18th AIAA Computational Fluid Dynamics Conference
25 - 28 June 2007, Miami, FL

AIAA 2007-4083

A Discontinuous Galerkin Method for Diffusion Based on Recovery

Bram van Leer* and Marcus Lo†
University of Michigan, Ann Arbor, MI 48105-2140, USA

Marc van Raalte‡
Centrum voor Wiskunde and Informatica, Kruislaan 413, 1090 GB Amsterdam, The Netherlands

We present the details of the recovery-based DG method for 2-D diffusion problems on unstructured grids. In the recovery approach the diffusive 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. Crucial is the choice of the proper basis for recovery of the smooth solution on the union of two elements. Some results on accuracy, stability and the range of eigenvalues are given, together with numerical solutions on rectangular grids.

I. Introduction

At the 17th AIAA Computational Fluid Dynamics Conference, Toronto, 2005, we presented the "recovery method" for computing the diffusive fluxes in a Discontinuous Galerkin (DG) scheme.[1] In this approach 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; recovery automatically generates a string of such terms, each with its optimal coefficient.[1–3] Analysis and applications of the recovery method were restricted to the case of one space dimension. It was found that the order of accuracy increased under p-refinement as 2^{p+1}, hence exponentially, and that steady solutions became exact in all but the highest two derivatives.

In the present paper the recovery principle is used to develop a full-blown two-dimensional, basis-independent DG method for diffusion. In the part devoted to analysis we present evidence of stability and accuracy in special cases; a general stability proof and a general accuracy estimate are still missing. However, the natural simplicity and the observed accuracy and robustness of recovery-based methods lead us to believe these have the potential to replace the traditional DG diffusion methods.

*Professor, Department of Aerospace Engineering, AIAA Fellow.

†Doctoral Candidate, Department of Aerospace Engineering, AIAA Student Member

‡Postdoctoral Researcher, Centrum voor Wiskunde and Informatica, Amsterdam, Netherlands.

II. Integrating the diffusion equation

We shall develop the DG recovery method for the 2-D diffusion equation

$$U_t = D(U_{xx} + U_{yy}) + s(x, y), \tag{1}$$

where D is the positive diffusion coefficient, assumed to be constant in space and time, and $s(x, y)$ represents a source. To obtain the update formulas for element Ω_j (triangle or quadrilateral) we begin by multiplying Eqn. (1) with a sequence of test functions $(v_k)_j,\ k = 1, ..., K$, that are polynomial on Ω_j and zero elsewhere. In a Galerkin method these also serve as the basis for the element-wise continuous approximate solution u. We then integrate the equation over the interior of the element and integrate by parts *twice*, yielding

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

The second integration by parts, an obvious choice for a second-order PDE, is *not* standard in DG methods, but is crucial to achieving optimal accuracy; we shall explain this toward the end of this section. For the sake of comparison we also give the formula resulting after integrating by parts once:

$$\begin{aligned}\int\int_{\Omega_j} (v_k)_j u_t d\Omega &= D\int_{\partial\Omega_j} (v_k)_j \hat{\mathbf{n}} \cdot \nabla u\ d\partial\Omega \\ &- D\int\int_{\Omega_j} \nabla u \cdot \nabla (v_k)_j d\Omega + \int\int_{\Omega_j} (v_k)_j s\ d\Omega, \quad k = 1, ..., K. \end{aligned} \tag{3}$$

Eqn. (2) is not usable as a DG scheme: the boundary integral is not defined because the solution is discontinuous at the element boundary. If we assume the boundary integral to be evaluated on the inner side of the boundary, the element will not be coupled to its neighbors, rendering the space-time scheme inconsistent. In Van Leer and Nomura[1] we demonstrated that coupling to a neighboring element Ω_{j+1} can be achieved naturally by *recovering* from the discontinuous discrete solution a smooth solution on the union of the element and its neighbor. The recovered solution is expressed in terms of basis functions $(w_k)_{j,j+1}$, $k = 1, ..., 2K$, that are polynomial on the union of the two elements and zero elsewhere; this basis is twice as large as the basis used within a single element, as it must accommodate information from two elements. By demanding that the recovered solution "tests out the same" (see Eqs. (5, 6) below) as the original solution on both elements, we make the recovered solution, in the weak sense, indistinguishable from the original solution. Figure 1 illustrates the recovery procedure in one dimension.

Using the recovered solution, denoted by f, to calculate the boundary integral, we achieve that Eqn. (2) becomes a valid DG scheme:

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

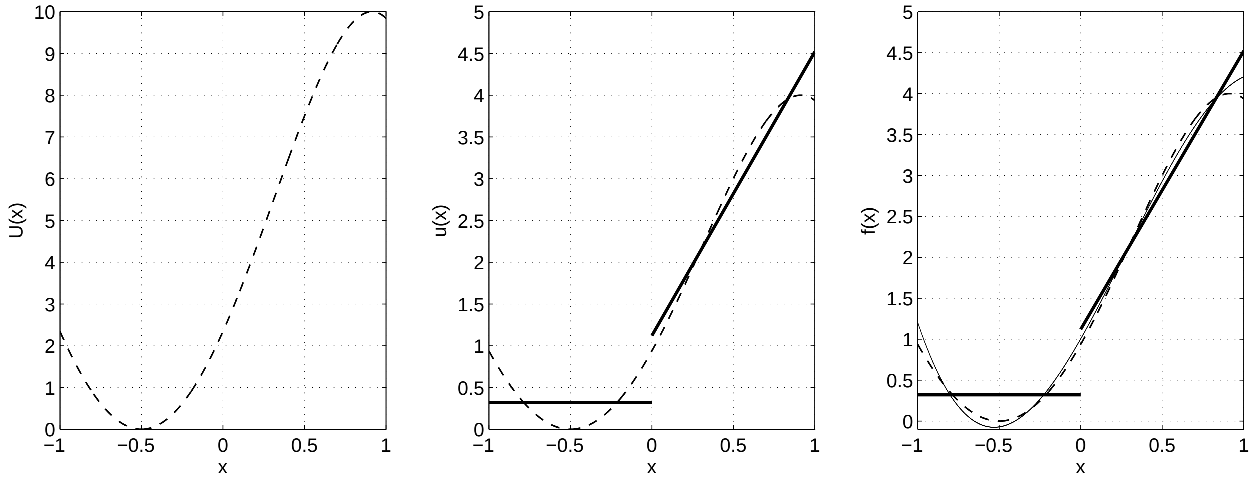

Figure 1. Recovery in one dimension, K=2. Shown are, from left to right, the original quartic initial values U (dashed), the piecewise linear discretization u (bold) together with U, and the cubic recovered distribution f (thin) together with u and U, on the adjacent intervals $(-1,0)$ and $(0,1)$. All three distributions yield the same value when taking their inner product with either test function on either interval, making them indistinguishable in the weak sense.

We must emphasize here that the values of f and its normal derivative at the element boundary follow from 3 or 4 separate recovery procedures, each centered on a different element face. Following are the formulas for recovering the smooth solution $f_{j,j+1}$ on the union of neighboring elements Ω_j and Ω_{j+1}:

$$\int\int_{\Omega_j} (v_k)_j f_{j,j+1} d\Omega \quad = \quad \int\int_{\Omega_j} (v_k)_j u \; d\Omega, \quad k = 1, ..., K, \tag{5}$$

$$\int\int_{\Omega_{j+1}} (v_k)_{j+1} f_{j,j+1} d\Omega \quad = \quad \int\int_{\Omega_{j+1}} (v_k)_{j+1} u \; d\Omega, \quad k = 1, ..., K. \tag{6}$$

Expanding u in terms of the basis functions v_k on both elements,

$$u(x,y) \quad = \quad \sum_{k=1}^{K} (a_k)_j (v_k(x,y))_j, \quad (x,y) \in \Omega_j, \tag{7}$$

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

and $f_{j,j+1}$ in terms of the basis functions w_k,

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

and inserting these into Eqs. (5,6) yields the following linear system for the coefficients defining f:

$$\mathbf{R}_{j,j+1} \mathbf{b}_{j.j+1} \quad = \quad \mathbf{M}_{j,j+1} \mathbf{a}_{j.j+1}, \tag{10}$$

$$\mathbf{a}_{j,j+1} \quad = \quad ((a_1)_j, ..., ((a_K)_j, (a_1)_{j+1}, ..., (a_K)_{j+1})^T, \tag{11}$$

$$\mathbf{b}_{j,j+1} \quad = \quad ((b_1)_{j,j+1}, ..., ((b_{2K})_{j,j+1})^T, \tag{12}$$

$$(M_{kl})_{j,j+1} = \begin{cases} \int\int_{\Omega_j} (v_k)_j (v_l)_j d\Omega, & k=1,...,K, \quad l=1,...,K, \\ 0, & k=1,...,K, \quad l=K+1,...,2K, \\ 0, & k=K+1,...,2K, \quad l=1,...,K, \\ \int\int_{\Omega_{j+1}} (v_{k-K})_{j+1} (v_{l-K})_{j+1} d\Omega, & k=K+1,...,2K, \quad l=K+1,...,2K, \end{cases} \tag{13}$$

$$(R_{kl})_{j,j+1} = \begin{cases} \int\int_{\Omega_j} (v_k)_j (w_l)_{j,j+1} d\Omega, & k=1,...,K, \quad l=1,...,2K, \\ \int\int_{\Omega_{j+1}} (v_{k-K})_{j+1} (w_l)_{j,j+1} d\Omega, & k=K+1,...,2K, \quad l=1,...,2K. \end{cases} \tag{14}$$

Thus, at each element face we find f from

$$\mathbf{b} = \mathbf{R}^{-1}\mathbf{M}\mathbf{a} = \mathbf{Q}\mathbf{a}; \tag{15}$$

the element mass matrix $\mathbf{M}$, the recovery mass matrix $\mathbf{R}$ and the recovery operator $\mathbf{Q}$ can be computed and stored at the start of a computation.

How to construct the basis $\{w_k\}_{j,j+1}$ defined on $\Omega_j \bigcup \Omega_{j+1}$ is not trivial and will be addressed in the next section.

At this point it is fitting to clarify our earlier choice of integrating by parts twice rather than once. The extra integration has two effects. To begin with, it doubles the number of boundary terms, a liability for traditional DG methods but of no concern when recovery is used. On the contrary, boundary terms obtained from the recovered solution are an asset: they have superior accuracy, being based on $2K$ rather than K basis functions.

In addition, the second integration changes the interior integral on the right-hand side of the scheme. The interior integral in Eqn. (2) is preferable over the one in Eqn. (3), because the numerical solution u itself, rather than its gradient, appears. The weight $\nabla \cdot \nabla v_k$, can be expressed in terms of the $\{v_k\}$, thus the integral is a linear combination of the integrals appearing on the right-hand side of the recovery equations (5,6)). In consequence, it makes no difference whether u or any one of the recovered solutions is used to evaluate the interior integral. This is not true for the interior integral in Eqn. (3); obtaining ∇u by differentiating the discrete solution causes a loss of accuracy[a].

Finally, we have found that the use of Eqn. (2) leads to an almost exact solution to simple steady-state problems; this will be discussed further in Sec. VI.

It is worth observing that there is a superficial similarity between our recovery procedure and Brezzi's[4] lifting operation. In both procedures a function is generated on neighboring elements using information from the discontinuous solution. In Brezzi's procedure, the only information used is the solution jump at the interface; in our recovery procedure *all* data defining the solution on adjacent elements are used. Specifically, the jumps and averages of higher derivatives contribute to f. Brezzi's operator vanishes if the solution does not jump, just as does its predecessor, Arnold's[5] interior penalty term. Brezzi constructed the operator in order to prove stability of the scheme in the DG norm; the recovery procedure is motivated by accuracy as well as stability. We hope that stability in the DG norm will eventually be proven.

[a] The loss could be prevented by replacing ∇u with $\nabla \bar{f}$, where $\bar{f}$ is an average of the recovered solutions used for the element under consideration. But avoiding the integration of ∇u is far more practical.

III. The polynomial basis for recovery

In our paper on 1-D recovery we used Legendre polynomials for the basis $\{v_k\}$ in each interval, because of their orthogonality, and non-orthogonal Taylor polynomials for $\{w_k\}$ on the union of two intervals, because orthogonality relations on the double interval do not come into play. Obviously, any *bona fide* basis will do on either domain.

From the point of view of numerical analysis the ideal basis for the double interval results from applying the recovery procedure to all $2K$ original basis functions used on the separate intervals; we call this the *recovery basis.* If the original basis is orthonormal on each interval, and the recovery basis is made orthonormal with respect to the original basis, then the expansion of the smooth recovered solution in terms of the smooth recovery basis is identical to the expansion of the discontinuous solution in terms of the discontinuous basis: their coefficients are identical. For details the reader is referred to Van Raalte and Van Leer.[2,3] Whether the use of the recovery basis leads to computational savings is not yet clear, but it greatly simplifies analysis. We used it to our advantage when deriving the expansion in bilinear terms ("penalty terms") equivalent to a recovery-based method.[2,3]

In two dimensions a new phenomenon enters: directionality. This makes the choice of the recovery basis nontrivial; a naive choice may fail. For example, assume that in each element a linear basis is used; in element Ω_j we might choose

$$\{1, \;\; x - x_j, \;\; y - y_j\}, \tag{16}$$

where (x_j, y_j) is, say, the center of mass of the element. The basis for $f_{j,j+1}$, defined on $\Omega_j \bigcup \Omega_{j+1}$, will consist of 6 polynomials; an obvious choice would be the full quadratic basis

$$\{1, \; x - x_{j,j+1}, \; y - y_{j,j+1}, \; (x - x_{j,j+1})^2, \; (x - x_{j,j+1})(y - y_{j,j+1}), \; y - y_{j,j+1})^2\}, \tag{17}$$

where $(x_{j,j+1}, y_{j,j+1})$ is, say, the midpoint of the interface. This choice, it turns out, makes the system (5,6) singular.

To understand this, consider first obtaining f by *strong* interpolation, *viz.*, by assuming f, f_x and f_y are given in both (x_j, y_j) and (x_{j+1}, y_{j+1}); see Figure 2. In this case it quickly becomes clear there is an embedded 1-D interpolation problem requiring a *cubic* basis. Introduce a rotated coordinate system (ξ, η) with the ξ-axis through (x_j, y_j) and (x_{j+1}, y_{j+1}); from f_x and f_y we obtain f_ξ and f_η in both interpolation points. To interpolate f along the ξ-axis we already need 4 basis functions, e. g.,

$$1, \; \xi, \; \xi^2, \; \xi^3; \tag{18}$$

this leaves two basis functions, e. g.,

$$\eta, \; \xi\eta, \tag{19}$$

to be used for matching f_η in the two points.

With *weak* interpolation as used in the recovery procedure (5,6), the ξ-axis must be chosen normal to the interface and the η-axis along it; see Figure 2. Otherwise, the argument goes the same[b].

[b] Assume the solution on both elements is constant in the η-direction, then the 1-D recovery procedure of Van Leer and Nomura[1] reappears.

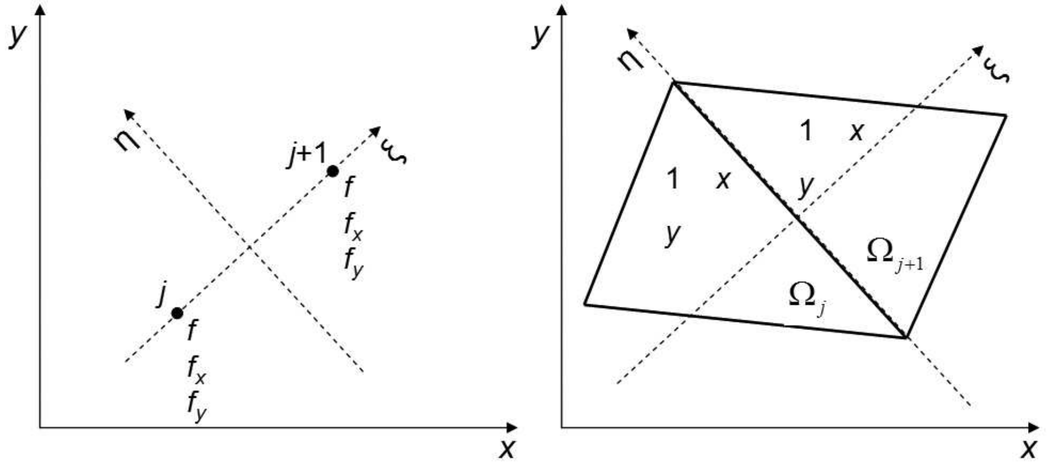

Figure 2. Strong and weak interpolation. (a) Strong interpolation using data in two points; (b) weak interpolation using inner products with test functions on two elements. In both cases the basis for the interpolant is best expressed in a rotated coordinate system (ξ, η).

If the basis $\{v_k\}$ is complete up to order p, requiring $K = \frac{1}{2}(p+1)(p+2)$, the recovery basis $\{w_k\}$ will contain $2K = (p+1)(p+2)$ basis functions. The embedded 1-D problem requires a basis of order $2p+1$, or $2p+2$ basis functions, in terms of ξ alone. In the remaining $(p-1)(p+2)$ basis functions, powers of η occur; for every factor η the maximum allowed power of ξ drops by 2. An example with $p = 3,\ K = 10$ is given in Table 1. That the accuracy of the recovery is greatest in the ξ-direction is a desirable property, since computing the diffusive flux requires differentiation of f in that direction.

1	x	x^2	x^3
y	yx	yx^2	$[yx^3]$
y^2	y^2x	$[y^2x^2]$	$[y^2x^3]$
y^3	$[y^3x]$	$[y^3x^2]$	$[y^3x^3]$

1	ξ	ξ^2	ξ^3	ξ^4	ξ^5	ξ^6	ξ^7
η	$\xi\eta$	$\xi^2\eta$	$\xi^3\eta$	$\xi^4\eta$	$\xi^5\eta$	$[\xi^6\eta]$	$[\xi^7\eta]$
η^2	$\xi\eta^2$	$\xi^2\eta^2$	$\xi^3\eta^2$	$[\xi^4\eta^2]$	$[\xi^5\eta^2]$	$[\xi^6\eta^2]$	$[\xi^7\eta^2]$
η^3	$\xi\eta^3$	$[\xi^2\eta^3]$	$[\xi^3\eta^3]$	$[\xi^4\eta^3]$	$[\xi^5\eta^3]$	$[\xi^6\eta^3]$	$[\xi^7\eta^3]$

Table 1. Example of solution and recovery bases for $p = 3$, $K = 10$. Left: Complete cubic basis for use in a single element (non-bracketed entries) and related tensor-product basis (all entries); x and y represent local Cartesian coordinates. Right: incomplete 7^{th}-degree basis used for the recovered solution on the union of two elements (non-bracketed entries) and tensor-product basis (all entries); ξ and η are coordinates normal to and aligned with the interface, respectively.

Once the basis functions have been determined in the rotated frame, they may, for other purposes, be expressed again in the original coordinate system.

We wish to point out that, on a grid of quadrilaterals, tensor-product bases may be used to represent the solution on each element as well as the recovered function on a pair of elements.

IV. Stability

In one dimension we have shown analytically for $p = 1$ that the eigenvalues of the DG operator approximating U_{xx} are nonpositive; we have checked it numerically up to $p = 5$. A general stability proof, based on classical finite-element arguments, is still being sought.

We note that 1-D stability implies 2-D stability if tensor-product bases are used, as can be done on quadrilateral meshes.

The range of the eigenvalues increases with the order p of the basis; a conservative estimate is:

$$\max_k |\lambda|_k \leq 4(p+1)^2. \tag{20}$$

This is comparable to the eigenvalue range found for most DG diffusion schemes. With regard to stability it shows that p-refinement, which introduces $p + 1$ independent solution parameters per mesh, is equivalent to dividing the mesh-size by $p + 1$.

In two dimensions we have numerically obtained the operator's eigenvalues for $p = 1$ on a grid of right triangles; these are nonnegative. Stencil and and spectrum are shown in Figure 3. So far, all our 2-D numerical experiments on rectangular grids have been stable for p up to 2 and a variety of boundary procedures. We do expect that a general stabilty proof of recovery-based schemes on an unstructured triangular mesh will eventually be found.

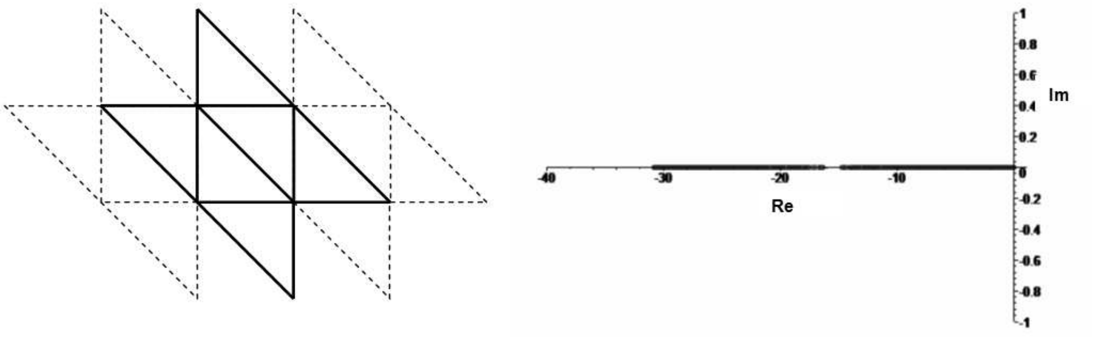

Figure 3. Stability on a triangular grid, $p = 1$. Left: stencil of right triangles used in the Fourier analysis of the recovery-based scheme. Right: eigenvalues in the complex plane.

V. Boundary procedures

When an element face lies on the domain boundary, the recovery procedure must be modified. Assume we use a complete basis of order p in the element. Along the boundary only η varies, so the number of basis functions needed to accomodate the boundary data is only $p+1$, making the total number of basis functions available for recovery $\frac{1}{2}(p+1)(p+4)$. Table 2 lists the basis functions for recovery at a boundary for $p = 3$, as in Table 1.

The loss of information in the direction normal to the boundary reduces the order of accuracy of the boundary scheme with respect to that of the interior scheme. This can be avoided rather easily on a

1
η
η^2
η^3

1	x	x^2	x^3
y	xy	x^2y	$[x^3y]$
y^2	xy^2	$[x^2y^2]$	$[x^3y^2]$
y^3	$[xy^3]$	$[x^2y^3]$	$[x^3y^3]$

1	ξ	ξ^2	ξ^3	ξ^4
η	$\xi\eta$	$\xi^2\eta$	$\xi^3\eta$	$[\xi^4\eta]$
η^2	$\xi\eta^2$	$\xi^2\eta^2$	$[\xi^3\eta^2]$	$[\xi^4\eta^2]$
η^3	$\xi\eta^3$	$[\xi^2\eta^3]$	$[\xi^3\eta^3]$	$[\xi^4\eta^3]$

Table 2. Example of boundary, element and recovery bases for $p = 3$. Left: Cubic basis needed for representing the boundary data; ξ and η are coordinates normal to and aligned with the boundary, respectively. Middle: Complete cubic basis for use in a single element (non-bracketed entries) and related tensor-product basis (all entries); x and y represent local Cartesian coordinates. Right: incomplete 4th-degree basis used for the recovered solution on the union of domain boundary and element (non-bracketed entries), and related tensor-product basis (all entries).

quadrilateral mesh by supplementing information from the next inward element; see Figure 4. When adding information from this element one must start with the lowest-order basis functions. Table 3 shows which functions must be contributed by the farthest element for $p = 3$.

1
η
η^2
η^3

1	x	x^2	x^3
y	xy	x^2y	
y^2	xy^2		
y^3			

1	ξ	ξ^2	
η	$\xi\eta$		
η^2			

Table 3. Example of boundary, element and recovery bases for $p = 3$. Left: Cubic basis needed for representing the boundary data; ξ and η are coordinates normal to and aligned with the boundary, respectively. Middle: Complete cubic basis for use in the element abutting the boundary; x and y represent local Cartesian coordinates. Right: Basis functions in the next cell used to make recovery at a domain boundary as accurate as at an interior boundary.

The 2-D results in Sec. VII on rectangular meshes were obtained with the boundary treatment of Figure 4.

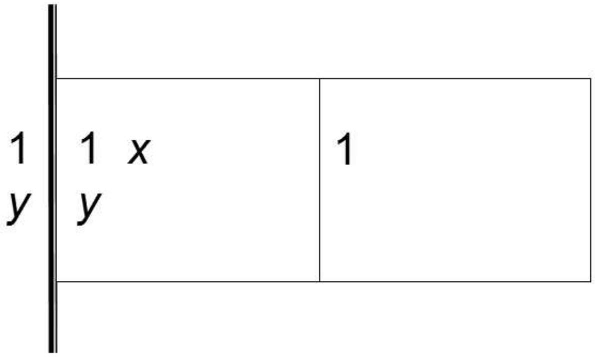

Figure 4. Recovery at the boundary, $p = 1$. Indicated are the basis functions used at the boundary, in the element abutting the boundary and the next element inward.

The same technique can be used on a triangular grid, as long as the adjacent elements providing the extra information extend sufficiently far inward. We have no experience yet with this implementation.

If no extra information is provided from inward cells, the boundary scheme may trail the interior scheme by one or more in order of accuracy. The error shows up undiluted in the L_∞ error norm, but is watered down in the L_2 norm, and even more in the L_1 norm; see Van Leer and Nomura.[1] Note that the accuracy at the boundary will still be greater than in traditional DG diffusion methods.

The above boundary procedure, whether involving one or more interior elements, creates some complex-conjugate eigenvalues; these may increase the spread of the eigenvalues, thus reducing the stability range.

VI. Accuracy

In one dimension, numerical results suggest the order of accuracy of recovery-based schemes is 2^{p+1}, that is, exponential. In addition we found that any steady solutions obtained were "almost" exact, starting from $p = 2$. We shall explain this below.

Assume we wish to solve a boundary-value problem for the 1-D Poisson equation

$$Du_{xx} + s(x, y) = 0,\ x \in (0, 1), \tag{21}$$

with a Dirichlet boundary condition at $x = 0$ and a Neumann[c] condition at $x = 1$. For the solution we use an orthogonal polynomial basis of order p, with each $v_k,\ k = 1, ..., p + 1$, a polynomial of degree $k - 1$. We may find the solution to this problem by marching the associated time-dependent problem to a steady state. The DG equations describing the steady state read

$$D\ \{v_k f_x - f(v_k)_x\}|_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} + D\int_{\Omega_j} u(v_k)_{xx}dx + \int_{\Omega_j} v_k s dx = 0, \quad k = 1, ..., p + 1. \tag{22}$$

Consider the first equation ($k = 1$, v_1 a constant); it reduces to

$$D\ v_1 f_x|_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} + \int_{\Omega_j} v_1 s dx = 0. \tag{23}$$

This equation is in conservation form; it relates the change of the diffusive flux f_x across a mesh to the source-term integral over the mesh. Since f_x is known exactly at the Neumann boundary, and the source-term integral can be computed exactly, we may sweep through the mesh starting from the Neumann boundary, and one by one obtain all face values of f_x exactly.

In the next equation ($k = 2$, v_2 linear) the values of f itself appear:

$$D\ \{v_2 f_x - f(v_2)_x\}|_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} + \int_{\Omega_j} v_2 s dx = 0. \tag{24}$$

Thus the values of f_x and f at the two mesh faces are coupled to a weighted source integral; the latter may again be computed exactly. With all face values of f_x already known, and the value of f known at the Dirichlet boundary, we may now sweep through the mesh starting from that boundary, and obtain one by one all face values of f exactly.

In the third equation ($k = 3$, v_3 quadratic) the interior integral

$$D\int_{\Omega_j} u(v_2)_{xx}dx \tag{25}$$

[c] The argument goes through as well for a pure Dirichlet problem, but is more involved.

kicks in. With $(v_3)_{xx}$ a constant, this integral is proportional to the mesh average $\bar{u}_j$ of the solution ($\equiv a_1$). Since all face values of f_x and f are known by now, and the weighted source integral can be computed exactly, the third equation yields the exact value of $\bar{u}_j$, for all j.

In each of the following equations, $k = 4, ..., p+1$, the interior integral includes the next higher moment of the solution on the mesh, equivalent to the $(k-3)^{\text{rd}}$ solution derivative, and its value is found exactly. Only the $(p-1)^{\text{st}}$ and p^{th} moment get no special equation; these will carry numerical errors of order $2^{p+1}-1$ and $2^{p+1}-2$, respectively.

Figure 5 shows the mesh convergence of solution properties for a 1-D Poisson problem, with $p = 2$. The error norms of the solution coefficients $a_3 \equiv \overline{\Delta^2 u}$ and $a_2 \equiv \overline{\Delta u}$ indicate 6th- and 7th-order accuracy; for $a_1 \equiv \bar{u}$ we would expect 8th-order accuracy, but the errors are computer-zero for all mesh-sizes.

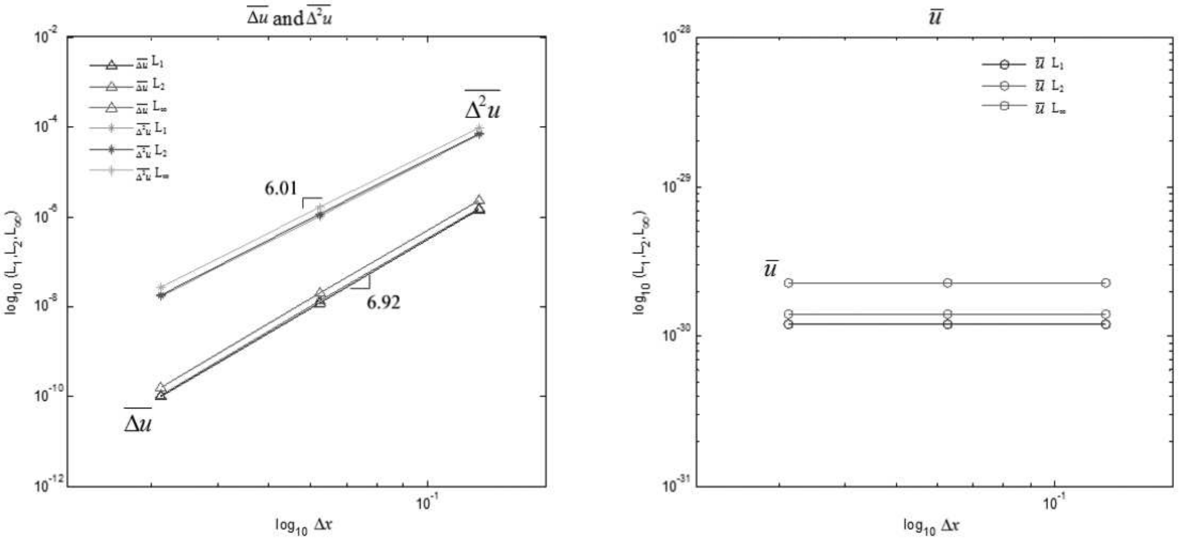

Figure 5. Mesh convergence for 1-D Poisson problem; $p = 2$. Left: convergence with mesh size of error norms of solution coefficients $\overline{\Delta^2 u}$ and $\overline{\Delta u}$. Right: convergence of $\bar{u}$.

While the above property of $(p-2)$-exactness may never play a role in practical problems, it shows the distinguished pedigree of the recovery method. Note that in the proof we have not made use of the details of recovery; what matters is: (1) a unique, differentiable function replaces the discontinuous solution at each element face; (2) the DG equations used are those resulting from integration by parts *twice*.

Note that if the source integral is evaluated with a numerical error, the solution coefficents will inherit that error.

Exponential accuracy $(p-2)$-exactness are is lost in two dimensions. Numerical solutions with $p = 1$ and 2 to 2-D Poisson problems on rectangular domains and meshes indicate the order of accuracy of the recovery method is $2(p+1)$. This relates to the order of accuracy of the recovered solution in the direction normal to the element boundary: the number of solution parameters doubles.

VII. Numerical results

Figure 6 shows properties of the solution to the following 2-D Poisson problem on $[0,1] \times [0,1]$:

$$\begin{aligned} U(x,0) &= \frac{1}{2}\cos(2\pi x), & (26)\\ U(x,1) &= \frac{1}{2}\cos(2\pi x), & (27)\\ U(0,y) &= \frac{1}{2}\cos(2\pi y), & (28)\\ U(1,y) &= \frac{1}{2}\cos(2\pi y), & (29)\\ s(x,y) &= 2\pi^2\{\cos(2\pi x)+\cos(2\pi y)\} & (30) \end{aligned}$$

The exact solution of the problem is

$$U(x,y) = \frac{1}{2}\{\cos(2\pi x)+\cos(2\pi y)-1\}. \qquad (31)$$

The numerical solution was taken to be piecewise linear ($p = 1$, $K = 3$) on square meshes, and obtained by marching the associated unsteady problem till convergence. The boundary treatment was as in Figure 4.

Figure 6, left plot, shows that the piecewise linear recovery-based DG method is 4$^{\text{th}}$-order accurate, even when measured in the maximum norm. The solution coefficients and errors shown on the right are for the 16×16 mesh. It is worth mentioning that the numerical solution exhibits all 4 symmetry axes of the exact solution.

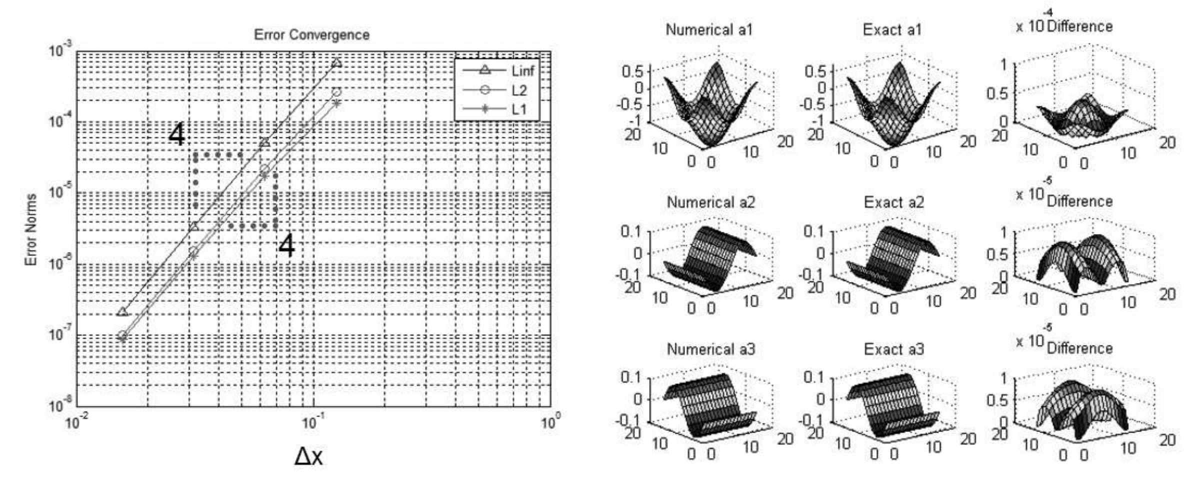

Figure 6. Mesh convergence for 2-D Poisson problem; $p = 1$. Left: convergence with mesh size of error norms of the mesh average $a_1 \equiv \bar{u}$. Right: computed solution coefficients a_1 (average value), a_2 ($\sim$ average x-derivative) and a_3 ($\sim$ average y-derivative); exact solution coefficients; coefficient errors.

An intriguing aspect of the 2-D recovery method for $p = 1$ is that the solution gradient is calculated with higher precision than the solution average: the coefficients a_2 and a_3 show 5$^{\text{th}}$-order mesh convergence in the L_1 norm. For $p = 2$ the method behaves again as expected, with 6$^{\text{th}}$-, 5$^{\text{th}}$- and 4$^{\text{th}}$-order accuracy for the solution average, gradient and second derivatives, respectively.

VIII. Conclusions and perspective

We have presented the details of the recovery-based DG method for 2-D diffusion problems, including the treatment of domain boundaries. The information provided is sufficient for programming the recovery scheme on an unstructured triangular grid. We expect the recovery scheme to remain stable and accurate on such a grid, although general proofs of stability and accuracy are missing.

Numerical experiments on a 2-D rectangular grid show stability and suggest an order of accuracy of $2(p+1)$; we expect the order to degrade on an unstructured grid. We further have shown stability of the method for $p = 1$ on a structured triangular grid.

We are currently gathering experience with including the recovery method in DG codes for linear and nonlinear advection-diffusion.

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, "Recovery bases for Discontinuous Galerkin diffusion operators," in *Proceedings of the 2007 Conference on Numerical Methods in Fluid Dynamics (ICFD 2007), 26-29 March 2007, Reading, UK*, 2007.

[3] M. Raalte and B. van Leer, "Bilinear forms for the recovery-based discontinuous galerkin method for diffusion," Tech. Rep. MAS-E0707, Center for Mathematics and Informatics (CWI), 2007.

[4] F. Brezzi, G. Manzini, D. Marini, P. Pietra, and A. Russo, "Discontinuous Galerkin approximations for elliptic problems," *Numerical Methods for Partial Differential Equations*, vol. 16, pp. 365–378, 2000.

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