



**Defense Nuclear Agency
Alexandria, VA 22310-3398**



DNA-TR-95-92

**Adaptive Refinement for Explicit
Nonlinear Finite Element Analysis**

**Ted Belytschko
Belytschko (Theodore B.)
18 Longmeadow Road
Winnetka, IL 60093**

June 1996

Technical Report

CONTRACT No. DNA 001-92-C-0072

DTIC QUALITY INSPECTED 3

Approved for public release;
distribution is unlimited.

19960611 176

Destroy this report when it is no longer needed. Do not return to sender.

**PLEASE NOTIFY THE DEFENSE NUCLEAR AGENCY,
ATTN: CSTI, 6801 TELEGRAPH ROAD, ALEXANDRIA, VA
22310-3398, IF YOUR ADDRESS IS INCORRECT, IF YOU
WISH IT DELETED FROM THE DISTRIBUTION LIST, OR
IF THE ADDRESSEE IS NO LONGER EMPLOYED BY YOUR
ORGANIZATION.**



DISTRIBUTION LIST UPDATE

This mailer is provided to enable DNA to maintain current distribution lists for reports. (We would appreciate your providing the requested information.)

- Add the individual listed to your distribution list.
- Delete the cited organization/individual.
- Change of address.

NOTE:

Please return the mailing label from the document so that any additions, changes, corrections or deletions can be made easily. For distribution cancellation or more information call DNA/IMAS (703) 325-1036.

NAME: _____

ORGANIZATION: _____

OLD ADDRESS

CURRENT ADDRESS

TELEPHONE NUMBER: () _____

DNA PUBLICATION NUMBER/TITLE

CHANGES/DELETIONS/ADDITIONS, etc.)

(Attach Sheet if more Space is Required)

DNA OR OTHER GOVERNMENT CONTRACT NUMBER: _____

CERTIFICATION OF NEED-TO-KNOW BY GOVERNMENT SPONSOR (if other than DNA):

SPONSORING ORGANIZATION: _____

CONTRACTING OFFICER OR REPRESENTATIVE: _____

SIGNATURE: _____

CUT HERE AND RETURN



DEFENSE NUCLEAR AGENCY
ATTN: IMAS
6801 TELEGRAPH ROAD
ALEXANDRIA, VA 22310-3398

DEFENSE NUCLEAR AGENCY
ATTN: IMAS
6801 TELEGRAPH ROAD
ALEXANDRIA, VA 22310-3398

REPORT DOCUMENTATION PAGE

Form Approved

OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE 960601	3. REPORT TYPE AND DATES COVERED Technical 920610 - 950210	
4. TITLE AND SUBTITLE Adaptive Refinement for Explicit Nonlinear Finite Element Analysis			5. FUNDING NUMBERS C - DNA 001-92-C-0072 PE - 62715H PR - AC TA - AK WU - DH323940	
6. AUTHOR(S) Ted Belytschko				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Belytschko (Theodore B.) 18 Longmeadow Road Winnetka, IL 60093			8. PERFORMING ORGANIZATION REPORT NUMBER 95-1	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Nuclear Agency 6801 Telegraph Road Alexandria, VA 22310-3398 WELR/Giltrud			10. SPONSORING/MONITORING AGENCY REPORT NUMBER DNA-TR-95-92	
11. SUPPLEMENTARY NOTES This work was sponsored by the Defense Nuclear Agency under RDT&E RMC Code B4662D AC AK 00003 4300A 25904D.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) An implementation of h-adaptivity was developed for DYNA2D, a nonlinear finite element program for two-dimensional solids. In h-adaptivity, elements are subdivided into smaller elements when an error or refinement indicator exceeds a user-specified threshold. Several error and refinement indicators are included, among them a local projection of the Green-Lagrange strain tensor, the effective stress, and the effective plastic strain. The h-adaptive part of the program is totally separate from the DYNA2D program. Adaptivity is achieved through the use of the restart and rezone options in DYNA2D. The following problems were solved with the h-adaptive program: the Taylor rod impact problem, an elastic beam and elastic-plastic beam. Comparisons were made with fine mesh solutions and show good agreement with substantial savings in computer time. Further improvements in the computational efficiency of the h-adaptive scheme could be achieved by adding subcycling in DYNA2D.				
14. SUBJECT TERMS Finite Elements Structural Dynamics			15. NUMBER OF PAGES 40	
Adaptivity Nonlinear Analysis			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT SAR	

NSN 7540-280-5500

Standard Form 298 (rev. 2-89)
Prescribed by ANSI Sta. Z39-18
298-102

CLASSIFIED BY:

N/A since Unclassified.

DECLASSIFY ON:

N/S since Unclassified.

CONVERSION TABLE

Conversion factors for U.S. Customary to metric (SI) units of measurement.

MULTIPLY \longrightarrow BY \longrightarrow TO GET
 TO GET \longleftarrow BY \longleftarrow DIVIDE

foot	3.048 000 X E -1	meter (m)
foot-pound-force	1.355 818	joule (J)
inch	2.540 000 X E -2	meter (m)
kip/inch ² (ksi)	6.894 757 X E +3	kilo pascal (kPa)
ounce	2.834 952 X E -2	kilogram (kg)
pound-force (lbs avoirdupois)	4.448 222	newton (N)
pound-force inch	1.129 848 X E -1	newton-meter (N·m)
pound-force/inch ² (psi)	6.894 757 4.535 924 X E -1	kilo pascal (kPa)
pound-mass (lbm avoirdupois)	4.535 924 X E -1 1.601 846 X E +1	kilogram (kg)
pound-mass/foot ³	1.601 846 X E +1	kilogram/meter ³ (kg/m ³)
slug	1.459 390 X E +1	kilogram (kg)
torr (mm Hg, 0° C)	1.333 22 X E -1	kilo pascal (kPa)

TABLE OF CONTENTS

Section	Page
CONVERSION TABLES	iii
FIGURES.....	v
1 INTRODUCTION	1
2 METHODOLOGY	3
2.1 OUTLINE AND NOTATION.....	3
2.2 COMPUTATIONAL PROCEDURE IN DYNA2D	3
2.3 ADAPTIVITY	5
2.4 ERROR AND REFINEMENT INDICATORS.....	9
2.5 IMPLEMENTATION.....	12
3 EXAMPLES OF H-ADAPTIVE CALCULATIONS	17
3.1 TAYLOR IMPACT PROBLEM.....	17
3.2 BEAM PROBLEMS	24
4 CONCLUSIONS	28
5 REFERENCES	31

FIGURES

Figure	Page
2-1 Example of h -adaptivity for a quadrilateral element; the left element has been refined on one level	8
2-2 Flow of computations in adaptivity with a go-back procedure	13
3-1 Deformed meshes for the Taylor impact problem for h -adaptive solution at $5\mu\text{s}$ intervals from $5\mu\text{s}$ to $40\mu\text{s}$	18
3-2 Deformed meshes for Taylor impact problem for h -adaptive solution $5\mu\text{s}$ intervals from $45\mu\text{s}$ to $80\mu\text{s}$	19
3-3 Deformed meshes for the fine mesh simulation at $10\mu\text{s}$ intervals from $10\mu\text{s}$ to $80\mu\text{s}$	20
3-4 Axial displacements of free-end point and midpoint on the axis of symmetry for Taylor rod problem; time is in μs	21
3-5 Axial velocities of free-end point and midpoint on the axis of symmetry for Taylor rod problem; time is in μs	22
3-6 The axial stress σ_{zz} at the midpoints of the axis of the rod; time is in μs	23
3-7 Description of beam problem	25
3-8 Displacement of free end of elastic beam as a function of time	25
3-9 Displacement of free end of elastic-plastic beam as a function of time	26
3-10 Deformed meshes for the elastic-plastic beam at four stages of deformation	27

SECTION 1 INTRODUCTION

The calculation of the large deformation response of structures by finite element methods is often a difficult task because the user has little guidance as to the size of the elements that need to be used. This is a particularly severe handicap in calculating the failure of structures because failure often results in localization of deformation.

Localization requires considerable refinement of the mesh, in the area of localization if the failure mode is to be calculated accurately. Therefore, the user must anticipate where localization is likely to occur in designing the mesh or make several runs to achieve an accurate solution. Adaptivity provides a way for reducing this burden by providing an automatic selection of appropriate element refinement. In adaptive programs, elements are subdivided in areas where refinement is needed so that the required accuracy is maintained. This refinement is guided by error indicators, which indicate an approximate level of the errors which are engendered by the finite element approximation; refinement criteria which are not based on error criteria can also be used.

This report describes the implementation of adaptivity in the computer program DYNA2D, (Whirley, et al 1992). The adaptivity is performed by subdividing elements into smaller elements according to error or refinement criteria, which is called *h*-adaptivity. The name *h*-adaptivity stems from the association of *h* with element size in the finite element literature. DYNA2D is a program originated by John Hallquist at Lawrence Livermore Laboratory. This program is a general purpose program for the nonlinear dynamics of solids and structures. It is based on explicit time integration, so it does not store or compute a stiffness matrix. *H*-adaptivity enables the user of the program to automatically refine portions of the mesh according to error or refinement criteria so that accuracy is increased for a given amount of computational resources. This is of particular importance in nonlinear failure problems because the most crucial areas for the placement of nodes and refinement of the mesh are in the failure areas. In *h*-adaptivity, the program will automatically refine the mesh in areas of the mesh in which the estimated error is large, and thus improve accuracy.

The program DYNA2D was chosen for this effort because it is widely used in the defense community and has a large range of capabilities for nonlinear analysis of weapons effects and related problems. Among the attractive features and capabilities of

DYNA2D are a large material library, an effective sliding interface for contact-impact, and the availability of capabilities such as rezoning and remeshing, postprocessing and interactive display, and the availability of a compatible preprocessor MAZE, (Hallquist 1983). Furthermore, the development of these procedures in DYNA2D has provided insight into adding adaptivity to other programs such as DYNA2D.

During the course of this work, it was found that it is very difficult to modify a program such as DYNA2D to internally implement h -adaptivity. The implementation of adaptivity requires so many changes throughout the program that our initial efforts in this direction were totally unsuccessful and could not even invoke a very limited subset of DYNA2D functionality. After further consideration of this approach, it also became clear that internal modifications would preclude its use by many other users, since many versions of DYNA2D are under separate development in the defense community.

Therefore, we selected an approach in which the programming for the h -adaptivity is completely external to DYNA2D and an h -adaptive run is executed through a series of DYNA2D restarts and rezones. This enabled us to keep all coding for the h -adaptivity completely separate from DYNA2D, thus making the h -adaptive procedures usable in later versions of DYNA2D and versions of DYNA2D which have been modified by other agencies.

This report is organized as follows. Section 2 describes the general framework of DYNA2D, of h -adaptivity and the procedures involved in its implementations and the error criteria which have been implemented in the program. Section 3 gives some examples of h -adaptive solutions, which is followed by conclusions in Section 4.

SECTION 2 METHODOLOGY

2.1 OUTLINE AND NOTATION.

This Section briefly describes the methodology of DYNA2D and then gives a description of h -adaptivity, the reasons for selection of h -adaptivity and the refinement and error criteria implemented in this work.

Vectors, tensors and matrices in this report are indicated by boldface. Indicinal notation is also used. In indicinal notation, the elements of a vector tensor or matrix are indicated by subscripts; for example, in indicinal notation the elements of vector \mathbf{x} are indicated by x_i . Upper case subscripts are used for nodal numbers, so x_{iI} are the coordinates at node I .

2.2 COMPUTATIONAL PROCEDURE IN DYNA2D.

DYNA2D is a finite element program for calculating the nonlinear response of solids in two dimensions; in addition, axisymmetric three dimensional structures can be treated. The program uses explicit time integration. In the nonlinear continuum mechanics formulation in DYNA2D, the deformation of the solid is described by

$$\mathbf{x} = \mathbf{x}(\mathbf{X}, t) \quad (2.1)$$

where $\mathbf{x}=[x,y]$ are the current spatial coordinates and $\mathbf{X}=[X,Y]$ are the material or original coordinates of a material point. In a finite element program, the deformation is approximated by subdividing the domain of the problem Ω into elements Ω_e such that $\Omega = \cup \Omega_e$. In each element, the deformation is approximated by shape functions N_I so that

$$\mathbf{x} = \sum_{I=1}^m N_I(\xi, \eta) x_{iI}^e(t) \quad (2.2)$$

where ξ, η are the coordinates of the element in the parent element plane and x_{iI}^e are the coordinates of node I of element e : $x_{iI}^e = [x_I^e, y_I^e]$; m is the number of nodes in the element.

The displacement field $u(\mathbf{X}, t)$ is approximated by the same shape functions

$$\mathbf{u}^e(\mathbf{X}, t) = \sum_{I=1}^m N_I(\xi, \eta) u_{II}^e(t) \quad (2.3)$$

where u_{II}^e are the nodal displacements of element e ; $u_{II} = [u_{xI}, u_{yI}]$. The nodal velocities are obtained from Equation (2.3) by taking a time derivative, which gives

$$\mathbf{v}^e(\mathbf{X}, t) = \sum_{I=1}^m N_I(\xi, \eta) v_{II}^e(t) = \sum_{I=1}^m \mathbf{N}_I \mathbf{v}_I \quad (2.4)$$

where v_{II}^e are the nodal velocities of element e . The nodal displacements and nodal velocities of the entire mesh are denoted by u_{II} and v_{II} , respectively; in vector notation, they are denoted by \mathbf{u}_I and \mathbf{v}_I , respectively.

The deformation in DYNA2D is measured by the velocity strain tensor, which is denoted by $\dot{\boldsymbol{\varepsilon}}$ and given by

$$\dot{\varepsilon}_{ij} = \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \quad \text{or} \quad \dot{\boldsymbol{\varepsilon}} = \text{sym} \nabla \mathbf{v} \quad (2.5)$$

The velocity strain is a rate of measure of deformation, rather than a measure of the total deformation, or strain, which has occurred at a point. The rate-of-deformation in an element given by

$$\dot{\boldsymbol{\varepsilon}} = \sum_{I=1}^m \text{sym} \nabla \mathbf{N}_I \mathbf{v}_I = \sum_{I=1}^m \mathbf{B}_I \mathbf{v}_I \quad (2.6)$$

where

$$\mathbf{B}_I = \text{sym} \nabla \mathbf{N}_I \quad (2.7)$$

In explicit programs, rate measures of deformation at times oscillate markedly in comparison to total measures of deformation. Furthermore, the integral of the velocity strain is not path independent, so the integral of the rate-of-deformation does not represent a good measure of the total strain.

The finite element procedure in DYNA2D uses a semidiscretization of the momentum equation. This leads to the finite element form of the equations of motion

$$\mathbf{M}_I \dot{\mathbf{v}}_I = \mathbf{f}_I^{\text{ext}} - \mathbf{f}_I^{\text{int}} = \mathbf{f}_I \quad (2.8)$$

where \mathbf{M}_I is a mass matrix, which is lumped, and consequently diagonal; a superposed dot denotes a time derivative and \mathbf{f}^{ext} and \mathbf{f}^{int} are the external and internal nodal forces; as indicated in the second equality, the difference between the external and internal nodal forces are often indicated as a net force. The internal nodal forces are given by

$$\dot{\mathbf{f}}_I^{int} = \int_{\Omega_I} \mathbf{B}_I^T \sigma d\Omega \quad (2.9)$$

where σ is the Cauchy stress.

Equation (2.8) is integrated in time by a central-difference explicit method. The central difference method is conditionally stable, and its time step depends on the smallest element in the mesh. Most simulations of nonlinear structural dynamics problems require many time steps. Generally, on the order of 10^3 or 10^5 time steps are used, but it is not unusual to have simulations with 10^6 time steps. This has important implications on the implementation of adaptivity, since the mesh can be refined only on the order of ten times in a simulation if the computational cost of adaptivity is to be reasonable.

2.3 ADAPTIVITY.

We will first describe why h -adaptivity was selected for DYNA2D among the various types of adaptivity available for finite elements analysis. There are basically three types of adaptivity:

1. r -adaptivity, in which the nodes are moved so that the degrees of freedom are concentrated in the areas where they are needed;
2. p -adaptivity, in which the order of the interpolating function in the element is increased when more accuracy is needed;
3. h -adaptivity, in which the elements are subdivided when more accuracy is needed.

The capabilities of r -adaptivity are quite limited in most practical meshes. For example, if the nodes are needed primarily on a diagonal line across a square mesh, when the starting point is a uniform mesh, it is impossible to move many of the nodes to the diagonal without inducing excessive mesh distortion. Mesh distortion degrades the accuracy of finite elements, so some of the benefits of moving the nodes are lost. Furthermore, nodes cannot be moved across material interfaces without invalidating

the model. Therefore, in general, very little additional accuracy can be gained by r -adaptivity.

In p -adaptivity, the order of the interpolating polynomial is raised where more accuracy is needed. For example if an element is initially a bilinear element, then p -adaptivity would increase the order of the element to a biquadratic element. P -adaptivity is unsuitable for this effort with explicit nonlinear-structural dynamics for two reasons:

1. It is not compatible with the approach chosen for the implementation of adaptivity, namely a paradigm in which all modifications for adaptivity are completely outside of the target computer program, because DYNA2D contains only low order elements.
2. This type of adaptivity is not advisable in explicit codes because explicit methods use diagonal mass matrices, and diagonal mass matrices of suitable accuracy have not been developed for higher order elements.

Based on these arguments, h -adaptivity was selected for this implementation of DYNA2D. In h -adaptivity, it is also possible to fuse elements, or in other words, to unrefine the mesh, when a coarser mesh is adequate. However, only refinement, which we will often call fission, is included in this implementation of h -adaptivity. The elements are not fused when the refinement is no longer necessary, because fusion of elements requires a complex data base and is quite difficult for general purpose programs.

The fission process for the elements in DYNA2D is shown in Figure 2-1. As can be seen from the figure, triangles are refined by splitting them into four triangles by connecting the midsides of the nodes. Quadrilaterals are refined by splitting them into four quadrilateral by connecting the midpoints of the sides. In the fission process, new nodes are also generated. As can be seen from Figure 2-1, the nodes generated by refinement are either free nodes or slave nodes. The motion of free nodes is completely unconstrained except for boundary conditions, so the addition of these nodes adds extra degrees of freedom to the model. The motion of slave nodes is constrained by the need to observe compatibility between adjacent elements. The nodes at the midpoints of sides of the larger elements are slave nodes; the treatment of these nodes is described later.

Each subdivision is called a level of fission. Elements may be refined an arbitrary number of levels, so that elements which are formed by splitting or subdivision of a previous element may be further subdivided. The number of subdivisions is limited in the program by an input parameter.

The fission process, the arrangement of elements is limited by the 1-irregular rule (Devloo, et al 1987). This rule requires that the difference in the level of subdivision between adjacent elements be at most two. As a consequence, at most one slave node can be inserted in any edge.

The motion of slave nodes is constrained by the motion of the adjacent corner nodes so that compatibility is maintained between the subdivided element and the larger element adjacent to it. The displacement and velocity fields for elements in DYNA2D, the three node triangle and four node quadrilateral, are linear along the edges. The motion of a slave node must conform to the linear velocity field of the larger element to satisfy compatibility. For example, if we consider nodes 1, 2 and 3 in Figure 2-1, then to maintain compatibility of the adjacent elements along this edge the velocity of the midpoint node, that is the slave node, must be given by

$$\mathbf{v}_2 = \frac{1}{2}(\mathbf{v}_1 + \mathbf{v}_3) \quad (2.10)$$

where \mathbf{v}_I are the nodal velocities; $\mathbf{v}_I = [v_{xI}, v_{yI}]$ where v_{xI} and v_{yI} are the x and y components of nodes I , respectively. The equations of motion, Equations 2.8, do not pertain to slave nodes. Therefore, the forces on the slave nodes must be transferred to master nodes. Nodes which are not constrained, such as nodes 1 and 3, are called master nodes. The internal force at node 2 is transferred to nodes 1 and 3 according to the following procedure based on the principle of virtual work. Let the internal forces at node I be denoted by $\mathbf{f}_I^{int} = [f_{xI}^{int}, f_{yI}^{int}]$. By the principle of virtual work, the virtual work of nodal force transferred from slave node 2 to the virtual work of the nodal forces transferred to master nodes 1 and 3. This means that

$$\delta \mathbf{v}_2 \cdot \mathbf{f}_2^{int} = \delta \mathbf{v}_1 \cdot \mathbf{f}_1^{int} + \delta \mathbf{v}_3 \cdot \mathbf{f}_3^{int} \quad (2.11)$$

From Equation (2.10),

$$\delta \mathbf{v}_2 = \frac{1}{2}(\delta \mathbf{v}_1 + \delta \mathbf{v}_3) \quad (2.12)$$

so substituting the above into Equation (2.11) yields

$$\frac{1}{2}(\delta \mathbf{v}_1 + \delta \mathbf{v}_3) \cdot \mathbf{f}_2^{int} = \delta \mathbf{v}_1 \cdot \mathbf{f}_1^{int} + \delta \mathbf{v}_3 \cdot \mathbf{f}_3^{int} \quad (2.13)$$

The above must hold for arbitrary $\delta \mathbf{v}_1$ and $\delta \mathbf{v}_3$ so it follows that

$$\mathbf{f}_1^{int} = \frac{1}{2} \mathbf{f}_2^{int} \quad (2.14)$$

$$\mathbf{f}_3^{int} = \frac{1}{2} \mathbf{f}_2^{int} \quad (2.15)$$

These nodal forces are added to the nodal forces which arise at nodes 1 and from the elements directly connected to these nodes.

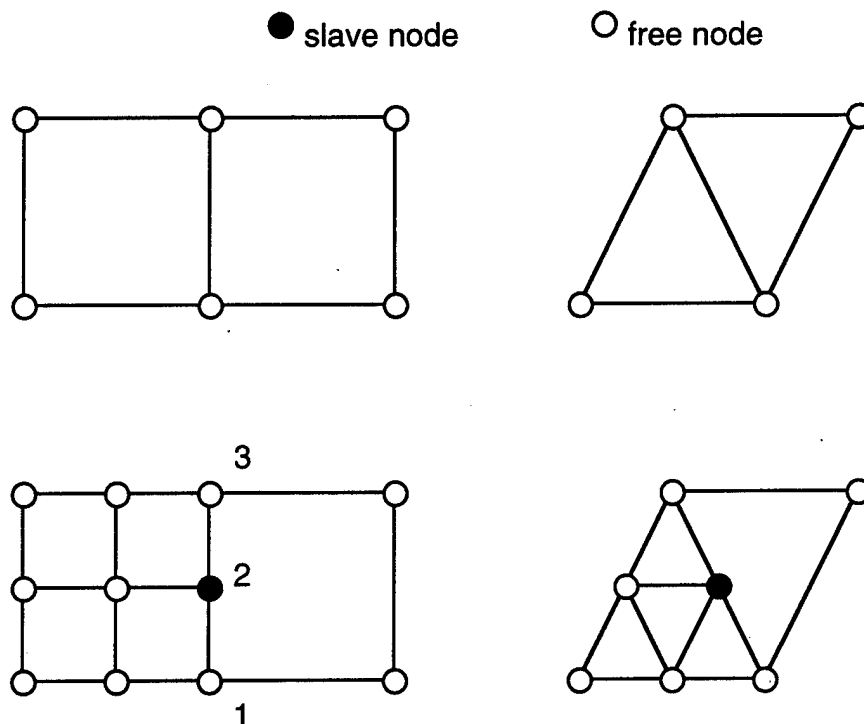


Figure 2-1. Example of h -adaptivity for a quadrilateral element; the left element has been refined one level.

2.4 ERROR AND REFINEMENT INDICATORS.

In adaptive methods, an error or refinement indicator must be computed to determine which elements need to be refined. Error indicators are distinguished from refinement indicators in that the latter make no attempt to estimate an error but simply refine on the basis of certain characteristics of the solution. In nonlinear, transient analysis, little theoretical justification is available for the error indicators which are used. However, preliminary studies (Belytschko and Tabbara 1993) show that the indicators used in linear analysis are also quite effective in nonlinear analysis. In addition, to an error indicator, several refinement indicators are included in this program. These refinement indicators use the magnitude of specific physical quantities to drive the adaptivity.

The error indicator selected here is based on an estimate of the error in the Green-Lagrange strain. The estimate is obtained by a projection of the finite element solution with the moving least-square interpolation described in (Tabbara, et al 1994). The projection criterion developed here is very similar to the local projection error indicator (Zienkiewicz and Zhu 1987) except that the projected solution is obtained by a moving least-square approximation, which has been found to be a more accurate in (Tabbara, et al 1994). This error criterion is an improvement of the earlier global projections of (Zienkiewicz and Zhu 1987). This procedure is also known as error estimation by a recovery technique.

In a recovery technique, a local projection is used to obtain an approximation to the exact solution denoted by $u^*(X,t)$. This projected solution is obtained by minimizing the L2 norm of the difference between the projected i.e. recovered solution and the finite element solution. The error in the Green-Lagrange stain in an element e of any function $f(X,t)$ is denoted by θ_e , and is given by

$$\theta_e = \Omega_e^{-1} \int_{\Omega_e} (E_{ij}^* - E_{ij}^h)(E_{ij}^* - E_{ij}^h) d\Omega \quad (2.16)$$

where E_{ij}^* is obtained from the projected solution and E_{ij}^h is the Green-strain computed from the finite element solution. As can be seen from the above, the integral of the error is divided by the domain (or area in two dimensions) of the element, so it is a density of error.

The rate-of-deformation tensor which is used in DYNA2D for constitutive evaluations Equation (2.5) is not suitable as an error criterion because it is not path-independent, furthermore, since it gives a rate of strain it tends to be noisy. The Green-Lagrange strain is path independent and a measure of strain. Therefore, it is more suitable for our own error criterion. The Green-Lagrange strain tensor E is given by

$$E_{ij} = \frac{1}{2}(F_{ki}F_{kj} - \delta_{ij}) = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \frac{\partial u_k}{\partial X_i} \frac{\partial u_k}{\partial X_j}\right) \quad (2.17)$$

where F_{ki} is the deformation gradient, which is given by

$$F_{ij} = \frac{\partial x_i}{\partial X_j} = \frac{\partial u_i}{\partial X_j} + \delta_{ij} \quad (2.18)$$

and δ_{ij} is the Kronecker delta, or unit matrix.

The projected Green-Lagrange strain is obtained from a moving least-square approximation for the displacement field. For the purpose of computing this approximation, a local circular domain Ω_g is defined about the centroid of the element of interest. The radius of the domain is denoted by R_g ; it is set by a procedure described subsequently. This domain is called the domain of influence of the approximation.

The projection of the displacement field $\mathbf{u}^*(\mathbf{X}, t)$ is obtained by a weight least square projection. For this purpose, the displacement fields are approximated in the subdomain Ω_g by

$$u_i^*(\mathbf{X}, t) = \mathbf{P}(\mathbf{X})\mathbf{a}_i(\mathbf{X}, t) \quad (2.19)$$

where $\mathbf{P}(\mathbf{X})$ is a monomial basis and $\mathbf{a}_i(\mathbf{X}, t)$ are coefficients which vary in space. A bilinear basis is used for $\mathbf{P}(\mathbf{X})$, so $\mathbf{P}(\mathbf{X})$ is given by

$$\mathbf{P}(\mathbf{X}) = [1, X, Y, XY] \quad (2.20)$$

In the moving least-square approximation, the parameters $\mathbf{a}_i(\mathbf{X}, t)$ at any time t are determined by minimizing the weighted quadratic form

$$S(\mathbf{a}_i(\mathbf{X}, t)) = \sum_{I=1}^n w(\mathbf{X} - \mathbf{X}_I) (\mathbf{P}(\mathbf{X}_I)\mathbf{a}_i(\mathbf{X}, t) - u_{Ie}^h)^2 \quad (2.21)$$

where u_i^h are the components of the finite element nodal displacements at node I and $w(\mathbf{X} - \mathbf{X}_I)$ is a weight function given by

$$w(\mathbf{X} - \mathbf{X}_I) = \exp(-\|\mathbf{X} - \mathbf{X}_I\| / (0.4d_m)^2) \quad (2.22)$$

and d_m is twice the maximum element diagonal for any element in the domain of influence. This choice for d_m is based on experience which shows that the domain of influence should include the elements which surround the element under consideration but that the exponential should have decayed markedly at the nodes which are not part of the element under consideration. The minimization of $S(\mathbf{a}_i(\mathbf{X}, t))$ leads to a system of linear equations in $\mathbf{a}_i(\mathbf{X}, t)$. The gradient of the displacement field is then obtained using Equations (2.19) and (2.20) which gives

$$\frac{\partial u_x^*}{\partial X} = a_{2x} + a_{4x}Y \quad (2.23)$$

$$\frac{\partial u_x^*}{\partial Y} = a_{3x} + a_{4x}X \quad (2.24)$$

$$\frac{\partial u_y^*}{\partial X} = a_{2y} + a_{4y}Y \quad (2.25)$$

$$\frac{\partial u_y^*}{\partial Y} = a_{3y} + a_{4y}X \quad (2.26)$$

These gradients are then substituted into Equation (2.17) to obtain the projected Green-strain E_{ij}^* . Note that the derivatives of the coefficients a_i are not considered in computing the gradients.

The remaining indicators included in the program are not based on an estimate of error but are refinement indicators which drive refinement on the basis of certain physical quantities that are likely to be high where refinement is needed. The following are included:

1. The effective stress which is given by

$$\bar{\sigma} = \left(\frac{3}{2} s_{ij} s_{ij} \right) \quad (2.27)$$

where s_{ij} is the deviatoric stress which is given by

$$s_{ij} = \sigma_{ij} - p\delta_{ij} \quad (2.28)$$

2. The trace of the stress tensor which is given by

$$trace(\sigma_{ij}) = \sigma_{kk} \quad (2.29)$$

3. The effective plastic strain is given by

$$\bar{\epsilon}^p = \int_0^t d\epsilon^p \quad (2.30)$$

where

$$d\epsilon^p = \left(\frac{2}{3} d\epsilon_{ij}^p d\epsilon_{ij}^p \right)^{1/2} \quad (2.31)$$

2.5 IMPLEMENTATION.

H-adaptivity was implemented in DYNA2D by developing an external program which performs the calculation of error criteria and the remeshing. The details of the implementation are described subsequently, after the procedure is described.

The basic procedure of *h*-adaptivity for nonlinear transient problems is summarized in Table 2-1. As can be seen from Table 2-1, the adaptive program updates the mesh by integrating the equations of motion a selected time interval, denoted by Δt_{adapt} , and then checks the refinement or error criteria for all of the elements in the mesh. Any elements for which the refinement criteria are exceeded are then subdivided.

The program then repeats the time interval Δt_{adapt} with the refined mesh and writes a restart tape; it then continues for the time interval Δt_{adapt} beyond the time at which the error criterion was previously checked before checking refinement criteria again. If the refinement criteria indicate that any elements need to be refined, the elements are subdivided and the computation is restarted with the restart tape which was previously written. The procedure is called a go-back procedure.

Table 2-1. Flowchart of Adaptive Procedure.

1. Initialize, set $t_{adapt} = \Delta t_{adapt}$.
2. Integrate the equations of motion, Equations 2-7, to a specified time t_{adapt} .
3. Check error or refinement indicators θ_e for all elements e ,
if $\theta_e >$ refinement threshold for element e , subdivide element e .
4. Set $t_{adapt} = t_{adapt} + \Delta t_{adapt}$ and go to 2.

A schematic of the go-back procedure is shown in Figure 2-2, which shows the time line for a computation with two refinement steps involving a total of three meshes: Mesh 1, Mesh 2, and Mesh 3. As can be seen from Figure 2-2, the computation starts with Mesh 1, which is refined at the first error check to obtain Mesh 2; Mesh 2 repeats the computation over the time interval from $(0 < t < \Delta t_{adapt})$ and then continues to $2\Delta t_{adapt}$. Time histories are stored for the interval $0 < t < \Delta t_{adapt}$ only for the Mesh 2 computations, not for the Mesh 1 computation. At $2\Delta t_{adapt}$, the next error check is made. In the literature, for example (Devloo, Oden and Strouboulis 1987), it is often advocated that the computation not proceed beyond the last error check until the error criteria are satisfied. Unfortunately, this leads to the possibility that the computation will iterate many times in a single time interval, which is not desirable in a production program.

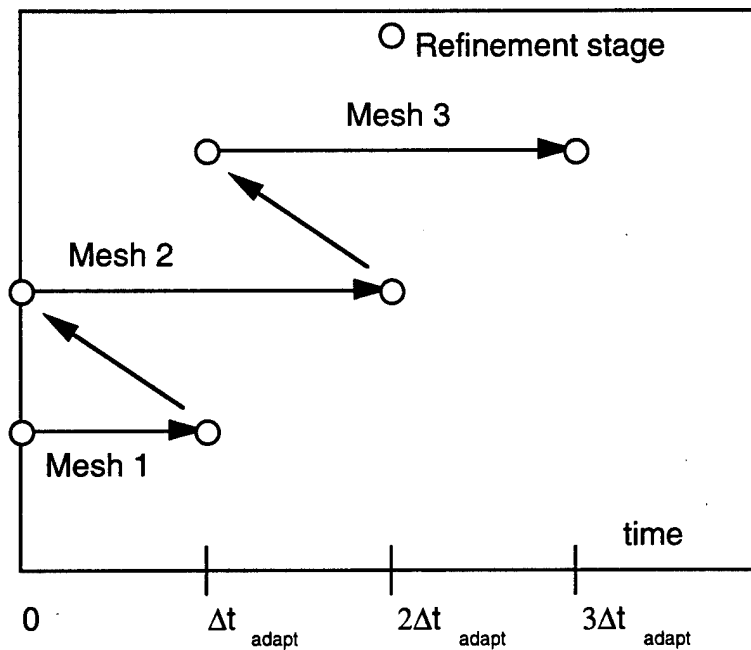


Figure 2-2. Flow of computations in adaptivity with a go-back procedure.

In the go-back procedure, two different meshes are used for the computations of any time interval. Thus the adaptive computation will be at least twice as expensive as a coarse mesh computation.

The refinement procedure involves four steps:

1. check the error or refinement criteria for all elements;
2. generate new nodes and elements where refinement is needed;
3. generate new starting data for new elements and nodes;
4. generate new data for boundary conditions, loads, etc.

In first step, the error or refinement indicators are computed by the external program, according to the equations in Section 2-4. These indicators are checked against a user-input tolerance, and any element which exceeds the tolerance is listed for subdivision. Only elements which have not exceeded a specified number of levels of refinement are refined. In the refinement procedure, new elements and nodes are generated. The new nodes and elements are numbered according to the following: the lower-left element retains the same number as the original element, and the other three new elements are added to the end of the element list; the new nodes are always added to the end of the node list. Nodes are classified as free nodes or slave nodes depending on the refinement of the contiguous elements.

Slave nodes which are generated by refinement are treated similarly to tielines in DYNA2D. The actual DYNA2D routines could not be used for the slave nodes so a modified routine was written which implements Equations (2.10) and (2.14-2.15).

The starting values of stresses and other state variables must be set in the elements which have been generated by the refinement, and the velocities and displacements are initiated for the new nodes. The new starting data are obtained by the REZONE algorithm in DYNA2D, which sets new values of element and nodal variables according to a least square projection. The descendant elements are assigned the same material properties as the parent element.

The fourth step requires the following data to be provided so that it reflects the addition of the new nodes to the model:

1. boundary conditions
2. nodal loads, i.e. f^{ext} .
3. slideline data

This fourth step is quite difficult, and it has led to problems in this program and many other adaptive programs, for it requires an interpretation of the problem statement from the discrete data. In other words, it is necessary to interpret the data for the initial mesh in order to develop the data for the new nodes.

For the boundary conditions, the following rule is used. The original nodes are called the edge parent nodes. The boundary condition of the new node is set according to the boundary conditions of the two parent nodes, if the parent nodes have the same boundary condition. If the two parent nodes have different boundary conditions, but the constraints on one node are less restrictive, then the less restrictive boundary conditions are assigned to the new node. For example, if the boundary conditions for parent nodes A and B are:

node A: x-component fixed, y-component fixed;

node B: x-component fixed;

then the x-component is fixed for the new node between nodes A and B. If the boundary conditions of the nodes are such that the conditions on one parent node are not a subset of the conditions of the other parent node, an error is flagged.

For external loads specified by a pressure or shear load card (page 157 in the DYNA2D manual) the new nodes are added to the data. Therefore, the pressure and shear loads are generated exactly as in a regular run.

A similar procedure was tried for the slideline data. However, we have not been able to successfully continue a run with slidelines in which adaptivity introduced new nodes along the slidelines. This option requires further work.

The program for these functions is in a block of subroutines with a root LP. These subroutines are called from DYNA2D. Two ways of making adaptive runs have been included:

1. automatic adaptivity, which completes a run using the go-back procedure previously;
2. interactive adaptivity procedure, in which the user can initiate adaptivity at any time on the basis on run-time plots by interrupting the run.

SECTION 3 EXAMPLES OF H-ADAPTIVE CALCULATIONS

3.1 TAYLOR IMPACT PROBLEM.

To illustrate the performance of the h -adaptive procedure, the Taylor rod impact problem, which is a standard DYNA2D problem was solved. The problem was solved in three ways: using a coarse mesh of 3×10 elements, using a fine mesh of 12×10 elements, and using the h -adaptive procedure starting from the coarse mesh. The problem is axisymmetric, so axisymmetric elements were used. In the h -adaptive simulation, the MLS error criterion was used with a refinement threshold of 0.002 and two levels of adaptivity were permitted. The total simulation time was 80ms. Error checks and adaptivity were performed every 5ms, so a changes were made in the mesh 16 times.

The rod is a copper rod with a Young's modulus 117GPa and a Poisson's ratio of 0.4. The material is isotropic-elastic-plastic with a yield stress of 400 MPa and a target modulus of 100 MPa. The initial velocity of the rod was 227m/s, and the bottom nodes were constrained in the z-direction at the start of the simulation to model an impact with a rigid wall.

The evolution of the adaptive mesh is shown in Figures 3-1 and 3-2. It can be seen that the refinement slowly progresses upward from the impacted surface, but that the top of the rod is never refined. For comparison, the deformed mesh for the fine-mesh solution is shown in Figure 3-3. Comparison of the two meshes reveals good agreement.

The displacement at the centerpoint of the free end and midpoint of the rod are compared for the three different meshes in Figure 3-4. It can be seen that the h -adaptive solution agrees exactly with the fine mesh solution, whereas the coarse mesh solution differs moderately from the two. The velocities for these two points are compared in Figure 3-5.

Again, the adaptive results compare very well with the fine mesh results. At several points in time, the h -adaptive solution exhibits moderate oscillations. In some instances, these appear to be associated with the refinement of the elements near the node where the velocity is output.

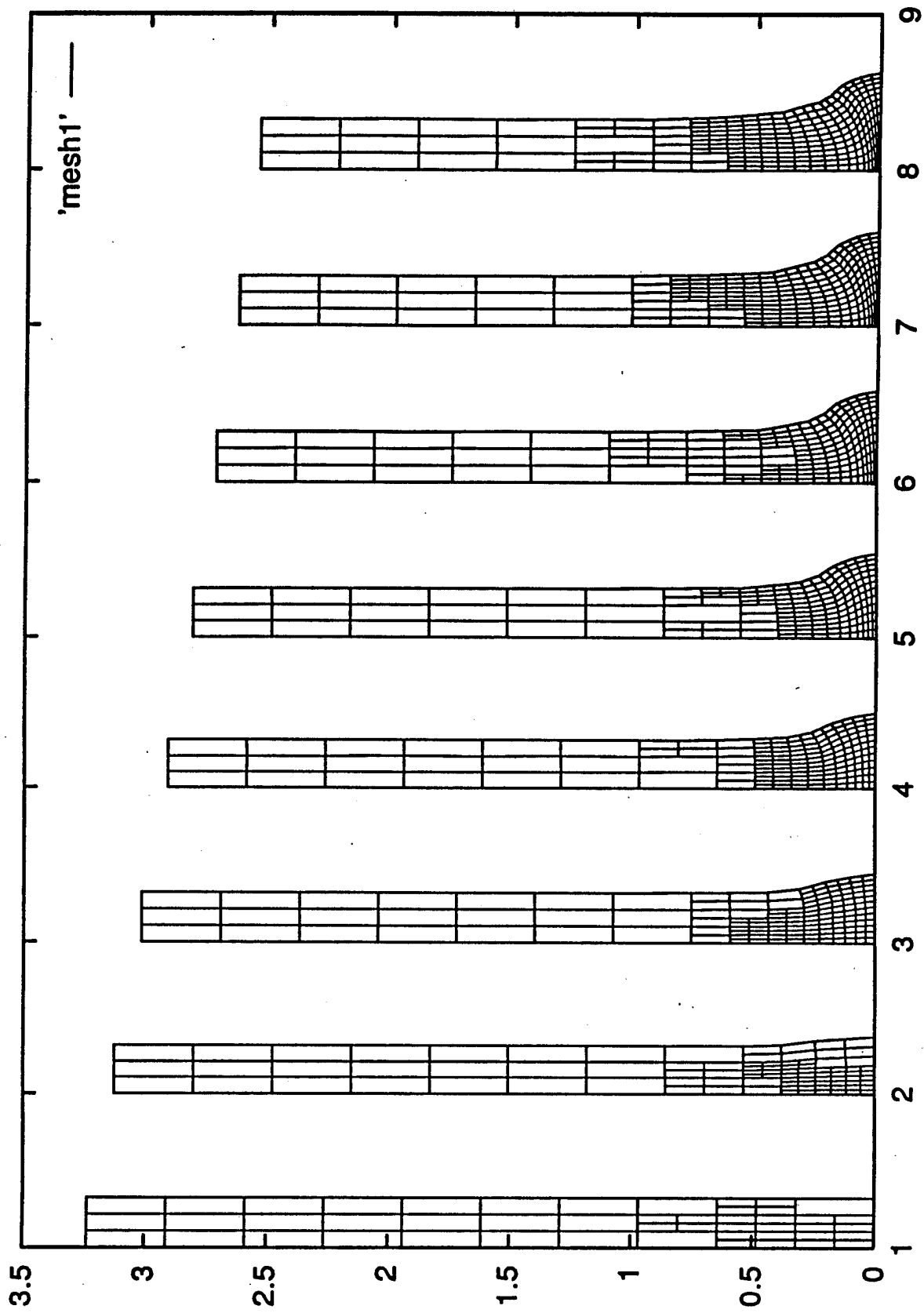


Figure 3-1. Deformed meshes for the Taylor impact problem for h -adaptive solution at $5\mu\text{s}$ intervals from $5\mu\text{s}$ to $40\mu\text{s}$.

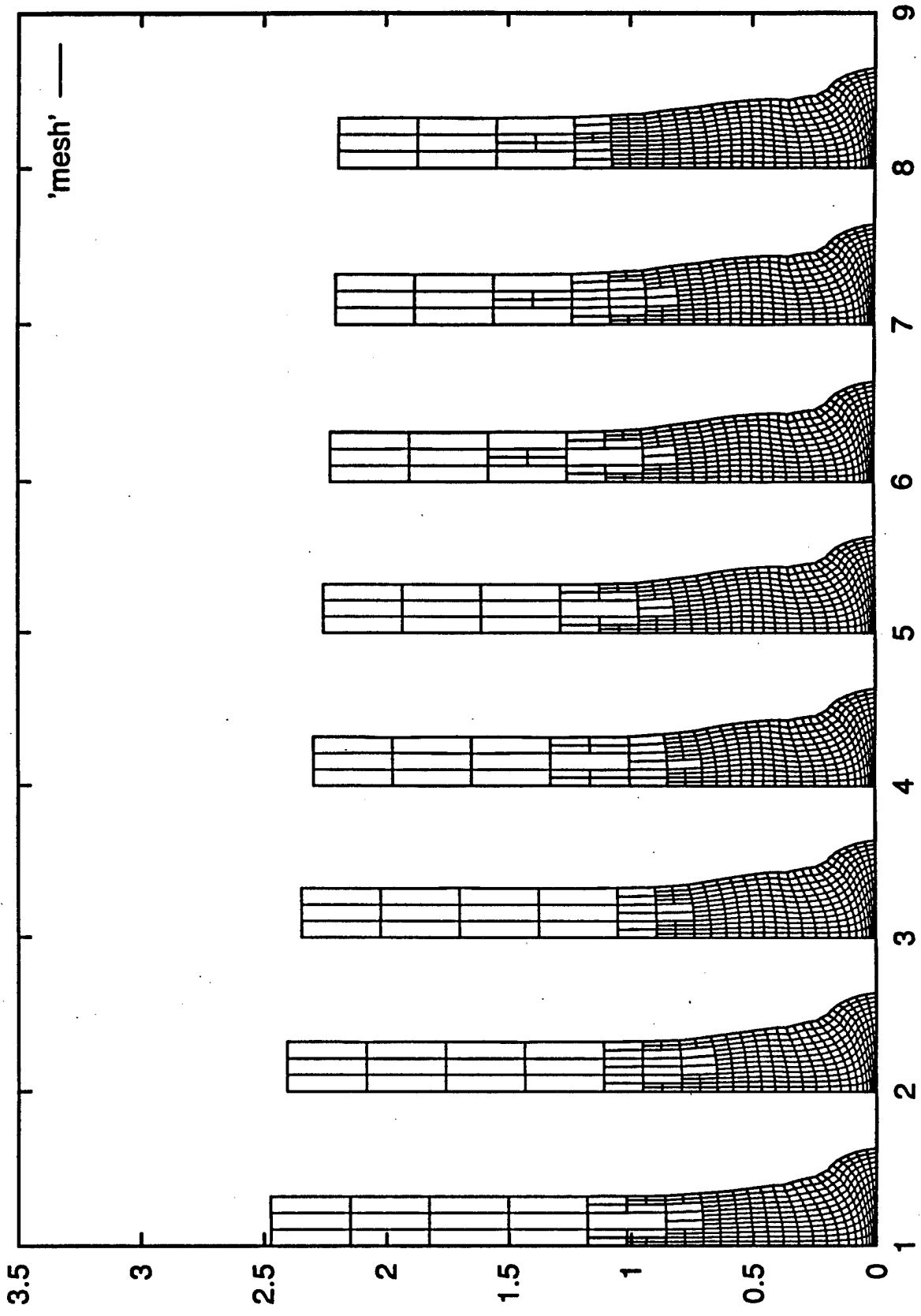


Figure 3-2. Deformed meshes for Taylor impact problem for h-adaptive solution $5\mu\text{s}$ intervals from $45\mu\text{s}$ to $80\mu\text{s}$.

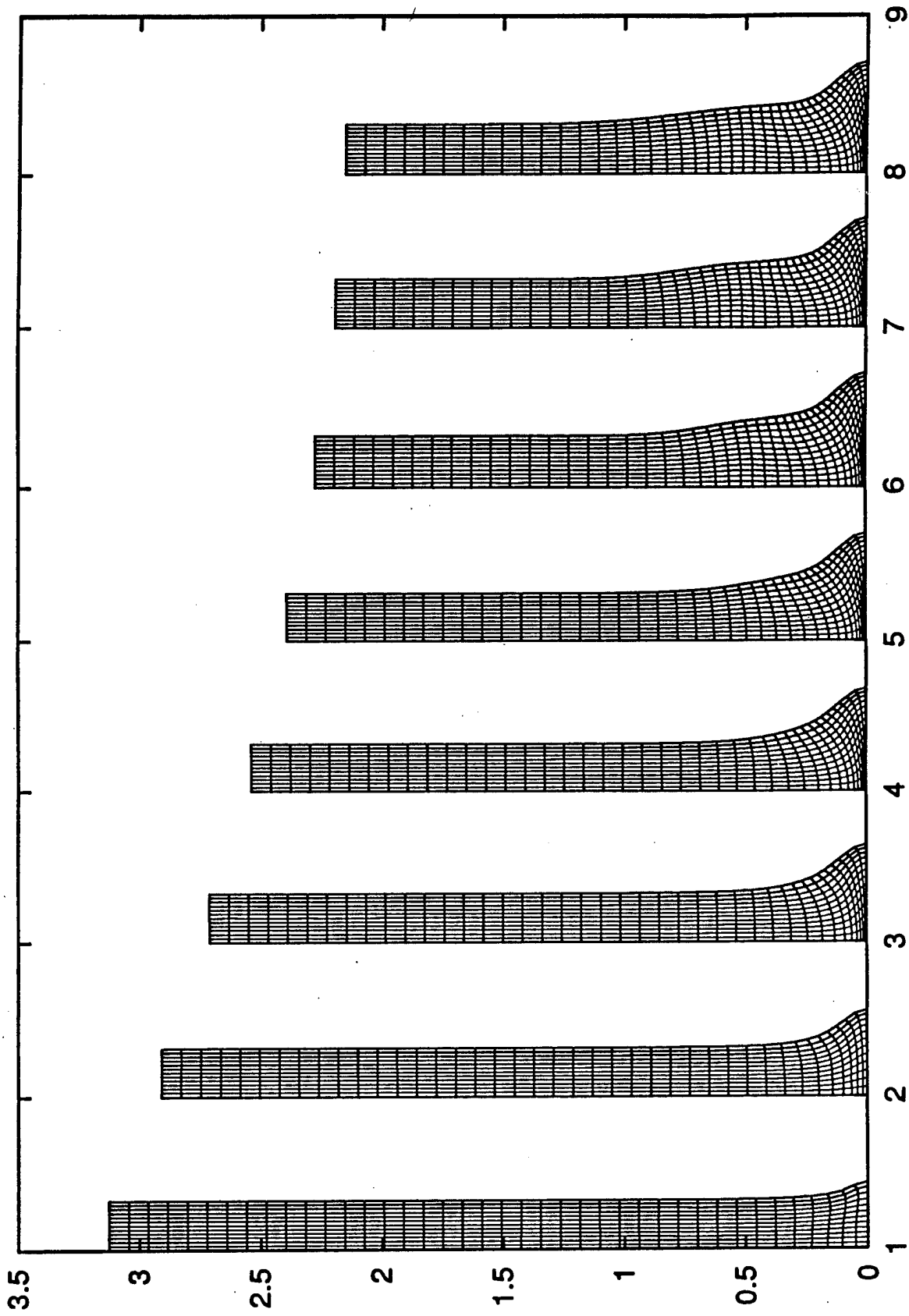


Figure 3-3. Deformed meshes for the fine mesh simulation at 10 μ s intervals from 10 μ s to 80 μ s.

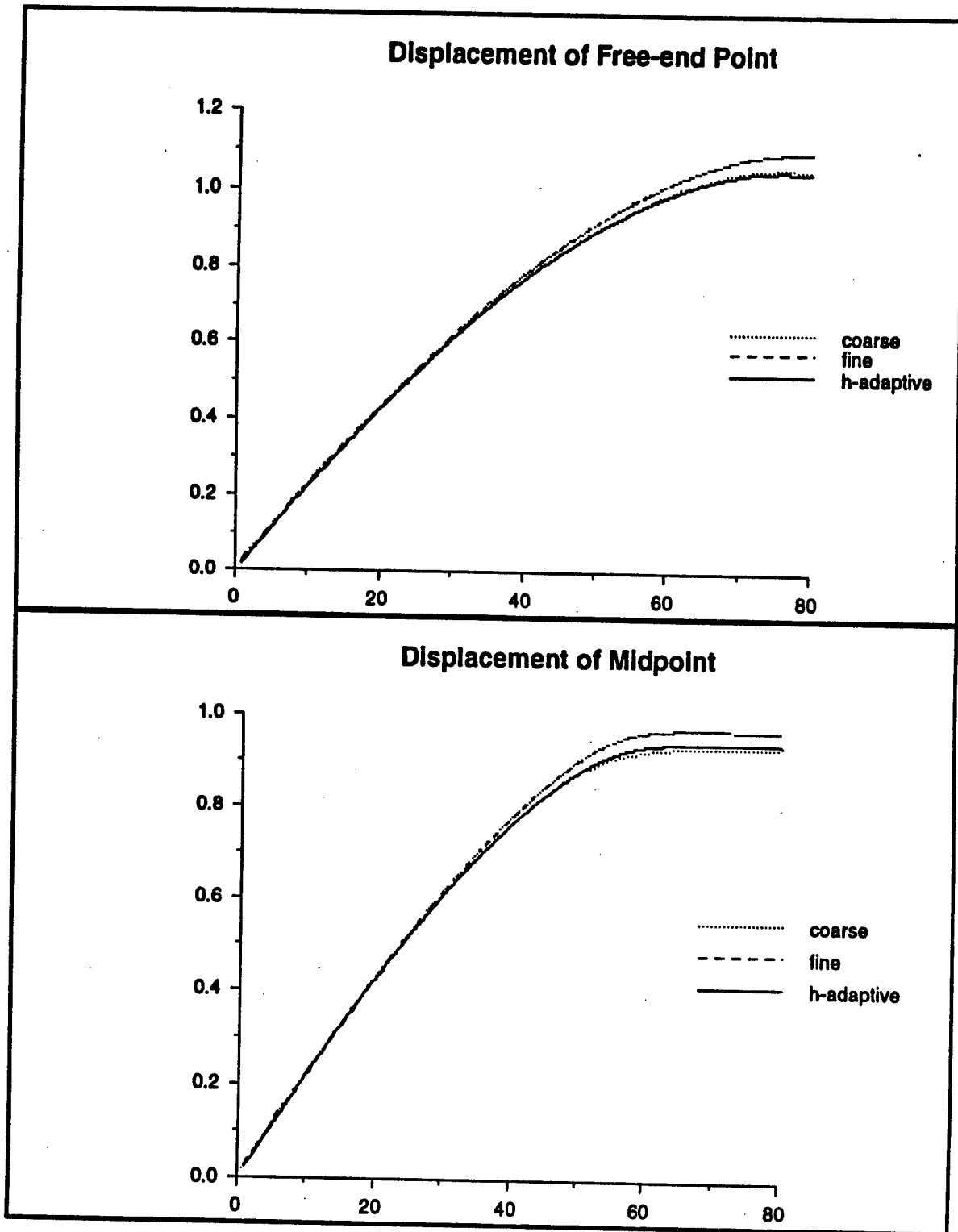


Figure 3-4. Axial displacements of free-end point and midpoint on the axis of symmetry for Taylor rod problem; time is in μs .

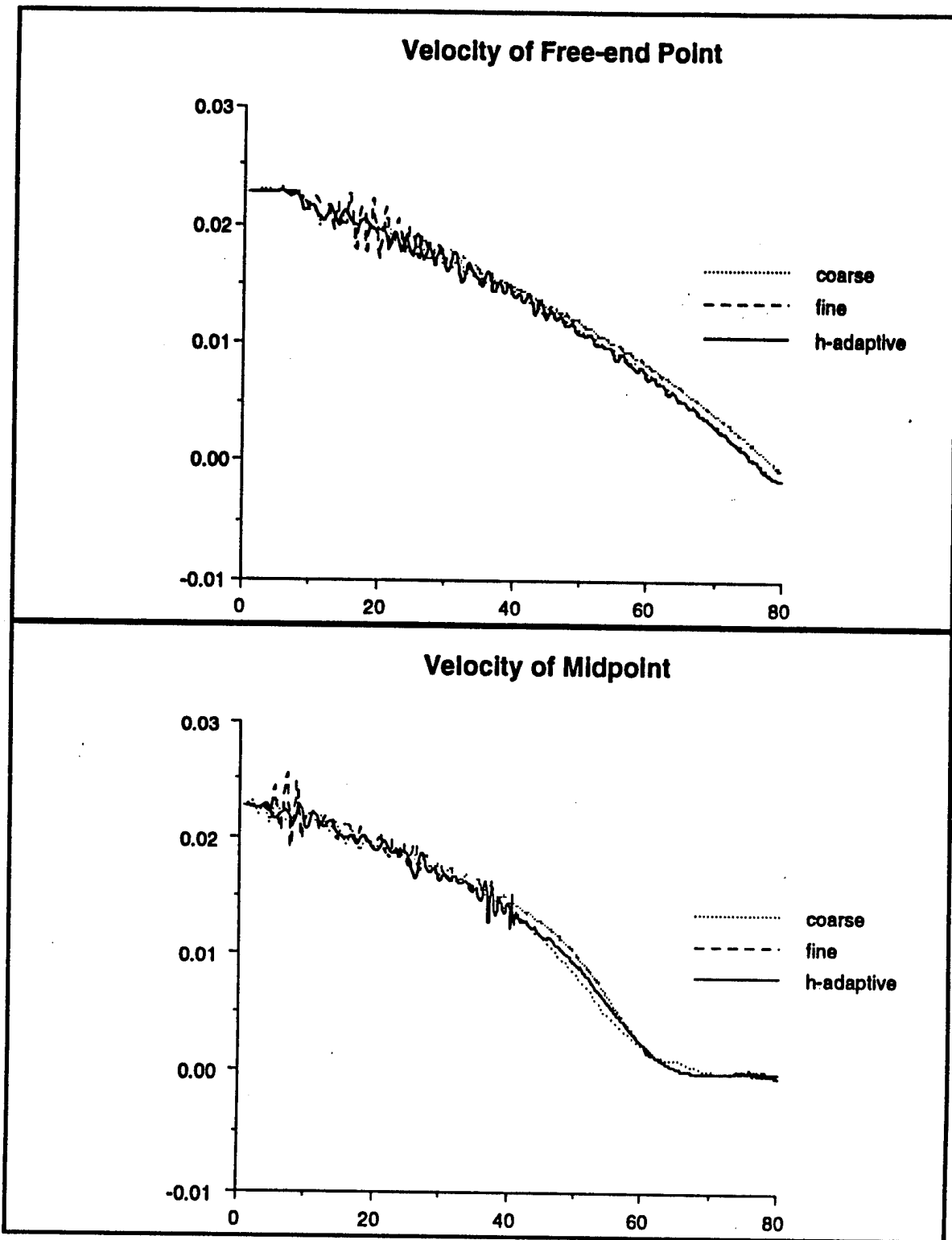


Figure 3-5. Axial velocities of free-end point and midpoint on the axis of symmetry for Taylor rod problem; time is in μs .

Figure 3-6 shows the axial stress σ_{rr} at the midpoint of the rod on the axis of symmetry. The h -adaptive agrees very well with the fine mesh solution, particularly at later times, but the h -adaptive solution exhibits severe noise at about 40ms. This corresponds to the time when refinement occurs in the area of the node for which the velocity is given. The noise decays quite rapidly in time and appears to have no permanent effect on the solution.

The fine mesh solution requires 55 minutes of computer time, the h -adaptive solution required 35 minutes. A SUN-SPARC 1 work station was used. The savings in computer time are moderate but nevertheless significant. It should be noted that substantially more time could be saved if the DYNA2D program had a subcycling capability so that different time steps could be used in different parts of the mesh. DYNA2D uses a single time step for the entire mesh, so that when the h -adaptivity introduces smaller elements, the integration time step for the entire mesh is needed.

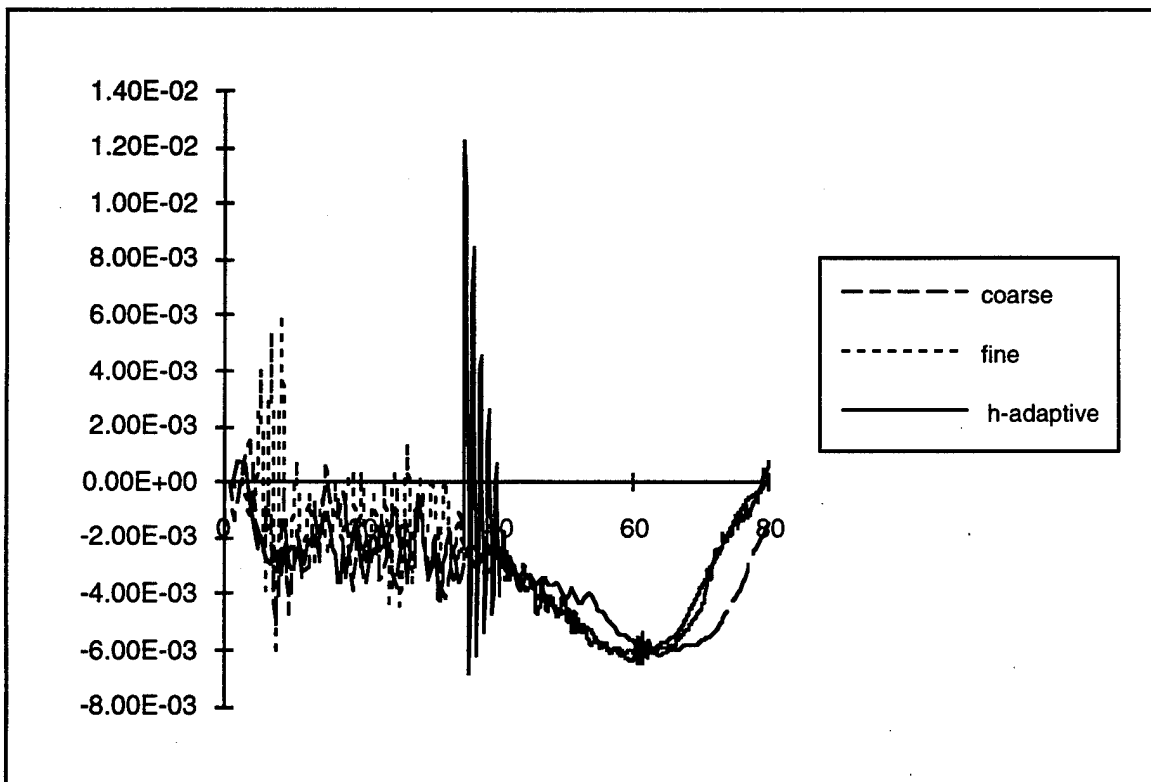


Figure 3.6 The axial stress σ_{zz} at the midpoint of the axis of the rod; time is in μ s.

3.2 BEAM PROBLEMS.

The response of a cantilever beam was simulated for an elastic material and an elastic-plastic material. The problem definition is given in Figure 3-7, all units are non-dimensional. A vertical load of magnitude 40 which remains in the vertical direction is applied as a step function in time at the free end; the load is a traction with a parabolic distribution which is maximum at the centerline and vanishes at the top and bottom. The material parameters are as follows: Young's modulus=10,000, Poisson's ratio=300, yield stress=300, the plastic tangent modulus=100, and the density =1; all units are non-dimensional.

The problem was solved with three meshes: a coarse mesh with 2 elements in the vertical direction, 10 elements in the horizontal direction; a fine mesh with 8 elements in the vertical direction and 40 elements in the horizontal direction; an h -adaptive solution starting with the coarse mesh and two levels of refinement. For the adaptive solution, the projected Green strain error indicator was used with a threshold of 0.005 for refinement.

The time-histories of the deflection of the end-point for the three runs for the elastic beam is shown in Figure 3-8; for the elastic solution, no yielding was allowed. It can be seen that the adaptive solution agrees reasonably well with the fine-mesh solution, although the agreement between the h -adaptive and fine mesh solutions is not perfect. Some of the discrepancy may be caused by the differences between the loading in the two solutions, which results from the fact that in the fine mesh run, the load was distributed over five nodes at the end, whereas for the coarse mesh solution the load is distributed over 3 nodes.

The time histories of the end-point deflections for the three runs with the elastic-plastic beam are shown in Figure 3-9. Again, there are moderate differences between the h -adaptive solution and the fine mesh solution, but the h -adaptive solution agrees much better with the fine mesh solution. The deformed meshes are shown in Figure 3-10. As can be seen, the refinement is concentrated at the supported edge where the strains and strain gradients are maximum.

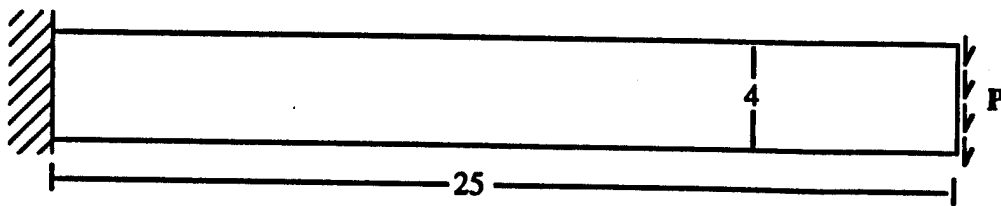


Figure 3-7. Description of beam problem.

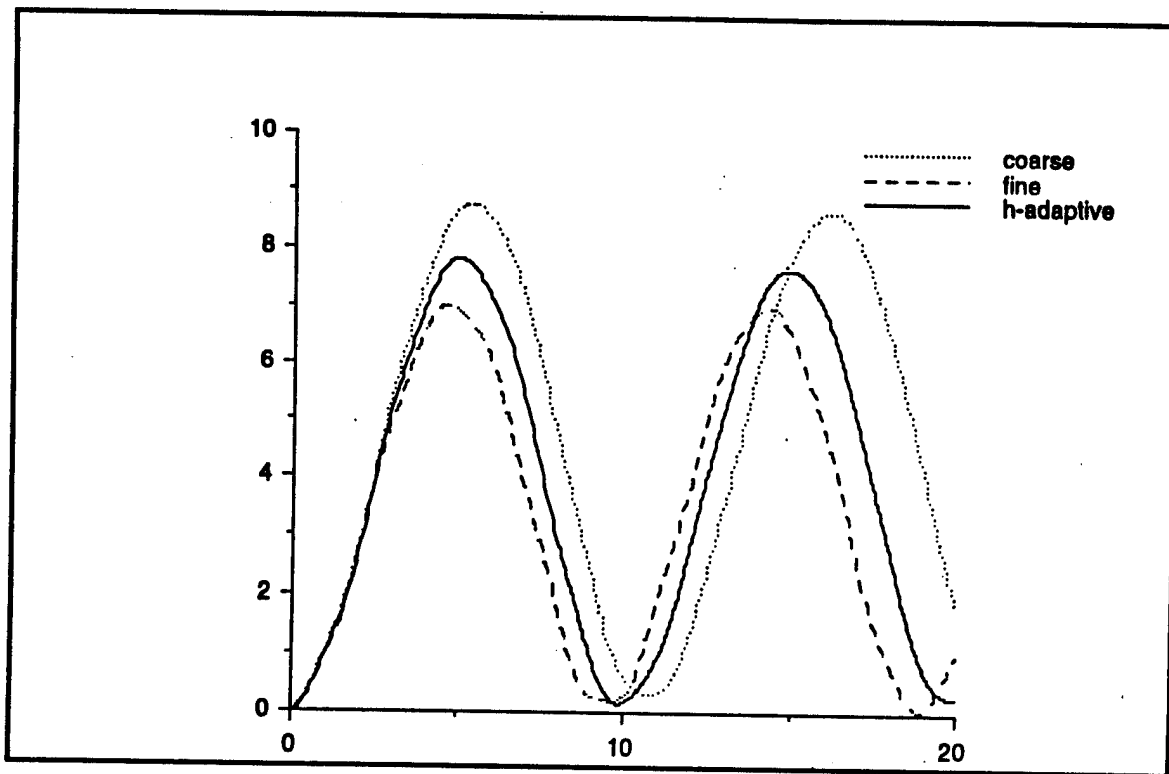


Figure 3-8. Displacement of free end of elastic beam as a function of time.

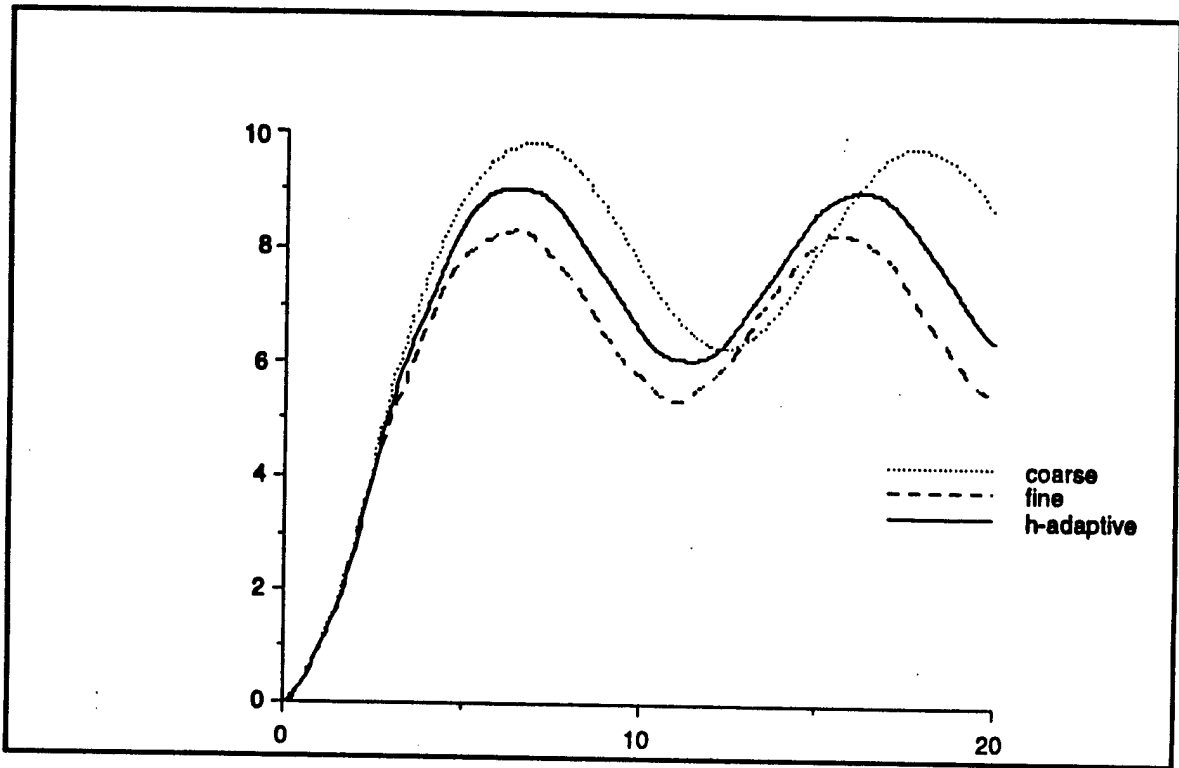


Figure 3-9. Displacement of free end of elastic-plastic beam as a function of time.

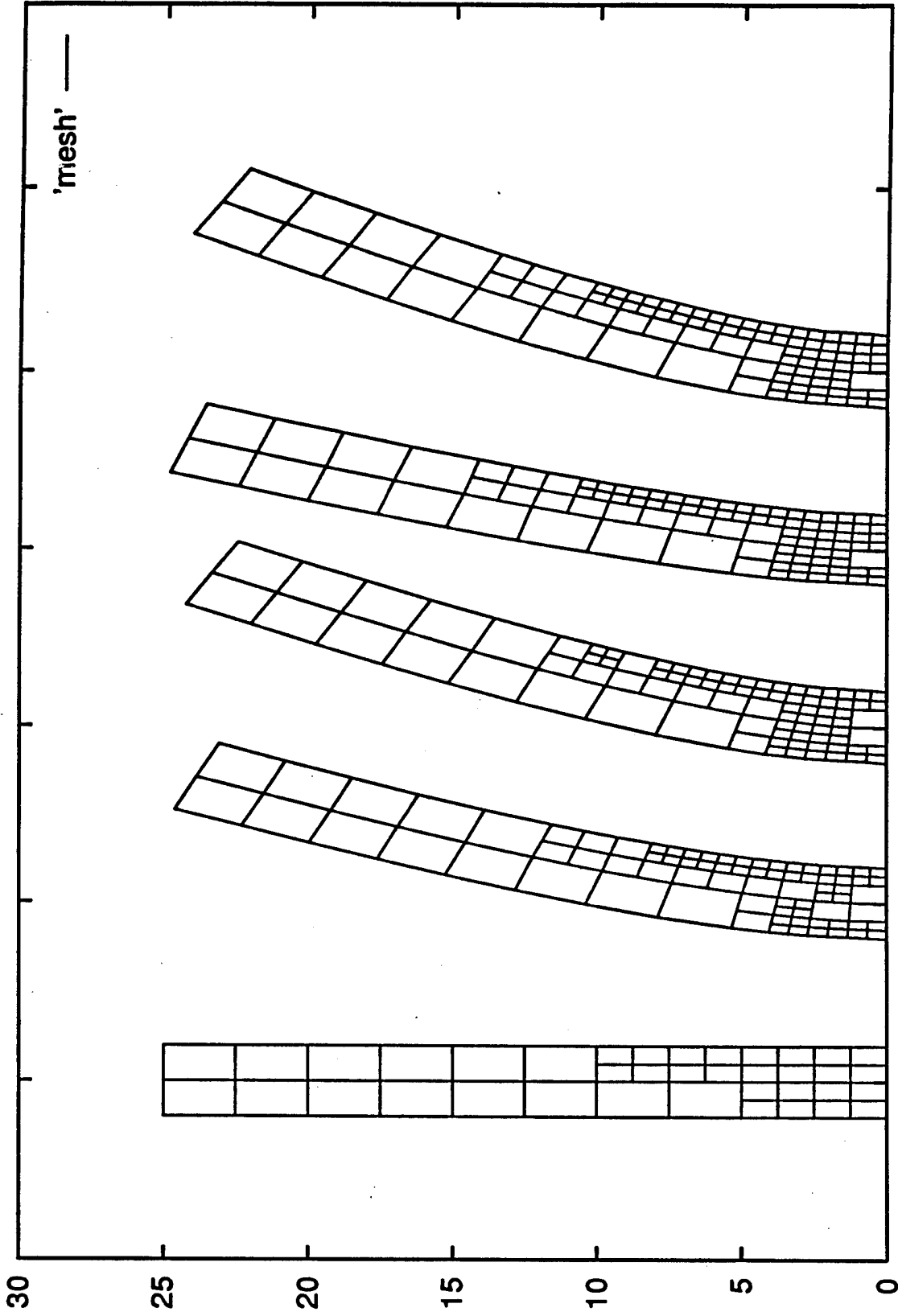


Figure 3-10. Deformed meshes for the elastic-plastic beam at four stages of deformation.

SECTION 4 CONCLUSIONS

An adaptive procedure has been implemented in the DYNA2D program for the nonlinear transient analysis of solids. A procedure was developed so that the program for implementing h -adaptivity is totally separate from the DYNA2D program. This enables the procedure to be used with other versions of the DYNA2D program, such as subsequent releases or modified versions developed in-house by various organizations. The new procedure utilizes the rezone and remap algorithms in DYNA2D in conjunction with the restart options. Thus the programming for adaptivity deals primarily with the implementation of the error and refinement criteria and with the selection of elements to be subdivided and is completely separate from DYNA2D.

In the h -adaptive procedure developed here, the mesh can be refined to several levels by subdividing quadrilaterals into four quadrilaterals. The error and refinement criteria select the elements which are to be refined. Several error and refinement indicators were incorporated in the program: a projection-based error indicator using the Green strain and refinement indicators based on effective stress and effective plastic strain. In preliminary studies, we also tested error indicators based on projections on the stress. We found that error indicators were not as useful for the class of nonlinear materials which were studied, elastic-plastic type materials, because the stress in the plastic zone is spatially quite uniform even when there are large variations in the strains. In many situations, a criterion for refinement based on the magnitude of the effective plastic strain works quite well, because refinement is usually needed in the areas of most severe nonlinearities and these are the areas where the effective plastic strain is very large. The projection criterion based on Green strain was quite effective in refining in areas of plastic response.

Comparisons were made between coarse mesh, fine mesh and h -adaptive solutions for three problems: the Taylor impact problem, an elastic beam problem and an elastic-plastic beam problem. The results show that the h -adaptive solutions compare quite well with the fine mesh solution even though the h -adaptive solution requires substantially fewer elements. It was found that the moving least square projection indicator based on Green strain developed here is quite effective in selecting the elements that need to be refined. The running time for the h -adaptive solutions is approximately 30-40% less than for the fine mesh solutions. This is a substantial saving

but has not been achieved, for example in h -adaptive procedures which are directly incorporated in a program. For comparable problems, fine mesh solutions generally took two and a half times as long as an adaptive solution.

There are two reasons for the failure to achieve larger savings of computation time:

1. the use of an external program to implement adaptivity involves the writing of restart files and the transfer between the two programs which is not necessary when h -adaptivity is built into a program;
2. the DYNA2D program does not have a subcycling, in which different time steps can be used in different parts of the mesh, so that the smaller elements can be integrated by a smaller time step, while the remainder of the mesh is integrated with larger time step.

The absence of subcycling is particularly costly in h -adaptive procedures because every level of refinement decreases the stable time step by a factor of two. Therefore it is important that explicit structural dynamics program with h -adaptivity have a subcycling feature. It is recommended that subcycling be incorporated in the DYNA2D codes for it not only improves the performance of the problems in h -adaptive problems but can also lead to substantial computational savings in standard solution procedures.

Another difficulty which we encountered was in defining boundary conditions, loads and sliding interface data for meshes subsequent to h -adaptivity. With the structure of current programs such as DYNA2D, the problem definition and consequently the boundary conditions, loads, and other data are defined discretely and no higher level description of the problem is available which could provide the primitive data is available. This is a particular difficulty in prescribing boundary conditions to descendant nodes which are generated in the h -adaptive procedure, because the boundary conditions on the new nodes must be developed from simple rules. Although in most cases, simple rules such as the ones devised here are able to assign the right boundary conditions, the procedure has a degree of uncertainty which is not desirable in production codes. The usability of programs such as DYNA2D could be enhanced if the descriptive data could be input in a much more general fashion for it would eliminate a substantial burden for the users.

The procedure developed here is a useful model for introducing the h -adaptivity into other general purpose programs. The addition of adaptivity by directly modifying a program is usually extremely difficult because of the large varieties of functionalities

which characterize these programs. When internal extensive modifications are made, it is difficult not to disrupt these functionalities. On the other hand, restart and rezoning should be a feature of any useful nonlinear solid mechanics program. By utilizing these features as described here, h -adaptivity can be implemented in a reasonably straightforward manner, although the efficiency will not match that of a program developed completely with h -adaptivity.

SECTION 5 REFERENCES

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

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

Belytschko, T., and Tabbara, M., "H-Adaptive Finite Element Methods for Dynamic Problems, with Emphasis on Localization," (UNCLASSIFIED) *International Journal for Numerical Methods in Engineering*, Vol. 36, pp. 4245-4265, 1993. (UNCLASSIFIED)

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

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

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

Hallquist, J.O., "MAZE- An Input Generation for DYNA2D and NIKE2D," (UNCLASSIFIED) Lawrence Livermore National Laboratory, Livermore CA, UCID-19029, Rev. 2, June 1983. (UNCLASSIFIED)

Whirley, R.G., Engleman, B.E., and Hallquist, J.O., "DYNA2D: A Nonlinear, Explicit, Two-Dimensional Finite Element Code For Solid Mechanics, User Manual," (UNCLASSIFIED) Lawrence Livermore National Laboratory, Livermore CA, UCRL-MA-110630, April 1992. (UNCLASSIFIED)

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

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

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

DISTRIBUTION LIST

DNA-TR-95-92

DEPARTMENT OF DEFENSE

DEFENSE NUCLEAR AGENCY

2 CY ATTN: ISST
ATTN: WEL
ATTN: WELE LTC JIM HODGE

DEFENSE TECHNICAL INFORMATION CENTER

2 CY ATTN: DTIC/OCP

FIELD COMMAND DEFENSE NUCLEAR AGENCY

ATTN: FCTO
ATTN: FCTT DR BALADI

DEPARTMENT OF THE ARMY

ARMY RESEARCH LABORATORIES

ATTN: AMSRL-SL-CE

U S ARMY ENGR WATERWAYS EXPER STATION

ATTN: C WELCH CEWES-SD-R
ATTN: CEWES-SS-R DR BALSARA
ATTN: D RICKMAN CEWES-SE-R
ATTN: E JACKSON CEWES-SD-R
ATTN: F DALLRIVA CEWES-SS-R

DEPARTMENT OF DEFENSE CONTRACTORS

APTEK, INC

ATTN: T MEAGHER

BELYTSCHKO (THEODORE B)

2 CY ATTN: T BELYTSCHKO

GENERAL ATOMICS, INC

ATTN: CHARLES CHARMAN

H & H CONSULTANTS, INC

ATTN: W HALL

HORIZONS TECHNOLOGY, INC

ATTN: B LEE

KAMAN SCIENCES CORP

ATTN: VERN SMITH

KAMAN SCIENCES CORP

ATTN: D MOFFETT

KAMAN SCIENCES CORPORATION

ATTN: DASIAC

LOGICON R & D ASSOCIATES

ATTN: D SIMONS
ATTN: LIBRARY

LOGICON R & D ASSOCIATES

ATTN: G GANONG

LOGICON R & D ASSOCIATES

ATTN: J WEBSTER
ATTN: T MAZOLLA

SCIENCE APPLICATIONS INTL CORP

ATTN: J STODDARD

SCIENCE APPLICATIONS INTL CORP

ATTN: D BACON

SRI INTERNATIONAL

ATTN: M SANAI

TECH REPS, INC

ATTN: F MCMULLAN

TITAN CORPORATION (THE)

ATTN: P HOOKUM

WEIDLINGER ASSOC, INC

ATTN: DARREN TENNANT
ATTN: H LEVINE

WEIDLINGER ASSOCIATES, INC

ATTN: M BARON