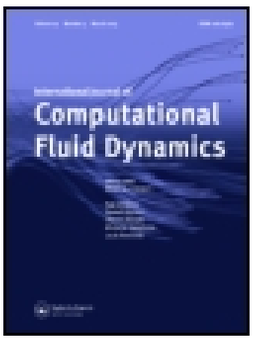


International Journal of Computational Fluid Dynamics

ISSN: 1061-8562 (Print) 1029-0257 (Online) Journal homepage: https://www.tandfonline.com/loi/gcfd20

Comparison of hybrid and standard discontinuous Galerkin methods in a mesh-optimisation setting

K. J. Fidkowski

To cite this article: K. J. Fidkowski (2019): Comparison of hybrid and standard discontinuous Galerkin methods in a mesh-optimisation setting, International Journal of Computational Fluid Dynamics

To link to this article: https://doi.org/10.1080/10618562.2019.1588962

Published online: 08 Mar 2019.

INTERNATIONAL JOURNAL OF COMPUTATIONAL FLUID DYNAMICS
https://doi.org/10.1080/10618562.2019.1588962

Taylor & Francis
Taylor & Francis Group

Comparison of hybrid and standard discontinuous Galerkin methods in a mesh-optimisation setting

K. J. Fidkowski

Aerospace Engineering, University of Michigan, Ann Arbor, MI, USA

ABSTRACT

This paper assesses the benefits of hybridization on the accuracy and efficiency of high-order discontinuous Galerkin (DG) discretizations. Two hybridized methods are considered in addition to DG: hybridized DG (HDG) and embedded DG (EDG). These methods offer memory and computational time savings by introducing trace degrees of freedom on faces that become the only globally-coupled unknowns. To mitigate the effects of solution singularities on accuracy, the methods are compared in an adaptive setting on meshes optimised for the accurate prediction of chosen scalar outputs. Compressible flow results for the Euler and Reynolds-averaged Navier-Stokes equations demonstrate that the hybridized methods offer cost savings relative to DG in memory and computational time. In addition, for the cases tested, EDG yields the lowest error levels for a given number of degrees of freedom. These benefits disappear on uniformly-refined meshes, indicating the importance of using order-optimised meshes when comparing the discretizations.

ARTICLE HISTORY
Received 14 November 2018
Accepted 22 February 2019

KEYWORDS
Discontinuous Galerkin; high order; hybridization; adjoints; adaptation

1. Introduction

Although discontinuous Galerkin (DG) methods (Reed and Hill 1973; Cockburn and Shu 2001) have enabled high-order accurate computational fluid dynamics simulations, their memory footprint and computational costs remain large. In addition to advances in solvers and preconditioning strategies (Franciolini, Fidkowski, and Crivellini 2018), another approach for reducing the expense of DG is modifying the discretization to reduce the number of globally coupled degrees of freedom. This reduces the computational time required by the global solver and the memory footprint of the Jacobian matrix. In the present work, we compare accuracy and cost trade-offs of two such modifications, both in the category of *hybridized* DG, in the context of output prediction on optimised meshes.

Hybridization of DG (Cockburn, Gopalakrishnan, and Lazarov 2009; Nguyen, Peraire, and Cockburn 2009) modifies the DG discretization to reduce its computational and memory expense for a given mesh. The high cost of DG arises from the coupling of the degrees of freedom used to approximate an element-wise discontinuous high-order polynomial solution: the residuals inside one element depend on the states on neighbouring elements. This coupling increases the memory requirements for solvers that store the residual Jacobian matrix, in entirety or portions there-of, even with an element-compact stencil. Hybridized discontinuous Galerkin (HDG) methods reduce the number of globally-coupled degrees of freedom by decoupling element solution approximations from each other. Instead, elemental degrees of freedom are linked to new face degrees of freedom through fluxes. Through a static condensation procedure, these face unknowns become the only globally-coupled degrees of freedom in the system. Since the number of face unknowns scales as $p^{\text{dim}-1}$ compared to the p^{dim} scaling for elements, HDG methods can be computationally cheaper and use less memory compared to DG. The embedded discontinuous Galerkin (EDG) method (Peraire, Nguyen, and Cockburn 2011; Fernandez, Nguyen, and Peraire 2017) employs a continuous face approximation space, further reducing the number of globally-coupled degrees of freedom. However, in both HDG and EDG, extra computations

CONTACT K. J. Fidkowski kfid@umich.edu

must be performed to create the statically-condensed system, and to back-solve for the element unknowns after the solution of the face unknowns.

High-order accurate methods are generally advantageous for problems in which the solutions are smooth. Shocks and other singular features, such as trailing edges and boundary layers in Reynolds-averaged simulations, do not generally lend themselves to efficient approximation with high-order polynomials. More beneficial for such features is *h*-adaptation, in which the mesh is refined to resolve the flow in these areas (Venkatakrishnan et al. 2003). Determining the proper balance of high order and *h*-adaptation is the central question of *hp*-adaptive methods (Burgess and Mavriplis 2011; Woopen et al. 2014). Much work has been done in adaptive methods that deliver the highest accuracy with the lowest cost, particularly in goal-oriented techniques (Becker and Rannacher 2001; Venditti and Darmofal 2003; Fidkowski and Darmofal 2011; Yano 2012).

This work compares DG, HDG, and EDG in terms of cost and accuracy on output-based adapted meshes. The use of adapted meshes allows for the realisation of high-order accuracy with increasing p, as otherwise singular features would pollute the results and set the convergence rate. Previous works have compared standard and hybridized DG methods (Woopen et al. 2014; Fidkowski 2016a), and the differentiating contributions of this work are (1) the consideration of EDG in addition to HDG, and (2) the use of meshes created by an output-based optimisation procedure that minimises numerical errors for a particular order and target cost.

2. Discretization

We consider the steady-state, compressible, Navier-Stokes equations,

$$\nabla \cdot \vec{\mathbf{H}}(\mathbf{u}, \nabla \mathbf{u}) + \mathbf{S}(\mathbf{u}, \nabla \mathbf{u}) = \mathbf{0}, \tag{1}$$

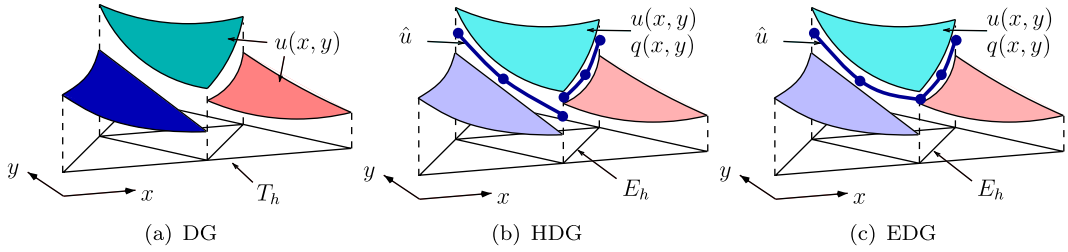
where $\mathbf{u} \in \mathbb{R}^s = [\rho, \rho\vec{v}, \rho E, \rho\tilde{\nu}]^{\mathrm{T}}$ is the conservative state vector of rank s, consisting of density, momentum, total energy, and turbulent viscosity components, $\vec{\mathbf{H}}(\mathbf{u}, \nabla\mathbf{u}) = \vec{\mathbf{F}}(\mathbf{u}) + \vec{\mathbf{G}}^{\mathrm{visc}}(\mathbf{u}, \nabla\mathbf{u})$ is the total flux, consisting of the convective and viscous components, and $\mathbf{S}(\mathbf{u}, \nabla\mathbf{u})$ is a source term that is used when modelling turbulent flow. We note that boldface denotes a state vector, whereas the arrow denotes a spatial vector. The viscous flux is assumed linear in the state gradients, $\mathbf{G}_i(\mathbf{u}, \nabla\mathbf{u}) = -\mathbf{K}_{ij}(\mathbf{u})\partial_j\mathbf{u}$, where $\mathbf{K}_{ij}$ is the diffusivity tensor (Fidkowski 2007).

2.1. Discontinuous Galerkin (DG)

Denote by T_h the set of N_{elem} elements in a non-overlapping tessellation of the computational domain, Ω. As shown in Figure 1(a), in DG, the state is approximated by polynomials of order p on each element, with no continuity constraints across elements. Formally, $\mathbf{u}_h \in \boldsymbol{\mathcal{V}}_h = [\mathcal{V}_h]^s$, where $\mathcal{V}_h = \{u \in L_2(\Omega) : u|_{\Omega_e} \in \mathcal{P}^p \ \forall\ \Omega_e \in T_h\}$, and $\mathcal{P}^p$ denotes polynomials of maximum order p on the reference space of element Ω_e. The weak form of (1) follows from multiplying the equation by test functions in the same space, integrating by parts, and coupling elements via unique fluxes. The result states: find $\mathbf{u}_h \in \boldsymbol{\mathcal{V}}_h$ such that $\forall \mathbf{w}_h \in \boldsymbol{\mathcal{V}}_h$,

$$\begin{aligned} &-\int_{\Omega_e} \nabla\mathbf{w}_h^{\mathrm{T}} \cdot \vec{\mathbf{H}}\, \mathrm{d}\Omega + \int_{\partial\Omega_e} \mathbf{w}_h^{\mathrm{T}} \hat{\mathbf{H}} \cdot \vec{n}\, \mathrm{d}s \\ &\quad - \int_{\partial\Omega_e} \partial_i \mathbf{w}_h^{+T} \mathbf{K}_{ij}^{+} [\![\mathbf{u}_h]\!] + \int_{\Omega_e} \mathbf{w}^{\mathrm{T}} \mathbf{S}(\mathbf{u}, \nabla\mathbf{u})\, \mathrm{d}\Omega = 0, \end{aligned} \tag{2}$$

where $(\cdot)^{\mathrm{T}}$ denotes transpose, $(\cdot)^+, (\cdot)^-$ denote quantities taken from the element or its neighbour, respectively, $[\![\cdot]\!] = (\cdot)^+ - (\hat{\cdot})$, and $(\hat{\cdot})$ is the face average or boundary value. $\hat{\mathbf{H}} \cdot \vec{n}$ denotes the numerical normal flux on faces and includes the convective (upwind) and diffusive (jump) stabilizations. We use the Roe



(a) DG (b) HDG (c) EDG

Figure 1. Solution using standard and hybridized discontinuous Galerkin methods. (a) DG, (b) HDG and (c) EDG.

approximate Riemann solver (Roe 1981) for the convective flux, and the second form of Bassi and Rebay (BR2) (Bassi and Rebay 2000) for the viscous flux. Choosing a basis for the test and trial spaces yields a system of nonlinear equations, $\mathbf{R}(\mathbf{U}) = \mathbf{0}$, where $\mathbf{U} \in \mathbb{R}^{N_h}$ is the discrete state vector of basis function coefficients, N_h is the number of unknowns, and $\mathbf{R} \in \mathbb{R}^{N_h}$ is the discrete steady residual vector function of the state. The nonlinear system of equations is solved using the Newton-Raphson method with pseudo-time continuation and the generalised minimal residual (GMRES) method, preconditioned with an element-block Jacobi smoother (Fidkowski and Ceze 2016).

2.2. Hybridized and embedded discontinuous Galerkin (HDG and EDG)

The starting point for a hybridized discretization is the conversion of (1) to a system of first-order equations,

$$\vec{\mathbf{q}} - \nabla \mathbf{u} = \vec{\mathbf{0}}, \tag{3}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \vec{\mathbf{H}}(\mathbf{u}, \vec{\mathbf{q}}) + \mathbf{S}(\mathbf{u}, \vec{\mathbf{q}}) = \mathbf{0}, \tag{4}$$

where $\vec{\mathbf{q}}$ is an approximation of the state gradient. Multiplying these equations by test functions $\vec{\mathbf{v}} \in [\mathcal{V}_h]^{\dim}$, $\mathbf{w} \in \mathcal{V}_h$ and integrating by parts over an element Ω_e yields the weak form: find $\mathbf{u}_h \in \mathcal{V}_h$, and $\vec{\mathbf{q}} \in [\mathcal{V}_h]^{\dim}$, such that $\forall\, \vec{\mathbf{v}}_h \in [\mathcal{V}_h]^{\dim}$ and $\forall\, \mathbf{w}_h \in \mathcal{V}_h$,

$$\int_{\Omega_e} \vec{\mathbf{v}}_h^{\mathrm{T}} \cdot \vec{\mathbf{q}}_h \, \mathrm{d}\Omega + \int_{\Omega_e} \nabla \cdot \vec{\mathbf{v}}_h^{\mathrm{T}} \mathbf{u}_h \, \mathrm{d}\Omega - \int_{\partial\Omega_e} \vec{\mathbf{v}}_h^{\mathrm{T}} \cdot \vec{n} \hat{\mathbf{u}}_h \, \mathrm{d}s = 0, \tag{5}$$

$$\int_{\Omega_e} \mathbf{w}_h^{\mathrm{T}} \frac{\partial \mathbf{u}_h}{\partial t} \, \mathrm{d}\Omega - \int_{\Omega_e} \nabla \mathbf{w}_h^{\mathrm{T}} \cdot \vec{\mathbf{H}} \, \mathrm{d}\Omega + \int_{\partial\Omega_e} \mathbf{w}_h^{\mathrm{T}} \hat{\mathbf{H}} \cdot \vec{n} \, \mathrm{d}s + \int_{\Omega_e} \mathbf{w}_h^{\mathrm{T}} \mathbf{S}(\mathbf{u}_h, \vec{\mathbf{q}}_h) \, \mathrm{d}\Omega = 0, \tag{6}$$

where $\hat{\mathbf{u}}$ is a new independent unknown: the state approximated on faces of the mesh. Note that element degrees of freedom are coupled to the face degrees of freedom, but not to each other. The additional unknowns call for additional equations, which come from weak enforcement of flux continuity across faces,

$$\int_{\sigma_f} \boldsymbol{\mu}^{\mathrm{T}} \{\hat{\mathbf{H}} \cdot \vec{n}|_L + \hat{\mathbf{H}} \cdot \vec{n}|_R\} \, \mathrm{d}s = 0 \quad \forall\, \boldsymbol{\mu} \in \boldsymbol{\mathcal{M}}_h. \tag{7}$$

In this equation, $\boldsymbol{\mathcal{M}}_h$ denotes the order-p approximation space on the faces $\sigma_f \in F_h$ of the mesh: $\boldsymbol{\mathcal{M}}_h = [\mathcal{M}_h]^s$, where $\mathcal{M}_h = \{u \in L_2(\sigma_f) : u|_{\sigma_f} \in \mathcal{P}^p \ \forall\ \sigma_f \in F_h\}$, and the subscripts L and R refer to the two sides of a face. Note that F_h is the set of *interior* faces of the mesh. As shown in Figure 1, both HDG and EDG introduce $\hat{\mathbf{u}}_h$, with the difference that in EDG, the approximation space $\boldsymbol{\mathcal{M}}_h$ is continuous at mesh nodes (and edges in three dimensions). This leads to a large reduction in global degrees of freedom, particularly at low or moderate orders.

The fluxes in (6) depend only on the state and gradient inside the element and the face state, $\hat{\mathbf{H}} \cdot \vec{n} = \vec{\mathbf{H}}(\hat{\mathbf{u}}, \vec{\mathbf{q}}) \cdot \vec{n} + \boldsymbol{\tau}(\hat{\mathbf{u}}, \mathbf{u}, \vec{n})$, where $\boldsymbol{\tau}$ consists of a Roe-like convective stabilisation and a BR2 viscous stabilisation (Fidkowski 2016a). The entropy fix in the Roe flux and the jump-penalty term in BR2 are the same in the hybridized and standard DG discretizations. For the cases tested, no differences were observed in the stability properties of the various solvers.

Choosing bases for the trial/test spaces in Equations (5), (6), (7) gives a nonlinear system of equations, $\mathbf{R}^Q = \mathbf{0}$, $\mathbf{R}^U = \mathbf{0}$, $\mathbf{R}^\Lambda = \mathbf{0}$, with the Newton update system

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{Q} \\ \Delta\mathbf{U} \\ \Delta\boldsymbol{\Lambda} \end{bmatrix} + \begin{bmatrix} \mathbf{R}^Q \\ \mathbf{R}^U \\ \mathbf{R}^\Lambda \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{8}$$

where $\mathbf{Q}$, $\mathbf{U}$, and $\boldsymbol{\Lambda}$ are the discrete unknowns in the approximation of $\vec{\mathbf{q}}$, $\mathbf{u}$, and $\hat{\mathbf{u}}$, respectively. $[\mathbf{A}, \mathbf{B}; \mathbf{C}, \mathbf{D}]$ is the primal Jacobian matrix partitioned into element-interior and interface unknown blocks. Note that $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ contain both $\mathbf{Q}$ and $\mathbf{U}$ components, and $\mathbf{A}$ is element-wise block diagonal, and hence easily invertible.

Statically condensing out the element-interior states gives a smaller system for the face degrees of freedom,

$$\boldsymbol{\mathcal{K}} \Delta\boldsymbol{\Lambda} + \left(\mathbf{R}^\Lambda - \mathbf{C}\mathbf{A}^{-1}\left[\mathbf{R}^Q; \mathbf{R}^U\right]\right) = \mathbf{0}, \qquad \boldsymbol{\mathcal{K}} \equiv \mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}. \tag{9}$$

This global system is solved using GMRES preconditioned by a point-block Jacobi smoother, where a 'point' corresponds to a $s \times s$ block of unknowns

associated with one degree of freedom. Following the global solve for $\Delta\mathbf{\Lambda}$, an element-local back-solve yields the updates to $\mathbf{Q}$ and $\mathbf{U}$.

2.3. Adjoint discretization

For a scalar output J, the discrete adjoint $\mathbf{\Psi}$ is a vector of sensitivities of J to residual source perturbations, and the associated adjoint equation is $(\partial\mathbf{R}/\partial\mathbf{U})^{\mathrm{T}}\mathbf{\Psi} + (\partial J/\partial\mathbf{U})^{\mathrm{T}} = \mathbf{0}$. For optimal output convergence and accurate error estimates, the discretization and output must be defined appropriately to ensure consistency of the discrete adjoint with the continuous adjoint (Lu 2005; Hartmann 2007). In the present work, adjoint consistency has been verified through output convergence studies and error effectivity tests for all of the discretizations. We solve this equation for the DG discretization, using the same preconditioned GMRES approach as in the implicit primal solver. When computed on a fine discretization space, the adjoint provides a weight on residuals in a measurement of output error.

2.4. Degrees of freedom and matrix sparsity

On a given mesh, DG, HDG, and EDG will have different degree of freedom counts and residual Jacobian sparsity patterns. Figure 2 presents an example of the degree of freedom placement for $p = 2$ approximation on a ten-element triangular mesh. In HDG and EDG, we do not introduce $\hat{\mathbf{u}}$ on boundary faces, as the flux there is computed in the same way as in DG. The number of matrix nonzeros for EDG (177) is about half that of HDG (330) and a sixths that of DG (1080).

3. Output error estimation

We use output-based error estimates computed from adjoint solutions to drive anisotropic mesh adaptation in the DG discretization. An adjoint solution can be used to estimate the numerical error in the corresponding output of interest, J, through the adjoint-weighted residual (Becker and Rannacher 2001; Fidkowski and Darmofal 2011). Let H denote a coarse/current discretization space, and h a fine one, e.g. obtained by increasing the approximation order by one. Denote by $\mathbf{U}_h^H$ the state injected from the coarse to the fine space. Computing the fine-space residual with the injected state and weighting it by the fine-space adjoint gives an estimate of the output error between the coarse and fine spaces, $J_h(\mathbf{U}_h^H) - J_h(\mathbf{U}_h) \approx -\delta\mathbf{\Psi}_h^{\mathrm{T}}\mathbf{R}_h(\mathbf{U}_h^H)$. For the fine space, we increment the approximation order by one on each element and obtain the fine-space adjoint by solving exactly on this fine space. We obtain $\delta\mathbf{\Psi}_h$ an injection of the coarse-space adjoint. The error estimates involving element residuals can be localised to element (e) contributions, resulting in the error indicator $\mathcal{E}_{e0} \equiv |\delta\mathbf{\Psi}_{h,e}^{\mathrm{T}}\mathbf{R}_{h,e}(\mathbf{U}_h^H)|$, where the subscript e denotes restriction to element e. Note that the fine-space residual on an element depends in general on neighbouring states.

4. Adaptation

Estimates of the output error not only provide information about the accuracy of a solution, but can also drive mesh adaptation. A fair comparison of high-order solution can only be made with meshes

Colour online, B/W in print

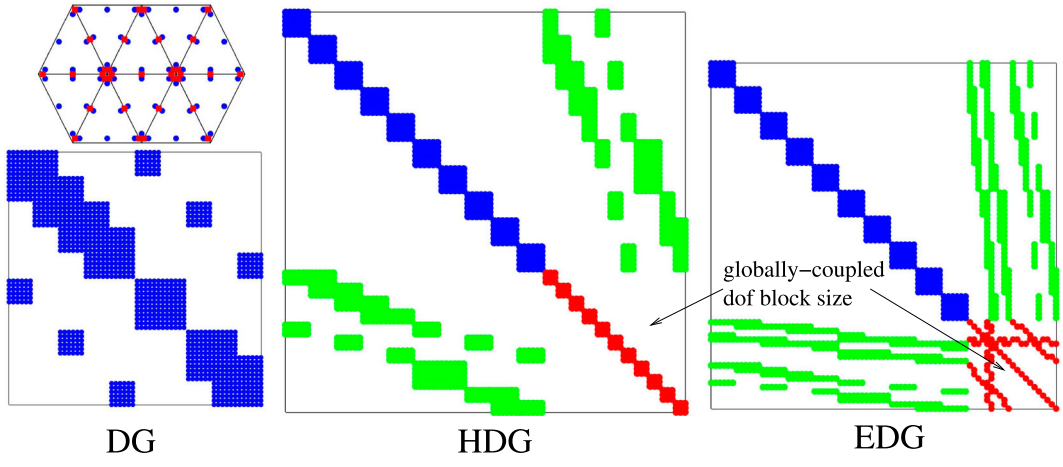
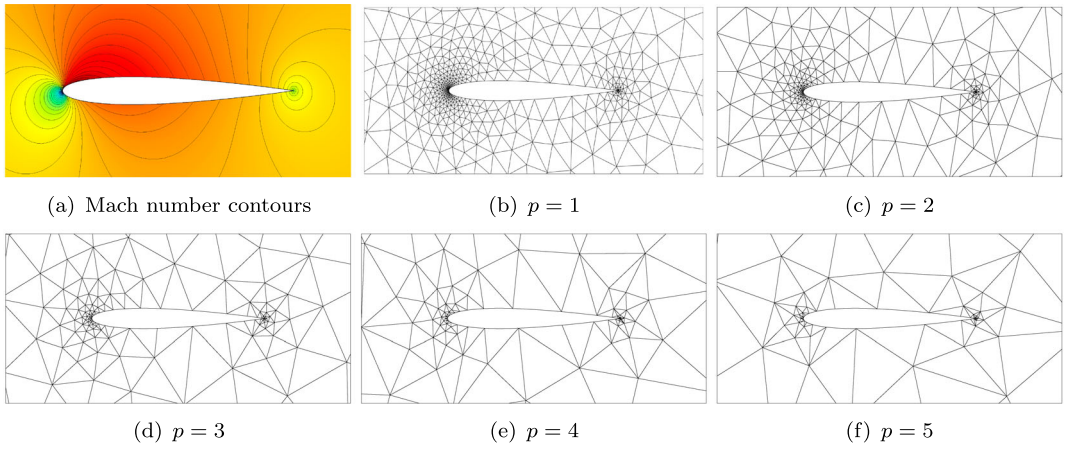


Figure 2. Degree of freedom placement (blue = DG, red = HDG) and residual Jacobian matrix sparsity patterns for a ten-element mesh. In EDG, face approximation unknowns are unique at nodes.

optimised to a particular order. In the present work, we assume that meshes optimised for the standard DG discretization are close to optimal for both HDG and EDG. We therefore generate an adapted sequence of meshes for DG, using an optimisation algorithm presented in Fidkowski (2016b), which builds on the work of Yano (Yano 2012). This algorithm is known as Mesh Optimization through Error Sampling and Synthesis (MOESS). It relies on a Riemannian metric field to encode information about the size and stretching of mesh elements, and thus enables anisotropic refinement. MOESS optimises the mesh to equidistribute the marginal error to cost ratio, where the error model is specific to each element and is identified through a sampling procedure. The references contain details of this method.

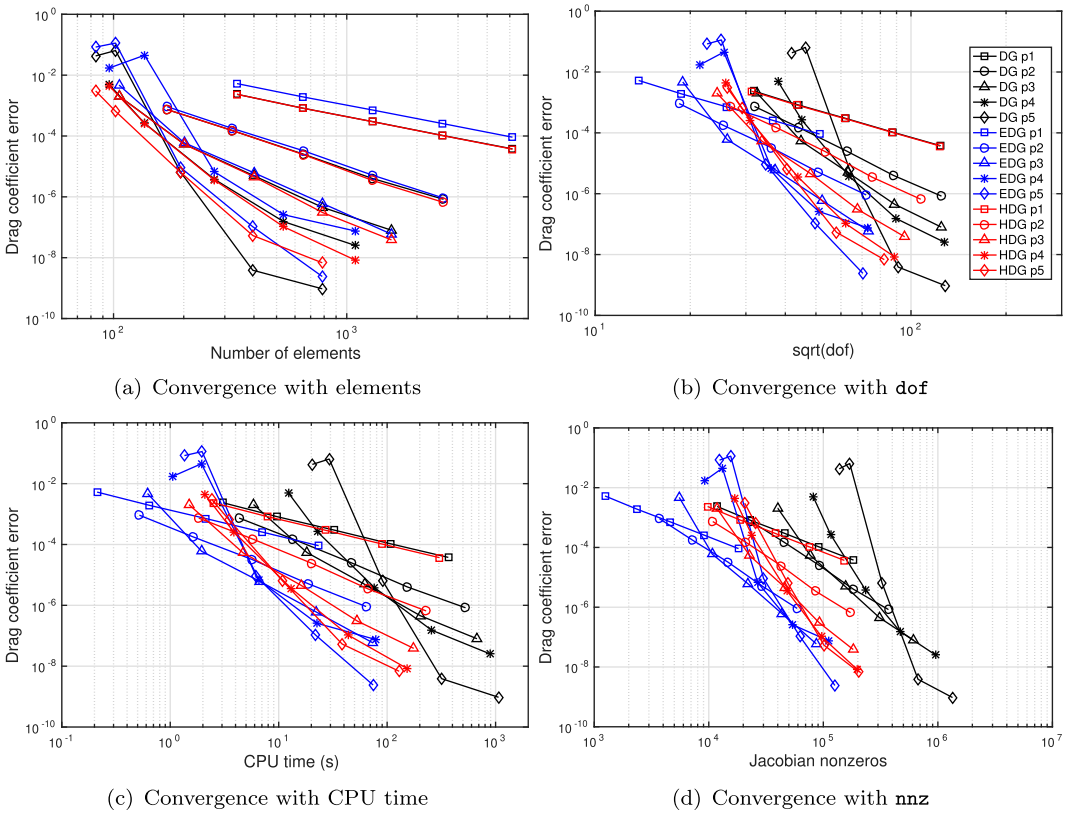


(a) Mach number contours (b) $p = 1$ (c) $p = 2$

(d) $p = 3$ (e) $p = 4$ (f) $p = 5$

Colour online, B/W in print

Figure 3. Inviscid flow: Mach contours and drag-adapted meshes for $\sim$ 4000 DG degrees of freedom. (a) Mach number contours, (b) $p = 1$, (c) $p = 2$, (d) $p = 3$, (e) $p = 4$ and (f) $p = 5$.



(a) Convergence with elements (b) Convergence with `dof`

(c) Convergence with CPU time (d) Convergence with `nnz`

Colour online, B/W in print

Figure 4. Inviscid flow: drag error convergence on adapted meshes plotted against various cost measures. (a) Convergence with elements, (b) Convergence with `dof`, (c) Convergence with CPU time and (d) Convergence with `nnz`.

5. Results

5.1. Inviscid flow

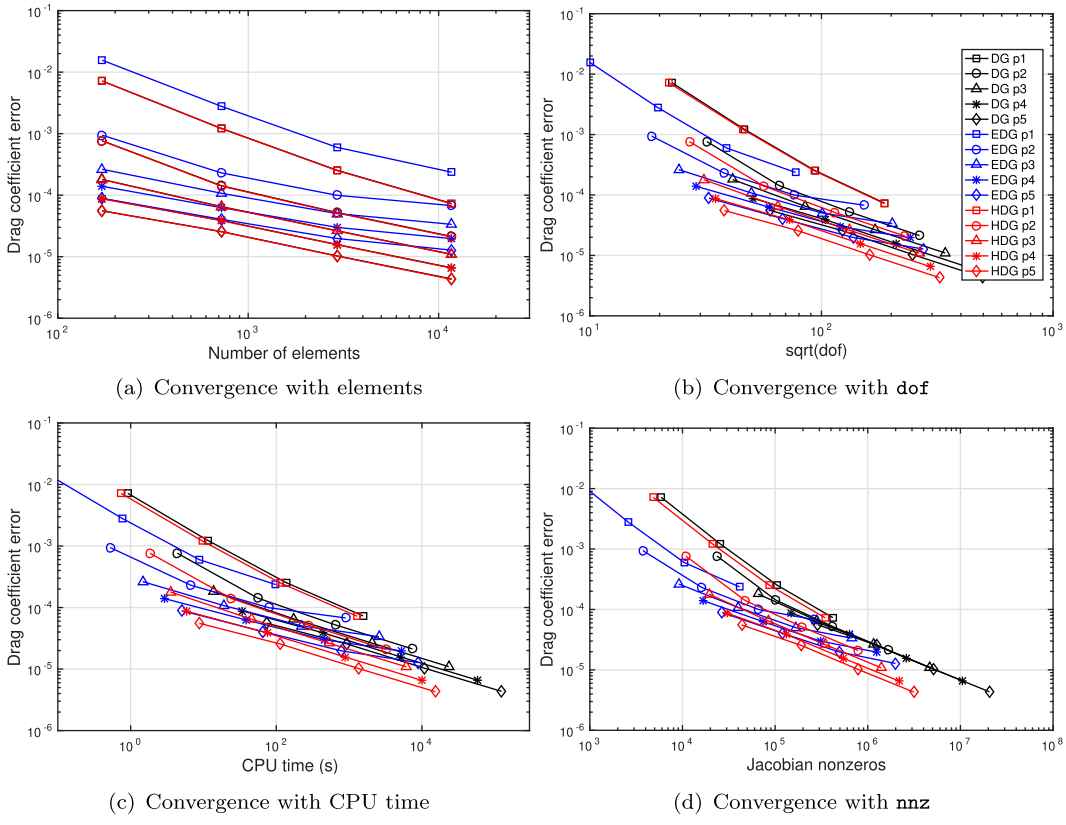
The first test case consists of a NACA 0012 airfoil in inviscid flow at $M = 0.5$ and $\alpha = 2^\circ$. Quintinc ($Q = 5$) elements approximate the geometry, and the farfield is 100 chord lengths away from the airfoil. We consider the prediction of drag using orders $p = 1-5$. For each order, we generate adapted meshes using DG at five target degrees of freedom, 1k, 2k, 4k, 8k, and 16k. To minimise scatter in the results, five adapted meshes from MOESS are used at each degree of freedom to generate an averaged output. Figure 3 shows the Mach contours and several adapted meshes. The key regions targeted for refinement are the leading and trailing edges. Figure 4 presents the convergence results of all three discretizations. Errors in the drag coefficient, computed relative to a finer truth solution, are shown versus several cost measures. Whereas EDG shows larger errors at lower orders, the comparable performance at higher orders combined with EDG's lower cost in memory (`dof` and Jacobian nonzeros) and CPU time make it the most attractive method considered.

To demonstrate the importance of comparing the methods on adapted meshes, Figure 5 repeats the convergence study on a sequence of uniformly-refined meshes, starting from one of the coarsest $p = 2$ adapted meshes. EDG now exhibits consistently higher errors, even at high p, for which its lower cost cannot compensate, with the result that HDG appears more attractive. Adapted meshes must therefore be used to fully realise the performance benefits of EDG.

5.2. Reynolds-averaged turbulent flow

The second test case consists of an RAE 2822 airfoil in Reynolds-averaged turbulent flow at $M = 0.3$, $\alpha = 2^\circ$, and $Re = 10^5$. The Spalart-Allmaras one equation model with a negative viscosity correction (Allmaras, Johnson, and Spalart 2012) is used. Quintinc ($Q = 5$) elements approximate the geometry, and the farfield is 100 chord lengths away from the airfoil. We consider the prediction of drag

Colour online, B/W in print



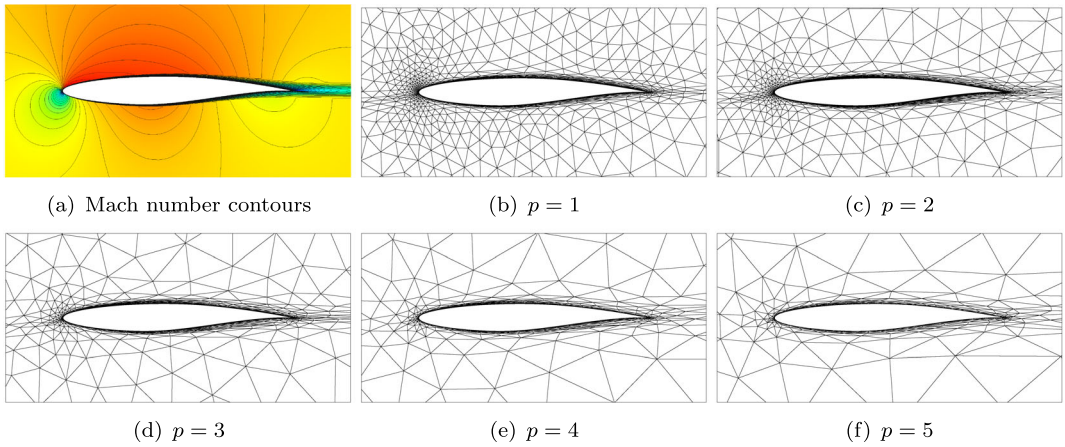
(a) Convergence with elements

(b) Convergence with `dof`

(c) Convergence with CPU time

(d) Convergence with `nnz`

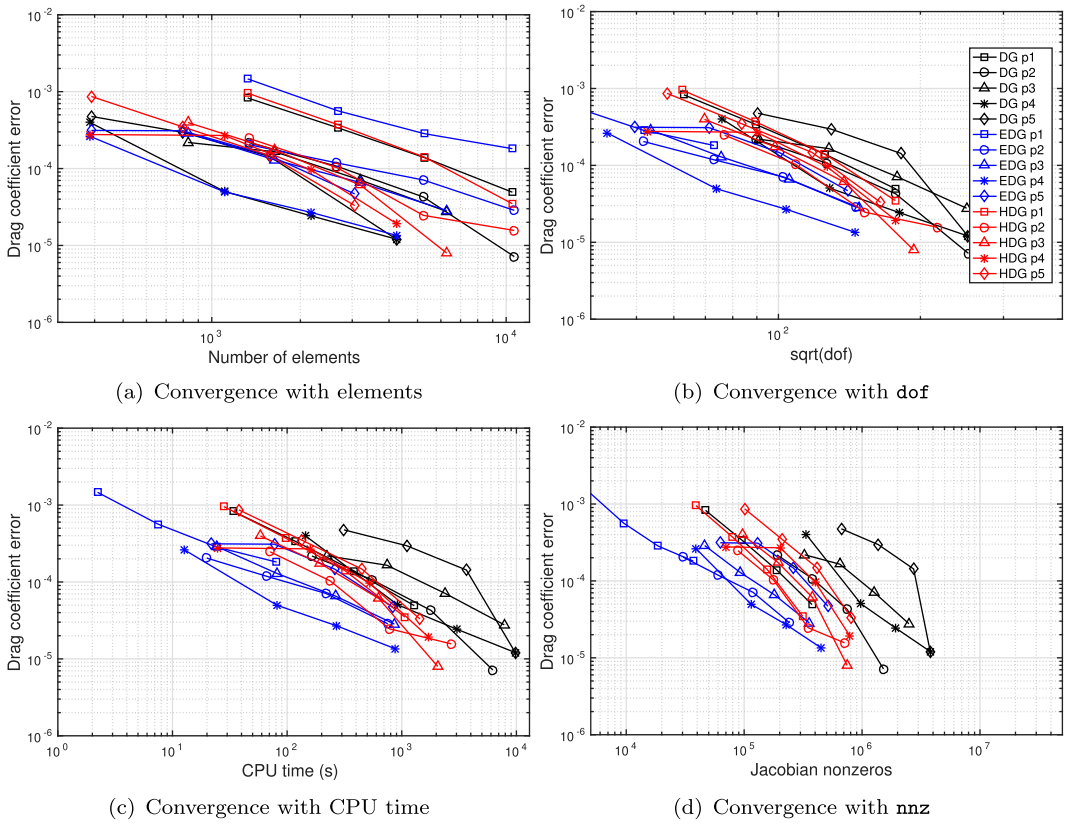
Figure 5. Inviscid flow: drag error convergence on a uniform-refinement mesh sequence. (a) Convergence with elements, (b) Convergence with `dof`, (c) Convergence with CPU time and (d) Convergence with `nnz`.



(a) Mach number contours (b) $p = 1$ (c) $p = 2$

(d) $p = 3$ (e) $p = 4$ (f) $p = 5$

Figure 6. RANS flow: Mach contours and drag-adapted meshes, all shown for $\sim$ 8000 DG degrees of freedom. (a) Mach number contours, (b) $p = 1$, (c) $p = 2$, (d) $p = 3$, (e) $p = 4$ and (f) $p = 5$.



(a) Convergence with elements (b) Convergence with `dof`

(c) Convergence with CPU time (d) Convergence with `nnz`

Figure 7. RANS flow: drag error convergence on adapted meshes plotted against various cost measures. (a) Convergence with elements, (b) Convergence with `dof`, (c) Convergence with CPU time and (d) Convergence with `nnz`.

using orders $p = 1-5$. For each order, we generate adapted meshes at 4k, 8k, 16k, 32k, and 64k `dof`. Figure 6 shows the Mach contours and several adapted meshes. Figure 7 presents the drag convergence results. As in the inviscid case, we see higher errors for EDG at lower orders, which diminish at higher orders and, once memory and CPU time costs are factored in, make EDG again the most attractive discretization.

6. Conclusions

This paper presents a comparison of standard and hybridized discontinuous Galerkin (DG) methods on output-based adapted meshes. Both hybrid (HDG) and embedded (EDG) variants of DG are considered, and these offer cost savings in globally-coupled degrees of freedom and Jacobian size relative to standard DG. EDG additionally eliminates duplicate

degrees of freedom at nodes and yields a much smaller system than DG and HDG. For the inviscid and RANS cases tested, at moderate and high orders, EDG delivers the lowest error versus memory and computational time cost measures. Adapted meshes are crucial to realising this advantage, which can disappear on sub-optimal uniformly-refined meshes. Future work will consider discretization-specific mesh optimisation algorithms, to further reduce the error levels for a given cost.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the Department of Energy (DE-FG02-13ER26146/DE-SC0010341) and the Boeing Company, technical monitor Dr. Mori Mani.

ORCID

K.J. Fidkowski http://orcid.org/0000-0002-5106-136X

References

Allmaras, S. R., F. T. Johnson, and P. R. Spalart. 2012. "Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model." Seventh international conference on computational fluid dynamics (ICCFD7) 1902, Hawaii, USA.

Bassi, F., and S. Rebay. 2000. "GMRES Discontinuous Galerkin Solution of the Compressible Navier-Stokes Equations." In *Discontinuous Galerkin Methods: Theory, Computation and Applications*, edited by Bernardo Cockburn, George Karniadakis, and Chi-Wang Shu, 197–208. Berlin: Springer.

Becker, R., and R. Rannacher. 2001. "An Optimal Control Approach to a Posteriori Error Estimation in Finite Element Methods." In *Acta Numerica*, edited by A. Iserles, 1–102. Cambridge: Cambridge University Press.

Burgess, Nicholas K., and Dimitri J. Mavriplis. 2011. "An *hp*-Adaptive Discontinuous Galerkin, Solver for Aerodynamic Flows on Mixed-Element Meshes." AIAA paper 2011–490, Orlando, Florida, USA.

Cockburn, Bernardo, Jayadeep Gopalakrishnan, and Raytcho Lazarov. 2009. "Unified Hybridization of Discontinuous Galerkin, Mixed, and Continuous Galerkin Methods for Second Order Elliptic Problems." *SIAM Journal on Numerical Analysis* 47 (2): 1319–1365.

Cockburn, Bernardo, and Chi-Wang Shu. 2001. "Runge-Kutta Discontinuous Galerkin Methods for Convection-dominated Problems." *Journal of Scientific Computing* 16 (3): 173–261.

Fernandez, P., N. C. Nguyen, and J. Peraire. 2017. "The hybridized Discontinuous Galerkin Method for Implicit Large-Eddy Simulation of Transitional Turbulent Flows." *Journal of Computational Physics* 336: 308–329.

Fidkowski, Krzysztof J. 2007. "A Simplex Cut-Cell Adaptive Method for High–order Discretizations of the Compressible Navier-Stokes Equations." PhD diss., Massachusetts Institute of Technology, Cambridge, Massachusetts. http://hdl.handle.net/1721.1/39701.

Fidkowski, Krzysztof J. 2016a. "A Hybridized Discontinuous Galerkin Method on Mapped Deforming Domains." *Computers and Fluids* 139 (5): 80–91.

Fidkowski, Krzysztof J. 2016b. "A Local Sampling Approach to Anisotropic Metric-Based Mesh Optimization." AIAA paper 2016–0835, San Diego, California, USA.

Fidkowski, Krzysztof J., and Marco A. Ceze. 2016. "High-Order Output-Based Adaptive Simulations of Turbulent Flow Over a Three Dimensionsional Bump." AIAA paper 2015–0862, San Diego, California, USA.

Fidkowski, Krzysztof J., and David L. Darmofal. 2011. "Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics." *American Institute of Aeronautics and Astronautics Journal* 49 (4): 673–694.

Franciolini, Matteo, Krzysztof J. Fidkowski, and Andrea Crivellini. 2018. "Efficient Discontinuous Galerkin Implementations and Preconditioners for Implicit Unsteady Compressible Flow Simulations." *arXiv preprint arXiv:1812.04789*.

Hartmann, Ralf. 2007. "Adjoint Consistency Analysis of Discontinuous Galerkin Discretizations." *SIAM Journal on Numerical Analysis* 45 (6): 2671–2696.

Lu, James. 2005. "An a Posteriori Error Control Framework for Adaptive Precision Optimization Using Discontinuous Galerkin Finite Element Method." PhD diss., Massachusetts Institute of Technology, Cambridge, Massachusetts.

Nguyen, N. C., J. Peraire, and B. Cockburn. 2009. "An Implicit High-Order Hybridizable Discontinuous Galerkin, Method for Linear Convection-Diffusion Equations." *Journal of Computational Physics* 228: 3232–3254.

Peraire, J., N. C. Nguyen, and B. Cockburn. 2011. "An Embedded Discontinuous Galerkin Method for the Compressible Euler and Navier-Stokes Equations." AIAA paper 2011–3228, Honolulu, Hawaii, USA.

Reed, W., and T. Hill. 1973. "Triangular Mesh Methods for the Neutron Transport Equation." Los Alamos Scientific Laboratory Technical Report LA-UR-73-479.

Roe, P. L. 1981. "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes." *Journal of Computational Physics* 43: 357–372.

Venditti, D. A., and D. L. Darmofal. 2003. "Anisotropic Grid Adaptation for Functional Outputs: Application to Two-dimensional Viscous Flows." *Journal of Computational Physics* 187 (1): 22–46.

Venkatakrishnan, V., S. R. Allmaras, D. S. Kamenetskii, and F. T. Johnson. 2003. "Higher Order Schemes for the Compressible Navier-Stokes Equations." AIAA Paper 2003–3987, Miami, Florida, USA.

Woopen, Michael, Aravind Balan, Georg May, and Jochen Schütz. 2014. "A Comparison of Hybridized and Standard DG Methods for Target-based hp-adaptive Simulation of Compressible Flow." *Computers & Fluids* 98: 3–16.

Yano, Masayuki. 2012. "An Optimization Framework for Adaptive Higher-Order Discretizations of Partial Differential Equations on Anisotropic Simplex Meshes." PhD diss., Massachusetts Institute of Technology, Cambridge, Massachusetts.