IMPLICIT A-POSTERIORI BOUNDS FOR DISCONTINUOUS GALERKIN METHODS

J.S. Wong* and J. Peraire†
Department of Aeronautics and Astronautics,
Massachusetts Institute of Technology, Cambridge, MA 02139

Abstract

We present an implicit a-posteriori method for the computation of upper and lower bounds for functional outputs of first-order systems of conservation laws approximated with the discontinuous Galerkin finite element method. The computed bounds can be shown to be strict with respect to the solution obtained on a conservatively refined discretization, whereas the cost and accuracy of these bounds is mostly determined by the size of a coarse grid (working) discretization. The proposed method is illustrated for linear and non-linear scalar equations as well as the two dimensional Euler equations of gas dynamics.

Introduction

Over the past few years we have develop a methodology for the computation of upper and lower bounds for functional outputs of partial differential equations.[5–7, 11, 15] Our methodology is based on a reformulation of the output of interest as a constrained minimization problem in which the objective function is convex, and the governing equations enter the problem as equality constraints. Computational advantage is obtained by decomposing, or hybridizing, the domain and imposing inter-domain continuity through additional constraints. Lower bounds for the output of interest are obtained by exploiting weak duality and relaxing the continuity and governing equation constraints. Approximate lagrange multipliers are obtained using an inexpensive coarse grid calculation. The final algorithm involves a global grid computation to evaluate approximations to the multipliers, and a series of local reduced size computations for each sub-domain, that can be trivially parallelized. The result is a strict lower bound for the solution that would be computed, at a much higher cost, using an arbitrarily refined fine mesh. Upper bounds can be computed by repeating the process and computing lower bounds for the negative of the output of interest. The bound gap, i.e. the difference between the upper and lower bounds, can be decomposed into a sum of positive contributions from each subdomain, which naturally leads to a mesh adaption indicator. Our strategy has been applied successfully to linear coercive problems including convection-diffusion, Stokes flows, linear stabilized methods for hyperbolic problems, and more recently, it has been extended to non-linear problems in large deformation elasticity, where the equilibrium equations are still derived from a variational principle.[12]

In this paper, we extend beyond our existing capability in a number of ways. First, we extend our methodology to discontinuous Galerkin approximations (DG). It turns out that in the discontinuous Galerkin approximation the solution is naturally discontinuous across boundaries, whereas the flux is continuous. The hybridization process is accomplished by introducing ghost nodes surrounding each sub-domain and constraining the fluxes in neighboring elements to be continuous. Second, the output of interest, is convexified using the natural "energy" of discontinuous Galerkin approximations, which involves the weighted squares of the inter-element jumps. Finally, the proposed procedure is applied to test problems involving the linear convection equation, a nonlinear scalar equation and the Euler equations in two dimensions.

Problem Statement

In order to present our method, we start from a general two dimensional non-linear scalar conservation law defined over domain Ω with boundary Γ

$$f_1(u)_{,1} + f_2(u)_{,2} + \sigma u = r \quad \text{in} \quad \Omega \qquad (1)$$

$$a^- u = a^- g \quad \text{on} \quad \Gamma \qquad (2)$$

In the above expression, $u(\mathbf{x}, t)$ is the unknown function, $f_i(u)$ is the flux in the i^{th} coordinate direc-

*Graduate Research Assistant
†Professor, Associate Fellow, AIAA

tion, σ is the source term coefficient and $r(\mathbf{x})$, is the non-homogeneous forcing function. The comma notation is used to denote partial differentiation (e.g. $u_{,j} = \partial u/\partial x_j$, denotes the partial derivative with respect to the j-th spatial coordinate). Also, $a^- = \frac{1}{2}(\mathbf{a}\cdot\mathbf{n} - \mid \mathbf{a}\cdot\mathbf{n} \mid)$, where $\mathbf{a} = [a_1, a_2]^T = [\partial f_1/\partial u, \partial f_2/\partial u]^T$, and $\mathbf{n} = [n_1, n_2]^T$, is the domain outward unit normal. The boundary data, g, is imposed at inflow (i.e. $\mathbf{a}\cdot\mathbf{n} < 0$). Note that to ensure well-posedness of the problem we assume that $\sigma \geq 0$.

Discontinuous Galerkin Formulation

We consider a discretization, $\mathcal{T}_H$, of the domain Ω consisting of N_H non-overlapping quadrilateral elements, T_H, with elemental boundaries γ_H such that $\Omega = \bigcup T_H$. We also introduce, for later use, the sets $\varepsilon(T_H)$ and $\varepsilon(\mathcal{T}_H)$, denoting the set of boundary edges of a typical element T_H, and the set of all interior boundaries in the discretization $\mathcal{T}_H$, respectively. Associated with this discretization, we introduce the space of discontinuous piecewise bilinear functions,

$$X_H = \{v | v \in (L_2(\Omega))^2, v|_{T_H} \in (\mathcal{Q}_1(T_H))^2, \forall T_H \in \mathcal{T}_H\} \ . \quad (3)$$

For an individual element, T_H, in $\mathcal{T}_H$, we can write, after multiplying (1) by a test function v and integrating by parts, the following expression

$$\sum_{\gamma_H \in \varepsilon(T_H)} \int_{\gamma_H \setminus \Gamma} v h_n(u^+, u^-)\, ds + \sum_{\gamma_H \in \varepsilon(T_H)} \int_{\gamma_H \cap \Gamma} v h_n(u, g)\, ds - \int_{T_H} (v_{,1} f_1 + v_{,2} f_2)\, dA + \int_{T_H} \sigma v\, dA = \int_{T_H} r v\, dA \quad (4)$$

or, in shorthand notation,

$$B(v,u)_{T_H} = L(v)_{T_H} \ . \quad (5)$$

Here, h_n, is the numerical flux function

$$h_n(v^+, v^-) = \frac{1}{2}(f(v^+;\mathbf{n}) + f(v^-;\mathbf{n})) - \frac{1}{2} h_n^d \ , \quad (6)$$

where

$$h_n^d = \mid \mathbf{a}(v^+, v^-;\mathbf{n}) \mid [v]_{\mathbf{x}^-}^{\mathbf{x}^+},$$

$f(v;\mathbf{n}) = n_1 f_1 + n_2 f_2$, and $\mathbf{a}(v^+, v^-;\mathbf{n})$ denotes an approximation to $\mathbf{a}\cdot\mathbf{n}$ evaluated at the interface between the two states $v^+ = v(\mathbf{x}^+)$ and $v^- = v(\mathbf{x}^-)$. The precise form of this approximation will be given below.

The discontinuous Galerkin method is defined by summing equation (5) over all elements and seeking a solution $u_H \in X_H$, such that

$$B(v_H, u_H) = L(v_H), \quad \forall v_H \in X_H \ . \quad (7)$$

Here, $B : X_H \times X_H \mapsto I\!R$, and $L : X_H \mapsto I\!R$, are given as

$$B(v,u) = \sum_{T_H \in \mathcal{T}_H} B(v,u)_{T_H}, \quad (8)$$

$$L(v) = \sum_{T_H \in \mathcal{T}_H} L(v)_{T_H} \ . \quad (9)$$

Finally, note that no restriction on the space X_H is imposed at inflow, and that the appropriate boundary conditions are treated weakly through the boundary flux.

Energy Stability of DG Discretization

The Discontinuous Galerkin finite element discretization for first-order hyperbolic conservation laws derives its stability from the upwinded nature of the numerical flux, through which numerical dissipation is applied. The symmetric part of $B(v,u)$, $B^s(v,v)$, be can be expressed as[1]

$$\begin{aligned} B(v,v) &= B^s(v,v) \\ &= \sum_{\gamma_H \in \varepsilon(\mathcal{T}_H)} B^s(v,v)_{\gamma_H} + \sum_{T_H \in \mathcal{T}_H} \int_{T_H} \sigma v^2\, dA \ , \end{aligned} \quad (10)$$

where,

$$B^s(v,v)_{\gamma_H} = \frac{1}{2}\int_{\gamma_H}\int_0^1 (1-2\theta)[v]_{\mathbf{x}^-}^{\mathbf{x}^+} \cdot \mathbf{a}(\bar{v}(\theta);\mathbf{n})[v]_{\mathbf{x}^-}^{\mathbf{x}^+}\, d\theta ds + \frac{1}{2}\int_{\gamma_H} [v]_{\mathbf{x}^-}^{\mathbf{x}^+} h^d\, ds \ , \quad (11)$$

and,

$$\bar{v}(\theta) = v(\mathbf{x}^-) + \theta [v]_{\mathbf{x}^-}^{\mathbf{x}^+} \ .$$

It is well known that for finite volume schemes, the interface fluxes need to be properly upwinded through the selection of a suitable Riemann solver in order to have a stable algorithm. The same is true for the DG discretization and it was shown in[1] that, for all v, $B(v,v)$ needs to satisfy

$$B(v,v) \geq B(v,v)_{SMV} \ , \quad (12)$$

for the numerical solution to possess the appropriate stability properties. Here, $B(v,v)_{SMV}$ is the operator (8) obtained by setting h_n^d in (6) equal to the Symmetric Mean Value flux, $h_{n\,SMV}^d$, which is given as,

$$h_{nSMV}^d = \int_0^1 (1-\theta) \mid \mathbf{a}(\bar{v}(\theta);\mathbf{n}) \mid [v]_{\mathbf{x}^-}^{\mathbf{x}^+} d\theta \ . \quad (13)$$

Bounds Formulation

Function Spaces

In addition to the working discretization, $\mathcal{T}_H$, already introduced, we consider an additional reference discretization, $\mathcal{T}_h$, which is obtained by uniformly subdividing the elements of $\mathcal{T}_H$. A typical element of $\mathcal{T}_h$, is denoted by T_h, and the set of interior element boundaries, γ_h, in $\mathcal{T}_h$, is denoted by $\varepsilon(\mathcal{T}_h)$. The discontinuos Galerkin approximation space associated with $\mathcal{T}_h$, is given as

$$X_h = \{v | v \in (L_2(\Omega))^2, v|_{T_h} \in (\mathcal{Q}_1(T_h))^2, \forall T_h \in \mathcal{T}_h\} \quad , \quad (14)$$

We also introduce the spaces, Q_H and Q_h, consisting of piecewise linear functions defined over the boundaries of the elements of $\mathcal{T}_H$ as

$$Q_H = \{q|_{\gamma_H} \in \mathcal{P}_1(\gamma_H), \forall \gamma_H \in \varepsilon(T_H)\} \quad (15)$$

$$Q_h = \{q|_{\gamma_h} \in \mathcal{P}_1(\gamma_h), \forall \gamma_h \in \varepsilon(\mathcal{T}_h) \cap \varepsilon(\mathcal{T}_H)\} \, . \quad (16)$$

Finally, we introduce the "broken" spaces $\hat{X}_H$ and $\hat{X}_h$, by augmenting the spaces X_H and X_h, with edge functions from Q_H and Q_h, respectively,

$$\hat{X}_H = \{\hat{v} = (v, v^{*+}, v^{*-}) | v \in X_H, v^{*+}, v^{*-} \in Q_H\} \quad (17)$$

$$\hat{X}_h = \{\hat{v} = (v, v^{*+}, v^{*-}) | v \in X_h, v^{*+}, v^{*-} \in Q_h\} \, . \quad (18)$$

That is, for each interior boundary in $\mathcal{T}_H$, we enrich the spaces X_H and X_h, by adding two functions which will be referred to as the "dummy" functions. Each of the two functions is associated to one of the two elements sharing that boundary. These functional spaces are illustrated in figure 1. It is clear from the above definitions that $X_H \subset X_h$, $\hat{X}_H \subset \hat{X}_h$ and $Q_H \subset Q_h$.

We also extend the forms $B(\cdot,\cdot)$ and $B^s(\cdot,\cdot)$, to operate on functions from the broken spaces; $B : \hat{X}_{H/h} \times X_{H/h} \mapsto I\!R$, and $B^s : \hat{X}_{H/h} \times \hat{X}_{H/h} \mapsto I\!R$. In this case, we shall employ different fluxes for the two elements sharing a coarse mesh interior boundary. That is, assuming that we have two elements denoted by left (L) and right (R), and that the unit normal for this edge points from left to right, the numerical flux to be used for the left element is $h_n^L(v^-, v^{*+})$, and the numerical flux to be used for the right element is $h_n^R(v^{*-}, v^+)$ (here the asterisk, *, denotes the "dummy" nodes surrounding each element in the broken mesh).

It is clear that the effect of introducing the "broken" spaces is that of decoupling the problem at the expense of making the solution indeterminate. In order remedy this situation, we introduce an operator which restores weak elemental coupling.

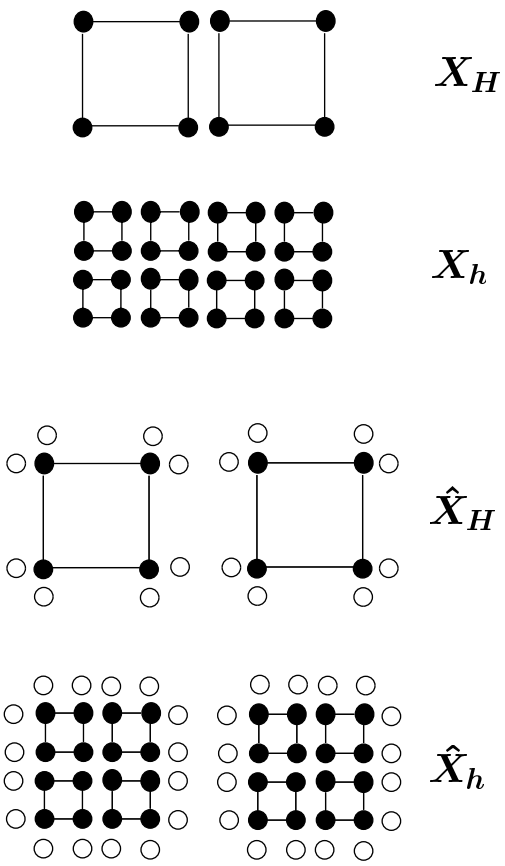

Fig. 1 Illustration of the functional spaces used in the bounds formulation. The figure shows two macro-elements (T_H), which have been subdivided into four elements each to produce the elements (T_h) of the reference discretization. The "broken" spaces are obtained by adding additional functions along the interior boundaries of the coarse coarse mesh elements. The black circles show the interpolation nodes for the functions in X_H and X_h, whereas the hollow circles show the interpolation nodes for the "dummy" edge functions.

Flux Continuity

Let the "jump" operator $b : \hat{X}_{H/h} \times Q_{H/h} \mapsto I\!R$, be defined as

$$b(\hat{v}, q) = \sum_{\gamma_H \in \varepsilon(\mathcal{T}_H)} \int_{\gamma_H} \mathcal{J}_{\gamma_H} h_n \, q|_{\gamma_H} \, ds \quad (19)$$

where $\mathcal{J}_{\gamma_H} h_n = h_n^R(v^-, v^{*+}) - h_n^L(v^{*-}, v^+)$, is the jump in h_n across γ_H. Given $\hat{v} \in \hat{X}_H$ (resp. $\hat{X}_h$), if,

$$b(\hat{v}, q) = 0, \qquad \forall q \in Q_H \text{ (resp. } Q_h) \, , \quad (20)$$

then, the numerical flux jumps at the macro-element interfaces will be zero, and as a consequence, we will have $h_n^R(v^-, v^{*+}) = h_n^L(v^{*-}, v^+) = h_n(v^-, v^+)$.

Energy Equality

The reference solution $u_h \in X_h$, satisfies

$$B(v_h, u_h) = L(v_h) \qquad \forall v_h \in X_h \ . \tag{21}$$

Setting $v_h = u_h$, we obtain

$$B^s(u_h, u_h) - L(u_h) = 0. \tag{22}$$

The symmetric form $B^s(\cdot,\cdot)$ is now decomposed into contributions corresponding to each of the elements in $\mathcal{T}_H$. Thus, one has,

$$B(v, v) = \sum_{T_H \in \mathcal{T}_H} B^s(v, v)_{T_H}, \tag{23}$$

where

$$B_{T_H}(v, v) = \int_{T_H} \sigma v^2 \, dA + \int_{\gamma_H \setminus \Gamma} \frac{1}{2} \beta h_n^d [v]_{\mathbf{x}^-}^{\mathbf{x}^+} \, ds$$
$$+ \int_{\gamma_H \cap \Gamma} (v h_n(v, g; \mathbf{n}) - F(v; \mathbf{n})) \, ds \tag{24}$$

with $F(v;\mathbf{n})$, any function satisfying $\partial F(v;\mathbf{n})/\partial v = f(v;\mathbf{n})$. Note that the role of the parameter β, is to assign the stabilizing contribution corresponding to the dissipative flux to the downstream element. Thus, β is chosen as,

$$\begin{aligned} \beta = 1, \quad \mathbf{a}\cdot\mathbf{n} \quad &< 0 \ , \\ \beta = 0, \quad \mathbf{a}\cdot\mathbf{n} \quad &\geq 0 \ . \end{aligned}$$

Lagrangian Formulation

We can now proceed with the bounding algorithm. Let us consider an output of interest $S \in I\!R$ which can be computed as a linear function of the solution, u, as $S = L^O(u)$. For the reference mesh solution u_h, we will have $S_h = L^O(u_h)$. Our goal is to compute inexpensive upper and lower bounds for S_h which do not require the computation of the reference solution u_h. We start by writing the following "trivial" identity,

$$S_h = \inf_{\hat{v}_h \in X_h} \sup_{\mu_h \in X_h, q_h \in Q_h} \mathcal{L}^-(\hat{v}_h, \mu_h, q_h) \ , \tag{25}$$

where the Lagrangian $\mathcal{L}^- : \hat{X}_h \times X_h \times Q_h \mapsto I\!R$, is defined as

$$\mathcal{L}^-(\hat{v}_h, \mu_h, q_h) = B^s(\hat{v}_h, \hat{v}_h) - L(\hat{v}_h)$$
$$+ L^O(\hat{v}_h) + B(\mu_h, \hat{v}_h) - L(\mu_h) + b(\hat{v}_h, q_h) \ . \tag{26}$$

We note that $\sup_{\mu_h \in X_h, q_h \in Q_h} \mathcal{L}(\hat{v}_h, \mu_h, q_h)$ will be ∞ whenever $\hat{v}_h \neq u_h$. This is so because the lagrangian is linear in μ_h and q_h. Therefore the infimization over $\hat{v}_h$, will result in $\hat{v}_h = u_h$, and as a result $S_h = L^O(u_h)$. A lower bound, S_h^-, for S_h, may now be obtained if the space Q_h is replaced by Q_H (a subset of Q_h). Thus, we can write

$$S_h^- = \inf_{\hat{v}_h \in \hat{X}_h} \sup_{\mu_h \in X_h, q_H \in Q_H} \mathcal{L}^-(\hat{v}_h, \mu_h, q_H) \ . \tag{27}$$

Note that this is a globally coupled problem, but due to the much weaker coupling between macro-elements it can be potentially solved at a much lower cost than problem (25). We shall see that, under certain circumstances, further reduction in computational cost may be achieved by freezing the multipliers and setting, $q_h = \bar{q}_h$ for any $\bar{q}_h \in Q_h$, thus obtaining

$$S_h^- = \inf_{\hat{v}_h \in \hat{X}_h} \sup_{\mu_h \in X_h} \mathcal{L}^-(\hat{v}_h, \mu_h, \bar{q}_h) \ . \tag{28}$$

In this way, one achieves complete decoupling and therefore the constrained minimization problem can be performed independently over each macro-element. For given multipliers $\bar{q}_h$, the resulting algorithm has a cost that is linear on the size of $\mathcal{T}_h$. Although the above inequality holds for any $\bar{q}_h$, the accuracy of the lower bound will depend on how close $\bar{q}_h$ is to the exact value of the multiplier q_h, p_h, at the saddle point. In practice one would perform a coarse grid computation to obtain p_H, and then set $\bar{q}_h = p_H$.

Finally, we note that an upper bound for S_h, S_h^+, can be obtained following an analogous procedure as

$$S_h^+ = - \inf_{\hat{v}_h \in \hat{X}_h} \sup_{\mu_h \in X_h, q_H \in Q_H} \mathcal{L}^+(\hat{v}_h, \mu_h, q_H) \ , \tag{29}$$

where, $\mathcal{L}^+ : \hat{X}_h \times X_h \times Q_h \mapsto I\!R$, is given as

$$\mathcal{L}^+(\hat{v}_h, \mu_h, q_h) = B^s(\hat{v}_h, \hat{v}_h) - L(\hat{v}_h) \tag{30}$$
$$- L^O(\hat{v}_h) + B(\mu_h, \hat{v}_h) - L(\mu_h) + b(\hat{v}_h, q_h) \ .$$

Solution Procedure

Except for some particular situations, problems (25, 29) result in a globally coupled set of equations. This set of nonlinear equations must be solved iteratively. Therefore, having a good initial guess for v_h, μ_h and q_H, greatly simplifies the iterative process. In order to obtain good initial values, we solve a problem in the working mesh, $\mathcal{T}_H$, and obtain a solution $u_H \in X_H$. Then, we can determine multipliers $\psi_H^{\pm} \in X_H$ and $p_H^{\pm} \in Q_H$ by enforcing stationarity of the Lagrangian with respect to v_H and the dummy variables v_H^{*+} and v_H^{*-}, respectively. This process is quite straightforward, but the details are quite tedious and are given elsewhere.[14]

The general bounding procedure can be summarized as follows:

1. Solve (1) on the coarse working discretization and obtain $u_H \in X_H$.

2. Calculate $\psi_H \in X_H$. This is an inexpensive computation that can be carried out independently on each element of $\mathcal{T}_H$.

3. Calculate $p_H \in Q_H$. This is also a local calculation that can be carried our independently over each edge $\gamma_H \in \varepsilon(\mathcal{T}_H)$.

4. Solve (25) for S^-.

5. Repeat steps 2, and 3, and solve (29) for S^+.

The solution of problems (25, 29) is accomplished using a Newton-Raphson iteration. This requires the solution of a sequence of linear equations, the size of which is $dim\{\hat{X}_h\} + dim\{X_h\} + dim\{Q_H\}$. In principle, this is a very large system and therefore it would seem that there is no computational advantage in solving this problem over solving for $u_h \in X_h$ directly on the reference mesh. However, the special structure of the linear system of equations to be solved, means that this system can often be solved at a much lower cost than the, in principle, smaller system required to calculate u_h directly. For a coarse mesh consisting of N_H macro-elements, the problem unknowns can be organized so that the Hessian matrix corresponding to the linear system of equations to be solved, has the following structure,

$$\begin{pmatrix} A_{11} & 0 & \cdots & & B_1 \\ 0 & A_{22} & 0 & \cdots & B_2 \\ \vdots & 0 & \ddots & & \vdots \\ & \vdots & & A_{N_H N_H} & B_{N_H} \\ B_1^T & B_2^T & \cdots & B_{N_H}^T & 0 \end{pmatrix}. \quad (31)$$

In 2D and for quadrilateral piecewise bi-linear elements, and assuming that the mesh $\mathcal{T}_h$ is obtained by subdividing each macro-element, T_H, into $m = N_h/N_H$ elements, the square matrices A_{kk} have dimension $8m + 8\sqrt{m}$, reflecting $4m + 8\sqrt{m}$ unknowns associated with v_h and $v_h^{*\pm}$, and $4m$ unknowns associated with μ_h. The matrices B_k are of size $(8m + 8\sqrt{m}) \times 4N_H$. Note that the matrices B_k are very sparse. Systems of equations having this structure can be solved efficiently by transforming them to a much smaller system; the Schur complement, which in this case has size $4N_H \times 4N_H$ (the same size of the coarse grid problem in $\mathcal{T}_H$).

Applications

The first example considered is the linear convection equation. Equation (1) is solved with $f_1 = u$, $f_2 = 0$, $\sigma = 1$ and $r = 0$. The computational domain is given by $0 < x < 1$, $-0.5 < y < 0.5$. Boundary conditions are imposed such that the exact solution is given by $u(x,y) = e^{-x}(1 - y^2)$. The output of interest $S = L^O(u)$ is the average of the solution over the computational domain, i.e. $L^O(v) = \int_\Omega v\, dA$. Figure 2, shows an intermediate $H = H_0/4$ working mesh, $\mathcal{T}_H$, used in the computation. The reference mesh is obtained by subdividing each element of an original coarse mesh $\mathcal{T}_{H_0}$ into $8 \times 8 = 64$ elements. Figure 3 shows the contours for the solution computed on the $\mathcal{T}_{H_0/4}$ mesh with our DG algorithm using the symmetric mean value flux (13). The value of the output on the reference mesh is $S_h = 0.57944399$. Keeping the reference mesh, $\mathcal{T}_h$, fixed, we have computed the bounds using a sequence of coarse mesh discretizations which were obtained by successively subdividing the initial discretization $\mathcal{T}_{H_0}$. For this problem, it is possible to "freeze" the multiplier $\bar{q}_h$ in (28), and still obtain a bounded minimum when performing the minimizations in (29, 25). This results in a much lower computational effort. In our computations we have set $\bar{q}_h = p_H$, where p_H is the exact multiplier that would result from solving problem (1) on the mesh $\mathcal{T}_H$. The table below shows the upper and lower bounds computed for the output of interest using different coarse meshes. As expected, when the coarse mesh is refined, p_H becomes a better approximation to the exact value of the multiplier, p_h, and as a result the bound gap decreases.

	S_h^-	S_h^+
$H = H_0$	0.57943741	0.57945057
$H = H_0/2$	0.57944346	0.57944451
$H = H_0/4$	0.57944395	0.57944402
$H = h = H_0/8$	0.57944399	0.57944399

Finally, figure 4 shows the convergence of the computed bounds for the reference mesh solution, as the coarse mesh discretization is refined. A higher than second order convergence is observed.

The second example considered is a scalar nonlinear conservation law. Equation (1) is solved with $f_1 = \frac{1}{2}u^2$, $f_2 = u$, $\sigma = 0$ and the boundary conditions and $r(x,y)$, are chosen so that the exact solution to the problem is $u(x,y) = (1+x)(1-y)^2$. We use the same computational domain, output of interest, and discretizations, as in the previous example. Figure 5 shows contours of the solution computed on the working mesh $\mathcal{T}_{H_0/4}$. The sym-

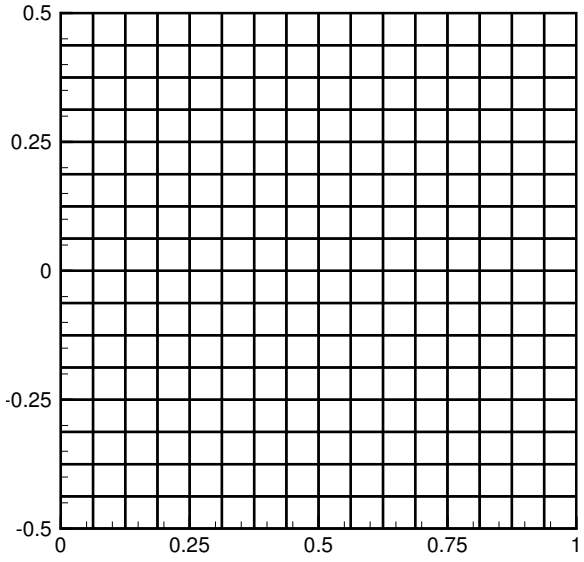

Fig. 2 Coarse Computational Mesh $\mathcal{T}_{H_0/4}$ used for the scalar computations.

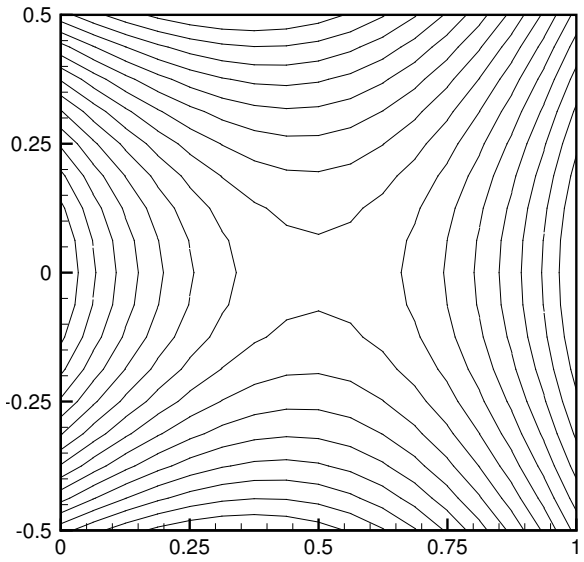

Fig. 3 Solution u_H for the linear convection problem computed on the discretization $\mathcal{T}_{H_0/4}$.

metric mean value flux (13) was also selected as the interface flux for this problem. The problem was solved by Newton-Raphson iteration, exploiting the block diagonal structure of the Hessian matrix. As a result, each Newton-Raphson did require the solution of a linear system of equation whose size was determined by the coarse grid discretization. The bounds computed for the different working meshes

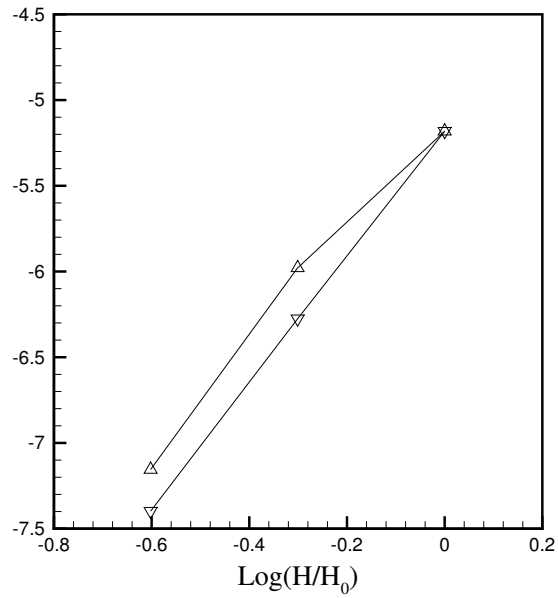

Fig. 4 Convergence of the upper and lower bounds to the reference mesh solution S_h, when the working mesh is refined. Plot shows Log($|S_h - S^{\pm}|$) vs. $Log(H/H_0)$

considered are given in the table below.

	S_h^-	S_h^+
$H = H_0$	1.372918	1.375463
$H = H_0/2$	1.374719	1.375111
$H = H_0/4$	1.374986	1.375007
$H = h = H_0/8$	1.375000	1.375000

Figure 6 shows the convergence of the computed bounds for the reference mesh solution, as the working discretization is refined. A higher than second order convergence is also observed for this problem.

The third numerical example involves preliminary results obtained for the Euler equations. In this example the reference mesh is obtained by enriching the polynomial approximation (p refinement) space rather than subdividing the elements. We have used piecewise bi-linear polynomials in the coarse working mesh and piecewise bi-cubic approximations for the reference solution. The extension of the approach presented above to deal with p refinement is quite straightforward and is described in detail elsewhere.[14] The Euler equations were written in terms of entropy variables[13] and a DG formulation using the HLL interface flux[2] was used. The problem considered is that of a diverging nozzle with an inflow Mach number of 0.5. The nozzle stretches from

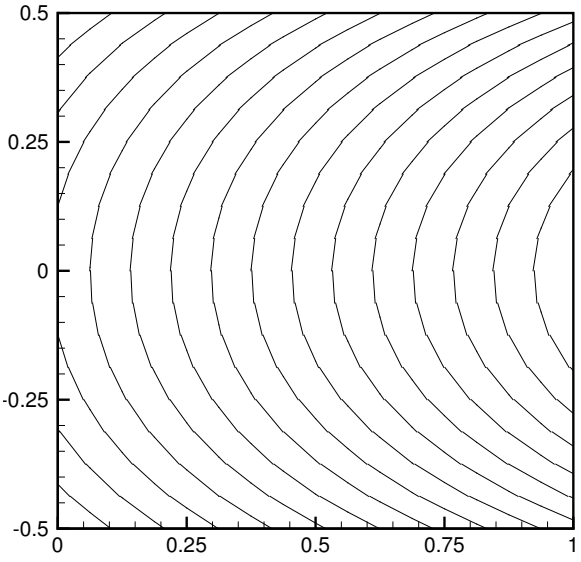

Fig. 5 Solution u_H for the non-linear problem computed on the discretization $\mathcal{T}_{H_0/4}$.

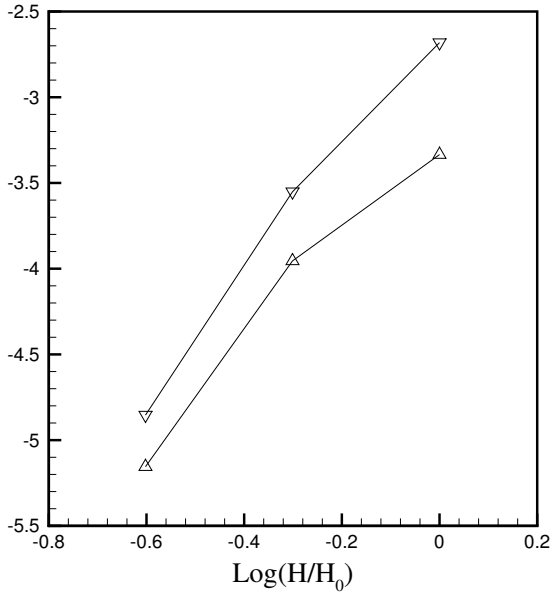

Fig. 6 Convergence of the upper and lower bounds to the reference mesh solution S_h, when the working mesh is refined. Plot shows Log($|S_h - S^{\pm}|$) vs. $Log(H/H_0)$

$x = 0$ to $x = 1$. The upper and lower walls of the nozzle vary according to $y = 1+0.05\sin^2(\pi(x-0.25))$ and $y = -0.05\sin^2(\pi(x-0.25))$ respectively, between $x = 0.25$ and $x = 0.75$. All boundary conditions were enforced weakly. The output of interest is taken to be the integrated horizontal pressure force on the wall. i.e. $S = \int_{\Gamma_w} n_1 p\, ds$, where Γ_w denotes the wall boundary and p is the static pressure. The coarsest computational mesh employed is shown in figure 7 and the contours of the Mach number solution computed on the reference mesh are shown in figure 8. For this problem, the straightforward use of Newton-Raphson iteration, to carry out the constrained minimization of problems (25) and (29), did not converge. This can be understood by looking at the structure of the Hessian matrix for a constrained minimization problem, which can be written as

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix}.$$

Here, the sub-matrix A is the Hessian associated with the unconstrained minimization problem, and the sub-matrices B and B^T enforce the constrains through the multipliers. It turns out that, the matrix A need only be positive definite in the null space of B at the constrained minimum for the problem to be well posed. For the Newton method to exhibit robust converge behavior, however, we require that the matrix A be positive definite in all directions. For the Euler equations this is not the case and therefore the convergence of the Newton method is not assured. A standard remedy is to use an augmented Lagrangian approach by which we multiply the constraint equations by cB, for some positive c, and add them to the primal equations so that one obtains a modified Hessian of the form

$$\begin{pmatrix} A + cBB^T & B \\ B^T & 0 \end{pmatrix}.$$

If the right hand side is modified accordingly, the solution of the equation system is unaltered, and, the matrix $A + cBB^T$ is positive definite for a sufficiently large c near the minimum. This approach was followed for the Euler test case, as it was found that this allowed for the constrained minimization iteration to converge. Unfortunately, this modification destroys the special structure of the Hessian and this, in turn, results in a much more costly solution procedure. A steepest descent algorithm coupled with first-order updates of the constraint variables was employed for the solution of this problem. The numerical results for the upper an lower bounds are shown in the table below

	$H = H_0$	$H = H_0/2$
S_h	-0.291359	-0.291487
S^-	-0.292874	-0.292136
S^+	-0.275970	-0.287646

Note that in this case, because the reference mesh is a function of H, the actual reference solution changes slightly as a function of the coarse mesh. Finally, figure 9 shows the convergence of the computed bounds as the coarse mesh is refined.

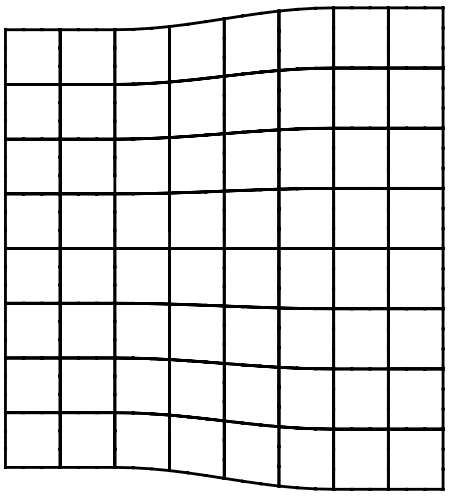

Fig. 7 Computational Mesh $\mathcal{T}_{H_0}$

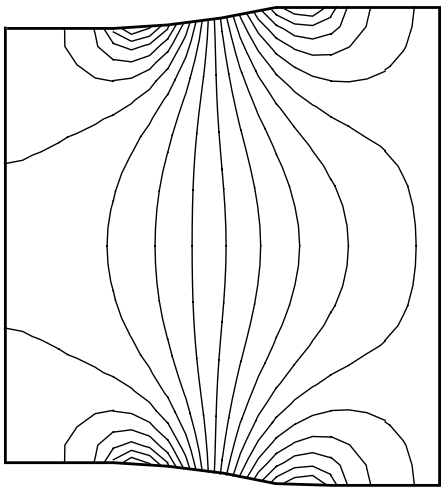

Fig. 8 Contours of the Mach solution on the reference mesh

Conclusions

We have presented a discontinuous Galerkin a-posteriori algorithm for the computation of bounds for functional outputs of solutions of systems of conservation laws. For certain classes of problems, including linear convection, the method is very efficient, and allows for the computation of bounds with a cost which is essentially linear in the number of elements in the fine reference mesh. For the non-linear scalar equations tested, fully decoupling is not possible and the algorithm requires the additional solution of a series of coarse mesh problems. For the Euler equations, fully decoupling is again not possible and, in addition, the resulting saddle point problem is not well conditioned and can not be solved directly using a Newton-Raphson method. In order to obtain a solution, in this case, it was necessary to modify the equation system, thereby destroying the matrix structure. The result is a considerably less efficient algorithm, but potentially still cheaper than computing the solution of the very refined reference mesh. More work is clearly needed to devise more effective and robust strategies in this case.

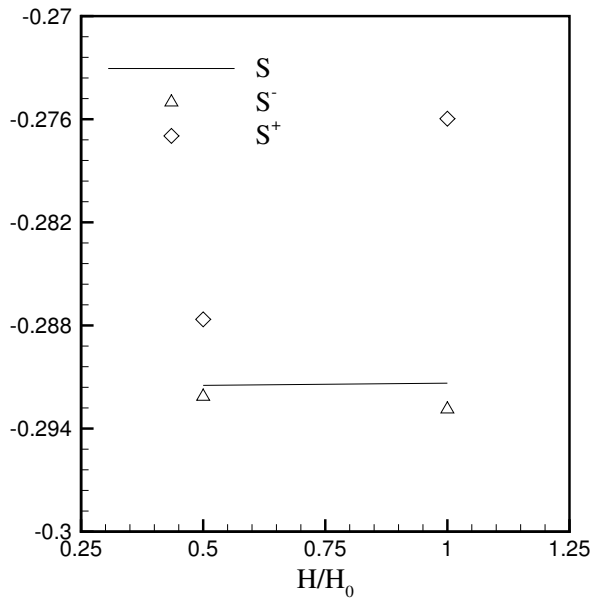

Fig. 9 Convergence of the upper and lower bounds to the reference solution S_h when the working mesh is refined

Acknowledgments

The authors would like to express their gratitude to the Singapore MIT Alliance for supporting this research.

References

[1] T. Barth and P. Charrier *Energy stable flux formulas for the discontinuous Galerkin discretization of first-order nonlinear conservation laws*, NASA Ames Research Center, Report NAS-01-001.

[2]A. Harten, P.D. Lax and B. van Leer *On upstream differencing and Godunov-type schemes for hyperbolic conservation laws*, SIAM Rev., 25, 35-61 (1983).

[3]M. Ainworth and J.T. Oden *A posteriori error estimation in finite element analysis*, TICAM Report 96-19, 1996.

[4]M. Ainworth and J.T. Oden *A unified approach to a posteriori error estimation using finite element residual methods*, Numer. Math., 65(1993), pp. 23-50

[5]J. Peraire and A.T. Patera *Bounds for linear-functional outputs of coercive partial differential equations : local indicators and adaptive refinement*, Advances in Adaptive Computational Methods in Mechanics, 1998, pp. 199-216.

[6]M. Paraschivoiu and A.T. Patera, *A hierarchical duality approach to bounds for the outputs of partial differential equations*, Comp. Meth. Appl. Mech. Engrg., 158 (1998) 389-407.

[7]M. Paraschivoiu, J. Peraire, and A.T. Patera, *A posteriori finite element bounds forlinear-functional outputs of elliptic partial differential equations*, Comp. Meth. Appl. Mech. Engrg., 150 (1997) 289-312.

[8]M. Paraschivoiu, *A Posteriori Finite Element Bounds forLinear-Functional Outputs of Coercive Partial Differential Equations and of the Stokes Problem*, Ph.D. Thesis, Department of mechanical Engineering, M.I.T., October, 1997.

[9]R. Becker and R. Rannacher *Weighted a posteriori error control in finite element methods*, IWR Preprint 96-1 (SFB 359), Heidelberg, 1996.

[10]R. Verfurth, *A posteriori error estimation and adaptive mesh-refinement techniques*, J. Comp. Appl. Math., 50(1994), pp. 67-83

[11]L. Machiels, A.T. Patera, J. Peraire, and Y. Maday, *A general framework for finite element a posteriori error control: application to linear and nonlinear convection-dominated problems*, ICFD Conference, March, 1998.

[12]J. Bonet, A. Huerta, and J. Peraire, *The efficient computation of bounds for functionals of finite element solutions in large strain elasticity*, Comp. Meth. Appl. Mech. Engrg., 191 (2002) 4807-4826.

[13]T.J.R. Hughes, L.P.Franca and M. Mallet, *A new finite element formulation for computational fluid dynamics : I. Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics*, Comp. Meth. in Appl. Mech. and Engnr., 54, 1986, pp. 223-234.

[14]J. Wong, *Implicit a-posteirori bounds for discontinuous Galerking methods*, Ph.D. Thesis, Massachusetts Insitute of Technology (in preparation).

[15]A.T. Patera and J. Peraire, *A general Lagrangian formulation for the computation of a-posteriori finite element bounds*, in Error Estimation and Adaptive Discretization Methods in Computationla Fluid Dynamics, Lecture Notes in Computational Science and Engineering, 25, T.J. Barth and H. Denconinck Eds., Springer 2003