STRUCTURAL DYNAMICS IN LMFBR CONTAINMENT ANALYSIS A BRIEF SURVEY OF COMPUTATIONAL METHODS AND CODES

Y. W. CHANG

Reactor Analysis and Safety Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, Illinois 60439, U.S.A.

SUMMARY

In recent years, the use of computer codes to study the response of primary containment of large, liquid-metal fast breeder reactors (LMFBR) under postulated accident conditions has been adopted by most fast reactor projects. Since the first introduction of REXCO-H containment code in 1969, a number of containment codes have evolved and reported in the literature. REXCO-HEP, ICECO, MICE, ASTARTE, SORBOUM, ARES, SEURBUNK, EURDYN, COQREV, and CEFRA are a few of them developed for LMFBR containments. Other computer codes developed for different purposes, such as PISCES, HEMP, ANSYS, TOODY, WHAM, STEALTH, CSQ, DYPLAS, and TRICO have also been applied to safety problems related to LMFBR containments.

However, before a computer code can be applied with confidence to safety analysis, it is necessary that the code has been validated by comparison with analytical solutions and experimental data. In the LMFBR containment analysis, because of the complexity of the problem, no appropriate analytical solutions are available, heavy reliance has to be placed on comparisons with experimental results. This validation process is very costly and time-consuming.

Recent developments in numerical analysis have added another aspect to the analysis of LMFBR containments. The number of numerical methods available for structural dynamics has increased to the point that the selection of a numerical method for the computer analysis has become just as difficult as obtaining a closed-form solution for HCDA analysis.

This paper gives a brief survey of the computational methods and codes available for LMFBR containment analysis. The various numerical methods commonly used in the computer codes are compared. It provides the reactor engineers to up-to-date information on the development of structural dynamics in LMFBR containment analysis. It can also be used as a basis for the selection of the numerical method in the future code development.

First, the commonly used finite-difference expressions in the Lagrangian codes will be compared. Sample calculations will be used as a basis for discussing and comparing the accuracy of the various finite-difference representations. The distortion of the meshes will also be compared; the techniques used for eliminating the numerical instabilities will be discussed and compared using examples. Next, the numerical methods used in the Eulerian formulation will be compared, first among themselves and then with the Lagrangian formulations. Special emphasis is placed on the effect of mass diffusion of the Eulerian calculation on the propagation of discontinuities. Implicit and explicit numerical integrations will be discussed and results obtained from these two techniques will be compared. Then, the finite-element methods are compared with the finite-difference methods. The advantages and disadvantages of the two methods will be discussed in detail, together with the versatility and ease of application of the method to containment analysis having complex geometries. It will also be shown that the finite-element equations for a constant-pressure fluid element is identical to the finite-difference equations using contour integrations. Finally, conclusions based on this study will be given.

1. Introduction

In recent years, the use of computer codes to study the response of primary containment of large, liquid-metal fast breeder reactors (LMFBR) under postulated accident conditions has been adopted by most fast reactor projects. Since the first introduction of REXCO-H containment code in 1969 [1], a number of containment codes have evolved and been reported in the literature.

However, the arithmetic operations in the computer is both discrete and finite. It is necessary in the computer code to formulate the equations of interest in a discrete form, and to replace the physical system by a discretized system. How closely the computer solution approximates the true solution of the system is a concern of the safety analysis. If the computer codes are to be used with confidence, they must first be validated by comparison with analytical solutions and experimental data. However, for reactor containment analysis, because of the complexity of the problem, no appropriate analytical solutions are available. Thus, heavy reliance had to be placed upon comparison with experimental results obtained from experiments having actual reactor geometries. This validation process is very costly and time-consuming.

Recent developments in numerical analysis have added another aspect to the analysis of the reactor primary containment using computer programs. The number of numerical methods available for containment analysis has increased to the point that the selection of a numerical method for the computer analysis has become just as difficult as obtaining a closed-form solution for HCDA analysis.

This paper briefly summarizes the various numerical methods commonly used in containment analysis in computer programs. They will be compared on the basis of truncation errors resulting in the numerical approximation, the method of integration, the resolution of the computed results, and the ease of programming in computer codes.

The aim of this paper is to provide enough information to an analyst so that he can suitably define his choice of method, and hence his choice of programs. It is hoped that the information provided in this paper can be used as a basis for the selection of a numerical method used in the future containment code development. Even though a code validation may still be imposed upon by the reactor licensing and regulatory authorities, the analyst can have more confidence in the ability of the computer program that he developed and in the accuracy of the numerical results which he obtained in the computer analysis.

2. Discrete Representation of a Physical System

As mentioned earlier, the arithmetic operation in the computer is both discrete and finite. It is necessary in the computer analysis that a physical system be replaced by a discretized system. In discretizing a continuum system, the continuum is replaced by a group of meshes. Two types of meshes are commonly used. If the mesh point is attached to a material particle and moves with the material, the mesh is called Lagrangian. If the mesh point is fixed in space and is invariant in time, the mesh is called Eulerian.

As the Lagrangian meshes move with the material particle the material interfaces, if they initially coincide with the Lagrangian mesh lines, will remain at the mesh lines. Therefore no complex procedures are needed at material interfaces and at the boundaries. Another important feature in the Lagrangian discretization is that the continuum system is always approximated by the same number of meshes. Thus, the initial accuracy of the approximation is, in general, maintained throughout the calculation. Furthermore, the number of meshes

needed to achieve a certain degree of accuracy is surprisingly small compared with other formulations. Lagrangian calculations have also proven to be very accurate as long as the meshes remain regular and undistorted. Therefore, Lagrangian calculations are ideal for solving problems where the meshes will not be badly distorted. Because of the ease of programming in material interfaces, computer codes developed for containment analysis are almost invariably utilizing the Lagrangian discretization.

Although Lagrangian codes have been successfully used for analyzing the pressure wave propagations, coolant slug impacts, and containment responses under HCDAs, their analysis are still limited to the early stage of the excursion, because the excessive zone distortion will deteriorate the accuracy of the numerical results. Although rezoning of the distorted Lagrangian mesh can extend the calculations to a longer time, there is an upper limit on how far one can push this rezoning technique in the Lagrangian calculation. If a rezoning is done at every few time steps, it means, essentially, that the fixed mesh of a Eulerian discretization is more appropriate.

As the Eulerian meshes are fixed in space, they do not move with the fluid in the numerical calculation. Therefore, the Eulerian equations of hydrodynamics are ideal for treating excessive fluid distortions. Thus, for problems involving large material distortions, Eulerian discretization becomes more attractive. Especially, in the calculation of sodium spillage and core gas-bubble migration, one has to use the Eulerian codes to avoid excessive mesh distortions.

Unfortunately, the very feature of the Eulerian method which makes it useful in handling material distortions is also the one which makes it unsatisfactory for performing calculations having material interfaces and moving boundaries. In reactor containments, the reactor vessel, cover head, and internal structures are not rigidly fixed in space. As they displace under the applied pressure loads, their boundaries, even if they initially coincided with an Eulerian mesh, will no longer coinside with the Eulerian lines. Complex procedures are needed at the material interfaces and boundaries for treating the arbitrarily shaped Eulerian meshes and for identifying the materials at the interfaces. Because of this, the Eulerian computer programs developed to date still cannot treat a reactor in an HCDA analysis to have complex material interfaces and configurations. Thus, the Eulerian computer programs developed to date still cannot treat a reactor configuration in the HCDA analysis to have the same complexity as in the Lagrangian analysis. As the mass convection terms in the Eulerian method will introduce a diffusion effect on the propagation of discontinuities, the Eulerian codes can be expected to be somewhat more dispersive than the Lagrangian codes. Because of the complex procedure in the treatment of the interfaces and boundaries, the Eulerian codes are more difficult to program compared to Lagrangian codes. Usually, the running time of a Eulerian code is also somewhat longer than that of a Lagrangian code.

It should become apparent that the two types of representation of a physical system, Lagrangian and Eulerian, have both merits and drawbacks. In order to be able to analyze the entire sequence of events which occurred in an HCDA, one should use both the Lagrangian and Eulerian computer programs: one for the determination of the pressure wave propagation, the response of the reactor internals and the deformation of reactor vessel wall during the phase immediately following the accident, and the other to study the excursion phenomena such as slug impact, sodium spillage and core gas-bubble expansion and migration during the

latter phase of the excursion. Other approaches such as coupled Lagrangian Eulerian method [2] and arbitrary Lagrangian and Eulerian method [3] have been suggested as an alternate for the Lagrangian or Eulerian method. Basically, these methods perform two types of calculations, one in Lagrangian and the other in Eulerian and they provide the capability of automatic continuous rezoning. Thus, meshes may be moved with the fluid in the usual Lagrangian manner or be fixed in space in the Eulerian manner or move in some arbitrary way. Therefore, the analyst can have a choice in the representation of the physical system, so that a part of the continuum system can be discretized into Lagrangian meshes and the other part of the continuum into Eulerian meshes.

3. Computational Methods in Hydrodynamics

3.1 Finite-difference Method

The most widely used numerical method in the computer programs in containment analysis for shock and wave propagation in fluids is the finite-difference method. In discretizing a continuum system and in performing numerical calculations, one needs a time-mesh and a space-mesh. Unlike the case for space variables, the creation of a time-mesh can be achieved by an integration procedure. Therefore, the derivatives in time and space can be treated differently. Here, only the finite-difference expressions of the spatial variables in two dimensions (r,z) are discussed; the differencing of the time variable will be discussed in Section 5 in the time integration procedure.

3.1.1 Lagrangian Hydrodynamics

In Lagrangian hydrodynamics, the spatial derivatives that need to be evaluated are the pressure gradients $(\frac{\partial p}{\partial r}, \frac{\partial p}{\partial z})$ in the momentum equation. The commonly used finite-difference expressions for the spatial derivatives can be grouped into two families: (1) Taylor's expansion; (2) Line integration.

Taylor's Expansion

Consider a physical system of cylindrical symmetry, which is represented by quadrilateral Lagrangian meshes as shown in Fig. 1. If the pressure at points 1,2,3 and 4 is known, the pressure gradients at point 0 can be evaluated by applying Taylor's expansion between points 0 and 1, 0 and 2, 0 and 3, and 0 and 4. This yields four equations of the form

$$p_1=p_o + (r_1-r_o)\frac{\partial p}{\partial r} + (z_1-z_o)\frac{\partial p}{\partial z} + \tfrac{1}{2}(r_1-r_o)^2\frac{\partial^2 p}{\partial r^2} + (r_1-r_o)(z_1-z_o)\frac{\partial^2 p}{\partial r \partial z}$$

$$+ \tfrac{1}{2}(z_1-z_o)^2\frac{\partial^2 p}{\partial z^2} + \ldots .$$

However, the system is overdetermined for $\frac{\partial p}{\partial r}$ and $\frac{\partial p}{\partial z}$. Kolsky [4] proposed to solve first for (p_1-p_3) and (p_2-p_4), and then for $\frac{\partial p}{\partial r}$ and $\frac{\partial p}{\partial z}$. The pressure gradients solved in this manner are of the form

$$\frac{\partial p}{\partial r} = -\frac{1}{2A_1}\left[(p_1-p_3)(z_2-z_4) - (p_2-p_4)(z_1-z_3)\right]$$
$$\frac{\partial p}{\partial z} = \frac{1}{2A_1}\left[(p_1-p_3)(r_2-r_4) - (p_2-p_4)(r_1-r_3)\right] \qquad (1)$$

where

$$A_1 = \tfrac{1}{2}\left[(z_1-z_3)(r_2-r_4) - (z_2-z_4)(r_1-r_3)\right]$$

represents the area of the quadrilateral 1234. The second- and higher-order terms in the mesh size, i.e., (z_1-z_3), (r_1-r_3), etc., are omitted. Amurud and Orr [5] proposed to apply the Taylor's expansion between points 0 and 5, 0 and 6, 0 and 7, and 0 and 8 (shown in Fig. 1) to avoid the reversal of signs of the accelerations occurring in the distorted meshes. By doing so, they obtained the pressure gradients of the form

$$\frac{\partial p}{\partial r} = -\frac{1}{2A_2}\left[(p_5-p_7)(z_6-z_8) - (p_6-p_8)(z_5-z_7)\right]$$
$$\frac{\partial p}{\partial z} = \frac{1}{2A_2}\left[(p_5-p_7)(r_6-r_8) - (p_6-p_8)(r_5-r_7)\right] \qquad (2)$$

where $2A_2$ is the area of the quadrilateral 5678. This method is often called "midpoint" method and was used in REXCO-H, REXCO-HEP [6], and CEFRA [7] codes.

Line Integration

Green's theorem in two dimensions states

$$\oint_C p\, n_1 ds = \int_A p_{,1} dA$$

where n_1 is the outward normal vector, C is a piecewise smooth curve, and A is the closed region bounded by C. In r-z coordinates, one obtains

$$\oint_C p\,dz = \int_A \frac{\partial p}{\partial r} dA = \left(\frac{\partial p}{\partial r}\right)_{ave} A$$

$$\oint_C p\,dr = -\int_A \frac{\partial p}{\partial z} dA = -\left(\frac{\partial p}{\partial z}\right)_{ave} A$$

Taking the line integration along the path ABCD in Fig. 1 and considering the pressure on line AB to be the pressure at point 2, etc., the pressure gradients at 0 are

$$\frac{\partial p}{\partial r} = -\frac{1}{2A_3}\left[p_1(z_A-Z_D) + p_2(z_B-z_A) + p_3(z_C-z_B) + p_4(z_D-z_C)\right]$$
$$\frac{\partial p}{\partial z} = \frac{1}{2A_3}\left[p_1(r_A-r_D) + p_2(r_B-r_A) + p_3(r_C-r_B) + p_4(r_D-r_C)\right] \qquad (3)$$

where $2A_3$ is the area of the quadrilateral ABCD. This difference scheme was used in PISCES, HEMP, TOODY, and STEALTH codes [8-11]. An alternate way to perform the line integration is to take four paths on lines A2O1, 2B3O, 3C4O, and 4D1O and to take the average value of the four paths as the pressure gradients at point 0. This method was used in the ASTARTE code [12].

As can be seen from Eqs. (2) and (3) the two methods, midpoint and line integration, which are commonly used in the containment computer codes, are equivalent to Taylor series expansions with terms of first order in the mesh distortion. As can be seen from the above derivations, the second order terms in pressure gradients contain the terms $(\frac{1}{2}(r_1+r_3)-r_0)$, etc., which are measures of the asymmetry of the mesh. For rectangular equally spaced meshes, the measure of the asymmetry becomes zero. Therefore, for small mesh distortions, the numerical expressions used in the midpoint and line integration methods should be of second-order accuracy. The results obtained by the two methods are in close agreement. For distorted meshes, there will be some differences in the two methods. This has been studied in detail

by Herrmann [13]. He showed the growth of the error of the spatial difference expressions as a function of the mesh deformation. For the methods he studied, the truncation errors all grew to very large proportions as the distortion became large. When one mesh area became negative, all Lagrangian methods gave unsatisfactory results. He concluded that a choice of the finite-difference expression may be made on the basis of computational convenience, which requires the least number of arithmetic steps for solution.

3.1.2 Eulerian Hydrodynamics

In Eulerian hydrodynamics, the governing equations are usually expressed in the conservative form; the physical system is represented by a set of rectangular meshes fixed in space. The density, pressure, and internal energy, are defined at the center of the mesh at i,j point, whereas, the velocities are defined at mesh boundaries with u at $i \pm \frac{1}{2}$ points, and v at $j \pm \frac{1}{2}$ points as shown in Fig. 2. One of the difficulties in the finite-difference of the Eulerian hydrodynamic equations is in the expression of the convective fluxes. Take, for example, the momentum flux term $(\rho u^2)_i$, where the density is defined at the center of the mesh where the flux is to be evaluated, but the velocities are defined at $i \pm \frac{1}{2}$. Therefore, some type of interpolation is needed. The commonly used convective flux expressions are:

Centered: $(\rho u^2)_i \rightarrow \rho_i (u_{i-\frac{1}{2}} + u_{i+\frac{1}{2}})^2/4$

Zip: $(\rho u^2)_i \rightarrow \rho_i u_{i-\frac{1}{2}} u_{i+\frac{1}{2}}$

Donor:
$$(\rho u^2)_i \rightarrow \begin{cases} \rho_i u^2_{i-\frac{1}{2}} \text{ for } (u_{i-\frac{1}{2}} + u_{i+\frac{1}{2}}) > 0 \\ \rho_i u^2_{i+\frac{1}{2}} \text{ for } (u_{i-\frac{1}{2}} + u_{i+\frac{1}{2}}) < 0 \end{cases} \qquad (4)$$

The properties of these expressions have been discussed by Harlow and Amsden [14]. Some of their comments are given below.

The centered flux form is one order more accurate in δx than the donor cell flux. However, this advantage is often weakened by the tendency to numerical instability which is produced by not time centering the convective flux. It should be mentioned that this tendency to instability can usually be mitigated by the addition of a counterbalancing diffusive flux (either real or artificial). This form was used in MAC [15] and SORBOUM [16] codes.

The ZIP form is also one order more accurate in δx than the donor cell flux. In addition, it has the advantages of eliminating a nonlinear contribution to instability. For these reasons, the ZIP flux is often preferred for the momentum convection. It was used in ICECO, MICE, and ICEPEL codes [17-19]. It should be mentioned that the ZIP flux also needs a counterbalancing diffusive flux when it is not centered in time.

Since the donor cell flux is not space centered, it has low order in δx. The larger truncation errors introduced in the donor cell method contribute a positive diffusive effect and, therefore, tend to stabilize the numerical calculations. The magnitude, however, can be excessive. Thus, it may produce erroneous results. Accordingly, the donor cell flux representation is not being used by most analysts except for very special circumstances. In LMFBR containment analysis, the circumstance in which a donor-cell flux appears to be desirable is in the early stages of excursions where violent initial conditions exist or at the time of slug impact where again violent conditions may exist.

3.2 Finite-element Method

Finite-element formulations for fluid were first developed by Oden and Somogyi [20], who first discretized the Navier-Stokes equations for viscous flow in Lagrangian formulations. The derivation of these equations are quite similar to those standard large-deformation formulations in solid mechanics by the finite-element method. It can be shown that the discrete form of the equation of motion of the finite-element method for a constant pressure fluid element is equivalent to that of the finite-difference method using line integration technique. This has been demonstrated by Belytschko et al. [21] in 2-D for a quadraliteral element, and by Chang and Gvildys [22] in 3-D for a hexahedron. Thus, there appear to be no basic mathematical differences between the two methods. Therefore, results obtained by the two methods should have the same degree of accuracy in the numerical computations.

One of the major differences between the finite-element and finite-difference methods is in the formulation of the equations of motion. In finite-difference methods the equations of motion are expressed directly in terms of the pressure gradients of the neighboring meshes whereas in the finite-element method they are formulated through the use of the intermediate nodal forces. The advantages of this aspect of the finite-element method are substantial when solving problems having different types of meshes connected together. This is because the nodal forces are completely defined within each mesh. Therefore, they do not depend on the shape of the neighboring mesh. However, this is not the case in the finite-different method. Although different types of meshes connected arbitrarily impose no difficulty in the analysis, the finite-difference equations must be formulated separately. This becomes rather troublesome in problems having complex geometries.

Another major difference between the finite-element and finite-difference methods is in the numbering of the meshes. In finite-difference programs, the meshes are not numbered. They rely on a regularity in the mesh to implicitly establish the connectivity information. The I-J system used in finite-difference methods requires each portion of the mesh in the physical space to be such that it can be mapped onto a rectangular grid in the computational space. Therefore, it cannot be used for problems having complex geometries. However, this is different in finite-element programs where the connectivity of the mesh is explicity stored. Therefore, the finite-element method can be used for problems involving complex geometrical shapes.

It should be noted that the above shortcomings of the finite-difference programs are not inherent in the finite-difference method. They can be easily removed in the finite-difference programs. For example, the finite-difference technique used in YAQUI code [23] is similar to the nodal force technique in the finite-element method. Likewise, the connectivity information can be easily stored in the finite-difference programs. It is important to note that the versatility of a computer code is often achieved at the expense of large computer storage and CPU time.

4. Computational Methods in Structural Dynamics

The structural members such as core barrel, core-support structure, and reactor vessel in a reactor containment, are made of shells and plates. If continuum theories are used for these structural members, the thickness of the structure will have to be divided up into at least three to five meshes, so that the bending strength of the structure member can be properly included in the analysis. This not only brings the requirement for a large number of meshes, but also limits the time steps to very small values if an explicit integration

scheme is used. Therefore, it is generally preferable to use a bending theory for the structural members. Here, we shall further define that the structural members are to be slender, solid members in which the deformation is governed by the deformation of the midplane and for which the assumptions such as those of Kirchhoff or Reissner regarding to the motion of the structure relative to the midplane are applied.

In the analysis of the structural members, Lagrangian meshes have been used almost exclusively. They can be analyzed by either a finite-difference or a finite-element method.

Consider a shell of revolution undergoing large deflections (shown in Fig. 3). The equations of equilibrium are:

$$\begin{aligned}
&\frac{\partial}{\partial s}\,[N_\phi r\cos\phi] - \frac{\partial}{\partial s}\,[Q_\phi r\sin\phi] - N_\theta + pr\sin\phi - mr\ddot{r} = 0\\
&\frac{\partial}{\partial s}\,[N_\phi r\sin\phi] + \frac{\partial}{\partial s}\,[Q_\phi r\cos\phi] - pr\cos\phi - mr\ddot{z} = 0\\
&\frac{\partial}{\partial s}\,[M_\phi r] - M_\theta\cos\phi - Q_\phi r = 0
\end{aligned} \tag{5}$$

where s is the length of the shell along the meridian, measured from the vertex of the shell, m the mass of the shell per unit area, ϕ the angle of inclination of the element with respect to the r-direction, p the pressure, N_ϕ and N_θ two tangential forces, Q_ϕ the transverse force, and M_ϕ and M_θ the two bending moments.

Using the notations in Fig. 4., the spatial derivatives of N_ϕ, Q_ϕ and M_ϕ in finite-difference representation can be easily obtained. For example, $\frac{\partial}{\partial s}\,[N_\phi r\cos\phi]$ becomes

$$\left[N_{\phi_{LT}} r_{LT+\frac{1}{2}} \cos\phi_{LT} - N_{\phi_{LT-1}} r_{LT-\frac{1}{2}} \cos\phi_{LT-1}\right] \Big/ \tfrac{1}{2}\,(S_{LT}+S_{LT-1})$$

etc.

This formulation was used in the REXCO-HEP code. Detail expressions of the finite difference equations and the treatment of material nonlinearity are given in Ref [6].

In finite element method, the equation of motion and element relations are again developed in terms of nodal forces and nodal displacement. Several alternative theories are available for treating the fluxural element. We will here sketch the formulation given by Belytschko and Hsieh [24]. which is very efficient in computer calculations. Their formulation is applicable to arbitrarily large rotations and material nonlinearities, but restricts the variation of rotation within an element to be small. Because of the use of a convected coordinate system, the strains can be linearly related to the displacement of the element (relative to the convected coordinates), and similarly the nodal force can be linearly related to the element stresses.

The convected $\hat{x}$ axis is taken to be the line joining the nodes 1 and 2. Cubic polynomial shape functions in $\hat{x}$ are used for the transverse displacements, linear shape functions for the axial displacements. The strain-displacement equations are

$$\begin{aligned}
\hat{\epsilon}_x &= \hat{\epsilon}_m - \hat{y}\,\frac{\partial\hat{\phi}(\hat{x})}{\partial\hat{x}}\\
\epsilon_\theta &= \frac{1}{r}\,(u_r - \hat{y}\,\cos\beta\,\frac{\partial u_y}{\partial x})
\end{aligned} \tag{6}$$

where $\hat{\varepsilon}_m = \dfrac{\partial \hat{u}_x^{def}}{\partial \hat{x}}$

$$\hat{\phi}(\hat{x}) = \frac{\hat{\phi}_1}{\ell^2}(\ell^2 - 4\ell\hat{x} + 3\hat{x}^2) + \frac{\hat{\phi}_2}{\ell^2}(3\hat{x}^2 - 2\ell\hat{x}) \tag{7}$$

in which $\hat{\phi}_1 = \phi_1 - \alpha$, subscript i equals 1 or 2, and α is the rigid body rotation.

The equations for internal nodal forces in the planar direction are

$$\begin{Bmatrix} m_1 \\ m_2 \\ \hat{f}_{2x} \end{Bmatrix} = -\frac{1}{\ell^2}\int_V \begin{bmatrix} (6\hat{x}-2\ell)\hat{y} \\ (6\hat{x}-2\ell)\hat{y} \\ \ell \end{bmatrix} \hat{\sigma}_x \, dV \tag{8}$$

The other nodal force are found by invoking the self-equilibration of the planar nodal forces

$$\hat{f}_{1y} = \hat{f}_{2y} = (m_1 + m_2)/\ell$$
$$\hat{f}_{1x} = \hat{f}_{2x} \tag{9}$$

The internal nodal forces due to the circumferential stresses are computed by the standard nodal force-stress relations. These nodal force are then transformed into the global coordinates to form the governing equations of the system. Detail derivations are given in [24]. This method was used in WHAM [25], STRAW [26] and EURDYN [27].

Although there are substantial differences in the mathematical treatments, the results obtained with the finite-difference and finite-element methods are comparable. The advantage of the finite-element method is in the use of shape functions which enable the variables to be defined not only at nodal points but also at any point within the element, whereas in the finite-difference method, the variables are defined only at the mesh points. Therefore, for structural members having strong geometric discontinuities, finite-element method appears to be more attractive.

5. Time Integration

As mentioned earlier, in performing numerical calculations, one needs both a time-mesh and a space-mesh, and the creation of a time-mesh can be achieved by an integration procedure. Thus, the displacements can be obtained from the velocities, and the velocities from the accelerations, through direct integration, while the accelerations are found from the equations of motion. If the acceleration at time step t^{n+1} is obtained from the values of pressures, densities, velocities, etc., at t^n, the integration method is explicit; if the acceleration at t^{n+1} is obtained from the values at t^{n+1}, the integration method is implicit. Similarly, if the displacements at t^{n+1} are obtained from velocities at t^n the integration is explicit; if the displacements at t^{n+1} are obtained from velocities at t^{n+1}, the integration is implicit. In general, implicit methods permit larger time steps but require the solution of a set of algebraic equations which are difference equations of the equations of motion combined with equations of mass, energy, and state, whereas explicit methods do not require the solution of a set of equations but are restricted to small time steps by numerical stability requirements.

In the explicit integration, the most commonly used integration scheme is the central difference method in which the displacements $\underline{u}$ are integrated on a time-mesh between the time

points t^n and t^{n+1} over the interval Δt from the velocities $\underline{\dot{u}}^{n+\frac{1}{2}}$ by

$$\underline{u}^{n+1} = \underline{u}^n + \underline{\dot{u}}^{n+\frac{1}{2}} \Delta t \tag{10}$$

and the velocities from the accelerations by

$$\underline{\dot{u}}^{n+\frac{1}{2}} = \underline{\dot{u}}^{n-\frac{1}{2}} + \underline{\ddot{u}}^n \Delta t \tag{11}$$

The most recently developed implicit integration scheme in Eulerian hydrodynamics is the ICE technique [14], in which the advanced-time densities and velocities are eliminated from the mass equation by the use of momentum equations and equation of state. Thus the mass equations in finite-difference form become a set of Poisson equations in the advance-time pressures that can be solved by matrix inversion or iterations. This method is extremely versatile and can be used for calculations in one, two, or three space dimensions. It was used in ICECO and ICEPEL codes. The commonly used implicit integration schemes in structural dynamics are the Newmark β-method, Wilson θ-method and Houbolt method. Among them, the most popular scheme is the Newmark β-method with $\beta=\frac{1}{4}$, which is equivalent to the trapezoidal method.

In the explicit integration scheme, the computation time depends principally on the number of meshes or elements and varies linearly with this number regardless of the size of the meshes or elements. In the implicit integration scheme, if the matrix inversion is used, the computational time of a particular problem then depends on the bandwidths of the matrix, which no longer varies linearly with the number of meshes or elements. Belytschko has compared the computational requirements of the two integration schemes, explicit verus implicit using matrix inversion for several problems [28]. For small bandwidth matrix, such as 100 node cylindrical shell elements, the two methods are comparable. For large bandwidth problems such as 31x62 node two-dimensional plane meshes, the implicit method requires 150 times as many computations per time step as the explicit method. It should be noted that the comparison of the two methods given above does not take into account the advantages of the larger time steps which are permitted in the implicit method.

If iteration is used in the solution of the set of algebraic equations in implicit schemes, the number of iteration cycles is, in general, independent of the bandwidth of the matrix. In other words, the computation time depends only on the number of meshes or elements and the convergence criterion of the iteration, regardless of the size of the meshes. However, the number of iteration cycles does depend on the magnitude of the time step used in the calculations. For large time steps, the number of iteration cycles increases rather rapidly. Results of a study performed on a simple pipe problem are given below to illustrate the relations between mesh size, time step and number of iteration cycles

15 cm-mesh					
Time step, μs	12.5	25	50	100	200
No. of Iterations	9	14	30	86	300
7.5 cm-mesh					
Time step, μs	6.25	12.5	25	50	
No. of Iterations	4	15	31	90	

6. Computer Programs

A number of containment codes have been developed since the introduction of REXCO-H in 1969. REXCO-HEP, ICECO, MICE, ICEPEL, ASTARTE, SORBOUM, ARES, SEURBUNK, EURDYN, COQREV, and CEFRA are a few of them developed for LMFBR containments. Other computer codes which were developed for different purposes, such as PISCES, HEMP, ANSYS, TOODY, WHAM, STEALTH, CSQ, DPYLAS, STRAW, SADCAT, and TRICO have also been applied to safety problems related to LMFBR containments. A complet review of the computer programs is beyond the scope of this paper. Here a brief summary is given in Table I. It should be mentioned that these codes were chosen primarily on the basis of their availability and the fact that they have been used for the analysis of the LMFBRs. It is by no means a complete list of the computer programs available in LMFBR containment analysis.

7. Acknowledgments

This work was performed in the Engineering Mechanics Section of the Reactor Analysis and Safety Division at Argonne National Laboratory, under the auspices of the U. S. Energy Research and Development Administration.

References

[1] Chang, Y. W., Gvidlys, J., and Fistedis, S. H., Two-dimensional Hydrodynamics Analysis for Primary Containment, ANL-7498 (November, 1969).

[2] Noh, W. F., "CEL: A Time-Dependent, Two-space-dimensional, Coupled Eulerian-Lagrangian Code," Methods in Computational Physics 3, pp. 117-179 (1964).

[3] Hirt, C. W. and Amsden, A. A., "An Arbitrary Lagrangian-Eulerian Computing Method for all Flow Speeds," J. Comp. Phys. 14, pp. 227-253 (1974).

[4] Kolsky, H. G., A Method for Numerical Solution of Transient Hydrodynamic Shock Problems in Two Space Dimensions, LA-1867 (April 1955).

[5] Amurud, L., and Orr, R. S., A Note on Inverted Centers of Pressure and Crossed Mass Points in a Two-dimensional Hydrodynamic Calculation, LASL Unpublished Note (May 1963).

[6] Chang, Y. W. and Gvildys, J., REXCO-HEP: A Two-dimensional Computer Code for Calculating the Primary System Response in Fast Reactors, ANL-75-19, (June 1975).

[7] Adamik, V., CEFRA-A Computational Programme for the Hydrodynamics Response of Primary Reactor Containment to Explosive Accident of Fast Reactor, Rep. of Power Research Institute - PRI (1973).

[8] PISCES 2DL Finite Difference Equations, 2DL-3, Physics International Co. (October, 1971).

[9] Wilkins, M. L., Calculation of Elastic-Plastic Flow, UCRL-7322, Rev. 1 (1969).

[10] Thorne, B. J. and Herrmann, W., TOODY, A Computer Program for Calculating Problems of Motion in Two Dimensions, SC-RR-66-602, (July, 1967).

[11] Hofmann, R., STEALTH, A Lagrange Explicit Finite Difference Code for Solids, Structure, and Thermodydraulic Analysis, EPRI NP-260 (August 1976).

[12] Cowler, M. S., ASTARTE-A 2-D Lagrangian Code for Unsteady Compressible Flow, Theoretical Description, AWRE-44-91-37, (March, 1974).

[13] Herrmann, W., Comparison of Finite Difference Expressions used in Lagrangian Fluid Flow Calculations, AFWL-TR-64-104 (November 1964).

[14] Harlow, F. H., and Amsden, A. A., "A Numerical Fluid Dynamics Calculation Method for All Flow Speeds," J. Comp. Phys. 8, pp. 197-213 (1971).

[15] Welch, J. E., Harlow, F. H., Shannon, J. P., and Daly, B. J., The MAC Method, A Computing Technique for Solving Viscous, Incompressible, Transient Fluid-Flow Problems Involving Free Surfaces, LA3425 (March 1966).

[16] Stievenart, M. Bonffioux, P., Egleme, M., Fabry, J. P., and Lamotte, H., "Analysis of LMFBR Explosion Model Experiments by means of the SURBOUM-II Code," E 3/5 Trans. 3rd Int'l. Conf. on Structural Mech. in Reactor Technology, London, UK, (September 1975).

[17] Wang, C. Y. ICECO-An Implicit Eulerian Method for Calculating Fluid Transient in Fast-Reactor Containment, ANL-75-81 (December 1975).

[18] Chu, H. Y., Chang, Y. W., and Fistedis, S. H., "MICE: A Two-eimensional Containment Code for Treating Multifield Fluid Flow with Interpenetrations," Trans. Am. Nucl. Soc. 24, pp. 277-278 (November 1976).

[19] A-Moneim, M. T., ICEPEL, A Two-dimensional Computer Program for the Transient Analysis of a Pipe-Elbow Loop, ANL-75-35 (July 1975).

[20] Oden, J. T., and Somogyi, D., "Finite-element Applications in Fluid Dynamics," Proc. ASCE, J. Eng. Mech. Div., 95, EM3, pp. 821-826 (1969).

[21] Belytschko, T. B., Kennedy, J. M., and Schoeberle, D. F., "On Finite Element and Difference Formulations of Transient Fluid Structure Problems," Proc. Conf. Comp. Methods Nucl. Eng., Charleston, S.C. (April 1975).

[22] Reactor Development Program Progress Report, ANL-RDP-58, 7.46 (February 1976)

[23] Amsden, A. A., and Hirt, C. W., YAQUI: An Arbitrary Lagrangian-Eulerian Computer Program for Fluid Flow at All Speeds, LA-5100 (March 1973).

[24] Belytschko, T. B., and Hsieh, B. J., "Nonlinear Transient Analysis of Shells and Solids of Revolution by Convected Elements," Proc. AIAA/ASME/SAE 14th Structures, Structural Dynamics and Materials Conf., Williamsburg, VA, (May 1973).

[25] Belytschko, T. B., WHAM, Waves in Hysteretic Arbitrary Media and Structures, Department of Materials Engineering Report, Univ. of Illinois at Chicago Circle, (Feb. 1974).

[26] Kennedy, J. M., Nonlinear Dynamic Response of Reactor-Core Subassemblies, ANL-8065 (Jan. 1974).

[27] Donea, J., Giuliani, S., and Halleux, J. P., "Prediction of the Nonlinear Dynamic Response of Structural Components using Finite Elements, T 2/4, Proc. Extreme Load Conditions and Limit Analysis Procedures for Structural Reactor Safeguards and Containment Structures, Berlin, Germany (September 1975).

[28] Belytschko, T. B., "Comparative Evaluation of Numerical Methods for Dynamic Structural Analysis," T 1/1, Proc. Extreme Load Conditions and Limit Analysis Procedures for Structural Reactor Safeguards and Containment Structures, Berlin, Germany (Sept. 1975).

[29] Hancock, S. L., Finite Difference Equations for PISCES 2DELK, A Coupled Euler Lagrange Continuum Mechanics Computer Program, TCAM 76-2, Physics International Co., (April 1976).

[30] Lauber, H., Private Communication.

[31] Thompson, S. L., CSQ-A Two-Dimensional Hydrodynamics Program with Energy, Flow and Material Strengths, SAND 72-D122, Sandia Laboratories (August 1975).

Table I Compilation of Containment Codes

Name Origin (Ref)	Problem Solved	Unique Capabilities	Discretization Fluid	Discretization Structure	Time Integration
REXCO-HEP ANL [6]	2D + Shell	Sliding lines for fluid/solid meshes and fluid/shell interfaces Rezoning by REZONE code	FD (Lagrangian)		Explicit
ICECO ANL [17]	2D + Shell	Can handle extended fluid motion (e.g. Bubble motion & Na spill)	FD (Euler.)	FE (Lagran.)	Implicit
MICE ANL [18]	2D + Shell	Multifields, allowing interpenetrations	FD (Euler.)	FE (Lagran.)	Implicit
ICEPEL ANL [19]	2D + Shell	Extensive models for piping components Can handle extended fluid motion	FD (Euler.)	FE (Lagran.)	Implicit
ASTARTE UKAEA [12]	2D + Shell	Slide, spill, void interfaces Auto. mesh generator	FD (Lagrangian)		Explicit
SORBOUM BN [16]	2D + Shell	Perforated dip plates Incompressible flow	FD (Euler.)	FD (Lagran.)	Implicit
ARES IA [30]	2D + Shell	Surface integral	FD (Lagrangian)		Explicit
EURDYN EU [27]	2D + Shell	Convected Coordinates Isotropic/Kinematic hardening laws for solid materials	FE (Lagrangian)		Explicit
CEFRA EGU [7]	2D + Shell	MFCI model	FD (Lagrangian)		Explicit
PISCES-2DL PI [8]	2D	Sliding lines for fluid/solid meshes, rezone sistatic option Auto. coordinate generator	FD (Lagrangian)		Explicit
PISCES-2DELK PI [29]	2D + Shell	Coupled Eulerian/Lagrangian Implicit and explicit option	FD (Euler.)	(Lagran.)	Explicit/ Implicit
HEMP LLL [9]	2D	Sliding lines capability for coupled fluid/solid meshes rezoning capability	FD (Lagrangian)		Explicit
TOODY SLA [10]	2D	Slide line capability for coupled fluid/solid meshes rezoning capability	FD (Lagrangian)		Explicit
STEALTH SAI [11]	2D + Shell	Continuous auto rezoner Thin shell option	FD (Lagrangian)	FE	Explicit
CSQ SLA [31]	2D	Extensive EOS options Conduction & radiation H.T. up to 10 fluids	FD (Eulerian)		Explicit
STRAW ANL [26]	2D	Convected Coordinates Isotropic strain hardening	FE (Lagrangian)		Explicit/ Implicit
WHAM UICC [25]	2D + Shell	Convected Coordinates Isotropic strain hardening	FE (Lagrangian)		Explicit/ Implicit

FD=finite-difference; FE=finite-element

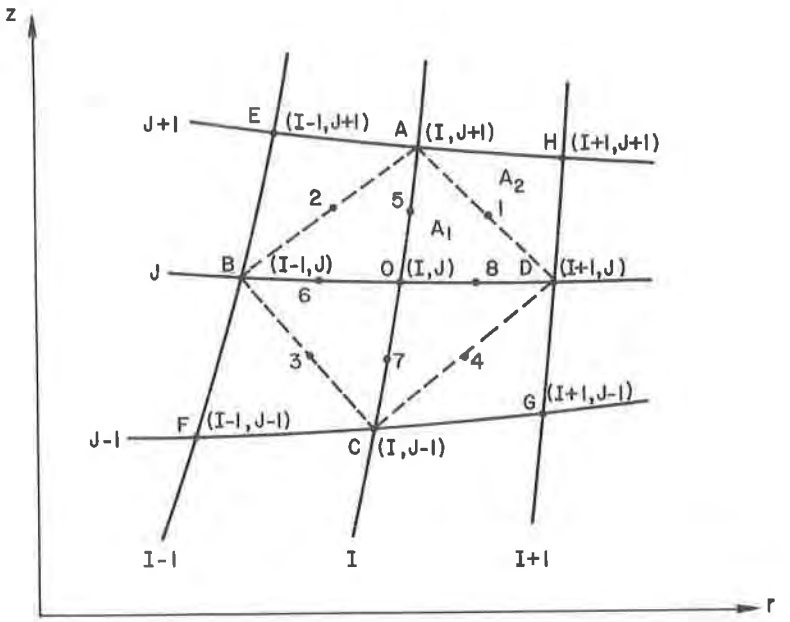

Fig. 1. A Typical Lagrangian Point with the Adjacent Meshes and Points, Illustrating the Notation Used in Finite-difference Equations

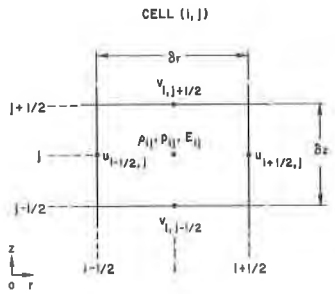

Fig. 2. A Typical Eulerian Cell, Illustrating the Notation Used in Finite-difference Equations

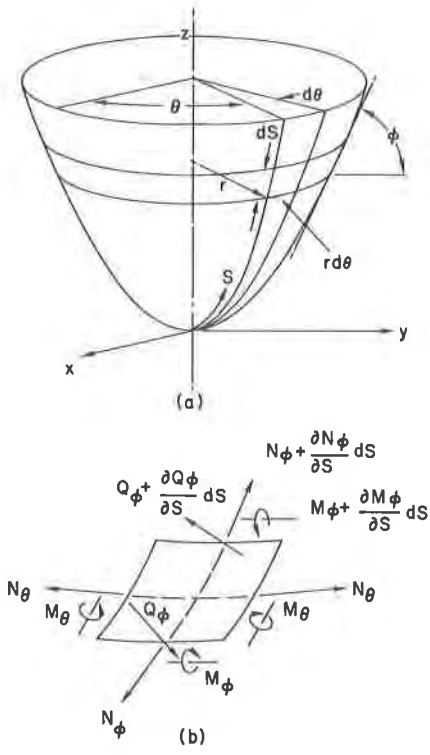

Fig. 3. Shell of Revolution

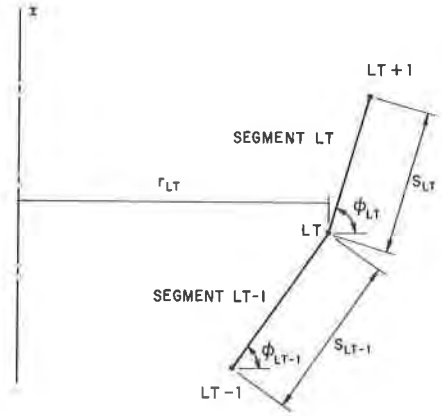

Fig. 4. A Typical Shell Node with the Adjacent Segments, Illustrating the Notation Used in Finite-difference Equations