AIAA 2009-3786

AIAA-2009-3786

Analysis and Implementation of Recovery-Based Discontinuous Galerkin for Diffusion[1]

Marcus Lo[2] and Bram van Leer[3]

Department of Aerospace Engineering
University of Michigan
khlo@umich.edu, bram@umich.edu

Abstract

Progress in the analysis and implementation of the Recovery-based DG method (RDG-2x) for diffusion operators is reported. Among the new theoretical insights in the method we present a general stability proof, and an arbitrary-order penalty-term formulation in one dimension. Regarding implementation we discuss the savings afforded by pre-computing at each interface the smooth recovery basis, which is invariant so long as the grid remains fixed. Furthermore, we discuss various versions of the RDG scheme in standard form (RDG-1x), and the extension of RDG to nonlinear diffusion terms such as appear in the Navier-Stokes equations. In addition we discuss recovery based on more than two neighboring cells, and not just for diffusion. A section with numerical results compares different implementations of RDG-1x with RDG-2x.

1 Introduction

Since its introduction at the 17th AIAA Computational Fluid Dynamics Conference, Toronto, 2005[1], and further reporting at the 18th conference in Miami, 2007[2], the development of the Recovery-based Discontinuous Galerkin (RDG) method has led to further theoretical and practical insights, in part owing to the opportunity we had to port RDG into the CFD software package HOME of HyPerComp (Westlake, CA) under an AFOSR-funded STTR contract, but also to the analysis, testing and applications contributed by other researchers [3, 4, 5].

In this paper we report the latest developments in RDG. We first review the definition and chief properties of RDG (Sec. 2). Then follow separate sections on progress in the analysis (Sec. 3) and the implementation (Sec. 4) of the method. Of particular importance are the availability of at least an *Ansatz* to a general stability proof (Sec. 3.1), and various ways to implement recovery in the "standard" DG form resulting after partially integrating only once (RDG-1x, Sec. 4.1). Next, a separate section (Sec. 5) is devoted to recovery based on more

[1]

[2]Doctoral candidate, member AIAA

[3]Professor, Fellow AIAA

than two neighboring cells, and not just for diffusion. The last section contains a numerical comparison of different implementations of RDG-1x with RDG-2x; the computations are done on a Cartesian grid, which brings out the accuracy differences most clearly.

2 Recovery and Discontinuous Galerkin

Recovery is a name we gave[4] to a *weak* interpolation technique for locally reconstructing, as well as possible with the available degrees of freedom, a function of which an L_2-projection is known on the meshes of a computational net; the recovered function must have the same L_2 projection. The latter property provides the "quality control" of the interpolation, just as in *strong* interpolation function- and/or derivative-values must be matched by the interpolant in *points*.

Recovery is especially useful when including diffusion in a DG discretization, as the solution and its gradient are discontinuous at the interfaces where diffusive fluxes must be computed. In the RDG approach the diffusive fluxes are computed from a smooth, locally recovered solution that in the weak sense is indistinguishable from the discontinuous discrete solution[1, 2]. This eliminates the introduction of *ad hoc* penalty or coupling terms found in traditional DG methods; in the simple case of 1-D diffusion the recovery principle can be shown to generate a string of penalty-like terms not encountered in standard DG methods (see Section 3.2).

The most attractive feature of RDG is the simplicity and generality of the recovery principle, which immediately allows one to create and further develop DG diffusion schemes. In addition, these schemes are of higher accuracy than any other DG diffusion schemes currently in use, while boasting the smallest eigenvalues, hence the largest stability range for explicit time-marching; see Huynh, 2009 [3]. This is particularly clear when comparing the methods on a uniform Cartesian mesh - Huynh analyzed and tested 16 of them. For example, if the numerical solution u lies in a piecewise polynomial space that is the tensor product of one-dimensional polynomial spaces of degree p, then the order of accuracy of RDG is $3p+2$ or $3p+1$ for p even or odd, respectively[5]. The largest negative-real eigenvalue for $p=3$, when RDG is 10th-order accurate, equals -134, to be compared with -340 for Bassi-Rebay-2 (6th-order accurate)[7], and -878 for CDG (8th-order accurate)[8]. While all these methods revert to the order of accuracy $p+1$ on an irregular grid, the pedigree of RDG remains visible in the lower error coefficient and smaller eigenvalues.

There are two basic versions of RDG: RDG-2x and RDG-1x. RDG-2x is the original version and starts from a weak form of the diffusion equation[6],

$$U_t = D\nabla \cdot \nabla U, \tag{1}$$

[4]The use of the term "recovery" in CFD is due to Bill Morton [6], who indicated with it the reconstruction of a shock discontinuity from the structure of a "captured" numerical shock. This is truly the opposite of what we are doing in RDG for diffusion: recovering a smooth function from a discontinuous solution.

[5]An earlier supposition[2] that the order might be exponential (2,4,8,... for p=0,1,2,...), turned out to be false.

[6]The diffusion equation is taken linear in this and other sections; Sec. 4.2 deals with the nonlinear-system case.

obtainable after multiplication with a test function and integration by parts *twice* over the interior of an element Ω_j with boundary $\partial\Omega_j$:

$$\begin{aligned}\int\int\int_{\Omega_j}(v_k)_j U_t d\Omega &= D\int\int_{\partial\Omega_j}\{(v_k)_j\nabla U - U\nabla(v_k)_j\}\cdot\hat{n}\; d\partial\Omega \\ &+ D\int\int\int_{\Omega_j} U\nabla\cdot\nabla(v_k)_j d\Omega, \quad k=1,...,K. \end{aligned} \tag{2}$$

Here U is the true solution, D is a constant scalar diffusion coefficient and the v_k are the test functions, also used as the basis functions; these are polynomial on Ω_j and zero elsewhere. In a DG method U is replaced in the above equation by a numerical approximation u that in each element lies in the span of the basis $\{v_k\}$. This numerical solution is discontinuous at $\partial\Omega_j$; if the boundary integral in (2) is evaluated on the inside of the boundary, the method will be inconsistent since there will be no coupling with the neighboring cells.

In RDG the coupling is provided by replacing u on a facet of the boundary by f, a smooth solution recovered from the information in the two abutting elements. The recovered solution $f_{j,j+1}$, centered at the interface between elements Ω_j and Ω_{j+1}, is uniquely determined by making it indistinguishable from u in the weak sense in the contributing elements, that is,

$$\int\int\int_{\Omega_j}(v_k)_j f_{j,j+1} d\Omega = \int\int_{\Omega_j}(v_k)_j u\; d\Omega, \quad k=1,...,K, \tag{3}$$

$$\int\int_{\Omega_{j+1}}(v_k)_{j+1} f_{j,j+1} d\Omega = \int\int_{\Omega_{j+1}}(v_k)_{j+1} u\; d\Omega, \quad k=1,...,K. \tag{4}$$

In order to satisfy these equations, f must lie in the space of the recovery basis w_k, with $k=1,..,2K$; for a detailed description of this basis see [2].

Each face of Ω_j generates a different recovered solution $f_{j,..}$; thus, there are as many surrogates for u or U as there are faces to the element. Inserting these into the boundary integral yields the RDG scheme:

$$\begin{aligned}\int\int\int_{\Omega_j}(v_k)_j u_t d\Omega &= D\int\int_{\partial\Omega}\{(v_k)_j\nabla f_{j,..} - f_{j,..}\nabla(v_k)_j\}\cdot\hat{n}\; d\partial\Omega \\ &+ D\int\int\int_{\Omega_j} u\nabla\cdot\nabla(v_k)_j d\Omega, \quad k=1,...,K. \end{aligned} \tag{5}$$

It is worth noticing that replacing u by any of the $f_{j,..}$, or an average of these, in the volume integral on the right-hand side does not change the value of the integral (see [2]).

For future reference we also give the weak form of the diffusion equation resulting after integrating by parts *once*:

$$\begin{aligned}\int\int\int_{\Omega_j}(v_k)_j U_t d\Omega &= D\int\int_{\partial\Omega_j}(v_k)_j\nabla U\cdot\hat{n}\; d\partial\Omega \\ &- D\int\int\int_{\Omega_j}\nabla U\cdot\nabla(v_k)_j d\Omega, \quad k=1,...,K; \end{aligned} \tag{6}$$

all traditional DG diffusion schemes are based on this form, which is why we have called it "standard." In [9] we indicated how to implement the DG form (6) using recovery, and called the resulting discretization RDG-1x. We will discuss several implementation options in Sec. 4.1.

3 Theoretical results

3.1 A stability proof

To assess the stability of RDG schemes, RDG-2x in particular, we replace the test function v_k in Eqn. (5) by the solution u, which combines all test functions, and integrate over the entire computational domain $\mathcal{V}$ with boundary $\mathcal{S}$; this yields an equation for the evolution of the energy of the solution on the domain:

$$\begin{aligned}\int\int\int_{\mathcal{V}} \frac{\partial}{\partial t}\left(\frac{u^2}{2}\right) d\mathcal{V} &= D\int\int_{\mathcal{S}} (u\nabla f - f\nabla u)\cdot\hat{n}\; d\mathcal{S} \\ &+ D\int\int_{\mathcal{E}} [u\nabla f - f\nabla u]\cdot\hat{n}\; d\mathcal{E} \\ &+ D\int\int\int_{\mathcal{V}} u\nabla\cdot\nabla u\; d\mathcal{V}. \end{aligned} \tag{7}$$

Here $\mathcal{E}$ represents the union of all interior cell faces; the jump $[\cdot]$ and normal $\hat{n}$ at an interface are matched such that if $\hat{n}$ is chosen to point from cell j_1 to cell j_2, $[\cdot]$ is defined as $(\cdot)_{j_1} - (\cdot)_{j_2}$. Note that we have replaced U and ∇U on the domain boundary by their recovered values f and ∇f; it is f, not u, to which the boundary condition must be applied.

We now *undo* the second partial integration so as to arrive at the following equation:

$$\begin{aligned}\int\int\int_{\mathcal{V}} \frac{\partial}{\partial t}\left(\frac{u^2}{2}\right) d\mathcal{V} &= D\int\int_{\mathcal{S}} \{u\nabla f + (u-f)\nabla u\}\cdot\hat{n}\; d\mathcal{S} \\ &+ D\int\int_{\mathcal{E}} [u\nabla f + (u-f)\nabla u]\cdot\hat{n}\; d\mathcal{E} \\ &- D\int\int\int_{\mathcal{V}} \nabla u\cdot\nabla u\; d\mathcal{V}. \end{aligned} \tag{8}$$

If the energy can be shown to decay in time, this is proof of global stability. It generally means proving that the last term on the right-hand side of Eqn. (8), i. e.,the volume integral, which is negative and of magnitude $O(1)$, dominates the other terms. For $p = 0$ the volume integral vanishes, and a separate proof is needed,

We assume that $\mathcal{V}$ and $\mathcal{S}$ are both of magnitude $O(1)$, and h is a measure of the average element size. Furthermore, we assume u lies in a polynomial space of order p and the solution is smooth to begin with (discontinuous initial values are excluded). This allows us to estimate the boundary- and interior-face terms.

The integral over the interior faces is larger than the integral over the domain boundary, because the area of $\mathcal{E}$ is $1/O(h)$, as compared to $O(1)$ for $\mathcal{S}$. Using angled brackets to indicate the interface average of a function, the integrant of the former integral can be reduced as follows:

$$\begin{aligned}[u\nabla f + (u-f)\nabla u]\cdot\hat{n} &= \{[u]\nabla f + [u-f]\,\langle\nabla u\rangle + \langle u-f\rangle\,[\nabla u]\}\cdot\hat{n} \\ &= \{[u](\nabla f + \langle\nabla u\rangle) + (\langle u\rangle - f)[\nabla u]\}\cdot\hat{n}. \end{aligned} \tag{9}$$

Here $[u]$ and $\langle u\rangle - f$ are both of order $O(h^{p+1})$, while $[\nabla u] = O(h^p)$ for $p \geq 1$ and 0 for $p = 0$. Hence, the two terms on the final right-hand side of (9) are $O(h^{p+1})$ and $O(h^{2p+1})$,

respectively, for $p \geq 1$. The first term dominates, so the integral over $\mathcal{E}$, with total area $\sim 1/h$, becomes $O(h^p)$ for $p \geq 1$. On a sufficiently fine grid this integral therefore becomes negligible compared to the negative volume integral.

Now consider the domain-boundary terms. Let us first assume that the boundary is adiabatic, i. e., the diffusive flux $D\nabla f \cdot \hat{n}$ vanishes at the boundary. This removes the first term of the boundary integral in (8). By realizing that $u = f + (u - f)$ we may reduce the integral to

$$D \int \int_{\mathcal{S}} \nabla \frac{(u-f)^2}{2} \cdot \hat{n} \; d\mathcal{S}, = O(h^{2p+1}), \tag{10}$$

which is small on a sufficiently fine grid, for all p. Thus, on a sufficiently fine grid the negative volume integral dominates both boundary- and interior-face integrals for $p \geq 1$; the solution energy decreases in time, implying stability.

For a Dirichlet boundary condition, e. g., $f = 0$ on $\mathcal{S}$, the integrant in the domain-boundary term reduces to

$$u\nabla f + (u-f)\nabla u = (u-f)\nabla(u+f) = O(h^{p+1}), \tag{11}$$

which vanishes with h for all p. Again, for $p \geq 1$ the negative volume integral dominates both boundary- and interior-face integrals on a sufficiently fine grid, causing the solution energy to decay.

For $p = 0$ we have $\nabla u = 0$, so the volume integral in (8) vanishes; the integral over $\mathcal{E}$ now takes over its role. The integrant (9) reduces to $[u]\nabla f \cdot \hat{n}$; in this case $[u] = O(h)$ and the recovery procedure yields

$$\nabla f \cdot \hat{n} = -\frac{[u]}{h_{cn}}, \tag{12}$$

where h_{cn} is the distance between the centroids of neighboring elements measured along their interface-normal. Therefore,

$$[u]\nabla f \cdot \hat{n} = -\frac{[u]^2}{h_{cn}} = -|O(h)|, \tag{13}$$

and the integral over $\mathcal{E}$ becomes negative of order $O(1)$. This dominates the domain-boundary integral in both Neumann and Dirichlet cases, so stability is ensured.

Note that (13) may be interpreted as stemming from the Baker[10]-Arnold[11] penalty term $-\mu D[v][u]/h$, with positive $\mu = O(1)$; cf. Sec. 3.2. For $p = 0$ this term has to provide the physical content of the diffusion operator, so the value of μ must be chosen correctly with regard to consistency [1]. The recovery procedure automatically generates the term with the proper value.

This completes the stability proof for RDG-2x; the proof for the "naive" form of RDG-1x (see Sec. 4.1) is obtained by omitting the terms proportional to $u - f$ in the integrals over $\mathcal{E}$ and $\mathcal{S}$ in Eqn. (8).

The global RDG form (8) is not just useful in a stability proof, but, as pointed out by Marshall Galbraith (University of Cincinnati), it actually suggests an excellent local

implementation formula that is equivalent to RDX-2x:

$$\begin{aligned}\int\int\int_{\Omega_j}(v_k)_j u_t d\Omega &= D\int\int_{\partial\Omega_j}\{(v_k)_j\nabla f+(u-f)\nabla(v_k)_j\}\cdot\hat{n}\ d\partial\Omega \\ &- D\int\int\int_{\Omega_j}\nabla(v_k)_j\cdot\nabla u\ d\Omega,\quad k=1,...,K. \end{aligned} \tag{14}$$

The formula looks like a hybrid of RDG-1x and RDG-2x: it has the volume integral of RDG-1x but shows two boundary terms as in RDG-2x. The great advantage is that one does not have to use recovered information in evaluating ∇u in the volume integral; one may use u "as is." The second boundary term looks somewhat like a penalty term; it contains the information to elevate the accuracy of naive RDG-1x (without recovered volume integral) to that of RDG-2x, most noticeable on a Cartesian grid. See further Secs. 4.1 and 6.

3.2 Penalty-term expansion in one dimension

We have succeeded in writing the 1-D form of the RDG-2x scheme of arbitrary order as an expansion in penalty-like terms. On a uniform grid it reads:

$$\begin{aligned}\frac{1}{D}\int_{\Omega_j} vu_t\ dx &= -\left(\langle u_x\rangle[v]|_{j+\frac{1}{2}}+\langle u_x\rangle[v]|_{j-\frac{1}{2}}\right)-\left(\langle u\rangle[v_x]|_{j+\frac{1}{2}}+\langle u\rangle[v_x]|_{j-\frac{1}{2}}\right)-\int_{\Omega_j}v_x u_x\ dx \\ &+\frac{R_0}{\Delta x}\left([u][v]|_{j+\frac{1}{2}}+[u][v]|_{j-\frac{1}{2}}\right) \\ &+R_1\Delta x\left([u_x][v_x]|_{j+\frac{1}{2}}+[u_x][v_x]|_{j-\frac{1}{2}}\right) \\ &+R_2\Delta x\left([u_{xx}][v]|_{j+\frac{1}{2}}+[u_{xx}][v]|_{j-\frac{1}{2}}\right) \\ &+R_3\Delta x^3\left([u_{xxx}][v_x]|_{j+\frac{1}{2}}+[u_{xxx}][v_x]|_{j-\frac{1}{2}}\right) \\ &+R_4\Delta x^3\left([u_{xxxx}][v]|_{j+\frac{1}{2}}+[u_{xxxx}][v]|_{j-\frac{1}{2}}\right) \\ &+R_5\Delta x^5\left([u_{xxxxx}][v_x]|_{j+\frac{1}{2}}+[u_{xxxxx}][v_x]|_{j-\frac{1}{2}}\right) \\ &+R_6\Delta x^5\left([u_{xxxxxx}][v]|_{j+\frac{1}{2}}+[u_{xxxxxx}][v]|_{j-\frac{1}{2}}\right)+\ldots; \end{aligned} \tag{15}$$

here the jump of a function at an interface is always computed as right value minus left value. It is seen that the i-th bilinear "penalty" term, with coefficient R_i, contains the jump in the i-th derivative of u and the jump in either v or v_x, depending on whether the index i is even or odd, respectively. For an RDG scheme based on a polynomial space of degree p, the index runs from 0 to p. The coefficients R_i are given in Table 1 for schemes up to $p = 5$.

Note that for $p > 1$ the spatial operators are not symmetric; however, the real part of their eigenvalues is found (by numerical evaluation) to be negative, allowing for stable update schemes. These discretizations are also highly accurate: the truncation error is of the order $3p + 2$ or $3p + 1$ for p even or odd, respectively.

p	$\mu = -R_0$	R_1	R_2	R_3	R_4	R_5
0	1					
1	$\frac{9}{4}$	$\frac{1}{12}$				
2	$\frac{15}{4}$	$\frac{3}{64}$	$-\frac{3}{80}$			
3	$\frac{175}{32}$	$\frac{1}{32}$	$-\frac{5}{192}$	$-\frac{1}{6720}$		
4	$\frac{945}{128}$	$\frac{35}{1536}$	$-\frac{5}{256}$	$-\frac{5}{86016}$	$\frac{5}{129024}$	
5	$\frac{4851}{512}$	$\frac{9}{512}$	$-\frac{63}{4096}$	$-\frac{1}{36864}$	$\frac{7}{369640}$	$\frac{1}{9461760}$

Table 1: Coefficients of the penalty-like terms in 1-D RDG for $p \leq 5$.

4 Progress in implementation

4.1 RDG schemes in standard form

As reported in [9], it is possible to insert the information obtained in the recovery procedure into the standard DG form (6) so as to arrive at an RDG-1x discretization that competes in accuracy with the RDG-2x discretization with equal p. In this section we review a hierarchy of three RDG-1x implementations based on (6), and one based on the quasi-standard form (14), which is really equivalent to RDG-2x.

The most naive form of RDG-1x has u replaced by f in the boundary term, but ∇u in the volume integral computed "as is;" this leads to an accuracy that is considerably lower than that of RDG-2x with equal p. For instance, for $p = 1$ on a 2-D Cartesian grid, "naive" RDG-1x is only second-order accurate, while RDG-2x is fourth-order accurate. The loss of accuracy is solely caused by the absence from Eqn. (6) of the $O(h^{2p+3}, h^{p+2})$ second boundary term present in Eqn. (14).

Next, we consider replacing ∇u in the volume integral by $\nabla \overline{f}$, where $\overline{f}$ is the average of the solutions recovered on cell Ω_j and each of its neighbors. The accuracy resulting from this procedure is expected to be better than that of naive RDG-1x, but only by a factor. The reason is that the recovered solutions are particularly accurate only in their dependence on the coordinate normal to the interface of the two participating cells [2]; their representation of the solution in the direction parallel to the interface is not better than order p, as for u itself.

Aware of the anisotropy in the accuracy of the recovered solutions, we recommended in [9] to evaluate ∇u using only the interface-normal derivatives $\partial f_{j,l}/\partial n_{j,l}$,of the solutions recovered on Ω_j and Ω_l, where l cycles through all neighbors of element j. Since there are more neighbors than dimensions, even for a simplex cell, this leads to a small least-squares problem that is always well conditioned; note that its size is independent of p. With the

gradient thus recovered, which we shall call $\nabla \hat{f}_j$, the accuracy of RDG-1x competes with that of RDG-2x on a Cartesian grid. But the implementation is more involved than the quasi-standard implementation (14), which we expect to become the standard of implementation of RDG for diffusion.

It is worth reiterating from [9] that in our first paper on RDG [1] we derived, by truncation-error minimization, a remarkably accurate member of the classical DG family [12]

$$\begin{aligned}\int_{I_j} v u_t dx &= -D\left(< u_x > [v]\Big|_{j+\frac{1}{2}} + < u_x > [v]\Big|_{j-\frac{1}{2}}\right) - D\int_{I_j} v_x u_x dx \\ &+ \sigma D\left(< v_x > [u]\Big|_{j+\frac{1}{2}} + < v_x > [u]\Big|_{j-\frac{1}{2}}\right) \\ &- \frac{\mu D}{\Delta x}\left([v][u]\Big|_{j+\frac{1}{2}} + [v][u]\Big|_{j-\frac{1}{2}}\right), \end{aligned} \tag{16}$$

with

$$\sigma = \frac{1}{4}, \quad \mu = \frac{9}{4}. \tag{17}$$

This scheme turns out to be an RDG-1x scheme with ∇u in the volume integral evaluated as $\nabla \hat{f}$ or $\nabla \overline{f}$ (these are equal in one dimension).

In Sec. 6 the above discretizations are numerically tested and compared on a Cartesian grid. Note that even on a regular triangular grid, let alone on an on an irregular grid, all RDG-1x and RDG-2x schemes reduce to the order of accuracy $p+1$ [9]; differences in error level and in eigenvalue magnitude may still remain. In particular, the eigenvalues of RDG-1x with recovery of ∇u appear to be more than 50% larger than those of RDG-2x for $p = 1$ and 2.

4.2 RDG for a nonlinear diffusion system

While the twice-integrated form (2) was essential for the development of the RDG approach, and leads to the simplest discretization of a linear scalar diffusion equation, it may not necessarily be the preferred choice when considering the DG discretization of a *nonlinear system* of diffusion operators, such as contained in the Navier-Stokes equations. Below the two weak nonlinear forms are displayed as they would appear in DG discretizations of the Navier-Stokes equations in conservation form (Euler terms suppressed); bold face indicates a vector or matrix in the 5-dimensional space of state variables, wheras an arrow indicates a vector in 3-D physical space. For instance, $\vec{\vec{\mathbf{D}}}$ indicates an 3×3 array of 5×5 matrix-valued diffusion coefficients corresponding to diffusive fluxes of all state quantities in all coordinate directions. Both the dot product and the gradient operator refer to physical space.

Differential form:

$$\mathbf{U}_t = \vec{\nabla} \cdot \vec{\vec{\mathbf{D}}} \vec{\nabla} \mathbf{U}; \tag{18}$$

Weak form once partially integrated (RDG-1x):

$$\begin{aligned}\int\int\int_{\Omega_j} (v_k)_j \mathbf{u}_t d\Omega &= \int\int_{\partial\Omega_j} (v_k)_j (\vec{\vec{\mathbf{D}}} \vec{\nabla} \mathbf{f}) \cdot \hat{n}\ d\partial\Omega \\ &- \int\int\int_{\Omega_j} (\vec{\vec{\mathbf{D}}} \vec{\nabla} \hat{\mathbf{f}}) \cdot \vec{\nabla} (v_k)_j d\Omega, \quad k = 1, ..., K; \end{aligned} \tag{19}$$

Weak form twice partially integrated (RDG-2x):

$$\begin{aligned}\int\int\int_{\Omega_j}(v_k)_j\mathbf{u}_t d\Omega &= \int\int_{\partial\Omega_j}[(v_k)_j\vec{\vec{\mathbf{D}}}\vec{\nabla}\mathbf{f} - \{\vec{\vec{\mathbf{D}}}\vec{\nabla}(v_k)_j\}\mathbf{f}]\cdot\hat{n}\; d\partial\Omega \\ &+ \int\int\int_{\Omega_j}[\vec{\nabla}\cdot\{\vec{\vec{\mathbf{D}}}\vec{\nabla}(v_k)_j\}]\mathbf{u}\; d\Omega, \quad k=1,...,K. \end{aligned} \tag{20}$$

The first weak form has the advantage that we may use the discrete solution $\mathbf{u}_j$ "as is" to compute the volume integral, rather than the set of recovered solutions $\mathbf{f}_{j,..}$ as recommended for the first form. This advantage, though, disappears when computing on irregular, unstructured grids, where all DG methods with a basis of order p have their accuracy reduced to the order $p+1$, so $\hat{\mathbf{f}}$ in the volume integral of Eqn. (19) may as well be simplified to $\mathbf{u}$.

Moreover, the first form includes $\mathbf{u}$ or $\mathbf{f}$ on the right-hand side solely in the combination $\vec{\vec{\mathbf{D}}}\vec{\nabla}\mathbf{u}$ or $\vec{\vec{\mathbf{D}}}\vec{\nabla}\mathbf{f}$; this expression greatly simplifies when switching to the primitive variables $\mathbf{w} = (\rho, u, v, w, T)^T$ favored for the Navier-Stokes equations. We have

$$\vec{\vec{\mathbf{D}}}\vec{\nabla}\mathbf{u} = \vec{\vec{\mathbf{M}}}\vec{\nabla}\mathbf{w}, \tag{21}$$

where the blocks of $\vec{\vec{\mathbf{M}}}$ are very sparse.

In contrast, the second weak form also contains the combinations $\{\vec{\vec{\mathbf{D}}}\vec{\nabla}(v_k)_j\}\mathbf{u}$ and $\{\vec{\vec{\mathbf{D}}}\vec{\nabla}(v_k)_j\}\mathbf{f}$, which do not benefit from simplification by switching to primitive variables. Even the quasi-standard form (14) extended to a nonlinear system, retains the expression in the second boundary term:

$$\begin{aligned}\int\int\int_{\Omega_j}(v_k)_j\mathbf{u}_t d\Omega &= \int\int_{\partial\Omega_j}[(v_k)_j\vec{\vec{\mathbf{D}}}\vec{\nabla}\mathbf{f} + \{\vec{\vec{\mathbf{D}}}\vec{\nabla}(v_k)_j\}(\mathbf{u}-\mathbf{f})]\cdot\hat{n}\; d\partial\Omega \\ &- \int\int\int_{\Omega_j}(\vec{\vec{\mathbf{D}}}\vec{\nabla}\mathbf{u})\cdot\vec{\nabla}(v_k)_j d\Omega, \quad k=1,...,K; \end{aligned} \tag{22}$$

Summarizing: RDG-1x, with the simplest possible evaluation of the volume integral, may be the most effective choice for use with the Navier-Stokes equations. However, there is no body of practical numerical evidence yet to support this presumption; furthermore, the aspect of a possibly reduced stability range associated with larger eigenvalues needs to be weighed in. The quasi-standard scheme (14) remains a strong contender.

4.3 Use of the recovery basis

As discussed at length in [13] and [9], the recovery procedure on $\Omega_j \bigcup \Omega_{j+1}$ can be first applied to the discontinuous basis functions $(v_k)_j, k=1,..,K$, and $(v_k)_{j+1},\ k=1,..,K$, yielding a set of smooth basis functions $(w_l)_{j,j+1},\ l=1,..,2K$. The principal recovery theorem says that the expansion of u on $\Omega_j \bigcup \Omega_{j+1}$ in the discontinuous basis functions and the expansion of f in the corresponding smooth basis functions have identical coefficients. Since the recovery process is only influenced by the geometry of Ω_j and Ω_{j+1}, the recovery basis will not change if the grid stays fixed. Hence, for calculations on a fixed grid it suffices to compute the smooth recovery bases for all pairs of neighboring elements at the start of a calculation, and

store these; no further recovery needs to be done at any later time. Recovery thus reduces to a pre-processing step for an entire calculation, yielding large computational savings. Our numerical experiments so far have always been based on fixed grids; for unstructured grids our RDG schemes have indeed been coded using pre-computed recovery bases.

5 Isotropic recovery and cell-centered recovery

So far we have limited ourselves to recovering a solution on the union of two neighboring, interface-sharing cells; we shall refer to this as a *binary* recovery. As pointed out before, such a recovery is anisotropic in its accuracy [2]: its representation of the solution in the direction normal to the interface is accurate to the order $2p+1$, while the representation in any direction parallel to the interface is only of the order p.

The anisotropy could be remedied, if so desired, by adding some information from neighboring cells that span a distance along the interface. Figure 1(a) shows, in two dimensions, the obvious extended Cartesian stencil, while the "butterfly" in Figure 1(b) is a possible extended stencil for an unstructured triangular grid. When recovering the solution on these stencils it makes sense to match fewer data in the more remote cells than in the principal pair.

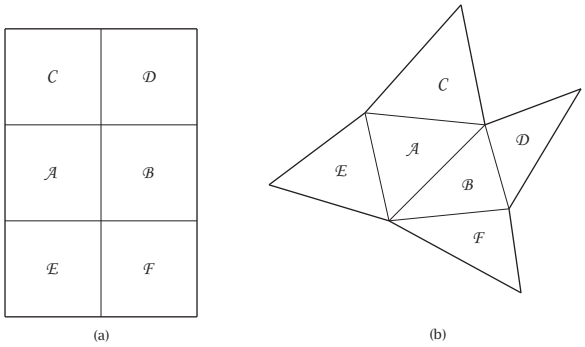

Figure 1: Extended Cartesian (a) and triangular "butterfly" (b) stencils for more isotropic interface-based recovery.

For instance, with $p=1$ the principal cells $\mathcal{A}$ and $\mathcal{B}$ each carry three independent data (defining a linear function) that must be matched by the recovered solution; in each of the four remote cells $\mathcal{C}$-$\mathcal{F}$ we shall only match the cell-average. Thus, the number of recovery equations increases from 6 to 10. The latter is the correct number of data for a complete cubic basis, but the collection of basis functions representable on the rectangle or "butterfly" is not quite right. Calling the coordinate normal to the principal interface r and the coordinate along the interface s, it is seen that we will fail to obtain a coefficient for the function $r^2 s$; instead, we obtain a coefficient for the function rs^3, not needed for the cubic basis. In the end, the recovery is only quadratically correct in all directions.

To reduce the "waste" we might constrain the recovered function to have the correct average value in $\mathcal{C} \bigcup \mathcal{D}$ and $\mathcal{E} \bigcup \mathcal{F}$, rather than in each of the four cells separately. This

reduces the number of recovery equations to 8; the coefficients of rs^2 and rs^3 no longer can be found. Left over is a quadratic basis augmented with r^3 and s^3.

Initially we asked ourselves if it would be possible to obtain the detailed s-dependence at the interface between $\mathcal{A}$ and $\mathcal{B}$ purely from binary recoveries, i. e., from the cell pairs $(\mathcal{A}, \mathcal{C})$, $(\mathcal{A}, \mathcal{E})$, $(\mathcal{B}, \mathcal{D})$ and $(\mathcal{B}, \mathcal{F})$. This is in the same vein as the gradient reconstruction (Sec. 4.1) for computing the RDG-1x volume integral, say, in cell $\mathcal{A}$; there we intentionally avoided performing a new, *cell-centered* recovery on the union of $\mathcal{A}$ and its neighbors $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{E}$. As it turned out, the binary-recovery information increases the accuracy of RDG-1x only on a Cartesian grid; even on a regular triangular grid [9] the accuracy falls back to the usual order $p + 1$. So we are pessimistic about re-using binary-recovery information to enhance RDG accuracy.

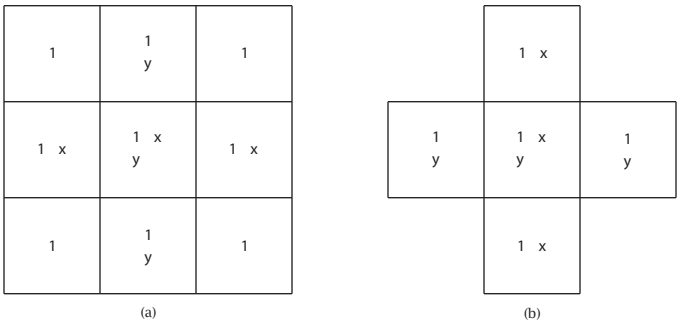

Figure 2: Cartesian stencil (a) used by Park et al. [5] for cRDG, and an especially thrifty stencil (b) for cubic cRDG.

Nourgaliev et al. [4, 5], though, have begun to explore cell-centered recovery to enhance the accuracy of DG; we shall refer too this as cRDG, to contrast it with interface-centered recovery-based DG or iRDG. Rather than removing an interface discontinuity, the goal of cRDG is to increase the order of the solution basis in the central cell by weak interpolation. The increase could of course simply be achieved by starting out with a higher-order DG basis; the advantage of increasing the order by recovery is that the number of unknowns does *not* increase. For Nourgaliev et al. this is the prime motivation, as their temporal discretization is implicit and treated with a Newton-Krylov solver. The computational expense of cell-centered recovery is small change compared to the expense of the implicit solver further down the line.

The 2-D numerical example of cRDG given in [5] is based on a manufactured traveling-wave Navier-Stokes solution; it starts from a $p = 1$ basis and uses the stencil and basis functions displayed in Figure 2(a) for the cell-centered recovery. This recovery-enhanced solution provides the input for the Riemann solver used to obtain the inviscid interface fluxes. Diffusive interface fluxes are computed on the stencil shown in Figure 1(a); not mentioned in the paper is whether cRDG and iRDG are applied in parallel, or cRDG pre-processes the data used in iRDG.

The cRDG-enhanced solution has 15 parameters but is not fully quartic; two mixed terms, x^3y and y^3x, are missing, while x^5 and y^5 are included. The numerical results show an increase from 2nd-order accuracy without cRDG to 6th-order accuracy with cRDG. The

former result is puzzling, as a Navier-Stokes DG calculation with upwind Euler fluxes and diffusive fluxes from binary recovery should yield third-order accuracy; the latter result may be flattered because the test problem could be too simple. Nevertheless, the increase in accuracy is impressive and suggests iRDG was applied on top of cRDG.

With regard to cutting waste we offer the observation that the combination of the Cartesian stencil and basis functions shown in Figure 2(b) is about as thrifty as it gets: from a DG basis with $p = 1$ it recovers a complete cubic basis, with only one parameter to spare. The extra parameter results from the fact that there are two ways to obtain the coefficient of the function xy in the central cell: by using horizontal neighbors or by using vertical neighbors. The difference between the two is a combination of 4th-order terms.

Based on our experience with iRDG, though, we expect the gains on an unstructured triangular grid to be rather modest. For instance, when enhancing a $p = 1$ basis in cell $\mathcal{A}$, using all the information from its neighbors $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{E}$ (12 data in total), still only a quadratic basis can be recovered without overly constraining the grid. In particular, if the centroids of 3 triangles line up, as in Figure 3, there is insufficient coverage of 2-D space. In the displayed stencil no coefficient can be found for the function xy^2. Similarly, if only the solution-averages are matched in the neighboring cells (6 data in total), the same degenerate geometry even prohibits going beyond a $p = 1$ basis. In this case no coefficient of xy can be found.

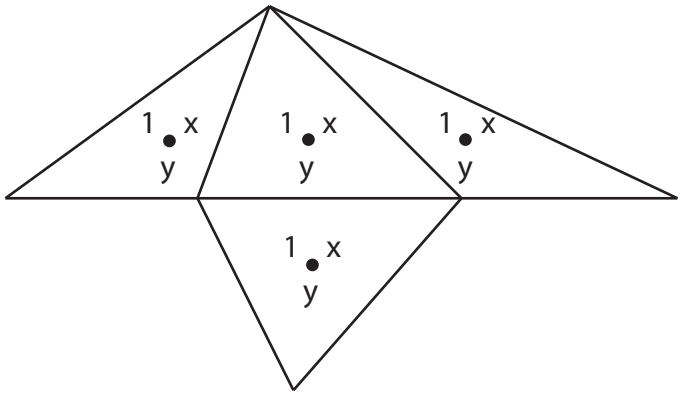

Figure 3: Degenerate triangular stencil: three centroids are aligned along the x-axis, spoiling the potential for cubic recovery.

We may regard cRDG as a cross between a finite-volume and a finite-element method; in fact, a finite-volume Euler method such as MUSCL may be interpreted as a cRDG method with $p = 0$, where the recovery aims at increasing the order of the solution representation to $p = 1$. The use of cell-centered recovery as a pre-processing step to enhance a DG solution offers a wealth of possibilities, especially when combined with interface-based recovery. We are bound to see more of this approach.

6 Numerical results

In order to demonstrate the differences in accuracy among the various versions of RDG-1x, and with respect to RDG-2x, we solved a Dirichlet problem on the square $[0,1] \times [0,1]$ with exact steady solution

$$U(x,y) = \cos(2\pi x) + \cos(2\pi y). \tag{23}$$

For our purpose it sufficed to study the case $p = 1$. At the boundary, recovery in the normal direction involved the next cell inward to make it cubic [2], as needed to match the accuracy at interior interfaces. We tested three RDG-1x schemes, with the following features (cf. Sec. 4.1):

(A) volume integral computed with ∇u,

(B) volume integral computed with $\nabla \overline{f}$,

(C) volume integral computed with $\nabla \hat{f}$;

for comparison, results were also obtained with RDG-2x cast in

(D) quasi-standard form (14).

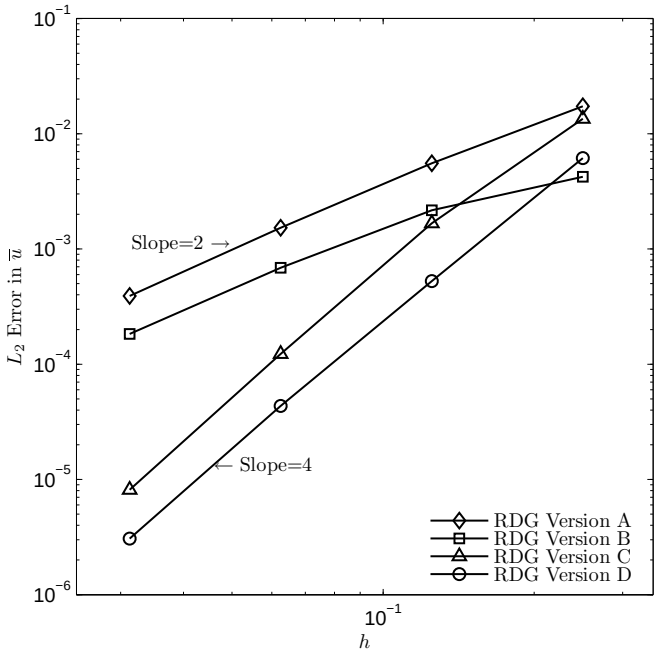

Figure 4: Grid-convergence study: L_2 error in cell-average versus mesh width for three RDG-1x schemes and the RDG-2x scheme (14). Dirichlet problem on Cartesian grid; $p = 1$.

The convergence results displayed in Figure 4 (L_2-error in cell-averages versus mesh-width h) confirm the theory-based prognosis made in Sec. 4.1. Versions (A) and (B) both are

second-order accurate, but (B) has about half the error of (A) owing to the averaging of accurate with inaccurate information in $\overline{f}$. Version (C) is fourth-order accurate, though its error coefficient is more than twice that of the RDG-2x version (D). Not shown is that for (C) and (D) the other solution coefficients, $\overline{u_x}$ and $\overline{u_y}$, are converging faster than $\overline{u}$, possibly heading toward 5th-order accuracy on still finer grids.

7 Conclusions

During the past year Recovery-based Discontinuous Galerkin methods have been put on a firm footing. There now is a sufficiently general stability proof based on an energy argument, and in one dimension we also know the precise string of penalty terms and coefficients that generate RDG for any order of the DG basis. Users can choose between various implementations in the standard, once-partially-integrated form RDG-1x, and the twice-partially-integrated form RDG-2x, which is more accurate and stable. The latter scheme can again be rewritten in the standard form, with a small but significant extra boundary term remaining. When implemented on a Cartesian grid, RDG-2x is the most accurate and most stable among all known DG diffusion methods, for a given order p of the basis (Huynh, 2009). Nevertheless, for a system with nonlinear diffusion terms like the Navier-Stokes equations, RDG-1x has the advantage of being somewhat easier to implement than RDG-2x.

Besides interface-based recovery (iRDG) for DG diffusion fluxes, there is now also cell-centered recovery (cRDG), developed by Nourgaliev et al. to enhance the DG basis; it is still restricted to Cartesian grids. This technique increases the accuracy of upwind advection fluxes and may also be used as a preprocessor for iRDG.

One may regard cRDG as a cross between DG and the Finite-Volume method. We expect to see much more of this, once a robust formulation for unstructured grids has been established.

Altogether, RDG appears to have a bright future.

8 Acknowledgement

This work was supported by AFOSR Grant Nr. FA9550-08-1-0307.

References

[1] B. van Leer and S. Nomura, "Discontinuous Galerkin for diffusion," AIAA Paper 2005-5108, 2005.

[2] B. van Leer, M. Lo, and M. van Raalte, "A Discontinuous Galerkin method for diffusion based on recovery," AIAA Paper 2007-4083, 2007.

[3] H. T. Huynh, "A reconstruction approach to high-order schemes including Discontinuous Galerkin for diffusion," AIAA Paper 2009-0403, 2009.

[4] R. Nourgaliev, T. Theofanous, H. Park, V. Mousseau, and D. Knoll, "Direct numerical simulation of interfacial flows," AIAA Paper AIAA-2008-1453, 2008.

[5] H. Park, R. Nourgaliev, V. Mousseau, and D. Knoll, "Recovery Discontinuous Galerkin Jacobian-free Newton Krylov method for all-speed flows," Tech. Rep. INL/CON-08-13822, Idaho National Laboratory, 2008.

[6] K. W. Morton and M. A. Rudgyard, "Shock recovery and the cell-vertex scheme for the steady Euler equations," *Lecture Notes in Physics*, vol. 323, pp. 424–428, 1989.

[7] F. Bassi and S. Rebay, "High-order accurate discontinuous finite element solution of the 2D Euler equations," *Journal of Computational Physics*, vol. 138, pp. 251–285, 1997.

[8] J. Peraire and P. O. Persson, "The compact discontinuous Galerkin method for elliptic problems," *SIAM Journal on Scientific Computing*, vol. 30, pp. 1806–1824, 2008.

[9] B. van Leer and M. Lo, "Unification of Discontinuous Galerkin methods for advection and diffusion," AIAA Paper 2009-0400, 2009.

[10] G. A. Baker, "Finite element methods for elliptic equations using nonconforming elements," *Mathematics of Computation*, vol. 31, pp. 45–59, 1977.

[11] D. N. Arnold, "An interior penalty finite element method with discontinuous elements," *SIAM Journal on Numerical Analysis*, vol. 19, pp. 742–760, 1982.

[12] M. H. van Raalte, *Multigrid Analysis and Embedded Boundary Conditions for Discontinuous Galerkin Discretization.* PhD thesis, University of Amsterdam, 2004.

[13] M. van Raalte and B. van Leer, "Bilinear forms for the recovery-based discontinuous Galerkin method for diffusion," *Communications in Computational Physics*, vol. 5, pp. 683–693, 2008.