AIAA AVIATION Forum
5-9 June 2017, Denver, Colorado
23rd AIAA Computational Fluid Dynamics Conference

10.2514/6.2017-3444

A New Family of Discontinuous Galerkin Schemes for Diffusion Problems: Interface Gradient Recovery

Philip E. Johnson* & Eric Johnsen†

Mechanical Engineering Department, University of Michigan, Ann Arbor, MI 48109

Novel Discontinuous Galerkin formulations for parabolic partial differential equations, such as the unsteady diffusion equation, are introduced. Fourier analysis of the various schemes is presented to determine order of accuracy and spectral radius. Additionally, computational examples for the 1D scalar diffusion equation are presented. The method's performance for 2D problems is also demonstrated. Some members of the family exhibit favorable performance in comparison to commonly used methods with regard to accuracy and computational efficiency. Additionally, the method can be adapted to fit within the Flux Reconstruction description of Huynh for diffusion problems; we explore the corresponding 1D formulation.

I. Introduction

The Discontinuous Galerkin (DG) Finite Element method and its application to flow physics problems have been the focus of much research in recent years, due to the method's excellent performance in the discretization of hyperbolic partial differential equations (PDEs).[1] The method exbihits high parallel efficiency and can naturally be extended to arbitrarily high orders of accuracy. Several variants are available for maintaining numerical stability even in the presence of physical discontinuities, such as the shock waves which commonly arise in solutions to the Euler equations for inviscid compressible flow. In the case of smooth solutions, there is no need for artificial stabilization techniques. However, the method in its raw form is poorly suited to parabolic/elliptic PDEs, and this deficiency has been a barrier to its widespread application for general advection-diffusion systems, such as the compressible Navier-Stokes equations. Many modifications to the conventional DG method have been proposed to remedy this deficiency, but these methods typically exhibit certain undesirable features, such as severe stability constraints, sub-optimal convergence in the global error norm, and ambiguously bounded stabilization parameters.

A unique scheme is the Recovery DG method of Van Leer & Nomura.[2] Compared to other DG approaches for handling parabolic/elliptic PDEs, Recovery DG is characterized by high order of accuracy, small spectral radius, and no ambiguously bounded stabilization parameters. The general concept of recovery has assisted in the development of similar schemes, for example the reconstruction-based DG method of Luo et al,[3] the enhanced stability recovery scheme of Ferrero et al.,[4] and the modified recovery scheme (MRDG-1x) of French et al.[5] Additionally, Recovery DG has demonstrated competence when applied to the viscous fluxes of the compressible Navier-Stokes equations in Direct Numerical Simulation of compressible turbulence by Johnsen et al.[6,7] For a given interface in the computational domain, the recovery process builds a smooth polynomial function (the "recovered" solution) across the union of the neighboring elements, maximizing the use of the solution data associated with each element and providing an unambiguous and accurate approximation for both the solution and its gradient along the interface. The effect of the recovery procedure is demonstrated

*Graduate Student, University of Michigan, 1231 Beal Ave. 2043 Walter E. Lay Automotive Laboratory, Ann Arbor, MI 48109-2133. AIAA Student Member

†Assistant Professor, University of Michigan, 1231 Beal Ave. 2043 Walter E. Lay Automotive Laboratory, Ann Arbor, MI 48109-2133. AIAA Senior Member

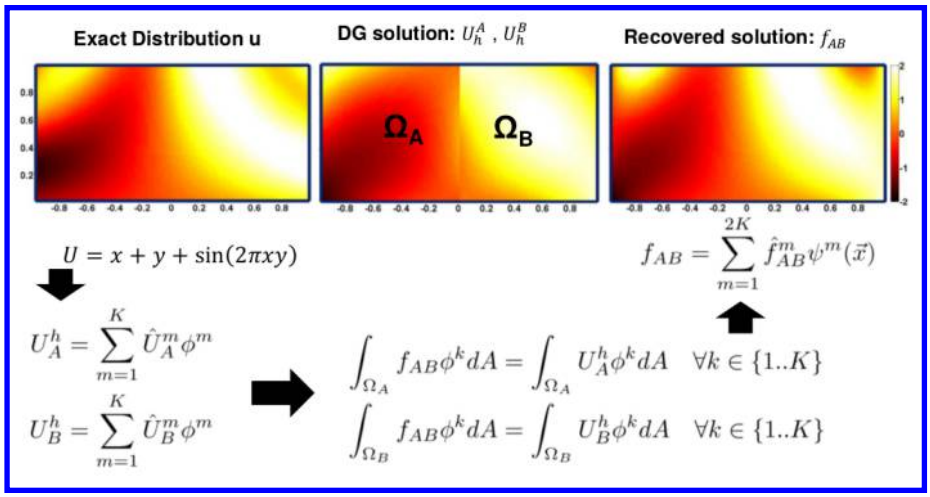


Figure 1: The Recovery process in 2D for a $p2$ discretization.

in Figure 1 for a $p2$ discretization, taken from our presentation at AIAA Aviation 2016.[8] The recovered solution corresponding to the interface $\mathcal{I}_{AB}$ between two neighboring elements Ω_A and Ω_B, denoted f_{AB}, recovers a very good approximation for a smooth exact solution by applying the recovery procedure to the discontinuous solution representation of U_A^h and U_B^h over $\Omega_A \cup \Omega_B$.

The Recovery DG method, extended to multidimensional problems by Lo & Van Leer,[9] shows great promise for purely parabolic PDEs in multiple dimensions, including initial boundary value problems (IBVPs) with time-dependent Dirichlet conditions.[8] However, for combined advection-diffusion problems, where the Recovery DG scheme is employed to handle the diffusive fluxes, it is difficult to design a complete advection-diffusion DG scheme that captures the advective fluxes with the same accuracy that Recovery DG achieves for the diffusive terms. While the accuracy of the combined method is constrained by the standard DG method's order of accuracy in capturing the advective fluxes, the cost and complication of the method is dominated by the Recovery DG scheme for the diffusive fluxes. This pair of performance bottlenecks is undesirable.

With these issues in mind, the aim of our recent studies has been to design new DG schemes for diffusion problems that benefit from the high-order reconstruction provided by the recovery concept while avoiding the complications of Lo & Van Leer's[9] Recovery DG formulation. Our umbrella term for this new family of schemes is "Interface Gradient Recovery," abbreviated IGR. Most of the schemes in this family maintain smaller spectral radii than presently popular DG schemes for diffusion. Simplicity is maintained by starting from the "mixed form" approach for parabolic PDEs, which is familiar to many DG practitioners. Within the mixed form, the IGR family is distinguished by the use of recovery for handling interface terms.

The focus of this work is the analysis and performance of the scheme for time-dependent diffusion problems. In section II, the scheme will be described for the multidimensional and 1D cases. Then, we show results from Fourier analysis, providing comparisons in terms of spectral radius and order of accuracy. Next, computational demonstrations are shown, focusing on the convergence rates in multiple error norms. Then, we detour from the DG framework to show how the IGR family can be applied to the Flux Reconstruction (FR) method before making concluding remarks.

II. The Interface Gradient Recovery Family

Before introducing the members of the Interface Gradient Recovery (IGR) family, we review the mixed form approach for parabolic/elliptic PDEs, focusing on the simple 1D case. Let the physical domain $\mathbf{\Omega} = \{x \in [0, L]\}$ be partitioned into M non-overlapping finite elements, such that $\mathbf{\Omega} = \cup_{e=0}^{M-1} \Omega_e$, where $\Omega_e = [x_e, x_{e+1}]$, $x_0 = 0$, $x_M = L$, and $x_{j-1} < x_j \quad \forall j$. To eliminate confusion, the exponential function of an input y in this

work will always be referred to as $\exp(y)$, and the lowercase e is reserved to refer to element addresses.

The global DG solution U^h is represented as a polynomial of degree p over each element Ω_e with no continuity restriction: $U^h(x) = \sum_{k=0}^{p} V_e^k(x)\ \hat{U}_e^k$ for all x in element Ω_e. In the expansion, $\hat{U}_e^k$ is the k^{th} DG degree of freedom (DOF) in element Ω_e. The shape function V_e^k is the k^{th} basis function with support on Ω_e; each element supports its own set of $p+1$ shape functions, and each shape function is nonzero only on its supporting element. In this work, it is assumed that each element uses the same polynomial degree p. For each interface location x_j, the DG solution U^h is in general multi-valued, as both of the neighboring elements that share the interface contribute some value for U^h (often called the trace) along the interface.

It is convenient to introduce a reference element $\Omega_{ref} = \{\xi \in [0,1]\}$. Let $\boldsymbol{\phi} = \{\phi^0(\xi), \phi^1(\xi), ...\phi^p(\xi)\}$ be a degree p polynomial basis defined over the unit domain, and for a given element, take $\xi(x) = \frac{1}{\Delta x_e}(x - x_e)$, where $\Delta x_e = x_{e+1} - x_e$. For each element, the shape functions are $V_e^k(x) = \phi^k(\xi(x))$. Now, for a given element Ω_e, the DG solution can be rewritten $U_e^h(x) = \sum_{k=0}^{p} \phi^k(\xi)\ \hat{U}_e^k$. Instead of the polynomial degree p, we can also desribe the solution by the DOF count per element, denoted K; in the 1D case, $K = p+1$. When a shape function ϕ^k appears in an integral over an element Ω_e or along its boundary $\partial\Omega_e$, it is assumed to be supported over the element.

Given an unsteady PDE system with a diffusive flux function $\mathcal{G}$,

$$\frac{\partial}{\partial t} U = \nabla \cdot \mathcal{G}(U, \nabla U), \tag{1}$$

to be satisfied in the weak sense (2a) over each element Ω_e, the DG weak form for both the general case (2b) and the simple 1D case (2c) is presented below; we require it to be satisfied for all Ω_e in the domain $\boldsymbol{\Omega}$.

$$\text{General}: \quad \int_{\Omega_e} \phi^k \frac{\partial}{\partial t} U^h d\mathbf{x} = \int_{\Omega_e} \phi^k \nabla \cdot \mathcal{G}(U^h, \nabla U^h) d\mathbf{x} \quad \forall k \in \{0, 1, ..., K-1\} \tag{2a}$$

$$\begin{aligned} \text{General}: \quad \int_{\Omega_e} \phi^k \frac{\partial}{\partial t} U^h d\mathbf{x} = &\int_{\partial\Omega_e} \phi^k (\mathcal{G}(\tilde{U}, \tilde{\sigma}) \cdot n^-) ds \\ &- \int_{\Omega_e} (\nabla \phi^k) \cdot (\mathcal{G}(U^h, \sigma)) d\mathbf{x} \quad \forall k \in \{0, 1, ..., K-1\} \end{aligned} \tag{2b}$$

$$\text{1D}: \quad \int_{\Omega_e} \phi^k \frac{\partial}{\partial t} U^h dx = [\phi^k\ \mathcal{G}(\tilde{U}, \tilde{\sigma})]_L^R - \int_{\Omega_e} (\nabla \phi^k) \cdot (\mathcal{G}(U^h, \sigma)) dx \quad \forall k \in \{0, 1, ..., p\} \tag{2c}$$

The term n^- is the outward normal along the boundary of an element, and the integration coordinate s traverses the perimeter of the element. In the 1D case, the surface integral around the element simplifies to the difference between a right-edge term and a left-edge term for each basis function within the element. All integrals are evaluated using Gauss-Legendre quadrature; this task requires that the flux at each quadrature point be populated using appropriately accurate approximations for U and ∇U.

In this work, the mixed form approach (well described in Arnold et al.[10]) is taken, where the auxiliary variable σ is introduced to track the gradient of the DG solution U^h. Specifically, for each element Ω_e in the domain $\boldsymbol{\Omega}$, $\nabla U_e^h \approx \sigma_e(x) = \sum_{k=0}^{p} \phi^k(\xi)\ \hat{\sigma}_e^k$. The auxiliary variable is an approximation for the gradient, which has one component for each spatial dimension, so while ϕ is a scalar, each coefficient $\hat{\sigma}_e^k$ has one component per spatial dimension. Over element interiors, the auxiliary variable is used to define the gradient for the diffusive flux function $\mathcal{G}(U, \nabla U)$; it is a finite-dimensional projection of ∇U^h, constrained to be indistinguishable from the gradient of the global DG solution U^h in the weak sense (3a). The enforcement of this condition (3b) takes advantage of the same integration-by-parts step used to form the usual DG weak form (2b). As with the description of the DG weak form, our description of the weak form for the auxiliary variable includes the general form, applicable to multidimensional problems (3b), and the simple 1D case (3c):

$$\text{General}: \quad \int_{\Omega_e} \phi^k \ \sigma_e d\mathbf{x} = \int_{\Omega_e} \phi^k \ \nabla U^h d\mathbf{x} \quad \forall k \in \{0, 1, ..., K-1\} \tag{3a}$$

$$\begin{aligned}\text{General}: \quad \int_{\Omega_e} \phi^k \ \sigma_e d\mathbf{x} &= \int_{\partial\Omega_e} \phi^k (\tilde{U} \cdot n^-) ds \\ &\quad - \int_{\Omega_e} U_e^h \ \nabla \phi^k d\mathbf{x} \quad \forall k \in \{0, 1, ..., K-1\}\end{aligned} \tag{3b}$$

$$\text{1D}: \quad \int_{\Omega_e} \phi^k \ \sigma_e dx = [\phi^k \ \tilde{U}]_L^R - \int_{\Omega_e} U_e^h \ \nabla \phi^k dx \quad \forall k \in \{0, 1, ..., p\} \tag{3c}$$

The DG method for the diffusive conservation law is completely described once the interface scalar flux $\tilde{U}$ and interface gradient $\tilde{\sigma}$ are defined. In this work, the interface scalar flux and interface gradient will also be referred to as the "common value" and "common gradient" of an interface, respectively. For a list of common value and common gradient choices corresponding to various DG schemes for diffusion, see Arnold et al.[10] The auxiliary variable in each element is constrained partially by the DG solution in the local element (3c); it is also influenced by the contributions of the common values $\tilde{U}$ on the interfaces of Ω_e, allowing for some exchange of information between elements.

A few special operators are used to define the interface scalar flux and gradient; these operators include the Recovery operator $\mathcal{R}$, the Generalized Binary Recovery operator $\mathcal{B}$, and the Localized Gradient Recovery operator $\mathcal{L}^\chi$. All operators recover a unique approximation for either $\tilde{U}$ or $\tilde{\sigma}$ on an interface using information from the two neighboring elements. Table 1 lists the members of the IGR family and distinguishes them according to the strategies employed for calculating $\tilde{U}$ and $\tilde{\sigma}$. Along an interface of an element Ω_e, y^+ indicates that the interface value for y is set from the neighbor element, while y^- indicates that the interface value for y is set from Ω_e. For a given interface, y_A and y_B denote the interface traces of the discontinuous polynomial approximations for the quantity y from the left and right elements, respectively. Following previous convention in DG methods, $\{\{y\}\} = \frac{1}{2}(y_A + y_B)$ indicates the average of the traces from the neighboring elements. The Local DG (LDG) method of Cockburn & Shu[11] (with one-sided fluxes, denoted LDG-OS) is also shown for comparison. The first seven IGR schemes, labelled GR-I through GR-VII, have non-compact stencils; only the special Compact Gradient Recovery (CGR-I) scheme features a compact stencil.

It remains to define the Recovery ($\mathcal{R}$), Generalized Binary Recovery ($\mathcal{B}$), and Localized Gradient Recovery ($\mathcal{L}^\chi$) operators. The Recovery operator is actually a special case of the Generalized Binary Recovery operator, so it suffices here to describe $\mathcal{B}$ and $\mathcal{L}^\chi$. Table 2 summarizes the operators. The Localized Gradient Recovery operator $\mathcal{L}^\chi$ is unique in that it takes the DOF $\hat{\mathbf{U}}$ of the two neighboring elements as an input and returns the interface gradient $\tilde{\sigma}$ instead of returning the interface's common value $\tilde{U}$.

Our definition of the Generalized Binary Recovery (GBR) operator is inspired by the interface-centered reconstruction schemes of Khieu & Johnsen,[12] specifically the icb$p$[0] schemes. Consider two elements Ω_A

Table 1: Configurations of DG schemes for diffusion.

Method	$\tilde{U}$	$\tilde{\sigma}$
GR-I	$\{\{U\}\}$	$\mathcal{R}(\sigma_A, \sigma_B)$
GR-II	$\mathcal{R}(U_A, U_B)$	$\mathcal{R}(\sigma_A, \sigma_B)$
GR-III	$\mathcal{R}(U_A, U_B)$	$\{\{\sigma\}\}$
GR-IV	U_A	$\mathcal{R}(\sigma_A, \sigma_B)$
GR-V	U^+	$\mathcal{R}(\sigma_A, \sigma_B)$
GR-VI	$\mathcal{B}_K^1(U_A, U_B)$	$\mathcal{B}_1^K(\sigma_A, \sigma_B)$
GR-VII	U_A	$\mathcal{B}_K^1(\sigma_A, \sigma_B)$
CGR-I	$\mathcal{R}(U_A, U_B)$	$\mathcal{L}^\chi(U_A, U_B)$
LDG-OS	U_A	σ_B

and Ω_B sharing an interface. Additionally, for the special 1D case, let Ω_A be the left element and Ω_B the right element. Define a reference coordinate r for the interface, such that $r = 0$ at the interface; the recovery basis $\boldsymbol{\Psi}$ is defined over $\Omega_A \cup \Omega_B$ using this interface coordinate. Let f denote the recovered solution across the entire union of the two elements; our specific interest is $f(0)$, the value at the interface. Assuming now that the two elements are of uniform width Δx, it is convenient to scale r such that $r = -1$ at the left edge of Ω_A and $r = 1$ at the right edge of Ω_B; excellent matrix conditioning is achieved by taking the shifted (unit domain) Legendre polynomials for the DG basis $\boldsymbol{\phi}$ and the bi-unit domain Legendre polynomials for the Recovery basis $\boldsymbol{\Psi}$. Since the GBR operation is the same for either $\tilde{U}$ or $\tilde{\sigma}$, the only difference being the input information, the procedure is described only in the context of recovering $\tilde{U}$:

$$f(r) = \sum_{n=0}^{C_A+C_B-1} \psi^n(r)\ \hat{f}^n \tag{4a}$$

$$\int_{\Omega_A} \psi^k\ f dx = \int_{\Omega_A} \psi^k\ U_A^h dx \quad \forall k \in \{0, 1, ..., C_A - 1\} \tag{4b}$$

$$\int_{\Omega_B} \psi^k\ f dx = \int_{\Omega_B} \psi^k\ U_B^h dx \quad \forall k \in \{0, 1, ..., C_B - 1\} \tag{4c}$$

The recovered solution f is required to be weakly equivalent to U_A^h over Ω_A with respect to C_A test functions and weakly equivalent to U_B^h over Ω_B with respect to C_B test functions. The recovery basis $\boldsymbol{\Psi}$ for an interface must be a degree $C_A + C_B - 1$ polynomial basis. Our implementation in the DG framework uses a hierarchical basis for $\boldsymbol{\Psi}$, and the lowest possible modes ($\psi = 1$, $\psi = r$, $\psi = r^2$, etc.) are taken for the $\{0, 1, ..., C\}$ testing functions over each element. It is possible to use instead a nodal basis for $\boldsymbol{\Psi}$, and we choose this configuration to implement recovery in the Flux Reconstruction[13] (FR) framework. The recovery constraints (4) are reorganized into matrix-vector form, such that for a given interface, the set of recovery coefficients $\hat{\mathbf{f}}$ can be calculated from the DOF vectors $\hat{\mathbf{U}}_A$ and $\hat{\mathbf{U}}_B$ using a pre-computed $(C_A + C_B) \times 2K$ matrix. Then, the coefficients $\hat{\mathbf{f}}$ are used to form $f(0) = \sum_{n=0}^{C_A+C_B-1} \psi^n(0)\ \hat{f}^n$, yielding a unique and unambiguous approximation for $\tilde{U}$ on the interface.

The description of the GBR operator is now concluded. The Recovery operator, originally designed by Van Leer & Nomura,[2] is the special case of $\mathcal{B}$ where K constraints are taken over each of the two elements: $\mathcal{R}(\sigma_A, \sigma_B) = \mathcal{B}_K^K(\sigma_A, \sigma_B)$. Additionally, for the full Recovery operator, instead of using the recovery basis $\boldsymbol{\psi}$, it is better to use the DG basis functions $\boldsymbol{\phi}$ of the two neighboring elements as the test functions in the constraint equations (4).

The χ-dependent Localized Gradient Recovery operator, abbreviated LGR and denoted $\mathcal{L}^\chi$, has been designed to maintain the advantage of a compact stencil. Other versions of IGR, because of how the recovery operators are applied, use a non-compact stencil. The non-compact stencil arises from the calculation of $\tilde{\sigma}$ on each interface. Specifically, if $\tilde{\sigma}$ is calculated using σ_e and σ_{e+1} from two neighboring elements, then $\tilde{\sigma}$ involves information from four different elements. The fix for this issue is clear: in order to calculate the interface gradient $\tilde{\sigma}$ between two elements Ω_e and Ω_{e+1}, special gradient approximations must be formed in each element using only information from Ω_e and Ω_{e+1}. Then, these approximations can be combined to yield the interface gradient $\tilde{\sigma}$.

To achieve a compact stencil, let $g_e^R(\xi) = \sum_{n=0}^{p} \phi^n(\xi)\ \hat{g}_e^{R,n}$ be a special gradient reconstruction in element Ω_e, known as the right-connected gradient. In forming this approximation, information from Ω_{e+1}

Table 2: Interface operators.

Operator	Input	Output	Constraints, Ω_A	Constraints, Ω_B
$\mathcal{R}(U_A, U_B)$	$\hat{\mathbf{U}}_A$, $\hat{\mathbf{U}}_B$	$\tilde{U}$	K	K
$\mathcal{B}_{C_A}^{C_B}(U_A, U_B)$	$\hat{\mathbf{U}}_A$, $\hat{\mathbf{U}}_B$	$\tilde{U}$	C_A	C_B
$\mathcal{L}^\chi(U_A, U_B)$	$\hat{\mathbf{U}}_A$, $\hat{\mathbf{U}}_B$	$\tilde{\sigma}$	K	K

is imported through the recovery process on the right edge of Ω_e, located at x_{e+1}, but the trace from Ω_e is used instead of $\tilde{U}$ on the left edge (5a). A similar operation is performed for Ω_{e+1}, except that the left-connected gradient g^L_{e+1} uses recovery for its left edge and the interior trace for the right edge. The factor χ is necessary for robustness in 2D applications. So far, it appears that for 2D cases, setting $\chi = 1$ is best for Cartesian grids with purely Laplacian diffusion while $\chi = 2$ is best for general cases.

$$\int_{\Omega_e} \phi^k \; g^R_e dx = \chi \; [\phi^k(\mathcal{R}(U^h_e, U^h_{e+1}) - U^h_e(1))]_{x_{e+1}} + \int_{\partial\Omega_e} (\phi^k \; U^h_e) \cdot n^- ds - \int_{\Omega_e} U^h_e \; \nabla\phi^k dx \quad \forall k \in \{0, 1, ..., p\} \tag{5a}$$

$$\int_{\Omega_{e+1}} \phi^k \; g^L_{e+1} dx = -\; \chi \; [\phi^k(\mathcal{R}(U^h_e, U^h_{e+1}) - U^h_{e+1}(0))]_{x_{e+1}} + \int_{\partial\Omega_{e+1}} (\phi^k \; U^h_{e+1}) \cdot n^- ds - \int_{\Omega_{e+1}} U^h_{e+1} \; \nabla\phi^k dx \quad \forall k \in \{0, 1, ..., p\} \tag{5b}$$

The $2K$ constraints of the interface's surrounding semi-connected gradients (5) must be organized in matrix-vector form to calculate the coefficient vectors $\hat{\boldsymbol{g}}^{\boldsymbol{R}}_{\boldsymbol{e}}$ and $\hat{\boldsymbol{g}}^{\boldsymbol{L}}_{\boldsymbol{e+1}}$ given $\hat{\mathbf{U}}_e$ and $\hat{\mathbf{U}}_{e+1}$. Then, the Recovery operator is applied to yield $\tilde{\sigma}$: specifically, $\tilde{\sigma} = \mathcal{R}(g^R_A, g^L_B)$. Both g^R_A and g^L_B are influenced only by $\hat{\mathbf{U}}_A$ and $\hat{\mathbf{U}}_B$, so the interface gradient $\tilde{\sigma}$ can be calculated directly from $\hat{\mathbf{U}}_A$ and $\hat{\mathbf{U}}_B$. Specifically, $\tilde{\sigma} = \mathcal{L}^\chi(U^h_A, U^h_B) = \mathcal{R}(g^R_A, g^L_B)$. As indicated in Table 1, the CGR-I method applies the LGR operator to calculate interface gradients $\tilde{\sigma}$, facilitating a compact stencil.

III. Fourier Analysis

For Fourier analysis of the IGR family, we maintain the assumption of a p-uniform grid with constant mesh spacing Δx. The governing conservation law is the heat equation with diffusivity of unity:

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} \tag{6}$$

Periodic boundary conditions (BC) are also assumed. The initial condition consists of a sine wave of wavenumber ω: $U(x, 0) = U_0(x) = \sin(\omega x)$. With periodic BC, the exact solution corresponding to this initial condition is $U(x, t) = \exp(-\omega^2 t)U_0(x)$. We define a new parameter $\beta = \omega\Delta x$; this value represents the wavenumber per element. High β corresponds to large solution variation within each element, while small β corresponds to small solution variation. This parameter will be useful in simplifying the update scheme. Based on the initial condition, the DG solution is assumed to vary (7a) according to the shift operator, $\exp(i\omega x)$, and the DG update scheme can be rewritten (7b) in terms of a $K \times K$ amplification matrix $\mathcal{A}$, which is itself a function of β:

$$\hat{\mathbf{U}}_{\mathbf{e+J}} = \exp(i\beta J) \cdot \hat{\mathbf{U}}_e \tag{7a}$$

$$\frac{\partial}{\partial t}\hat{\mathbf{U}}_e = \frac{1}{\Delta x^2} \cdot \mathcal{A}(\beta)\hat{\mathbf{U}}_e \tag{7b}$$

The procedure is explained in more detail by Huynh[14] for the FR method, but the approach is easily applicable to the DG method as well.

In the case of the heat equation, $\mathcal{A}$ should have an eigenvalue (one among its spectrum of K eigenvalues) that approximates $\lambda^{ex} = -\beta^2$ as β approaches zero. This numerical eigenvalue is labeled the consistent

eigenvalue, λ^c. The accuracy of the numerical scheme is described by the order of accuracy with which λ^c approximates λ^{ex}. The difference $|\lambda^{ex} - \lambda^c|$ converges to zero as β approaches zero, and this convergence rate, calculated using Huynh's approach,[14] is labelled as the scheme's order of accuracy.

The eigenvalues of $\mathcal{A}$ govern the maximum allowable timestep when explicit time integration is employed. The spectral radius, $R_s = max|Re(\lambda)|$, is inversely proportional to the maximum allowable timestep. In particular, for the 1D case using the forward Euler method, the stability constraint is $\Delta t \leq \frac{2(\Delta x)^2}{R_s}$. Figure 2 shows two example eigenvalue plots; these particular cases are the GR-II and CGR-I($\chi = 1$) schemes with $p = 2$. The real components of each of the $K = 3$ eigenvalues are plotted versus β. Each eigenvalue is a function of β, but our approach is unable to determine the analytical forms of the eigenvalues. Consequently, on both plots, all three eigenvalues are plotted in the same color for each discrete value of β. The maximum magnitude of any of the eigenvalues of $\mathcal{A}$ is approximately $R_s = 26.5$ for both methods. Fourier analysis has been performed for all members of the IGR family, and the spectral radii and orders of accuracy are shown in Tables 3 and 4 for $p \in \{1, 2, 3, 4, 5\}$. The properties of the LDG[11] and BR2[15] schemes are also shown for comparison. Our implementation of BR2 follows the description of Arnold et al.,[10] taking the stabilization parameter χ to be 2.

Table 3: Order of accuracy from Fourier analysis. **X** indicates an unstable scheme.

Scheme	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
GR-I	2	6	6	10	10
GR-II	4	8	10	14	16
GR-III	2	6	6	10	10
GR-IV	3	5	7	9	11
GR-V	0	2	2	4	**X**
GR-VI	4	6	8	10	12
GR-VII	3	5	7	9	11
LDG-OS	4	6	8	10	12
CGR-I($\chi = 1$)	4	4	8	8	12
CGR-I($\chi = 2$)	4	4	8	8	12
BR2($\chi = 2$)	2	4	6	8	10

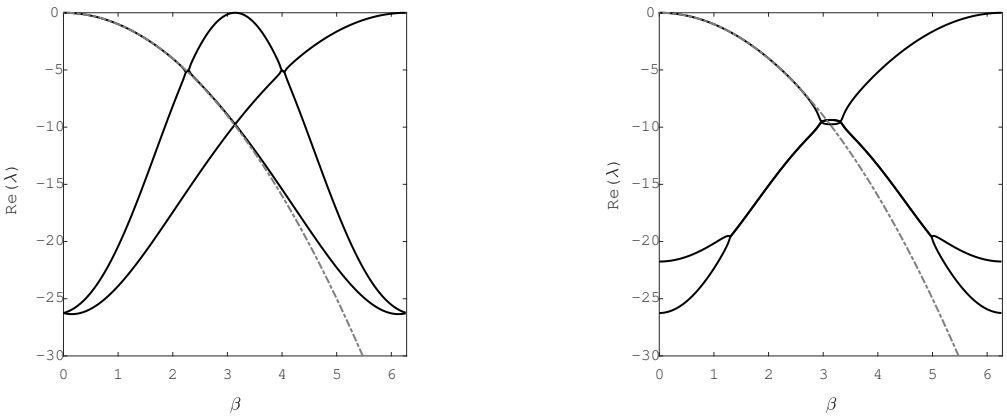
Out of all the schemes analyzed, the $p1$ case of GR-V is the only inconsistent scheme; the GR-V approach also produces an unstable scheme in the $p5$ case. The GR-IV and GR-VII methods show order $2p+1$ accuracy; for both schemes, the consistent eigenvalue has a nonzero imaginary component that converges to zero at rate $2p + 1$, so while the real component of λ^c is order $2p + 2$ accurate, the imaginary component reduces the schemes to order $2p + 1$ accuracy. The CGR-I scheme exhibits unusual behavior; the order of accuracy is $2p$ for even p and $2p + 2$ for odd p, regardless of the value chosen for the parameter χ. The spectral radius of this scheme increases with χ. This behavior is unsurprising, as it has also been mentioned for the BR2

Table 4: $|Re(\lambda)|_{max}$, rounded up to the nearest integer. **X** indicates an unstable scheme.

Scheme	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
GR-I	13	60	171	381	739
GR-II	9	27	54	90	135
GR-III	7	60	171	381	739
GR-IV	7	60	171	381	739
GR-V	19	60	171	381	**X**
GR-VI	7	98	336	859	1832
GR-VII	24	114	377	940	1974
LDG-OS	36	149	439	1046	2143
CGR-I($\chi = 1$)	9	27	50	86	132
CGR-I($\chi = 2$)	24	77	180	347	585
BR2($\chi = 2$)	36	147	420	976	1965

scheme in the work of Peraire & Persson.[16]

From the Fourier analysis, it appears that GR-II is the best scheme in terms of both spectral radius and order of accuracy. However, the scheme is not robust on non-Cartesian elements in 2D, and we suspect that the cause could be the scheme's excessive dependence on non-neighbor elements when forming the update equation for each element. The GR-IV, GR-V, GR-VI, and GR-VII schemes are attempts to lower the scheme's reliance on non-neighbor elements. For example, the GR-VI scheme makes use of biased recovery on interfaces, using the relationships of C_A and C_B in the BGR operator to regulate the flow of information.



(a) (b)

Figure 2: Eigenvalues of the amplification matrix $\mathcal{A}$ for the a) GR-II and b) CGR-I($\chi = 1$) schemes in the $p = 2$ case. The dotted gray line is the exact eigenvalue for heat equation, $\lambda^{ex} = -\beta^2$.

IV. Numerical Test: 1D Heat Equation

The family of IGR schemes and the popular methods analyzed in Table 3 are now compared by simulating the time-dependent heat equation with unity diffusivity coefficient and periodic boudary conditions, as described in the previous section. For this particular test, $\omega = 1$ and $\mathbf{\Omega} = [0, 2\pi]$. A uniform grid is used in all cases, such that $\Delta x = \frac{2\pi}{M}$, where M is the number of elements. Each method is applied to simulate the system from $t = 0$ to $t = 2$ using the explicit 8^{th} order Runge-Kutta scheme of Prince and Dormand.[17] Two error metrics are used for the investigation: the L_2 error in cell averages, denoted E_{CA}, and the global L_2 error norm, denoted E_{GLO}:

$$E_{CA} = \sqrt{\frac{1}{M}\sum_{e=0}^{M-1}(\overline{U}_e^h - \overline{U}_e)^2} \qquad E_{GLO} = \sqrt{\sum_{e=0}^{M-1}\int_{\Omega_e}(U_e^h - U)^2 dx} \tag{8}$$

The cell averages of the DG and exact solutions are denoted $\overline{U_e^h}$ and $\overline{U_e}$, respectively, in each element Ω_e. All error measurements are obtained by comparing the DG solution at the final time, $U^h(x, 2)$, to the exact solution at the final time, $U(x, 2) = \exp(-2)\sin(x)$. Every scheme is run on a set of successively refined meshes for $p \in \{1, 2, 3\}$, and the error's order of convergence is measured in both norms.

Through the numerical testing process, it has become clear that some of the methods are sub-optimal in the global error norm, meaning that the order of convergence in E_{GLO} is less than $p + 1$ for at least one choice of $p \in \{1, 2, 3\}$. These schemes are GR-I, GR-III, and GR-V; in fact, the sub-optimal behavior of the simpler GR-I and GR-III schemes was the inspiration for the design of the GR-II scheme. All of the remaining schemes from Table 3 are optimal in the global error norm, meaning that order $p+1$ convergence is achieved. No more reporting with respect to this error norm is necessary.

The convergence behavior in E_{CA} is now explored. Tables 5 lists the number of elements M, the cell-average error E_{CA}, and the order of convergence in E_{CA} for the grid refinement study of the GR-II scheme; the order of convergence is measured between each successive pair of meshes for each discretization order p. From the tabular data, it is evident that GR-II converges at rates four, eight, and ten for $p = 1$, $p = 2$, and $p = 3$, respectively. The convergence rates of all of the optimal schemes have been determined in the same manner and are listed in Table 6. For many schemes (GR-II included) the convergence rates in E_{CA} match the orders of accuracy given by Fourier analysis; in other cases, the two values are mismatched. For example, in the case of the GR-VII scheme, E_{CA} converges with order $2p+1$ for $p \in \{1, 2\}$, but order $2p+2$ for $p = 3$; alternatively, LDG-OS is order $2p + 2$ accurate from Fourier analysis, but sometimes only converges at rate $2p + 1$ in E_{CA}.

Table 5: GR-II Performance, 1D heat equation.

p	**M**	$\mathbf{E_{CA}}$	**Order**	**p**	**M**	$\mathbf{E_{CA}}$	**Order**	**p**	**M**	$\mathbf{E_{CA}}$	**Order**
1	10	$4.58E-05$	–	2	10	$3.05E-09$	–	3	3	$6.46E-08$	–
1	20	$2.87E-06$	3.99	2	20	$1.16E-11$	8.04	3	4	$3.38E-09$	10.26
1	30	$5.68E-07$	4.00	2	30	$4.50E-13$	8.01	3	6	$5.92E-11$	9.97
1	40	$1.80E-07$	4.00	2	40	$5.00E-14$	7.64	3	8	$3.35E-12$	9.98
1	50	$7.37E-08$	4.00	2	50	$1.00E-14$	7.21	3	10	$3.60E-13$	10.00

Table 6: Convergence rates in E_{CA} for 1D heat equation test.

Method	**p**	**Order**	**Method**	**p**	**Order**	**Method**	**p**	**Order**
GR-II	1	4	GR-II	2	8	GR-II	3	10
GR-IV	1	3	GR-IV	2	5	GR-IV	3	8
GR-VI	1	4	GR-VI	2	6	GR-VI	3	8
GR-VII	1	3	GR-VII	2	5	GR-VII	3	8
LDG-OS	1	3	LDG-OS	2	6	LDG-OS	3	7
CGR-I($\chi = 1$)	1	4	CGR-I($\chi = 1$)	2	4	CGR-I($\chi = 1$)	3	8
CGR-I($\chi = 2$)	1	4	CGR-I($\chi = 2$)	2	4	CGR-I($\chi = 2$)	3	8
BR2($\chi = 2$)	1	2	BR2($\chi = 2$)	2	4	BR2($\chi = 2$)	3	6

V. Extension to Two Dimensions

The method is now described in the context of 2D problems with periodic boundary conditions. Take the reference element Ω_{ref} to be the unit square with reference coordinates $(\xi, \eta) \in [0, 1]^2$; we restrict ourselves to quadrilateral elements. Take the solution basis $\boldsymbol{\phi}$ to be a tensor product basis of degree p in each direction, with a total of $K = (p + 1)^2$ members.

In the 2D case, the integration of the volume integrals in the DG weak form (2b) and the auxiliary weak form (3b) is performed with $GQ_v = K$ quadrature points over the interior of each element, with the quadrature nodes $\boldsymbol{\xi}_g = (\xi_g, \eta_g), \quad g \in \{1, 2, ..., GQ_v\}$ being laid out in tensor product fashion on the unit square. The quadrature nodes and weights are taken to be the usual Gauss-Legendre nodes and weights, transformed from the bi-unit to the unit domain.

Along interfaces, all integrals in the DG weak form (2b), the auxiliary weak form (3b), and the weak form for the semi-connected gradients of an interface (9) are evaluated using a $GQ_s = p+1$ point quadrature rule, again using the 1D Gauss-Legendre nodes and weights. The same quadrature resolution is applied for mass matrix formation, populating the DG residual, and all integrals arising from solving the mixed form for the auxiliary variable and the semi-connected gradients. Note that in 2D, the auxiliary variable and semi-connected gradients have two directional components per element in order to account for derivatives in both the x and y directions.

The semi-connected gradient solve (9) in the multidimensional case merits additional explanation. In the 2D quadrilateral case, since each element has four edges, there are four semi-connected gradient approxima-

tions associated with each element, but the scheme can be implemented in such a manner that the coefficients for the semi-connected gradients are not explicitly stored. Setting $\mathcal{I}_{AB}$ to be the interface connecting two elements Ω_A and Ω_B, the semi-connected gradient of Ω_A associated with $\mathcal{I}_{AB}$ uses information from only Ω_A and Ω_B:

$$\begin{aligned}\int_{\Omega_A} \phi^k \; g_A^{\mathcal{I}} d\mathbf{x} = \quad & \chi \int_{\mathcal{I}_{AB}} \phi^k (\tilde{U} - U_A^h) \cdot n^- ds + \\ & \int_{\partial\Omega_A} \phi^k (U_A^h \cdot n^-) ds - \int_{\Omega_A} U_A^h \; \nabla\phi^k d\mathbf{x} \quad \forall k \in \{0, 1, ..., K-1\}\end{aligned} \tag{9}$$

The semi-connected gradient in Ω_B is formed similarly. As with the 1D case, $\tilde{\sigma}_{\mathcal{I}_{AB}} = \mathcal{R}(g_A^{\mathcal{I}}, g_B^{\mathcal{I}})$. The recovery basis $\boldsymbol{\psi}$ in 2D is constructed as a tensor product basis; let $\mathcal{U} = \Omega_A \cup \Omega_B$ be the union of two elements, over which a recovery coordinate $\mathbf{r} = (r, s)$ is defined. Following Lo's implementation,[18] the origin of the recovery coordinate is the midpoint of the interface linking the two elements; r is the face-normal coordinate and s is orthogonal to r. For straight-edged interfaces, s is normalized so that it reaches a maximum magnitude of unity on the interface, and r is normalized to reach a maximum magnitude of unity at any of the vertices of $\mathcal{U}$. Then, $\boldsymbol{\psi}(\mathbf{r})$ is built as a tensor-product basis of order $2p-1$ in the r direction and order p in the s direction. In practice, we use a Gram-Schmidt orthonormalized[19] basis for $\boldsymbol{\phi}$ on each element and a tensor product basis of Legendre polynomials for $\boldsymbol{\psi}$. The details of the biased recovery setup in the 2D case can be found in Khieu et al.,[20] where it is applied to form the accurate and stable icb$p$[0] DG scheme for advection phenomena. When applying the GBR operator $\mathcal{B}_{C_A}^{C_B}$ for the GR-VI scheme, we restrict ourselves to the following two configurations: either $(C_A, C_B) = (K, p+1)$ or $(C_A, C_B) = (p+1, K)$.

For advection-diffusion scenarios, the governing equation includes an advective flux $\mathcal{F}$ which depends only on the field variables U and not the solution gradient:

$$\frac{\partial}{\partial t} U = \nabla \cdot \mathcal{G}(U, \nabla U) - \nabla \cdot \mathcal{F}(U) \tag{10}$$

The advective flux is discretized with the standard DG procedure, well-described by Cockburn & Shu.[1] The only detail that must be filled in here is the approximate Riemann solver used to populate the flux along element interfaces: we make use of the Rusanov flux.[21]

VI. Numerical Tests: 2D Navier-Stokes Equations

The performance of some of the IGR schemes (GR-II, GR-VI, and CGR-I) is now demonstrated with three 2D test problems under periodic boundary conditions. As with the 1D case, the numerical solution is initialized to the finite-dimensional projection of an exact initial condition $U(\mathbf{x}, 0)$. Then, the 5^{th} order explicit Runge-Kutta scheme of Cash & Karp[22] is paired with the DG discretization to integrate the solution forward in time. The timestep size Δt is chosen to be as large as possible while maintaining numerical stability, so it is inversely proportional to a given scheme's spectral radius. For all three tests, only the $p3$ configurations of the IGR schemes are applied; the established LDG and BR2 methods are also implemented for comparison. For the LDG method, we take the one-sided flux approach recommended by both Guo et al.[23] and Landmann et al.[24] The CGR-I and BR2 methods are both applied with $\chi = 2$. At the end of each simulation, the error is measured in two norms. The first norm is the global L_2 norm, denoted E_G and populated by integrating the square of the error over the entire domain. Second, the error in a target functional is measured, following the approach of Hartmann & Houston.[25] Let $J(Y) = \int_{\mathbf{\Omega}} \kappa(\mathbf{x}) \; Y(\mathbf{x}) d\mathbf{x}$, where κ is some user-specified kernel function and Y is some function populated over $\mathbf{\Omega}$. The error in the target functional, denoted E_J, is the magnitude of the difference between the target functionals of the exact and DG solutions.

$$E_G = \sqrt{\sum_{m=0}^{M-1} \int_{\Omega_m} (U_m^h - U)^2 d\mathbf{x}}, \qquad E_J = \left| \sum_{m=0}^{M-1} \int_{\Omega_m} \kappa \; U d\mathbf{x} \; - \; \sum_{m=0}^{M-1} \int_{\Omega_m} \kappa \; U_m^h d\mathbf{x} \right| \tag{11}$$

The first two tests involve the non-advective Navier-Stokes (NANS) equations. This set (12a) is the compressible Navier-Stokes (CNS) equations (12b) in 2D with advection terms ($\mathcal{F}$) removed:

$$\text{NANS}: \qquad \frac{\partial U}{\partial t} = \frac{\partial \mathcal{G}^x}{\partial x} + \frac{\partial \mathcal{G}^y}{\partial y} \tag{12a}$$

$$\text{CNS}: \qquad \frac{\partial U}{\partial t} = \frac{\partial (\mathcal{G}^x - \mathcal{F}^x)}{\partial x} + \frac{\partial (\mathcal{G}^y - \mathcal{F}^y)}{\partial y} \tag{12b}$$

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E_g \end{bmatrix} \qquad E_g = \frac{P}{\gamma - 1} + \frac{\rho}{2}(u^2 + v^2) \tag{12c}$$

$$\mathcal{F}^x = \begin{bmatrix} \rho u \\ \rho u^2 + P \\ \rho u v \\ u(E_g + p) \end{bmatrix} \qquad \mathcal{G}^x = \begin{bmatrix} 0 \\ \frac{2\mu}{3}(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}) \\ \mu(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) \\ u(\frac{2\mu}{3}(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y})) + v(\mu(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})) + K_d \mu(\frac{\partial}{\partial x}(T)) \end{bmatrix} \tag{12d}$$

$$\mathcal{F}^y = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + P \\ v(E_g + p) \end{bmatrix} \qquad \mathcal{G}^y = \begin{bmatrix} 0 \\ \mu(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) \\ \frac{2\mu}{3}(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}) \\ v(\frac{2\mu}{3}(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}) + u(\mu(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})) + K_d \mu(\frac{\partial}{\partial y}(T)) \end{bmatrix} \tag{12e}$$

where ρ, ρu, ρv, and E_g represent the fluid density, momentum in the x direction, momentum in the y direction, and energy, respectively. The fluid pressure, viscosity, and thermal diffusivity coefficient are denoted P, μ, and K_d, respectively. The thermal diffusivity coefficient is dependent on the Prandtl number: $K_d = \frac{\gamma}{Pr \cdot (\gamma - 1)}$, with $Pr = \frac{4\gamma}{9\gamma - 5}$. The temperature ($T$) appearing in the energy flux is calculated using the ideal gas law, $p = \rho R_g T$. In this work, we always take the gas constant R_g to be unity, therefore $P = \rho\, T$.

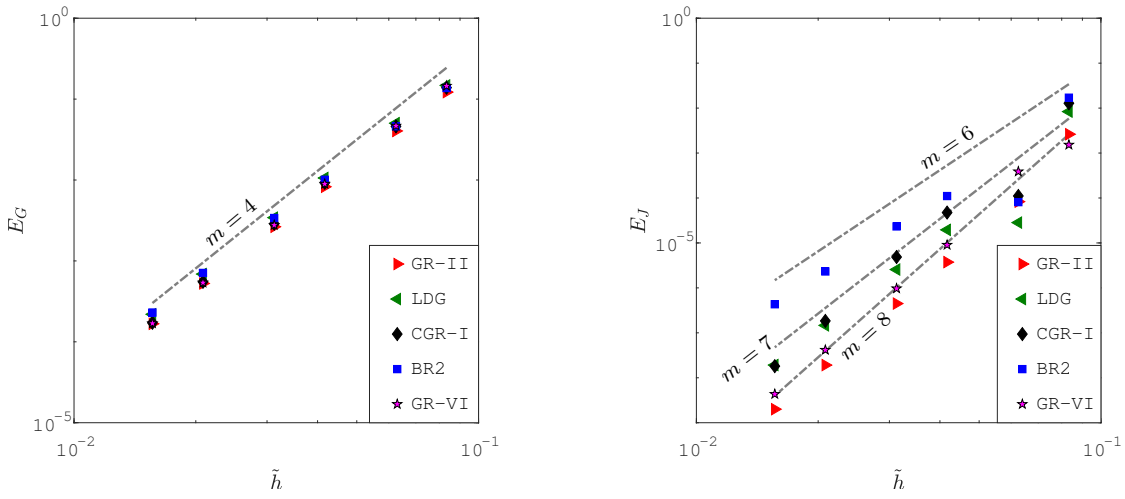
For the first test, the DG method is applied to solve the NANS equations on a square domain with uniform square elements. The domain is $\mathbf{\Omega} = [0, \pi]^2$, partitioned into M square elements, such that each element is a square with side length $h = \frac{\pi}{\sqrt{M}}$. We force an unsteady exact solution:

$$U(\mathbf{x}, t) = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E_g \end{bmatrix} = (\exp(-t) + 1 - \exp(-1)) \begin{bmatrix} 4 + \sin(2x + 2y) \\ \frac{1}{5}\sin(2x + 2y) \\ \frac{1}{5}\sin(2x + 2y) \\ (4 + \sin(2x + 2y))^2 \end{bmatrix} \tag{13}$$

using the method of manufactured solutions, which requires a time-dependent source term to be added to the governing equations (12a). The fluid parameters are as follows: the ratio of specific heats is $\gamma = 1.4$ and the viscosity is constant at $\mu = 1$. Even with constant viscosity, the flux law remains nonlinear because the velocity is present in the energy flux component. This test problem is a modified version of the steady-state problem employed by Hartmann & Houston[25] to test a version of the interior penalty DG method. Once initialized, the DG solution is integrated forward in time to $t = 1$. We imitate Hartmann & Houston[25] by taking $J = \sin(\pi x)\sin(\pi y)$ to measure the functional error.

The global and functional errors are measured in the energy field, E_g. The exact target functional to 13 digits is $J(E_g) = 3.227287522641$. The convergence behavior for test 1 is shown in Figure 3; the error is

plotted against the characteristic mesh width, $\tilde{h} = nDOF^{-0.5}$, where $nDOF$ is the total count of degrees of freedom per field variable in a given simulation.



(a) Global error

(b) Functional error

Figure 3: Results from test 1 (uniform Cartesian elements). Gray lines illustrate the convergence rates, denoted by m, with respect to the characteristic mesh width $\tilde{h}$. (a): Convergence in the global error norm. All schemes show 4^{th} order convergence. (b): Convergence in the functional error norm. The BR2, LDG, and IGR schemes show 6^{th}, 7^{th}, and 8^{th} order convergence, respectively.

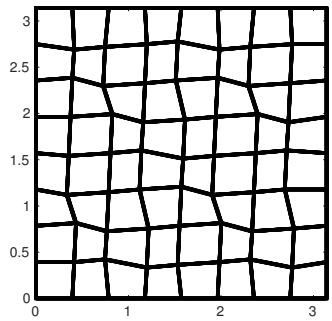
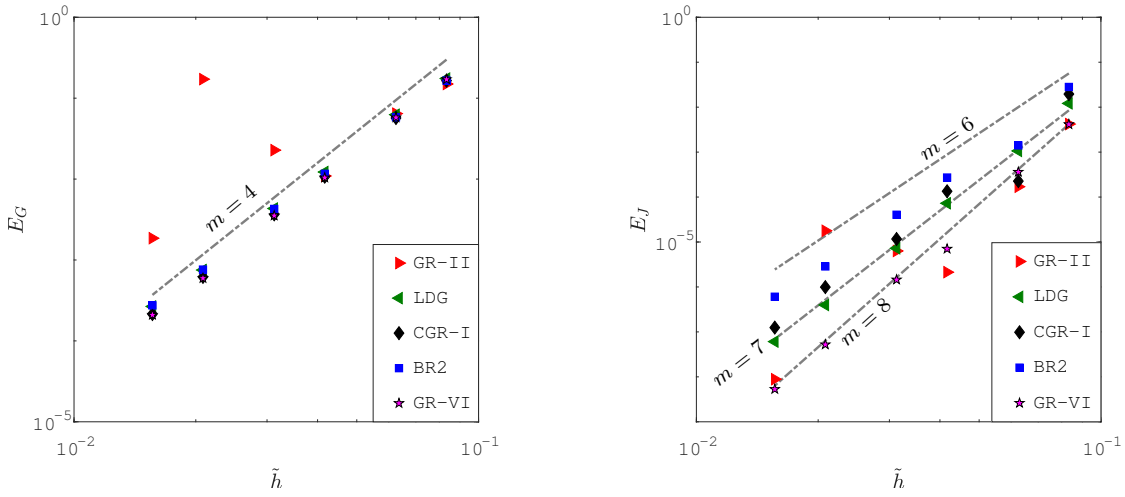


Figure 4: Sample $M = 8 \times 8$ mesh for test 2

In the global error norm, all five schemes converge at the optimal rate, specifically $p + 1 = 4$, giving similar error on all meshes. The GR-II scheme is by far the fastest computation because its relatively small spectral radius allows for larger timesteps than the other methods. Per timestep, the compact methods (CGR-I and BR2) are faster, but the GR-II scheme's spectral radius advantage (requiring fewer timesteps to reach $t = 1$) makes it much faster overall. Specifically, on the finest mesh used for this test ($M = 256$ elements), CGR-I took 1.93 times as long as the GR-II calculation, and BR2($\chi = 2$) took 4.59 times as long. Typically, BR2 is applied with a larger χ value than 2, and the scheme's spectral radius increases[16] with χ. While the errors of all schemes are relatively close, the IGR methods appear marginally more accurate in the global error norm than LDG and BR2 on the finer meshes.

In the functional error norm, different convergence rates are observed for the different schemes. The three members of the IGR family all show 8^{th} order convergence in this norm. LDG-OS is reduced to 7^{th}



(a) Global error

(b) Functional error

Figure 5: Results from test 2 (nonuniform quadrilateral elements). Gray lines illustrate the convergence rates, denoted by m, with respect to the characteristic mesh width $\tilde{h}$. (a): Convergence in the global error norm. Most schemes show 4^{th} order convergence. (b): Convergence in the functional error norm.

order convergence, and BR2 achieves only 6^{th} order convergence. With the exception of the LDG-OS and GR-II approaches, the orders of convergence in this norm match the order of accuracy results from the 1D Fourier analysis. We remind the reader that the smaller spectral radius of the GR-II scheme allows for a faster computation, and it is the best scheme in terms of both accuracy and computational time. Initially, the authors suspected that the reduced convergence rate of the GR-II scheme (8^{th} order here, as opposed to the 10^{th} order convergence observed in the 1D heat equation test) was due to difficulties in handling the nonlinearity of the flux function. However, extensive testing in 2D with linear flux functions has shown that when a shear stress term is present (as in the Navier-Stokes viscous flux), the scheme's convergence is reduced to order $2p + 2$. Consequently, the scheme loses some of its appeal in multidimensional implementations, but it still gives superior performance compared to other DG methods. Between the two compact schemes shown (CGR-I and BR2), CGR-I is superior in terms of both accuracy and computational time, owing to a higher convergence rate and smaller spectral radius than BR2.

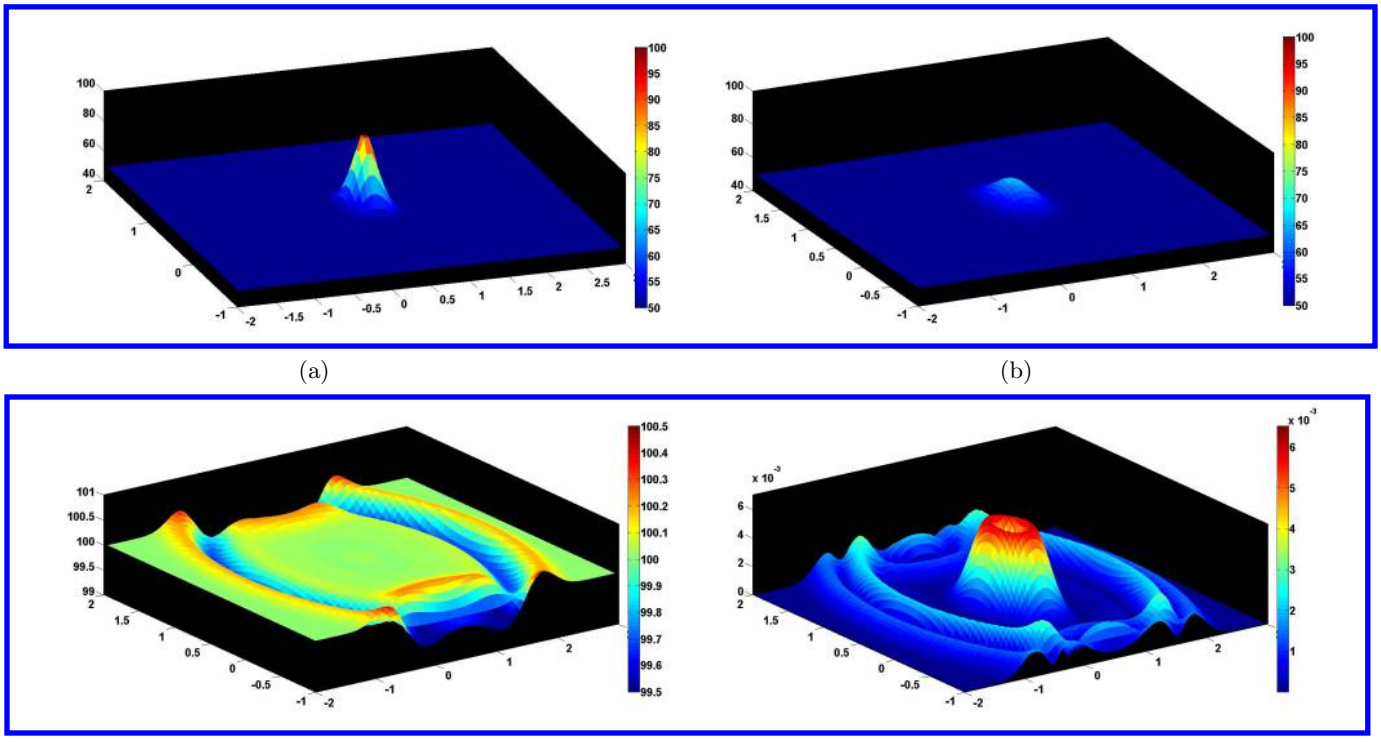
For the second test in the 2D environment, we simulate the same solution as in test 1, but instead of using uniform Cartesian meshes, the elements are now perturbed quadrilaterals. For a given element count M, beginning with the uniform Cartesian mesh, each element vertex is perturbed by up to 15 percent of the average element width in both the x and y directions, for a maximum perturbation distance of approximately 21 percent of the average element width. Additionally, the quadrature resolution is increased to $GQ_v = (p + 2)^2$ in order to improve the calculation of the volume integrals, since the Jacobian matrix involved in the quadrature process is non-constant over each element. A sample perturbed mesh, for the $M = 8 \times 8$ case, is shown in Figure 4.

The results of test 2 are shown in Figure 5. Again, the convergence in both the global and functional error norms with respect to the characteristic mesh width $\tilde{h}$ is presented. The BR2 method is again implemented with $\chi = 2$; it was also tested with $\chi = 4$, following the conventional approach of taking χ to be the number of sides per element, but for this test, the method performs better with $\chi = 2$. The GR-II scheme, which was the best in test 1, shows erratic behavior in test 2; specifically, the solution actually becomes less accurate as the mesh is refined over the mid-resolution cases. This behavior is undesirable. With the exception of the GR-II scheme, all schemes remain optimal in the global error norm. With respect to the functional error

norm, the CGR-I method is only slightly better than BR2 in terms of accuracy, but remains computationally less expensive than the LDG and BR2 methods because of its smaller spectral radius. The GR-VI scheme maintains 8^{th} order convergence in the functional error across all meshes, making it the superior scheme in this test.

For the final test, the DG method is applied to solve the 2D compressible Navier-Stokes equations (12b) with periodic boundary conditions. The standard DG approach is used to handle the advective fluxes, denoted $\mathcal{F}$, and the IGR family, alongside the LDG and BR2 approaches, is applied to discretize the diffusive fluxes, denoted $\mathcal{G}$. The initial condition (14) is a mostly quiescent, uniform field, with a severe density perturbation prescribed around the epicenter $(x, y) = (0.5, 0.5)$. The fluid has ratio of specific heats $\gamma = 1.4$, freestream density $\rho_\infty = 2$, freestream pressure $P_\infty = 100$, and viscosity $\mu = 0.1$.

The domain is a rectangle, $\mathbf{\Omega} = [-2, 3] \times [-1, 2]$. The mesh is a set of $M = 5R \times 3R$ elements, where R is some specified resolution and all elements in each mesh are uniform squares, each with side length of $h = \frac{1}{R}$. Once the DG solution has been initialized to the prescribed initial condition, it is integrated forward in time to $t = 0.1$.



(a) (b)

(c) (d)

Figure 6: The flow behavior from test 4; the result shown is from the reference simulation, but only cell averages are plotted. (a): Initial temperature distribution; the temperature peaks at $T = 100$. (b): Temperature distribution at $t = 0.1$. (c): Pressure distribution at $t = 0.1$. (d): Mach number distribution at $t = 0.1$.

The behavior of the flow is illustrated in Figure 6; specifically, we show the temperature distribution at initialization (Figure 6a) in addition to the temperature (Figure 6b), pressure (Figure 6c), and Mach number (Figure 6d) distributions at $t = 0.1$. The initial density perturbation yields a temperature spike with maximum temperature $T = 100$. The thermal diffusion process results in an acoustic wave that radiates outward from the temperature spike, which is rapidly diffused. The peak in the Mach number distribution is focused around the temperature spike; the flow in this neighborhood is converging towards $(x, y) = (0.5, 0.5)$.

$$U(\mathbf{x},0) = \begin{bmatrix} \rho \\ \\ \rho u \\ \rho v \\ E_g \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \rho_\infty & \text{for} \quad x \notin [0,1]^2 \\ \rho_\infty(1-\frac{1}{2}\sin^3(\pi x)\sin^3(\pi y)) & \text{for} \quad x \in [0,1]^2 \end{bmatrix} \\ \\ 0 \\ 0 \\ \frac{P_\infty}{\gamma-1} \end{bmatrix} \tag{14}$$

The error at the end of each simulation is measured exclusively in the functional error norm. In the absence of an analytical solution to the prescribed problem (14), the reference value for the functional is set from a reference solution, specifically a $p4$ simulation using the CGR-I method with $M = 3840$ elements ($R = 16$). Taking the kerkel κ to be an abbreviated sum of sine waves:

$$\kappa(x,y) = \sum_{j=2}^{4} \left(\sin(\frac{j}{4}\pi x) - \sin(\frac{j}{4}\pi y) \right), \tag{15}$$

the reference functional for the energy, to ten digits, is $\int_{\mathbf{\Omega}} \kappa \; E_g d\mathbf{x} = 150.9473163$. The error of a given simulation is taken to be the difference between that simulation's target functional and the reference target functional, $E_J = |J - J^{ref}| = |J - 150.9473163|$.

The convergence of all schemes with respect to the characteristic mesh width $\tilde{h}$ is shown in Figure 7. The standard DG method for the advective fluxes has order of accuracy $2p + 1$; the highest asymptotic convergence rate of any advection-diffusion DG scheme presented in this work is consequently limited to order $2p + 1$. Figure 7 shows that on the coarser half of the meshes, the five methods presented do not vary appreciably in accuracy. On the finer half of meshes, the GR-VI and CGR-I schemes achieve better agreement with the $2p + 1$ order of convergence that would be expected for a purely advective problem. On all meshes, the CGR-I and GR-II methods require less computational time than the other methods because the spectral radius of the diffusion scheme influences the maximum safe timestep. Specifically, on the finest mesh, the CGR-I and BR2 simulations took 1.13 and 1.91 times as long as the GR-II simulation, respectively.

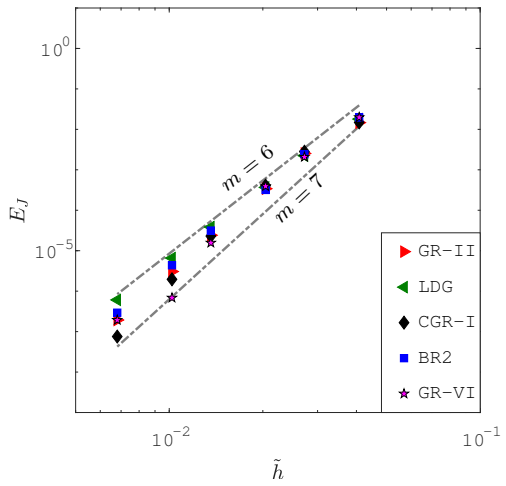


Figure 7: Functional error from test 4 (uniform Cartesian elements). Gray lines illustrate approximate convergence rates, denoted by m, with respect to the characteristic mesh width $\tilde{h}$. On the coarser meshes, all methods work equally well. On the finer meshes, the CGR-I and GR-VI methods are more accurate.

VII. Extension to the Flux Reconstruction Method

We detour from the DG framework and move to the more general Flux Reconstruction (FR) method of Huynh.[13] The FR method solves the differential form of a governing differential equation (as opposed to the weak form solved by the DG method). The method was extended from hyperbolic PDE problems to parabolic/elliptic problems by Huynh[14] in 2009. In this section, we show how the CGR-I method can be paired with the FR framework for solving the 1D heat equation. The FR method is well-described in the work of Huynh,[14] and we give only a summary here.

To begin, the domain $\mathbf{\Omega} = [x_0, x_M]$ is partitioned into M finite elements, with $\Omega_e \in [x_e, x_{e+1}]$ for $e \in \{0, 1, ..., M-1\}$. Let the numerical solution $U^h(x)$ be a piecewise continuous polynomial described by $p+1 = K$ degrees of freedom in each element Ω_e. Specifically, taking $\boldsymbol{\phi}$ to be a Lagrange basis of degree p (defined on the reference element Ω_{ref}) and $\hat{\mathbf{U}}_\mathbf{e}$ to be the nodal values of U^h at the K solution points within each Ω_e, we form a polynomial expansion:

$$U^h(x \in \Omega_e) = U_e^h(\xi(x)) = \sum_{k=0}^{p} \phi^k(\xi)\ \hat{U}_e^k, \tag{16}$$

where x is the physical coordinate and $\xi \in [0,1]$ is the reference coordinate, defined by $\xi(x) = \frac{x - x_e}{x_{e+1} - x_e}$ over each element. We take the solution points to be the K Gauss-Legendre points on the reference element. Given an arbitrary diffusive flux law

$$\frac{\partial U}{\partial t} = \frac{\partial \mathcal{G}}{\partial x}, \tag{17}$$

with $\mathcal{G}$ depending on U and ∇U, we seek to approximate the divergence of the flux at each solution point $x_{e,k}^S$ within each element ($x_{e,k}^S$ is the k^{th} solution point in element Ω_e). For the simple case of the heat equation with unity diffusivity, $\mathcal{G} = \nabla U$. Since $\boldsymbol{\phi}$ is a nodal basis, the temporal derivative of each DOF $\hat{U}_e^k$ is the divergence of the flux at the corresponding solution point $x_{e,k}^S$.

The method requires some mechanism to allow elements to exchange information. Towards this end, on the reference element, we define two degree $p+1$ correction functions, one corresponding to each endpoint of Ω_{ref}. Specifically, $C^L(\xi)$ is equal to unity at $\xi = 0$ and equal to 0 at $\xi = 1$. Similarly, $C^R(\xi)$ is equal to unity at $\xi = 1$ and equal to zero at $\xi = 0$. In the work of Huynh,[14] these functions, known as the left and right correction functions, are denoted g, but we have denoted them C in this work to avoid confusion with the semi-connected gradient approximation of the CGR-I method. The correction functions have a symmetry relation: $C^R(\xi) = C^L(1-\xi)$, so one is known once the other has been defined. Within each element, a connected solution (18a), denoted U^C, is formed by combining the discontinuous solution U^h with a correction based on interface jump terms:

$$U^C(x \in \Omega_e) = U_e^C(\xi(x)) = U_e^h(\xi) + (\tilde{U}_L - U_e^h(0))\ C^L(\xi) + (\tilde{U}_R - U_e^h(1))\ C^R(\xi) \tag{18a}$$

$$\frac{\partial U_e^C}{\partial x}(\xi) = \sum_{k=0}^{p} \hat{U}_e^k\ \frac{\partial \phi^k}{\partial x}(\xi) + (\tilde{U}_L - U_e^h(0))\ \frac{\partial C^L}{\partial x}(\xi) + (\tilde{U}_R - U_e^h(1))\ \frac{\partial C^R}{\partial x}(\xi) \tag{18b}$$

The calculated values $\tilde{U}_L$ and $\tilde{U}_R$ are the common values on the left and right interfaces of Ω_e, respectively. For the CGR-I method, given an interface $\mathcal{I}_{AB}$ shared by Ω_A and Ω_B, we take $\tilde{U} = \mathcal{R}(U_A^h, U_B^h)$ on the interface. Thanks to the use of the correction functions, U^C is globally continuous, unlike the numerical solution U^h. For every element Ω_e, the connected solution is differentiated (18b) to populate the solution derivative at the K solution points $\{x_{e,0}^S, x_{e,1}^S, ..., x_{e,p}^S\}$ of the element. Then, the derivative is used to calculate the viscous flux $\mathcal{G}$ at each of the solution points.

The next step is to calculate the flux along element interfaces. Now, focusing on a specific interface $\mathcal{I}_{e,e+1}$ between Ω_e and Ω_{e+1}, take $U_{e+1}^{SC,L}$ be the left-connected solution in Ω_{e+1} and $U_e^{SC,R}$ to be the right-connected solution in Ω_e:

$$U_e^{SC,R}(\xi) = U_e^h(\xi) + \chi\ (\tilde{U} - U_e^h(1))\ C^R(\xi) \tag{19a}$$

$$U_{e+1}^{SC,L}(\xi) = U_{e+1}^h(\xi) + \chi\ (\tilde{U} - U_{e+1}^h(0))\ C^L(\xi) \tag{19b}$$

These solutions add corrections to the numerical solution U^h, but each only includes information from the two elements sharing the interface. Again, the common value $\tilde{U}$ on the interface is taken to be the recovered solution. We emphasize that the right-connected and left-connected solutions of the elements Ω_e and Ω_{e+1} are particular to the shared interface; for the interface on the left side of Ω_e, namely $\mathcal{I}_{e-1,e}$, we would be interested in the left-connected solution of Ω_e and the right-connected solution of Ω_{e-1}.

As with the DG case, let g_e^R and g_{e+1}^L be the semi-connected gradients over the two neighboring elements. These are formed as polynomial expansions (20) using the basis functions $\boldsymbol{\phi}$ over each element:

$$g_e^R(\xi) = \sum_{k=0}^{p} \phi^k(\xi)\ \nabla U_e^{SC,R}(\xi_k) \tag{20a}$$

$$g_{e+1}^L(\xi) = \sum_{k=0}^{p} \phi^k(\xi)\ \nabla U_{e+1}^{SC,L}(\xi_k) \tag{20b}$$

The coefficients in the semi-connected gradient approximation correspond to nodal values of the solution gradient. The gradients of the semi-connected solutions (19) are simple to calculate because the gradients of the correction functions are known analytically, and the gradient of the discontinuous solution U^h can be calculated by differentiating the basis functions $\boldsymbol{\phi}$. Once the semi-connected gradient approximations (20) native to a given interface have been populated, the recovery procedure is performed to calculate the interface gradient: $\tilde{\sigma} = \mathcal{R}(g_e^R, g_{e+1}^L)$. Then, the interface gradient is applied to calculate the viscous flux on the interface: $\tilde{\mathcal{G}} = \mathcal{G}(\tilde{U}, \tilde{\sigma})$.

With the flux $\mathcal{G}$ populated for all interfaces and element solution points, the next step is to differentiate the flux distribution. For a given element Ω_e, take $\tilde{\mathcal{G}}_L$ to be the calculated flux on the left-side interface (using the interface's common gradient), take $\tilde{\mathcal{G}}_R$ to be the calculated flux on the right-side interface, and take $\hat{\mathcal{G}}_e^k$ to be the calculated flux at an element's solution point $x_{e,k}^S$. The nodal flux values $\hat{\boldsymbol{\mathcal{G}}}_{\boldsymbol{e}}$ allow a flux polynomial (21a) to be formed within the element, denoted $\mathcal{G}_e^h(\xi)$. Additionally, two more terms are necessary before we proceed. Take $\mathcal{G}_L^e$ and $\mathcal{G}_R^e$ to be viscous fluxes calculated along the element's left and right edges using the fully connected solution. Specifically, $\mathcal{G}_e^L = \mathcal{G}(U_e^C(0), \frac{\partial U_e^C}{\partial x}(0))$ and $\mathcal{G}_e^R = \mathcal{G}(U_e^C(1), \frac{\partial U_e^C}{\partial x}(1))$. The globally continuous flux polynomial (21b) is formed by combining a pair of correction functions C^L and C^R with the flux differences at the left and right boundaries of the element. With $\mathcal{G}^C(x)$ being a continuous function (21b) approximating the flux across the entire domain, $\nabla \cdot \mathcal{G}^C$ provides the divergence calculation (21c) at each solution point:

$$\mathcal{G}_e^h(\xi) = \sum_{k=0}^{p} \phi^k(\xi)\ \hat{\mathcal{G}}_e^k \tag{21a}$$

$$\mathcal{G}^C(x \in \Omega_e) = \mathcal{G}_e^C(\xi(x)) = \mathcal{G}_e^h(\xi) + (\tilde{\mathcal{G}}_L - \mathcal{G}_L^e)\ C^L(\xi) + (\tilde{\mathcal{G}}_R - \mathcal{G}_R^e)\ C^R(\xi) \tag{21b}$$

$$\frac{\partial \mathcal{G}_e^C}{\partial x}(\xi) = \sum_{k=0}^{p} \hat{\mathcal{G}}_e^k\ \frac{\partial \phi^k}{\partial x}(\xi) + (\tilde{\mathcal{G}}_L - \mathcal{G}_L^e)\ \frac{\partial C^L}{\partial x}(\xi) + (\tilde{\mathcal{G}}_R - \mathcal{G}_R^e)\ \frac{\partial C^R}{\partial x}(\xi) \tag{21c}$$

In accordance with the governing PDE (17), the divergence (21c) of the continuous flux polyomial provides the temporal derivative for the nodal degree of freedom $\hat{U}_e^k$ associated with each solution point $x_{e,k}^S$. Then, the temporal derivative calculation can be paired with an arbitrary explicit time integration scheme to march the solution forward in time.

The CGR-I method for the FR framework described here is fully defined once three properties are chosen: the value of the χ parameter, the correction functions C^L and C^R applied to calculate the connected (18) and semi-connected (19) solutions, and the correction functions C^L and C^R used to calculate the globally continuous flux polynomial (21b).

As with the DG case, Fourier analysis has been performed to investigate the properties of the newly proposed CGR-I scheme within the FR framework. A thorough exploration of the various CGR-I configurations in the $p5$ case has indicated that while the correction function chosen for the connected and semi-connected

solutions governs the spectral radius, it is the correction function chosen for the flux polynomial that governs the order of accuracy. Additionally, the χ parameter has an influence on stability; taking $\chi = 1$ produces many unstable schemes (the choice shown in Table 7 is an exception), so we favor the value $\chi = 1.2$ to maintain stability for various choices of correction functions. Three sample CGR-I configurations, all using a $p5$ discretization ($K = 6$ DOF per element), are listed in Table 7. For each scheme, we give the correction function for populating U^C and U^{SC}, the correction function for populating the globally continuous flux $\mathcal{G}^h$, the method's order of accuracy, and the method's spectral radius. The correction function g_{DG} is the degree K Radau polynomial, well-described by Huynh.[13] The correction function $g_{Lump,Lo}$ is an invention of Huynh;[13] its derivative is zero at $K - 1$ of the Lobatto points on the reference element.

The first scheme in Table 7 has been analyzed for $K \in \{2, 3, 4, 5, 6\}$, and the results are given in Table 8. The orders of accuracy for the CGR-I approach in the FR framework are identical to the orders of accuracy observed within the DG framework. Huynh's manuscript on applying the FR framework to diffusion problems[14] lists the spectral radius and order of accuracy for many FR schemes with $K \in \{2, 3, 4\}$. Specifically, for the $K = 2$ case, three fourth-order accurate schemes (specifically schemes 11, 14, and 15) are listed; our CGR-I scheme has a smaller spectral radius than all three of those schemes. For the $K = 3$ and $K = 4$ cases, our CGR-I method is outperformed by scheme 14 of Huynh,[14] which is a close cousin of Lo & Van Leer's[9] RDG-1x+ method. The commonly applied BR2 approach, described as scheme 2 in Hunyh's work,[14] shows larger spectral radius and equal or worse order of accuracy than our CGR-I scheme for $K \in \{2, 3, 4\}$. Specifically, that scheme has spectral radii of 13, 60, and 170 for $K = 2$, 3, and 4, respectively, while achieving order of accuracy $2(K - 1)$.

Table 7: CGR-I scheme configurations in FR framework, $K = 6$.

χ	C for solution	C for flux	order of accuracy	spectral radius
1.2	g_{DG}	g_{DG}	12	273
1.2	$g_{Lump,Lo}$	g_{DG}	12	225
1.0	g_{DG}	g_{DG}	12	196

Table 8: Fourier analysis of CGR-I scheme in FR framework. $\chi = 1.2$, $C = g_{DG}$ for solution and flux.

K	spectral radius	order of accuracy
2	11	4
3	30	4
4	74	8
5	150	8
6	273	12

VIII. Conclusions

The Interface Gradient Recovery (IGR) Discontinuous Galerkin family for discretizing diffusion problems has been shown to contain multiple attractive numerical schemes; in particular, the GR-II scheme stands out as having small spectral radii and high orders of accuracy relative to the established BR2 and LDG schemes. The GR-II, GR-VI, and CGR-I methods of the family have been shown to generalize to 2D problems and perform well on Cartesian elements, at least in the $p3$ case. However, the GR-II method may be unsuitable for non-Cartesian elements.

The GR-II scheme is the most attractive for 1D problems, but requires the update scheme in each element to rely heavily on information from non-neighbor elements. Modified versions of GR-II may provide improved robustness in multidimensional problems on unstructured grids, as they have been designed to reduce the influence of an element's non-neighbors during the population of the DG residual. The results presented here indicate that the GR-VI method is more robust on non-Cartesian grids, at least in the $p3$ case. Most of the members of the IGR family are free of ambiguously bounded numerical parameters. In the case of the χ parameter of the CGR-I scheme, testing in the 2D case has suggested that $\chi = 2$ is the best

choice in general, but additional analysis could uncover a better choice in the future. The IGR approach for discretizing diffusive fluxes has been effectively combined with the standard DG approach in order to fully discretize advection-diffusion problems. The use of the CGR-I scheme for capturing diffusive fluxes allows the full DG scheme to achieve order $2p+1$ convergence, at least in the $p3$ case.

The CGR-I method is favored by the authors on account of many properties: it has smaller spectral radii than established DG schemes across all values of p analyzed, it is order $2p+2$ accurate for odd p, and it appears robust on non-Cartesian elements. Perhaps most importantly, it maintains the compact computational stencil of the conventional DG method for advective problems. Future work will involve pairing this scheme with an improved DG advection scheme in order to achieve order $2p+2$ convergence for advection-diffusion problems. From Fourier analysis, the CGR-I method also appears effective within the Flux Reconstruction (FR) framework. Compared to most FR schemes for diffusion, it yields relatively small spectral radii while either matching or exceeding their orders of accuracy.

References

[1]Cockburn, B. and Shu, C.-W., "Runge Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems," *Journal of Scientific Computing*, Vol. 16, No. 3, 2001, pp. 89.

[2]Van Leer, Bram; Nomura, S., "Discontinuous Galerkin for Diffusion," *17th AIAA Computational Fluid Dynamics Conference, Toronto, Ontario*, 2005.

[3]Luo, H., Luo, L., Nourgaliev, R., Mousseau, V. A., and Dinh, N., "A reconstructed discontinuous Galerkin method for the compressible NavierStokes equations on arbitrary grids," *Journal of Computational Physics*, Vol. 229, No. 19, sep 2010, pp. 6961–6978.

[4]Ferrero, A., Larocca, F., and Puppo, G., "A robust and adaptive recovery-based discontinuous Galerkin method for the numerical solution of convection-diffusion equations," *International Journal for Numerical Methods in Fluids*, Vol. 77, No. 2, jan 2015, pp. 63–91.

[5]French, D. A., Galbraith, M. C., and Osorio, M., "Error analysis of a modified discontinuous Galerkin recovery scheme for diffusion problems," *Applied Mathematics and Computation*, Vol. 218, No. 13, 2012, pp. 7144–7154.

[6]Johnsen, E., Nair, A., and Varadan, S., "Recovery Discontinuous Galerkin Method for Compressible Turbulence," *21st AIAA Computational Fluid Dynamics Conference*, Fluid Dynamics and Co-located Conferences, American Institute of Aeronautics and Astronautics, jun 2013.

[7]Johnsen, E., Varadan, S., and Van Leer, B., "A Three-Dimensional Recovery-Based Discontinuous Galerkin Method for Turbulence Simulations," *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, American Institute of Aeronautics and Astronautics, Reston, Virigina, jan 2013.

[8]Johnson, P. E. and Johnsen, E., "Progress Towards the Application of the Recovery-Based Discontinuous Galerkin Method to Practical Flow Physics Problems," *46th AIAA Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, Washington, D.C., jun 2016.

[9]Lo, M. and Van Leer, B., "Recovery-Based Discontinuous Galerkin for Navier-Stokes Viscous Terms," *20th AIAA Computational Fluid Dynamics Conference*, Fluid Dynamics and Co-located Conferences, American Institute of Aeronautics and Astronautics, jun 2011.

[10]Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, L. D., "Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems," *SIAM J. Numer. Anal.*, Vol. 39, No. 5, 2001, pp. 1749–1779.

[11]Cockburn, B. and Shu, C.-W., "The Local Discontinuous Galerkin Method for Time-Dependent Convection-Diffusion Systems," *SIAM Journal on Numerical Analysis*, Vol. 35, No. 6, 1997, pp. 2440–2463.

[12]Khieu, L. H. and Johnsen, E., "Analysis of Improved Advection Schemes for Discontinuous Galerkin Methods," *7th AIAA Theoretical Fluid Mechanics Conference*, AIAA Aviation, American Institute of Aeronautics and Astronautics, jun 2014.

[13]Huynh, H. T., "A Flux Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin Methods," *18th AIAA Computational Fluid Dynamics Conference*, Miami, Florida, 2007.

[14]Huynh, H. T., "A Reconstruction Approach to High-Order Schemnes Including Discontinuous Galerkin for Diffusion," *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, American Institute of Aeronautics and Astronautics, Orlando, Florida, jan 2009.

[15]Bassi, F., Crivellini, A., Rebay, S., and Savini, M., "Discontinuous Galerkin solution of the Reynolds-averaged Navier-Stokes and kω turbulence model equations," *Computers & Fluids*, Vol. 34, No. 4-5, 2005, pp. 507–540.

[16]Peraire, J. and Persson, P.-O., "The Compact Discontinuous Galerkin (CDG) Method For Elliptic Problems," *J. Comput. Phys. Lecture Notes in Phys*, Vol. 30, No. 58, jan 1997, pp. 1806–1824.

[17]Prince, P. and Dormand, J., "High order embedded Runge-Kutta formulae," *Journal of Computational and Applied Mathematics*, Vol. 7, No. 1, 1981, pp. 67–75.

[18]Lo, K., *A Space-Time Discontinuous Galerkin Method for Navier-Stokes with Recovery*, Dissertation, University of Michigan, 2011.

[19]Bassi, F., Botti, L., Colombo, A., Di Pietro, D., and Tesini, P., "On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations," *Journal of Computational Physics*, Vol. 231, No. 1, 2012, pp. 45–65.

[20]Khieu, L. H., Fidkowski, K., and Johnsen, E., "Discontinuous Galerkin for Advection with Interface-Centered Reconstruction," *54th AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, San Diego, California, jan 2016.

[21]Toro, E. F., *Riemann solvers and numerical methods for fluid dynamics : a practical introduction*, Springer-Verlag, 1999.

[22]Cash, J. R. and Karp, A. H., "A Variable Order Runge-Kutta Method for Initial Value Problems with Rapidly Varying Right-Hand Sides," *ACM Transactions on Mathematical Software*, Vol. 16, No. 3, 1990, pp. 201–222.

[23]Guo, W., Zhong, X., and Qiu, J.-M., "Superconvergence of discontinuous Galerkin and local discontinuous Galerkin methods: Eigen-structure analysis based on Fourier approach," *Journal of Computational Physics*, Vol. 235, 2013, pp. 458–485.

[24]Landmann, B., Kessler, M., Wagner, S., and Krämer, E., "A parallel, high-order discontinuous Galerkin code for laminar and turbulent flows," *Computers and Fluids*, Vol. 37, 2008, pp. 427–438.

[25]Hartmann, R. and Houston, P., "An optimal order interior penalty discontinuous Galerkin discretization of the compressible NavierStokes equations," *Journal of Computational Physics*, Vol. 227, No. 22, 2008, pp. 9670–9685.