On the Use of Adjoint-Based Sensitivity Estimates to Control Local Mesh Refinement

Joseph D. Frazier[1], Peter K. Jimack[2,*] and Robert M. Kirby[1]

[1] *School of Computing, University of Utah, Salt Lake City, UT, USA*
[2] *School of Computing, University of Leeds, Leeds, UK*

Abstract. The goal of efficient and robust error control, through local mesh adaptation in the computational solution of partial differential equations, is predicated on the ability to identify in an *a posteriori* way those localized regions whose refinement will lead to the most significant reductions in the error. The development of *a posteriori* error estimation schemes and of a refinement infrastructure both facilitate this goal, however they are incomplete in the sense that they do not provide an answer as to where the maximal impact of refinement may be gained or what type of refinement – elemental partitioning (h-refinement) or polynomial enrichment (p-refinement) – will best lead to that gain. In essence, one also requires knowledge of the sensitivity of the error to both the location and the type of refinement. In this communication we propose the use of adjoint-based sensitivity analysis to discriminate both where and how to refine. We present both an adjoint-based and an algebraic perspective on defining and using sensitivities, and then demonstrate through several one-dimensional model problem experiments the feasibility and benefits of our approach.

Key words: adaptivity, error estimation, hp-finite elements, sensitivity, discrete adjoint

1 Introduction

The use of adaptivity is widely accepted as an essential component in the efficient and reliable implementation of finite element (FE) algorithms for the solution of a wide range of partial differential equations (PDEs) [4, 12]. In particular, for problems whose solution exhibits steep fronts or sharp layers (*e.g.* boundary layers), so-called *h*-refinement is usually an appropriate strategy for producing a mesh size that is of the same order as the feature in question [3, 9, 10]. Conversely, once the mesh size is suitable, polynomial enrichment (also known as *p*-refinement) is generally the most accurate and cost effective form of refinement [1, 11]. In the last three decades, as practitioners have sought to

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

develop algorithms that both incorporate and drive these adaptive procedures fully automatically, there has been an enormous research focus on the development of reliable and accurate *a posteriori* error estimates [2–5]. These estimates typically provide information on the total error of a computed solution, the distribution of this error throughout the domain and/or information on the error in some derived solution-dependent quantity. Traditionally, this information has been used both to decide when a computed solution is of sufficient accuracy for no further calculations to be required and, in the case where the accuracy is deemed to be insufficient, to decide how to adapt the FE trial space. Typically, the approach used is to refine in those regions where the estimated error is the greatest: either by refining those elements whose error is within a certain percentage of the total [3], or refining the elements with the greatest error until the cumulative total is within a given percentage of the estimated error [12]. The criteria for deciding whether this refinement should be in h or p is rather more varied but is typically based upon some form of estimate as to which is likely to be the most beneficial [1,8].

This short communication introduces a new approach for controlling local adaptivity within an hp-finite element code. The goal is to explore whether it is possible to use more information from the estimated error for deciding both where to adapt and/or how to adapt. The approach is based upon the assumption that we have a reliable *a posteriori* error estimate (E say), that is easily computable, and then to attempt to compute the sensitivity of this estimate to the addition of further p- or h- degrees of freedom. A standard adjoint approach is used to compute these sensitivities efficiently and it is demonstrated that an adaptive strategy based upon these values can have advantages over other, more traditional, approaches.

2 Notation and Formulation

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

$$\frac{d}{dx}(a(x)\frac{du}{dx})+b(x)\frac{du}{dx}+c(x)=0 \tag{2.1}$$

subject to Dirichlet boundary conditions on the domain (x_0,x_N), where $a(x)>0$. Suppose the domain is divided into N intervals, $x_0<x_1<...<x_{N-1}<x_N$, and let $\{\phi_0^1,...,\phi_N^1\}$ be the usual basis (of local hat functions) for the space of continuous piecewise polynomials of degree one on this mesh. For simplicity assume that $u(x_0)=u(x_N)=0$, so the corresponding piecewise linear FE trial function takes the form:

$$u^1(x)=\sum_{i=1}^{N-1}u_i^1\phi_i^1(x), \tag{2.2}$$

where the coefficients u_i^1 are prescribed by the usual FE weighted residual equations:

$$R_j=\int_{x_0}^{x_N}\left(-a(x)\frac{du^1}{dx}\frac{d\phi_j^1}{dx}+b(x)\frac{du^1}{dx}\phi_j^1+c(x)\phi_j^1\right)dx=0 \text{ for } j=1,...,N-1. \tag{2.3}$$

Now consider the possibility of adapting a computed solution of this form by increasing the polynomial degree locally (p-refinement) or by bisecting some of the elements (h-refinement). On element e, for $e=1,...,N$, let $\phi_e^2(x)$ be a bubble function of degree 2 and let $\psi_e(x)$ be the piecewise linear hat function associated with bisecting element e (i.e., $\psi_e(x)$ 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). Note that it is possible to write (2.2) as:

$$u^1(x)=\sum_{i=1}^{N-1} u_i^1\phi_i^1(x)+\sum_{i=1}^{N}\left(u_i^2\phi_i^2(x)+v_i\psi_i(x)\right) \tag{2.4}$$

provided that we impose $0=u_1^2=...=u_N^2=v_1=...=v_N$. If we now make the definitions $\underline{u}\in\Re^{N-1}=(u_1^1,...,u_{N-1}^1)^T$ and $\underline{s}\in\Re^{2N}=(u_1^2,...,u_N^2,v_1,...,v_N)^T$, then equations (2.3) may be expressed as

$$\underline{R}(\underline{u},\underline{s})=\underline{0}, \tag{2.5}$$

where $\underline{s}=\underline{0}$. The system (2.5) is the same as (2.3) except u^1 is now of the form (2.4) and the test functions are ϕ_j^2 and ψ_j for $j=1,...,N$, as well as ϕ_j^1 for $j=1,...,N-1$. Having computed the finite element solution (i.e. solved (2.3)) 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 (see for example [2,4,5]).

Suppose that we can evaluate $\frac{dE}{d\underline{s}}$. Then it would be possible to assess to which of these additional degrees of freedom the error is most sensitive, and to use this information to decide both where to refine and how to refine. Of course, the practicality of this approach is dependent upon being able to compute $\frac{dE}{d\underline{s}}$ both reliably and efficiently. Fortunately this may be achieved through the use of adjoint-based methods [6,7].

3 Sensitivity via the adjoint solution

Suppose we have successfully computed an FE solution and a corresponding error estimate. That is, we have solved $\underline{R}(\underline{u},\underline{s}=\underline{0})=\underline{0}$ and then computed $E(\underline{u},\underline{s}=\underline{0})$. Note that

$$\frac{dE}{d\underline{s}}=\frac{\partial E}{\partial\underline{u}}\cdot\frac{\partial\underline{u}}{\partial\underline{s}}+\frac{\partial E}{\partial\underline{s}}. \tag{3.1}$$

However, we are only interested in calculating E when the FE equations are satisfied, which means that $\underline{R}(\underline{u},\underline{s})=0$ for all $\underline{s}$, and so

$$\frac{d\underline{R}}{d\underline{s}}=\frac{\partial\underline{R}}{\partial\underline{u}}\cdot\frac{\partial\underline{u}}{\partial\underline{s}}+\frac{\partial\underline{R}}{\partial\underline{s}}=\mathbf{0}. \tag{3.2}$$

From (3.2) it follows that, for any $\underline{\Psi}\in\Re^{2N}$,

$$(\underline{\Psi}^T\frac{\partial\underline{R}}{\partial\underline{u}})\cdot\frac{\partial\underline{u}}{\partial\underline{s}}+\underline{\Psi}^T\frac{\partial\underline{R}}{\partial\underline{s}}=\underline{0}. \tag{3.3}$$

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

$$\underline{\Psi}^T \frac{\partial \underline{R}}{\partial \underline{u}} = \frac{\partial E}{\partial \underline{u}}, \tag{3.4}$$

so that combining (3.1), (3.3) and (3.4) yields:

$$\frac{dE}{d\underline{s}} = -\underline{\Psi}^T \frac{\partial \underline{R}}{\partial \underline{s}} + \frac{\partial E}{\partial \underline{s}}. \tag{3.5}$$

Note that using (3.5), the derivative $\frac{dE}{d\underline{s}}$ may be computed by: evaluating $\frac{\partial \underline{R}}{\partial \underline{u}}$, evaluating $\frac{\partial E}{\partial \underline{u}}$, solving (3.4), evaluating $\frac{\partial \underline{R}}{\partial \underline{s}}$ and evaluating $\frac{\partial E}{\partial \underline{s}}$. Significantly, there is no need to compute $\frac{\partial \underline{u}}{\partial \underline{s}}$. Furthermore $\frac{\partial \underline{R}}{\partial \underline{u}}$, the Jacobian of the FE equations, will already have been computed as part of the solution process and the other expressions are straightforward to evaluate: where $\frac{\partial E}{\partial \underline{u}}$ and $\frac{\partial E}{\partial \underline{s}}$ depending upon the precise choice of error estimate, E.

It is also possible to consider the above argument from a more algebraic perspective for the finite element equations discussed in the previous section. For example, the system (2.3) may be expressed in matrix from as

$$K^{11}\underline{u}^1 = \underline{f}^1, \tag{3.6}$$

where $\underline{u}^1$ is the vector of unknown coefficients. Similarly, the system (2.5) may be written in matrix form as

$$\begin{bmatrix} K^{11} & K^{12} \\ K^{21} & K^{22} \end{bmatrix} \begin{bmatrix} \underline{u} \\ \underline{s} \end{bmatrix} = \begin{bmatrix} \underline{f}^1 \\ \underline{f}^2 \end{bmatrix}, \tag{3.7}$$

where one may view the solution of the smaller problem (3.6) as being equivalent to solving the larger problem (3.7) but with the vector $\underline{s}$ constrained to be $\underline{0}$. That is, when $\underline{s} = \underline{0}$ in (3.7), $\underline{u} = \underline{u}^1$. This view may be extended by thinking of $\underline{u}$ in (3.7) as depending upon the prescribed value of $\underline{s}$. That is, $\underline{u} = \underline{u}(\underline{s})$, where $\underline{u}(\underline{0}) = \underline{u}^1$. The goal of the sensitivity analysis is therefore to calculate the sensitivity of $\underline{u}(\underline{s})$ in the region of $\underline{s} = \underline{0}$ (this in turn allows the sensitivity of E to be obtained via (3.1)).

Note that in (3.1) the terms $\frac{dE}{d\underline{s}}$, $\frac{\partial E}{\partial \underline{u}}$ and $\frac{\partial E}{\partial \underline{s}}$ are all assumed to be row vectors: re-writing (3.1) with each of these terms assumed to be column vectors gives

$$\frac{dE}{d\underline{s}} = \left(\left(\frac{\partial E}{\partial \underline{u}} \right)^T \frac{\partial \underline{u}}{\partial \underline{s}} \right)^T + \frac{\partial E}{\partial \underline{s}}. \tag{3.8}$$

Observing that, from (3.7), $\underline{u}(\underline{s}) = (K^{11})^{-1}(\underline{f}^1 - K^{12}\underline{s})$, one can compute the derivative of $\underline{u}$ with respect to $\underline{s}$ and so it follows from (3.8) that,

$$\frac{dE}{d\underline{s}} = -(K^{12})^T (K^{11})^{-T} \frac{\partial E}{\partial \underline{u}} + \frac{\partial E}{\partial \underline{s}}. \tag{3.9}$$

This last expression is equivalent to (3.5) in the case where R is the discretization of a linear elliptic operator. Furthermore, in the special case where the PDE is self-adjoint, so the stiffness matrix is symmetric, (3.9) simplifies further to

$$\frac{dE}{d\underline{s}} = -K^{21}(K^{11})^{-1}\frac{\partial E}{\partial \underline{u}} + \frac{\partial E}{\partial \underline{s}}. \tag{3.10}$$

4 Some Corroborating Numerical Tests

The first test problem that we consider in this section takes the form:

$$-\frac{d^2u}{dx^2} = f(x) \quad -1 < x < 1, \tag{4.1}$$

where $f(x)$ and the Dirichlet boundary conditions at $x = -1$ and $x = 1$ are chosen so as to permit the exact solution $u(x) = (\cos(n\pi x))/(1+25x^2)$. In the calculations described here, we take $n = 4$ and the sensitivities of the *exact* error are computed with respect to both h-refinement and p-refinement on each element. It is also important to remark that all calculations are undertaken using high degree quadrature to ensure that quadrature errors are not a factor in any of these results. In what follows, we compare h-refinement, through adding a linear bubble function, versus p-refinement, through adding a polynomial (quadratic) bubble function (ψ_e and ϕ_e^2 respectively) as previously described.

Initially we computed a piecewise linear solution on ten equally-spaced elements. The resulting L_2 error, along with the errors and sensitivities on each element, are shown in Table 1(top), where $\tilde{E}_e$ is L_2^2 error on each element. The table shows that the largest L_2^2 error is found on the centre two elements, and therefore the conventional adaptive approach would be to refine these elements first. On the other hand, the sensitivities suggest that most benefit will be gained by refining elements 3 and 8 (with slightly more impact resulting from p-refinement). Table 1(bottom) shows the errors actually observed in the computed solutions when different refinement strategies are compared. It is apparent from this table that refining the elements with the largest error is not the best short-term strategy in this case and that, as predicted by the sensitivity calculations, it is better to refine elements 3 and 8. Furthermore, also as predicted by the sensitivity calculations, it is better to employ p-refinement in this case.

A further set of computational tests were undertaken for this problem, this time to illustrate performance on non-equally-spaced grids. Initially (4.1) was solved using five equally-spaced piecewise linear elements. Table 2 shows the resulting errors and sensitivities. This table clearly shows the need to refine the middle element first and, on this occasion, it is predicted by the sensitivities that h-refinement will be superior to p-refinement. This is indeed confirmed by experiment, where h-refinement of the centre element leads to a new total L_2 error of 0.17167 whereas p-refinement yields a total L_2 error of 0.21070. We may now repeat the error estimation and sensitivity calculations for the solution on the new mesh (consisting of six piecewise linear elements of two different

Table 1: (Top) For a piecewise linear solution of (4.1) using ten elements the global L_2 error, E, is 0.089354, with contributions from each element as shown below, followed by the entry of $\frac{dE^2}{ds}$ corresponding to h- and p-refinement respectively; (Bottom) Actual total L_2 errors obtained following different refinement strategies .

Elements	1 and 10	2 and 9	3 and 8	4 and 7	5 and 6
Error, $\vec{E}_e$	3.835×10^{-5}	2.854×10^{-4}	8.935×10^{-4}	8.657×10^{-4}	$\mathbf{1.909\times10^{-3}}$
h-Sensitivity	-9.833×10^{-4}	2.734×10^{-3}	-4.790×10^{-3}	4.210×10^{-3}	1.934×10^{-3}
p-Sensitivity	-9.905×10^{-4}	2.758×10^{-3}	$\mathbf{-4.841\times10^{-3}}$	4.283×10^{-3}	1.824×10^{-3}

Refinement scheme	Total L_2 error	Refinement scheme	Total L_2 error
h-ref (elements 3 and 8)	0.079548	p-ref (elements 3 and 8)	**0.078911**
h-ref (elements 4 and 7)	0.081982	p-ref (elements 4 and 7)	0.081293
h-ref (elements 5 and 6)	0.087786	p-ref (elements 5 and 6)	0.087990

Table 2: For a piecewise linear solution of (4.1) using five elements the total L_2 error, E, is 0.55404, with contributions from each element as shown below, followed by the entry of $\frac{dE^2}{ds}$ corresponding to h- and p-refinement respectively .

Element	1	2	3	4	5
Error, $\vec{E}_e$	1.611×10^{-3}	1.122×10^{-2}	$\mathbf{2.813\times10^{-1}}$	1.122×10^{-2}	1.611×10^{-3}
h-Sensitivity	8.417×10^{-3}	-2.251×10^{-2}	$\mathbf{-1.217\times10^{-1}}$	-2.251×10^{-2}	8.417×10^{-3}
p-Sensitivity	8.467×10^{-3}	-2.208×10^{-2}	-1.187×10^{-1}	-2.208×10^{-2}	8.467×10^{-3}

sizes). The results of this are shown in Table 3. This shows that the error is now greatest in elements 2 and 5, and the sensitivity calculations imply that these are also the best elements to refine. Furthermore, it is predicted that h-refinement will again be superior to p-refinement in this case. Additional numerical tests verify all of these conclusions: for example, the total L_2 error after h-refinement of elements 2 and 5 is 0.10276, as opposed to 0.10630 using p-refinement on these elements.

Finally in this section we present illustrative results for the solution of a second test problem, namely:

$$-\left(\frac{d}{dx}(\pi+x)\frac{du}{dx}\right)=f(x), \quad -1<x<1, \tag{4.2}$$

where $f(x)$ and the Dirichlet boundary conditions are chosen so that $u(x)=\sin(2\pi x)$. We again see that the sensitivity calculations provide excellent predictions as to how and where to refine. For example, when five equally-spaced linear elements are used, the computed total L_2 error is 0.50505, and Table 4 shows the errors and sensitivities for each element. The computed errors for different refinements of the mesh rank just as predicted by the sensitivities in Table 4. For example, p-refinement of element 1 leads to the smallest total L_2 error (0.40076), followed by h-refinement of element 1 (0.40559), then p-refinement of element 5 (0.40943), h-refinement of element 5 (0.41588), *etc.*

Table 3: For a piecewise linear solution of (4.1) using six (unequally-spaced) elements the total L_2 error, E, is 0.17617, with contributions from each element as shown below, followed by the entry of $\frac{dE^2}{d\underline{s}}$ corresponding to h- and p-refinement respectively .

Elements	1 and 6	2 and 5	3 and 4
Error, $\tilde{E}_e$	1.6×10^{-3}	$\mathbf{1.12\times10^{-2}}$	1.9×10^{-3}
h-Sensitivity	8.4×10^{-3}	$\mathbf{-2.25\times10^{-2}}$	1.9×10^{-3}
p-Sensitivity	8.5×10^{-3}	-2.21×10^{-2}	1.8×10^{-3}

Table 4: For a piecewise linear solution of (4.2) using five elements the total L_2 error, E, is 0.50505, with contributions from each element as shown below, followed by the entry of $\frac{dE^2}{d\underline{s}}$ corresponding to h- and p-refinement respectively .

Element	1	2	3	4	5
Error, $\tilde{E}_e$	$\mathbf{1.012\times10^{-1}}$	2.74×10^{-2}	2.8×10^{-3}	4.27×10^{-2}	8.09×10^{-2}
h-Sensitivity	-7.24×10^{-2}	3.83×10^{-2}	-1.9×10^{-3}	-4.41×10^{-2}	6.58×10^{-2}
p-Sensitivity	$\mathbf{-7.32\times10^{-2}}$	3.85×10^{-2}	-2.0×10^{-3}	-4.468×10^{-2}	8.467×10^{-3}

A significant number of additional computational tests have been undertaken on these and other equations. In each case they demonstrate that the sensitivity calculations give reliable predictions as to how and where to refine. Indeed, tests have been carried out with p-refinement up to degree ten, and numerous examples have been found where refining the elements with the largest error is less effective than refining different elements, for which the sensitivity is greater.

5 Discussion

In this paper we suggest that using the sensitivity of an error estimate, with respect to additional FE degrees of freedom, may have potential value in informing the adaptive strategy used within an hp-refinement algorithm. In particular, we note that simply refining where the error is greatest is not always the optimal strategy, and that assessing where the estimated error is most sensitive to the addition of degrees of freedom can provide an effective alternative criterion for deciding where to refine. Furthermore, we have demonstrated that this approach may also be used successfully to decide which of two different refinement procedures (h-refinement or p-refinement) should be employed. Future work will consider these observations within the context of an overall refinement algorithm. For example, rather than adapting a percentage of the elements with the greatest errors one might adapt a percentage of the elements with the greatest sensitivities. It will also be necessary to consider the use of a wide variety of *a posteriori* error estimates:

preliminary work using a simple estimate from [5] is very encouraging.

Finally we make three observations on sensitivity values that are computed throughout this work. Firstly, the sign of $\frac{dE}{ds_j}$ is not important (only its magnitude). Secondly, the basis that is used for the sensitivity calculations need not be the same as that is used for subsequent FE calculations after refinement. For example, when h-refinement occurs we always use the usual basis functions with support over just two elements rather than a hierarchical basis. Finally, and perhaps most importantly, it is essential to appreciate that the values of the entries in $\frac{dE}{ds_j}$ depend upon the scaling of the basis functions that are used. In order to be able to make a comparison between h-refinement and p-refinement, for example, we scaled the basis functions such that on each element e: $\|\phi_e^2\|_2^2 = \|\psi_e\|_2^2$. Note however that these values must be different for different choices of e when the element sizes are different (with $\|\psi_e\|_2^2$ proportional to the length of e).

Acknowledgements

The work of the third author was supported in part by NSF Career Award CCF0347791.

References

[1] M. Ainsworth and B. Senior, An adaptive refinement strategy for hp-finite element computations, Applied Numerical Mathematics, 26 (1998), 165-178.

[2] M. Ainsworth and J.T. Oden, A posteriori error estimation in finite element analysis, Computer Methods in Applied Mechanics and Engineering, 142 (1997), 1-88.

[3] T. Apel, S. Grosman, P.K. Jimack, and A. Meyer, A new methodology for anisotropic mesh refinement based upon error gradients. Applied Numerical Mathematics, 50 (2004), 329-341.

[4] W. Bangerth and R. Rannacher, Adaptive Finite Element Methods for Differential Equations, Birkhäuser-Verlag, 2003.

[5] R.E. Bank and A. Weiser, A posteriori error estimators for elliptic partial differential equations, Mathematics of Computation, 44 (1985), 283-301.

[6] M.B. Giles and N.A. Pierce, An introduction to the adjoint approach to design, Flow Turbulence and Combustion, 65 (2002), 393-415.

[7] M.D. Gunzburger, Sensitivities, adjoints and flow optimization, International Journal for Numerical Methods in Fluids, 31 (1999), 53-78.

[8] J. Kurtz and L. Demkowicz, A fully automatic hp-adaptivity for elliptic PDEs in three dimensions, Computer Methods in Applied Mechanics and Engineering, 196 (2007), 3534-3545.

[9] R. Lohner and J.D. Baum, Adaptive h-refinement on 3d unstructured grids for transient problems, International Journal for Numerical Methods in Fluids, 14 (1992), 1407-1419.

[10] J. Rosam, P.K. Jimack and A.M. Mullis, A fully implicit, fully adaptive time and space discretization method for phase-field simulation of binary alloy solidification, Journal of Computational Physics, 225 (2007), 1271-1287.

[11] Ch. Schwab, p- and hp- Finite element methods: theory and applications to solid and fluid mechanics, Oxford University Press (USA), 1999.

[12] R. Stevenson, Optimality of a standard finite element method, Foundations of Computational Mathematics, 7 (2007), 245-269.