Computers & Fluids 46 (2011) 277–281

Contents lists available at ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

Towards the development of an h–p-refinement strategy based upon error estimate sensitivity

Peter K. Jimack [a,*], Robert M. Kirby [b]

[a] School of Computing, University of Leeds, Leeds, UK
[b] School of Computing, University of Utah, Salt Lake City, UT, USA

ARTICLE INFO

Article history:
Received 30 April 2010
Received in revised form 3 August 2010
Accepted 5 August 2010
Available online 10 August 2010

Keywords:
Mesh adaptivity
Error estimation
Discrete adjoint

ABSTRACT

The use of (*a posteriori*) error estimates is a fundamental tool in the application of adaptive numerical methods across a range of fluid flow problems. Such estimates are incomplete however, in that they do not necessarily indicate where to refine in order to achieve the most impact on the error, nor what type of refinement (for example h-refinement or p-refinement) will be best. This paper extends preliminary work of the authors (Comm Comp Phys, 2010;7:631–8), which uses adjoint-based sensitivity estimates in order to address these questions, to include application with p-refinement to arbitrary order and the use of practical *a posteriori* estimates. Results are presented which demonstrate that the proposed approach can guide both the h-refinement and the p-refinement processes, to yield improvements in the adaptive strategy compared to the use of more orthodox criteria.

1. Introduction

The use of adaptive finite element methods for the solution of fluid flow problems is now widespread, e.g. [6,9,12], with the most popular techniques being based upon element sub-division (h-refinement) [14] or polynomial order enrichment (p-refinement) [5]. Indeed, combinations of these approaches are now beginning to appear quite frequently, since p-refinement is superior in regions where the solution is smooth, whilst h-refinement is generally superior elsewhere [1,10,11,15].

In order to drive adaptivity in a robust manner, reliable *a posteriori* error estimates are required. These estimates may approximate the error itself, as in [2,4], a norm of the error, as in Zienkiewicz and Zhu [20], or they may estimate the error in some derived quantity, e.g. [3,19]. In each case, when the estimated error is greater than some desired threshold then refinement should take place. Typically this refinement is restricted to those elements which contribute most to the overall error [18], whilst a variety of techniques have been proposed for selecting between h- and p-refinement locally [1,10,11,15].

In this paper we present an extension of the preliminary results reported in [7], in which a new approach to controlling local refinement was proposed. The idea is to compute the sensitivity of the error (estimate) to the potential addition of degrees of freedom, and then select those for which this sensitivity is greatest. Here we apply the approach to a wider variety of test problems than in [7], and also demonstrate its applicability both to p-refinement of arbitrary order and alongside a practical *a posteriori* error estimate. Extensions to nonlinear problems and a wider range of error estimates are also discussed.

2. Methodology

2.1. Notation

Consider as our model problem a linear second-order two-point boundary value problem (BVP) of the form:

$$\frac{d}{dx}\left(a(x)\frac{du}{dx}\right)+b(x)\frac{du}{dx}+c(x)=f \tag{1}$$

on the domain (x_0,x_N), where $a(x)>0$ and zero Dirichlet boundary conditions are assumed for simplicity. This problem may be expressed in weak form as: find $u\in H_0^1(\Omega)$ such that

$$a(u,v)=(f,v)\quad \forall v\in H_0^1(\Omega), \tag{2}$$

where

$$a(u,v)=-\int_{x_0}^{x_N}a(x)\frac{du}{dx}\frac{dv}{dx}dx+\int_{x_0}^{x_N}b(x)\frac{du}{dx}v\,dx+\int_{x_0}^{x_N}c(x)uv\,dx \tag{3}$$

and

$$(f,v)=\int_{x_0}^{x_N}fv\,dx. \tag{4}$$

* Corresponding author.
E-mail addresses: p.k.jimack@leeds.ac.uk (P.K. Jimack), kirby@sci.utah.edu (R.M. Kirby).

0045-7930/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2010.08.003

The finite element method seeks an approximation $\tilde{u}$ from a finite dimensional subspace $V \subset H_0^1(\Omega)$ such that

$$a(\tilde{u}, v) = (f, v) \quad \forall v \in V. \tag{5}$$

Now consider two such finite dimensional spaces, V_1 and V_2, for which $V_1 \subset V_2$ with the following basis sets:

$$V_1 = \text{span}\{\phi_1, \ldots, \phi_{N1}\}, \tag{6}$$

$$V_2 = \text{span}\{\phi_1, \ldots, \phi_{N1}, \phi_{N1+1}, \ldots, \phi_{N1+N2}\}. \tag{7}$$

When V in (5) is equal to V_1, the finite element problem may be expressed as

$$K^{11}\underline{u}^1 = \underline{f}^1, \tag{8}$$

where

$$K_{ij}^{11} = a(\phi_i, \phi_j) \quad \text{for } i,j \in \{1, \ldots, N1\}, \tag{9}$$

$$f_i^1 = (f, \phi_i) \quad \text{for } i \in \{1, \ldots, N1\}, \tag{10}$$

and

$$\tilde{u} = \sum_{j=1}^{N1} u_j^1 \phi_j. \tag{11}$$

Alternatively, when V in (5) is equal to V_2 the finite element problem may be expressed as

$$\begin{bmatrix} K^{11} & K^{12} \\ K^{21} & K^{22} \end{bmatrix} \begin{bmatrix} \underline{u} \\ \underline{s} \end{bmatrix} = \begin{bmatrix} \underline{f}^1 \\ \underline{f}^2 \end{bmatrix}, \tag{12}$$

where

$$K_{ij}^{11} = a(\phi_i, \phi_j) \quad \text{for } i,j \in \{1, \ldots, N1\}, \tag{13}$$

$$K_{ij}^{12} = a(\phi_i, \phi_{N1+j}) \quad \text{for } i \in \{1, \ldots, N1\} \quad \text{and } j \in \{1, \ldots, N2\}, \tag{14}$$

$$K_{ij}^{21} = a(\phi_{N1+i}, \phi_j) \quad \text{for } i \in \{1, \ldots, N2\} \quad \text{and } j \in \{1, \ldots, N1\}, \tag{15}$$

$$K_{ij}^{22} = a(\phi_{N1+i}, \phi_{N1+j}) \quad \text{for } i,j \in \{1, \ldots, N2\}, \tag{16}$$

$$f_i^1 = (f, \phi_i) \quad \text{for } i \in \{1, \ldots, N1\}, \tag{17}$$

$$f_i^2 = (f, \phi_{N1+i}) \quad \text{for } i \in \{1, \ldots, N2\} \tag{18}$$

and

$$\tilde{u} = \sum_{j=1}^{N1} u_j \phi_j + \sum_{j=1}^{N2} s_j \phi_{N1+j}. \tag{19}$$

Note that one may view the solution of the smaller problem (8) as being equivalent to solving the larger problem (12) but with the vector $\underline{s}$ constrained to be $\underline{0}$. That is, when $\underline{s} = \underline{0}$ in (12), $\underline{u} = \underline{u}^1$. This view may be extended by thinking of $\underline{u}$ in (12) as depending upon the prescribed value of $\underline{s}$. That is, $\underline{u} = \underline{u}(\underline{s})$, where $\underline{u}(\underline{0}) = \underline{u}^1$.

2.2. Connection with adaptivity

Suppose the domain is divided into N intervals:

$$x_0 < x_1 < \ldots < x_{N-1} < x_N, \tag{20}$$

and let $\{\phi_0^1, \ldots, \phi_N^1\}$ be the usual basis (of local hat functions) for the space of continuous piecewise polynomials of degree 1 on this mesh. Given that $u(x_0) = u(x_N) = 0$, the corresponding piecewise linear finite element trial function takes the form:

$$u^1(x) = \sum_{i=1}^{N-1} u_i^1 \phi_i^1(x). \tag{21}$$

On each element $e = 1, \ldots, N$ also define local bubble functions (i.e. polynomials which are zero at the element boundaries) up to degree p : $\{\phi_e^2, \ldots, \phi_e^p\}$. With this notation the piecewise polynomial trial function of degree p takes the form:

$$u^p(x) = \sum_{i=1}^{N-1} \sum_{d=1}^{p} u_i^d \phi_i^d(x) + \sum_{d=2}^{p} u_N^d \phi_N^d(x). \tag{22}$$

We now consider the possibility of adapting a computed solution by increasing the polynomial degree p locally. Let us begin with the simplest possible case, where $p = 1$ on every element. The finite element solution in this case is given by (21), where the unknown coefficients are prescribed by the finite element equations (8). We would like to predict the consequences of increasing the polynomial degree (from $p = 1$ to $p = 2$ say) on each element. Note that it is possible to write

$$u^1(x) = \sum_{i=1}^{N-1} u_i^1 \phi_i^1(x) + \sum_{i=1}^{N} u_i^2 \phi_i^2(x), \tag{23}$$

provided that we impose

$$0 = u_1^2 = \ldots = u_N^2, \tag{24}$$

If we make the following definitions:

$$\underline{u} \in \Re^{N-1} = \left(u_1^1, \ldots, u_{N-1}^1\right)^T, \tag{25}$$

$$\underline{s} \in \Re^{N} = \left(u_1^2, \ldots, u_N^2\right)^T, \tag{26}$$

then the resulting finite element equations take the same form as (12), where based upon (24), $\underline{s} = \underline{0}$. In other words, the piecewise linear finite element solution on the grid (20), is given by the solution of (12) when $\underline{s} = \underline{0}$.

Having computed this finite element solution it is then possible to compute an *a posteriori* error estimate $E = E(\underline{u}, \underline{s})$ (again, this is computed for the value $\underline{s} = \underline{0}$). This error estimate will typically take the form of a single number, either for the size of the error itself or for the error in some quantity that may be derived from the solution. Alternatively, the estimate may produce an error function whose magnitude may be determined either locally or globally. See, for example [2–4] for a selection of typical results.

We now have a computed finite element solution and a computed error estimate E. If this error estimate is sufficiently small then there is no need to continue; however if we require E to be smaller, then there is a need for adaptivity. Suppose we are able to compute the value of $\frac{dE}{d\underline{s}}$. When a component of this vector is large (in magnitude) this tells us that moving the corresponding component of $\underline{s}$ away from its present value (which is zero) is likely to have a large impact on the error. Conversely, when a component of $\frac{dE}{d\underline{s}}$ is small, it is likely that moving the corresponding component of $\underline{s}$ away from zero will have less effect. The idea that we present in this work is the suggestion that this information may provide an effective means of deciding where to refine locally.

In addition to using this approach to determine where it may be most beneficial to refine, it is also possible to consider *how* it might be best to refine. Specifically, it is possible to consider the sensitivity of the computed error to both h- and p-refinement. Continuing with the example above, let $\psi_e(x)$ be the piecewise linear hat function associated with bisecting element e (for $e = 1, \ldots, N$). That is, $\psi_e(x)$ is equal to 1.0 at the midpoint of element e and to 0.0 at the end points, and is zero on every other element. Instead of (23) we may now write

$$u^1(x) = \sum_{i=1}^{N-1} u_i^1 \phi_i^1(x) + \sum_{i=1}^{N} v_i \psi_i(x), \tag{27}$$

provided that we impose

$$0 = v_1 = \ldots = v_N. \tag{28}$$

We now define

$$\underline{u} \in \Re^{N-1} = \left(u_1^1, \ldots, u_{N-1}^1\right)^T, \tag{29}$$

$$\underline{s} \in \Re^{N} = \left(v_1, \ldots, v_N\right)^T, \tag{30}$$

in Eq. (12). Again, if we can evaluate $\frac{dE}{d\underline{s}}$, we can assess the sensitivity of the error to each of these additional degrees of freedom. This

information may be used, along with the sensitivity to the addition of the higher degree basis functions, to decide both, where to refine and how to refine. It should be noted however that, in order to make a meaningful comparison, the different basis functions must be scaled appropriately: specifically their L_2 norm should be the same on each element [7]. Furthermore, it is only the magnitude of the entries of $\frac{dE}{d\underline{s}}$ that is significant.

Of course, the above discussion assumes that it is possible to calculate $\frac{dE}{d\underline{s}}$ both reliably and efficiently. At first sight this may seem uneconomical. For example, one could use the estimate

$$\frac{dE}{ds_k} \approx \frac{E(\underline{u}, \underline{s} + \delta \underline{e}_k) - E(\underline{u}, \underline{s})}{\delta}, \tag{31}$$

for $k = 1, \ldots, N$ and some small value of δ (where $\underline{e}_k \in \Re^N$ is the vector whose entries are all zero except for entry k, which is one). However this would require an excessive computational cost, which would defeat the objective of using local mesh adaptivity. Consequently, a more efficient means computing $\frac{dE}{d\underline{s}}$ is required. It is in meeting this requirement that the adjoint approach becomes important [8,9,13,17].

3. Sensitivity

In this section we focus on the efficient evaluation of $\frac{dE}{d\underline{s}}$. It will be shown that this simply requires the solution of a single adjoint problem, which in this case reduces to the transpose of the finite element stiffness matrix K_{11}. The extension to more general nonlinear problems is discussed in Section 5 below.

3.1. Formulation

The goal of the sensitivity analysis is to calculate the sensitivity of the estimated error to variations in $\underline{s}$ in the region of $\underline{s} = \underline{0}$. Note that we may express this error as $E = E(\underline{u}, \underline{s})$, and so

$$\frac{dE}{d\underline{s}} = \frac{\partial E}{\partial \underline{u}} \frac{\partial \underline{u}}{\partial \underline{s}} + \frac{\partial E}{\partial \underline{s}}, \tag{32}$$

where $\frac{dE}{d\underline{s}}$ is a row vector of dimension $N2 = N - 1$. Noting that, from (12),

$$\underline{u}(\underline{s}) = (K^{11})^{-1}(\underline{f}^1 - K^{12}\underline{s}), \tag{33}$$

it follows from (32) that,

$$\left(\frac{dE}{d\underline{s}}\right)^T = -(K^{12})^T (K^{11})^{-T} \left(\frac{\partial E}{\partial \underline{u}}\right)^T + \left(\frac{\partial E}{\partial \underline{s}}\right)^T. \tag{34}$$

In the special case, where the underlying PDE is self-adjoint, and so the stiffness matrix is symmetric, this simplifies further to

$$\left(\frac{dE}{d\underline{s}}\right)^T = -K^{21}(K^{11})^{-1} \left(\frac{\partial E}{\partial \underline{u}}\right)^T + \left(\frac{\partial E}{\partial \underline{s}}\right)^T. \tag{35}$$

3.2. Error estimation

In this paper we just consider one typical, and relatively simple, *a posteriori* error estimate from [4]. Given a piecewise linear approximation (21), this error estimate may be computed on each element $i = 1, \ldots, N$ by finding η_i such that

$$\alpha(\eta_i \phi_i^2, \phi_i^2) = \langle f, \phi_i^2 \rangle - \alpha(u^1, \phi_i^2), \tag{36}$$

where $\alpha(\cdot,\cdot)$ and $\langle \cdot,\cdot \rangle$ are given by (3) and (4) respectively. Hence, this estimate gives

$$\eta_i = \frac{\langle f, \phi_i^2 \rangle - \alpha(u^1, \phi_i^2)}{\alpha(\phi_i^2, \phi_i^2)}, \tag{37}$$

from which we may define the estimated error function

$$\eta(x) = \sum_{i=1}^{N} \eta_i \phi_i^2(x). \tag{38}$$

Finally, we take as our error the square of the L_2 norm:

$$E = \|\eta(x)\|_2^2 = \sum_{i=1}^{N} \eta_i^2 \int_{x_{i-1}}^{x_i} \left(\phi_i^2\right)^2 dx. \tag{39}$$

From the (37) it follows that

$$\frac{\partial \eta_i}{\partial u_j} = \frac{-\alpha\left(\phi_j^1, \phi_i^2\right)}{\alpha\left(\phi_i^2, \phi_i^2\right)}, \tag{40}$$

and therefore (39) implies that

$$\frac{\partial E}{\partial u_j} = 2\left[\eta_j \frac{-\alpha\left(\phi_j^1, \phi_j^2\right)}{\alpha\left(\phi_j^2, \phi_j^2\right)} \int_{x_{j-1}}^{x_j} \left(\phi_j^2\right)^2 dx + \eta_{j+1} \frac{-\alpha\left(\phi_j^1, \phi_{j+1}^2\right)}{\alpha\left(\phi_{j+1}^2, \phi_{j+1}^2\right)} \int_{x_j}^{x_{j+1}} \left(\phi_{j+1}^2\right)^2 dx\right], \tag{41}$$

for $j = 1, \ldots, N$. Similarly, in the case of h-refinement using the piecewise linear hat functions $\psi_i(x)$, it follows from (37) that

$$\frac{\partial \eta_i}{\partial s_j} = \frac{-\alpha(\psi_j, \phi_i^2)}{\alpha(\phi_i^2, \phi_i^2)}, \tag{42}$$

and consequently, from (39), that

$$\frac{\partial E}{\partial s_j} = 2\eta_j \frac{-\alpha\left(\psi_j, \phi_j^2\right)}{\alpha\left(\phi_j^2, \phi_j^2\right)} \int_{x_{j-1}}^{x_j} \left(\phi_j^2\right)^2 dx. \tag{43}$$

Using expressions (41) and (43) it is therefore possible to compute (34) as required.

4. Numerical results

4.1. Exact error

In this subsection we illustrate the feasibility of the proposed approach using the exact error, as in [7]. The main extensions here being that we demonstrate the sensitivity to the addition of polynomial basis functions of higher degree (i.e. p-refinement) and we consider a convection-dominated problem. The specific equation that we consider is

$$-\frac{d}{dx}\left(x\frac{du}{dx}\right) + 10\frac{du}{dx} = 100\pi^2 x^{19} \sin(\pi x^{10}), \tag{44}$$

subject to zero Dirichlet boundary conditions on the domain (0,1). This problem is selected to have the known solution $u = \sin(\pi x^{10})$ which is almost identically zero in the left half of the domain before gradually rising to a local maximum at $x \approx 0.933$, and then falling sharply down to zero again at $x = 1.0$.

Initially this problem is solved using 10 equally-spaced elements of degree 1, yielding a result with an L_2 error of 0.10564. Table 1 shows the local error on the right-most elements, along with the sensitivity to increasing the polynomial degree up to 5 on each of these elements (the errors and sensitivities on the other elements are not shown as they are so much smaller).

It is clear that the local error is greatest on the final element, however the sensitivity information tells us more than this. For example, if two degrees of freedom are to be added in the refinement process then Table 1 suggests that it is better to use basis functions of degrees 2 and 3 on element 10 than to use basis func-

Table 1
Local error and sensitivity to addition of basis functions of different polynomial degrees on the right-most elements of the domain following the initial solve.

Element	L_2 error	degree = 2	degree = 3	degree = 4	degree = 5
8	0.00151	0.00054	0.00044	0.00019	0.00001
9	0.05665	0.00207	0.00121	0.00019	0.00004
10	0.08786	−0.01255	−0.00267	0.00298	0.00037

tions of degree 2 on both elements 9 and 10. This may be confirmed by undertaking both refinement patterns and comparing the errors in the resulting solutions: the former case leading to an error of 0.00710 compared to an error of 0.01264 in the latter case. Table 2 shows the local errors and sensitivities following the first of these solves (with basis functions of degrees 2 and 3 on the final element).

At this stage it is now clear that the greatest error is in element 9 but, unlike the previous case, the sensitivity now indicates that the best way to add the next two degrees of freedom is by adding a polynomial of degree 2 to each of two elements (8 and 9). When different adaptive schemes are tried and compared the conclusion is again confirmed. For example adding polynomial basis functions of degree 2 on elements 8 and 9 reduces the error to 0.00332, whereas adding polynomial basis functions of degrees 2 and 3 on element 9, only reduces the error to 0.00661.

4.2. Error estimation

In this subsection we demonstrate the feasibility of the approach when applied to a practical *a posteriori* error estimate [4]. Because this error estimate is designed specifically for use with linear elements the following description is restricted to using the sensitivity estimates to guide h-refinement. Possible extensions to other error estimates, including those that are better suited for guiding p-refinement, are discussed in Section 5.

The error estimate that we use here is that described in Section 3.2, and so expressions (41) and (43) are used to evaluate $\frac{dE}{d\underline{s}}$ in (34). The equation that we consider is

$$\frac{d^2u}{dx^2} + 3\frac{du}{dx} = 0, \tag{45}$$

with zero Dirichlet boundary conditions on the domain $(0,1)$. This linear problem also has a known solution and so it is possible to verify the accuracy of the error estimate and to assess the quality of different refinement strategies in terms of error reduction.

Table 3 presents the results, in terms of the actual errors achieved, of different refinement strategies based upon using the *a posteriori* error estimate in two distinct ways. The first strategy, similar to [18], simply refines a proportion of the elements with the highest error (in this case, a fixed proportion of 20% is used at each step). The second strategy also refines 20% of the elements at each step: this time selecting those with the greatest sensitivity. The initial mesh is uniform, containing just 10 elements (yielding an L_2 error of 0.000894), and it can be seen that for the first two steps both approaches select the same elements to refine. After the third refinement step however different elements are selected and, in this particular case, the strategy based upon sensitivity reduces the error at a faster rate.

Table 2
Local error and sensitivity to addition of basis functions of different polynomial degrees on the right-most elements of the domain following solution after the first refinement.

Element	L_2 error	degree = 2	degree = 3	degree = 4	degree = 5
8	0.00353	0.700e−3	0.088e−3	−0.193e−3	−0.013e−3
9	0.00468	0.661e−3	0.038e−3	−0.192e−3	0.043e−3
10	0.00181	–	–	0.125e−3	0.062e−3

Table 3
Actual error for refinement strategy based upon: (a) refining 20% of elements with the largest error and (b) refining 20% of elements with greatest sensitivity.

Refinement step	L_2 Error (a)	L_2 Error (b)
1	5.42×10^{-4}	5.42×10^{-3}
2	3.78×10^{-4}	3.78×10^{-4}
3	2.80×10^{-4}	2.71×10^{-4}
4	2.23×10^{-4}	2.09×10^{-4}

5. Discussion

5.1. Conclusions

We have presented extensions of the work in [7] by considering the use of high order polynomial basis functions and a practical *a posteriori* error estimate. The sample results presented in the previous section clearly illustrate that calculations of the sensitivity of the error to the addition of different degrees of freedom can provide valuable information to inform decisions relating to local mesh refinement. In [7] it is shown that these decisions can help to differentiate between whether to refine in h or in p, and in this paper we have also shown that sensitivity information can assist in deciding how much refinement to undertake in each element, or that it may provide an alternative criterion for deciding where to refine (compared to using the highest local error). Having demonstrated the potential value of this approach the challenge now is to understand how to incorporate this additional information into a robust algorithm. This is likely to be particularly complex since the sensitivity calculations provide a purely local perspective at the point $\underline{s} = \underline{0}$, and so can only be an indicator of where it is likely to be most beneficial to add further degrees of freedom. Such information will therefore need to be used in addition to, rather than instead of, existing techniques and criteria.

5.2. Nonlinear problems

In this paper we have considered only linear equations. The extension to nonlinear problems is relatively straightforward, in principle at least, through the use of standard discrete adjoint arguments [8,17,19]. For a nonlinear problem we can replace the discrete system (12) by a nonlinear algebraic system of the form $\underline{R}(\underline{u},\underline{s}) = \underline{0}$. Since this must be satisfied for all $\underline{s}$ it follows that

$$\frac{d\underline{R}}{d\underline{s}} = \frac{\partial \underline{R}}{\partial \underline{u}}\frac{\partial \underline{u}}{\partial \underline{s}} + \frac{\partial \underline{R}}{\partial \underline{s}} = 0, \tag{46}$$

and so, for any $\underline{\Psi} \in \Re^{N1}$,

$$\left(\underline{\Psi}^T \frac{\partial \underline{R}}{\partial \underline{u}}\right)\frac{\partial \underline{u}}{\partial \underline{s}} + \underline{\Psi}^T \frac{\partial \underline{R}}{\partial \underline{s}} = \underline{0}. \tag{47}$$

In particular, we may choose $\underline{\Psi}$ such that it satisfies

$$\underline{\Psi}^T \frac{\partial \underline{R}}{\partial \underline{u}} = \frac{\partial E}{\partial \underline{u}}, \tag{48}$$

so that, by (47),

$$\frac{\partial E}{\partial \underline{u}}\frac{\partial \underline{u}}{\partial \underline{s}} = -\underline{\Psi}^T \frac{\partial \underline{R}}{\partial \underline{s}}. \tag{49}$$

Substituting (49) into the total derivative of $E(\underline{u},\underline{s})$ (32), then yields:

$$\frac{dE}{d\underline{s}} = \frac{\partial E}{\partial \underline{u}}\frac{\partial \underline{u}}{\partial \underline{s}} + \frac{\partial E}{\partial \underline{s}} = -\underline{\Psi}^T \frac{\partial \underline{R}}{\partial \underline{s}} + \frac{\partial E}{\partial \underline{s}}. \tag{50}$$

Table 4
Actual local errors and computed sensitivities to local h-refinement when solving Eq. (51) using ten equally-spaced elements and $\epsilon = 0.01$.

Element	1	2	3	...	9	10
L_2 error	0.0038	0.0104	0.0076	...	0.0039	0.0010
Sensitivity	0.0004	−0.0028	0.0006	...	−0.0001	0.0132

This last expression is a generalization of (34) to nonlinear problems, requiring just a single linear solve for the adjoint variable $\underline{\Psi}$ (48).

To illustrate the potential use of expression (50) for nonlinear problems we consider the differential equation

$$u\frac{du}{dx} + \epsilon\frac{d^2u}{dx^2} = 0, \quad 0 < x < 1, \tag{51}$$

subject to Dirichlet boundary conditions that are consistent with the analytic solution $u(x) = 2\epsilon/(2\epsilon + x)$. Table 4 shows the exact local error and the sensitivity of this exact error to local h-refinement when ten equally-spaced elements are used and $\epsilon = 0.01$. The global L_2 error prior to refinement is 0.23405 and the contribution to this error is greatest in element 2. When this element is refined however the solution of the resulting finite element system yields a negligible reduction in the error (unchanged to five decimal places). Conversely, refinement of element 10, which has a much smaller local error but the greatest sensitivity, reduces the error quite substantially, to 0.07185. Hence we again see, for this example at least, the value of using the sensitivity information in selecting where to refine.

5.3. Further extensions

So far just one particular error estimation technique has been considered, however the specific error estimate used in Section 3.2 may also be generalized. For example, the goal-based approach of [19] assumes that one is primarily interested in some quantity which depends upon the solution u of (1), $I(u)$ say. An estimate is developed which allows the calculation of this quantity on a given grid to be updated (based upon the solution of an additional, adjoint, problem on this grid). Our approach allows us to compute the sensitivity of this estimate to the use of additional basis functions. When $I(u)$ is a linear functional this sensitivity calculation is simplified further. Generalizations to other error estimators, such as [2,20], may also be considered in a similar manner.

Finally, we acknowledge the preliminary nature of this work, and the need to implement and test the sensitivity approach on problems in two and three space dimensions. In particular, we are keen to incorporate this within a discontinuous Galerkin framework, e.g. [12,16], since this allows the maximum flexibility in hp-refinement in higher dimensions. Clearly there is a wide range of fluid flow problems to which the approach may be applied – the overall goal being to develop more robust and more efficient adaptive software.

Acknowledgement

The second author would like to acknowledge support under NSF-IIS-0914564.

References

[1] Ainsworth M, Senior B. An adaptive refinement strategy for hp-finite element computations. Appl Numer Math 1998;26:165–78.
[2] Ainsworth M, Oden JT. A posteriori error estimation in finite element analysis. Comput Methods Appl Mech Eng 1997;142:1–88.
[3] Bangerth W, Rannacher R. Adaptive finite element methods for differential equations. Birkhäuser-Verlag; 2003.
[4] Bank RE, Weiser A. A posteriori error estimators for elliptic partial differential equations. Math Comput 1985;44:283–301.
[5] Babuska I, Rheinboldt WC. A-posteriori error estimates for finite-element method. Int J Numer Methods Eng 1978;12:1597–615.
[6] Deville MO, Mund EH, Fischer PF. High order methods for incompressible fluid flow. Cambridge University Press (UK); 2002.
[7] Frazier JD, Jimack PK, Kirby RM. On the use of adjoint-based sensitivity estimates to control local mesh refinement. Commun Comput Phys 2010;7:631–8.
[8] Giles MB, Pierce NA. An introduction to the adjoint approach to design. Flow Turbul Combust 2002;65:393–415.
[9] Gunzburger MD. Sensitivities, adjoints and flow optimization. Int J Numer Methods Fluids 1999;31:53–78.
[10] Karniadakis GE, Sherwin SJ. Spectral/hp element methods for CFD. 2nd ed. Oxford University Press (UK); 2005.
[11] Kurtz J, Demkowicz L. A fully automatic hp-adaptivity for elliptic PDEs in three dimensions. Comput Methods Appl Mech Eng 2007;196:3534–45.
[12] Lu H, Berzins M, Goodyer CE, Jimack PK. High order discontinuous Galerkin method for EHL line contact problems. Commun Numer Methods Eng 2005;21:643–50.
[13] Maute K, Nikbay K, Farhat C. Sensitivity analysis and design optimization of three-dimensional non-linear aeroelastic systems by the adjoint method. Int J Numer Methods Eng 2003;56:911–33.
[14] Rosam J, Jimack PK, Mullis AM. A fully implicit, fully adaptive time and space discretization method for phase-field simulation of binary alloy solidification. J Comput Phys 2007;225:1271–87.
[15] Schwab Ch. p- and hp-Finite element methods: theory and applications to solid and fluid mechanics. Oxford University Press (USA); 1999.
[16] Sherwin SJ, Kirby RM, Peiro J, Taylor RL, Zienkiewicz OC. On 2D elliptic discontinuous Galerkin methods. Int J Numer Methods Eng 2006;65:752–84.
[17] Schneider R, Jimack PK. Anisotropic mesh adaption based on a posteriori estimates and optimisation of node positions. In: Lube G, Rapin G, editors. Proceedings of BAIL 2006: international conference on boundary and interior layers. Germany: University of Gottingen; 2006.
[18] Stevenson R. Optimality of a standard finite element method. Found Comput Math 2007;7:245–69.
[19] Venditti DA, Darmofal DL. Grid adaptation for functional outputs: application to two-dimensional inviscid flows. J Comput Phys 2002;176:40–69.
[20] Zienkiewicz OC, Zhu JZ. A simple error estimator and adaptive procedure for practical engineering analysis. Int J Numer Methods Eng 1987;24:337–57.