

## Selecting the Numerical Flux in Discontinuous Galerkin Methods for Diffusion Problems

Robert M. Kirby<sup>1</sup> and George Em Karniadakis<sup>2</sup>

*Received August 6, 2003; accepted (in revised form) March 5, 2004*

---

In this paper we present numerical investigations of four different formulations of the discontinuous Galerkin method for diffusion problems. Our focus is to determine, through numerical experimentation, practical guidelines as to which numerical flux choice should be used when applying discontinuous Galerkin methods to such problems. We examine first an inconsistent and weakly unstable scheme analyzed in Zhang and Shu, *Math. Models Meth. Appl. Sci. (M<sup>3</sup>AS)* **13**, 395–413 (2003), and then proceed to examine three consistent and stable schemes: the Bassi–Rebay scheme (*J. Comput. Phys.* **131**, 267 (1997)), the local discontinuous Galerkin scheme (*SIAM J. Numer. Anal.* **35**, 2440–2463 (1998)) and the Baumann–Oden scheme (*Comput. Math. Appl. Mech. Eng.* **175**, 311–341 (1999)). For an one-dimensional model problem, we examine the stencil width,  $h$ -convergence properties,  $p$ -convergence properties, eigenspectra and system conditioning when different flux choices are applied. We also examine the ramifications of adding stabilization to these schemes. We conclude by providing the *pros* and *cons* of the different flux choices based upon our numerical experiments.

---

**KEY WORDS:** Discontinuous Galerkin methods; spectral/ $hp$  elements; parabolic flux choices; stabilization.

### 1. INTRODUCTION

Although the original thrust of most discontinuous Galerkin (DG) research was in solving hyperbolic problems, the general proliferation of the DG methodology has also spread to the study of parabolic and elliptic problems. For example, works such as [4], in which the viscous compressible Navier–Stokes equations were solved, required that a DG formulation be

---

<sup>1</sup> School of Computing, University of Utah. E-mail: kirby@cs.utah.edu

<sup>2</sup> Division of Applied Mathematics, Brown University. E-mail: gk@dam.brown.edu

extended beyond the hyperbolic advection terms to the viscous terms of the Navier–Stokes equations. Concurrently, both in [9] and [5] other DG formulations for parabolic and elliptic problems were proposed. In an effort to classify all the efforts made toward the use of DG methods for elliptic problems, Arnold *et al.*, first in [1] and then more fully in [2], published a unified analysis of DG methods for elliptic problems.

In [2] a mathematical framework is provided for studying different versions of DG approaches for elliptic problems. We first recognize from [2] that the problem of solving

$$-\Delta u = f \quad \text{in } \Omega \tag{1}$$

$$u = 0 \quad \text{on } \partial\Omega \tag{2}$$

can be formulated in the discrete case as follows.

Assume we are given a tessellation  $\mathcal{T}_h = \{K\}$  of the domain  $\Omega$ . We define the following two spaces:

$$V_h := \{v \in L^2(\Omega) : v|_K \in P(K) \forall K \in \mathcal{T}_h\},$$

$$\Sigma_h := \{\tau \in [L^2(\Omega)]^2 : \tau|_K \in \Sigma(K) \forall K \in \mathcal{T}_h\},$$

where  $P(K) = \mathcal{P}_p(K)$  is the space of polynomial functions of degree at most  $p \geq 1$  on  $K$  and  $\Sigma(K) = [\mathcal{P}_p(K)]^2$ . Following [2] we now define the discrete solution of Eq. (1) as the problem of finding  $u_h \in V_h$  and  $\sigma_h \in \Sigma_h$  such that for all  $K \in \mathcal{T}_h$

$$\int_K \sigma_h \cdot \tau \, dx = - \int_K u_h \nabla_h \cdot \tau \, dx + \int_{\partial K} \hat{u}_K n_K \cdot \tau \, ds, \tag{3}$$

$$\int_K \sigma_h \cdot \nabla v \, dx = \int_K f v \, dx + \int_{\partial K} v \hat{\sigma}_K \cdot n_K \, ds, \tag{4}$$

where the numerical fluxes  $\hat{\sigma}_K$  and  $\hat{u}_K$  are approximations to  $\sigma = \nabla u$  and to  $u$ , respectively, on the boundary of  $K$ . Given this general unified formulation of the discrete problem, the two remaining choices which determine exactly which DG methodology is used is the choice of the numerical fluxes  $\hat{\sigma}_K$  and  $\hat{u}_K$ . Although theoretical considerations are discussed, the reader is still left with the question of which flux choice should be used and why.

There have been several attempts to provide performance information concerning the flux choices, both by the developers of different flux choices (e.g. [9,5]) and by those interested in flux choice comparisons (e.g. [16,3,15,8]). For an overview of many of the properties of the DG method, from the theoretical perspective, the performance perspective, and the usage perspective, we refer the reader to the review article [10] and

the references therein. From our perspective, however, there does not appear to be clear-cut guidelines within the literature for aiding someone in determining what are the computational trade-offs involved in the flux choice.

In an attempt to ascertain the trade-offs between the different flux choices, we set out to study several of the different formulations presented in [2]. In Table I we present the methodologies and the corresponding numerical fluxes for which we will present results. The operator  $\{\cdot\}$  denotes averaging across the interface while  $[[\cdot]]$  denotes the jump difference across the interface as described in [2].

Our goal is to determine, through numerical investigation, the trade-offs between different fluxes. To accomplish this numerical investigation, we will present a very simple model problem, and will investigate the stencil width,  $h$ -convergence,  $p$ -convergence, eigenspectra and system conditioning associated with different flux choices.

### 1.1. Model Problem and Notation

The model problem which we will use for our evaluation of the various methods is diffusion, i.e.,

$$\frac{\partial u(x, t)}{\partial t} = \frac{\partial^2 u(x, t)}{\partial x^2}, \quad x \in (0, 2\pi) \tag{5}$$

with periodic boundary conditions and an initial condition  $u(x, t = 0) = \sin(x)$ .

As in [16], let us denote  $I_j = [x_{j-(1/2)}, x_{j+(1/2)}]$ , for  $j = 1, \dots, N$  as our elemental mesh on  $[0, 2\pi]$  where  $x_{1/2} = 0$  and  $x_{N+(1/2)} = 2\pi$ . We define the following set of piecewise polynomials:

$$V_P = \{v : v \text{ is a polynomial of degree at most } P \text{ for } x \in I_j, j = 1, \dots, N\},$$

which will be used for both our trial and test spaces. Unless otherwise stated, the orthogonal Legendre basis [7] was used for all experiments.

**Table I.** Proposed DG Methodologies for Elliptic Problems and the Flux Choices They Represent

Method	$\hat{u}_K$	$\hat{\sigma}_K$
Bassi–Rebay [4]	$\{u_h\}$	$\{\sigma_h\}$
LDG [9]	$\{u_h\} - \beta \cdot [[u_h]]$	$\{\sigma_h\} + \beta [[\sigma_h]] - \alpha_j ([[u_h]])$
Baumann–Oden [5]	$\{u_h\} + n_K \cdot [[u_h]]$	$\{\nabla_h u_h\}$

All computations were accomplished with respect to modal expansion coefficients; as such, the model problem initial condition specified above was first projected to the space of piecewise polynomials based upon the elemental decomposition and polynomial order per element. All inner product calculations were accomplished using Gauss–Legendre quadrature [7] of sufficient order to guarantee exact numerical integration of the inner products of the polynomials used. Error ( $L_2$ ) calculations were accomplished numerically using the same quadrature rules as used for formulating the polynomial inner products. The computed approximate solution was compared against the true exact solution, not the projected (to the piecewise polynomial space) solution in all cases, and hence due to the quadrature rules employed the error presented herein is a numerical approximation of the true  $L_2$  error.

The three primary fluxes which we will study in this paper are the Bassi–Rebay flux [4] which we will denote with the initials BR, the LDG flux [9] which we will denote with the initials LDG, and the Baumann–Oden flux [5] which we will denote with the initials BO (as summarized in Table I). To accomplish our study, we will follow the work of Shu in [15], and by algebraic manipulation rewrite Eqs. (3 and 4) to eliminate the auxiliary variable  $\sigma$  from the formulation (taking into account the proper flux when manipulating the variable out of the expression). For our model problem this manipulation leads to the following systems for BR, LDG, and BO, respectively:

$$\frac{d\hat{u}_j}{dt} = A_{-2}^{\text{BR}}\hat{u}_{j-2} + A_{-1}^{\text{BR}}\hat{u}_{j-1} + A_0^{\text{BR}}\hat{u}_j + A_1^{\text{BR}}\hat{u}_{j+1} + A_2^{\text{BR}}\hat{u}_{j+2}, \quad (6)$$

$$\frac{d\hat{u}_j}{dt} = A_{-1}^{\text{LDG}}\hat{u}_{j-1} + A_0^{\text{LDG}}\hat{u}_j + A_1^{\text{LDG}}\hat{u}_{j+1}, \quad (7)$$

$$\frac{d\hat{u}_j}{dt} = A_{-1}^{\text{BO}}\hat{u}_{j-1} + A_0^{\text{BO}}\hat{u}_j + A_1^{\text{BO}}\hat{u}_{j+1}, \quad (8)$$

where  $\hat{u}_j$  denotes a vector of the modal coefficients of the polynomial expansion on an element  $j$ , and the matrices  $A_k$  are formulated based upon the choice of the numerical fluxes  $\hat{\sigma}_K$  and  $\hat{u}_K$  in Eqs. (3) and (4). The subscript  $k$  on each matrix  $A_k$  denotes the offset from the current element  $j$  for which the solution is being sought. The particular LDG stencil above corresponds to a choice of  $\beta=1/2$  as in the works of [16,15]. A different choice of the  $\beta$  parameter may lead to a wider stencil and different numerical properties for LDG. For the purposes of this paper we limit ourselves to examining the cases of  $\beta=0$  (which, when no stabilization is added, reverts to the BR scheme), and  $\beta=1/2$  as in [15,16]. We can now write our numerical approximation of the model problem in the following form:

$$\frac{d\hat{u}_g}{dt} = \mathbf{A}\hat{u}_g, \tag{9}$$

where  $\hat{u}_g$  denotes the concatenation of modal coefficients of each element (hence if given  $N$  elements, each having  $M$  modal coefficients, the size of  $\hat{u}_g$  is  $N \times M$ ), and  $A$  is a size( $\hat{u}_g$ )  $\times$  size( $\hat{u}_g$ ) square matrix. When examining eigenspectra, we will examine the matrix  $\mathbf{A}$  associated with different flux choices, and will denote the choice with a subscript (such as  $\mathbf{A}_{\text{BR}}$  for the matrix based upon the BR flux choice).

For all numerical tests accomplished in this paper, we will use the second-order implicit Crank–Nicolson scheme, which can be written as

$$\mathbf{L}_{\text{CN}}\hat{u}_g^{n+1} = \left( \mathbf{M} + \frac{1}{2}\Delta t\mathbf{A} \right)\hat{u}_g^n, \tag{10}$$

$$\mathbf{L}_{\text{CN}} = \left( \mathbf{M} - \frac{1}{2}\Delta t\mathbf{A} \right), \tag{11}$$

where  $\mathbf{M}$  denotes the mass matrix and  $\mathbf{A}$  denotes the spatial operator matrix as described above. When discussing the conditioning of the system, we will examine  $\mathbf{L}_{\text{CN}}$ , since this is the matrix term which requires inversion in the above expression.

### 1.2. Outline

This paper is divided as follows. We examine four different one-dimensional formulations of the DG method for our model problem. In Sec. 2 we examine the “inconsistent scheme” analyzed in [15,16]. In Sec. 3, we present a convergence study, eigenspectra and conditioning information for the BR formulation. In Sec. 4 we present similar information for LDG and in Sec. 5 we present information for the BO formulation. In Sec. 6 we present some effects of stabilization. Finally, in Sec. 7, we summarize our finding by providing the trade-offs for using each method based upon this numerical study.

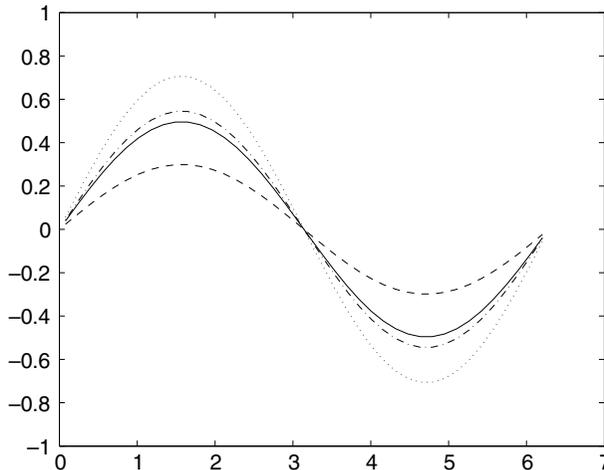
## 2. FORMULATION 1: THE “INCONSISTENT” SCHEME

The first scheme we investigate is the “inconsistent” scheme – a scheme deemed to be both inconsistent and weakly unstable in the analysis of [16]. The solution of our model problem using this scheme is to find  $u \in V_P$  such that

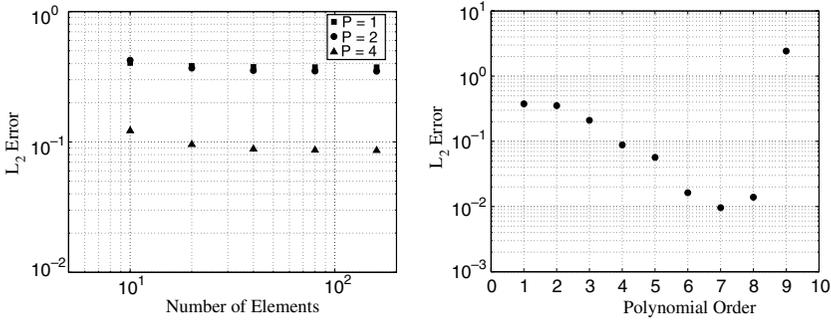
$$\int_{I_j} u_t v \, dx + \int_{I_j} u_x v_x \, dx - (\hat{u}_x)_{j+\frac{1}{2}} v_{j+\frac{1}{2}}^- + (\hat{u}_x)_{j-\frac{1}{2}} v_{j-\frac{1}{2}}^+ = 0$$

for all test function  $v \in V_P$ . Since there is no upwinding mechanism in the diffusion problem of interest, we will take  $(\hat{u}_x)_{j+(1/2)} = (1/2) ((\hat{u}_x^+)_{j+(1/2)} + (\hat{u}_x^-)_{j+(1/2)})$ . In Fig. 1 we present solutions to the model problem using three different polynomial orders per element:  $P = 1$ ,  $P = 2$  and  $P = 4$ . Forty evenly spaced elements were used for all three polynomial orders, and a time step of  $10^{-5}$  was employed with the CN time stepping scheme. Results are shown at  $T = 0.7$ .

To help elucidate the statements about consistency made in [16], we examine both the  $h$ -convergence and  $p$ -convergence of this scheme. In Fig. 2 we present both  $h$ -convergence (left) and  $p$ -convergence (right) plots. We observe that for a fixed polynomial order, the method does not converge upon elemental refinement. This is consistent with the claims made in [16]. For a fixed number of elements (40 evenly spaced elements), upon  $p$ -refinement, we observe what appears to be the initial signs of convergence. As the polynomial order is increased, however, the solution starts to diverge from the true solution. This phenomenon is consistent with the analysis shown in [16] in which an  $\mathcal{O}(1/\Delta x)$  instability is predicted. As the polynomial order is increased, the wavelength support (a measure of spatial resolution) is increased, and hence the instability mentioned in [16] becomes prominent. Also,  $h$ -convergence is insufficient to demonstrate this instability as the wavelength support added by increasing the elemental resolution is slow compared to polynomial



**Fig. 1.** Solution of the model problem using formulation 1. The exact solution (solid) and polynomial orders  $P = 1$  (dotted),  $P = 2$  (dashed) and  $P = 4$  (dot-dashed) are presented at time  $T = 0.7$ . Forty evenly spaced elements were used.



**Fig. 2.** Convergence study of formulation 1 based upon the model problem evaluated at  $T = 0.7$ : On the left, we present the  $L_2$  error vs. the number of evenly spaced elements having polynomial orders  $P = 1$  (squares),  $P = 2$  (circles) and  $P = 4$  (triangles). On the right we present the  $L_2$  error vs. polynomial order for a mesh consisting of 40 evenly spaced elements.

refinement. To verify that this phenomenon is not a function of the CN time stepping algorithm, we also studied the  $p$ -convergence when using the implicit first-order Euler–Backward scheme. The convergence diagram did not change (for instance, the  $L_2$  error for the Euler–Backward scheme for a ninth-order discretization was 2.52 compared to 2.43 using CN; the small discrepancy is due to the difference in the time integration order).

### 3. FORMULATION 2: BASSI-REBAY FLUX CHOICE

The first consistent scheme that we examine is given by splitting the solution of the model problem into two equations. We seek to find  $u, q \in V_P$  such that, for all test functions  $v, w \in V_P$ ,

$$\int_{I_j} u_t v \, dx + \int_{I_j} q v_x \, dx - \hat{q}_{j+\frac{1}{2}} v_{j+\frac{1}{2}}^- + \hat{q}_{j-\frac{1}{2}} v_{j-\frac{1}{2}}^+ = 0,$$

$$\int_{I_j} q w \, dx + \int_{I_j} u w_x \, dx - \hat{u}_{j+\frac{1}{2}} w_{j+\frac{1}{2}}^- + \hat{u}_{j-\frac{1}{2}} w_{j-\frac{1}{2}}^+ = 0,$$

where for flux choices we make the choice of BR [4]:

$$\hat{u}_{j+\frac{1}{2}} = \frac{1}{2} \left( u_{j+\frac{1}{2}}^+ + u_{j+\frac{1}{2}}^- \right), \quad \hat{q}_{j+\frac{1}{2}} = \frac{1}{2} \left( q_{j+\frac{1}{2}}^+ + q_{j+\frac{1}{2}}^- \right).$$

The scheme above has been shown in [2] to be both consistent and stable for all polynomial orders.

The first observation that can be made, as in [15], is that averaging in both the primary and the auxiliary variable yields a five element wide stencil. This observation will become important in discussing the eigenspectra and the system conditioning. We will now proceed to examine the convergence rate, eigenspectra and system conditioning for this flux choice.

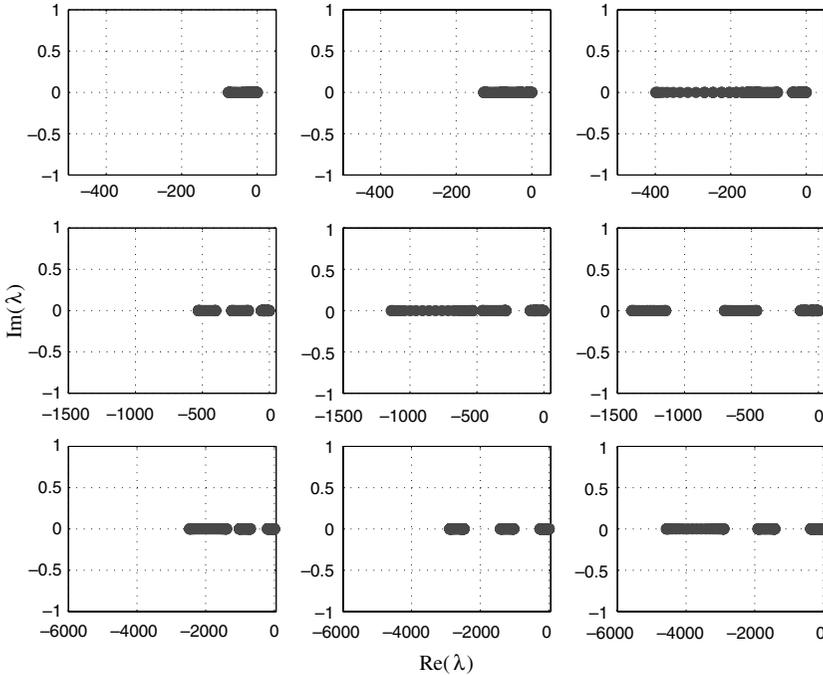
### 3.1. Convergence Rate

In Table II we present a convergence study using the BR flux. For this study, we examine five different numbers of evenly spaced elements (10, 20, 40, 80, 160) with polynomial orders varying systematically from  $P = 1$  to  $P = 6$ . For this test, the model problem was solved up to time  $T = 0.7$  using the second-order CN scheme with a time step of  $\Delta t = 10^{-5}$ . In Table II we present the error defined as the  $L_2$  difference between the approximate and exact solution. In the table, the symbol ‘-’ denotes when the error due to the spatial discretization is less than  $10^{-10}$ , and hence the time error becomes the dominant error. As was shown in [3,15], the order of accuracy is  $P$  when the polynomial order is *odd* (sub-optimal) and  $P + 1$  when the polynomial order is *even* (optimal). In Figs. 6 and 4 we present a comparison of the  $h$ -convergence and  $p$ -convergence between BR, LDG and BO, respectively. In Fig. 6 we examine the  $h$ -convergence of the method (denoted with circles) for two different polynomial orders,  $P = 1$  (solid line) and  $P = 2$  (dashed line). In Fig. 4, we examine the  $p$ -convergence of the method (denoted with circles) when 40 evenly spaced elements are used. The method exhibits a stair-case convergence as the polynomial order is increased, consistent with the optimal and sub-optimal estimates mentioned above. With respect to the optimal parity (even), the scheme exhibits

**Table II.** BR Convergence Data:  $L_2$  Error Computed when Solving the Model Problem Evaluated at  $T = 0.7$ .

Polynomial order	$N = 10$	$N = 20$	$N = 40$	$N = 80$	$N = 160$
1	4.1349e-02	2.0084e-02	9.9664e-03	4.9737e-03	2.4856e-03
2	7.2334e-04	8.6986e-05	1.0776e-05	1.3441e-06	1.6792e-07
3	8.8529e-05	1.0827e-05	1.3457e-06	1.6797e-07	2.0988e-08
4	9.0255e-07	2.7175e-08	8.4172e-10	-	-
5	7.3355e-08	2.2518e-09	-	-	-
6	5.3352e-10	-	-	-	-

Evenly spaced elements were used in space; second-order Crank–Nicolson with at time step of  $\Delta t = 10^{-5}$  was used in time. Entries denoted with ‘-’ represent cases where the spatial error is less than  $10^{-10}$ , and hence the time stepping error becomes the dominant error.



**Fig. 3.** Eigenspectra of the spatial operator for BR  $\mathbf{A}_{BR}$ . Forty elements were used in all cases; each plot denotes a different polynomial order. The polynomial order runs from first-order  $P=1$  to ninth-order  $P=9$  in row-major order. The ordinate of each plot is the complex imaginary axis, and the abscissa is the complex real axis. Note that the axes scales are only consistent across rows due to the large magnitude variation in the spectra due to polynomial order.

exponential convergence. Comparative statements between the methods will be made later in the paper.

### 3.2. Eigenspectra

In Fig. 3 we present the eigenspectra of the spatial operator  $\mathbf{A}_{BR}$  formed using BR flux. The operator is a real symmetric matrix, and hence yields eigenvalues which are real. We present eigenspectra diagrams for nine different polynomial orders running from first-order  $P=1$  to ninth-order  $P=9$  in row-major order.

As we would expect, increasing the polynomial order increases the absolute maximum eigenvalue. In Table III we present the maximum absolute eigenvalue ( $\max |\lambda_i|$ ) for different element number and polynomial order combinations. In Fig. 8, we present a graph of maximum absolute

**Table III.** BR Maximum Absolute Eigenvalue Study: Maximum Absolute Eigenvalue of the Discrete Operator  $\mathbf{A}_{BR}$  Approximating the Second-order Spatial Derivative Operator for Different Number of Elements and Polynomial Order Per Element.

Polynomial order	$N = 10$	$N = 20$	$N = 40$	$N = 80$	$N = 160$
1	1.9099e+01	3.8197e+01	7.6394e+01	1.5279e+02	3.0558e+02
2	3.1831e+01	6.3662e+01	1.2732e+02	2.5465e+02	5.0930e+02
3	9.9688e+01	1.9938e+02	3.9875e+02	7.9750e+02	1.5950e+03
4	1.3268e+02	2.6535e+02	5.3071e+02	1.0614e+03	2.1228e+03
5	2.8474e+02	5.6949e+02	1.1390e+03	2.2779e+03	4.5559e+03
6	3.4837e+02	6.9673e+02	1.3935e+03	2.7869e+03	5.5738e+03
7	6.1846e+02	1.2369e+03	2.4739e+03	4.9477e+03	9.8954e+03
8	7.2296e+02	1.4459e+03	2.8919e+03	5.7837e+03	1.1567e+04
9	1.1448e+03	2.2897e+03	4.5793e+03	9.1586e+03	1.8317e+04
10	1.3004e+03	2.6009e+03	5.2017e+03	1.0403e+04	2.0807e+04
11	1.9078e+03	3.8156e+03	7.6312e+03	1.5262e+04	3.0525e+04
12	2.1247e+03	4.2494e+03	8.4989e+03	1.6998e+04	3.3995e+04
13	2.9513e+03	5.9026e+03	1.1805e+04	2.3610e+04	4.7221e+04
14	3.2398e+03	6.4795e+03	1.2959e+04	2.5918e+04	5.1836e+04
15	4.3193e+03	8.6386e+03	1.7277e+04	3.4554e+04	6.9109e+04
16	4.6895e+03	9.3791e+03	1.8758e+04	3.7516e+04	7.5033e+04

Evenly spaced elements were used in all cases.

eigenvalue vs. polynomial order for a 40 evenly spaced element mesh (circles denote BR). The increase in the magnitude is of order  $P^4$  where  $P$  is the order of the polynomial approximation used. This coincides with the commonly used  $1/P^4$  estimate for the diffusion number when using spectral methods for solving parabolic problems.

### 3.3. Conditioning

When solving our model problem implicitly, we are interested in inverting the operator  $\mathbf{L}_{CN}$  as described above when formed using the spatial operator  $\mathbf{A}_{BR}$ . In Table IV we examine the  $L_2$  condition number of the matrix  $\mathbf{L}_{CN}$  before and after diagonal preconditioning (denoted by multiplying by a matrix  $\mathbf{Z}$  which consists of the inverse diagonal operator). For this experiment, a 40 evenly spaced element discretization using a time step of  $\Delta t = 10^{-5}$  was used. This time step was chosen so as to yield a time stepping error on the order of  $10^{-10}$  when using the second-order CN scheme. A different choice of time step will change the absolute numbers presented, however trends can be assessed. It is also important to note that variations in the elemental spacing and in the choice of basis may strongly influence the condition number [11]; this must be considered when interpreting the conditioning results presented herein.

**Table IV.** Condition Number Comparison Before and After Diagonal Preconditioning for the Linear Operator  $L_{CN}$  formed Using BR.

Polynomial order	$\kappa_2(L_{CN})$	$\kappa_2(ZL_{CN})$
1	3.0073	1.0037
2	5.0122	1.0141
3	7.0570	1.0446
4	9.0733	1.0929
5	11.1912	1.1970
6	13.2259	1.3310

A mesh consisting of 40 evenly spaced elements and a time step of  $\Delta t = 10^{-5}$  was used.

As the polynomial order is increased, the condition number of the system increases (as expected). It is interesting to note that the growth in the condition number appears linear with respect to the polynomial order. Recall that we are not examining the condition number of the spatial operator  $A_{BR}$  as done in [8], but rather the condition number of  $L_{CN}$ , which is the matrix that we must invert due to the implicit time stepping algorithm which we are using. The CN scheme applied to this system produces a system which is diagonally dominant, and hence diagonal preconditioning works well. The new system, which is symmetric and has a condition number near one, is now a prime candidate for using conjugate gradient methods. Numerical experiments found that the number of iterations necessary to solve the preconditioned system was at least an order of magnitude lower than the rank of the original system.

When less stringent time stepping errors are required, and hence larger time steps are used, the effect of the diagonal preconditioner becomes less pronounced. For instance, given a time step of  $10^{-3}$  with sixth-order polynomials, diagonal preconditioning reduces the condition number of the system by a factor of 1.2.

#### 4. FORMULATION 3: LOCAL DISCONTINUOUS GALERKIN (LDG) FLUX CHOICE

The second consistent scheme that we examine is given by splitting the solution of the model problem into two equations. We seek to find  $u, q \in V_P$  such that, for all test functions  $v, w \in V_P$ ,

$$\int_{I_j} u_t v \, dx + \int_{I_j} q v_x \, dx - \hat{q}_{j+\frac{1}{2}} v_{j+\frac{1}{2}}^- + \hat{q}_{j-\frac{1}{2}} v_{j-\frac{1}{2}}^+ = 0,$$

$$\int_{I_j} qw \, dx + \int_{I_j} uw_x \, dx - \hat{u}_{j+\frac{1}{2}} w_{j+\frac{1}{2}}^- + \hat{u}_{j-\frac{1}{2}} w_{j-\frac{1}{2}}^+ = 0,$$

where for flux choices we make the choice of Cockburn and Shu [9]

$$\hat{u}_{j+\frac{1}{2}} = u_{j+\frac{1}{2}}^+, \quad \hat{q}_{j+\frac{1}{2}} = q_{j+\frac{1}{2}}^-.$$

The scheme above has been shown in [2] to be both consistent and stable for all polynomial orders.

The first observation that can be made, as in [15], is that this scheme yields a three element stencil. The “flip-flopping” of the flux choice yields a three element wide stencil, which is tighter spatially than the BR flux discussed previously. This observation will become important in discussing the eigenspectra and the system conditioning. We will now proceed to examine the convergence rate, eigenspectra and system conditioning for this flux choice.

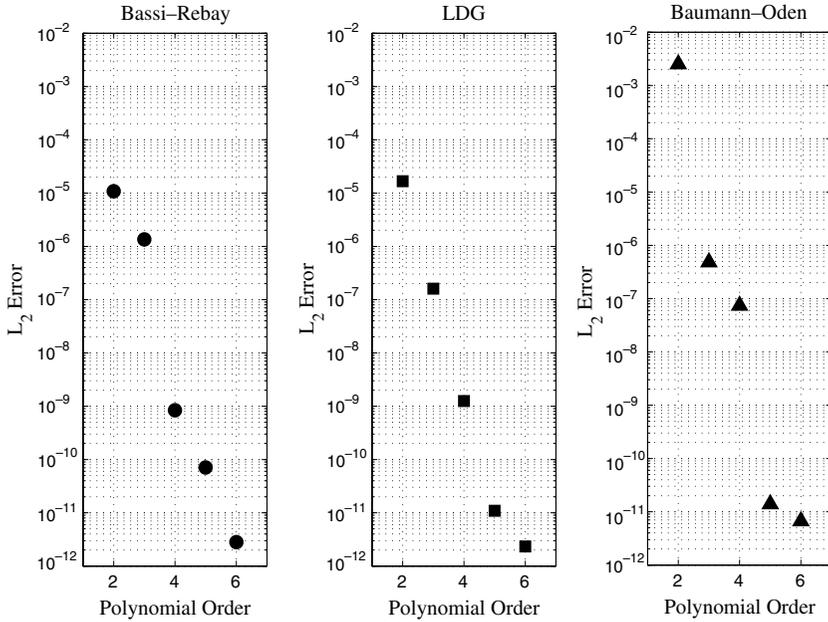
#### 4.1. Convergence

In Table V we present a convergence study using the LDG flux. For this study, we examine five different numbers of evenly spaced elements (10, 20, 40, 80, 160) with polynomial orders varying systematically from  $P = 1$  to  $P = 6$ . For this test, the model problem was solved up to time  $T = 0.7$  using the second-order CN scheme with a time step of  $\Delta t = 10^{-5}$ . In Table V we present the error defined as the  $L_2$  difference between the approximate and exact solution. In the Table, the symbol ‘-’ denotes when the error due to the spatial discretization is less than  $10^{-10}$ , and hence the

**Table V.** LDG Convergence Data:  $L_2$  Error Computed when Solving the Model Problem Evaluated at  $T = 0.7$

Polynomial Order	$N = 10$	$N = 20$	$N = 40$	$N = 80$	$N = 160$
1	2.1270e-02	5.2941e-03	1.3221e-03	3.3045e-04	8.2607e-05
2	1.0662e-03	1.3319e-04	1.6646e-05	2.0807e-06	2.6009e-07
3	4.1068e-05	2.5706e-06	1.6072e-07	1.0046e-08	6.2812e-10
4	1.2779e-06	4.0010e-08	1.2510e-09	-	-
5	3.3266e-08	5.2098e-10	-	-	-
6	7.4372e-10	-	-	-	-

Evenly spaced elements were used in space; second-order CN with at time step of  $\Delta t = 10^{-5}$  was used in time. Entries denoted with ‘-’ represent cases where the spatial error is less than  $10^{-10}$ , and hence the time stepping error becomes the dominant error.

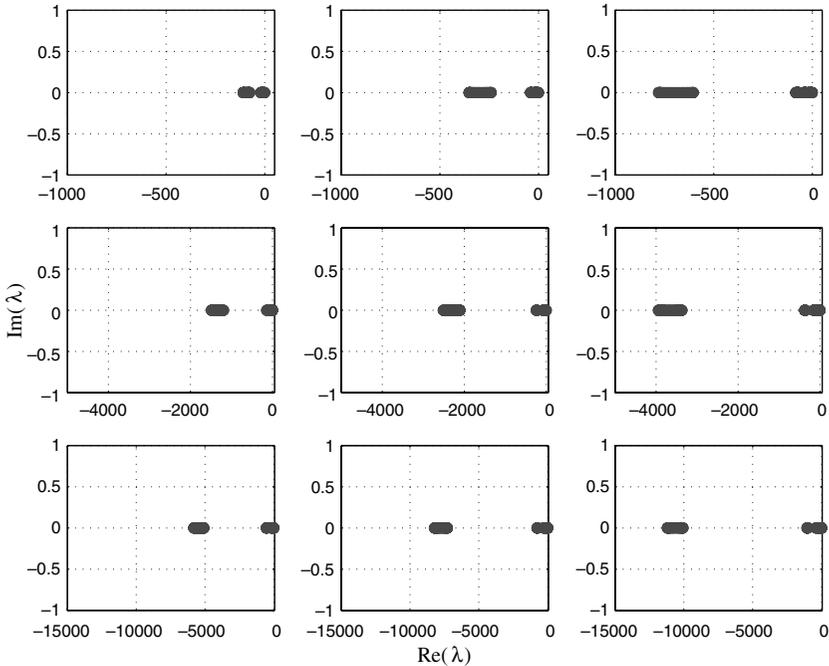


**Fig. 4.**  $p$ -Convergence study comparison of BR (circles), LDG (squares) and BO (triangles) Based upon the model problem evaluated at  $T=0.7$ ; We present the  $L_2$  error vs. polynomial order for a mesh consisting of 40 evenly spaced elements.

time error becomes the dominant error. As was shown in [16,3], the order of accuracy is  $P + 1$  (optimal) irrespective of polynomial order. In Fig. 6 we examine the  $h$ -convergence of the method (denoted with squares) for two different polynomial orders,  $P = 1$  (solid line) and  $P = 2$  (dashed line). In Fig. 4, we examine the  $p$ -convergence of the method (denoted with squares) when 40 evenly spaced elements are used. The method exhibits exponential convergence as the polynomial order is increased, independent of the parity. Comparative statements between the methods will be made later in the paper.

### 4.2. Eigenspectra

In Fig. 5 we present the eigenspectra of the spatial operator  $\mathbf{A}_{LDG}$  formed using LDG fluxes. The operator is a real symmetric matrix, and hence yields eigenvalues which are real. We present eigenspectra diagrams for nine different polynomial orders running from first-order  $P = 1$  to ninth-order  $P = 9$  in row-major order. Observe the nice clustering property of the



**Fig. 5.** Eigenspectra of the spatial operator for LDG  $\mathbf{A}_{\text{LDG}}$ . Forty elements were used in all cases; each plot denotes a different polynomial order. The polynomial order runs from first-order  $P=1$  to ninth-order  $P=9$  in row-major order. The ordinate of each plot is the complex imaginary axis, and the abscissa is the complex real axis. Note that the axes scales are only consistent across rows due to the large magnitude variation in the spectra due to polynomial order.

LDG eigenvalues; this clustering property makes LDG a prime candidate for preconditioning techniques.

In Table VI we present the maximum absolute eigenvalue ( $\max |\lambda_i|$ ) for different element number and polynomial order combinations. In Fig. 8, we present a graph of maximum absolute eigenvalue vs. polynomial order for a 40 evenly spaced element mesh (squares denote LDG). The increase in the maximum absolute eigenvalue is of order  $P^4$  where  $P$  is the order of the polynomial approximation used. This coincides with the commonly used  $1/P^4$  estimate for the diffusion number when using spectral methods for solving parabolic problems.

We observe that the maximum absolute value of LDG is about three times that of BR for comparative element number and polynomial order. This is consistent with the observations made in [3]. This implies that when

**Table VI.** LDG Maximum Absolute Eigenvalue Study: Maximum Absolute Eigenvalue of the Discrete Operator  $A_{LDG}$  Approximating the Second-order Spatial Derivative Operator for Different Number of Elements and Polynomial Order per Element.

Polynomial order	$N = 10$	$N = 20$	$N = 40$	$N = 80$	$N = 160$
1	2.7392e+01	5.4785e+01	1.0957e+02	2.1914e+02	4.3828e+02
2	8.8096e+01	1.7619e+02	3.5239e+02	7.0477e+02	1.4095e+03
3	1.9486e+02	3.8971e+02	7.7942e+02	1.5588e+03	3.1177e+03
4	3.7235e+02	7.4470e+02	1.4894e+03	2.9788e+03	5.9576e+03
5	6.2813e+02	1.2563e+03	2.5125e+03	5.0250e+03	1.0050e+04
6	9.8623e+02	1.9725e+03	3.9449e+03	7.8898e+03	1.5780e+04
7	1.4545e+03	2.9090e+03	5.8180e+03	1.1636e+04	2.3272e+04
8	2.0568e+03	4.1136e+03	8.2272e+03	1.6454e+04	3.2909e+04
9	2.8011e+03	5.6022e+03	1.1204e+04	2.2409e+04	4.4818e+04
10	3.7112e+03	7.4224e+03	1.4845e+04	2.9690e+04	5.9379e+04
11	4.7951e+03	9.5902e+03	1.9180e+04	3.8361e+04	7.6722e+04
12	6.0766e+03	1.2153e+04	2.4306e+04	4.8613e+04	9.7225e+04
13	7.5636e+03	1.5127e+04	3.0255e+04	6.0509e+04	1.2102e+05
14	9.2801e+03	1.8560e+04	3.7120e+04	7.4240e+04	1.4848e+05
15	1.1234e+04	2.2468e+04	4.4935e+04	8.9871e+04	1.7974e+05
16	1.3449e+04	2.6898e+04	5.3795e+04	1.0759e+05	2.1518e+05

Evenly spaced elements were used in all cases.

using an explicit time stepping scheme with the same elemental and polynomial discretization, LDG will require a time step approximately three times smaller than BR for stability.

### 4.3. Conditioning

As mentioned earlier, when solving our model problem implicitly, we are interested in inverting the operator  $L_{CN}$  as described above when formed using the spatial operator  $A_{LDG}$ . In Table VII we examine the  $L_2$  condition number of the matrix  $L_{CN}$  before and after diagonal preconditioning (denoted by multiplying by a matrix  $Z$  which consists of the inverse diagonal operator). For this experiment, a 40 evenly spaced element discretization using a time step of  $\Delta t = 10^{-5}$  was used.

As the polynomial order is increased, the condition number of the system increases (as expected). As in the BR case, the condition number of  $L_{CN}$  appears to grow linearly with the polynomial order. The CN scheme applied to this system produces a system which is diagonally dominant, and hence diagonal preconditioning works well. One observation, however, is that the condition number of the preconditioned LDG system is not as low as the conditioned number for the preconditioned BR system. This may be

**Table VII.** Condition Number Comparison Before and After Diagonal Preconditioning for the Linear Operator  $\mathbf{L}_{CN}$  Formed using LDG.

Polynomial order	$\kappa_2(\mathbf{L}_{CN})$	$\kappa_2(\mathbf{ZL}_{CN})$
1	3.0024	1.0056
2	4.9772	1.0308
3	6.8769	1.0918
4	8.6917	1.2362
5	10.4851	1.4847
6	12.3185	1.9161

A mesh consisting of 40 evenly spaced elements and a time step of  $\Delta t = 10^{-5}$  was used.

attributed to the tighter LDG stencil. Because the LDG stencil is tighter, LDG is less diagonally dominant (in the sense of monitoring the ratio of the absolute row sums over the diagonal element) than BR, and hence diagonal preconditioning is less effective than in the BR case. However, the new system, which is symmetric and has a condition number near one, is also a prime candidate for using conjugate gradient methods. Numerical experiments found that the number of iterations necessary to solve the preconditioned system was at least an order of magnitude lower than the rank of the original system, however the number of iterations is greater than or equal to the number of iterations needed for the BR system.

It is interesting to note that when less stringent time stepping errors are required, and hence larger time steps are used, diagonal preconditioning still has a greater relative effect on the BR system compared to the LDG system.

## 5. FORMULATION 4: BAUMANN–ODEN FLUX CHOICE

The consistent scheme that we examine is given by a modification of formulation 1 to make it consistent. We seek to find  $u, q \in V_P$  such that, for all test functions  $v, w \in V_P$ ,

$$\int_{I_j} u_t v dx + \int_{I_j} u_x v_x dx - (\hat{u}_x)_{j+\frac{1}{2}} v_{j+\frac{1}{2}}^- + (\hat{u}_x)_{j-\frac{1}{2}} v_{j-\frac{1}{2}}^+ - \frac{1}{2} (v_x)_{j+\frac{1}{2}}^- \left( u_{j+\frac{1}{2}}^+ - u_{j+\frac{1}{2}}^- \right) - \frac{1}{2} (v_x)_{j-\frac{1}{2}}^+ \left( u_{j-\frac{1}{2}}^+ - u_{j-\frac{1}{2}}^- \right) = 0,$$

where we take  $(\hat{u}_x)_{j+(1/2)} = (1/2)((\hat{u}_x^+)_{j+(1/2)} + (\hat{u}_x^-)_{j+(1/2)})$  as with the inconsistent scheme. The modification above yields a consistent scheme

**Table VIII.** BO Convergence Data:  $L_2$  Error Computed when Solving the Model Problem Evaluated at  $T=0.7$

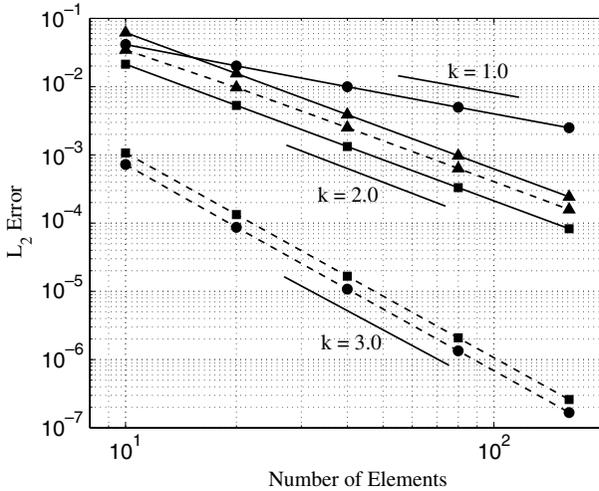
Polynomial order	$N = 10$	$N = 20$	$N = 40$	$N = 80$	$N = 160$
1	6.1733e-02	1.5530e-02	3.8852e-03	9.7141e-04	2.4286e-04
2	3.4457e-02	9.7002e-03	2.5055e-03	6.3165e-04	1.5824e-04
3	1.3137e-04	7.8184e-06	4.8267e-07	3.0076e-08	1.8786e-09
4	1.7944e-05	1.1723e-06	7.4127e-08	4.6490e-09	2.8931e-10
5	8.7873e-08	1.3167e-09	-	-	-
6	7.3241e-09	1.2006e-10	-	-	-

Evenly spaced elements were used in space; second-order CN with at time step of  $\Delta t = 10^{-5}$  was used in time. Entries denoted with ‘-’ represent cases where the spatial error is less then  $10^{-10}$ , and hence the time stepping error becomes the dominant error.

for all polynomial orders greater than or equal to one. The sacrifice that is made, however, is that the modification above yields a non-symmetric scheme, which will be evident when examining the eigenspectra. We will now proceed to examine the convergence rate, eigenspectra and system conditioning for this flux choice.

### 5.1. Convergence Properties

In Table VIII we present a convergence study using the BO flux. For this study, we examine five different numbers of evenly spaced elements (10, 20, 40, 80, 160) with polynomial orders varying systematically from  $P = 1$  to 6. For this test, the model problem was solved up to time  $T = 0.7$  using the second-order CN scheme with a time step of  $\Delta t = 10^{-5}$ . In Table II we present the error defined as the  $L_2$  difference between the approximate and exact solution. In the table, the symbol ‘-’ denotes when the error due to the spatial discretization is less than  $10^{-10}$ , and hence the time error becomes the dominant error. As was shown in [15,3], the order of accuracy is  $P + 1$  when the polynomial order is *odd* (optimal) and  $P$  when the polynomial order is *even* (sub-optimal). In Fig. 6 we examine the  $h$ -convergence of the method (denoted with triangles) for two different polynomial orders,  $P = 1$  (solid line) and  $P = 2$  (dashed line). In Fig. 4, we examine the  $p$ -convergence of the method (denoted with triangles) when 40 evenly spaced elements are used. The method exhibits a stair-case convergence as the polynomial order is increased, consistent with the optimal and sub-optimal estimates mentioned above. With respect to the optimal parity (odd), the scheme exhibits exponential convergence. Comparative statements between the methods will be made later in the paper.



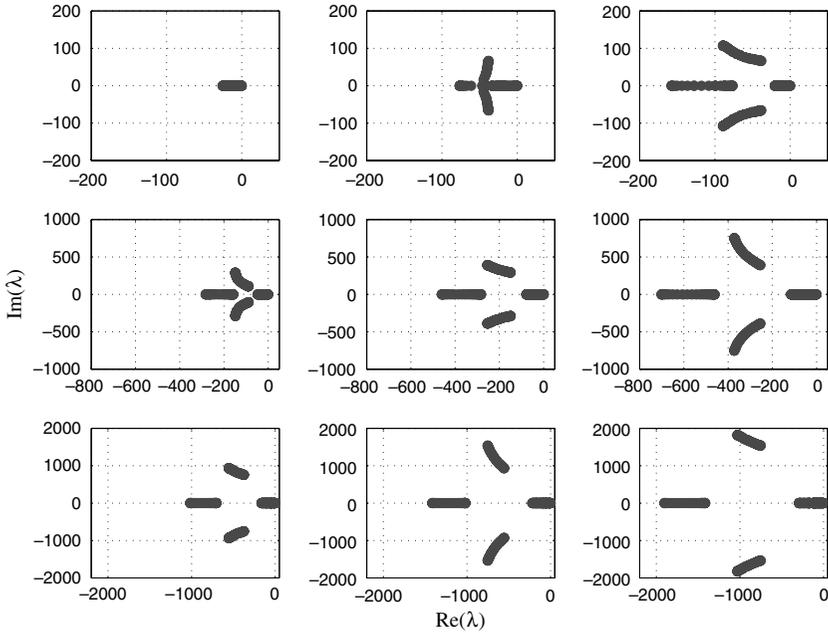
**Fig. 6.**  $h$ -Convergence study comparison of BR (circles), LDG (squares) and Baumann–Oden (triangles) based upon the model problem evaluated at  $T=0.7$ ; we present the  $L_2$  error vs. the number of evenly spaced elements when using polynomial order  $P=1$  (solid) and  $P=2$  (dashed).

Based upon the  $h$ -convergence results presented in Fig. 6, we observe that when the polynomial order is odd ( $P=1$  for the experiment in the figure), LDG provides the best convergence properties followed by Baumann–Oden and Bassi–Rebay (in descending order). We observe that when the polynomial order is even ( $P=2$  for the experiment in the figure), BR provides the best convergence properties followed by LDG and BO (in descending order). These observations are consistent with the convergence studies accomplished in [3]. Observe that with respect to  $p$ -convergence, BR and LDG provide nearly identical convergence results, both which are better than BO.

## 5.2. Eigenspectra

In Fig. 7 we present the eigenspectra of the spatial operator  $\mathbf{A}_{BO}$  formed using BO fluxes. The operator is a real but not symmetric, and hence admits the possibility of eigenvalues which are complex. We present eigenspectra diagrams for nine different polynomial orders running from first-order  $P=1$  to ninth-order  $P=9$  in row-major order.

Observe that the eigenspectra of this operator clearly demonstrate the non-symmetric nature of the operator. Complex eigenvalues denote the



**Fig. 7.** Eigenspectra of the spatial operator for BO  $A_{BO}$ . Forty elements were used in all cases; each plot denotes a different polynomial order. The polynomial order runs from first-order  $P=1$  to ninth-order  $P=9$  in row-major order. The ordinate of each plot is the complex imaginary axis, and the abscissa is the complex real axis. Note that the axes scales are only consistent across rows due to the large magnitude variation in the spectra due to polynomial order.

dispersive properties of the modification made to the inconsistent scheme. The other observation which can be made is that when solving the BO scheme explicitly, special care must be taken to use a time stepping scheme whose region of convergence contains a sufficient amount of the complex half-plane to encompass the dispersive eigenvalues.

In Table IX we present the maximum absolute eigenvalue ( $\max |\lambda_i|$ ) for different element number and polynomial order combinations. In Fig. 8, we present a graph of maximum absolute eigenvalue vs. polynomial order for a 40 evenly spaced element mesh (triangles denote BO). We observe that the maximum absolute value of LDG is about five times that of BO for comparative element number and polynomial order.

In Fig. 8 we compare the maximum absolute eigenvalue vs. polynomial order for the three consistent schemes. A 40 evenly spaced elemental mesh was used. As one would expect, all three flux choices exhibit  $\mathcal{O}(P^4)$  growth. Observe that LDG has the largest absolute eigenvalue, implying that LDG

**Table IX.** BO Maximum Absolute Eigenvalue Study: Maximum Absolute Eigenvalue of the Discrete Operator  $\mathbf{A}_{\text{BO}}$  Approximating the Second-order Spatial Derivative Operator for Different Number of Elements and Polynomial Order per Element.

Polynomial order	$N = 10$	$N = 20$	$N = 40$	$N = 80$	$N = 160$
1	6.3662e+00	1.2732e+01	2.5465e+01	5.0930e+01	1.0186e+02
2	1.9099e+01	3.8197e+01	7.6394e+01	1.5279e+02	3.0558e+02
3	3.9423e+01	7.8846e+01	1.5769e+02	3.1539e+02	6.3077e+02
4	8.1525e+01	1.6305e+02	3.2610e+02	6.5220e+02	1.3044e+03
5	1.1647e+02	2.3294e+02	4.6587e+02	9.3175e+02	1.8635e+03
6	2.0980e+02	4.1959e+02	8.3918e+02	1.6784e+03	3.3567e+03
7	2.7169e+02	5.4339e+02	1.0868e+03	2.1735e+03	4.3471e+03
8	4.2726e+02	8.5452e+02	1.7090e+03	3.4181e+03	6.8361e+03
9	5.2385e+02	1.0477e+03	2.0954e+03	4.1908e+03	8.3816e+03
10	7.5720e+02	1.5144e+03	3.0288e+03	6.0576e+03	1.2115e+04
11	8.9621e+02	1.7924e+03	3.5849e+03	7.1697e+03	1.4339e+04
12	1.2229e+03	2.4458e+03	4.8917e+03	9.7833e+03	1.9567e+04
13	1.4121e+03	2.8241e+03	5.6483e+03	1.1297e+04	2.2593e+04
14	1.8477e+03	3.6953e+03	7.3907e+03	1.4781e+04	2.9563e+04
15	2.0947e+03	4.1893e+03	8.3787e+03	1.6757e+04	3.3515e+04
16	2.6547e+03	5.3095e+03	1.0619e+04	2.1238e+04	4.2476e+04

Evenly spaced elements were used in all cases.

will be the most restrictive when applying an explicit time stepping algorithm. BR is less restrictive than LDG (as stated previously, about three times less restrictive), and BO is the least restrictive based upon maximum eigenvalue magnitude. For BO, however, we must remember that the explicit time stepping scheme must contain a large region of the complex half-plan to encompass the dispersive eigenvalues of the BO spatial operator.

### 5.3. Conditioning

As mentioned earlier, when solving our model problem implicitly, we are interested in inverting the operator  $\mathbf{L}_{\text{CN}}$  as described above when formed using the spatial operator  $\mathbf{A}_{\text{BO}}$ . In Table X we examine the  $L_2$  condition number of the matrix  $\mathbf{L}_{\text{CN}}$  before and after diagonal preconditioning (denoted by multiplying by a matrix  $\mathbf{Z}$  which consists of the inverse diagonal operator). For this experiment, a 40 evenly spaced element discretization using a time step of  $\Delta t = 10^{-5}$  was used.

As in the BR and LDG cases, the condition number of  $\mathbf{L}_{\text{CN}}$  appears to grow linearly with the polynomial order. Observe that diagonal preconditioning modifies this system significantly also. The caveat, however, is that the BO system is not symmetric, and hence conjugate gradient methods

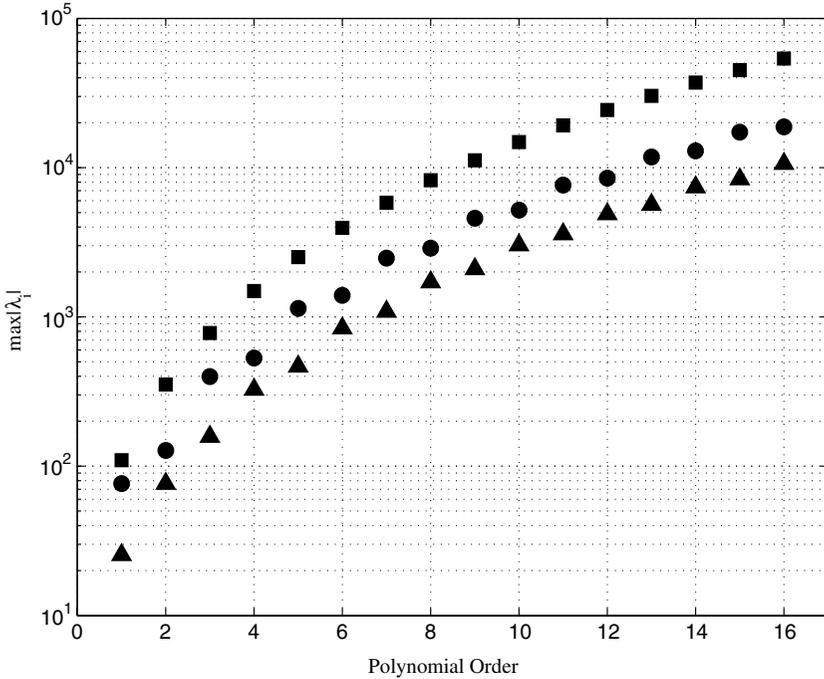


Fig. 8. Maximum absolute eigenvalue  $\max_i |\lambda_i|$  vs. polynomial order for BR (circles), LDG (squares) and BO (triangles). A mesh consisting of 40 evenly spaced elements was used.

cannot be applied; one must resort to methods such as generalized residual methods (e.g., GMRES).

### 6. STABILIZATION

For the three consistent flux choices presented above, stabilization factors are sometimes added when solving elliptic problems. In the case of solving purely elliptic problems, these stabilization factors quite often help to guarantee that the null space of the discrete operator is trivial or modify the scheme so that optimal convergence rates can be achieved [2]. For instance, in the case of discretizing the model parabolic problem, only the constant function should exist in the discrete null space of the spatial operator. One form of the stabilization factor commonly used is the term  $-\eta_e h_e [[u_h]]$ , which is appended to the  $\hat{\sigma}_K$  flux. The term  $\eta_e$  is basically a penalization factor taken to be greater than or equal to zero,  $h_e$  is related to the length of the edge on which the penalization is to occur

**Table X.** Condition Number Comparison Before and After Diagonal Preconditioning for the Linear Operator  $\mathbf{L}_{\text{CN}}$  Formed Using BO.

Polynomial order	$\kappa_2(\mathbf{L}_{\text{CN}})$	$\kappa_2(\mathbf{ZL}_{\text{CN}})$
1	2.9927	1.0008
2	4.9400	1.0096
3	6.7697	1.0237
4	8.3936	1.0823
5	9.7528	1.1423
6	10.9273	1.3039

A mesh consisting of 40 evenly spaced elements and a time step of  $\Delta t = 10^{-5}$  was used.

(and in the one-dimensional case is taken to be one), and  $[[u_h]]$  is a measure of the jump in the solution [2]. For LDG with  $\beta = 0$  (which in the absence of stabilization reduces to the original BR scheme), the inclusion of this term implies that  $\hat{\sigma}_K = \{\sigma_h\} - \eta_e h_e [[u_h]]$ , for LDG (in general)  $\hat{\sigma}_K = \{\sigma_h\} + \beta \cdot [[\sigma_h]] - \eta_e h_e [[u_h]]$  and for BO  $\hat{\sigma}_K = \{\nabla_h u_h\} - \eta_e h_e [[u_h]]$  (a new variation on BO stabilization has recently been presented in [14], but will not be discussed here). The addition of this elementary stabilization is designed to be consistent with the LDG stabilization factor found in [2], and is similar to adding an additional penalty term [12] which penalizes jumps in the solution. The larger  $\eta_e$  is chosen to be, the more penalized the method; asymptotically the scheme becomes a  $C^0$  method because the stabilization factor more strongly enforces continuity across element interfaces. Several other stabilization options have been proposed and studied in the literature, for instance: “stabilized” BR [6], variants of the non-symmetric interior penalty Galerkin (NIPG) method [13], and the aforementioned penalization in terms of jumps in derivatives [14]. None of these will be considered in this paper, although similar tests could be accomplished to understand the influence of the penalty parameters.

For parabolic problems, two natural questions are: why would stabilization be necessary, and what is the effect of stabilization? To attempt to understand the first of these questions, we attempted to quantify the size of the discrete null space of the discretized operators formed using BR, LDG and BO. To accomplish this task, we examined carefully the eigenvalues of the discrete operator  $\mathbf{A}$  which is the DG approximation of the second-order derivative operator on a periodic interval. The continuous operator, in this case, has only the constant function in its null space. We would desire that this also be true of the discrete operator. After ordering the eigenvalues, we declared the size of the discrete null space to be the number of eigenvalues

that, in absolute magnitude, are less than  $10^{-13}$ . We expect that only one such eigenvalue exists for the discrete operators. In Table XI, we present the size of the null space for the three different formulations. We compute for two different evenly spaced element numbers ( $N = 10$  and  $11$ ) and for polynomial orders  $P = 1$  to  $10$ .

Observe that LDG exhibits exactly what we expect; upon examination, only the constant solution is in the null space. BO exhibits what we expect except for one case:  $N = 10$  with  $P = 1$ . It is discussed in [2] that for  $P < 2$  such problems may exist. More importantly, however, is that (as predicted in [2]) the BR operator has a null space which contains, under certain circumstances, more than a constant mode. This study shows that the size of the discrete null space does not grow above two with polynomial order, and apparently the size is effected by a combination of the parity of the element number and polynomial order. In Fig. 9 we plot as an example the non-constant function within the discrete null space for BR on ten evenly spaced elements with sixth order polynomials.

The concern which arises for BR is that, when combined with non-linear advection (such as in the Navier–Stokes equations), BR may, in some instances, leave some solutions untouched with respect to dissipation. Consistent with [2], we affirmed numerically that stabilization can be added to BR which reduces the null space to contain only the constant mode.

To understand the effect of stabilization, we examined the eigenspectra of the new operator formed by stabilization. In Fig. 10, we present on the left the eigenspectra of the LDG  $\beta = 0$  (i.e., the reduction to BR)

**Table XI.** Numerical Evaluation of the Dimension of the Null Space ( $\lambda_i \leq 1 \times 10^{-13}$ ) for Different Polynomial Order Expansions  $P$  when Partitioning the Domain into Evenly Spaced Elements

Polynomial order	BR: $N = 10$	BR: $N = 11$	LDG: $N = 10$	LDG: $N = 11$	BO: $N = 10$	BO: $N = 11$
1	2	2	1	1	2	1
2	2	1	1	1	1	1
3	2	2	1	1	1	1
4	2	1	1	1	1	1
5	2	2	1	1	1	1
6	2	1	1	1	1	1
7	2	2	1	1	1	1
8	2	1	1	1	1	1
9	2	2	1	1	1	1
10	2	1	1	1	1	1

All three schemes are presented; ' $N$ ' denotes the number of elements used.

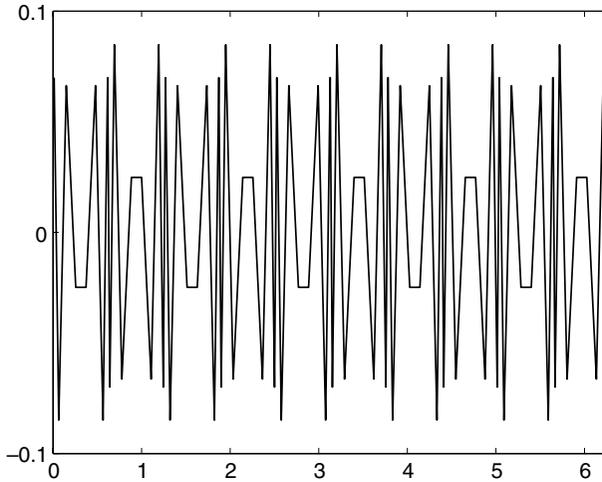


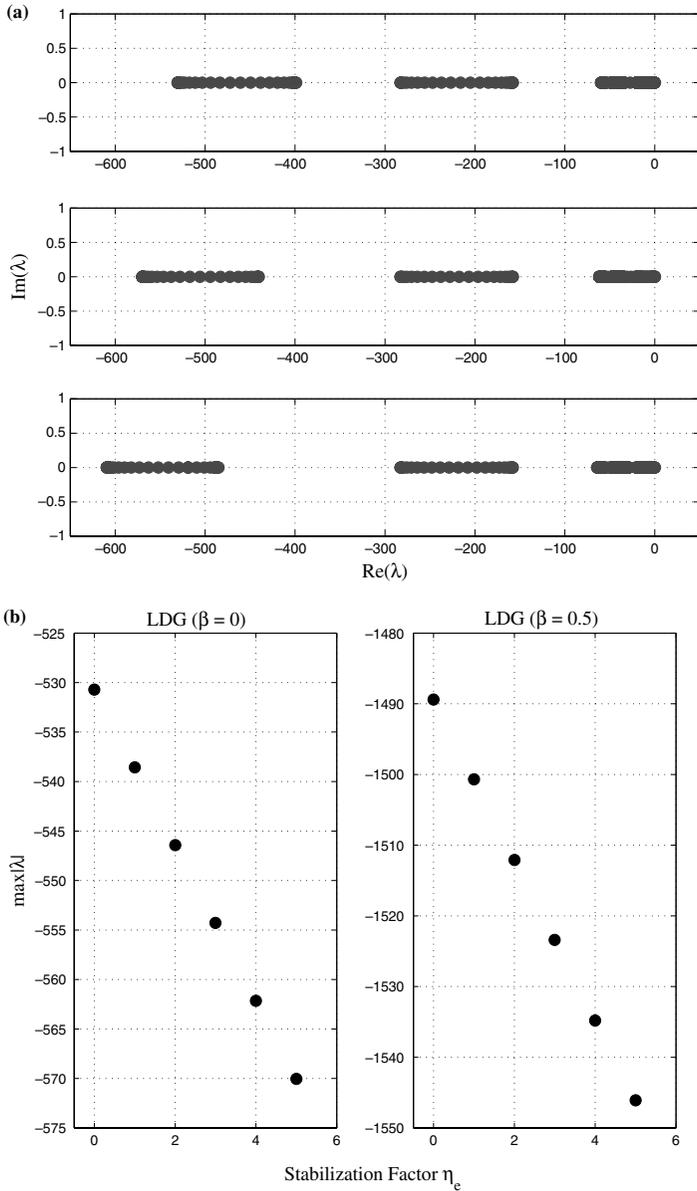
Fig. 9. Plot of the non-constant function which exists in the null space of the classic (unstable) BR discrete operator. Ten evenly-spaced elements with sixth order polynomials were used.

operator when the elementary stabilization factor described above is added. The three plots denote the eigenspectra when the stabilization factor  $\eta_e$  is taken to be zero, five and ten from top to bottom, respectively. On the right we present the maximum absolute magnitude of the eigenspectra for both LDG  $\beta = 0$  and LDG  $\beta = 0.5$  when a 40 element discretization using 4th order polynomials were employed.

Figure 10 shows that the effect of the stabilization factor is to move the eigenvalues to the left. More specifically, the stabilization factor makes the scheme more dissipative (which is what one would expect of a stabilization factor). In terms of the schemes that we are examining, the major ramification of this movement of the eigenvalues is the further restriction on the time step which moving the eigenvalues incurs. This behavior is consistent with the observations of [12]; increasing the stabilization penalty parameter more strongly enforces continuity at the sacrifice of a more stringent time step restriction.

## 7. SUMMARY

In this paper we have sought to provide the *pros* and *cons* of different flux choices when solving diffusion problems using the DG method through an investigation of a model one-dimensional problem. We began by examining an “inconsistent” scheme, and then proceeded to examine three



**Fig. 10.** On the left we present the eigenspectra of the spatial operator for LDG  $\beta=0$  (BR ABR) with stabilization added. Forty elements with fourth-order polynomials were used in all cases. Three different choices of the stabilization factor are shown:  $\eta_e = 0$  (left-top),  $\eta_e = 5$  (left-center) and  $\eta_e = 10$  (left-bottom). On the right we present the maximum eigenvalue in magnitude vs. the stabilization parameter  $\eta_e$  for LDG  $\beta=0$  and LDG  $\beta=0.5$ .

commonly used flux choices: BR, LDG and BO. In particular, we provided numerical evaluations of the  $h$ -convergence rate, the  $p$ -convergence rate, the eigenspectra and the system conditions. From our examination, the following observations can be made:

- For the one-dimensional system considered, the LDG (with  $\beta = 0.5$ ) and BO schemes produce tighter elemental stencils than BR. In the case of parallel computation, this implies that LDG and BO require less communication than BR. A similar result for two-dimensions was discussed in [8].
- LDG has optimal  $h$ -convergence independent of the polynomial order. Both BR and BO can observe suboptimal convergence depending on the parity of the polynomial order.
- When solving the model problem with an explicit time-stepping method, LDG requires a smaller time step. This is observed by examining the spectra of the operator.
- For the cases considered, diagonal preconditioning works better for BR than LDG. Both BR and LDG benefit from diagonal preconditioning, and since they are symmetric, both BR and LDG can use conjugate gradient methods.
- For the cases considered, diagonal preconditioning works well for BO. The trade-off is that BO is not a symmetric system, and hence conjugate gradient methods cannot be employed. Rather, generalized residual methods (e.g., GMRES) must be employed.
- Stabilization factors move the eigenvalues to the left on the stability diagram, and hence decrease the time step when using an explicit method.

Examination of the one-dimensional model problem presented herein provides some insight into how to make appropriate flux choices when solving diffusion problems with the DG method. Further examinations of the type presented in this paper for two- and three-dimensional spatial discretizations will be accomplished and presented in the future.

## ACKNOWLEDGMENTS

We would like to thank Professors Chi-Wang Shu and Jan Hesthaven of Brown University and Professor Bernardo Cockburn of University of Minnesota for their helpful comments. We gratefully acknowledges the support of this work by the Air Force Office of Scientific Research (Computational Mathematics Program) under Grant number F49620-01-1-0035.

## REFERENCES

1. Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, D. (2000). Discontinuous Galerkin Methods for Elliptic Problems. In Cockburn, B., Karniadakis, G.E., and Shu, C.-W. (eds.), *Discontinuous Galerkin Methods: Theory Computation and Applications*, Springer, Berlin.
2. Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, L. D. (2002). Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.* **39**, 1749.
3. Atkins, H., and Shu, C.-W. (1999). Analysis of the discontinuous Galerkin method applied to the diffusion operator. In *14th AIAA Computational Fluid Dynamics Conference AIAA*, pp. 99–3306.
4. Bassi, F., and Rebay, S. (1997). A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. *J. Comp. Phys.* **131**, 267.
5. Baumann, C. E., and Oden, J. T. (1999). A discontinuous hp finite element method for convection-diffusion problems. *Comp. Meth. Appl. Mech. Eng.* **175**, 311–341.
6. Brezzi, F., Manzini, G., Marini, D., Pietra, P., and Russo, A. (1999). Discontinuous finite elements for diffusion problems. *Atti Convegno in onore di F. Brioschi (Milano 1997) Istituto Lombardo Accademia di Scienze e Lettere*, pp. 197–217.
7. Canuto, C., Hussaini, M. Y., Quarteroni, A., and Zang, T. A. (1987). *Spectral Methods in Fluid Mechanics*, Springer-Verlag, New York.
8. Castillo, P. (2002). Performance of discontinuous Galerkin Methods for elliptic problems. *SIAM J. Numer. Anal.* **24**(2), 524–547.
9. Cockburn, B., and Shu, C.-W. (1998). The local discontinuous Galerkin for convection-diffusion systems. *SIAM J. Numer. Anal.* **35**, 2440–2463.
10. Cockburn, B., and Karniadakis, G. E., and Shu, C.-W. (2000). The development of discontinuous Galerkin methods. In *Discontinuous Galerkin Methods: Theory Computation and Applications*, Cockburn, B., Karniadakis, G.E., and Shu, C.-W. (eds.), Springer, Berlin.
11. Helenbrook, B., Mavriplis, D., and Atkins, H. (2003). Analysis of p-multigrid for continuous and discontinuous finite element discretizations. In *16th AIAA Computational Fluid Dynamics Conference AIAA*, pp. 99–3989.
12. Hesthaven, J. S., and Gottlieb, D. (1996). A stable penalty method for the compressible Navier–Stokes Equations. I. Open boundary conditions. *SIAM J. Sci. Comp.* **17**(3), 579–612.
13. Rivière, B., Wheeler, M. F., and Girault, V. (2001). A priori error estimates for finite element methods based on discontinuous approximation spaces for elliptic problems. *SIAM J. Numer. Anal.* **39**(3), 902–931.
14. Romkes, A., Prudhomme, S., and Tinsley Oden, J. (2003). A posteriori error estimation for a new stabilized discontinuous Galerkin method. *Appl. Math. Lett.* **16**(4), 447–452.
15. Shu, C. -W. (2001). Different formulations of the discontinuous Galerkin method for the viscous terms. In Shi, Z.-C., Mu, M., Xue, W., and Zou, J. (eds.), *Advances in Scientific Computing*, Science Press, mascou, pp. 144–155.
16. Zhang, M., and Shu, C.-W. (2003). An analysis of three different formulations of the discontinuous Galerkin method for diffusion equations. *Math. Models Meth. Appl. Sci. (M<sup>3</sup>AS)* **13**, 395–413.