47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition
5 - 8 January 2009, Orlando, Florida

AIAA 2009-400

Unification of Discontinuous Galerkin Methods for Advection and Diffusion[1]

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

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

Abstract

The Recovery-based Discontinuous Galerkin discretization for diffusion terms is shown to combine well with the upwind Discontinuous Galerkin discretization for advection terms. The difference in treatment between the two kinds of terms is reduced to a switch from the discontinuous basis to a smooth basis of that is indistinguishable in the weak sense; the two bases are orthonormal to each other.

We address some implementation issues, in particular, how to avoid bad conditioning of the recovery equations.

There now are two versions of RDG, based on different formulations of the DG weak formulation. Integrating by parts twice lewads to RDG-2x, the original version; integrating by parts only once leads to RDG-1x. In numerical experiments on regular and irregular triangular grids both methods appear to be of the order of accuracy $p + 1$, where p is the order of the elemental polynomial basis. There is some evidence that RDG-1x has a slightly lower truncation error than RDG-2x; this possible benefit is offset by the disadvantage of RDG-1x having a lower stability limit than RDG-2x.

1   Introduction: history of RDG

At the 17th AIAA Computational Fluid Dynamics Conference, Toronto, 2005, we introduced the "recovery method" for computing the diffusive fluxes in a Discontinuous Galerkin (DG) scheme [1]. In this approach, which we shall refer to as RDG (R for Recovery), the fluxes are based on a smooth, locally recovered solution that in the weak sense is indistinguishable from the discontinuous discrete solution. This eliminates the introduction of *ad hoc* penalty or coupling terms found in traditional DG methods; instead, recovery automatically generates a string of such terms, each with its optimal coefficient [1, 2]. Likewise, it eliminates the *ad hoc* choice of a unique solution value at an interface discontinuity (e. g., the average or a one-sided value) introduced in newer methods (see Huynh [3]).

---

[1]©2009 by Bram van Leer and Marcus Lo. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

[2]Professor, Fellow AIAA

[3]Doctoral candidate, member AIAA

Analysis and applications of the recovery method in 2005 were restricted to the case of one space dimension.

A noteworthy feature of RDG was that it started from the weak form resulting from partially integrating the diffusion equation *twice*; the standard DG form for diffusion follows from partially integrating only once. The repeated partial integration appeared to contribute to the high order of accuracy achieved by RDG on a Cartesian grid. For example, if the numerical solution 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 as seen in the convergence of the cell-averaged solution values, is $3p + 2$ or $3p + 1$ for p even or odd, respectively[4] [3].

At the time of the next AIAA CFD Conference, Miami, 2007, RDG had developed into a full-blown multidimnensional mthod. In our conference paper[4] we offered a two-dimensional formulation on an unstructured grid; by then the non-trivial 3-D extension was also known.

A key theoretical result presented was the existence of the "recovery basis," a unique basis orthonormal to the discontinuous solution basis; the "Principal Recovery Theorem" says that the expansions of the discontinuous and recovered solutions in terms of their respective bases are *identical*. This is of fundamental importance in developing a unified DG method for equations that contain both advection and diffusion terms – which is the main theme of the current paper. The recovery basis and associated theorem were treated in detail in a subsequent journal paper[2].

Further developments in the analysis and implementation of RDG are summarized in a paper submitted to the 19th AIAA CFD Conference, San Antonio, TX, June 2009 [5], and include the derivation of the general penalty-term form of 1-D RDG, an *Ansatz* to a general stability proof, and the development of a new family of RDG schemes based on the standard, once-partially-integrated DG form.

The latest research subject is reduced-order recovery, which makes sense for reasons of computational efficiency. Here, the recovered solution on the union of two elements has fewer than twice the number of parameters of a single element. Standard recovery would lead to an overdetermined linear system; how to suppress the least significant information while maintaining a well-conditioned and stable method is still under investigation.

Altogether, 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 of high accuracy.

In the remainder of this paper we first review the definition and chief properties of RDG (Section 2); this then serves as the basis of a unified theory of DG discretization for advection and diffusion (Section 3). Next follows Section 4 on implementation issues; this clears the way for Section 5, which contains numerical results regarding the accuracy of RDG on regular and irregular triangular grids. 'Section 6 lists the main conclusions from the material presented.

---

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

2 Definition and properties of RDG

RDG starts from a weak form of the diffusion equation obtained after integration by parts *twice* over the interior of an element Ω_j with boundary $\partial\Omega_j$:

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

Here U is the true solution, D is a constant diffusion coefficent and the $(v_k)_j$ 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 $\{v_k\}_j$. This numerical solution is discontinuous at $\partial\Omega_j$; if the boundary integral in (1) is evaluated on the inside 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 defined by making it indistinguishable from u in the weak sense in the contributing elements, that is,

$$
\int\int\int_{\Omega_j} (v_k)_j f_{j,j+1} d\Omega \;=\; \int\int\int_{\Omega_j} (v_k)_j u\; d\Omega, \quad k = 1,...,K,
\tag{2}
$$

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

In order to satisfy these equations, f must lie in the space of the smooth basis functions $\{w_k\}_{j,j+1}$, $k = 1,..,2K$, defined on $\Omega_j \bigcup \Omega_{j+1}$; for a detailed description of this space see [4].

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:

$$
\begin{aligned}
\int\int\int_{\Omega_j} (v_k)_j u_t d\Omega \;&=\; D\int\int_{\partial\Omega} \{(v_k)_j \nabla f_{j,..} - f_{j,..}\nabla(v_k)_j\} \cdot \hat{n}\; d\partial\Omega \\
&+\; D\int\int\int_{\Omega_j} u\nabla\cdot\nabla(v_k)_j d\Omega, \quad k = 1,...,K.
\end{aligned}
\tag{4}
$$

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

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

all DG diffusion schemes other than RDG are based on this form.

3 Unifying DG for advection and diffusion

3.1 Recovery basis and Principal Recovery Theorem

For the benefit of this subsection and the next one we make the notation more compact by merging the bases for elements Ω_j and Ω_{j+1}, each including K functions, into one basis of $2K$ functions for use on $\Omega_j \bigcup \Omega_{j+1}$:

$$(v_k)_{j,j+1} = (v_k)_j, \quad k = 1, .., K; \tag{6}$$

$$(v_k)_{j,j+1} = (v_{k-K})_{j+1}, \quad k = K+1, .., 2K; \tag{7}$$

We may now expand the discontinuous solution and the recovered solution similarly in terms of their respective bases on $\Omega_j \bigcup \Omega_{j+1}$:

$$u(x,y) = \sum_{k=1}^{2K} (a_k)_{j,j+1} (v_k(x,y))_{j,j+1}, \tag{8}$$

$$(f(x,y))_{j,j+1} = \sum_{k=1}^{2K} (b_k)_{j,j+1} (w_k(x,y))_{j,j+1}, \tag{9}$$

$$(x,y) \in \Omega_j \bigcup \Omega_{j+1}. \tag{10}$$

Inserting these expansions into Eqs. (2,3) yields a linear system of the form

$$\mathbf{Rb} = \mathbf{Ma}, \tag{11}$$

where the column vectors $\mathbf{a}$ and $\mathbf{b}$ contain the coefficients $(a_k)_{j,j+1}$ and $(b_k)_{j,j+1}$, $k = 1, .., 2K$, respectively. The $2K \times 2K$ "element mass matrix" $\mathbf{M}$ contains the inner products of the basis functions $(v_k)_{j,j+1}$ with each other, while the $2K \times 2K$ "recovery mass matrix" $\mathbf{R}$ contains the inner products of the basis functions $(w_k)_{j,j+1}$ with the basis functions $(v_l)_{j,j+1}$. (For detailed formulas see [4].)

We may rewrite the above equation as

$$\mathbf{b} = \mathbf{R}^{-1}\mathbf{Ma} \equiv \mathbf{Qa}; \tag{12}$$

where $\mathbf{Q}$ is the "recovery operator." Note that $\mathbf{M}$, $\mathbf{R}$ and $\mathbf{Q}$ can be pre-computed and stored for use during an entire numerical simulation on a fixed grid.

Now consider the columns of the matrix $\mathbf{Q}$; let us call these $\mathbf{q}_l$, $l = 1, ..2K$. This set of column vectors spans the vector space in which $\mathbf{b}$ is found and may therefore serve as its basis. If we expand $\mathbf{b}$ in terms of this basis, Eqn. (12) tells us that the coefficients are simply the $(a_l)_{j,j+1}$:

$$\mathbf{b} = \sum_{l=1}^{2K} (a_l)_{j,j+1} \mathbf{q}_l. \tag{13}$$

Translated back into a statement about basis *functions* on $\Omega_j \bigcup \Omega_{j+1}$ this means that by switching to the "recovery basis"

$$(\tilde{w}_k)_{j,j+1} = \sum_{l=1}^{2K} q_{kl} (w_l)_{j,j+1}, \quad k = 1, ..2K, \tag{14}$$

4

f inherits the expansion coefficients of u:

$$(f(x,y))_{j,j+1} = \sum_{k=1}^{2K}(a_k)_{j,j+1}(\tilde{w}_k(x,y))_{j,j+1}, \quad (x,y) \in \Omega_j \bigcup \Omega_{j+1}. \tag{15}$$

We call this the Principal Recovery Theorem; for a more detailed its proof see [2].

By making one of the $(a_k)_{j,j+1}$ equal to one and all the other ones zero, it is seen that each of the recovery-basis functions is the recovered, smooth version of an original, discontinuous basis function on $\Omega_j \bigcup \Omega_{j+1}$. These pairs of basis functions are indistinguishable in the weak sense.

If we start with f expressed in the new basis according to (15), the linear system (11) changes into

$$\tilde{\mathbf{R}}\mathbf{a} = \mathbf{M}\mathbf{a}, \tag{16}$$

where $\tilde{\mathbf{R}}$ contains the inner products of the basis functions $(\tilde{w}_k)_{j,j+1}$ with the basis functions $(v_l)_{j,j+1}$. If the basis $\{v_k\}$ is orthonormalized, $\mathbf{M}$ becomes the $2K \times 2K$ identity matrix, and $\mathbf{R}$, therefore, too. This means that the bases $\{\tilde{w}_k\}$ and $\{v_l\}$ are orthonormal to each other.

3.2   Implications for advection and diffusion

The Principal Recovery Theorem allows us to unify the treatment of advection and diffusion terms in DG methods; specifically, it directs to the upwind/recovery-based (URDG) method. Here's the bottom line.

When including advective and diffusive fluxes in a Discontinuous Galerkin discretization, one must use distinct solution representations. This is perfectly fine as long as these representations are indistinguishable in the weak sense.

For the advective fluxes one may express the solution in terms of the original discontinuous basis functions, as long as a Riemann solver is used to physically resolve the discontinuities.

For the diffusive fluxes one must express the solution in terms of smooth basis functions that in the weak sense are indistinguishable from the original discontinuous basis functions. The process of constructing these basis functions is called recovery.

According to the Principal Recovery Theorem, the expansions of the recovered smooth solution and the original discontinuous solution in terms of their respective bases are identical, i. e., they have equal coefficients. Thus, the recovered and discontinuous solution representations are indeed indistinguishable in the weak sense.

Just as the discontinuous basis is unsuited for computing diffusive fluxes, the recovery basis is unsuited for computing advective fluxes: it leads to (unstable) central differencing. *In URDG the distinction between treating advection and diffusion is reduced to the essential minimum: switching between bases.*

4 Implementation issues

4.1 RDG schemes in standard form

It is possible to insert the information obtained in the recovery procedure into the standard DG form (5); the schemes generated this way are different from those of the form (4). We shall denote these schemes by RDG-1x and RDG-2x, respectively, referring to how many times integration by parts is performed.

In both RDG versions we use f at the boundary to achieve coupling between elements; in RDG-1x we must also use it to calculate the volume integral with the greatest possible accuracy[5]. This means we must replace ∇U in the interior of the element by a maximally accurate expression based on the various surrogates $f_{j,..}$ of U recovered from u.

A crucial observation is that the recovery procedure at the interface $\partial\Omega_{j,j+1}$ accurately recovers *only* the dependence of the solution on the *normal* coordinate $n_{j,j+1}$ (in the direction of $\hat{n}_{j,j+1}$), namely, with a polynomial of degree $2p+1$. Since there are more directions $\hat{n}_{j,j+1}$ than dimensions, recovery of the solution gradient involves a least-squares procedure. Let N_j be the matrix whose rows are the vectors $\hat{n}_{j,l}$, where index l cycles through the neighbors Ω_l of Ω_j, $l = l_1, ..l_L$, and let $g_j(\vec{x})$ be the vector of recovered normal derivatives $\partial f_{j,l}(\vec{x})/\partial n_{j,l}$, evaluated in a point $\vec{x} \in \Omega_j$; that is,

$$N_j = \begin{pmatrix} (n_{l_1})_x & (n_{l_1})_y & (n_{l_1})_z \\ : & : & : \\ (n_{l_L})_x & (n_{l_L})_y & (n_{l_L})_z \end{pmatrix}, \quad g_j(\vec{x}) = \begin{pmatrix} \frac{\partial f_{j,l_1}(\vec{x})}{\partial n_{j,l_1}} \\ : \\ \frac{\partial f_{j,l_L}(\vec{x})}{\partial n_{j,l_L}} \end{pmatrix}. \tag{17}$$

The recovered gradient $\nabla f_j(\vec{x})$ then follows from

$$N_j^T N_j \nabla f_j(\vec{x}) = N_j^T g_j(\vec{x}). \tag{18}$$

This system will have to be solved in all quadrature points used in evaluating the volume integral on the right-hand side of Eqn. (5).

In one dimension the above scheme reduces to

$$\int_{I_j} v_k u_t dx = D(v_k f_x)\big|_{j+\frac{1}{2}}^{j+\frac{1}{2}} - D \int_{I_j} (v_k)_x \overline{f}_x dx, \quad k = 1,..,p+1, \tag{19}$$

where

$$\overline{f}(x) = \frac{1}{2}\{f_{j,j+1}(x) + f_{j,j-1}(x)\}, \quad x \in I_j. \tag{20}$$

For $p = 1$ this recovery scheme can be written as a classical DG scheme of the form studied in [1],

$$\int_{I_j} v u_t dx = -D\left(<u_x>[v]\big|_{j+\frac{1}{2}} + <u_x>[v]\big|_{j-\frac{1}{2}}\right) - D \int_{I_j} v_x u_x dx$$

---

[5]In RDG-2x, replacing u by any of the $f_{j,..}$, or an average of these, in the volume integral does not change the value of the integral; see [4].

$$+ \quad \sigma D \left(<v_x> [u] \big|_{j+\frac{1}{2}} + <v_x> [u] \big|_{j-\frac{1}{2}} \right)$$

$$- \quad \frac{\mu D}{\Delta x} \left([v][u] \big|_{j+\frac{1}{2}} + [v][u] \big|_{j-\frac{1}{2}} \right), \tag{21}$$

with

$$\sigma = \frac{1}{4}, \quad \mu = \frac{9}{4}. \tag{22}$$

This scheme was derived in [1] by minimizing truncation error, without reference to recovery. It is gratifying that the scheme turns out to be a recovery scheme, after all. It was shown to be 4th-order accurate, with a truncation error comparable to that of the original recovery scheme for $p = 1$.

While conforming to the standard form may seem an advantage of RDG-1x, we have found that its stability condition is more restrictive than that of RDG-2x for all values of p tested. Section 5 includes a comparison of the two forms on regular triangular grids.

4.2   Recovery and condition number

Recovery requires the solution of linear systems of the form

$$\mathbf{Rb} = \mathbf{Ma}, \tag{23}$$

if only for the pre-computation of the recovery basis functions at all cell faces. This calls for the inversion of $\mathbf{R}$. The matrix $\mathbf{M}$ also appears on the left-hand side of the DG update equations and therefore needs inversion as well.

When the degree p of the elemental polynomial space is raised, the condition of the system quickly worsens; raising yjr numbrt of dimensions has a similar effect. When working on keeping the condition number of $\mathbf{R}$ as low as possible, we discovered a number of practical rules that need to be followed.

1. Normalize inner products. when computing inner products like those in Eqs. (2, 3), to obtain the elements of $\mathbf{M}$ and $\mathbf{R}$, always divide by the element volume V_Ω, e. g.,

$$(v \cdot w)_j \equiv \frac{1}{V_{\Omega_j}} \int \int \int_{\Omega_j} vw \; d\Omega. \tag{24}$$

2. *Make the elemental basis $\{v_k\}$ orthonormal.* The best results were obtained with analytical orthonormalization, starting from Legendre polynomials. For a triangle, e. g., one starts with Legendre polynomials on a square, then one side of the square is collapsed onto its midpoint. A further linear mapping (with constant Jacobian) will accomodate any other triangular shape.

3. *Try to make the provisional basis $\{w_k\}$, used as a stepping stone to obtain $\{\tilde{w}_k\}$, reasonably well-behaved.* Since inner products on two elements are involved this is not easy except on the most regular grids. *At least make sure to normalize the columns of $\mathbf{R}$ to unit length, with corresponding rescaling of the elements of $\mathbf{b}$.*

4. If Eqn. (23) follows from reduced-order revovery, meaning **b** is a shorter vector than **a** and **R** is rectangular, *do not pre-multiply by* $\mathbf{R}^T$ *for a least-squaressolution;* this almost squares the condition number.

5. On the other hand, do not worry too much about the condition of the least-squares system (18); it is independent of p and only weakly dependent on dimensionality,
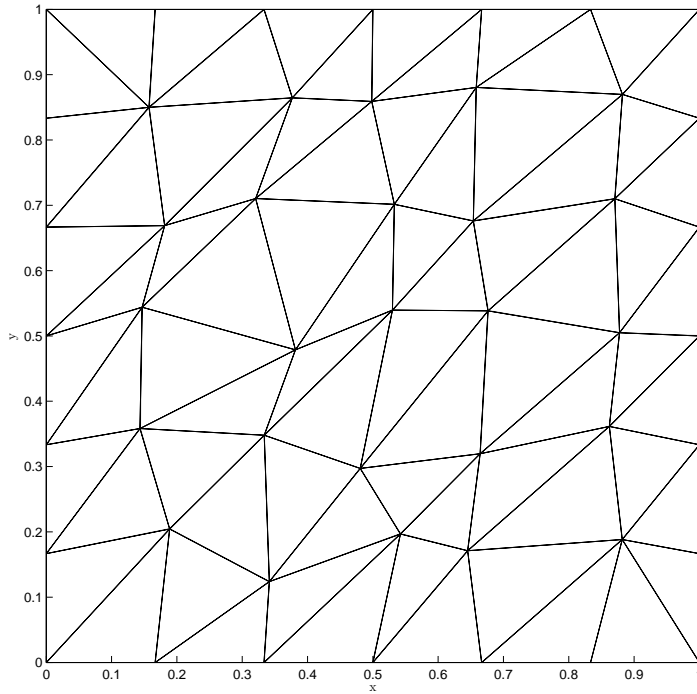
5   Numerical results

Figure 1: An irregular triangular grid (72 elements) obtained from a regular grid by randomly perturbing the nodes with a maximum displacement of 30% of the mesh width.

To illustrate the accuracy of RDG-2x we solve a Dirichlet problem on the square $[0, 1] \times [0, 1]$ with exact steady solution

$$U(x, y) = x^3 + 3x^2 y - 3xy^2 - y^3, \tag{25}$$

on a sequence of irregular grids. Each is derived by perturbing a regular triangular grid resulting from dividing square meshes by diagonals in the same direction[6]. The perturbations

---
[6]Except in two corner squares, where we wished to avoid triangles with two boundary faces.

of the nodal points are randomly distributed, with a maximum displacement in each coordinate direction of 15% or 30% of the average face length. An example of a 30%-perturbed grid is given in Figure 1.

Figure 2 shows the results of a convergence study on the 15%-perturbed grids, for $p = 1$, 2 and 3. The order of accuracy of RDG-2x is seen to be $p + 1$.
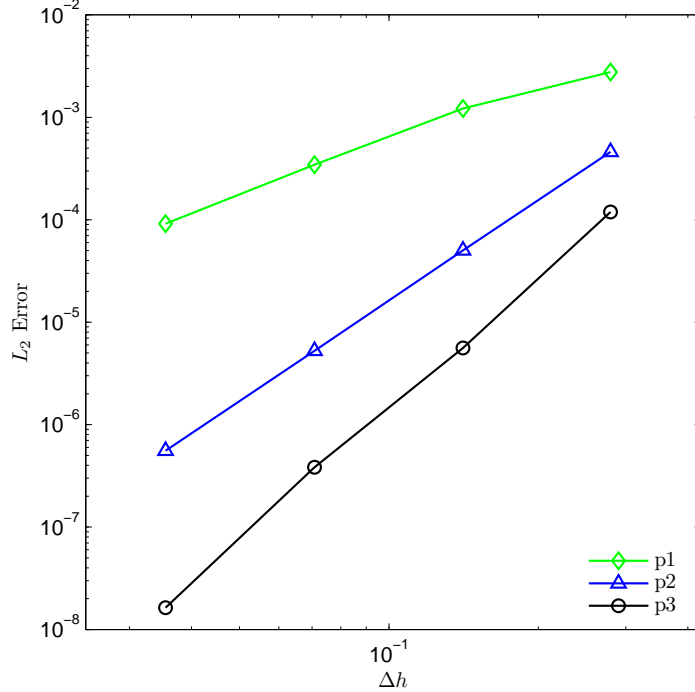

Figure 2: L_2-error convergence for RDG-2x on a sequence of 15%-perturbed grids, for $p = 1$, 2, 3.

Figure 3 shows, for $p = 2$, how the amplitude of the grid perturbation affects the accuracy of RDG-2x. As expected, the solution error increases with the grid perturbation, although on the coursest grid there is a surprising reversal of order between the 15%- and 30%-perturbed grids[7]. On the finer grids the order of accuracy is 3 regardless of the amplitude of the perturbations.

Next, we solve a different Dirichlet problem on $[0, 1] \times [0, 1]$, taken from[6]; its steady solution is

$$u(x, y) = \exp^{-0.1 sin(-5.1x + 6.2y) + 0.3 cos(4.3x + 3.4y)}, \tag{26}$$

The triangular grid used is orthogonal in the sense that the connector between the centroids of two adjacent triangles is perpendicular to their interface; see Figure 4.

---

[7]This phenomenon persisted when averaging over a few runs on different 15%-perturbed grids.
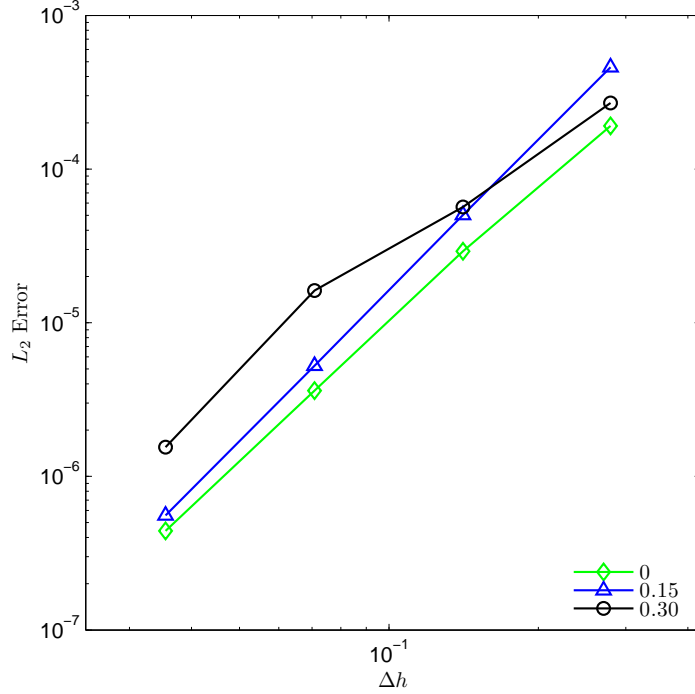
Figure 3: Dependence of l_2-error on grid irregularity for RDG-2x, with $p = 2$.

Finally, Figure 5 combines the error-convergence results of RDG-2x and RDG-1x, for $p = 1, 2, 3$. There are indications that RDG-1x is more accurate than RDG-2x, particularly on finer grids. A disadvantage of RDG-1x, though, is that its stability condition is more restrictive than that of RDG-2x. For $p = 0$ the allowable time step is 21% smaller; for $p = 3$ the difference has grown to 36%, and we expect it to grow further with increasing p.

It is worth mentioning that the error levels shown in this figure are an order of magnitude lower than the corresponding ones presented in [6] for the Compact DG (CDG) scheme, and all other schemes tested.

6 Conclusions

The Recovery-based Discontinuous Galerkin discretization for diffusion terms combines well with the upwind Discontinuous Galerkin discretization for advection terms. The difference in treatment between the two kinds of terms can be reduced to a switch from the discontinuous basis to a smooth basis that is indistinguishable in the weak sense; the two bases are orthonormal to each other.

There now are two versions of RDG, based on different formulations of the DG weak
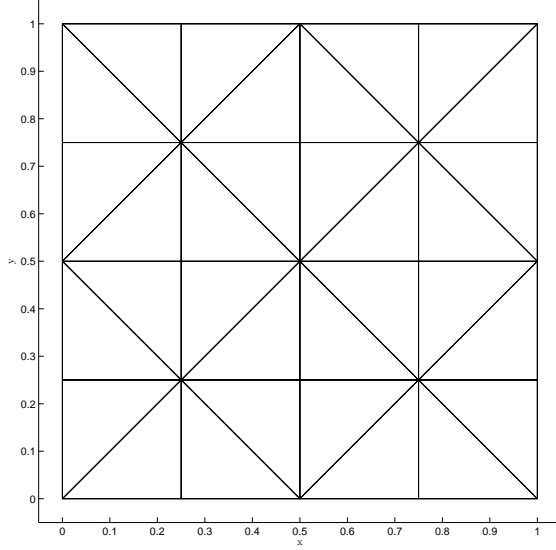
Figure 4: Example of orthogonal triangular grid (32 meshes) used for some of the numerical tests.

formulation. Integrating by parts twice leads to RDG-2x, the original version; integrating by parts only once leads to RDG-1x. There is some evidence that RDG-1x has a lower truncation error than RDG-2x; this possible benefit is offset by the disadvantage of having a lower stability limit than RDG-2x.

RDG is straightforward to apply, owing to the simplicity and generality of the recovery principle, and for a given value of p it appears to be more accurate than any other DG diffusion scheme currently in use.

References

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

[2] 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.

[3] H. T. Huynh, "A reconstruction approach to high-order schemes including Discontinuous Galerkin for diffusion," 2009. to be presented at the 47th AIAA Aeropace Sciences Meeting, 5-8 January 2009, Orlando, FL.
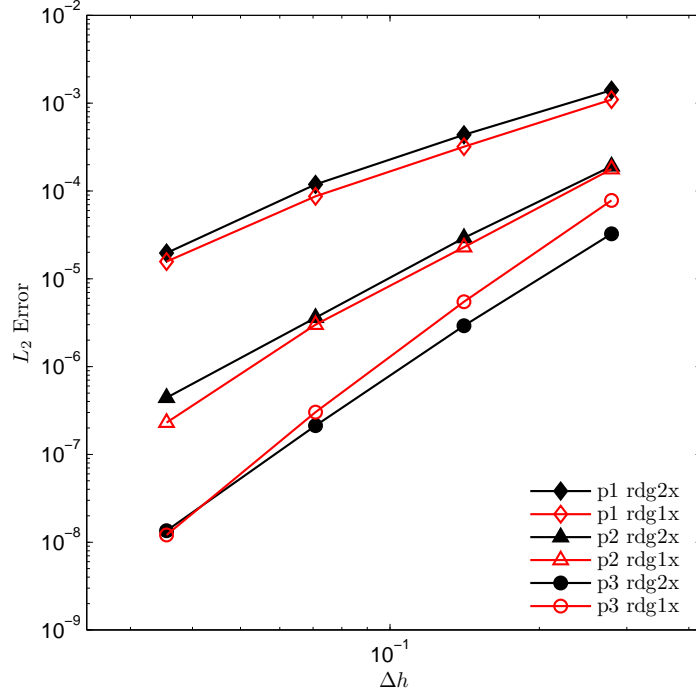
Figure 5: *Comparison of L_2*-error convergence for RDG-2x and RDG-1x on a sequence of orthogonal grids, for $p = 1, 2, 3$.

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

[5] B. van Leer and M. Lo, "Analysis and implementation of the Recovery-based Discontinuous Galerkin method for diffusion," aiaa paper, 2008. submitted to the 19th AIAA Computational Fluid Dynamics Conference, 22-25 June 2009, San Antonio, TX.

[6] 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.