrid preconditioner, we consider W schemes (see Birken, Bull, and Jameson 2018). A pseudo-time derivative is added to Equation (11) to yield

\[
\frac{\partial s}{\partial t^\gamma} + As - b = 0.
\]

(16)

A W smoother is given by

\[
\begin{align*}
s_{n} &= \bar{s}_{n}, \\
s_{j} &= s_{n} - \alpha_{j}\Delta t^\gamma W_{j}^{-1}(A_{j} - b), \quad j = 1, \ldots, s, \\
s_{n+1} &= s_{s},
\end{align*}
\]

(17)

where \( W \approx I + \eta \Delta t^\gamma A \). The free parameters are \( \eta \), usually taken from the range \([0.25, 1.5]\), as well as \( \alpha_{j} \in [0, 1] \) and a local \( \Delta t_{j} \), given by a pseudo-CFL number \( c_{j} \) depending on the maximal eigenvalue of the Jacobian \( \lambda_{\text{max}} \):

\[
\Delta t_{j} = c_{j} \frac{\Delta x_{l}}{\lambda_{\text{max}}}.
\]

(18)

Smoothers of form (17) can be written as a linear iterative scheme

\[
\bar{s}_{n+1} = M_{S}s_{n} + N_{S}^{-1}b_{l},
\]

(19)

where for a 3-stage W smoother, we obtain

\[
N_{S}^{-1} = \alpha_{3}\Delta t^\gamma W_{3}^{-1} - \alpha_{3}\alpha_{2}\Delta t^\gamma W_{2}^{-1}A W_{1}^{-1} + \alpha_{3}\alpha_{2}\alpha_{1}\Delta t^\gamma (W_{3}^{-1}A) W_{2}^{-1}.
\]

(20)

The approximation of \( W^{-1} \) will be explained in the next section. The coefficients are given by \([\alpha_{1}, \alpha_{2}, \alpha_{3}] = [0.1481, 2/5, 1] \).

5.1.1. SGS preconditioner

The specific approximation to define \( W \) is based on a symmetric Gauss–Seidel (SGS) approach. Such a method was originally suggested in Swanson, Turkel, and Rossov (2007) and further developed by several
6. Numerical results

We consider the one-dimensional Euler equations on the interval [0, 10] and study one step of implicit Euler where we look at the convergence rate of GMRES with maximal 100 iterations in the first Newton step. All results are produced in Python. Measuring the CPU time will not give a great insight into performance in the one-dimensional case and is therefore not discussed in the following, but is important to consider in higher dimensions.

We equip the Euler equations with periodic boundary conditions and consider two different initial conditions: A subsonic case \((\rho_0, v_0, p_0) = (1 + 0.1 \sin(2\pi x/10), 1, 28)\) with Mach number 0.16 and a transonic case \((\rho_0, v_0, p_0) = (1 + 0.1 \sin(2\pi x/10), 1, 1)\) with Mach number 0.85. We choose a CFL number of 100 and test the discussed MG preconditioner with W3 smoother for a V cycle, see (15). A simple block Jacobi preconditioner with blocks corresponding to the \(3 \times 3\) systems does not improve the convergence rate of the GMRES cycle compared to no preconditioner. This motivates to consider more sophisticated preconditioners for the given problem. We also note that applying the W3 presmoothing step several times gives almost no convergence improvement while being very expensive in terms of computational costs. The same holds for using a W cycle or two consecutive V cycles, as well as for the method of nonsymmetric Restriction Aggregation (NSR) from Sala and Tuminaro (2008).

We test the framework for 4th- and 8th-order DG methods with 240 and 480 DOFs, respectively. In order to have a reference for efficiency, we construct a preconditioner based on the Jacobian of FV discretisation (12). The new multigrid preconditioner approximates the inverse of the FV Jacobian and thus cannot be expected to behave superior to the reference preconditioner. The reference preconditioner is applied by using GMRES with tolerance 1e-14 and maximal 300 iterations. This is very expensive in terms of computations and only suggested to compare how well the proposed MG preconditioner approximates the inverse of the FV Jacobian.

6.1. Subsonic case

In the subsonic case, the pseudo-CFL number for the W3 smoother is \(c^* = 10, \eta = 1.4\) and \(d = 0.1\). We consider two different MG preconditioners: one with only one presmoothing step on each level and one with one
pre- and one postsmoothing step on each level. The convergence results are shown in Figure 2.

We see that the reference preconditioner (FV) gives very good results for all four test cases, but works better for 4th order. In particular, there is fast initial convergence, which is crucial to get fast termination within an inexact Newton’s method. The suggested MG preconditioners yield a very good approximation to the reference preconditioner, especially within the first 20 GMRES iterations. The MG preconditioner with pre- and postsmoothing gives the best results, outperforming the MG preconditioner equipped with only presmoothing slightly. This holds for DG solvers of 4th and 8th order as well as for DOFs 240 and 480. Increasing the DOFs does not have a visible impact on the convergence rate of the reference preconditioner for both 4th and 8th order while we can notice small differences in the behaviour of the MG preconditioners. The residual after 100 GMRES iterations differs slightly more for the two MG preconditioners when increasing the DOFs for both 4th and 8th order. Since the FV replacement operator is of the first order, the question arises how it behaves for the increased order of the DG method. When going from 4th to 8th order for 240 DOFs, there are two orders of magnitude decrease of the reference preconditioner, whereas the decrease for the W3 preconditioners is very small and they behave very similarly. For 480 DOFs, there is a 1.5 order of magnitude decrease of the reference preconditioner and the MG preconditioners behave similarly. It is noticeable that the MG preconditioners mimic the FV preconditioner better when increasing the order of the DG discretisation. We expect a performance improvement of the smoother for optimised $c^*$, $\eta$ and $d$. 

Figure 2. Convergence history for GMRES for one Newton iteration, subsonic case, DG order 4 (top) and 8 (bottom), 240 DOFs (left) and 480 DOFs (right).
6.2. Transonic case

In the transonic case, the pseudo-CFL number for the W3 smoother is \( c^* = 2, \eta = 0.7 \) and \( d = 0.1 \). We consider the same two different MG preconditioners as in the transonic case. The convergence results are shown in Figure 3.

Both reference FV and MG preconditioners perform worse than in the subsonic case. Such a loss of performance has been reported for a \( p \)-multigrid method in Premasuthan et al. (2009). In the transonic case, the reference preconditioner works very similar for 4th and 8th order and different DOFs. We notice only a slight decrease in performance when increasing the order as well as when increasing the DOF. For 240 DOFs, the first MG preconditioner mimics the performance of the reference preconditioner with approximately 0.5 order of magnitude degradation and works slightly better for the 4th order. The second MG preconditioner does not work well for 240 DOFs and the 4th and 8th order, respectively. When increasing the DOFs, the MG preconditioner with pre- and post-smoothing slightly outperforms the one with pre-smoothing only. In the case of 480 DOFs, the second MG preconditioner works around 1 order of magnitude worse than the reference preconditioner, giving for both 4th and 8th order methods a result approximately 0.5 order of magnitude degradation compared to 240 DOFs. Again the order affects the performance only slightly for different DOFs.

7. Conclusions

We presented a new multigrid preconditioning strategy for use in JFNK methods for the solution of equation systems arising from DG discretisations. The core idea is to make use of an auxiliary first-order FV
discretisation that refines the original DG mesh, but can still be implemented algebraically. As smoother, we consider W3. Numerical results show the potential of the approach as a proof of concept for the one-dimensional Euler equations. A simple block Jacobi preconditioner does not improve the convergence rate compared to no preconditioner at all, which justifies the necessity of using this more sophisticated preconditioner. The convergence results of the proposed preconditioner are promising, being close to a quasi-exact preconditioner in the subsonic case. Our results indicate that the performance of the preconditioner is only weakly influenced by the order of the DG discretisation. A possible extension of the preconditioner to multiple spatial dimensions could be based on a tensor product strategy, which is also the natural approach for the extension of DG-SEM to multiple spatial dimensions. It should be noted that W3 smoothers are designed for high aspect ratio grids and have been shown to achieve even better performance on those compared to equidistant meshes (Birken, Bull, and Jameson 2018). This is of special interest for Navier–Stokes equations with a boundary layer.

Disclosure statement
No potential conflict of interest was reported by the authors.

Funding
Philipp Birken and Lea Versbach gratefully acknowledge the support of the Swedish Research Council – Vetenskapsrådet [grant number 2015-04133]. Gregor Gassner has been supported by the European Research Council (ERC) under the EU’s 8th Framework Program Horizon 2020 with the research project Extreme [ERC grant agreement no. 714487].

References
Allaneau, Yves, Lala Y. Li, and Antony Jameson. 2012. “Convergence Acceleration of High Order Numerical Simulations using a Hybrid Spectral Difference – Finite Volume Multigrid Method.” In 7th International Conference on Computational Fluid Dynamics (ICCFD7), Big Island, Hawaii, July 9–13.
Birken, Philipp. 2012. “Numerical Methods for the Unsteady Compressible Navier-Stokes Equations.” Habilitation Thesis, University of Kassel.
Birken, Philipp, Jonathan Bull, and Antony Jameson. 2018. “Preconditioned Smoothers for the Full Approximation Scheme for the RANS Equations.” Journal of Scientific Computing 78 (2): 995–1022.
Birken, Philipp, Gregor Gassner, Mark Haas, and Claus-Dieter Munz. 2013. “Preconditioning for Modal Discontinuous Galerkin Methods for Unsteady 3D Navier-Stokes Equations.” Journal of Computational Physics 240: 20–35.
Birken, Philipp, and Antony Jameson. 2010. “On Nonlinear Preconditioners in Newton-Krylov-Methods for Unsteady Flows.” International Journal of Numerical Methods in Fluids 62: 565–573.
Fidkowski, Krzysztof J., Todd A. Oliver, James Lu, and David L. Darmofal. 2005. “p-Multigrid Solution of High-order Discontinuous Galerkin Discretizations of the Compressible Navier-Stokes Equations.” Journal of Computational Physics 207: 92–113.
Fisher, Travis C., Mark H. Carpenter, Jan Nordström, Nail K. Yamaalev, and Charles Swanson. 2013. “Discretely Conservative Finite-difference Formulations for Nonlinear Conservation Laws in Split Form: Theory and Boundary Conditions.” Journal of Computational Physics 234: 353–375.
Gassner, Gregor J. 2013. “A Skew-symmetric Discontinuous Galerking Spectral Element Discretization and Its Relation to SBP-SAT Finite Difference Methods.” Journal of Scientific Computing 35: A1233–A1253.
Hindenlang, Florian, Gregor J. Gassner, Christoph Altmann, Andrea Beck, Mark Staudenmaier, and Claus-Dieter Munz. 2012. “Explicit Discontinuous Galerkin Methods for Unsteady Problems.” Computers Fluids 61 (May): 86–93.
Huynh, H. T. 2007. “A Flux Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin Methods.” In AIAA, 2007–4079. Miami, FL.
Knoll, Dana A., and David E. Keyes. 2004. “Jacobian-free Newton-Krylov Methods: A Survey of Approaches and Applications.” Journal of Computational Physics 193: 357–397.
Kopriva, David A. 2009. Implementing Spectral Element Methods for Partial Differential Equations. Dordrecht: Springer Scientific Computing.
Kopriva, David A., S. L. Woodruff, and M. Y Hussaini. 2002. “Computation of Electromagnetic Scattering with a Non-conforming Discontinuous Spectral Element Method.” International Journal for Numerical Methods in Engineering 53: 105–122.
Premasuthan, Sachin, Chunlei Liang, Antony Jameson, and Z. J. Wang. 2009. “A P-Multigrid Spectral Difference Method for Viscous Compressible Flow Using 2D Quadrilateral Meshes.” AIAA Paper 2009-950. Orlando, FL.
Sala, Marzio, and Raymond S. Tuminaro. 2008. “A New Petrov-Galerkin Smoothed Aggregation Preconditioner for Non-symmetric Linear Systems.” SIAM Journal on Scientific Computing 31 (1): 143–166.
Swanson, R. Charles, Eli Turkel, and Cord-Christian Rossow. 2007. “Convergence Acceleration of Runge-Kutta Schemes for Solving the Navier-Stokes Equations.” Journal of Computational Physics 224 (1): 365–388.
Versbach, Lea M., Philipp Birken, and Gregor J. Gassner. 2018. “Finite Volume Based Multigrid Preconditioners for Discontinuous Galerkin Methods.” PAMM, Vol. 18. Munich.
Vincent, Peter E., and Antony Jameson. 2011. “Facilitating the Adoption of Unstructured High-Order Methods Amongst a Wider Community of Fluid Dynamists.” *Mathematical Modelling of Natural Phenomena* 6 (3): 97–140.

Vincent, Peter E., Freddie Witherden, Brian Vermeire, Jin S. Park, and Arvind Iyer. 2016. “Towards Green Aviation with Python at Petascale.” SC16: International Conference for High Performance Computing, Networking, Storage and Analysis. Article No. 1. Salt Lake City, UT.