Adaptivity in Crashworthiness Calculations

INTRODUCTION

The major cost in crashworthiness analysis arises from model preparation. In many cases, two to three man-months or more are required to prepare the finite element model for a crash analysis. Some of this burden arises because of the inability to use existing linear models as a basis for the crash model, for usually considerable refinement is needed where severe deformations are expected, whereas coarse meshes are used for the rest of the model to save computer time. Furthermore, different models have to be developed for each of the crash scenarios that must be considered, which include frontal crash, rear impact, side impact, and others.

A substantial part of this burden can be alleviated by using adaptive procedures wherein the computer program automatically refines the mesh where it is needed for accuracy. Adaptive procedures for linear stress analysis and computational fluid dynamics are already available commercially, and have found enthusiastic acceptance. Although more computationally demanding than programs without adaptivity, they can save substantial amounts of engineering time.

This article describes some of the methods and experiences with two adaptive finite element programs for nonlinear transient analysis: WHAMS, a research program developed at Northwestern University and SUPERWHAMS, a commercial program developed by KBS2. The latter is an outgrowth of WHAMS. $h$-Adaptivity and its implementation is described. We cursorily address the issues of subcycling and contract-impact, which are essential for an effective implementation of $h$-adaptivity. The last sections give some numerical studies and concluding remarks.

ADAPTIVITY

The $h$-adaptive methodology for explicit nonlinear structural dynamics used here was described in Belytschko, Wong, Plaskacz (1989). In the $h$-adaptive process, elements are subdivided into smaller elements where more accuracy is needed; this process is called fission. The elements involved in the fission process are subdi-
vided into elements with sides $h/2$, where $h$ is the length of the side of the original elements. This is illustrated in Fig. 1 for a quadrilateral element. In fission, each quadrilateral is subdivided into four quadrilaterals (as indicated in Fig. 1) by using the midpoints of the sides and the centroid of the element to generate four new quadrilaterals. The fission process for a triangular element is shown in Fig. 2 where the element is subdivided into four triangles by using the midpoints of the three sides.

The adaptive process can consist of several levels of fission. Figure 1 shows one subdivision, which is called one level of adaptivity. In subsequent steps, the fissioned elements can again be fissioned in a second level of adaptivity, and these elements can again, in turn, be fissioned in a third level of adaptivity (Fig. 3). The number of levels of adaptivity is unrestricted, although it has been limited to three in most computations so far. The adaptivity in a mesh is restricted by three rules:

1. The number of levels is unrestricted in the program, but a maximum level of adaptivity is set in each calculation, which has been limited to three in our calculations.
2. The levels of adaptivity implemented in a mesh must be such that the levels of adaptivity implemented in adjacent elements differ by, at most, one level.
3. The total number of elements cannot exceed the limit specified by the user. Once the specified maximum number of elements is reached, no further fission is permitted.

The second rule is used to enforce a one-irregular rule given by Devloo, Oden, and Strouboulis (1987), which restricts the number of elements along the side of any element in the mesh to two. The enforcement of this rule is necessary to accommodate limitations in the data structure.

In WHAMS, elements were also fused or combined when it was felt that they were no longer needed. It was found that the implementation of fusion procedures for general meshes, such as occur in typical applications of commercial programs, is too complex, so only fission is included in SUPERWHAMS.

The original mesh provided by the user is known as the parent mesh, the elements of this mesh are called the parent elements, and the nodes are called parent nodes. Any elements that are generated by the adaptive process are called descendant elements, and any nodes that are generated by the adaptive process are called descendant nodes. Elements generated by the first level of adaptivity are called first-generation elements, those generated by the second levels of adaptivity are called second-generation elements, etc.

The coordinates of the descendant nodes are generated by using the midpoint of a line between immediate parent nodes. In subsequent steps, nodes that are not corner nodes of all attached elements are treated as slave nodes. They are handled by the tieline algorithm, which enforces compatibility of this node and consistently transfers its nodal forces to the adjacent master nodes.

Several error criteria have been studied. The first error criterion is an energy-normalized interpolation error criterion. The underlying motivation is a standard result in interpolation theory in one-dimension that for linear interpolants

$$
\|u - u^h\| \leq \|u - u_i\| \leq Ch^2 \|u_{x\times}\|
$$

where $\|\|$ is the $L_2$ norm, $C$ is a constant, $h$ is the characteristic element length, $u$ is the function to be interpolated, and $u_i$ the interpolate. We use
the arguments of Diaz, Kikuchi, Papalambros, and Taylor (1983) that the right-hand side can be approximated sufficiently by an interpolant \( u^\ast \) generated by the nodal values of \( u \) to indicate the error. Thus, the norm in the error can be approximated over a subdomain by

\[
\| \text{error} \| = Ch^\gamma \| u^\ast, \alpha \| .
\] (2)

It can be shown for a plate in bending approximated by bilinear functions that the right-hand side in the above can be approximated by the change in angle between two elements. In order to obtain a normalized value, the change in angle is multiplied by the moment and divided by a suitable normalizing energy. The error indicator is then given by

\[
e_{\text{indic}} = \frac{\Delta \theta m_\theta}{\bar{W}}
\] (3)

where \( \Delta \theta \) is the change in angle across a side, \( m_\theta \) a corresponding moment, and \( \bar{W} \) the normalizing energy.

The other error indicators we used are of the Zienkiewicz-Zhu type. Lee (1993) has implemented an error criterion based on Zienkiewicz and Zhu’s superconvergent patch (1992a,b), while Yeh (1992) has used an error criterion based on a global \( L^2 \) projection.

A difficulty in both cases is that the components of the stresses and strains in contiguous elements do not refer to a common coordinate system. In corotational elements, the stress and strain components are referred to corotational coordinates that differ between adjacent elements; even in total Lagrangian formulations for shells, the local coordinates systems of adjacent shell elements differ.

Therefore, the error indicator was expressed in terms of the invariants of the strains

\[
e_{\text{indic}} = \int_{\Omega^t} \sum_{i=1}^{3} (I_i^e(x) - I_i^e(x)^\ast)^2 dA
\] (4)

where \( I_i^e \) are the strain invariants computed by the finite element procedure and \( I_i^e^\ast \) are the strain invariants computed by the projection. The error indicator is obtained by integrating with \( 2 \times 2 \) quadrature over the top and bottom surfaces of the shell element, which are indicated by \( \Omega^s_t \) and \( \Omega^s_b \), respectively. Integrating over the volume is substantially more expensive and does not improve results.

In Lee (1993), the projected invariants \( I_i^e^\ast \) were obtained by using a local patch of four elements using a least squares fit of a bilinear polynomial to the superconvergent points of the finite element solution. Thus

\[
I_i^e(x) = P(x)a_i
\] (5)

where

\[
P(x) = [1, x, y, xy]
\] (6)

and \( a_i \) are four parameters obtained by minimizing \( J \) given by

\[
J = \sum_{Q=1}^{4} \sum_{i=1}^{3} [I_i^e(x_Q) - I_i^e(x_Q)^\ast]^2
\] (7)

where \( x_Q \) are the superconvergent points of the elements, which correspond to the quadrature points in one-quadrature point elements. The coordinates \( x \) and \( y \) are measured on a projected plane by a stretching-mosaic technique. In Belytschko and Yeh (1992), a global \( L^2 \) projection was used to obtain \( I_i^e \). Global projections are not as accurate as local projections (Zienkiewicz and Zhu, 1992a,b), so their efficacy as error indicators diminishes.

A fourth error criterion is obtained by first fitting the displacements at the nodes by a polynomial. The strains can then be computed by the strain-displacement equations. An error criterion is given by

\[
e_{\text{indic}} = \int_{\Omega^t} \sum_{i=1}^{3} \| e^\ast(x) - e^h(x) \|^2 dA
\] (8)

where \( e^h \) is the finite element strains.

Note that in Eqs. (4) and (8), a total measure of strain must be used. In these programs, like in DYNA-3D, deformation is measured by the rate-of-deformation tensor, so to obtain a path independent measure of strain, the Green strain tensor is used for the error calculations.

The process of determining whether to fission any elements is called an assessment or judgment. Because the assessment process and associated remeshing is time consuming, it should not be done frequently in a run (maybe 5–20 times). In the assessment process, the error indicators are computed for all elements and sorted. Those elements that have error indicators that exceed the user–input tolerance are fissioned, provided that the total number of elements does not exceed the limit set on the number of elements.
Table I. Graphical Explanation of Go-Back Adaptive Algorithm

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Tw</th>
<th>TA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(×FA)</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

FA: Frequency of assessments
Tw: Time along the solid analysis path at which the temporary analysis results are saved, then used for the repeat
TA: Time along the dashed analysis path at which the adaptive assessment is done and, once a new mesh has been obtained, go back to Tw
—: Analysis path along which the results are the accepted results
—: Analysis path along which the results are only used for the adaptive assessment

It has been found through many numerical studies that to continue forward in time with the refined (or fissioned) mesh is often insufficient to achieve the benefits of adaptivity. The reason for this is that violation of the error criterion indicates that significant errors have occurred since the last assessment. Therefore, a procedure indicated in Table 1 is adopted. As can be seen from Table 1, whenever a judgment or assessment is made, the resulting finite element mesh is used to repeat the solution from the previous judgment. The output obtained by the coarser mesh is not retained; solutions that are not retained are indicated by dashed lines in Table 1. The final output stream of time histories then consists of the mesh-time path indicated by solid lines in Table 1.

SUBCYCLING AND CONTACT-IMPACT

When h-adaptivity is implemented with explicit time integration, mixed time integration (often called subcycling) is crucial for efficiency because the addition of smaller elements by fission reduces the stable time step (by a factor of 8 for three levels of fission). In WHAMS and SUPERWHAMS, mixed time integration as described in Belytschko and Liu (1982) is available. With this method, the time steps vary throughout the mesh so that each element is integrated with an optimal time step.

Contact-impact in an adaptive method is facilitated by a simple algorithm called the pinball method (Belytschko and Neal, 1991). The pinball algorithm embeds a sphere in each element, regardless of whether it is a shell or solid element, and then makes an interpenetration check by determining whether the pinballs have overlapped. This is a very simple check, because it only involves a comparison of the current distance between the pinballs with the sum of the radii. In the penalty form of the algorithm, whenever overlap of pinballs is detected, equal and opposite forces proportional to the magnitude of the interpenetration are then applied to the centers of the pinballs. These forces are then transferred to the nodes of the elements in which the pinball is embedded. Because the algorithm is very simple and identical regardless of what type of contact is involved, it is very amenable to vectorization and leads to very fast contact-impact algorithms. For example, the collapsing box beam problem that Zhong (1988) reported to take 64 min of CPU was solved in 16 min by the pinball algorithm. Only 15% of the total CPU time was expended by the pinball algorithm on a CRAY-YMP. In comparison to the Belytschko-Lin (1987) algorithm, the pinball algorithm was 5 times as fast on a vector compiler, 1.25 times as fast on a scalar compiler.

This algorithm was first developed to facilitate the vectorization of contact-impact algorithms for high velocity impact and penetration of solids in which erosion of elements occurs. It became apparent in subsequent work that the pinball method has a striking superiority: it is not necessary to distinguish different types of contact, because edge to surface and surface to solid contact can all be treated by the same algorithm (Belytschko and Yeh, 1993). Therefore the user can be spared from guessing the types of contact that may occur. Moreover, the applicability of a single algorithm to all contacts that occur in a model obviously facilitates vectorization.

NUMERICAL EXAMPLES

Several examples are given to demonstrate the performance of h-adaptivity. The formulations of the elements used here are given in Belytschko, Lin, and Tsay (1984) and Belytschko, Wong, and Chiang (1992). The materials used in the sample
Table II. Material Properties for Examples

<table>
<thead>
<tr>
<th>Material Properties</th>
<th>Box Beam</th>
<th>S-Beam</th>
<th>Cylindrical Panel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elastic modulus ((E))</td>
<td>(2.06 \times 10^{11} \text{ N/m}^2)</td>
<td>(3.00 \times 10^9 \text{ psi})</td>
<td>(1.05 \times 10^7 \text{ psi})</td>
</tr>
<tr>
<td>Plastic modulus ((E_p))</td>
<td>(6.30 \times 10^8 \text{ N/m}^2)</td>
<td>(0.00 \text{ psi})</td>
<td>(0.00 \text{ psi})</td>
</tr>
<tr>
<td>Yield stress ((\sigma_y))</td>
<td>(2.00 \times 10^6 \text{ N/m}^2)</td>
<td>(3.60 \times 10^4 \text{ psi})</td>
<td>(4.40 \times 10^4 \text{ psi})</td>
</tr>
<tr>
<td>Density ((\rho))</td>
<td>(7.84 \times 10^3 \text{ kg/m}^2)</td>
<td>(7.40 \times 10^{-4} \text{ lb.-s}^2/\text{in.}^4)</td>
<td>(2.50 \times 10^{-4} \text{ lb.-s}^2/\text{in.}^4)</td>
</tr>
<tr>
<td>Poisson’s ratio ((\nu))</td>
<td>(0.30)</td>
<td>(0.30)</td>
<td>(0.33)</td>
</tr>
</tbody>
</table>

analyses are elastic-plastic with isotropic strain hardening; their properties are listed in Table 2.

The first example problem is a thin-walled box beam that impacts a fixed rigid wall. Figure 4 gives the geometry for this problem. It was previously analyzed by Benson and Hallquist (1987), Zhong (1988), and Belytschko and Neal (1991). Because of symmetry, only one quarter of this beam is modeled. Note that contact can occur between any two elements. Although roundoff error will sometimes trigger buckling even in 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. An imperfection with different signs but the same magnitude on the sides and bottom as shown in Fig. 5 was used.

Figure 6 shows the rigid wall forces. Results from the fine nonadaptive meshes of 756 elements are used as our benchmark to show the improvement by the adaptive method. The improvement of accuracy and reduction of CPU time by the use of the adaptive method are also shown in Fig. 6, from which we can see a 40% reduction of CPU time, yet the adaptive result still coincides with the fine mesh result. Figure 7 shows the evolution of the adaptive mesh with time.

Figure 8 shows an original coarse mesh and adaptive results for an S-beam that strikes a rigid wall. Two levels of adaptivity were used. Note that refinement takes place at the corners where local buckling occurs. This local refinement substantially improves the accuracy of the solution.

The next example is a cylindrical panel that is loaded impulsively over a portion of the top surface; details of this problem are given in Kennedy, Belytschko, and Lin (1986). Experimental results have been reported by Balmer and Witmer (1964). Figure 9 shows the undeformed and deformed configurations at selected times and the deployment of elements by \(h\)-adaptivity; two
levels of subdivision were used along with the error criterion of Lee (1993).

Table 3 gives the maximum displacements for several nonadaptive meshes and adaptive meshes. The experimental results are also given. Note the improvement in accuracy for $h$-adaptivity. Experimental results are not an ideal benchmark for assessing adaptivity because discrepancies between experimental and numerical results may arise from inadequacies in the material law, such as neglecting strain-rate effects, and uncertainties in material properties, loading, and boundary conditions. Therefore, results obtained on a massively parallel computer, the CM-2, are included as the benchmark. Note that the adaptive results converge much faster to the CM-2 benchmark than fixed meshes. It is also interesting that meshes with the resolution often used in crashworthiness ($8 \times 16$) exhibit large error. These errors can be explained by the inability of the coarse mesh to resolve the elastic-plastic hinge lines.

We next discuss several crashworthiness calculations. Table 4 shows the timings with and without subcycling. As can be seen, the running time decreases by a factor of more than 2 with subcycling. Figure 10 shows the front end of a full model of a Ford van used for a frontal crash simulation with SUPERWHAMS. A modified pinball method was used for contact-impact; note that it effectively prevents interpenetration. A hypothetical rear end simulation (the rear end model does not contain the engineering detail required for a realistic analysis) was used for a two car collision. The model required about 90,000 elements. Running time increased by about a factor of 2 for the adaptive calculation as compared to the coarse mesh solution. However, this mesh could subsequently be used for similar rear end simulations at far less cost.

Figure 11 shows a portion of a side rail taken from a full car model solved with an adaptive mesh with one level of adaptivity; the deformed original coarse mesh is also shown. Note the significantly larger deformations of the $h$-adaptive mesh, which is also reflected in increased overall crush and lower acceleration levels.
Table 5 gives the breakdown of the CPU time required by various parts of the SUPERWHAMS program in a crashworthiness analysis. With the pinball algorithm, only 15% of running time is attributable to the contact-impact algorithm. In some explicit codes, more than 60% is often required for the contact-impact algorithm in crash analysis. Thus the pinball algorithm saves substantial time in the calculation of contact-impact.
CONCLUSIONS

Adaptive finite element techniques that are applicable to crashworthiness analysis have been described. Such analyses have a substantial potential for reducing model data preparation time and improving the quality of results.

The state of crash simulation today still requires considerable model tuning in order to match test results. Some of this tuning is probably due to insufficient resolution in the models. As illustrated by the simple cylindrical panel problem, very fine meshes are needed where plastic hinge lines occur to achieve good accuracy. The resolution of current finite element models for crashworthiness does not approach this level, and it is not possible with uniform meshes because of excessive computational costs. Limitations on sizes of models do not arise only from excessive computer time; computer memory requirements are also an important barrier. Adaptive methods have the potential for overcoming this accuracy barrier; by concentrating refinement where it is needed, high resolution and accuracy can be achieved with reasonable computer resources.

Multi-time step integration, often called subcycling, is an essential element for efficient h-adaptivity in explicit programs. Without subcycling, the local refinement of the mesh will re-
Adaptivity in Crashworthiness Calculations

\[ \text{time} = 0.000 \times 10^0 \]

\[ \text{time} = 4.001 \times 10^1 \text{ ms} \]

FIGURE 10 Original and deformed shapes of a vehicle front end with pinball contact-impact algorithm.

duce the time step over the entire mesh, which negates much of the advantage of adaptivity. Even without adaptivity, subcycling can provide savings of a factor of two in crash simulations.

We gratefully acknowledge the support of the U.S. Office of Naval Research and the National Center for Supercomputer Applications at the University of Illinois for computer time.

REFERENCES


Belytschko, T., Wong, B. L., and Chiang, H. Y., 1992, "Advances in One-Point Quadrature Shell


Submit your manuscripts at http://www.hindawi.com