Sang-Ho Lee
Ted Belytschko
Robert R. McCormick School of Engineering and Applied Science
Department of Civil Engineering
The Technological Institute
Northwestern University
Evanston, IL 60208-3109

H-Adaptive Methods for Nonlinear Dynamic Analysis of Shell Structures

The implementation and application of h-*adaptivity in an explicit finite element program for nonlinear structural dynamics is described. Particular emphasis is placed on developing procedures for general purpose structural dynamics programs and efficiently handling adaptivity in shell elements. New projection techniques for error estimation and projecting variables on new meshes after fission or fusion are described. Several problems of severe impact are described.* © *1995 John Wiley & Sons, Inc.*

INTRODUCTION

H-adaptivity is a technique for enhancing the accuracy of nonlinear finite element simulations; moreover, it eliminates the user's need to anticipate where refinement will be needed to capture nonlinear phenomena adequately. Two characteristics of nonlinear response favor adaptivity:

1. localization of deformation, which means that high strains occur in very small portions of a component, for example in a buckled thin-walled beam, the high strains will be localized areas that appear as kinks;
2. the unpredictability of where high strains are going to occur.

Adaptive finite element methods place the unknowns, that is, the degrees-of-freedom, where they are needed, so that the most accurate solution can be obtained for a given amount of resources. This is of particular importance in buckling and localization problems, because the areas where localization occur are not known a priori.

This article deals with the development of *h*-adaptivity for general purpose structural dynamics programs with explicit time integration such as DYNA3D (Hallquist, 1983). This class of programs is widely used for accident analysis, automotive crashworthiness, etc.

Adaptive methods have reached an advanced stage of development in linear stress analysis and in computational fluid dynamics. Extending the concepts developed in those contexts to general purpose structural dynamics programs is not straightforward. General purpose programs usually contain large libraries of elements and materials and interface with preprocessors and postprocessors. The design of an effective adaptive strategy must take these factors into account for a general purpose program.

The application of many of the error criteria reported in the literature is quite difficult for shell elements. Shell finite elements for nonlinear

Received October 26, 1994; Accepted December 16, 1994.
Shock and Vibration, Vol. 2, No. 3, pp. 193–204 (1995)

CCC 1070-9622/95/030193-12

analysis are usually developed from the degenerated continuum approach. For these discretizations, residual methods of error criterion in which the error is related to the unbalance in the momentum equation is almost impossible to apply because the actual surface of the shell, and hence the divergence operator in the momentum equation, are never explicitly defined. Similarly, projection error criteria, such as the Zienkiewicz–Zhu criterion (1987, 1992a, 1992b), are difficult to apply directly to the stresses.

Here we have focused on some techniques for shell elements that lead to simpler algorithms and data structures. The motivation is that simplicity is the key toward successful implementation in a general purpose finite element program. It is the simplicity of finite element methods and particularly explicit finite element programs that has made them so robust and useful. The intent here was to develop adaptivity so that this simplicity is maintained as much as possible. This is accomplished by using projections based on moving least square interpolations. These provide so-called smooth solutions that can be used for two purposes:

1. as a reference solution that can be used to make error estimates by comparing it with the finite element solutions, as in the Zienkiewicz–Zhu criterion;
2. to provide nodal coordinates and element variables for the nodes and elements created by the fission process.

Another difficulty in the implementation of adaptivity is the burdensome requirements of data generation and manipulation and the long running times. Therefore, we have developed several schemes for reduction in running time.

The moving least square method for solution recovery is described in the following section. Then, an error criterion is described. This is followed by a section on adaptive strategies for dealing with the evolution of the mesh. Some numerical examples are given. Finally, there is a section for discussion and conclusion.

MOVING LEAST SQUARE METHOD (MLSM) FOR ERROR ESTIMATE

In the finite element approximation, the error is defined as

$$\mathbf{e} = \mathbf{s} - \mathbf{s}^h \tag{1}$$

where $\mathbf{s}$ is an exact solution such as displacement $\mathbf{u}$, strain $\boldsymbol{\varepsilon}$, or stress $\boldsymbol{\sigma}$, and $\mathbf{s}^h$ is the finite element solution. In most practical nonlinear dynamic problems, the exact solution is unknown, so that the exact solution $\mathbf{s}$ is replaced by approximated exact solution $\mathbf{s}^*$, which is often called projected solution. Then the error is defined by

$$\mathbf{e} \approx \mathbf{e}^* = \mathbf{s}^* - \mathbf{s}^h. \tag{2}$$

The MLS interpolant $\mathbf{s}^*(\mathbf{x})$ of the solution $\mathbf{s}^h(\mathbf{x})$ is defined in the domain Ω by

$$\mathbf{s}^*(\mathbf{x}) = \sum_j^m p_j(\mathbf{x}) a_j(\mathbf{x}) \equiv \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) \tag{3}$$

where $\mathbf{p}(\mathbf{x})$ are polynomials in the space coordinates for example, $\mathbf{p}^T(\mathbf{x}) = [1, x, y, z]$ and $m = 4$ in three-dimensional space, and $\mathbf{a}(\mathbf{x})$ are the corresponding unknown coefficients. The coefficients $a_j(\mathbf{x})$ in the above equation are also functions of $\mathbf{x}$; $\mathbf{a}(\mathbf{x})$ are obtained at any point $\mathbf{x}$ by minimizing a weighted, discrete L_2 norm as follows:

$$\mathcal{J} = \sum_I^n w(\mathbf{x} - \mathbf{x}_I)[\mathbf{p}^T(\mathbf{x}_I)\mathbf{a}(\mathbf{x}) - s_I^h]^2 \tag{4}$$

where n is the number of sampling points in the neighborhood (domain of influence) of $\mathbf{x}$, $w(\mathbf{x} - \mathbf{x}_I)$ is a weight function that is nonzero over the domain of influence, and s_I^h is the finite element solution at $\mathbf{x} = \mathbf{x}_I$.

Nodal displacement values are used to recover the approximated exact displacement solution, whereas for strains or stresses the values evaluated at the reduced integration points are used to increase the accuracy of the projected solution. Note that to avoid singularity of the linear equations for $\mathbf{a}(\mathbf{x})$, the number of sampling points n should be equal to or greater than m.

The stationary condition of $\mathcal{J}$ in (4) with respect to $\mathbf{a}(\mathbf{x})$ leads to the following linear relation:

$$\mathbf{A}(\mathbf{x})\mathbf{a}(\mathbf{x}) = \mathbf{B}(\mathbf{x})\mathbf{s}^h \tag{5}$$

or

$$\mathbf{a}(\mathbf{x}) = \mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{s}^h \tag{6}$$

where $\mathbf{A}(\mathbf{x})$ and $\mathbf{B}(\mathbf{x})$ are the matrices defined by

$$\mathbf{A}(\mathbf{x}) = \sum_{I}^{n} w_I(\mathbf{x})\mathbf{p}^T(\mathbf{x}_I)\mathbf{p}(\mathbf{x}_I), \quad w_I(\mathbf{x}) \equiv w(\mathbf{x} - \mathbf{x}_I) \tag{7}$$

$$\mathbf{B}(\mathbf{x}) = [w_1(\mathbf{x})\mathbf{p}(\mathbf{x}_1), w_2(\mathbf{x})\mathbf{p}(\mathbf{x}_2), \ldots, w_n(\mathbf{x})\mathbf{p}(\mathbf{x}_n)] \tag{8}$$

$$\mathbf{s}^{h^T} = [s_1^h, s_2^h, \ldots, s_n^h]. \tag{9}$$

Hence, the approximated exact solution at any point $\mathbf{x}$ is obtained by

$$s^*(\mathbf{x}) = \sum_{I}^{n} \sum_{j}^{m} p_j(\mathbf{x})(\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x}))_{jI} s_I^h. \tag{10}$$

The weight function $w(\mathbf{x} - \mathbf{x}_I)$ is chosen to be relatively large for the $\mathbf{x}_I$ close to $\mathbf{x}$, and relatively small as $\|\mathbf{x} - \mathbf{x}_I\|$ increases; in other words, they should decrease in magnitude at the distance from $\mathbf{x}$ to $\mathbf{x}_I$ increases. Therefore, the weight function is considered to be dependent only on the distance between two points $\mathbf{x}$ and $\mathbf{x}_I$ as follows:

$$w(\mathbf{x} - \mathbf{x}_I) = w_I(d) \tag{11}$$

where $d = \|\mathbf{x} - \mathbf{x}_I\|$ is the distance between two points.

In this study, we used the following bell-shaped exponential weight function:

$$w_I(d_I^{2k}) = \begin{cases} \dfrac{e^{-(d_I/c)^{2k}} - e^{-(d_{mI}/c)^{2k}}}{1 - e^{-(d_{mI}/c)^{2k}}}, & d_I \le d_{mI} \\ 0, & d_I > d_{mI} \end{cases} \tag{12}$$

where k is a positive constant; c is a constant that controls the relative weights; d_{mI} is the size of the support for the weight function and determines the domain of influence of $\mathbf{x}_I$. Note that when c decreases, we obtain larger weights on points $\mathbf{x}_I$ close to $\mathbf{x}$ and lower weights on points far removed from $\mathbf{x}$. In this study, we used $k = 1$, and $c = \alpha h_{\max}$ where $h_{\max}$ is the maximum size of elements and $1 \le \alpha \le 2$.

REMARK 1. The MLSM can recover projected solutions accurately even in irregular meshes or the meshes with partial refinement by adaptivity.

REMARK 2. In the MLSM, the accuracy of recovered solution can be further improved with a larger domain of influence, however big d_{mI} often deteriorates the accuracy of solution in highly nonlinear deformations of shell structures unless the distance $d_I = \|\mathbf{x} - \mathbf{x}_I\|$ is measured along the surface of the structure.

REMARK 3. The result of MLSM is identical to that of the solution recovery technique by Zienkiewicz and Zhu (1992a) when the weight function is constant and the number of sampling points in the domain of influence equals the number of sampling points in the local patch—in the Z–Z solution recovery technique, the patch consists of the elements that surround a recovery point (nodal point $\mathbf{x}$).

ERROR CRITERION

For dynamic finite element analyses, the sources of error are:

1. time integration;
2. space discretization;
3. numerical integration over the element domain to compute internal forces.

In explicit codes, time steps are small to assure the stability of solutions, so error from time integration is not a major issue except for certain problems such as those with large rigid body motion or strain softening. We will therefore focus on an error criterion's ability to predict errors from spatial discretization and numerical spatial integration. A useful error criterion should indicate when the error from these two major contributions is large.

Belytschko, Wong, and Plaskacz, (1989) and Belytschko and Yeh (1993) used the change in angles between shell elements as a measure of error. This is a useful criterion for highly nonlinear three-dimensional problems when bending is the dominant behavior, but it is ineffective for problems where membrane behavior dominates. For example, it cannot indicate the error for membrane bending, in which the angle change is zero for all elements. Belytschko et al. (1992) also added a error criterion based on hourglass energy for the one-point integration quadrature element, which represents error due to numerical integration.

A well-known error criterion is the Zienkiewicz–Zhu criterion (1987, 1992a, 1992b) where the measure of error in a linear problem is:

$$\theta^2 = \int_{\Omega_e} (\sigma_{ij} - \sigma_{ij}^h)(\varepsilon_{ij} - \varepsilon_{ij}^h)\, d\Omega. \tag{13}$$

It is then assumed that the error can be estimated by replacing σ_{ij}, ε_{ij} by an approximated exact

solution, so

$$\theta_{\text{app}}^2 \approx \int_{\Omega_e} (\tilde{\sigma}_{ij} - \sigma_{ij}^h)(\tilde{\varepsilon}_{ij} - \varepsilon_{ij}^h)\, d\Omega \qquad (14)$$

where $\tilde{\sigma}_{ij}$, $\tilde{\varepsilon}_{ij}$ is the projected or approximated exact solution obtained by a least square fit to σ_{ij}^h or ε_{ij}^h, respectively. A similar criterion can be found in Torigaki (1989), where the nodal averaging method is used to obtain the approximated exact solution. This criterion performs very well for linear continuum problems. However, the computation of the projection is quite complicated in shell problems, because the stress and strain components for each element are computed in local coordinates. These local components need to be transformed to global components to be assembled into the global stress and strain matrices in both the minimization and the nodal averaging methods. This entails many transformations, and the transformations are ambiguous in a finite element model because a consistent set of global components cannot be defined unless a continuous surface is specified for the shell. Therefore, because the local coordinate systems are often related to the element node numbering, simply interpolating the local components of tensors is not satisfactory.

In this work, norms of strain invariants, as in Belytschko and Yeh (1993), with MLS projection are used for evaluating an error estimate. The reason for choosing the invariants of strain as an error indicator is that they characterize the elongations of a solid and they are scalars with sign.

The measure of strains in nonlinear structural dynamics codes such as DYNA3D is usually the rate of deformation. However, this is a rate measure and its integral is not path independent. A rate measure of deformation is not satisfactory as an error measure in an explicit code because it tends to be rather oscillatory. Therefore, the invariants of the Green–Lagrange strain are used as an error measure (the invariants are chosen for reasons already cited).

Shell elements are in a state of plane stress (we treat the transverse shears as penalties to enforce the Kirchhoff–Love hypothesis), so the state of strain in the local system has three independent components and three invariants. They are given by the standard equation

$$\mathbf{E} = \frac{1}{2}(\mathbf{F}^T\mathbf{F} - \mathbf{I}) \qquad (15)$$

and $\mathbf{F}$ is the deformation gradient

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}. \qquad (16)$$

The invariants of $\mathbf{E}$ are

$$I_1 = \text{tr } \mathbf{E} \qquad (17)$$

$$I_2 = \frac{1}{2}(\mathbf{E}:\mathbf{E} - (\text{tr } \mathbf{E})^2) \qquad (18)$$

$$I_3 = \det \mathbf{E}. \qquad (19)$$

The smoothed interpolants I_i^* are obtained by the MLSM described earlier.

The error criterion for element e is then given by

$$\theta_e = \sum_{i=1}^{3} \|I_i^*(\mathbf{x}) - I_i^h(\mathbf{x})\|_{\Omega_e} \qquad (20)$$

where I_i^h is the invariant computed from the finite element solution.

ADAPTIVE ALGORITHM

In transient analysis, the structure of the adaptive strategy is crucial in obtaining accuracy and reasonable running time. In the algorithm used in Belytschko et al. (1989) and Sarwas (1989), at judgment time t_n the mesh was reconfigured in accordance with an error check if any elements exceeded the tolerance; this mesh was used for the subsequent time segment (called adaptation interval). With this algorithm, we have found that for reasonable values of the adaptation interval, fission (refinement) occurs too late and the lost accuracy can never be recovered. If the adaptation interval is too small, considerable noise can be generated by successive fission and fusion, and the cost is substantial. Bajer (1988) drew similar conclusions.

Figure 1 shows the go-back adaptive algorithm used by Belytschko et al. (1992) and Belytschko and Yeh (1993). In this procedure, the time interval during the adaptation interval prior to any adaptive assessment is repeated with the newly generated mesh to recover the lost accuracy. In Devloo, Oden, and Strouboulis (1987), a time interval is repeated until error tolerance is met. This scheme is even more time consuming, but it does insure that a higher level of accuracy will be

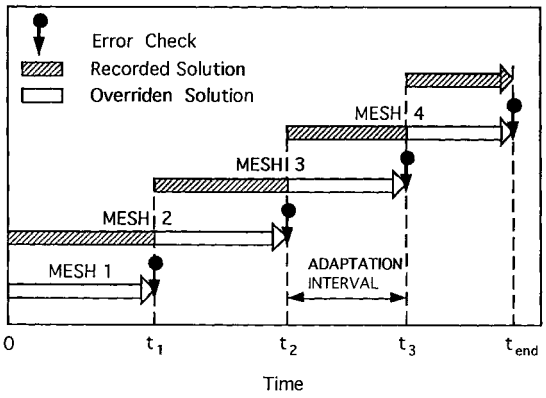


FIGURE 1 Go-back adaptive procedure by Belytschko et al. (1992) and Belytschko and Yeh (1993).

maintained. These two methods have the following difficulties:

1. programming is complicated because the data structure changes dramatically during the calculation;
2. the writing of output tapes and other aspects of postprocessing are difficult because the data structure changes and while running any time segment.

In the algorithm used here (shown in Figure 2) from Belytschko and Yeh (1992), some of these difficulties are avoided. The procedure is structured as follows. First, the problem is completely solved with a coarse mesh. During the coarse mesh run, assessments are made at selected intervals, from which the level of adaptation for each adaptation interval is selected. On the basis of these assessments, a second complete run is then made with several meshes. A small adaptation interval (such as 10 or 20 time steps) is used to make sure that critical phenomena such as bifurcation can be captured and that the dependence of results on the adaptive periods is circumvented. This new strategy simplifies data structures and postprocessing because:

1. geometric and material information do not need to be stored temporarily for the use of the go-back;
2. the element arrangements for the adaptive runs are known prior to the run so that data files for time histories are easier to write.

Of course, the total CPU time spent in the initial coarse mesh run and the error checks may appear burdensome. Because we make error checks only in the coarse mesh run, the CPU time used for the second run is less than with procedures such as in Figure 1. Further, compared to a uniformly refined mesh, this two run approach is quite cost efficient. The CPU time for a mesh with uniform refinement of level NL is 8^{NL} times the initial mesh. This figure reflects the fact that one level of refinement in a shell mesh quadruples the number of elements and entails a twofold reduction in the time step. The CPU time spent in the test run is therefore only a small part of the total, and if the adaptive mesh can save a significant fraction of the time required for a uniform refinement, then the potential CPU savings of this method are significant. Our results show that this strategy saves about 50% of CPU time compared to a computation with a uniformly refined mesh.

The method makes it necessary to relate the error in the coarse mesh to the number of subdivisions needed to obtain the requisite accuracy. Because a constant resource approach has been taken here, this is not of concern. The error in the second run can be tracked, and if it is unacceptable, the run can be repeated. We found that in most runs, the behavior of the two meshes is not radically different, and the second run serves primarily to obtain more accuracy. Of course, if the behavior of the two runs is radically different, then careful monitoring of error in the second run is essential.

Adaptivity

H-adaptivity consists of two processes: fission (or refinement) and fusion (or unrefinement).

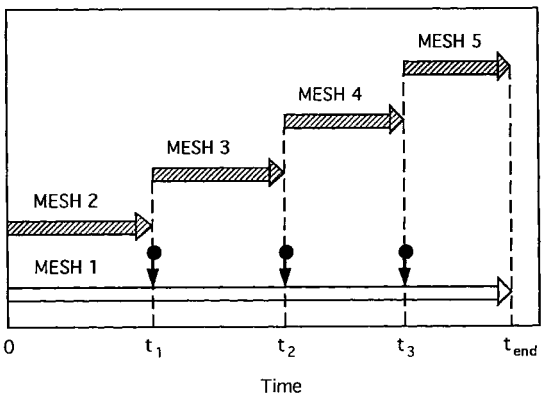


FIGURE 2 New adaptive procedure using initial coarse mesh run (Belytschko and Yeh, 1992).

These procedures have been extensively described in Belytschko et al. (1989, 1993) and Belytschko and Yeh (1992). In this article only the forward and reverse projections needed in fission and fusion, respectively, are described. In our previous work we used very simple projections, and we have found improvements in the performance of h-adaptivity by using better projections based on MLS approximations.

The following variables are subject to forward projection:

1. the coordinates $\mathbf{x}_I$ of the new nodes;
2. the velocities of the new nodes $\mathbf{v}_I$;
3. the Cauchy stresses $\sigma(\mathbf{x}_I^Q)$ and state variables, such as yield strength, etc. at the quadrature points $\mathbf{x}_I^Q$.

For the purpose of making the projection, it is convenient to construct a reference plane for each element. This plane is constructed by a weighted least square fit to the nodes within the domain of influence of the weight functions.

These projections must account for discontinuities in the shell structures: smooth projections should not be continued across discontinuities in the surfaces. For example, in a box beam, projections should not extend across the edges. To implement this, nodes on discontinuities in surfaces must be labeled in the initial mesh. These labeled nodes are used to construct lines of discontinuity, Γ_ξ^D which are then projected onto the reference plane. In computing a projection at any point $\mathbf{x}_I$, data at a point $\mathbf{x}_Q$ is excluded from the domain of influence if the line from ξ_I to ξ_Q intersect Γ_ξ^D.

Prediction of Mesh Refinement

The algorithm for determining levels of refinement used in Belytschko and Yeh (1993) is summarized in Table 1. Any element in which the error (ERR) exceeds the error criterion (CRIT) is fissioned one more level. Belytschko et al. (1989) use a similar algorithm, except they fission and fusion a fixed percentage of elements so the number of elements remains constant. These algorithms have two shortcomings.

1. CRIT is a factor that crucially affects the quality of the results (the other is the adaptation interval). A small value of CRIT will result in the fission of too many elements and lead to the waste of CPU time; too large a value of CRIT does not yield any improvement. An appropriate value of CRIT can only be obtained by experience from many test runs and is quite problem dependent.
2. The initial refinement level given by the old algorithm is at most one level. As we described before, lost accuracy cannot be recovered no matter how you fission your mesh, so if you start with a mesh in which all elements are fissioned at too low a level, the results will never reach the desired accuracy.

Table 1. Refinement Algorithm in Belytschko and Yeh (1992)

```
for each adaptation assessment IT = 1 to NT:
  for each element JE = 1 to NELE:
    if ERR (JE) ≥ CRIT then
      LEVEL(JE, IT + 1) = LEVEL(JE, IT) +1
    endif
  enddo for JE
enddo for IT
NT              : the number of adaptation assessments.
CRIT            : the error criterion input by the user.
LEVEL(JE, IT) : level of refinement for element JE in the adaptation assessment IT.
```

Devloo et al. (1987) circumvented the second difficulty by repeating the adaptive time segments until the error is reduced to the desired level at the assessment; however, this can lead to long and unpredictable running times that may be undesirable in practice.

Here we describe a more easily used procedure for setting the error criterion and the refinement. To avoid more than one iteration over a given time interval, we have used the procedure shown in Table 2 to estimate the errors for several levels of refinement based on the coarse mesh solution. In this procedure, two parameters are required: the maximum level of refinement (MAXLEV) and the target deviation e ($0 \leq e \leq 1$); e is defined as the ratio of the error difference between the coarsest mesh and the adaptive mesh and the error difference between the coarsest mesh and the MAXLEV level finest mesh:

$$e = \frac{(\text{error of adaptive mesh} - \text{error of finest mesh})}{(\text{error of coarsest mesh} - \text{error of finest mesh})}. \quad (21)$$

Table 2. Refinement Level Selection Algorithm

1. Use the extrapolation method to net nodal invariant values.
2. Approximate exact solution (AES) is constructed over the finite element mesh by assigning the interpolated values at the centers of the elements in the MAXLEV+1 level mesh, which is the finest mesh.
3. Follow the same procedures of step (2) to obtain invariant values at the centers for all levels IL, $0 \le \text{IL} \le \text{MAXLEV}$. These will be called the lower level solution.
4. After getting the AES from step (2) and the lower level solution from step (3), we use the strain invariant norms of each element for each level IL, which is called CRIT(JE, IL).
5. Take the sum of CRIT(JE, IL) as the norm of the whole object for level IL = 0 and MAXLEV, i.e.,

$$\text{WCRIT(IL)} = \sum_{\text{JE}=1}^{\text{NELE}} \text{CRIT(JE, IL)}$$

6. Based on the given target deviation, TAGDEV, the required error can be estimated by

$$\text{RCRIT} = \text{WCRIT(MAXLEV)} + \text{TAGDEV} \times (\text{WCRIT(0)} - \text{WCRIT(MAXLEV)})$$

7. Calculate the average of the required invariant error by

$$\text{ARI} = \text{RCRIT/NELE}$$

which will then be used as our fission criterion;

8. Use the following procedure to obtain the fission level of element JE, LEVEL(JE).

```
a. for each element 1 ≤ JE ≤ NELE:
b.    for level 0 ≤ IL ≤ MAXLEV:
c.       If CRIT(JE, IL) ≤ ARI then
d.          LEVEL(JE) = IL
e.             GOTO step i
f.       endif
g.    enddo for IL
h.    LEVEL(JE) = MAXLEV
 i. enddo for JE
```

when $e = 1$, no elements will be refined;
when $e = 0$, all elements will be refined to the level MAXLEV.

NUMERICAL EXAMPLES

The examples were run on an HP-720 system. All use the one-point integration element developed by Belytschko and Leviathan (1994), which is an extension of the element (Belytschko and Tsay, 1983). It uses physical hourglass control. All examples use elastic–plastic materials with the Mises yield criterion and five integration points are used through the thickness of the shell element; the material parameters of the example problems are given in Table 3. Several levels of adaptation are employed. For contact–impact problems the pinball contact–impact algorithm of Belytschko and Yeh (1993) was used. These problems were previously reported in Belytschko and Yeh (1993). They are rerun here using our new adaptive algorithm and the split-pinball scheme.

The examples show that the stresses, strains, and displacements obtained by this adaptive mesh are very close to those obtained by the finest mesh, and a great deal of CPU time is saved. The normalized CPU time (CPU time/CPU time of coarsest mesh) for different meshes are shown in Table 4, which shows that our adaptive algorithm can save 20–30% of the CPU time with one level of fission, 35–50% with two levels of fission. For a given problem, a higher level of fission results in greater savings of CPU time.

The first problem, shown in Figure 3, is an elastic, perfectly plastic, 120° cylindrical panel that is loaded impulsively with an initial radial velocity over a portion of the top surface. On the loaded area of 3.08×10.205 in., an initial velocity of 5650 in./s is applied to the panel. Two sides of this panel are clamped and two ends are hinged. Half of the panel is modeled due to symmetry. The initial mesh used in the coarse

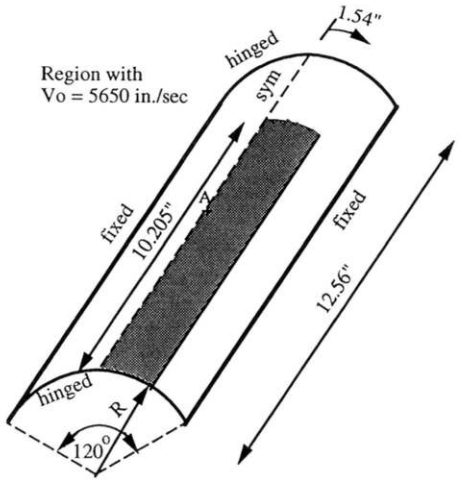


FIGURE 3 Cylindrical panel problem.

Table 3. Material Properties of Test Problems

Material Properties	Cylindrical Panel	S-Beam	Box Beam	Impacting Tubes
Elastic modulus (E)	1.05×10^7 psi	3.00×10^7 psi	2.06×10^{11} N/m^2	0.25×10^{11} N/m^2
Plastic modulus (E_p)	0.00 psi	0.00 psi	6.30×10^8 N/m^2	2.30×10^8 N/m^2
Yield stress (σ_y)	4.40×10^4 psi	3.60×10^4 psi	2.00×10^8 N/m^2	1.00×10^8 N/m^2
Density (ρ)	2.50×10^{-4} lb-s^2/in.^{4}	7.40×10^{-4} lb-s^2/in.^{4}	7.84×10^3 kg/m^2	7.64×10^3 kg/m^2
Poisson's ratio (ν)	0.33	0.30	0.30	0.30

mesh run consists of 300 (12×25) elements. The time history of displacements at the center point A is given in Figure 4 for the coarse mesh, the fine mesh (1200 elements), and the one and two level adaptive meshes. Figure 5 shows the evolution of the deformed geometry at selected times for two-levels adaptive solution.

The second problem shown in Figure 6 is a S-shaped T-section beam, fixed at one end and with axial loading applied to the other end. Axial displacement and velocity of node A for the two-level adaptive mesh are shown in Figures 7 and 8, respectively. The deformed configurations and the development of element refinement by h-adaptivity are shown in Figure 9.

The third example is a thin-walled box beam that impacts a fixed rigid wall. Figure 10 gives the geometry for this problem. Only one-quarter of this beam is analyzed due to the symmetry. It was previously analyzed by Benson and Hallquist (1987), Sarwas (1989), and Belytschko and Neal (1991). The solution reported by Belytschko and Neal (1991) used imperfections of the same sign on the sides and top and bottom; in Belytschko and Yeh (1993), the imperfection on the sides and bottom are of different sign but of the same magnitude. Although roundoff errors

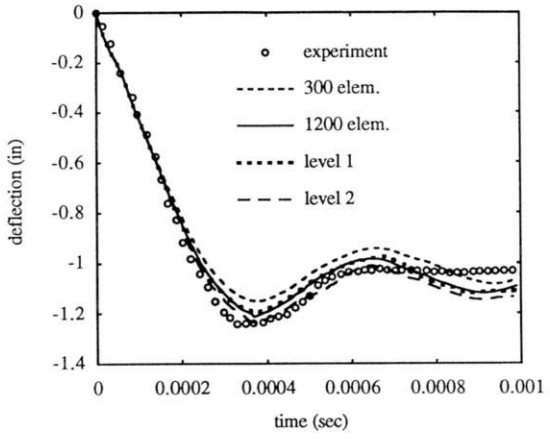


FIGURE 4 Deflections at the center of cylindrical panel.

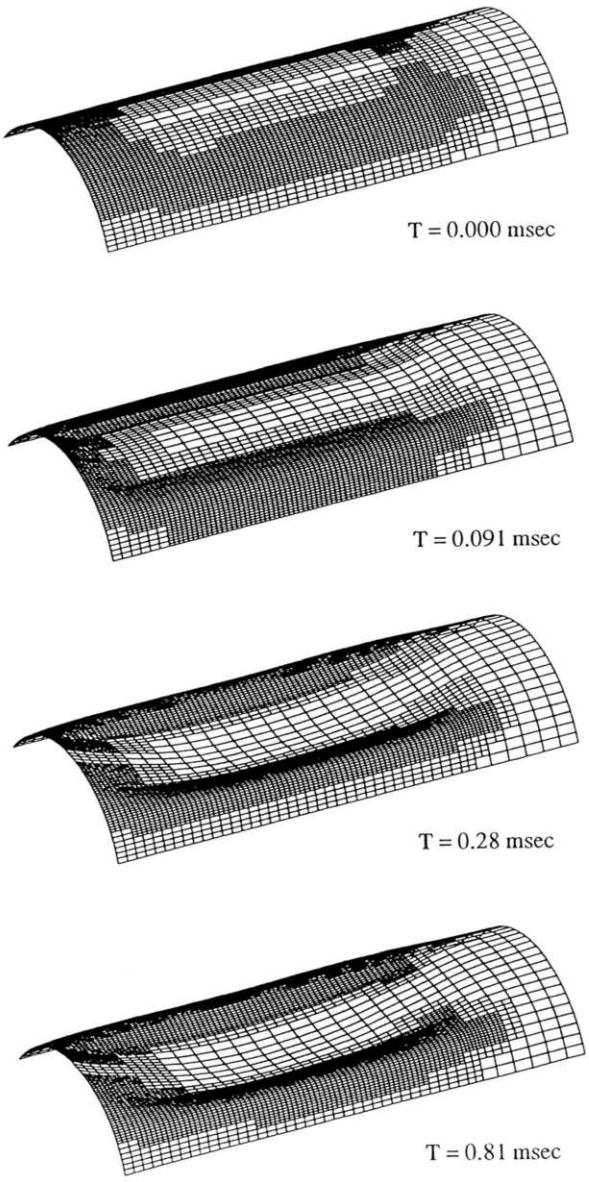


FIGURE 5 Deformed shapes of cylindrical panel.

Table 4. Normalized CPU Time for Sample Problems

	Cylindrical Panel		S-Beam		Box Beam		Impacting Tubes	
MESH	Elem. #	CPU	Elem. #	CPU	Elem. #	CPU	Elem. #	CPU
L1finest	1200	8	960	7	756	13	1536	15.3
AdapL1	720	5.5	510	4.2	456	9	993	9.2
L2finest	4800	64	3840	51	NA	NA	NA	NA
AdapL2	2560	31.6	1970	24.9	NA	NA	NA	NA

L1, one-level; L2, two-level. NA, not available.

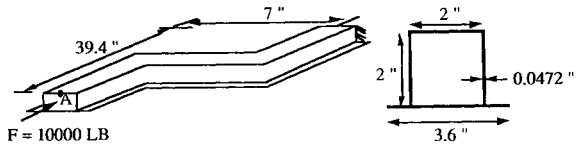


FIGURE 6 Geometry of S-beam.

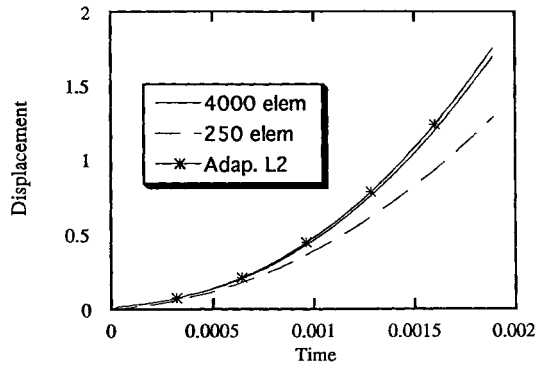


FIGURE 7 Axial displacement of point A in S-beam.

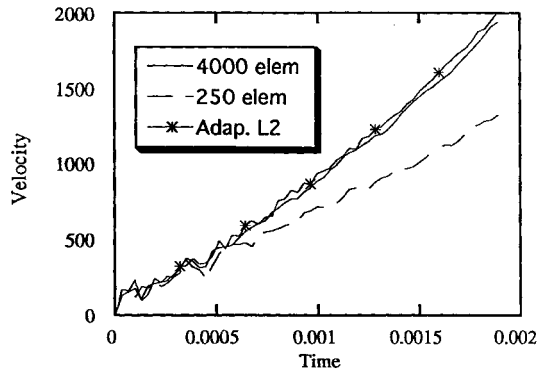


FIGURE 8 Axial velocity of point A in S-beam.

will sometimes trigger folding type instabilities even in very symmetric structures such as this box beam, we found that the buckling modes can vary widely unless an imperfection is used to seed the buckling mode. Incidentally, in experiments on this box beam, it has been reported that the buckling modes are also not easily reproducible. The initial mesh for the coarse mesh run consists of 189 elements. The comparison of velocity at point A (where $x = 0$) is shown in Figure 11. Figure 12 shows the one-level adaptive meshes.

The last example is two impacting tubes shown in Figure 13. The initial coarse mesh has 300 (10 × 15 × 2) elements. The comparison of velocity at point A is shown in Figure 14. Figure 15 presents the adaptive mesh.

DISCUSSION AND CONCLUSION

A simplified *h*-adaptive strategy was studied for nonlinear structural dynamics with explicit integration. The algorithm was quite successful in a series of problems involving large displacement, elastic–plastic response with contact–impact. The procedure provides marked savings in compute time over solutions run with uniform meshes. Although the problems treated are not of the complexity encountered in general engineering analysis, they do contain some of the complicating features of these problems. Therefore, the success of these solutions is quite encouraging. *H*-adaptive methods provide the following advantages:

1. they eliminate the need to mesh a problem on the basis of what is expected in various parts of the mesh;

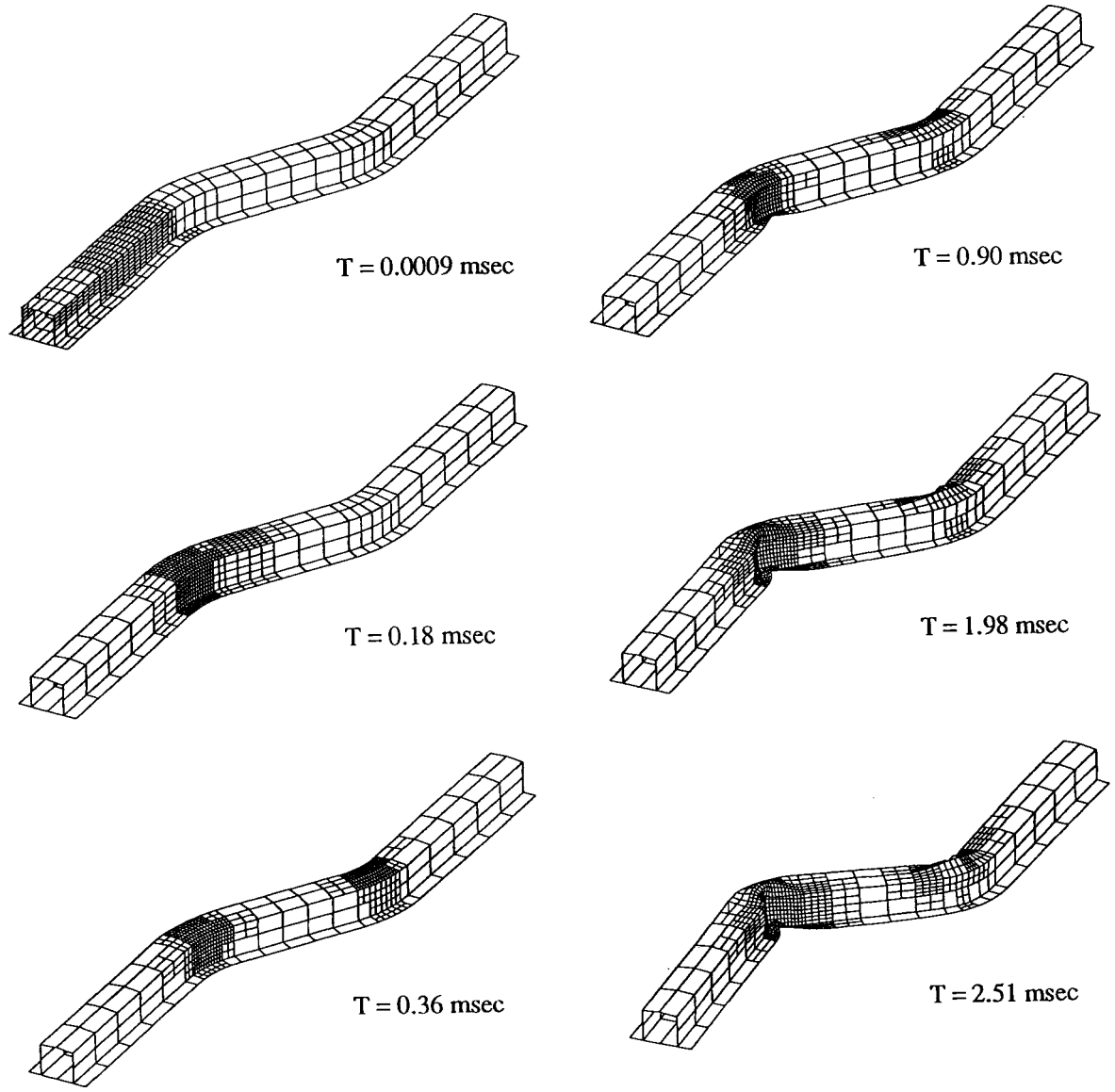
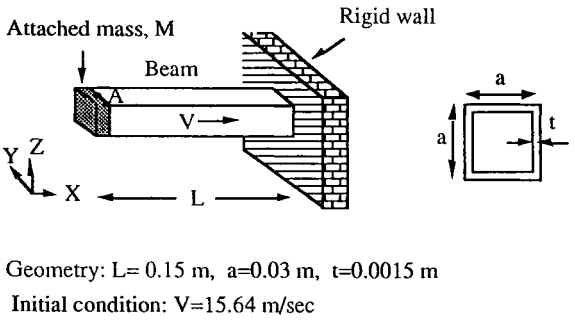


FIGURE 9 Two-level adaptive mesh of S-beam.



Geometry: L= 0.15 m, a=0.03 m, t=0.0015 m
Initial condition: V=15.64 m/sec

FIGURE 10 Box beam problem.

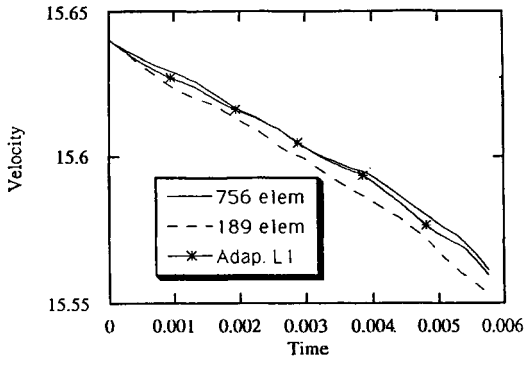


FIGURE 11 Velocity of point A ($x = 0$) in box beam.

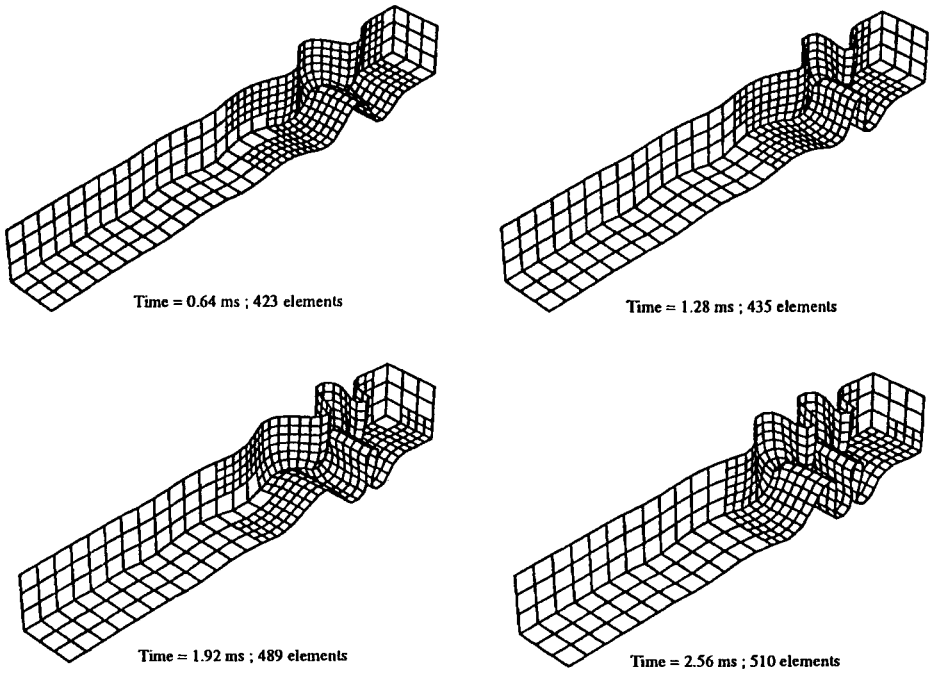
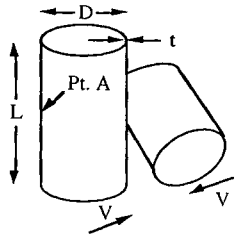


FIGURE 12 One-level adaptive mesh of box beam.



Geometry :
L = 0.460 m
D = 0.200 m
t = 0.006 m

Initial condition :
V = 35 m/s
Gap = 0.05 m

FIGURE 13 Two impacting tubes problem.

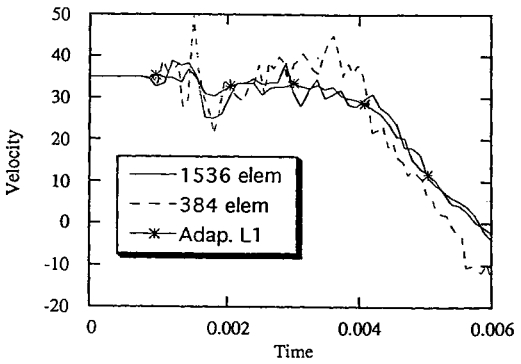


FIGURE 14 Velocity of point A on impacting tubes.

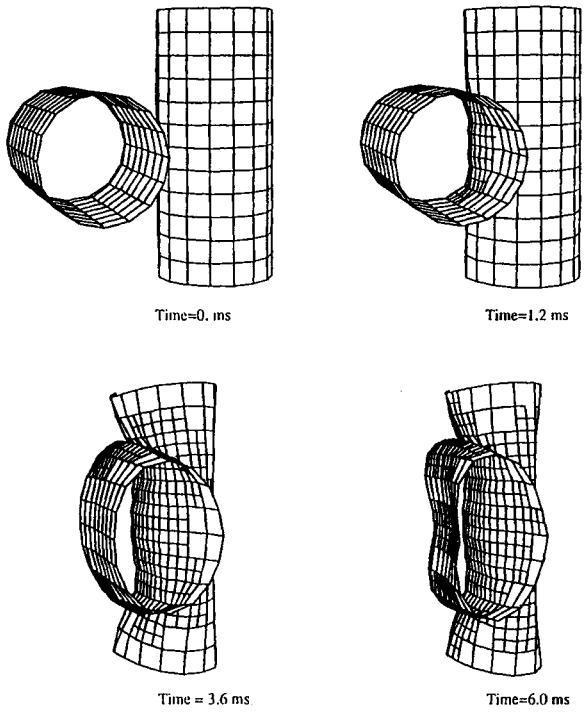


FIGURE 15 One-level adaptive mesh of impacting tubes.

2. by refining the mesh where needed, and only where it is needed, they provide more accuracy in nonlinear simulations at less cost.

Error criteria and adaptive strategies have been developed for nonlinear dynamics. The error criterion circumvents some of the difficulties associated with the treatment of shells by expressing errors in terms of strain invariants. As a consequence there is no need to deal with different coordinate system transformations.

The support of the Office of Naval Research is gratefully acknowledged.

REFERENCES

Bajer, C., 1988, "Dynamics of Contact Problem by Adaptive Simple-Shaped Space-Time Approximation," *Journal of Theoretical and Applied Mechanics*, Special Issue 7, pp. 235–248.

Belytschko, T., Lee, S.-H., Yeh, I. S., Lin, J. I., Tsay, C. S., and Kennedy, J. M., 1993, "Adaptivity in Crashworthiness Calculations," *Journal of Shock and Vibration* Vol. 1(2), pp. 97–106.

Belytschko, T., and Leviathan, I., 1994, "Physical Stabilization of the 4-Node Shell Element with One-Point Quadrature," *Computer Methods in Applied Mechanics and Engineering*, Vol. 113, pp. 321–350.

Belytschko, T., and Neal, M. O., 1991, "Compact–Impact by the Pinball Algorithm with Penalty and Lagrangian Methods," *International Journal for Numerical Methods in Engineering*, Vol. 31, pp. 547–572.

Belytschko, T., and Tsay, C. S., 1983, "A Stabilization Procedure for the Quadrilateral Plate Element with One-Point Quadrature," *International Journal for Numerical Methods in Engineering*, Vol. 19, pp. 405–419.

Belytschko, T., Wong, B. L., and Plaskacz, E. J., 1989, "Fission–Fusion Adaptivity in Finite Elements for Nonlinear Dynamics of Shells," *Computers and Structures*, Vol. 33, pp. 1307–1323.

Belytschko, T., and Yeh, I. S., "Adaptivity in Nonlinear Structural Dynamics with Contact–Impact," in A. K. Noor, *Adaptive, Multilevel, and Hierarchical Computational Strategies*, 1992, AMD-157, ASME, New York.

Belytschko, T., and Yeh, I. S., 1993, "The Splitting Pinball Method for Contact–Impact Problems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 105, pp. 375–393.

Belytschko, T., Yeh, I. S., Lee, S.-H., Lin, J. I., Tsay, C. S., and Kennedy, J. M., 1992, "Adaptivity and Pinball Contact–Impact for Automobile Crashworthiness," *Automotive Applications of Vector/Parallel Computers: State of the Art*, SP-923, pp. 29–40.

Benson, D. J., and Hallquist, J. O., 1987, "A Single Surface Contact Algorithm for the Postbuckling Analysis of Shell Structures," Report to the University of California at San Diego, CA.

Devloo, P., Oden, J. T., and Strouboulis, T., 1987, "Implementation of an Adaptive Refinement Technique for the SUPG Algorithm," *Computer Methods in Applied Mechanics and Engineering*, Vol. 61, pp. 339–358.

Hallquist, J. O., 1983, "Theoretical Manual for DYNA3D," Report UCID-19401, University of California, Lawrence Livermore National Laboratory, Livermore, CA.

Sarwas, R. E., 1989, "Hidden Line Elimination and a Modified Pinball Algorithm for Finite Element Contact–Impact Problems with Shell Elements," MS Thesis, Northwestern University, Evanston, IL.

Torigaki, T., 1989, "Contact and Impact Problems Using Adaptive Finite Element Methods," PhD Thesis, University of Michigan, Ann Arbor, MI.

Zienkiewicz, O. C., and Zhu, J. Z., 1987, "A Simple Erorr Estimator and Adaptive Procedure for Practical Engineering Analysis," *International Journal for Numerical Methods in Engineering*, Vol. 24, pp. 337–357.

Zienkiewicz, O. C., and Zhu, J. Z., 1992a, "The Superconvergent Patch Recovery and a Posteriori Error Estimates. Part 1: The Recovery Technique," *International Journal for Numerical Methods in Engineering*, Vol. 33, pp. 1331–1364.

Zienkiewicz, O. C., and Zhu, J. Z., 1992b, "The Superconvergent Patch Recovery and a Posteriori Error Estimates. Part 2: Error Estimates and Adaptivity," *International Journal for Numerical Methods in Engineering*, Vol. 33, pp. 1365–1382.