Adaptive Discontinuous Galerkin Finite Element Methods for Compressible Fluid Flows

Paul Houston
University of Leicester

Ralf Hartmann
University of Heidelberg

Endre Süli
University of Oxford

1 Introduction

Multi-dimensional compressible fluid flows are modelled by nonlinear conservation laws whose solutions exhibit a wide range of localised structures, such as shock waves, contact discontinuities and rarefaction waves. The accurate numerical resolution of these features necessitates the use of locally refined, adaptive computational meshes. A majority of adaptive algorithms used to model compressible fluid flows will simply refine or adjust the computational grid according to an *ad hoc* criterion, such as a large gradient in a physical quantity, for example. Although this intuitive approach has had some success, it does not provide guaranteed error control. Moreover, *ad hoc* refinement strategies may not provide the most economical mesh design for the control of a given error quantity of interest, cf. [2, 3], for example. The aim of this paper is to discuss the *a posteriori* error analysis and adaptive mesh design for discontinuous Galerkin finite element approximations to systems of conservation laws. In Section 2, we introduce the model problem and formulate its discontinuous Galerkin finite element approximation. Section 3 is devoted to the derivation of weighted *a posteriori* error bounds for linear functionals of the solution. Finally, in Section 4 we present some numerical examples to demonstrate the performance of the resulting adaptive finite element algorithm.

2 Model problem and discretisation

Given an open bounded polyhedral domain Ω in $\mathbb{R}^n$, $n \geq 1$, with boundary $\partial\Omega$, we consider the following problem: find $\mathbf{u} : \Omega \to \mathbb{R}^m$, $m \geq 1$, such that

$$\mathrm{div}\mathcal{F}(\mathbf{u}) = 0 \quad \text{in } \Omega, \tag{2.1}$$

where, $\mathcal{F} : \mathbb{R}^m \to \mathbb{R}^{m\times n}$ is continuously differentiable. We assume that the system of conservation laws (2.1) may be supplemented by appropriate initial/boundary conditions. For example, assuming that $B(\mathbf{u}, \nu) := \sum_{i=1}^n \nu_i \nabla_{\mathbf{u}} \mathcal{F}_i(\mathbf{u})$ has m real eigenvalues and a complete set of linearly independent eigenvectors for all $\nu = (\nu_1, \ldots, \nu_n) \in \mathbb{R}^n$; then at inflow/outflow boundaries, we require that $B^-(\mathbf{u}, \mathbf{n})(\mathbf{u} - \mathbf{g}) = \mathbf{0}$, where $\mathbf{n}$ denotes the unit outward normal vector to $\partial\Omega$, $B^-(\mathbf{u}, \mathbf{n})$ is the negative part of $B(\mathbf{u}, \mathbf{n})$ and $\mathbf{g}$ is a (given) real–valued function.

To formulate the discontinuous Galerkin finite element method (DGFEM, for short) for (2.1), we first introduce some notation. Let $\mathcal{T}_h = \{\kappa\}$ be an admissible subdivision of Ω into open element domains κ; here h is a piecewise constant mesh function with $h(\mathbf{x}) = \mathrm{diam}(\kappa)$

when $\mathbf{x}$ is in element κ. For $p \in \mathbb{N}_0$, we define the following finite element space

$$S_{h,p} = \{\mathbf{v} \in [L_2(\Omega)]^m : \mathbf{v}|_\kappa \in [\mathcal{P}_p(\kappa)]^m \quad \forall \kappa \in \mathcal{T}_h\}, \tag{2.2}$$

where $\mathcal{P}_p(\kappa)$ denotes the set of polynomials of degree at most p over κ. Given that $\mathbf{v} \in [H^1(\kappa)]^m$ for each $\kappa \in \mathcal{T}_h$, we denote by $\mathbf{v}^+$ (resp., $\mathbf{v}^-$) the interior (resp., exterior) trace of $\mathbf{v}$ on $\partial\kappa$. The DGFEM for (2.1) is defined as follows: find $\mathbf{u}_h \in S_{h,p}$ such that

$$\sum_{\kappa \in \mathcal{T}_h} \left\{ -\int_\kappa \mathcal{F}(\mathbf{u}_h) \cdot \nabla \mathbf{v}_h \, \mathrm{d}\mathbf{x} + \int_{\partial\kappa} \mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}_\kappa) \, \mathbf{v}_h^+ \, \mathrm{d}s + \int_\kappa \varepsilon \nabla \mathbf{u}_h \cdot \nabla \mathbf{v}_h \, \mathrm{d}\mathbf{x} \right\} = 0 \tag{2.3}$$

for all $\mathbf{v}_h \in S_{h,p}$, cf. [2,5], for example. Here, $\mathcal{H}(\cdot,\cdot,\cdot)$ denotes a *numerical flux* function, assumed to be Lipschitz continuous, consistent and conservative. We emphasize that the choice of the numerical flux function is completely independent of the finite element space employed; in Section 4, we employ the (local) Lax–Friedrichs flux. Further, ε denotes the coefficient of artificial viscosity defined by $\varepsilon = C_\varepsilon h^{2-\beta} |\mathrm{div}\mathcal{F}(\mathbf{u}_h)|$, where C_ε is a positive constant and $0 < \beta < 1/2$; see [5]. For elements $\kappa \in \mathcal{T}_h$ whose boundary intersects $\partial\Omega$, we replace $\mathbf{u}_h^-$ by appropriate boundary/initial conditions on $\partial\kappa \cap \partial\Omega$.

3 A posteriori error analysis

In this section, we shall be concerned with controlling the error in the numerical solution measured in terms of a given linear functional $J(\cdot)$. Here, $J(\cdot)$ is the physical quantity of interest which may, for example, represent the mean flow across a curve, a point value of the solution, or the lift or drag coefficients of a body immersed into an inviscid fluid, cf. [3].

We begin by first introducing some notation. Letting $\mathcal{N}(\mathbf{u}_h, \mathbf{v}_h)$ denote the left–hand side of (2.3), we write $\mathcal{M}(\mathbf{u}, \mathbf{u}_h; \cdot, \cdot)$ to denote the mean–value linearisation of $\mathcal{N}(\cdot,\cdot)$ given by

$$\mathcal{M}(\mathbf{u}, \mathbf{u}_h; \mathbf{u} - \mathbf{u}_h, \mathbf{v}) = \mathcal{N}(\mathbf{u}, \mathbf{v}) - \mathcal{N}(\mathbf{u}_h, \mathbf{v}) = \int_0^1 \mathcal{N}'_{\mathbf{u}}[\theta \mathbf{u} + (1-\theta)\mathbf{u}_h](\mathbf{u} - \mathbf{u}_h, \mathbf{v}) \, \mathrm{d}\theta \tag{3.1}$$

for all $\mathbf{v}$ in V, where V is a suitable function space such that $S_{h,p} \subset V$. Here, $\mathcal{N}'_{\mathbf{u}}[\mathbf{w}](\cdot, \mathbf{v})$ denotes the Fréchet derivative of $\mathbf{u} \mapsto \mathcal{N}(\mathbf{u}, \mathbf{v})$, for $\mathbf{v} \in V$ fixed, at some $\mathbf{w}$ in V. We remark that the linearisation defined in (3.1) is only a *formal* calculation, in the sense that $\mathcal{N}'_{\mathbf{u}}[\mathbf{w}](\cdot,\cdot)$ may not in general exist. Instead, a suitable approximation to $\mathcal{N}'_{\mathbf{u}}[\mathbf{w}](\cdot,\cdot)$ must be determined, for example, by computing appropriate finite difference quotients of $\mathcal{N}(\cdot,\cdot)$, cf. [2]. For the proceeding analysis, we *assume* that the linearisation (3.1) is well–defined. Under this hypothesis, we introduce the following *dual* problem: find $\mathbf{z} \in V$ such that

$$\mathcal{M}(\mathbf{u}, \mathbf{u}_h; \mathbf{w}, \mathbf{z}) = J(\mathbf{w}) \quad \forall \mathbf{w} \in V. \tag{3.2}$$

We assume that (3.2) possesses a unique solution. Clearly, the validity of this assumption depends on both the definition of $\mathcal{M}(\mathbf{u}, \mathbf{u}_h; \cdot, \cdot)$ and the choice of the linear functional under consideration, cf. [2]. For the proceeding error analysis, we must therefore *assume* that the dual problem (3.2) is well–posed. Under this assumption, we have the following result.

Theorem 1 *Let $\mathbf{u}$ and $\mathbf{u}_h$ denote the solutions of (2.1) and (2.3), respectively, and suppose that the dual problem (3.2) is well–posed. Then,*

$$J(\mathbf{u}) - J(\mathbf{u}_h) = \mathcal{E}_\Omega \equiv \sum_{\kappa \in \mathcal{T}_h} \eta_\kappa, \tag{3.3}$$

where

$$\eta_\kappa = \int_\kappa R_h \left(\mathbf{z} - \mathbf{z}_h\right) d\mathbf{x} + \int_{\partial\kappa} r_h \left(\mathbf{z} - \mathbf{z}_h\right)^+ ds - \int_\kappa \varepsilon \nabla \mathbf{u}_h \cdot \nabla(\mathbf{z} - \mathbf{z}_h) \, d\mathbf{x}$$

for all $\mathbf{z}_h$ *in* $S_{h,p}$. *Here,* $R_h|_\kappa = -\mathrm{div}\mathcal{F}(\mathbf{u}_h)$ *and* $r_h|_\kappa = \mathcal{F}(\mathbf{u}_h) \cdot \mathbf{n}_\kappa - \mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}_\kappa)$ *denote the internal and boundary finite element residuals, respectively, defined on each* $\kappa \in \mathcal{T}_h$.

Proof Choosing $\mathbf{w} = \mathbf{u} - \mathbf{u}_h$ in (3.2), recalling the linearity of $J(\cdot)$, and exploiting the Galerkin orthogonality property $\mathcal{N}(\mathbf{u}, \mathbf{v}_h) - \mathcal{N}(\mathbf{u}_h, \mathbf{v}_h) = 0$ for all $\mathbf{v}_h$ in $S_{h,p}$, we get

$$J(\mathbf{u}) - J(\mathbf{u}_h) = J(\mathbf{u} - \mathbf{u}_h) = \mathcal{M}(\mathbf{u}, \mathbf{u}_h; \mathbf{u} - \mathbf{u}_h, \mathbf{z}) = \mathcal{M}(\mathbf{u}, \mathbf{u}_h; \mathbf{u} - \mathbf{u}_h, \mathbf{z} - \mathbf{z}_h) = -\mathcal{N}(\mathbf{u}_h, \mathbf{z} - \mathbf{z}_h)$$

for all $\mathbf{z}_h$ in $S_{h,p}$. Equation (3.3) now follows by employing the divergence theorem. ■

Based on the general error representation formula derived in Theorem 1, *a posteriori* error estimates bounding the error in the computed functional $J(\cdot)$ may be deduced. Before we proceed, let us first recall the classification into Type I and Type II *a posteriori* error bounds introduced in [4]. Type I *a posteriori* error bounds, also referred to as *weighted a posteriori* error bounds, cf. [3], involve the multiplication of the computable terms R_h, r_h and $\varepsilon\nabla\mathbf{u}_h$, by the difference between the dual solution $\mathbf{z}$ and its projection (or interpolant) $\mathbf{z}_h$. On the other hand, Type II *a posteriori* bounds are in the spirit of the error analysis of Johnson *et al.* [1], and do not depend explicitly on the dual solution. These latter error estimates are derived from Type I *a posteriori* bounds by employing standard results from approximation theory to estimate the projection (or interpolation) error, together with well–posedness results for the dual problem. The resulting Type II error bounds only involve certain norms of the residuals and artificial viscosity term, the mesh function h, interpolation constants and the stability factor of the dual problem.

However, as we shall see in Section 4, the elimination of the 'weighting terms' involving the difference between $\mathbf{z}$ and $\mathbf{z}_h$ may adversely affect the performance of our adaptive finite element method. Indeed, mesh refinement strategies based on Type II *a posteriori* error bounds which do not require the computation of the dual solution may, under mesh refinement, lead to suboptimal convergence of the error $J(\mathbf{u}) - J(\mathbf{u}_h)$ in the computed functional, resulting in uneconomical mesh design and an inefficient adaptive algorithm, cf. [3]. For this reason, we only consider the derivation of the following Type I *a posteriori* error bound.

Corollary 1A *Under the assumptions of Theorem 1, we have*

$$|J(\mathbf{u}) - J(\mathbf{u}_h)| \leq \mathcal{E}_{|\Omega|} \equiv \sum_{\kappa \in \mathcal{T}_h} |\eta_\kappa| \,. \tag{3.4}$$

Proof The error bound (3.4) follows from (3.3) by application of the triangle inequality. ■

We end this section by noting that for nonlinear hyperbolic conservation laws, both the error representation formula (3.3) and the Type I *a posteriori* error bound (3.4) depend on the unknown analytical solution to the primal and dual problems. Thus, in order to render these quantities computable, both $\mathbf{u}$ and $\mathbf{z}$ must be replaced by suitable approximations. To this end, the linearisation leading to $\mathcal{M}(\mathbf{u}, \mathbf{u}_h; \cdot, \cdot)$ is performed about $\mathbf{u}_h$ and the dual solution $\mathbf{z}$ is replaced by a discontinuous Galerkin approximation $\hat{\mathbf{z}}$ computed on the same mesh $\mathcal{T}_h$ used for $\mathbf{u}_h$, but with a higher degree polynomial, i.e., $\hat{\mathbf{z}} \in S_{h,\hat{p}}$, $\hat{p} > p$.

4 Numerical experiments

In this section we present some numerical examples to highlight the advantages of designing an adaptive finite element algorithm based on the weighted error indicators $|\eta_\kappa|$ in comparison

Nodes	Elements	DOF	$J(u-u_h)$	$\sum_\kappa \eta_\kappa$	θ_1	$\sum_\kappa \|\eta_\kappa\|$	θ_2
81	128	384	0.2441E-01	0.4158E-01	1.70	0.9703E-01	3.97
135	230	690	-0.2797	0.4037E-01	-0.14	3.750	13.41
254	464	1392	-0.2507	-0.3753E-01	0.15	0.2777	1.11
566	1079	3237	-0.1173	-0.4558E-01	0.39	0.8735E-01	0.74
1100	2142	6426	0.3788E-03	0.3023E-03	0.80	0.2829E-02	7.47
2005	3942	11826	0.1801E-04	0.1877E-04	1.04	0.4967E-03	27.58
3624	7180	21540	0.2559E-05	0.2476E-05	0.97	0.2387E-03	93.27
6938	13808	41424	0.3654E-06	0.3650E-06	1.00	0.1045E-03	286.01
13560	27052	81156	0.8224E-07	0.8445E-07	1.03	0.4244E-04	516.11
27209	54343	163029	0.4271E-08	0.4551E-08	1.07	0.1718E-04	4023.63

Table 1: Example 1. Adaptive algorithm for Burgers' equation.

with traditional refinement strategies which do not require the solution of the dual problem (3.2). To this end, we consider the *ad hoc* error indicator $\eta_\kappa^{\mathrm{adhoc}} = \|hR_h\|_{L_2(\kappa)} + \|h^{1/2}r_h\|_{L_2(\partial\kappa)}$, which stems from a Type II *a posteriori* error analysis, cf. [4]. Here, we employ the fixed fraction mesh refinement algorithm with refinement and derefinement fractions set to 20% and 10%, respectively. Finally, we note that all computations are performed with $p = 1$ and $\hat{p} = 2$.

4.1 Burgers' equation

Here, we consider the one–dimensional unsteady inviscid Burgers' equation for the scalar variable $\mathbf{u} \equiv u$; i.e., writing x_2 to denote time, we have $u_{x_2} + \left((1/2)\,u^2\right)_{x_1} = 0$, on the (space–time) domain $\Omega = (0,1)^2$, subject to the initial condition

$$u(x_1,0) = \begin{cases} 3/2, & \text{for } \ x_1 \le 0, \\ 0, & \text{for } \ 0 < x_1 \le 1/8, \\ (3/8)(\cos(4\pi(x_1 - 3/8)) + 1), & \text{for } \ 1/8 < x_1 \le 5/8, \\ 0, & \text{for } \ x_1 > 5/8, \end{cases}$$

and boundary condition $u(0,x_2) = 3/2$, for $x_2 \in [0,1]$. The analytical solution to this problem consists of two shocks in the space–time domain separated by a smooth region containing a rarefaction wave. The two shocks eventually merge to form a single line of discontinuity in the space–time plane. In this example we select the functional of interest $J(\cdot)$ to be the value of the solution just before the shocks interact with one another. More precisely, we choose $J(u) = u(0.8, 0.9)$; thereby, the true value of the functional is $J(u) = 0.564318816016671$.

In Table 1, we demonstrate the performance of the adaptive algorithm with $C_\varepsilon = 1/4$ and $\beta = 1/10$. Here, we show the number of nodes, elements and degrees of freedom (DOF) in $S_{h,1}$, the true error in the functional $J(u - u_h)$, the computed error representation formula (3.3), the *a posteriori* error bound (3.4) and their respective effectivity indices θ_1 and θ_2. We see that initially on very coarse meshes the quality of the computed error representation formula $\mathcal{E}_\Omega$ is very poor, in the sense that $\theta_1 = \mathcal{E}_\Omega / J(u - u_h)$ is not close to one; however, as the mesh is refined the effectivity index θ_1 approaches unity. On the other hand, we observe that the Type I *a posteriori* error bound is not sharp, in the sense that the second effectivity index θ_2 overestimates the true error in the computed functional by three orders of magnitude on the finest mesh; this loss of sharpness is attributed to the loss of inter–element cancellation of the local error indicators η_κ, when the triangle inequality is employed. Thereby, it is clear

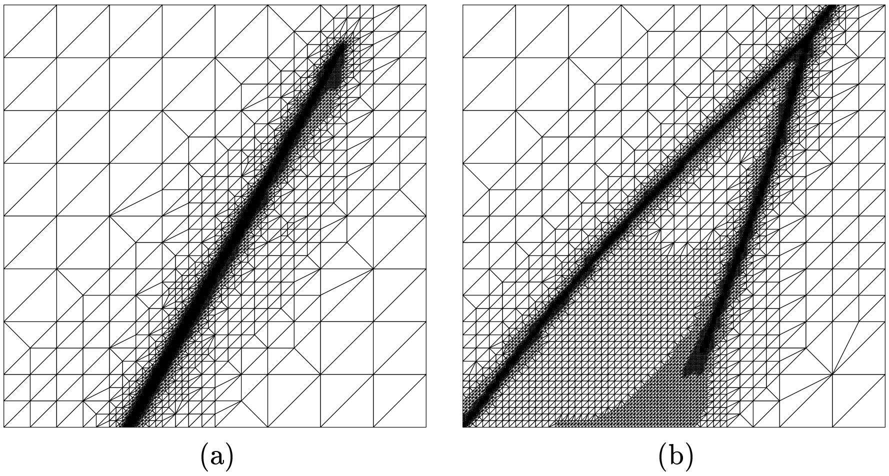


Figure 1: Example 1. (a) Mesh constructed using dual error indicator with 27209 nodes and 54343 elements ($|J(u) - J(u_h)| = 4.271 \times 10^{-9}$); (b) Mesh constructed using *ad hoc* error indicator with 17288 nodes and 34432 elements ($|J(u) - J(u_h)| = 1.358 \times 10^{-5}$).

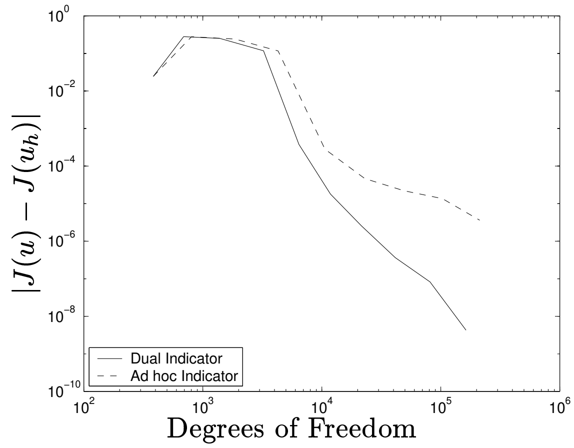


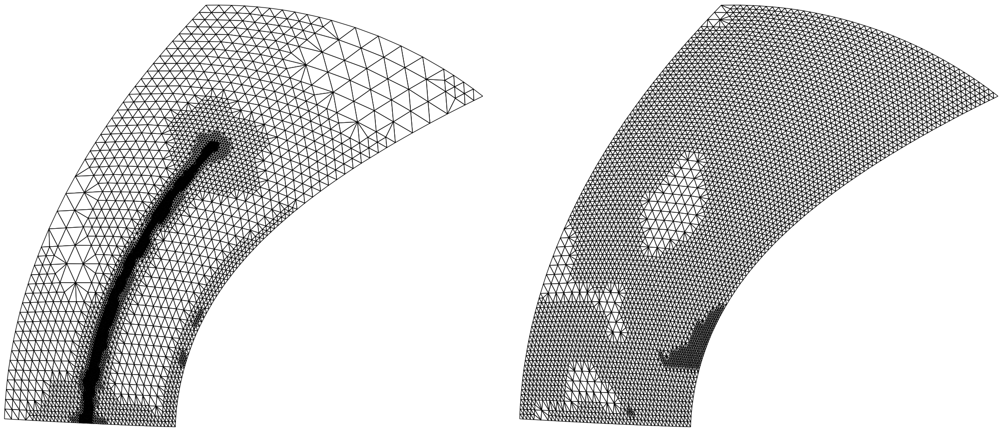
Figure 2: Example 1. Convergence of $J(u) - J(u_h)$ using the dual and *ad hoc* error indicators.

that any further bounding performed *en route* to deriving a Type II *a posteriori* estimate will further adversely affect the quality of the computed error bound.

In Figure 1 we show the meshes generated using both the weighted error indicator $|\eta_\kappa|$ and the *ad hoc* error indicator $\eta_\kappa^{\mathrm{adhoc}}$. From Figure 1(a), we see that there is virtually no refinement in the regions of the computational domain where the shocks are located when $|\eta_\kappa|$ is employed. Indeed, the most of the mesh refinement is concentrated in the neighbourhood upstream of the point of interest. In contrast, the mesh produced using the *ad hoc* error indicator is largely concentrated the vicinity of the two shock waves, with some refinement of the smooth region between them. In Figure 2, we compare the true error in the computed functional $J(\cdot)$ using the two mesh refinement strategies. Here, we clearly observe the superiority of the weighted *a posteriori* error indicators; on the final mesh the true error in the linear functional is over three orders of magnitude smaller than $J(u-u_h)$ computed on the sequence of meshes produced using $\eta_\kappa^{\mathrm{adhoc}}$. This clearly indicates that a good numerical resolution of the shocks is irrelevant for the accurate approximation of the functional of interest.

Nodes	Elements	DOF	$J(u-u_h)$	$\sum_\kappa \eta_\kappa$	θ_1	$\sum_\kappa \lvert\eta_\kappa\rvert$	θ_2
91	144	1728	-0.6734E-03	-0.6822E-03	1.01	0.6595E-02	9.79
157	274	3288	0.1488E-03	0.1473E-03	0.99	0.2827E-02	19.00
323	596	7152	0.3327E-04	0.3216E-04	0.97	0.1114E-02	33.47
564	1060	12720	-0.1006E-04	-0.9445E-05	0.94	0.7351E-03	73.04
1010	1931	23172	-0.1250E-04	-0.1241E-04	0.99	0.3145E-03	25.16
1905	3702	44424	-0.8657E-05	-0.8545E-05	0.99	0.2257E-03	26.07
3706	7263	87156	-0.2951E-05	-0.2933E-05	0.99	0.8870E-04	30.05
7215	14228	170736	-0.1087E-05	-0.1079E-05	0.99	0.3792E-04	34.88

Table 2: Example 2. Adaptive algorithm for Ringleb's flow.



(a) (b)

Figure 3: Example 2. (a) Mesh constructed using dual error indicator with 7215 nodes and 14228 elements ($|J(\mathbf{u}) - J(\mathbf{u}_h)| = 1.087 \times 10^{-6}$); (b) Mesh constructed using *ad hoc* error indicator with 4627 nodes and 8965 elements ($|J(\mathbf{u}) - J(\mathbf{u}_h)| = 2.332 \times 10^{-5}$).

4.2 Ringleb's flow

In this final example we consider the steady two–dimensional compressible Euler equations; here, $\mathbf{u}$ represents the vector of conserved quantities $(\rho, \rho u, \rho v, \rho E)$, where ρ, (u, v) and E represent the density, Cartesian velocity and total energy per unit mass, respectively. Here, we consider Ringleb's flow for which an analytical solution may be obtained using the hodograph method. This problem represents a transonic flow which turns around an obstacle; the flow is mostly subsonic, with a small supersonic region around the nose of the obstacle.

Given that the solution to Ringleb's flow is smooth, no artificial viscosity is required; thereby, we set $C_\varepsilon = 0$. Further, we take the functional of interest to be the value of the total energy at the point $(-0.4, 2)$, i.e., $J(\mathbf{u}) = (\rho E)(-0.4, 2)$; consequently the true value of the functional is given by $J(\mathbf{u}) = 1.574181283916596$. In Table 2 we show the performance of our adaptive algorithm; here we see that the quality of the computed error representation formula is extremely good, with $\theta_1 \approx 1$ even on very coarse meshes. Furthermore, the Type I *a posteriori* error bound (3.4) is sharper for this smooth problem than for Example 1; indeed, here $\mathcal{E}_{|\Omega|}$ overestimates $|J(\mathbf{u} - \mathbf{u}_h)|$ by just over an order of magnitude.

The meshes produced using both our weighted error indicator $|\eta_\kappa|$ and the *ad hoc* error

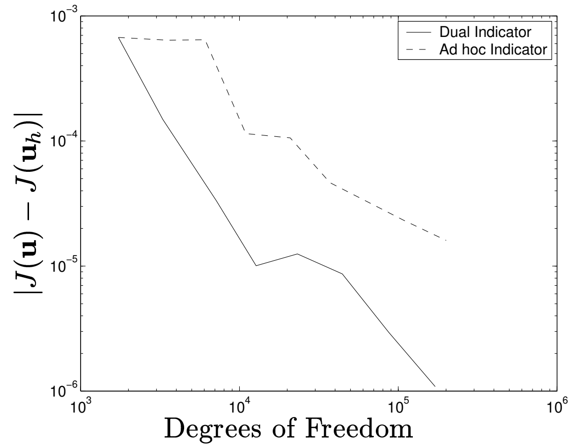


Figure 4: Example 2. Convergence of $J(\mathbf{u})-J(\mathbf{u}_h)$ using the dual and *ad hoc* error indicators.

indicator $\eta_\kappa^{\mathrm{adhoc}}$ are shown in Figure 3. Here, we see that, as in the previous example, the mesh constructed using $|\eta_\kappa|$ is mostly concentrated in the neighbourhood upstream of the point of interest. However, due to the elliptic nature of the flow in the subsonic region, a circular region containing the point of interest is also refined, together with a strip of cells on the wall of the obstacle enclosing the supersonic region of the flow. In contrast, the mesh constructed using $\eta_\kappa^{\mathrm{adhoc}}$ is largely concentrated in the region near the obstacle, with further almost uniform refinement of the rest of the computational domain; we note that subsequent mesh refinement leads to an even greater concentration of elements near the obstacle. Finally, in Figure 4 we compare the true error in the computed functional $J(\cdot)$ using the two mesh refinement strategies. Again, as in Example 1, we clearly see that our weighted *a posteriori* error indicators produce more economical meshes than when the traditional *ad hoc* error indicator is employed, in the sense that $J(\mathbf{u}-\mathbf{u}_h)$ is smaller for a given number of degrees of freedom. Indeed, on the final mesh the true error in the computed functional is over an order of magnitude smaller when the former error indicator is used.

Acknowledgments

Ralf Hartmann acknowledges the financial support of the DFG Priority Research Program and the SFB 359 at the IWR, University of Heidelberg. Paul Houston acknowledges the financial support of the EPSRC (Grant GR/N24230).

References

[1] P. Hansbo and C. Johnson, *Streamline diffusion finite element methods for fluid flow.* von Karman Institute Lectures, 1995.

[2] R. Hartmann and P. Houston, *Adaptive Discontinuous Galerkin Finite Element Methods for Nonlinear Hyperbolic Conservation Laws.* In preparation.

[3] P. Houston, R. Rannacher and E. Süli, *A posteriori error analysis for stabilised finite element approximations of transport problems.* Comput. Meth. Appl. Mech. Engrg. 190(11-12):1483–1508, 2000.

[4] P. Houston and E. Süli, *hp–Adaptive discontinuous Galerkin finite element methods for first–order hyperbolic problems.* Leicester University Technical Report 2000/31, 2000. (Submitted).

[5] J. Jaffre, C. Johnson and A. Szepessy, *Convergence of the discontinuous Galerkin finite element method for hyperbolic conservation laws.* Math. Models Methods Appl. Sci. 5(3):367–386, 1995.