

SAND98-1760/3
Unlimited Release
Printed March 1999

SEACAS Theory Manuals: Part III. Finite Element Analysis in Nonlinear Solid Mechanics

T.A. Laursen
Duke University
Durham, NC 27706

S.W. Attaway
Material and Structural Mechanics Department
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0443

R.I. Zadoks
University of Texas at El Paso
El Paso, TX 79968

Abstract

This report outlines the application of finite element methodology to large deformation solid mechanics problems, detailing also some of the key technological issues that effective finite element formulations must address. The presentation is organized into three major portions: first, a discussion of finite element discretization from the global point of view, emphasizing the relationship between a virtual work principle and the associated fully discrete system; second, a discussion of finite element technology, emphasizing the important theoretical and practical features associated with an individual finite element; and third, detailed description of specific elements that enjoy widespread use, providing some examples of the theoretical ideas already described. Descriptions of problem formulation in nonlinear solid mechanics, nonlinear continuum mechanics, and constitutive modeling are given in three companion reports.

Finite Element Formulation: Table of Contents

Introduction	1
Weak Form Revisited	2
Discretization	5
Introduction	5
Galerkin Finite Element Methods	6
Generation of Matrix Equations	8
Localization and Assembly	12
Quasistatics	17
Introduction	17
Internal Force Vector	17
External Force Vector	18
Incremental Load Approach	18
Dynamics	21
Introduction	21
The Semidiscrete Approach	21
Time-Stepping Procedures	21
Explicit Finite Element Methods	22
Implicit Finite Element Methods	25
Nonlinear Equation Solving	26
Introduction	26
Newton Raphson Framework	26
Line Search	28
Quasi-Newton Methods	30
Conjugate Gradient Methods	33
Preconditioning	38
Basics of Element Design	41
Introduction	41
Convergence	41
Parameterization	43
Shape Functions	47
Quadrature	50
Local Arrays	53
Advanced Element Design Issues	57
Introduction	57
Constrained Media and Locking	57
Selective/Fully Reduced Integration	59
Hourglass Control	60
Eight-Node Uniform Strain Element	63
Introduction	63
Element Force Vector	64
Lumped Mass Matrix	67

Finite Rotation Algorithm	68
Determination of Effective Moduli	70
Determination of the Stable Time Increment	71
Hourglass Control Algorithm	73
Artificial Bulk Viscosity	75
Four-Node Corotational Shell	78
Introduction	78
Shell Kinematics	79
Constitutive Assumptions	81
Shell Element Coordinate Systems	81
Element Equations	83
Computation of Internal Force and Moment Resultants	84
Hourglass Control	86
Calculation of the Stable Time Increment	87
Constitutive Models	88
Time-Stepping Algorithm	92
Bibliography	95

Introduction

In this chapter we explore the finite element techniques utilized in the description of large deformation problems in solid mechanics. Beginning with the notational framework and problem description discussed in Formulation of Nonlinear Problems and utilizing the nonlinear continuum mechanics and material modeling issues discussed in Nonlinear Continuum Mechanics and Table of Contents, we discuss in this chapter how discrete approximations to the governing nonlinear field equations are generated and solved.

The discussion will take place in three general stages. The first stage, consisting of the first five sections of the report, emphasizes the **global formulation of the finite element method** and treats aspects best understood by considering the discretized system in its entirety. Topics to be discussed in this way include a brief presentation of weak forms appropriate for large deformation problems, in Weak Form Revisited, a general discussion of Discretization, time independent and dependent problems (i.e., Quasistatics and Dynamics), and Nonlinear Equation Solving. These sections will emphasize the derivation of discrete system equations from the underlying variational principle, the form of these system equations in matrix form, and the iterative solution of these equations that is required for nonlinear applications.

The next stage treats **element technology**, presenting the fundamentals necessary to formulate and implement the basic building block of the finite element method: the finite element. Indeed, the most basic advantage of the finite element method over other more classical variational methods is its modularity. That is to say that the method of discretization is tailored to small systematically generated subdomains of the problem of interest (i.e., elements) making the method applicable to a myriad of geometrical situations. Importantly much of finite element technology is sufficiently generic so that many aspects of element formulation are virtually unchanged from application to application. We will discuss these aspects in two sections. The first, Basics of Element Design, will cover the most essential features of element design including requirements for global convergence, shape function definition, and numerical integration to produce local contributions to the global equations. The second section, Advanced Element Design Issues, deals with concerns more specific to large deformation solid mechanics with the primary concern being near incompressibility of materials and the effect numerical treatment of such phenomena.

The third stage of our presentation will consist of **specific element examples**, summarizing some formulations that are in particularly prevalent use in computational solid mechanics. In Eight-Node Uniform Strain Element, we present some of the implementational details associated with an element widely used for the description of three-dimensional continuous media, particularly in explicit dynamic and matrix-free quasistatic applications. In Four-Node Corotational Shell, a common structural element is discussed in some detail.

Weak Form Revisited

We begin by providing a brief review of the field equations to be considered. The problem to be solved is as shown schematically in Figure 1.7, where the finite deformation response of a body, denoted Ω in its reference configuration, is to be computed. Assuming that this time-dependent configuration mapping is denoted by $\varphi_{\mathbf{t}}$, the following problem is to be solved for each time, \mathbf{t} , in the time interval of interest:

$$\nabla \cdot \mathbf{T} + \mathbf{f} = \rho \mathbf{a} \text{ on } \varphi_{\mathbf{t}}(\Omega), \quad (3.1)$$

$$\varphi_{\mathbf{t}} = \bar{\varphi}_{\mathbf{t}} \text{ on } \varphi_{\mathbf{t}}(\Gamma_{\mathbf{u}}), \quad (3.2)$$

and

$$\mathbf{t} = \bar{\mathbf{t}} \text{ on } \varphi_{\mathbf{t}}(\Gamma_{\sigma}), \quad (3.3)$$

where all notations are as discussed in Notational Framework. In particular \mathbf{a} is the material acceleration expressed in spatial coordinates, \mathbf{f} is the body force per unit (spatial) volume, and \mathbf{T} is the Cauchy stress tensor. The vector \mathbf{t} is the Cauchy traction vector, obtained via $\mathbf{t} = \mathbf{T}\mathbf{n}$, where \mathbf{n} is the outward unit normal to the spatial surface $\varphi_{\mathbf{t}}(\Gamma_{\sigma})$.

The problem is also subject to initial conditions of the form

$$\varphi(\mathbf{x}, 0) = \varphi_0(\mathbf{x}) \text{ on } \Omega, \quad (3.4)$$

and

$$\frac{\partial \varphi}{\partial \mathbf{t}}(\mathbf{x}, 0) = \mathbf{v}_0(\mathbf{x}) \text{ on } \Omega. \quad (3.5)$$

Recall that although Eqs. (3.1)-(3.3) are written in the so-called spatial configuration, we still consider ourselves to be working in a Lagrangian framework, where all quantities are ultimately indexed to material points through the mapping $\mathbf{x} = \varphi_{\mathbf{t}}(\mathbf{X})$ (see Lagrangian and Eulerian Descriptions).

A prerequisite of the finite element method is that a weak, or variational, form of the above field equations be available for discretization. This can be obtained, following the general procedure outlined for linear problems in Weak Forms, by considering weighting functions φ^* , defined over Ω , which satisfy the following condition:

$$\varphi^* = 0 \text{ on } \Gamma_{\mathbf{u}} \quad (3.6)$$

(cf. (1.82)), where we also assume that all φ^* are sufficiently smooth so that any desired partial derivatives can be computed. In treating large deformation problems, it is useful to consider spatial forms of the functions φ^* , obtained by composition with the (unknown) mapping φ_t^{-1} . We denote these spatial variations in the sequel by \mathbf{w} , and note that they may be obtained via

$$\mathbf{w}(\mathbf{x}) = \varphi^*(\varphi_t^{-1}(\mathbf{x})) \quad (3.7)$$

for any $\mathbf{x} \in \varphi_t(\mathbf{X})$. This causes the condition

$$\mathbf{w} = \mathbf{0} \text{ on } \varphi_t(\Gamma_u) \quad (3.8)$$

to be satisfied, and provided the configuration mapping φ_t is smooth (which we assume to be the case), all required partial derivatives of \mathbf{w} can be computed.

With these definitions in hand, the development in Weak Forms can be reproduced in the current context to provide the following spatial representation of the variational form for large deformations:

Given the boundary conditions $\bar{\mathbf{e}}$ on $\varphi_t(\Gamma_\sigma)$, $\bar{\varphi}_t$ on $\varphi_t(\Gamma_u)$, the initial conditions φ_0 and \mathbf{V}_0 on Ω , and the distributed body force \mathbf{f} on $\varphi_t(\Omega)$, find $\varphi_t \in S_t$ for each time $t \in (0, T)$ such that:

$$\begin{aligned} & \int_{\varphi_t(\Omega)} \rho \mathbf{w} \cdot \mathbf{a} \, dv + \int_{\varphi_t(\Omega)} (\nabla \mathbf{w}) : \mathbf{T} \, dv \\ & = \left(\int_{\varphi_t(\Omega)} \mathbf{w} \cdot \mathbf{f} \, dv + \int_{\varphi_t(\Gamma_\sigma)} \mathbf{w} \cdot \bar{\mathbf{e}} \, da \right) \end{aligned} \quad (3.9)$$

for all admissible \mathbf{w} , where S_t is as defined as

$$S_t = \{ \varphi_t \mid \varphi_t = \bar{\varphi}_t \text{ on } \Gamma_u, \varphi_t \text{ is smooth} \} \quad (3.10)$$

and where admissible \mathbf{w} are related in a one-to-one manner via (3.7) to the material variations $\varphi^* \in W$ with the definition of W being

$$W = \{ \varphi^* \mid \varphi^* = \mathbf{0} \text{ on } \Gamma_u, \varphi^* \text{ is smooth} \}. \quad (3.11)$$

Note that in contrast to previous development, the constitutive relation governing \mathbf{T} is left unspecified: it can, in general, be subject to both geometric and material nonlinearities. The notation \mathbf{a} for the acceleration is to be understood as the material acceleration, as defined by (2.27) in Material and Spatial Velocity and Acceleration.

In addition, the solution φ is subject to the following conditions at $\tau = 0$:

$$\int_{\Omega} \varphi^* \cdot (\varphi|_{\tau=0} - \varphi_0) d\Omega = 0 \quad (3.12)$$

and

$$\int_{\Omega} \varphi^* \cdot \left(\left. \frac{\partial \varphi}{\partial \tau} \right|_{\tau=0} - \mathbf{v}_0 \right) d\Omega = 0, \quad (3.13)$$

both of which must hold for all $\varphi^* \in W$.

Discretization

Introduction

The process of numerically approximating a continuous problem is generically called **discretization**. In the finite element method, the entity discretized is a weak form (alternatively, variational equation). In the current context the variational form to be considered is that described in Weak Form Revisited. We now refer the reader to Figure 3.1, which gives the general notation to be used in description of the discretization process for the problem at hand.

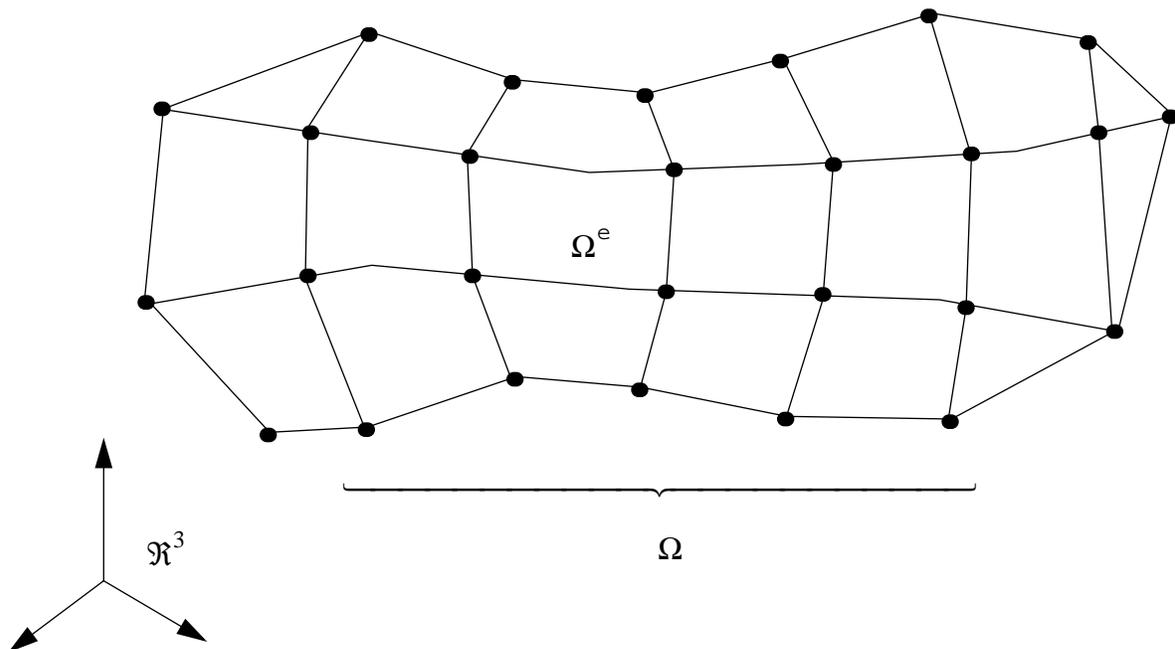


Figure 3.1 General notation for finite element discretization of the reference domain.

As referred to in Figure 3.1, the reference domain Ω is subdivided into a number of element subdomains, Ω^e , where the superscript e is an index to the specific element in question, running between 1 and n_{e1} , where n_{e1} is the total number of elements required for the discretization.

We assume in the figure and throughout the ensuing discussion that Ω is a subset of \mathcal{R}^3 , with the two-dimensional case readily obtained as a special case of the theory we will discuss.

Note also from Figure 3.1 that a number of nodal points are indicated by the dots. We shall assume that all degrees of freedom in the discrete system to be proposed will be associated with these nodes. As one might also notice, these nodes may lay at corners, edges, and in interiors of the elements with which they are associated. A key feature of the finite element method will be that a specific element can be completely characterized by the coordinates and degrees of

freedom associated with the nodes attached to it. In the following we will index the nodes with uppercase letters A, B, C, etc., with such indices running between 1 and n_{np} , the total number of nodal points in the problem.

Galerkin Finite Element Methods

The essence of any finite element method lies in the discretization of a weak or variational form. This discretization process involves two important approximations: approximation of a typical member of the solution space S_τ , and approximation of the weighting space W . These approximations are typically expressed as an expansion in terms of prescribed shape or interpolation functions, usually associated with specific nodal points in the mesh. Since the number of nodal points is obviously finite, the expansion is likewise finite, giving rise to the concept of a finite-dimensional approximation of a space.

Roughly speaking, the idea of discretization is as follows. We know from earlier chapters that if the variational equation is enforced considering all $\varphi_\tau \in S_\tau$ and $\varphi^* \in W$ as mandated by its definition, then the solution of the weak form is completely equivalent to that of the strong form (i.e., the governing partial differential equation with boundary/initial conditions). This fact results because of the arbitrary nature of the φ^* and because of the very general definitions for S_τ and W . If we restrict our attention only to some subset of the above spaces, we now make an error with the solution of our approximated weak form no longer being identical to the solution of the strong form. If our choice for the type of shape functions to be used is reasonable, however, we can represent the full solution and weighting spaces with arbitrarily closeness by increasing the number of nodal points and/or the degree of polynomial approximation utilized in the interpolation functions. In the limit of such refinement, we should expect recovery of the exact solution (i.e., convergence).

Let us represent the shape function associated with node A as N_A , and assume it to be as follows:

$$N_A: \bar{\Omega} \rightarrow \Re. \quad (3.14)$$

Given a time, τ , the finite-dimensional counterpart of φ_τ will be denoted as φ_τ^h and is expressed in terms of the shape functions as

$$\varphi_\tau^h = \sum_{B=1}^{n_{np}} N_B \mathbf{d}_B(\tau), \quad (3.15)$$

where $\mathbf{d}_B(\tau)$ is a 3-vector containing the (in general unknown) coordinates of nodal point B at time τ . Given a prescribed set of nodal shape functions $\{N_B\}$, $B = 1, \dots, n_{np}$, the finite dimensional solution space S_τ^h is defined as the collection of all such ϕ_τ^h :

$$S_\tau^h = \left\{ \phi_\tau^h = \sum_{B=1}^{n_{np}} N_B \mathbf{d}_B(\tau) \mid \phi_\tau^h \approx \bar{\phi}_\tau(\mathbf{x}) \text{ for all } \mathbf{x} \in \Gamma_u \right\}. \quad (3.16)$$

In other words, we require members of the discrete solution space to (approximately) satisfy the displacement boundary condition on Γ_u . The approximation comes about because, in general, we only force ϕ_τ^h to interpolate the nodal values of $\bar{\phi}_\tau$ on Γ_u with the N_B serving as the interpolation functions. We might also note that Γ_u itself is typically geometrically approximated by the finite element discretization, contributing also to the approximation.

This defines the discretization procedure for ϕ_τ^h , at least notationally. It still remains, however, to approximate the weighting space. The (Bubnov-) Galerkin finite element method is characterized by utilizing the same shape functions to approximate W as were used to approximate S_τ .

Accordingly, we define a member of this space, ϕ^{*h} , via

$$\phi^{*h} = \sum_{A=1}^{n_{np}} N_A \mathbf{c}_A, \quad (3.17)$$

where the \mathbf{c}_A are 3-vectors of nodal constants. We can then express the finite dimensional weighting space W^h via

$$W^h = \left\{ \phi^{*h} = \sum_{A=1}^{n_{np}} N_A \mathbf{c}_A \mid \phi^{*h} = 0 \text{ for all } \mathbf{x} \in \Gamma_u \right\}. \quad (3.18)$$

Analogous to the situation for S_τ^h , Eq. (3.18) features a discrete version of the boundary condition on Γ_u . In other words, W^h consists of all functions of the form (3.17) resulting in satisfaction of this condition. Note that the only restriction on the \mathbf{c}_A is that they result in satisfaction of the homogeneous boundary condition on Γ_u ; they are otherwise arbitrary.

With these ideas in hand, the approximate Galerkin solution to the initial/boundary value problem takes the form described below.

Given the boundary conditions $\bar{\mathbf{e}}$ on $\phi_t^h(\Gamma_\sigma)$, $\bar{\phi}_t$ on $\phi_t^h(\Gamma_u)$, the initial conditions ϕ_0 and \mathbf{v}_0 on Ω , and the distributed body force \mathbf{f} on $\phi_t^h(\Omega)$, find $\phi_t^h \in S_t^h$ for each time $t \in (0, T)$ such that:

$$\begin{aligned} & \int_{\phi_t^h(\Omega)} \rho \mathbf{w}^h \cdot \mathbf{a}^h dv + \int_{\phi_t^h(\Omega)} (\nabla \mathbf{w}^h) : \mathbf{T}^h dv \\ &= \int_{\phi_t^h(\Omega)} \mathbf{w}^h \cdot \mathbf{f} dv + \int_{\phi_t^h(\Gamma_\sigma)} \mathbf{w}^h \cdot \bar{\mathbf{e}} da \end{aligned} \quad (3.19)$$

for all admissible \mathbf{w}^h , where S_t is as defined in (3.16) and where admissible \mathbf{w}^h are related to the material variations $\phi^{*h} \in W^h$ via

$$\mathbf{w}^h(\mathbf{x}) = \phi^{*h}((\phi_t^h)^{-1}(\mathbf{x})). \quad (3.20)$$

In Eq. (3.19) \mathbf{T}^h refers to the Cauchy stress field computed from the discrete mapping ϕ_t^h through the constitutive relations, whereas \mathbf{a}^h is the discrete material acceleration.

The initial conditions are ordinarily simplified in the discrete case to simply read:

$$\mathbf{d}_B(0) = \bar{\phi}_0(\mathbf{x}_B) \quad (3.21)$$

and

$$\dot{\mathbf{d}}_B(0) = \mathbf{v}_0(\mathbf{x}_B), \quad (3.22)$$

both of which must hold for all nodes $B = 1, \dots, n_{np}$, where \mathbf{x}_B are the reference coordinates of the node in question.

Generation of Matrix Equations

We are now in a position to summarize the discrete equations that will result from Eq. (3.19). Before doing so, let us develop one more notational necessity. We can reexpress the nodal vectors \mathbf{c}_A and \mathbf{d}_B in terms of their components via:

$$\mathbf{c}_A = \{c_{iA}\}, i = 1, 2, 3 \quad (3.23)$$

and

$$\mathbf{d}_B = \{d_{jB}\}, j = 1, 2, 3. \quad (3.24)$$

Note that indices i and j are spatial indices, in general. It is useful in generating matrix equations to have indices referring not to nodes A and B or spatial directions i and j , but rather to degree of freedom numbers in the problem. Toward this end we define for notational convenience the concept of an ID array that is set up as follows:

$$\text{ID}(i, A) = P \text{ (global degree of freedom number)}. \quad (3.25)$$

In other words, the ID array takes the spatial direction index and nodal number as arguments and assigns a global degree of freedom number to the corresponding unknown. In general, the number of degrees of freedom is n_{dof} , given by

$$n_{\text{dof}} = 3 \times n_{\text{np}}. \quad (3.26)$$

With this notation in hand, the equation numbers P and Q are defined as follows:

$$P = \text{ID}(i, A) \quad (3.27)$$

and

$$Q = \text{ID}(j, B). \quad (3.28)$$

We now generate the discrete equations by substitution of Eqs. (3.15) and (3.17) into (3.19), causing the variational equation to read:

$$\begin{aligned} & \int_{\phi_t^h(\Omega)} \rho \left(\sum_{A=1}^{n_{\text{np}}} N_A(\phi_t^{-1}(\mathbf{x})) \mathbf{c}_A \right) \cdot \left(\sum_{B=1}^{n_{\text{np}}} N_B(\phi_t^{-1}(\mathbf{x})) \ddot{\mathbf{d}}_B(t) \right) dV \\ & + \int_{\phi_t^h(\Omega)} \left(\sum_{A=1}^{n_{\text{np}}} \nabla N_A(\phi_t^{-1}(\mathbf{x})) \otimes \mathbf{c}_A \right) : \mathbf{T}^h dV, \quad (3.29) \\ & = \int_{\phi_t^h(\Omega)} \left(\sum_{A=1}^{n_{\text{np}}} N_A(\phi_t^{-1}(\mathbf{x})) \mathbf{c}_A \right) \cdot \mathbf{f} dV + \int_{\phi_t^h(\Gamma_\sigma)} \left(\sum_{A=1}^{n_{\text{np}}} N_A(\phi_t^{-1}(\mathbf{x})) \mathbf{c}_A \right) \cdot \bar{\mathbf{t}} da \end{aligned}$$

where we note in particular that \mathbf{T}^h is a function of $\phi_t^h = \sum_{B=1}^{n_{\text{np}}} N_B \mathbf{d}_B(t)$ through the strain-displacement relations (nonlinear, in general) and the constitutive law (as yet unspecified and perhaps likewise nonlinear).

Proceeding now to examine (3.29) term-by-term, the inertial term can be expanded as follows

$$\begin{aligned}
& \int_{\varphi_t^h(\Omega)} \rho \left(\sum_{A=1}^{n_{np}} N_A(\varphi_t^{-1}(\mathbf{x})) \mathbf{c}_A \right) \cdot \left(\sum_{B=1}^{n_{np}} N_B(\varphi_t^{-1}(\mathbf{x})) \ddot{\mathbf{d}}_B(t) \right) dv \\
&= \sum_{A=1}^{n_{np}} \sum_{i=1}^3 \int_{\varphi_t^h(\Omega)} \left(\rho N_A(\varphi_t^{-1}(\mathbf{x})) c_{iA} \left(\sum_{B=1}^{n_{np}} N_B(\varphi_t^{-1}(\mathbf{x})) \ddot{d}_{iB} \right) dv \right) \\
&= \sum_{A=1}^{n_{np}} \sum_{i=1}^3 c_{iA} \left[\sum_{B=1}^{n_{np}} \sum_{j=1}^3 \int_{\varphi_t^h(\Omega)} \rho N_A(\varphi_t^{-1}(\mathbf{x})) \delta_{ij} N_B(\varphi_t^{-1}(\mathbf{x})) dv \ddot{d}_{jB} \right], \\
&= \sum_{P=1}^{n_{dof}} c_P \left(\sum_{Q=1}^{n_{dof}} M_{PQ} \ddot{d}_Q \right)
\end{aligned} \tag{3.30}$$

where M_{PQ} is defined as follows:

$$M_{PQ} = \int_{\varphi_t^h(\Omega)} \rho N_A(\varphi_t^{-1}(\mathbf{x})) \delta_{ij} N_B(\varphi_t^{-1}(\mathbf{x})) dv. \tag{3.31}$$

The second term of (3.29) can be simplified via

$$\begin{aligned}
& \int_{\varphi_t^h(\Omega)} \left(\sum_{A=1}^{n_{np}} \nabla N_A(\varphi_t^{-1}(\mathbf{x})) \otimes \mathbf{c}_A \right) : \mathbf{T}^h dv \\
&= \int_{\varphi_t^h(\Omega)} \left(\sum_{A=1}^{n_{np}} \sum_{i=1}^3 \sum_{j=1}^3 N_{A,j}(\varphi_t^{-1}(\mathbf{x})) c_{iA} T_{ij}^h \right) dv, \\
&= \sum_{P=1}^{n_{dof}} c_P F_P^{int}
\end{aligned} \tag{3.32}$$

where

$$F_P^{int} = \int_{\varphi_t^h(\Omega)} \left[\sum_{j=1}^3 N_{A,j}(\varphi_t^{-1}(\mathbf{x})) T_{ij}^h \right] dv. \tag{3.33}$$

Finally, the last two terms of (3.29) can be treated as

$$\int_{\varphi_t^h(\Omega)} \left(\sum_{A=1}^{n_{np}} N_A(\varphi_t^{-1}(\mathbf{x})) \mathbf{c}_A \right) \cdot \mathbf{f} dv + \int_{\varphi_t^h(\Gamma_\sigma)} \left(\sum_{A=1}^{n_{np}} N_A(\varphi_t^{-1}(\mathbf{x})) \mathbf{c}_A \right) \cdot \bar{\mathbf{t}} da = \sum_{P=1}^{n_{dof}} c_P F_P^{ext}, \tag{3.34}$$

where

$$\mathbf{F}_P^{\text{ext}} = \int_{\Phi_t^h(\Omega)} N_A(\Phi_t^{-1}(\mathbf{x})) f_i dV + \int_{\Phi_t^h(\Gamma_\sigma)} N_A(\Phi_t^{-1}(\mathbf{x})) \cdot \bar{t}_i da. \quad (3.35)$$

We now define the following vectors and matrices of global variables, all with dimension of n_{dof} :

$$\begin{aligned} \mathbf{c} &= \{c_p\} \\ \mathbf{d}(t) &= \{d_Q(t)\} \\ \mathbf{F}^{\text{int}}(\mathbf{d}(t)) &= \{F_P^{\text{int}}\} . \\ \mathbf{F}^{\text{ext}} &= \{F_P^{\text{ext}}\} \\ \mathbf{M} &= [M_{PQ}] \end{aligned} \quad (3.36)$$

The results of Eqs. (3.30)-(3.35) can now be summarized as follows:

$$\mathbf{c}^T [\mathbf{M}\ddot{\mathbf{d}}(t) + \mathbf{F}^{\text{int}}(\mathbf{d}(t)) - \mathbf{F}^{\text{ext}}] = 0, \quad (3.37)$$

which must hold for all n_{dof} -vectors \mathbf{c} that result in satisfaction of the homogeneous boundary condition imposed on W (i.e., Eq. (3.18)).

Finally, we make the observation that not all of the members of $\mathbf{d}(t)$ are unknown; for those nodes lying on Γ_u , these degrees of freedom are prescribed. Furthermore, the corresponding entries of \mathbf{c} at these nodes are typically taken to be zero, so that the aforementioned condition on W^h is obeyed. Since the remainder of the vector \mathbf{c} is arbitrary, it must be the case that the elements of the bracketed term in (3.37) corresponding to unprescribed degrees of freedom must be identically zero, so that (3.37) will hold for arbitrary combinations of the c_p . Thus we can write the following nonlinear equation that expresses the discrete equations of motion:

$$\mathbf{M}\ddot{\mathbf{d}}(t) + \mathbf{F}^{\text{int}}(\mathbf{d}(t)) = \mathbf{F}^{\text{ext}}. \quad (3.38)$$

Here we employ a slight abuse of notation because we have asserted in (3.36) that all vectors and matrices have dimension n_{dof} ; and yet we only enforce Eq. (3.38) for unprescribed degrees of freedom. Denoting the number of unprescribed degrees of freedom as n_{eq} , one can account for this difference in practice by calculating the vector and matrix entries for all degrees of freedom and then by merely disregarding the $n_{\text{dof}} - n_{\text{eq}}$ equations corresponding to the prescribed degrees of freedom. The members of $\mathbf{d}(t)$ that are prescribed do need to be retained in its

definition, however, since they enter into both terms on the left-hand side of (3.38). It should simply be remembered that only n_{eq} members of $\mathbf{d}(\tau)$ are, in fact, unknown.

Localization and Assembly

The development to this point is mostly a matter of mathematical manipulation with little insight gained into the character of the interpolation functions, N_A . In fact, the basic nature of these interpolation functions distinguishes the finite element method from other variational solution techniques.

The details of shape function construction will be discussed in Basics of Element Design in the context of element programming. However, it is useful to discuss now the basic character of finite element approximation functions to give general insight into the structure of the method. We refer then to Figure 3.2, which depicts a node, A, in Ω and some generic elements attached to it. A basic starting point for the development of a finite element method is as follows: the shape function associated with Node A, N_A , is only nonzero in that subportion of Ω encompassed by the elements associated with Node A and is zero everywhere else in Ω .

This property of the shape functions is crucial to the modular character of the finite element method. Shape functions N_A having this property are said to possess local support.

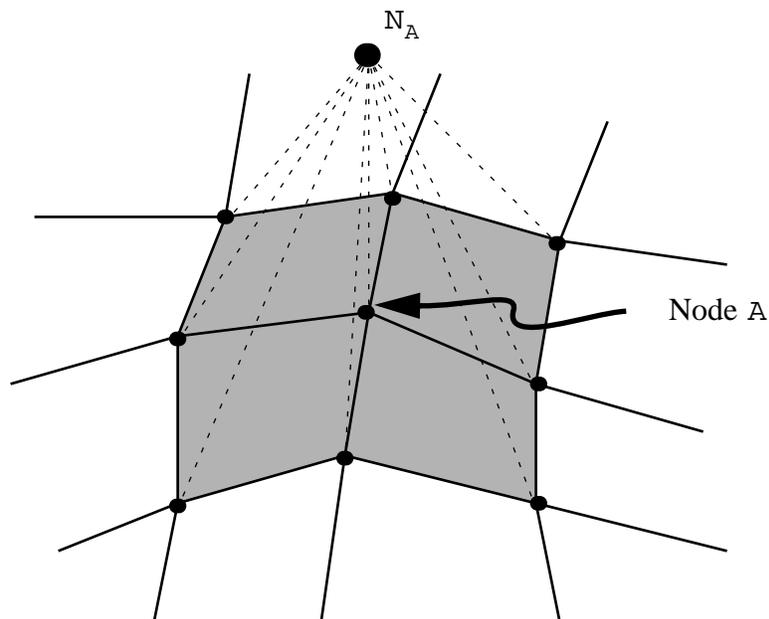


Figure 3.2 Local support of finite element interpolation functions. Region of support for N_A shown as shaded.

To gain insight into the effect of this property, let us examine the expression given in Eq. (3.31) for an element of the mass matrix M_{PQ} . We note in particular that the integrand of (3.31) will only be nonzero if both Nodes A and B share a common element in the mesh; otherwise, M_{PQ} must be zero. If we fix our attention on a given Node A in the mesh, we can, therefore, conclude that very few Nodes B will produce nonzero entries in \mathbf{M} . This matrix is, therefore, sparse; and it would be a tremendous waste of time to try to compute \mathbf{M} by looping over all the possible combinations of node numbers and spatial indices without regard to elements and the node numbers attached to them.

Instead the global matrices and vectors needed in the solution of (3.38) are more typically computed using two important concepts: localization and assembly. Still considering the matrix \mathbf{M} as an example, we note that by the elementary properties of integration, we can write:

$$\begin{aligned}
M_{PQ} &= \int_{\varphi_t^h(\Omega)} \rho N_A(\varphi_t^{-1}(\mathbf{x})) \delta_{ij} N_B(\varphi_t^{-1}(\mathbf{x})) d\mathbf{v} \\
&= \sum_{e=1}^{n_{e1}} \int_{\varphi_t^h(\Omega^e)} \rho N_A(\varphi_t^{-1}(\mathbf{x})) \delta_{ij} N_B(\varphi_t^{-1}(\mathbf{x})) d\mathbf{v}, \\
&= \sum_{e=1}^{n_{e1}} M_{PQ}^e
\end{aligned} \tag{3.39}$$

where

$$M_{PQ}^e = \int_{\varphi_t^h(\Omega^e)} \rho N_A(\varphi_t^{-1}(\mathbf{x})) \delta_{ij} N_B(\varphi_t^{-1}(\mathbf{x})) d\mathbf{v}. \tag{3.40}$$

Thus, the global mass matrix can be computed as the sum of a number of element mass matrices. This fact in itself is not especially useful because each of the \mathbf{M}^e is extremely sparse, even more so than \mathbf{M} . In fact, the only entries of \mathbf{M}^e that will be nonzero will be those for which both P and Q are degrees of freedom associated with element e.

This fact can be exploited by defining another local element matrix \mathbf{m}^e containing only degrees of freedom associated with that element. We introduce element degree of freedom indices p and q, as indicated generically in Figure 3.3. Assuming that p and q can take on values between 1 and n_{edf} , where n_{edf} is the number of degrees of freedom associated with the element, an $n_{edf} \times n_{edf}$ matrix \mathbf{m}^e is constructed as follows:

$$\mathbf{m}^e = [m_{pq}^e]. \tag{3.41}$$

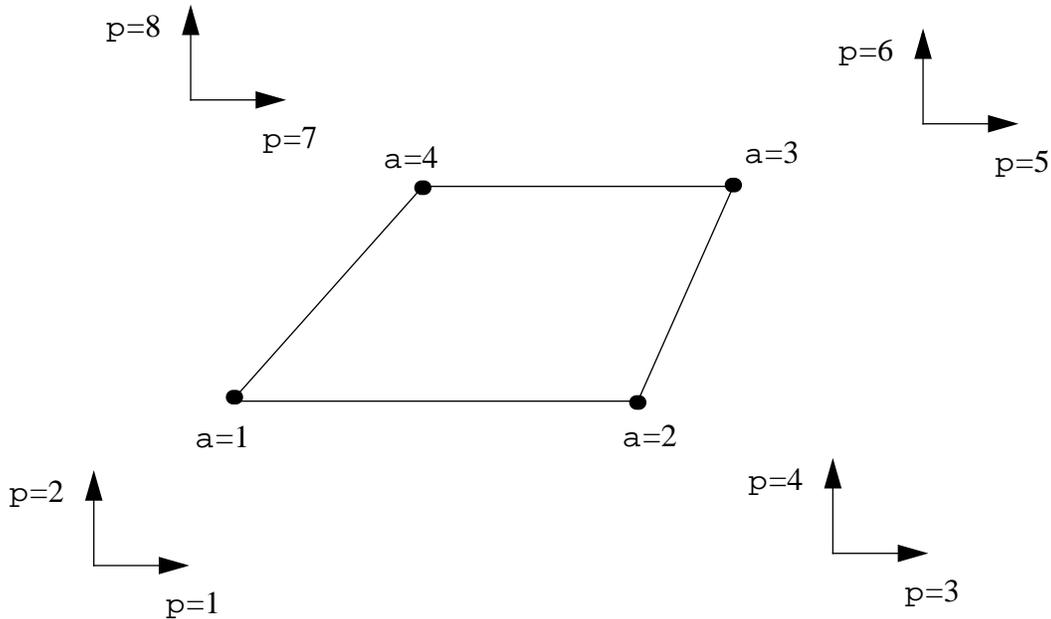


Figure 3.3 Element (local) degrees of freedom for a sample finite element.

The m_{pq}^e can be specified by introducing the concept of a local node number a or b , as also shown in Figure 3.3. With these definitions we can write

$$m_{pq}^e = \int_{\phi_t^h(\Omega^e)} \rho N_a(\phi_t^{-1}(\mathbf{x})) \delta_{ij} N_b(\phi_t^{-1}(\mathbf{x})) dV, \quad (3.42)$$

where a sample relationship between indices i , a , and p appropriate for the element at hand might be

$$p = (a - 1) \times 2 + i \quad (3.43)$$

(similarly for j , b and q). The notation N_a simply refers to the shape function associated with local Node a . By definition it is the restriction of the global interpolation function N_A to the element domain.

Calculation of local element entities, such as \mathbf{m}^e , turns out to be a highly modular procedure whose form remains essentially unchanged for any element in a mesh. Detailed discussion of this calculation is deferred until Basics of Element Design. Let us suppose for a moment, however, that we have a procedure in hand for calculating this matrix. We might then propose the following procedure for calculating the global mass matrix \mathbf{M} and internal force vector \mathbf{F}^{int} :

Step 1: Zero out \mathbf{M} , \mathbf{F}^{int} .

Step 2: For each element e , $e = 1, \dots, n_{e1}$:

- a) Prepare local data necessary for element calculations – e.g., \mathbf{x}^e (n_{edf} -vector of element nodal coordinates), \mathbf{d}^e (n_{edf} -vector of element nodal configuration mappings), etc.
- b) Calculate element internal force vector $\mathbf{f}^{int}|_p^e = \{f^{int}|_p^e\}$ and element mass matrix $\mathbf{m}^e = [m_{pq}^e]$ via

$$\mathbf{f}^{int}|_p^e = \int_{\phi_t^h(\Omega^e)} \left[\sum_{j=1}^3 N_{a,j}(\phi_t^{-1}(\mathbf{x})) T_{ij}^h \right] d\mathbf{v} \quad (3.44)$$

and Eq. (3.42).

- c) Assemble the element internal force vector and element mass matrix into their global counterparts by performing the following calculations for all local degrees of freedom p and q :

$$M_{PQ} = M_{PQ} + m_{pq}^e \quad (3.45)$$

and

$$F_P^{int} = F_P^{int} + f^{int}|_p^e, \quad (3.46)$$

where local degrees of freedom are related to global degrees of freedom via the LM array, defined so that

$$P = LM(p, e) \quad (3.47)$$

and

$$Q = LM(q, e). \quad (3.48)$$

Step 2a) above is referred to as localization; given a particular element, e , it extracts the local information from the global arrays necessary for element level calculations. Step 2b) consists of element level calculations; these computations will be discussed in detail in Basics of Element Design. Step 2c) is the process known as assembly and takes the data produced by the element level calculations and places them in the proper locations of the global arrays.

We can thus now summarize the effect of localization and assembly in a finite element architecture. They act as pre- and post-processors to the element-level calculations, enabling the entities needed for global equilibrium calculations to be computed in a modular manner as a summation of element contributions. Of course, the effectiveness of this procedure, as well as the convergence behavior of the numerical method in general, depends crucially on the interpolation

functions chosen and their definitions in terms of elements. We defer this topic for now and concentrate in the coming sections on the classes of problems and global equation-solving strategies to be utilized.

Quasistatics

Introduction

As discussed previously in the context of Linear Elastic IBVP in The Quasistatic Approximation, the quasistatic approximation is appropriate when inertial forces are negligible compared to the internal and applied forces in a system. As discussed in Discretization, the quasistatic system of equations is obtained by omission of the inertial term in the discrete equations of motion. Thus in this section we discuss solution of problems of the form:

$$\mathbf{F}^{\text{int}}(\mathbf{d}(\tau)) = \mathbf{F}^{\text{ext}}(\tau) \quad (3.49)$$

subject to only one initial condition of the form

$$\mathbf{d}(0) = \mathbf{d}_0. \quad (3.50)$$

Note that the time variable τ may correspond to real time (e.g., if rate-dependent material response is considered) but need not have physical meaning for rate-independent behavior. For example, it is common for τ to be taken as a generic parameterization for the applied loading on the system, as discussed below in Incremental Load Approach.

It could also be noted that if the initial condition were taken as the same as the reference configuration of the body, then

$$\mathbf{d}_0|_{\mathbf{A}} = \mathbf{x}_{\mathbf{A}}. \quad (3.51)$$

Internal Force Vector

The quantity $\mathbf{F}^{\text{int}}(\mathbf{d}(\tau))$ is known as the internal force vector and consists of that set of forces that are variationally consistent with the internal stresses in the body undergoing analysis. The generic expression for an element in this vector is

$$\mathbf{F}_p^{\text{int}} = \int_{\phi_{\tau}^h(\Omega)} \left(\sum_{j=1}^3 N_{\mathbf{A},j}(\phi_{\tau}^{-1}(\mathbf{x})) \mathbf{T}_{ij}^h \right) d\mathbf{v}, \quad (3.52)$$

as given in Generation of Matrix Equations. This vector-valued operator is, in general, a nonlinear function of the unknown solution vector $\mathbf{d}(\tau)$ due to the possible Material Nonlinearity and/or Geometric Nonlinearity inherent in the definition of the Cauchy stress \mathbf{T}_{ij}^h

in (3.52). As implied by our notation, we assume the solution vector \mathbf{d} to be smoothly parameterized by τ , which may represent time or some other loading parameter.

External Force Vector

The external load vector $\mathbf{F}^{\text{ext}}(\tau)$ must equilibrate the internal force vector, as is clear from Eq. (3.49). As first presented in Generation of Matrix Equations, the expression for an element $\mathbf{F}_p^{\text{ext}}$ of $\mathbf{F}^{\text{ext}}(\tau)$ is as follows

$$\mathbf{F}_p^{\text{ext}} = \begin{bmatrix} \int_{\phi_\tau^h(\Omega)} N_A(\phi_\tau^{-1}(\mathbf{x})) \mathbf{f}_i(\tau) d\mathbf{v} \\ + \int_{\phi_\tau^h(\Gamma_\sigma)} N_A(\phi_\tau^{-1}(\mathbf{x})) \cdot \bar{\mathbf{t}}_i(\tau) d\mathbf{a} \end{bmatrix}, \quad (3.53)$$

where the explicit dependence of \mathbf{f}_i and $\bar{\mathbf{t}}_i$ upon τ has been indicated and where

$\mathbf{P} = \text{ID}(i, a)$, as given in (3.27). In other words, we assume that the prescribed internal force loadings \mathbf{f}_i and prescribed surface tractions $\bar{\mathbf{t}}_i$ are given functions of τ .

Equation (3.53) as written implies no dependence of either $\bar{\mathbf{t}}_i$ or \mathbf{f}_i upon $\phi_\tau(\mathbf{x})$ (and thus \mathbf{d}). Provided no such dependence exists, the external force vector is completely parameterized by τ , and the sole dependence of the equilibrium equations upon \mathbf{d} occurs through \mathbf{F}^{int} . However, it is important to realize that some important loading cases are precluded by this assumption, with perhaps the most important being the case of pressure loading, where the direction of applied traction is opposite to the outward surface normal, which in large deformation problems depends upon $\phi_\tau(\mathbf{x})$. Such a load is sometimes called a follower force and will, in general, contribute additional nonlinearity to the problem. Such complications are readily handled but are not encompassed by our current notational framework for the sake of simplicity.

Incremental Load Approach

We may now summarize the global solution strategy most commonly applied to quasistatic nonlinear solid mechanics applications. We assume that we are interested in the solution $\mathbf{d}(\tau)$ over some interval of interest for τ :

$$\tau \in [0, T]. \quad (3.54)$$

We subdivide this interval of interest into a set of subintervals via

$$[0, T] = \bigcup_{n=0}^{N-1} [t_n, t_{n+1}], \quad (3.55)$$

where n is an index on the time steps or intervals, and N is the total number of such increments. We assume that $t_0 = 0$ and that $t_N = T$, but we do not, in general, assume that all time intervals $[t_n, t_{n+1}]$ have the same width.

With this notation in hand, the incremental load approach attempts to solve the following problem successively in each time interval $[t_n, t_{n+1}]$:

Given the solution \mathbf{d}_n corresponding to time level t_n , find \mathbf{d}_{n+1} corresponding to t_{n+1} satisfying:

$$\mathbf{F}^{\text{int}}(\mathbf{d}_{n+1}) = \mathbf{F}^{\text{ext}}(t_{n+1}). \quad (3.56)$$

This governing equation is also often expressed by introducing the concept of a residual vector $\mathbf{R}(\mathbf{d}_{n+1})$:

$$\mathbf{R}(\mathbf{d}_{n+1}) = \mathbf{F}^{\text{ext}}(t_{n+1}) - \mathbf{F}^{\text{int}}(\mathbf{d}_{n+1}). \quad (3.57)$$

Solution of (3.56), therefore, amounts to finding the root of the equation

$$\mathbf{R}(\mathbf{d}_{n+1}) = 0. \quad (3.58)$$

The physical meaning of this approach is depicted graphically in Figure 3.4. Starting with an initial equilibrium state at t_n , so that $\mathbf{R}(\mathbf{d}_n) = 0$, we introduce a prescribed load increment $\Delta \mathbf{F}^{\text{ext}} = \mathbf{F}^{\text{ext}}(t_{n+1}) - \mathbf{F}^{\text{ext}}(t_n)$ and attempt to find that displacement increment, $\mathbf{d}_{n+1} - \mathbf{d}_n$, that will restore equilibrium (i.e., result in satisfaction of (3.58)). This will require a nonlinear equation solving technique for determination of \mathbf{d}_{n+1} , a topic that will be discussed further in Nonlinear Equation Solving.

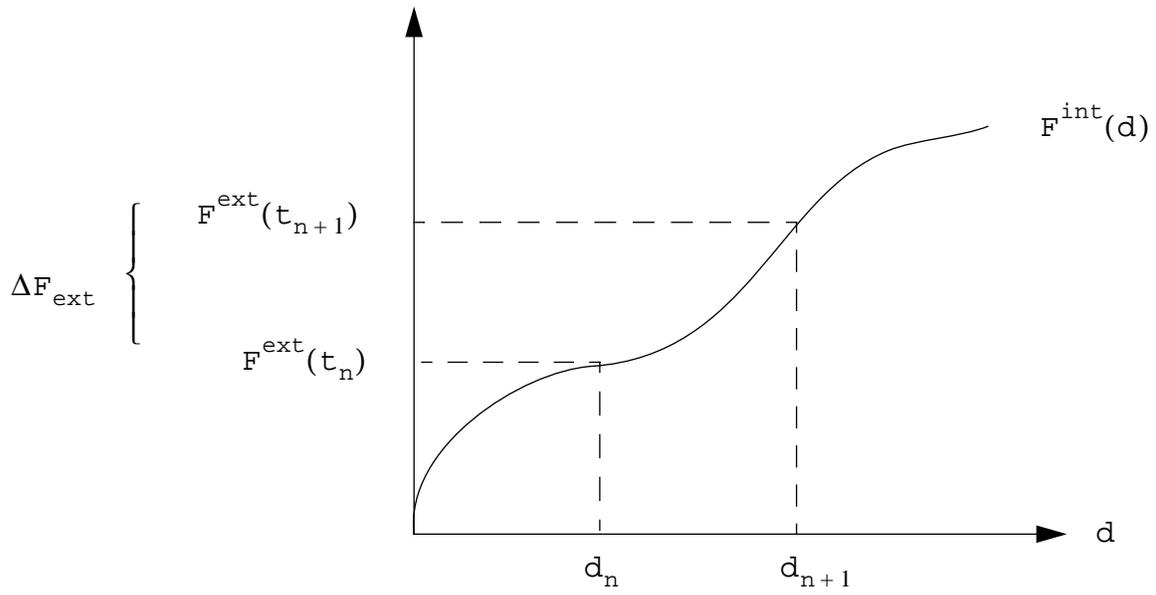


Figure 3.4 Simple illustration of the incremental load approach to quasistatic problems.

Dynamics

Introduction

We now restore the inertial terms to the discrete equation system and examine prospective techniques for solution. To recap the key result of Generation of Matrix Equations, the problem we consider now takes the form

$$\mathbf{M}\ddot{\mathbf{d}}(\tau) + \mathbf{F}^{\text{int}}(\mathbf{d}(\tau)) = \mathbf{F}^{\text{ext}}, \quad (3.59)$$

to be solved for $\tau \in [0, T]$, subject to the initial conditions

$$\mathbf{d}(0) = \mathbf{d}_0 \quad (3.60)$$

and

$$\dot{\mathbf{d}}(0) = \mathbf{v}_0. \quad (3.61)$$

The Semidiscrete Approach

It might be noted from Eq. (3.59) that time remains continuous in our formulation at this point, whereas the spatial discretization has already been achieved by the finite element interpolations summarized in Discretization. This type of finite element approach to transient problems is sometimes referred to as the semidiscrete finite element method, since the approximation in space is performed first, leaving a set of equations discrete in space but still continuous in time. To complete the approximation, a finite differencing procedure is generally applied in time, as discussed below.

Time-Stepping Procedures

As discussed in Quasistatics, we subdivide the time interval of interest $[0, T]$ via

$$[0, T] = \bigcup_{n=0}^{N-1} [\tau_n, \tau_{n+1}] \quad (3.62)$$

and consider the following generic problem. Given algorithmic approximations for the solution vector (\mathbf{d}_n), velocity (\mathbf{v}_n), and acceleration (\mathbf{a}_n) at time τ_n , find approximations \mathbf{d}_{n+1} , \mathbf{v}_{n+1} and \mathbf{a}_{n+1} for these quantities at time τ_{n+1} . Note that in contrast to the quasistatic problem, the variable τ here does have the interpretation of actual time.

Several time-stepping algorithms have been proposed for this incremental problem we have posed. So that we might have a template with which to work, we will consider perhaps the most pervasive of these schemes: the Newmark family of temporal integrators ([Newmark, N.M., 1959]). This algorithm can be summarized in a time step $[t_n, t_{n+1}]$ as follows:

$$\begin{aligned} \mathbf{M}\mathbf{a}_{n+1} + \mathbf{F}^{\text{int}}(\mathbf{d}_{n+1}) &= \mathbf{F}^{\text{ext}}(t_{n+1}) \\ \mathbf{d}_{n+1} &= \mathbf{d}_n + \Delta t \mathbf{v}_n + \frac{\Delta t^2}{2} [(1 - 2\beta)\mathbf{a}_n + 2\beta\mathbf{a}_{n+1}], \\ \mathbf{v}_{n+1} &= \mathbf{v}_n + \Delta t [(1 - \gamma)\mathbf{a}_n + \gamma\mathbf{a}_{n+1}] \end{aligned} \quad (3.63)$$

where β and γ are algorithmic parameters that define the stability and accuracy characteristics of the method.

Although, obviously, a wide range of algorithms exist corresponding to the different available choices of β and γ , two algorithms in particular are prevalent in common use:

1. Central differences ($\beta = 0, \gamma = \frac{1}{2}$). This integrator is second order accurate and only conditionally stable, meaning that linearized stability is only retained when Δt is less than some critical value. This algorithm is an example of an explicit finite element integrator, to be discussed in Explicit Finite Element Methods.
2. Trapezoidal rule ($\beta = \frac{1}{4}, \gamma = \frac{1}{2}$). This integrator is second-order accurate and unconditionally stable for linear problems, meaning that the spectral radii of the integrator remain less than 1 in modulus for any time step Δt (in linear problems). This algorithm is an example of an implicit finite element integrator, to be discussed in Implicit Finite Element Methods.

Explicit Finite Element Methods

Examining the central differences algorithm as an example, let us take $\beta = 0, \gamma = \frac{1}{2}$ and substitute into Eq. (3.63). Upon doing so, we obtain the following algorithm:

$$\begin{aligned} \mathbf{a}_{n+1} &= \mathbf{M}^{-1}(\mathbf{F}^{\text{ext}}(t_{n+1}) - \mathbf{F}^{\text{int}}(\mathbf{d}_{n+1})) \\ \mathbf{d}_{n+1} &= \mathbf{d}_n + \Delta t \mathbf{v}_n + \frac{\Delta t^2}{2} \mathbf{a}_n, \\ \mathbf{v}_{n+1} &= \mathbf{v}_n + \frac{\Delta t}{2} [\mathbf{a}_n + \mathbf{a}_{n+1}] \end{aligned} \quad (3.64)$$

where the first equation has been written as solved for \mathbf{a}_{n+1} .

Equation (3.64) can be used to explain why this formulation is termed explicit. Consider the case where \mathbf{M} is a diagonal matrix. This is not, in general, the case if we strictly follow the variational formulation; reference to Eq. (3.31) will verify that unless two shape functions N_A and N_B are mutually orthogonal, the mass matrix will not, in general, be diagonal. However, it is common practice, as will be discussed in Basics of Element Design, to diagonalize the mass matrix. In the event that this is done, Eq. (3.64) shows that given the three vectors $\{\mathbf{a}_n, \mathbf{v}_n, \mathbf{d}_n\}$, the data at t_{n+1} , $\{\mathbf{a}_{n+1}, \mathbf{v}_{n+1}, \mathbf{d}_{n+1}\}$ can be computed explicitly (i.e., without the need for solution of coupled equations).

Although this form of the central difference formulation is readily obtained from the Newmark formulae, it does not give insight into the source of the “central difference” terminology and, in fact, does not represent the manner in which the integrator is ordinarily implemented. To see the usual form, let us define the following auxiliary algorithmic velocity vector:

$$\mathbf{v}_{n+\frac{1}{2}} = \mathbf{v}_n + \frac{1}{2}\Delta t \mathbf{a}_n, \quad (3.65)$$

which also implies a corresponding relation for the previous time step:

$$\mathbf{v}_{n-\frac{1}{2}} = \mathbf{v}_{n-1} + \frac{1}{2}\Delta t \mathbf{a}_{n-1}. \quad (3.66)$$

Subtracting Eq. (3.66) from (3.65) gives

$$\mathbf{v}_{n+\frac{1}{2}} - \mathbf{v}_{n-\frac{1}{2}} = \mathbf{v}_n - \mathbf{v}_{n-1} + \frac{1}{2}\Delta t (\mathbf{a}_n - \mathbf{a}_{n-1}). \quad (3.67)$$

However, evaluation of (3.64) during the time step $[t_{n-1}, t_n]$ reveals that

$$\mathbf{v}_n - \mathbf{v}_{n-1} = \frac{1}{2}\Delta t (\mathbf{a}_n + \mathbf{a}_{n-1}), \quad (3.68)$$

so that upon substitution into (3.67) we find

$$\mathbf{v}_{n+\frac{1}{2}} - \mathbf{v}_{n-\frac{1}{2}} = \Delta t \mathbf{a}_n. \quad (3.69)$$

Furthermore, substitution of (3.65) into the second equation of (3.64) gives

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \Delta t \mathbf{v}_{n+\frac{1}{2}}. \quad (3.70)$$

Thus by collecting these latest two results, together with the equilibrium equation evaluated at t_n , we can reexpress the algorithm completely equivalently as

$$\begin{aligned} \mathbf{a}_n &= \mathbf{M}^{-1}(\mathbf{F}^{\text{ext}}(t_n) - \mathbf{F}^{\text{int}}(\mathbf{d}_n)) \\ \mathbf{v}_{n+\frac{1}{2}} &= \mathbf{v}_{n-\frac{1}{2}} + \Delta t \mathbf{a}_n \\ \mathbf{d}_{n+1} &= \mathbf{d}_n + \Delta t \mathbf{v}_{n+\frac{1}{2}} \end{aligned} \quad (3.71)$$

Note that the velocity and displacement updates emanate from centered difference approximations to the acceleration \mathbf{a}_n and velocity $\mathbf{v}_{n+\frac{1}{2}}$, respectively, giving the algorithm its

name. The velocity measures that are utilized by the algorithm are shifted by a half step from the time values at which the acceleration and configuration are monitored.

As mentioned above, explicit finite element schemes are only conditionally stable, meaning that they only remain stable when the time increment Δt is less than some critical limit. This limit, sometimes called the Courant stability limit, can be shown to be as follows

$$\Delta t \leq \frac{2}{\omega}, \quad (3.72)$$

where ω is the highest modal natural frequency in the mesh. It can also be shown that this frequency can be conservatively estimated via

$$\omega \approx 2 \left(\frac{c}{h} \right)_{\max}, \quad (3.73)$$

where c and h are the sound speed and characteristic mesh size, respectively, associated with the element in the mesh having the largest ratio of these two quantities. Combining Eqs. (3.72) and (3.73) we find that

$$\Delta t \leq \left(\frac{h}{c} \right)_{\min}. \quad (3.74)$$

In other words, the time step may be no larger than the amount of time required for a sound wave to traverse the element in the mesh having the smallest transit time. This fact tells us immediately that explicit finite element methods are most appropriate for those problems featuring very high frequency response or wave-like phenomena. For problems featuring low frequency response, literally thousands of time steps may be required to resolve even a single period of vibration due to the stringent stability limit posed by (3.74). For such problems an unconditionally stable algorithm is highly desirable, albeit at the cost of explicit updates in each increment.

Implicit Finite Element Methods

To introduce the concept of an implicit finite element method, we examine the trapezoidal rule, which is simply that member of the Newmark family obtained by setting $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$.

Substitution of these values into Eq. (3.63) yields

$$\begin{aligned}\mathbf{M}\mathbf{a}_{n+1} + \mathbf{F}^{\text{int}}(\mathbf{d}_{n+1}) &= \mathbf{F}^{\text{ext}}(t_{n+1}) \\ \mathbf{d}_{n+1} &= \mathbf{d}_n + \Delta t \mathbf{v}_n + \frac{\Delta t^2}{4} [\mathbf{a}_n + \mathbf{a}_{n+1}]. \\ \mathbf{v}_{n+1} &= \mathbf{v}_n + \frac{\Delta t}{2} [\mathbf{a}_n + \mathbf{a}_{n+1}]\end{aligned}\tag{3.75}$$

Insight into this method can be obtained by combining the first two equations in (3.75) and solving for \mathbf{d}_{n+1} . Doing so gives

$$\begin{aligned}\begin{bmatrix} \frac{4}{\Delta t^2} \mathbf{M} \mathbf{d}_{n+1} \\ + \mathbf{F}^{\text{int}}(\mathbf{d}_{n+1}) \end{bmatrix} &= \begin{bmatrix} \mathbf{F}^{\text{ext}}(t_{n+1}) \\ + \mathbf{M} \left(\mathbf{a}_n + \Delta t \mathbf{v}_n + \frac{4}{\Delta t^2} \mathbf{d}_n \right) \end{bmatrix} \\ \mathbf{a}_{n+1} &= \frac{4}{\Delta t^2} (\mathbf{d}_{n+1} - \mathbf{d}_n) \left(-\frac{4}{\Delta t} \mathbf{v}_n - \mathbf{a}_n \right) \\ \mathbf{v}_{n+1} &= \mathbf{v}_n + \frac{\Delta t}{2} [\mathbf{a}_n + \mathbf{a}_{n+1}]\end{aligned}\tag{3.76}$$

Clearly solving the first equation in (3.76) is the most expensive procedure involved in updating the solution from t_n to t_{n+1} . This equation is not only fully coupled, but also is highly nonlinear, in general, due to the internal force vector. In fact, we could write the first equation of (3.76) in terms of a dynamic incremental residual \mathbf{R}_{n+1} via

$$\begin{aligned}\mathbf{R}(\mathbf{d}_{n+1}) &= \begin{bmatrix} \mathbf{F}^{\text{ext}}(t_{n+1}) + \mathbf{M} \left(\mathbf{a}_n + \Delta t \mathbf{v}_n + \frac{4}{\Delta t^2} \mathbf{d}_n \right) \\ - \left(\frac{4}{\Delta t^2} \mathbf{M} \mathbf{d}_{n+1} + \mathbf{F}^{\text{int}}(\mathbf{d}_{n+1}) \right) \end{bmatrix} \\ &= 0\end{aligned}\tag{3.77}$$

This system has the same form as (3.57), which suggests that the same sort of nonlinear solution strategies are needed for implicit dynamic calculations as in Quasistatics. Some common equation-solving alternatives are discussed in Nonlinear Equation Solving.

Nonlinear Equation Solving

Introduction

In this section we explore some of the alternatives available for solving the nonlinear discrete equations associated with computation of an unknown state at t_{n+1} , in either the context of a quasistatic problem (i.e., Eq. (3.57)) or an implicit dynamic formulation (Eq. (3.77)). In either case, the equation to be solved takes the form

$$\mathbf{R}(\mathbf{d}_{n+1}) = 0, \quad (3.78)$$

where the residual $\mathbf{R}(\mathbf{d}_{n+1})$ is considered to be a nonlinear function of the solution vector \mathbf{d}_{n+1} .

Newton Raphson Framework

We now return to the general concept of a Newton-Raphson iterative solution technique, as discussed in the one-dimensional context in Material Nonlinearity. To review, a Newton-Raphson solution technique for (3.78) is defined in iteration i by

$$\mathbf{R}(\mathbf{d}_{n+1}^i) + \left[\frac{\partial \mathbf{R}}{\partial \mathbf{d}} \right]_{\mathbf{d}_{n+1}^i} \Delta \mathbf{d} = 0, \quad (3.79)$$

followed by the update

$$\mathbf{d}_{n+1}^{i+1} = \mathbf{d}_{n+1}^i + \Delta \mathbf{d}. \quad (3.80)$$

Iterations on i typically continue until the Euclidean norm $\|\mathbf{R}(\mathbf{d}_{n+1}^i)\|$ is less than some tolerance; $\|\Delta \mathbf{d}\|$ is smaller than some tolerance, the quantity $\mathbf{R}(\mathbf{d}_{n+1}^i) \cdot \Delta \mathbf{d}$ is small, or some combination of these three conditions.

It is instructive to examine the form taken by Eq. (3.79) for the quasistatic and implicit dynamic cases. For the quasistatic case $\mathbf{R}(\mathbf{d}_{n+1}^i)$ takes the form

$$\mathbf{R}(\mathbf{d}_{n+1}^i) = \mathbf{F}^{\text{ext}}(t_{n+1}) - \mathbf{F}^{\text{int}}(\mathbf{d}_{n+1}^i), \quad (3.81)$$

so that Eq. (3.79) can be rewritten as

$$\mathbf{K}(\mathbf{d}_{n+1}^i) \Delta \mathbf{d} = \mathbf{R}(\mathbf{d}_{n+1}^i), \quad (3.82)$$

where the incremental stiffness matrix $\mathbf{K}(\mathbf{d}_{n+1}^i)$ is given by

$$\mathbf{K}\mathbf{d}_{n+1}^i = -\left[\frac{\partial \mathbf{R}}{\partial \mathbf{d}}\right]_{\mathbf{d}_{n+1}^i} = \left[\frac{\partial \mathbf{F}^{\text{int}}}{\partial \mathbf{d}}\right]_{\mathbf{d}_{n+1}^i}. \quad (3.83)$$

Thus application of the Newton-Raphson method to quasistatic problems amounts to solution of successive linear problems, as defined by (3.82).

In the implicit dynamics case, let us consider the trapezoidal rule as a template. In this case, the residual is of the form

$$\mathbf{R}(\mathbf{d}_{n+1}^i) = \left[\begin{array}{c} \mathbf{F}^{\text{ext}}(t_{n+1}) + \mathbf{M}\left(\mathbf{a}_n + \Delta t \mathbf{v}_n + \frac{4}{\Delta t^2} \mathbf{d}_n\right) \\ -\left(\frac{4}{\Delta t^2} \mathbf{M}\mathbf{d}_{n+1}^i + \mathbf{F}^{\text{int}}(\mathbf{d}_{n+1}^i)\right) \end{array} \right] = 0. \quad (3.84)$$

This causes (3.81) to take the form

$$\left[\frac{4}{\Delta t^2} \mathbf{M} + \mathbf{K}(\mathbf{d}_{n+1}^i) \right] \Delta \mathbf{d} = \mathbf{R}(\mathbf{d}_{n+1}^i), \quad (3.85)$$

where the stiffness matrix $\mathbf{K}(\mathbf{d}_{n+1}^i)$ is as given in (3.83). In either case solution of the global incremental equations will require the assembly of the coefficient matrix on the left-hand side. Following the same assembly procedures outlined in Localization and Assembly, this matrix is given as an assembly of element stiffness matrices $\mathbf{k}^e(\mathbf{d}_{n+1}^i)$, each of which can be expressed generally as

$$\mathbf{k}^e(\mathbf{d}_{n+1}^i) = [k_{pq}^e(\mathbf{d}_{n+1}^i)], \quad (3.86)$$

where

$$k_{pq}^e(\mathbf{d}_{n+1}^i) = \frac{\partial f_p^{\text{int}}}{\partial d_q^e}(\mathbf{d}_{n+1}^i), \quad (3.87)$$

where f_p^{int} is as given in Eq. (3.44).

We can, therefore, conclude that for a Newton-Raphson treatment of either a quasistatic or implicit dynamic system, an important function of the element subroutine is to return an element stiffness in addition to the internal force vector and mass matrix that may also be required. We will discuss in detail the mechanics of this operation in Basics of Element Design.

Line Search

It is noteworthy that the Newton-Raphson method is only guaranteed to be convergent in an asymptotic sense, subject also to some smoothness and differentiability conditions. This means that solution updates may not be effective if one is excessively far from the solution or if significant nonsmoothnesses are present in the equation system. Indeed, in many problems the early displacement updates in a given load increment given by (3.82) or (3.85) may actually be counterproductive in that they take one farther from the solution rather than closer. It is, therefore, imperative to have a technique that controls the manner in which the solution is sought such that bad displacement updates, as predicted by the linearized kernel, are not allowed to carry one too far from the desired solution.

The concept of line search, pervasive in nonlinear equation solving, is employed for this purpose. To motivate the concept, we consider the case of a so-called quadratic system, where the total system energy $\Pi(\mathbf{d})$ can be expressed as a quadratic function of the solution vector \mathbf{d} via

$$\Pi(\mathbf{d}) = \frac{1}{2} \mathbf{d}^T \mathbf{K} \mathbf{d} - \mathbf{F}^{\text{ext}T} \mathbf{d}, \quad (3.88)$$

where for simplicity we assume \mathbf{K} and \mathbf{F}^{ext} to be constant. We seek the minimizer of $\Pi(\mathbf{d})$, which, of course, can be equivalently expressed as the solution of

$$\mathbf{K} \mathbf{d} = \mathbf{F}^{\text{ext}}. \quad (3.89)$$

In fact, this problem statement is a finite-dimensional analogue of that discussed in Weak Forms, where it was asserted that the linearly elastic boundary problem can be solved by finding the displacement field, minimizing the total potential energy of the system. We could, therefore, think of the problem we have posed as a finite element discretization of such a system. Although the system we consider here is quasistatic, the technique we motivate is utilized for solution of nonlinear dynamic systems as well.

Neglecting the fact that this problem could be solved via Gaussian elimination, we consider a generic iterative procedure for solving it. Suppose we have a current iterate \mathbf{d}^i as well as a proposed displacement increment $\Delta \mathbf{d}$. In a Newton-Raphson method $\Delta \mathbf{d}$ would be computed in a given iteration by solving (3.82), which would produce the exact solution (to machine precision) after the displacement update (3.80). If $\Delta \mathbf{d}$ is not such a good choice for the displacement increment, however, we would like a method for detecting this fact and for controlling growth in the residual. In this discussion we take $\Delta \mathbf{d}$ as a search direction in n_{eq} -space and look for solutions in this direction reducing (or at least controlling growth in) the residual.

Given \mathbf{d}^i and $\Delta \mathbf{d}$, then, we introduce a search parameter s and consider an update of the form:

$$\mathbf{d}^{i+1}(\mathfrak{s}) = \mathbf{d}^i + \mathfrak{s}\Delta\mathbf{d}. \quad (3.90)$$

The line search parameter is chosen such that the update produced by (3.90) is in some sense optimal. In this spirit we choose \mathfrak{s} as the minimizer of:

$$\hat{\Pi}(\mathfrak{s}) = \begin{bmatrix} \frac{1}{2}(\mathbf{d}^i + \mathfrak{s}\Delta\mathbf{d})^T \mathbf{K}(\mathbf{d}^i + \mathfrak{s}\Delta\mathbf{d}) \\ -(\mathbf{d}^i + \mathfrak{s}\Delta\mathbf{d})^T \mathbf{F}^{\text{ext}} \end{bmatrix}, \quad (3.91)$$

which can be found by finding the solution of $\frac{\partial}{\partial \mathfrak{s}} \hat{\Pi}(\mathfrak{s}) = 0$. If we assume that \mathbf{K} is symmetric, positive definite, one finds in taking this derivative that \mathfrak{s} is given as the solution of

$$\Delta\mathbf{d}^T (\mathbf{K}(\mathbf{d}^i + \mathfrak{s}\Delta\mathbf{d}) - \mathbf{F}^{\text{ext}}) = 0. \quad (3.92)$$

Two forms of this equation are useful under various circumstances. First, in the linear system we now consider, (3.92) is readily solved to explicitly yield \mathfrak{s} :

$$\mathfrak{s} = \frac{\Delta\mathbf{d}^T \mathbf{R}^i}{\Delta\mathbf{d}^T \mathbf{K} \Delta\mathbf{d}}, \quad (3.93)$$

where $\mathbf{R}^i = \mathbf{F}^{\text{ext}} - \mathbf{K}\mathbf{d}^i$. This form of the line search is actually used in some implementations but depends, strictly speaking, on linearity to be effective. Thus a more generally used form is generated by reexpressing (3.92) as the following problem:

Find \mathfrak{s} such that

$$\Delta\mathbf{d}^T \mathbf{R}(\mathbf{d}^i + \mathfrak{s}\Delta\mathbf{d}) = 0. \quad (3.94)$$

In (3.94), $\mathbf{R}(\mathbf{d}^i + \mathfrak{s}\Delta\mathbf{d}) = \mathbf{F}^{\text{ext}} - \mathbf{K}(\mathbf{d}^i + \mathfrak{s}\Delta\mathbf{d})$ in the linear case. The advantage of Eq. (3.94), however, is that it admits more general representations of the residual; for a nonlinear quasistatic problem, we can use (3.94) with $\mathbf{R}(\mathbf{d}^i + \mathfrak{s}\Delta\mathbf{d})$ given by

$$\mathbf{R}(\mathbf{d}^i + \mathfrak{s}\Delta\mathbf{d}) = \mathbf{F}^{\text{ext}} - \mathbf{F}^{\text{int}}(\mathbf{d}^i + \mathfrak{s}\Delta\mathbf{d}). \quad (3.95)$$

Similar generalizations for the dynamic case are, of course, also possible with the dynamic residual given for the trapezoidal rule in (3.84).

From (3.95) we conclude that a line search procedure looks for updated iterates where the search direction is orthogonal to the residual. This is equivalent to an energy minimization in the linear case, whereas the interpretation in the nonlinear case is not quite so straightforward.

Furthermore, in the nonlinear case it is not efficient or even necessary to find the root of (3.94) to

machine precision. More commonly one uses some sort of root finder to find an s between 0 and 1 that satisfies (3.94) to some tolerance. Making the definition

$$G(s) = \Delta \mathbf{d}^T \mathbf{R}(\mathbf{d}^i + s\Delta \mathbf{d}), \quad (3.96)$$

a typical algorithm to find s could be outlined as:

Given \mathbf{d}^i and $\Delta \mathbf{d}$

- IF ($|G(1)| > \text{TOL} \times |G(0)|$ or $G(1) \times G(0) < 0$) THEN

$$\text{Iterate for } s \in (0, 1] \text{ such that } |G(s)| < (\text{TOL})|G(0)| \quad (3.97)$$

- ELSE

$$s = 1. \quad (3.98)$$

- ENDIF

The check in the IF statement amounts to checking whether a full step (with $s = 1$) leads to an unreasonably large increase in G and whether a root might reasonably be expected in the interval $(0, 1]$.

Quasi-Newton Methods

One can establish that Newton-Raphson iteration is quadratically convergent asymptotically, meaning that the error associated with a given iteration tends to the square of the previous iteration's error as iterations proceed. One can, in fact, roughly see the reason for this fact from Eq. (1.21), which states that the Newton-Raphson update is motivated by a first order Taylor series expansion of the residual about the current solution vector iterate. We might, therefore, expect that the error incurred from this approximate update should tend toward the square of the displacement update as iterations proceed, which is, indeed, the case. A much more rigorous derivation of this property can be found in [Kelley, C.T., 1995]. This quadratically convergent behavior is a highly desirable property and makes the Newton-Raphson method much more rapidly convergent than many other equation-solving alternatives.

However, it is also true to say that a full Newton-Raphson method can be tremendously expensive due to the necessity of solving the successive fully coupled linear systems implied by (3.82) and (3.85). If we choose to use a direct equation-solving technique for solving these systems, such as Gaussian elimination, the cost of solving each linearized problem will vary as the cube of the number of equations (n_{eq}). For very large problems this cost can become prohibitive.

In response to this situation, a number of methods, known collectively as quasi-Newton (alternatively, secant) methods, have been developed. These methods replace most of the

Newton-Raphson iterations with a cheaper update to the solution vector, sacrificing convergence performance but making the average equilibrium iteration much less expensive. The interested reader should consult [Dennis, J.E. and Schnabel, R.B., 1996] or [Kelley, C.T., 1995] for excellent overviews of these methods from a generic, nonlinear equation-solving viewpoint.

Before discussing specific quasi-Newton methods, let us motivate them through consideration of the term “secant method”. Suppose we have a scalar-valued, nonlinear equation

$$\mathbf{R}(\mathbf{d}) = 0 \quad (3.99)$$

and wish to employ an iterative method to obtain the root. If we are currently performing iteration i and wish to obtain the next iterate $i + 1$, a secant method will do this by replacing

the tangent to the curve, $\mathbf{K}_i = \frac{d}{d\mathbf{d}} [\mathbf{R}(\mathbf{d}_i)]$, with the secant $\tilde{\mathbf{K}}_i = \frac{\mathbf{R}(\mathbf{d}_i) - \mathbf{R}(\mathbf{d}_{i-1})}{\mathbf{d}_i - \mathbf{d}_{i-1}}$ in the

Newton-Raphson updating scheme (see Figure 3.5). Thus, the next iterate is obtained via

$$\mathbf{d}_{i+1} = \mathbf{d}_i - \frac{\mathbf{R}(\mathbf{d}_i) - \mathbf{R}(\mathbf{d}_{i-1})}{\mathbf{d}_i - \mathbf{d}_{i-1}} \mathbf{R}(\mathbf{d}_i). \quad (3.100)$$

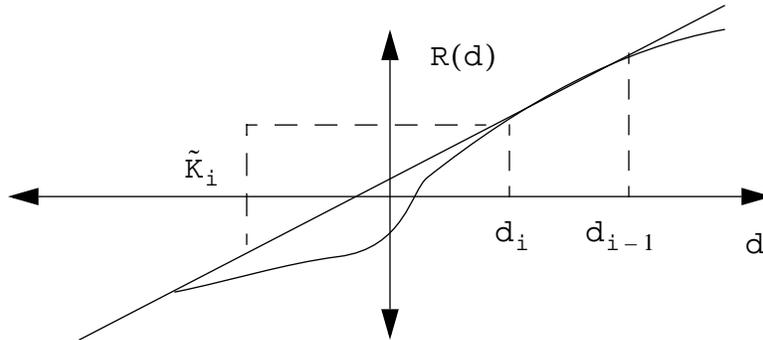


Figure 3.5 One-dimensional illustration of quasi-Newton (secant) iteration.

In generalizing this concept to multiple directions, we seek to approximate so-called consistent tangents needed by Newton-Raphson updates (see (3.83)) with stiffnesses that will be cheaper to compute and invert. In a secant method, using the one-dimensional example as motivation, we demand that these approximate tangents obey the so-called quasi-Newton equation:

$$\tilde{\mathbf{K}}_i(\mathbf{d}_i - \mathbf{d}_{i-1}) = \mathbf{R}(\mathbf{d}_i) - \mathbf{R}(\mathbf{d}_{i-1}), \quad (3.101)$$

or in terms of the inverse:

$$\mathbf{d}_i - \mathbf{d}_{i-1} = \tilde{\mathbf{K}}_i^{-1}(\mathbf{R}(\mathbf{d}_i) - \mathbf{R}(\mathbf{d}_{i-1})). \quad (3.102)$$

In the one-dimensional case either expression implies uniquely the secant method already discussed but with multiple unknowns, expressions (3.101) or (3.102) place a less stringent restriction. There are, therefore, a multitude of such methods, collectively termed quasi-Newton

or secant methods, whose defining feature is the satisfaction of (3.101) and (3.102). Here we concentrate on one particular method, the BFGS (Broyden-Fletcher-Goldfarb-Shanno) method, proposed originally and most coherently for specific use in finite element calculations by [Matthies, H. and Strang, G., 1979].

In the BFGS method, one typically starts with an assembled Newton-Raphson tangent given, for example, by (3.87). One performs one iteration with this tangent (probably including a line search). Rather than repeating this procedure for subsequent iterations, the BFGS method takes the tangent from the Newton-Raphson iteration and updates it in a manner consistent with (3.102) and uses this tangent to compute the next iterate for the solution (also probably including a line search in the update).

The BFGS method is effective in many circumstances because the update to the tangent matrix is inexpensive and is actually done to a previously determined inverse so that no matrix inversion is necessary in most equilibrium iterations. To be more specific, we suppose that the last tangent utilized in an iteration process is $\tilde{\mathbf{K}}_{i-1}$. The BFGS update is defined as

$$\tilde{\mathbf{K}}_i^{-1} = (\mathbf{I} + \mathbf{v}_i \mathbf{w}_i^T) \tilde{\mathbf{K}}_{i-1}^{-1} (\mathbf{I} + \mathbf{w}_i \mathbf{v}_i^T), \quad (3.103)$$

where

$$\mathbf{v}_i = \frac{\Delta \mathbf{d}_{i-1}}{\Delta \mathbf{d}_{i-1} \cdot (\mathbf{R}(\mathbf{d}_i) - \mathbf{R}(\mathbf{d}_{i-1}))}, \quad (3.104)$$

$$\mathbf{w}_i = -(\mathbf{R}(\mathbf{d}_i) - \mathbf{R}(\mathbf{d}_{i-1})) + \alpha_i \mathbf{R}(\mathbf{d}_{i-1}), \quad (3.105)$$

where

$$\alpha_i = \left[\frac{-s_{i-1} (\mathbf{R}(\mathbf{d}_i) - \mathbf{R}(\mathbf{d}_{i-1})) \cdot \Delta \mathbf{d}_{i-1}}{\mathbf{R}(\mathbf{d}_{i-1}) \cdot \Delta \mathbf{d}_{i-1}} \right]^{\frac{1}{2}}. \quad (3.106)$$

Throughout the above equations we have indexed the search directions and line search parameters such that

$$\mathbf{d}_i = \mathbf{d}_{i-1} + s_{i-1} \Delta \mathbf{d}_{i-1}. \quad (3.107)$$

The next search direction $\Delta \mathbf{d}_i$ is then computed via

$$\begin{aligned} \Delta \mathbf{d}_i &= \tilde{\mathbf{K}}_i^{-1} \mathbf{R}(\mathbf{d}_i) \\ &= (\mathbf{I} + \mathbf{v}_i \mathbf{w}_i^T) \tilde{\mathbf{K}}_{i-1}^{-1} (\mathbf{I} + \mathbf{w}_i \mathbf{v}_i^T) \mathbf{R}(\mathbf{d}_i). \end{aligned} \quad (3.108)$$

One may discern from (3.108) why a BFGS iteration is so much cheaper than a Newton-Raphson iteration. Keeping in mind that $\tilde{\mathbf{K}}_{i-1}^{-1}$ is typically stored in practice as a factorized stiffness

matrix, one can compute $\Delta \mathbf{d}_i$ efficiently by proceeding right to left in the second line of (3.108). Thinking in this manner, the update consists only of dot products, scalar vector multiplies, and a backsolve procedure.

The BFGS method is, like other quasi-Newton methods, superlinear in convergence rate, meaning that the error decreases in a manner faster than linear but not as fast as the quadratic rate displayed by Newton-Raphson. Thus it is most effectively used in large problems, where this disadvantage in convergence behavior is offset by its great savings in the iteration process. It should also be noted that the success of BFGS depends critically upon the incorporation of line search. Since the iterations are based less directly on the underlying mechanics of the system than they would be in Newton-Raphson, it is particularly important that the line search prevent excessive excursions away from the solution in the case of bad search directions. Typically, BFGS solvers also contain provisions to compute new Newton-Raphson tangents to restart the iteration process in the event the BFGS iterations are ineffective.

Conjugate Gradient Methods

The desire to solve very large problems has recently led researchers to consider so-called indirect iterative, or matrix-free strategies, where the set of nonlinear equations is iteratively solved without any need to compute, store, or invert a tangent matrix at any stage of the iteration process. Perhaps the most celebrated iterative technique for solving linear equations in the last two to three decades has been the conjugate gradient method. In this section we briefly consider its extension to matrix-free, nonlinear equation solving. To do so, however, it is useful to examine the linear case first, from which the nonlinear algorithms are readily derived.

We begin then by considering the same linear system and potential energy function Π given in (3.88) and consider that this potential energy is to be minimized. We note that at any prospective solution point \mathbf{d} , the steepest descent direction of the objective function Π is given by the negative of the gradient:

$$-\frac{\partial}{\partial \mathbf{d}}(\Pi) = \mathbf{F}^{\text{ext}} - \mathbf{K}\mathbf{d} = \mathbf{R}(\mathbf{d}). \quad (3.109)$$

In other words, the steepest descent direction at any prospective solution \mathbf{d} is merely given by the residual $\mathbf{R}(\mathbf{d})$. One of the most elementary methods from nonlinear equation solving/optimization, the steepest descent method, utilizes at each iteration the current steepest descent direction as the search direction, with a subsequent line search defining the update to the solution vector. This method, while intuitive, is not as effective as other alternatives including, in particular, the conjugate gradient strategy we now discuss.

The difficulty with steepest descent is that in many cases, successive search directions “zig-zag,” meaning that a given search direction will contain significant components in the direction of previous steps. It is readily imagined that such repetition is wasteful, since presumably these

earlier iterations have already eliminated, or at least markedly reduced, the error in their respective search directions.

This discussion is made more quantitative by recalling the formula for a generic line search, Eq. (3.94). In the case of a linear problem, this condition takes the form

$$\begin{aligned}\Delta\mathbf{d}^T\mathbf{R}(\mathbf{d}^i + s\Delta\mathbf{d}) &= \Delta\mathbf{d}^T(\mathbf{F}^{\text{ext}} - \mathbf{K}(\mathbf{d}^i + s\Delta\mathbf{d})) \\ &= \Delta\mathbf{d}^T\mathbf{K}(\mathbf{d} - (\mathbf{d}^i + s\Delta\mathbf{d})) = 0\end{aligned}\quad (3.110)$$

In (3.110) the term $\mathbf{d} - (\mathbf{d}^i + s\Delta\mathbf{d})$ is recognized as the error associated with the next iterate $\mathbf{d}^i + s\Delta\mathbf{d}$. Thus we can see from (3.110) that the line search criterion causes the error associated with the next iterate for the solution vector to be \mathbf{K} -orthogonal to the search direction. Thus if we consider the solution space to be described by a sequence of vector spaces of increasing dimension, where search directions form the basis and the stiffness matrix \mathbf{K} serves as a metric, then each iteration with line search removes all error in that direction. It is, therefore, wasteful to have subsequent search directions that have non-zero components in previous directions. The aim of the conjugate gradient method is to orthogonalize this direction as iterations proceed.

To begin derivation of the method, let us use the notations \mathbf{p} for a search direction and α for a line search parameter, as opposed to the $\Delta\mathbf{d}$ and s used previously (in large part to conform to usage in the literature). The restriction we will place on the search directions is that they will be \mathbf{K} -orthogonal:

$$\mathbf{p}_i^T\mathbf{K}\mathbf{p}_j = 0, \quad (3.111)$$

where i and j are indices for two different iterations. In each iteration a line search will be performed eliminating all error in the current search direction via

$$\mathbf{d}_{i+1} = \mathbf{d}_i + \alpha_i\mathbf{p}_i, \quad (3.112)$$

where α_i is given by (cf. (3.93)):

$$\alpha_i = \frac{\mathbf{p}_i^T\mathbf{R}(\mathbf{d}_i)}{\mathbf{p}_i^T\mathbf{K}\mathbf{p}_i}. \quad (3.113)$$

Since the method operates by eliminating all error in each successive search direction, which is orthogonal to all previous directions, this method will yield the exact solution in n_{eq} iterations in perfect arithmetic. In this sense conjugate gradients can be viewed as a direct method, although in practice iterations are terminated far before this point, making the method approximate.

Yet to be discussed is the generation of the orthogonal search directions. This is done by applying a Gram-Schmidt orthogonalization procedure to a set of linearly independent vectors, taken in the case of conjugate gradients as residual vectors. The process is begun by taking the original search direction as the residual, which would be the same direction taken by steepest descent:

$$\mathbf{p}_0 = \mathbf{R}(\mathbf{d}_0). \quad (3.114)$$

Subsequent search directions are defined via

$$\mathbf{p}_i = \mathbf{R}(\mathbf{d}_i) + \sum_{k=0}^{i-1} \beta_{ik} \mathbf{p}_k, \quad (3.115)$$

where the coefficients β_{ik} must be found to ensure \mathbf{K} -orthogonality of \mathbf{p}_i with previous search directions. This calculation can be done by taking the \mathbf{K} -inner product of (3.115) with \mathbf{p}_j :

$$\begin{aligned} 0 &= \mathbf{p}_i^T \mathbf{K} \mathbf{p}_j \\ &= \mathbf{R}(\mathbf{d}_i)^T \mathbf{K} \mathbf{p}_j + \sum_{k=0}^{i-1} \beta_{ik} \mathbf{p}_k^T \mathbf{K} \mathbf{p}_j. \\ &= \mathbf{R}(\mathbf{d}_i)^T \mathbf{K} \mathbf{p}_j + \beta_{ij} \mathbf{p}_j^T \mathbf{K} \mathbf{p}_j \end{aligned} \quad (3.116)$$

Eq. (3.116) then allows us to write a formula for β_{ij}

$$\beta_{ij} = -\frac{\mathbf{R}(\mathbf{d}_i)^T \mathbf{K} \mathbf{p}_j}{\mathbf{p}_j^T \mathbf{K} \mathbf{p}_j}. \quad (3.117)$$

At first glance it would appear that to find search direction \mathbf{p}_i , all β_{ij} would need to be calculated for $j < i$. However, we can apply further reasoning to see that, in fact, only one of the coefficients required by (3.115) is nonzero. To begin, since the error in a given iteration is \mathbf{K} -orthogonal to earlier search directions, we can write

$$\mathbf{p}_j^T \mathbf{K} \mathbf{e}_i = 0, \quad j = 0, \dots, i-1, \quad (3.118)$$

where $\mathbf{e}_i = \mathbf{d}_i - \mathbf{d}$. Since $\mathbf{K} \mathbf{e}_i = -\mathbf{R}(\mathbf{d}_i)$, it follows that

$$\mathbf{p}_j^T \mathbf{R}(\mathbf{d}_i) = 0, \quad j = 0, \dots, i-1. \quad (3.119)$$

We can further conclude that since the search direction in a given iteration is a linear combination of all the residuals that preceded it,

$$\mathbf{R}(\mathbf{d}_j)^T \mathbf{R}(\mathbf{d}_i) = 0, j = 0, \dots, i-1. \quad (3.120)$$

We can write the residual at the $j+1$ iteration as

$$\begin{aligned} \mathbf{R}(\mathbf{d}_{j+1}) &= -\mathbf{K}\mathbf{e}_{j+1} = -\mathbf{K}(\mathbf{e}_j + \alpha_j \mathbf{p}_j) \\ &= (\mathbf{R}(\mathbf{d}_j) - \alpha_j \mathbf{K}\mathbf{p}_j). \end{aligned} \quad (3.121)$$

Taking the inner product of this equation with $\mathbf{R}(\mathbf{d}_i)$, we find

$$\alpha_j \mathbf{R}(\mathbf{d}_i)^T \mathbf{K}\mathbf{p}_j = \mathbf{R}(\mathbf{d}_i)^T \mathbf{R}(\mathbf{d}_j) - \mathbf{R}(\mathbf{d}_i)^T \mathbf{R}(\mathbf{d}_{j+1}). \quad (3.122)$$

Using now the orthogonality property (3.120), we find

$$\mathbf{R}(\mathbf{d}_i)^T \mathbf{K}\mathbf{p}_j = \begin{cases} -\frac{1}{\alpha_{i-1}} \mathbf{R}(\mathbf{d}_i)^T \mathbf{R}(\mathbf{d}_i), & j = i-1 \\ 0, & j = 0, \dots, i-2 \end{cases}. \quad (3.123)$$

Thus examining (3.115) and (3.117), we find that only one of the needed coefficients is non-zero – namely, $\beta_{i,i-1}$. Naming this quantity β_i subsequently we can rewrite (3.115) as

$$\mathbf{p}_i = \mathbf{R}(\mathbf{d}_i) + \beta_i \mathbf{p}_{i-1}, \quad (3.124)$$

where

$$\beta_i = \frac{1}{\alpha_{i-1}} \frac{\mathbf{R}(\mathbf{d}_i)^T \mathbf{R}(\mathbf{d}_i)}{\mathbf{p}_{i-1}^T \mathbf{K}\mathbf{p}_{i-1}}. \quad (3.125)$$

Eq (3.125) can be written even more simply by noting that

$$\begin{aligned} \alpha_{i-1} \mathbf{p}_{i-1}^T \mathbf{K}\mathbf{p}_{i-1} &= \mathbf{p}_{i-1}^T \mathbf{K}(\mathbf{d}_i - \mathbf{d}_{i-1}) \\ &= \mathbf{p}_{i-1}^T (\mathbf{R}(\mathbf{d}_{i-1}) - \mathbf{R}(\mathbf{d}_i)) \\ &= \mathbf{p}_{i-1}^T \mathbf{R}(\mathbf{d}_{i-1}) \\ &= (\mathbf{R}(\mathbf{d}_{i-1}) + \beta \mathbf{p}_{i-2})^T \mathbf{R}(\mathbf{d}_{i-1}) \\ &= \mathbf{R}(\mathbf{d}_{i-1})^T \mathbf{R}(\mathbf{d}_{i-1}) \end{aligned} \quad (3.126)$$

Thus,

$$\beta_i = \frac{\mathbf{R}(\mathbf{d}_i)^T \mathbf{R}(\mathbf{d}_i)}{\mathbf{R}(\mathbf{d}_{i-1})^T \mathbf{R}(\mathbf{d}_{i-1})}. \quad (3.127)$$

Eq. (3.127) is the Fletcher-Reeves version of the orthogonalization [Fletcher, R. and Reeves, C.M., 1964]. Another commonly used form, due to Polak and Ribiere, is trivially obtained from the orthogonality property as

$$\beta_i = \frac{\mathbf{R}(\mathbf{d}_i)^T(\mathbf{R}(\mathbf{d}_i) - \mathbf{R}(\mathbf{d}_{i-1}))}{\mathbf{R}(\mathbf{d}_{i-1})^T \mathbf{R}(\mathbf{d}_{i-1})}. \quad (3.128)$$

Collecting all of these results, the conjugate gradient algorithm for linear problems can be summarized as

$$\begin{aligned} \mathbf{p}_0 &= \mathbf{R}(\mathbf{d}_0) \\ \alpha_i &= \frac{\mathbf{p}_i^T \mathbf{R}(\mathbf{d}_i)}{\mathbf{p}_i^T \mathbf{K} \mathbf{p}_i} = \frac{\mathbf{R}(\mathbf{d}_i)^T \mathbf{R}(\mathbf{d}_i)}{\mathbf{p}_i^T \mathbf{K} \mathbf{p}_i} \\ \mathbf{d}_{i+1} &= \mathbf{d}_i + \alpha_i \mathbf{p}_i \\ \mathbf{R}(\mathbf{d}_{i+1}) &= \mathbf{R}(\mathbf{d}_i) - \alpha_i \mathbf{K} \mathbf{p}_i \\ \beta_{i+1} &= \begin{cases} \frac{\mathbf{R}(\mathbf{d}_{i+1})^T \mathbf{R}(\mathbf{d}_{i+1})}{\mathbf{R}(\mathbf{d}_i)^T \mathbf{R}(\mathbf{d}_i)} \text{ (Fletcher-Reeves)} \\ \frac{\mathbf{R}(\mathbf{d}_{i+1})^T (\mathbf{R}(\mathbf{d}_{i+1}) - \mathbf{R}(\mathbf{d}_i))}{\mathbf{R}(\mathbf{d}_i)^T \mathbf{R}(\mathbf{d}_i)} \text{ (Polak-Ribiere)} \end{cases} \\ \mathbf{p}_{i+1} &= \mathbf{R}(\mathbf{d}_{i+1}) + \beta_{i+1} \mathbf{p}_i \end{aligned} \quad (3.129)$$

With our derivation of the linear conjugate gradient method now complete, we return to the actual problem of interest here, which is nonlinear conjugate gradients. At this point we should distinguish between two alternatives. One could still adopt a Newton-Raphson nonlinear equation-solving strategy and employ a conjugate gradient-type algorithm to solve the linear system of equations. This type of algorithm is sometimes termed a Newton-iterative method (see [Kelley, C.T., 1995]). One should note, however, that use of this method still requires formation of the stiffness matrix, so that even if equation-solving savings are realized by using CG over a direct solver, the memory requirements of the method will tend to be extensive. Also if determination of the global Jacobian matrix is difficult, expensive, or impossible, then this method will be limited just as its more traditional ancestors are.

Thus one is led to consider the possibility of using conjugate gradient iterations themselves as the nonlinear solution iterates. The algorithms of this type commonly used look remarkably like the linear algorithm summarized by (3.129). We summarize here perhaps the most successful of these, termed the Polak-Ribiere algorithm after the form of β , used in the (now approximate) orthogonalization:

$$\begin{aligned}
\mathbf{p}_0 &= \mathbf{R}(\mathbf{d}_0) = \mathbf{F}^{\text{ext}} - \mathbf{F}^{\text{int}}(\mathbf{d}_0) \\
\alpha_i &\text{ obtained from nonlinear line search} \\
\mathbf{d}_{i+1} &= \mathbf{d}_i + \alpha_i \mathbf{p}_i \\
\mathbf{R}(\mathbf{d}_{i+1}) &= \mathbf{F}^{\text{ext}} - \mathbf{F}^{\text{int}}(\mathbf{d}_{i+1}) \\
\beta_{i+1} &= \frac{\mathbf{R}(\mathbf{d}_{i+1})^T (\mathbf{R}(\mathbf{d}_{i+1}) - \mathbf{R}(\mathbf{d}_i))}{\mathbf{R}(\mathbf{d}_i)^T \mathbf{R}(\mathbf{d}_i)} \\
\mathbf{p}_{i+1} &= \mathbf{R}(\mathbf{d}_{i+1}) + \beta_{i+1} \mathbf{p}_i
\end{aligned} \tag{3.130}$$

Really only one difference from the linear algorithm is obvious. The line search in (3.130) must now be given by an expression appropriate for nonlinear problems. Although several alternatives might be devised, one of the simplest (attributed to [Bartels, R. and Daniel, J.W., 1973]) is generated by considering only the first Newton-Raphson iterate (with initial guess $\alpha_i = 0$) to the line search equation

$$\mathbf{p}_i^T (\mathbf{R}(\mathbf{d}_i + \alpha_i \mathbf{p}_i)) = 0, \tag{3.131}$$

which would lead one to consider

$$\alpha_i = \frac{\mathbf{p}_i^T \mathbf{R}(\mathbf{d}_i)}{\mathbf{p}_i^T \mathbf{K} \mathbf{p}_i}, \tag{3.132}$$

where \mathbf{K} is in practice a diagonalized estimate of the tangent (perhaps a secant) evaluated at the last iteration.

Preconditioning

It is widely recognized that conjugate gradient methods are best behaved when the condition number of the underlying stiffness matrix is small (i.e., when the eigenvalues are clustered together). Due to this fact conjugate gradient algorithms are almost never applied without preconditioning, a term referring to the act of converting an equation system to one having the same solution while possessing a tighter clustering of eigenvalues.

Thinking somewhat simplistically about this idea, we might conceive that the ultimate in a well-conditioned coefficient matrix would be the identity matrix, which has all eigenvalues equal to one. Relying once more on a linear system to motivate our ideas, let us consider that our generic linear system

$$\mathbf{Kd} = \mathbf{F}^{\text{ext}} \tag{3.133}$$

is ill-conditioned and that we wish to employ a well-behaved CG algorithm to solve it. If we devise a matrix, \mathbf{M} , that we feel to be a good approximation to the inverse of \mathbf{K} , we might choose to iteratively solve the equation

$$\mathbf{MKd} = \mathbf{MF}^{\text{ext}}, \quad (3.134)$$

since the matrix \mathbf{MK} should have a tight cluster of eigenvalues about one (if \mathbf{M} is, indeed, a good approximation of \mathbf{K}^{-1}).

However, CG is only applicable to symmetric systems, and \mathbf{MK} is not necessarily symmetric. If the \mathbf{M} we select is symmetric positive-definite, we can find a matrix, \mathbf{E} , such that

$$\mathbf{EE}^T = \mathbf{M}^{-1} \quad (3.135)$$

and consider solution of the system

$$\mathbf{E}^{-1}\mathbf{KE}^{-T}\widehat{\mathbf{d}} = \mathbf{E}^{-1}\mathbf{F}^{\text{ext}}, \quad (3.136)$$

where $\widehat{\mathbf{d}} = \mathbf{E}^T\mathbf{d}$. It is to be noted that this system also will be satisfied by the solution of (3.133), while remaining symmetric and positive-definite. If we straightforwardly apply the linear CG algorithm (3.129) to (3.135), assuming the Polak-Ribiere form, we obtain

$$\begin{aligned} \widehat{\mathbf{p}}_0 &= \widehat{\mathbf{R}}(\widehat{\mathbf{d}}_0) = \mathbf{E}^{-1}\mathbf{F}^{\text{ext}} - \mathbf{E}^{-1}\mathbf{KE}^{-T}\widehat{\mathbf{d}}_0 \\ \alpha_i &= \frac{\widehat{\mathbf{R}}(\widehat{\mathbf{d}}_i)^T \widehat{\mathbf{R}}(\widehat{\mathbf{d}}_i)}{\widehat{\mathbf{p}}_i^T \mathbf{E}^{-1}\mathbf{KE}^{-T}\widehat{\mathbf{p}}_i} \\ \widehat{\mathbf{d}}_{i+1} &= \widehat{\mathbf{d}}_i + \alpha_i \widehat{\mathbf{p}}_i \\ \widehat{\mathbf{R}}(\widehat{\mathbf{d}}_{i+1}) &= \widehat{\mathbf{R}}(\widehat{\mathbf{d}}_i) - \alpha_i \mathbf{EKE}^T \widehat{\mathbf{p}}_i \\ \beta_{i+1} &= \frac{\widehat{\mathbf{R}}(\widehat{\mathbf{d}}_{i+1})^T (\widehat{\mathbf{R}}(\widehat{\mathbf{d}}_{i+1}) - \widehat{\mathbf{R}}(\widehat{\mathbf{d}}_i))}{\widehat{\mathbf{R}}(\widehat{\mathbf{d}}_i)^T \widehat{\mathbf{R}}(\widehat{\mathbf{d}}_i)} \\ \widehat{\mathbf{p}}_{i+1} &= \widehat{\mathbf{R}}(\widehat{\mathbf{d}}_{i+1}) + \beta_{i+1} \widehat{\mathbf{p}}_i \end{aligned} \quad (3.137)$$

Algorithm (3.137) can be written in a form not explicitly involving the matrix \mathbf{E} or the supplementary vectors $\widehat{\mathbf{R}}$, $\widehat{\mathbf{p}}$, and $\widehat{\mathbf{d}}$ by noting that $\widehat{\mathbf{R}}(\widehat{\mathbf{d}}_i) = \mathbf{E}^{-1}\mathbf{R}(\mathbf{d}_i)$, $\mathbf{M} = \mathbf{E}^{-T}\mathbf{E}^{-1}$, and by taking $\widehat{\mathbf{p}}_i = \mathbf{E}^T\mathbf{p}_i$. We therefore write the Preconditioned Conjugate Gradient algorithm for linear systems as:

$$\begin{aligned}
\mathbf{R}(\mathbf{d}_0) &= \mathbf{F}^{\text{ext}} - \mathbf{K}\mathbf{d}_0 \\
\mathbf{p}_0 &= \mathbf{M}\mathbf{R}(\mathbf{d}_0) \\
\alpha_i &= \frac{\mathbf{R}(\mathbf{d}_i)^\top \mathbf{M}\mathbf{R}(\mathbf{d}_i)}{\mathbf{p}_i^\top \mathbf{K}\mathbf{p}_i} \\
\mathbf{d}_{i+1} &= \mathbf{d}_i + \alpha_i \mathbf{p}_i \\
\mathbf{R}(\mathbf{d}_{i+1}) &= \mathbf{R}(\mathbf{d}_i) - \alpha_i \mathbf{K}\mathbf{p}_i \\
\beta_{i+1} &= \frac{\mathbf{R}(\mathbf{d}_{i+1})^\top \mathbf{M}(\mathbf{R}(\mathbf{d}_{i+1}) - \mathbf{R}(\mathbf{d}_i))}{\mathbf{R}(\mathbf{d}_i)^\top \mathbf{M}\mathbf{R}(\mathbf{d}_i)} \\
\mathbf{p}_{i+1} &= \mathbf{M}\mathbf{R}(\mathbf{d}_{i+1}) + \beta_{i+1} \mathbf{p}_i
\end{aligned} \tag{3.138}$$

Finally, we are in a position to discuss a preconditioned matrix-free conjugate gradient structure for a fully nonlinear system. We might summarize such an algorithm as

$$\begin{aligned}
\mathbf{R}(\mathbf{d}_0) &= \mathbf{F}^{\text{ext}} - \mathbf{F}^{\text{int}}(\mathbf{d}_0) \\
\mathbf{p}_0 &= \mathbf{M}\mathbf{R}(\mathbf{d}_0) \\
\alpha_i &\text{ obtained from nonlinear line search} \\
\mathbf{d}_{i+1} &= \mathbf{d}_i + \alpha_i \mathbf{p}_i \\
\mathbf{R}(\mathbf{d}_{i+1}) &= \mathbf{F}^{\text{ext}} - \mathbf{F}^{\text{int}}(\mathbf{d}_{i+1}) \\
\beta_{i+1} &= \frac{\mathbf{R}(\mathbf{d}_{i+1})^\top \mathbf{M}(\mathbf{R}(\mathbf{d}_{i+1}) - \mathbf{R}(\mathbf{d}_i))}{\mathbf{R}(\mathbf{d}_i)^\top \mathbf{M}\mathbf{R}(\mathbf{d}_i)} \\
\mathbf{p}_{i+1} &= \mathbf{M}\mathbf{R}(\mathbf{d}_{i+1}) + \beta_{i+1} \mathbf{p}_i
\end{aligned} \tag{3.139}$$

Following the lead of the last section, one alternative for the line search would be the first iteration for a Newton-Raphson strategy to find α_i :

$$\alpha_i = \frac{\mathbf{p}_i^\top \mathbf{M}^{-1} \mathbf{R}(\mathbf{d}_i)}{\mathbf{p}_i^\top \mathbf{K}\mathbf{p}_i} \tag{3.140}$$

To conclude, it should be remarked that the simplest choice for preconditioning, Jacobi preconditioning, is accomplished when the matrix \mathbf{M} is chosen as diagonal. One choice might be to take \mathbf{M} to be the inverse of the diagonal of the stiffness matrix.

Basics of Element Design

Introduction

In this section we explore the basic issues associated with the design of finite elements, which are the building blocks of the methods we have discussed. In particular we will discuss how definitions and manipulations are done at the local level to produce the elemental quantities like \mathbf{m}^e , \mathbf{f}^{int^e} , and \mathbf{k}^e that are needed for assembly of the global equations of motion. We concentrate in this section on one field problems (i.e., where only the deformation mapping ϕ_t is discretized). It will turn out that many nonlinear solid mechanics applications of interest, including nearly incompressible elasticity and metal plasticity, require more sophisticated approximations in which other variables (like pressure) must be explicitly included in the formulation. Discussion of such advanced methods is deferred to Advanced Element Design Issues.

Convergence

Before introducing in detail the manipulations necessary at the element level, it is worthwhile to discuss in general terms the general requirements usually placed upon shape function definitions. It should be noted that these conditions are sufficient but not necessary, so that many formulations exist that violate one or more of them. However, it is also fair to say that most finite elements in wide use satisfy the conditions we will now place. The discussion we give now is brief and rather qualitative; the interested reader should consult [Hughes, T.J.R., 1987] or [Strang, G. and Fix, G.J., 1973] for more technical discussions of these points (notably within the context of linear problems).

We begin by defining m , which will denote the highest order of shape function spatial derivative present in the expression for the stiffness matrix. For the class of problems we have considered in this text, we recall from Newton Raphson Framework that the element stiffness matrix takes the form

$$k_{pq}^e(\mathbf{d}_{n+1}^{e^i}) = \frac{\partial f_p^{int}}{\partial d_q^e}(\mathbf{d}_{n+1}^{e^i}). \quad (3.141)$$

The internal force vector required in (3.141) was given generically in Localization and Assembly as

$$\mathbf{f}^{\text{int}}|_p^e = \int_{\phi_t^h(\Omega^e)} \left[\sum_{j=1}^3 N_{a,j}(\phi_t^{-1}(\mathbf{x})) \mathbf{T}_{i,j}^h \right] d\mathbf{v}. \quad (3.142)$$

Performing the differentiation indicated in (3.141) will produce no higher than first-order derivatives of the shape functions; therefore, $m = 1$.

The three general convergence requirements we wish to mention are as follows.

- The global shape functions N_A should have global continuity of the order $m - 1$. In mathematical terms they should be C^{m-1} on Ω^h .
- The restriction of the global shape functions to individual elements (i.e., the $\{N_a\}$) should be C^m on element interiors.
- The elemental shape functions $\{N_a\}$ should be complete.

The first two of these requirements are fairly simple to understand. The first, the C^{m-1} continuity requirement, simply means that all derivatives up to $m - 1$ of the shape functions should not undergo jumps as element boundaries are crossed. In the current case this means that all N_A should be C^0 . Since the approximation to the configuration mapping ϕ_t^h is a linear combination of these shape functions, we see that the physical restriction placed by this condition amounts to no more than a requirement that the displacement be single-valued throughout the domain (i.e., gaps and interpenetrations at element boundaries may not occur).

The second requirement on element interiors simply states that the shape functions should be sufficiently smooth so that the element stiffness expression is integrable. Physically speaking, the first derivatives of the configuration mapping produces strain measures, so we simply require that the strains be well-behaved on element interiors by this restriction. Note that global smoothness of the strains (and, therefore, stresses) is not required. This point is of some importance in the reporting of results.

The third requirement, the completeness requirement, is somewhat more involved to explain and yet corresponds fairly directly to physical ideas. We say that a given element is complete when setting the element degrees of freedom according to a given low-order polynomial forces the solution (in this case ϕ_t^h) to be interpolated according to the same polynomial pointwise within the element. The degree of polynomials for which we place this requirement is referred to as the degree of completeness for the element.

In the current case where we deal with solid continua, the usual degree of completeness demanded is 1. This means that all polynomials, up to and including order 1, should be exactly interpolated by the element. It is worthwhile to consider an example of this point. Suppose we are in three dimensions, and set the element degrees of freedom via

$$\mathbf{d}_a^e = c_0 + c_1 X_a^e \mathbf{e}_x + c_2 Y_a^e \mathbf{e}_y + c_3 Z_a^e \mathbf{e}_z, \quad (3.143)$$

where $c_0 - c_3$ are arbitrary constants and X_a^e, Y_a^e, Z_a^e are the (reference) coordinates for local node number a . The completeness condition requires that

$$\begin{aligned} \varphi_t^h(\mathbf{x}^e) &= \sum_{a=1}^{nen} N_a(\mathbf{x}^e) \mathbf{d}_a^e \\ &= (c_0 + c_1 X^e \mathbf{e}_x + c_2 Y^e \mathbf{e}_y + c_3 Z^e \mathbf{e}_z) \end{aligned} \quad (3.144)$$

hold for all $\mathbf{x}^e \in \Omega^e$ and for all values of the arbitrary constants.

As mentioned above, this requirement has a physical interpretation as well. In solid mechanics we have already pointed out that the first spatial derivatives of the displacements produce strains. Since we require that an element be able to reproduce arbitrary linear polynomials, this also implies that any state where the first derivatives (i.e., strains) are constant should be exactly representable. Thus a complete element should be able to exactly represent any uniform strain state. A practical way to test for this condition is to impose a boundary value problem on an arbitrary patch of elements having a constant strain (and thus stress) solution and then to demand exactness of the numerical solution. Such a test is called a “patch test” and has become one of the standard benchmarks by which any new proposed element formulation is tested and evaluated.

Parameterization

With the three criteria in hand for element definitions, we proceed to define a recipe through which element definitions and manipulations can be systematically performed. The most basic definition to be made toward this end is the concept of the local (or parent) parameterization of an element. In effect we seek to define a local coordinate system that will be the same for every element in a problem, contributing in great part to the modularity we will desire for element level operations.

We will denote a vector of these local variables by \mathbf{r} , with \mathbf{r} being a 2-vector in two dimensions and a 3-vector in three dimensions. Specifically we define \mathbf{r} as

$$\mathbf{r} = \begin{cases} \begin{bmatrix} r \\ s \end{bmatrix} & \text{(two dimensions)} \\ \begin{bmatrix} r \\ s \\ t \end{bmatrix} & \text{(three dimensions)} \end{cases} . \quad (3.145)$$

The local variables r , s , and t are all assumed to range between -1 and 1, so that the domain definition is likewise standardized among all elements in a given problem. The domain of \mathbf{r} is often referred to as the parent domain. In two dimensions it is a biunit square, and in three dimensions it is a biunit cube (see Figure 3.6).

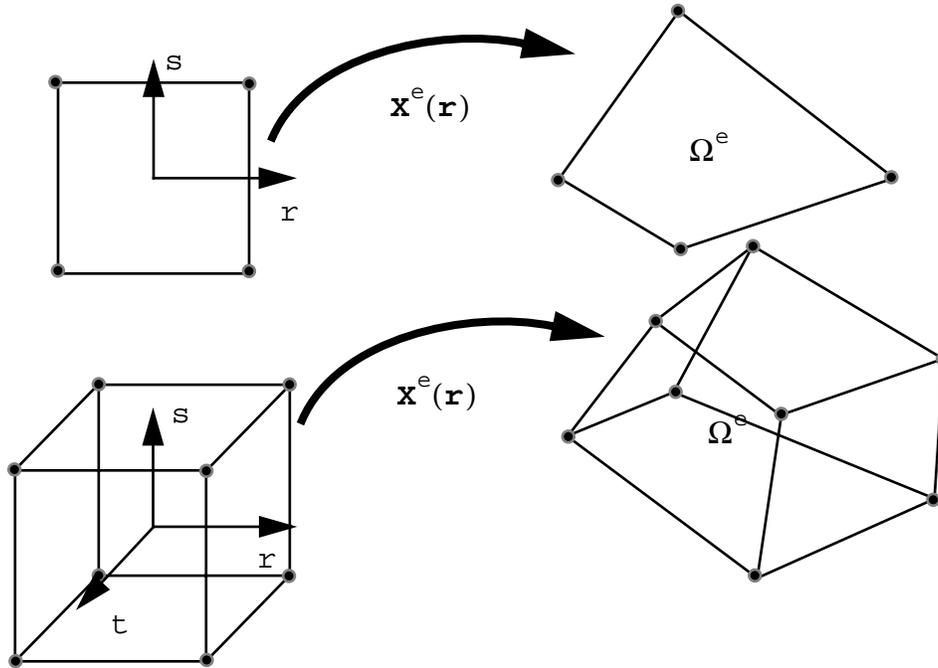


Figure 3.6 Local parameterizations and coordinate mappings in two and three dimensions.

Of course, for this alternative element coordinate system to be of any use, its relationship with the global coordinate system must be defined. This is accomplished through a shape function expansion via

$$\mathbf{x}^e(\mathbf{r}) = \sum_{a=1}^{n_{en}} \tilde{N}_a(\mathbf{r}) \mathbf{x}_a^e, \quad (3.146)$$

where \mathbf{x}^e is the global (reference) coordinate mapping covering element e and where \mathbf{x}_a^e are the element nodal (reference) coordinates, as before. Note also that in (3.146) the shape functions have been written using the parent coordinates as the independent variables. This is the reason for the superposed tilde on the shape function. One could think of \mathbf{r} as an material point label within the element, so that \mathbf{x}^e and \mathbf{r} are two reference coordinate systems for the element that are related according to (3.146).

The most important generic class of finite elements is comprised of so-called isoparametric elements. Such elements are defined by utilizing the same shape functions for definition of $\phi_t^h(\mathbf{x}^e)$ (see Eq. (3.144)) as for the element coordinates \mathbf{x}^e (as in (3.146)). One can show (and

the reader should) that providing all element shape functions sum to one at any point in the element, an isoparametric element automatically satisfies the completeness condition outlined in Convergence. Thus provided we choose shape functions that sum to one, are suitably smooth on element interiors, and match neighboring element descriptions on element boundaries, the convergence criteria are automatically satisfied. We will concentrate on the mechanics of shape function definition in the next section.

In the meantime let us consider the implications of the isoparametric approach for the Lagrangian description of large deformation solid mechanics we now consider. So that we may distinguish carefully between mappings taking \mathbf{r} as an argument and between those taking \mathbf{x} , we will use superposed tildes for the former (as in Eq. (3.146)). If an element is isoparametric, then by definition the configuration mapping over an element is given by

$$\tilde{\varphi}_t^h(\mathbf{r}) = \sum_{a=1}^{nen} \tilde{N}_a(\mathbf{r}) \mathbf{d}_a^e, \quad (3.147)$$

where the shape functions $\tilde{N}_a(\mathbf{r})$ are exactly the same as in (3.146). However, it should also be the case that the function $\tilde{\varphi}_t^h(\mathbf{r})$ should be attainable from the composition of $\varphi_t^h(\mathbf{x}^e)$ (defined according to (3.144)) with $\mathbf{x}^e(\mathbf{r})$ (defined according to (3.146)). Thus we can write

$$\sum_{a=1}^{nen} \tilde{N}_a(\mathbf{r}) \mathbf{d}_a^e = \tilde{\varphi}_t^h(\mathbf{r}) = \sum_{a=1}^{nen} N_a(\mathbf{x}^e(\mathbf{r})) \mathbf{d}_a^e. \quad (3.148)$$

Comparing the leftmost and rightmost expressions of (3.148) and realizing that the equality must hold for any given combination of the element degrees of freedom \mathbf{d}_a^e , we are led to conclude that the alternative shape function expressions $\tilde{N}_a(\mathbf{r})$ and $N_a(\mathbf{x}^e)$ must be related by mere composition via

$$\tilde{N}_a = N_a \circ \mathbf{X}^e. \quad (3.149)$$

Thus we have the option of defining the shape functions over whatever domain is convenient, and since the parent domain is the one that is standardized, we typically begin with an expression for \tilde{N}_a and then derive the implied expression for N_a according to

$$N_a = \tilde{N}_a \circ \mathbf{X}^{e^{-1}}. \quad (3.150)$$

Equation (3.150) has important implications in practice for, in general, we have no guarantee that the inverse mapping $\mathbf{x}^{e^{-1}}$ of \mathbf{x}^e is well behaved, which it must be for the shape functions N_a to make sense. Fortunately, according to the implicit function theorem, the inverse function to

(3.146) is smooth and one-to-one, provided the Jacobian of the indicated transformation is nonzero. This essentially amounts to a geometric restriction on elements in the reference domain. In two dimensions, using a four-noded element, the implication is that all interior angles in each element must be less than 180° (see Figure 3.7).

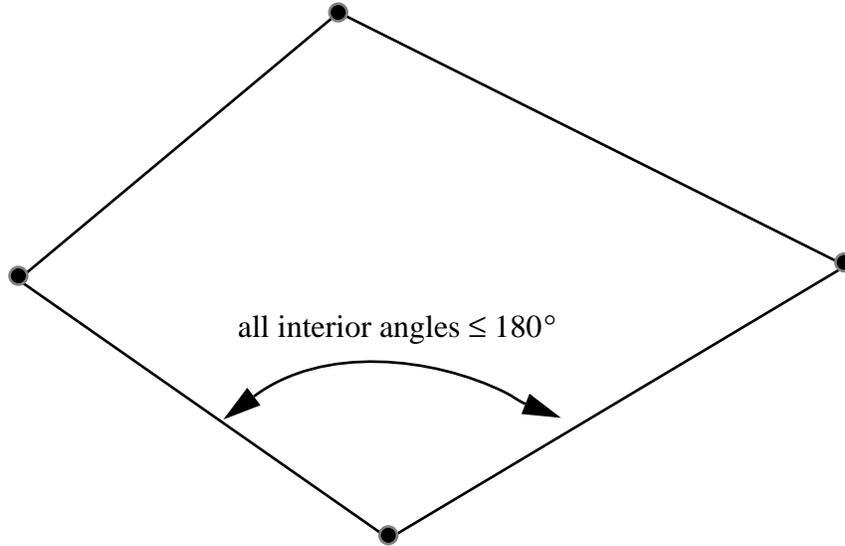


Figure 3.7 Geometric restrictions on a four-noded element to retain well-posedness of the coordinate and configuration mappings.

Finally, let us introduce the notation \widehat{N}_a for shape functions that take the current coordinates $\mathbf{x}^e = \boldsymbol{\varphi}_t^h(\mathbf{X}^e)$ as arguments. Such an expression is needed – for example, in Eq. (3.142) (note the abuse in notation) – where the spatial derivatives $\widehat{N}_{a,i} = \frac{\partial}{\partial x_i^e} \widehat{N}_a$ must be computed.

Following similar reasoning to the above, one can conclude that the functions \widehat{N}_a must obey

$$N_a = \widehat{N}_a \circ \tilde{\boldsymbol{\varphi}}_t^{h^{-1}}. \quad (3.151)$$

Again for the needed function $\tilde{\boldsymbol{\varphi}}_t^{h^{-1}}$ to be well-behaved, the Jacobian of the transformation (3.147) must be nonzero. This restriction amounts to:

$$\det \left[\frac{\partial \tilde{\boldsymbol{\varphi}}_t^h}{\partial \mathbf{r}} \right] = \det \left[\frac{\partial \tilde{\boldsymbol{\varphi}}_t^h}{\partial \mathbf{x}^e} \right] \det \left[\frac{\partial \mathbf{x}^e}{\partial \mathbf{r}} \right] \neq 0. \quad (3.152)$$

Provided the original element definitions are not overly distorted, the second term on the right-hand side of (3.152) will be nonzero. Thus the well-posedness of the spatial shape functions \widehat{N}_a

requires that $\det \left[\frac{\partial \tilde{\phi}_t^h}{\partial \mathbf{x}^e} \right]$ be nonzero. The reader will recognize this as the approximated

determinant of the deformation gradient \mathcal{J} , as defined in Measures of Deformation. According to Eq. (2.10) \mathcal{J} must be positive pointwise for the concept of volume change to have any physical meaning. Thus provided the approximated deformation mapping remains kinematically admissible (i.e., $\mathcal{J} > 0$), the spatially defined shape functions are guaranteed to be well-behaved.

With these arguments as background, we now turn our attention to definition of the N_a , according to the parent domain. To keep notational complexity to a minimum, we will drop the explicit distinction between N_a , \tilde{N}_a , and \widehat{N}_a , referring to all these objects as simply N_a in the sequel.

Shape Functions

Most continuum-based finite elements rely on Lagrange polynomials for their shape function definitions. Beginning with expressions appropriate for one-dimensional domains, let us suppose that we have a one-dimensional element with nen nodes, which are equally spaced over the $[-1, 1]$ domain. Use of $nen - 1$ order Lagrange polynomials $L_a^{nen-1}(r)$ for definition of each of the element shape functions N_a leads to

$$N_a(r) = L_a^{nen-1}(r) = \frac{\prod_{b=1, b \neq a}^{nen} (r - r_b)}{\prod_{b=1, b \neq a}^{nen} (r_a - r_b)}, \quad (3.153)$$

where the r_b refer to the local (parent) coordinates of the individual element nodes. The reader may care to verify that these shape functions have two useful properties. First, that

$$N_a(r_b) = \delta_{ab} \quad (3.154)$$

and second, that

$$\sum_{a=1}^{nen} N_a(r) = 1 \text{ for all } r \in [-1, 1]. \quad (3.155)$$

Equation (3.155) is noteworthy in that it provides an important ingredient of the completeness argument (see Parameterization), whereas Eq. (3.154) ensures that the nodal degrees of freedom have the actual interpretation that

$$\mathbf{d}_a^e = \boldsymbol{\varphi}_t^h(\mathbf{x}_a^e) = \mathbf{x}_a^e \quad (3.156)$$

(see Eq. (3.144)).

Before proceeding to the more interesting multidimensional case, some examples may be useful. For $nen = 2$ the element coordinates are $r_1 = -1$ and $r_2 = 1$. The corresponding element shape functions are computed from (3.153) as $N_1(r) = \frac{1}{2}(1 - r)$ and $N_2(r) = \frac{1}{2}(1 + r)$, thereby providing the basis for the one-dimensional linear finite element. For $nen = 3$ the local nodal coordinates are $r_1 = -1$, $r_2 = 0$, and $r_3 = 1$, with the shape functions turning out to be $N_1(r) = \frac{1}{2}r(r - 1)$, $N_2(r) = 1 - r^2$, and $N_3(r) = \frac{1}{2}r(r + 1)$, thereby defining the one-dimensional quadratic element. These shape functions are plotted in Figure 3.8.

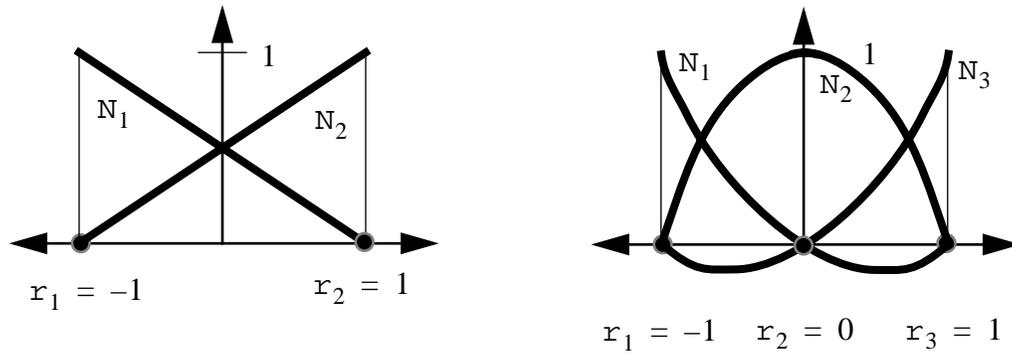


Figure 3.8 Low-order, one-dimensional Lagrange shape functions.

Next we turn to the generalization of these concepts to two and three dimensions. This can be accomplished by building up “products” of one-dimensional shape functions, as indicated schematically for the four-noded quadrilateral element depicted in Figure 3.9. In general, let us suppose that local Node a in a two-dimensional element has local coordinates (r_c, s_d) , where indices c and d refer to the node number in the r and s directions, respectively. The two-dimensional shape function is given by

$$N_a(r, s) = L_c^{nen_r - 1}(r) \times L_d^{nen_s - 1}(s), \quad (3.157)$$

where nen_r and nen_s are the numbers of nodes in the r and s directions, respectively.

Taking the four-noded quadrilateral depicted in Figure 3.9 as an example, the shape functions N_a are found to be $N_1 = \frac{1}{4}(1-r)(1-s)$, $N_2 = \frac{1}{4}(1+r)(1-s)$, $N_3 = \frac{1}{4}(1+r)(1+s)$, and $N_4 = \frac{1}{4}(1-r)(1+s)$.

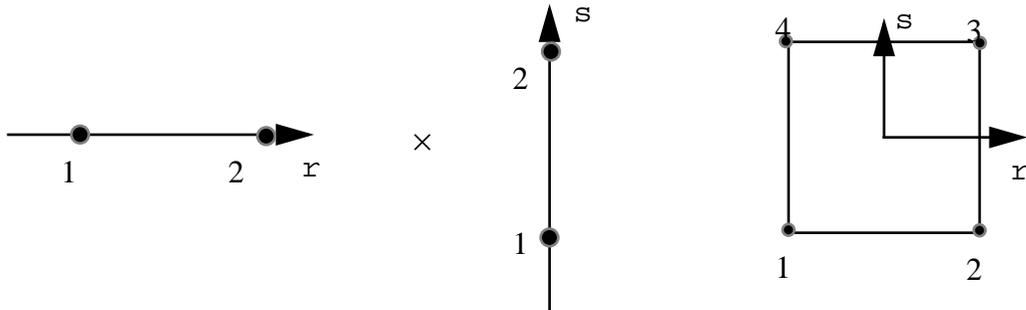


Figure 3.9 Definition of element shape functions for two-dimensional, four-noded quadrilateral.

The three-dimensional case can be treated analogously, and in doing so we use the trilinear brick element depicted in Figure 3.10 as a template. Here we consider that local Node a has local coordinates (r_c, s_d, t_e) and write the three-dimensional shape functions as:

$$N_a(r, s) = L_c^{\text{nen}_r - 1}(r) \times L_d^{\text{nen}_s - 1}(s) \times L_e^{\text{nen}_t - 1}(t). \quad (3.158)$$

The appropriate shape functions for the trilinear brick turn out to be:

$$\begin{aligned} N_1 &= \frac{1}{4}(1-r)(1-s)(1-t), \quad N_2 = \frac{1}{4}(1+r)(1-s)(1-t), \quad N_3 = \frac{1}{4}(1+r)(1+s)(1-t), \\ N_4 &= \frac{1}{4}(1-r)(1+s)(1-t), \quad N_5 = \frac{1}{4}(1-r)(1-s)(1+t), \quad N_6 = \frac{1}{4}(1+r)(1-s)(1+t), \\ N_7 &= \frac{1}{4}(1+r)(1+s)(1+t), \quad \text{and } N_8 = \frac{1}{4}(1-r)(1+s)(1+t). \end{aligned}$$

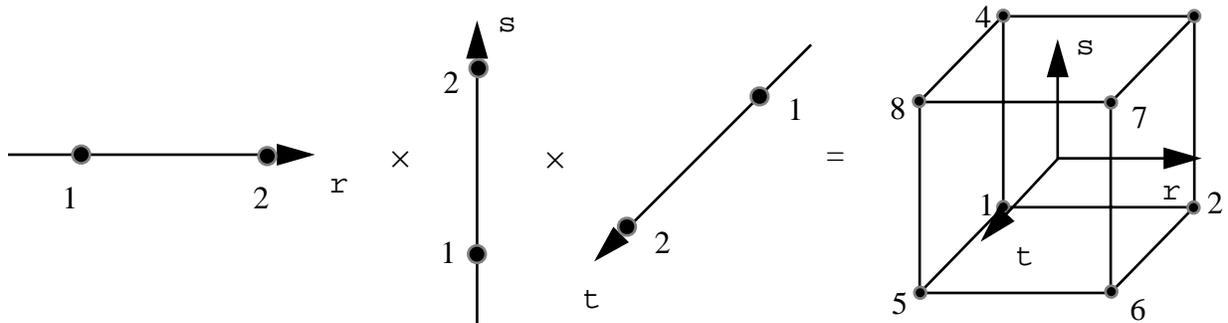


Figure 3.10 Definition of element shape functions for three-dimensional, eight-noded brick.

Quadrature

With the element parameterizations and shape functions now defined, we are in a position to discuss how element-level calculations are performed to evaluate such quantities as \mathbf{m}^e , $\mathbf{f}^{\text{int}^e}$, and \mathbf{k}^e . Notably all these calculations involve integrals over the element domain, as evidenced by the expression for $\mathbf{f}^{\text{int}^e}$ given in (3.142). One can evaluate these integrals analytically only for a few highly specialized cases, meaning that numerical integration (i.e., quadrature) is required for any type of generality to be present in the element formulation. Accordingly, we are led in this section to consider the generic problem of integrating a function, f , over the element domain via

$$\int_{\phi_t^h(\Omega^e)} f(\mathbf{x}^e) d\mathbf{v}, \quad (3.159)$$

where f could, in principle, be scalar, vector, or tensor-valued.

The first step in evaluation of (3.159) is generally to perform a change of variables, converting the integral in the current element physical domain $\phi_t^h(\Omega^e)$ to one over the parent domain, which we shall denote by \square . This is accomplished using the standard change-of-variables formula from multivariate calculus,

$$\int_{\phi_t^h(\Omega^e)} f(\mathbf{x}^e) d\mathbf{v} = \int_{\square} f(\mathbf{x}^e(\mathbf{r})) j(\mathbf{r}) d\square, \quad (3.160)$$

where $j(\mathbf{r})$ is the Jacobian of the transformation from parent coordinates \mathbf{r} to spatial coordinates \mathbf{x}^e :

$$j(\mathbf{r}) = \det \left[\frac{\partial \mathbf{x}^e}{\partial \mathbf{r}} \right], \quad (3.161)$$

where \mathbf{x}^e is as given in (3.147).

The advantage of (3.160) over (3.159) is that the integration takes place over a standardized domain, for which quadrature rules are readily tabulated. One typically approximates the integral in (3.160) by applying quadrature via

$$\int_{\square} f(\mathbf{x}^e(\mathbf{r})) j(\mathbf{r}) d\square \approx \sum_{l=1}^{n_{\text{int}}} f(\mathbf{x}^e(\mathbf{r}_l)) j(\mathbf{r}_l) W_l, \quad (3.162)$$

where n_{int} is the number of integration (quadrature) points in the element, \mathbf{r}_1 is the parent coordinate of quadrature Point 1, and w_1 is the weight associated with quadrature Point 1. The choice of these quadrature point coordinates and weights effectively defines the numerical integration scheme and the accuracy associated with it.

The most prevalent quadrature schemes in finite elements are based on Gaussian quadrature rules, which may be derived in terms of Legendre polynomials. While this derivation is unnecessary for our present purposes, its result is that in one dimension the Gaussian quadrature rules are optimally accurate in that no greater accuracy can be achieved for lesser cost. By cost we mean the number of integration points used, whereas by accuracy we mean the lowest order polynomial not integrated exactly by a given quadrature rule. The Gaussian rules in one dimension have the property that given n_{int} integration points, $2 \times n_{int}$ order accuracy is achieved, meaning that a $2 \times n_{int} - 1$ order polynomial in \mathbf{r} will be integrated exactly. Below are listed the first few Gauss integration rules over the domain $[-1, 1]$.

- $n_{int} = 1$: $r_1 = 0$, $w_1 = 2$ (second order accurate).
- $n_{int} = 2$: $r_1 = -\frac{1}{\sqrt{3}}$, $r_2 = \frac{1}{\sqrt{3}}$, $w_1 = w_2 = 1$ (fourth order accurate).
- $n_{int} = 3$: $r_1 = -\sqrt{\frac{3}{5}}$, $r_2 = 0$, $r_3 = \sqrt{\frac{3}{5}}$, $w_1 = w_3 = \frac{5}{9}$, $w_2 = \frac{8}{9}$ (sixth order accurate).

Now returning to the problem of interest, multidimensional quadrature, we can use very similar reasoning to that used to define multidimensional shape functions in the last section. Since integration in a multidimensional domain involves integrating with respect to each variable separately while holding the others constant, we might expect that numerical integration in successive directions is done in just the same way. The result of this fact is that we can define multidimensional quadrature rules as products of one-dimensional ones, just as was done for shape function definitions. It turns out the optimality property present in the one-dimensional formulation is lost but that a highly systematic and effective integration procedure results.

Beginning in two dimensions we refer to Figure 3.11, which depicts a four-point quadrature scheme in two dimensions that we will use as a template. Let us consider quadrature Point 1, having local coordinates (r_{1_r}, s_{1_s}) , where indices 1_r and 1_s refer to the appropriate quadrature point number in the r and s directions, respectively. The two-dimensional weight is given by the product of the appropriate weights from the one-dimensional rules, i.e.:

$$w_1 = w_{1_r} \times w_{1_s}. \quad (3.163)$$

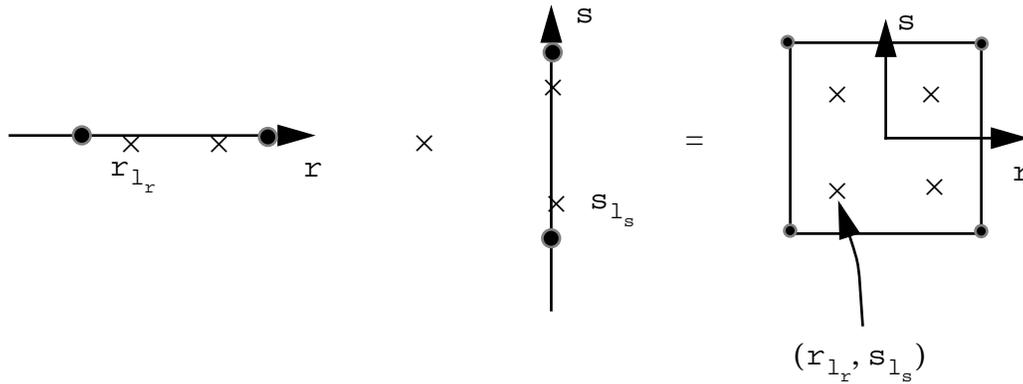


Figure 3.11 Quadrature rule definition in two dimensions: four-point Gaussian quadrature.

Taking the two-dimensional, four-point quadrature rule depicted in Figure 3.11 as an example, the appropriate parameters are found to be: $W_1 = W_2 = W_3 = W_4 = 1$,

$$(r_1, s_1) = \left(-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right), (r_2, s_2) = \left(\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right), (r_3, s_3) = \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right), \text{ and}$$

$$(r_4, s_4) = \left(-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right).$$

Three-dimensional quadrature rules are similarly conceived; the reader should consult Figure 3.12 for a template of the procedure. We consider quadrature Point 1, having local coordinates $(r_{1_r}, s_{1_s}, t_{1_t})$, where indices 1_r , 1_s , and 1_t refer to the appropriate quadrature point number in the r , s , and t directions, respectively. The three-dimensional weight is given by the product of the appropriate weights from the one-dimensional rules, i.e.:

$$W_1 = W_{1_r} \times W_{1_s} \times W_{1_t}. \quad (3.164)$$

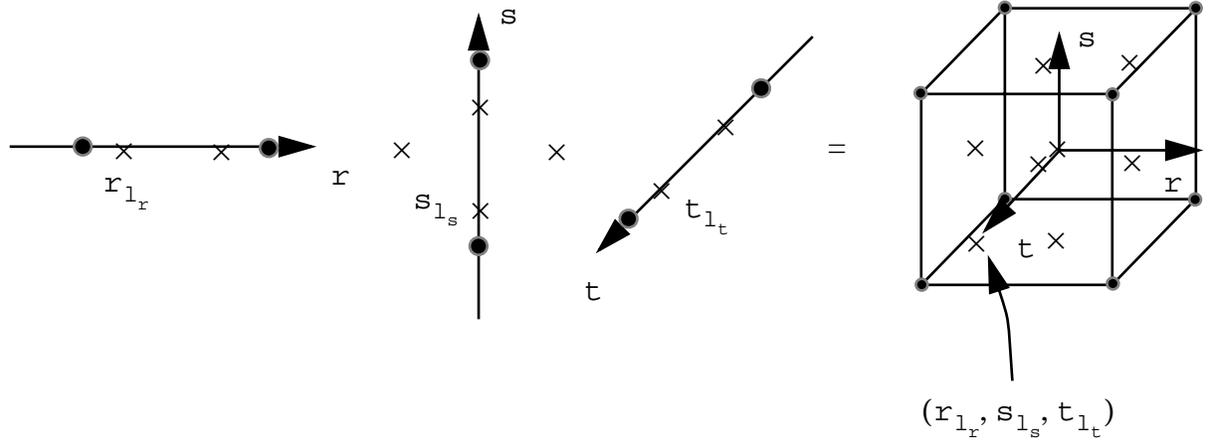


Figure 3.12 Quadrature rule definition in three dimensions: eight-point Gaussian quadrature.

Considering the case in Figure 3.12 as a specific example, we find the following parameters:

$$W_1 - W_8 = 1, (r_1, s_1, t_1) = \left(-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right), (r_2, s_2, t_2) = \left(\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right),$$

$$(r_3, s_3, t_3) = \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right), (r_4, s_4, t_4) = \left(-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right),$$

$$(r_5, s_5, t_5) = \left(-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right), (r_6, s_6, t_6) = \left(\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right),$$

$$(r_7, s_7, t_7) = \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right), \text{ and } (r_8, s_8, t_8) = \left(-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right).$$

Local Arrays

The final task in the section is to give brief prescriptions for how element-level calculations are done to find \mathbf{m}^e , $\mathbf{f}^{\text{int}^e}$, and \mathbf{k}^e for a given element, e . These quantities are needed by the global assembly algorithm to form \mathbf{M} and \mathbf{F}^{int} (as discussed in Localization and Assembly) and \mathbf{K} (as discussed in Newton Raphson Framework). Beginning first with \mathbf{m}^e we recall the general expression for the element mass matrix

$$\mathbf{m}_{pq}^e = \int_{\varphi_t^h(\Omega^e)} \rho N_a(\varphi_t^{-1}(\mathbf{x})) \delta_{ij} N_b(\varphi_t^{-1}(\mathbf{x})) d\mathbf{v}. \quad (3.165)$$

One can apply a change of variables to the reference configuration to find that

$$\begin{aligned}
m_{pq}^e &= \int_{\Omega^e} \rho N_a(\mathbf{x}^e) \delta_{ij} N_b(\mathbf{x}^e) J dV \\
&= \int_{\Omega^e} \rho_0 N_a(\mathbf{x}^e) \delta_{ij} N_b(\mathbf{x}^e) dV ,
\end{aligned} \tag{3.166}$$

where the second line of (3.166) holds because $\rho_0 = J\rho$ (conservation of mass, see Eq. (2.74)). Looking at the second line of (3.166), we see that the element mass matrix is independent of the deformation. It is straightforwardly calculated using quadrature via

$$m_{pq}^e \approx \sum_{l=1}^{nint} \rho_0(\mathbf{r}_l) N_a(\mathbf{r}_l) \delta_{ij} N_b(\mathbf{r}_l) j_0(\mathbf{r}_l) W_l , \tag{3.167}$$

where

$$j_0(\mathbf{r}_l) = \det \left[\frac{\partial \mathbf{x}^e}{\partial \mathbf{r}} \right] . \tag{3.168}$$

According to the discussion of the last section, the ordinary strategy in applying quadrature would be to use a sufficiently accurate rule so that (3.167) is evaluated exactly (at least if the reference density ρ_0 is constant). This would lead one to employ a four-point Gauss quadrature rule for a four-noded quadrilateral in two dimensions and an eight-point Gauss rule for an eight-noded brick in three dimensions. Following this procedure produces a “consistent” mass matrix \mathbf{m}^e .

The difficulty with a consistent mass matrix, however, is twofold. First, it is, in general, banded but not diagonal (as would be preferable for an explicit dynamics application, for example), and second, experience shows that better accuracy is often exhibited in dynamics problems with “lumped mass”, where the rows of (3.168) are actually summed and the result placed on the diagonal of the mass matrix. Use of this row sum technique produces the following alternative, more widely used expression for element mass:

$$m_{pq}^e \Big|_{\text{lumped}} \approx \delta_{ij} \delta_{ab} \sum_{l=1}^{nint} \rho_0(\mathbf{r}_l) N_a(\mathbf{r}_l) j_0(\mathbf{r}_l) W_l . \tag{3.169}$$

Turning attention now to \mathbf{f}^{int^e} , we begin by applying the change of variables to (3.142):

$$\mathbf{f}^{int} \Big|_p^e \approx \sum_{l=1}^{nint} N_{a,j}(\mathbf{r}_l) T_{ij}^h j_0(\mathbf{r}_l) W_l . \tag{3.170}$$

Two requirements of (3.170) are notable: the determination of the stress \mathbf{T}_{ij}^h (dependent, as indicated on the current element deformation field through the constitutive law), and the need for the spatial derivatives $N_{a,j}$ of the shape functions. In fact, derivatives of this type are also needed for the stress calculation, which will ordinarily involve the approximated deformation gradient:

$$\mathbf{F}^h = \left[\frac{\partial \phi_t^h}{\partial \mathbf{x}^e} \right] = \left[\sum_{a=1}^{n_{en}} \frac{\partial N_a}{\partial \mathbf{x}^e} \mathbf{d}_a^e \right]. \quad (3.171)$$

Thus calculation of both $\frac{\partial N_a}{\partial \mathbf{x}^e}$ and $\frac{\partial N_a}{\partial \mathbf{x}^e}$ is typically necessary to obtain \mathbf{f}^{int^e} . These derivatives are usually produced by a shape function subroutine, called by the element subroutine for each quadrature point. Taking the spatial derivative $\frac{\partial N_a}{\partial \mathbf{x}^e}$ as an example, the chain rule can be invoked to obtain the appropriate expressions via:

$$\left\{ \begin{array}{c} \frac{\partial N_a}{\partial \mathbf{x}_j^e} \end{array} \right\} = \left\{ \begin{array}{c} \frac{\partial N_a}{\partial r_k} \end{array} \right\} \left[\begin{array}{c} \frac{\partial r_k}{\partial \mathbf{x}_j^e} \end{array} \right] = \left\{ \begin{array}{c} \frac{\partial N_a}{\partial r_k} \end{array} \right\} \left[\begin{array}{c} \frac{\partial \mathbf{x}_j^e}{\partial r_k} \end{array} \right]^{-1}, \quad (3.172)$$

where $(r_k) = (r, s)$ in two dimensions, and $(r_k) = (r, s, t)$ in three dimensions. The reader

will recognize that $\frac{\partial N_a}{\partial r_k}$ can be computed through simple differentiation of the local shape

functions, whereas $\frac{\partial \mathbf{x}_j^e}{\partial r_k}$ is found by differentiation of (3.147). Calculation of the required inverse

is rather simple and is readily done in closed form, since the matrix involved is 2x2 in two dimensions and 3x3 in three dimensions.

Completing our discussion of element-level calculations, the element stiffness matrix \mathbf{k}^e is given generically by

$$k_{pq}^e(\mathbf{d}^{e,i}) = \frac{\partial f_p^{int}}{\partial d_q^e}(\mathbf{d}^{e,i}), \quad (3.173)$$

where the internal force vector is as given in (3.142). Some manipulation of (3.142) is useful at this point:

$$\begin{aligned}
\mathbf{f}^{\text{int}}|_p^e &= \int_{\Omega^e} [N_{a,J}(\mathbf{x}^e) F_{Jj}^{-1} T_{ij}^h] J dV \\
&= \int_{\Omega^e} [N_{a,J}(\mathbf{x}^e) P_{iJ}^h] dV
\end{aligned} \tag{3.174}$$

where in the second line of (3.174), the Piola stress P_{iJ}^h has been introduced in accordance with (2.54). There are many ways in which the derivative indicated in (3.173) can be expressed; here we do it by computing the derivative of the Piola stress with respect to the deformation gradient (Eq. (3.171)) and invoking the chain rule:

$$\begin{aligned}
k_{pq}^e(\mathbf{d}^{e^i}) &= \frac{\partial}{\partial \mathbf{d}_{q\Omega^e}^e} \int_{\Omega^e} [N_{a,J}(\mathbf{x}^e) P_{iJ}^h] dV \\
&= \int_{\Omega^e} N_{a,J}(\mathbf{x}^e) \frac{\partial P_{iJ}^h}{\partial F_{kL}^h} \frac{\partial F_{kL}^h}{\partial \mathbf{d}_q^e} dV
\end{aligned} \tag{3.175}$$

Simplification of the second line of (3.175) results in the following expression:

$$k_{pq}^e(\mathbf{d}^{e^i}) = \int_{\Omega^e} N_{a,J}(\mathbf{x}^e) C_{iJjL}(\varphi_t^h) N_{b,L}(\mathbf{x}^e) dV, \tag{3.176}$$

where the material moduli C_{iJjL} are defined as

$$C_{iJjL} = \frac{\partial P_{iJ}^h}{\partial F_{jL}^h}. \tag{3.177}$$

Application of the quadrature rule to (3.176) gives

$$k_{pq}^e(\mathbf{d}^{e^i}) \approx \sum_{l=1}^{n_{\text{int}}} N_{a,J}(\mathbf{r}_l) C_{iJjL}(\mathbf{r}_l) N_{b,L}(\mathbf{r}_l) j_0(\mathbf{r}_l) W_l. \tag{3.178}$$

The required reference coordinate shape function derivatives can be calculated as discussed above for each quadrature point, as can the Jacobian j_0 . The material moduli C_{iJjL} are, typically, the most difficult to compute, as they require linearization of the tensor-valued constitutive relation with respect to a tensor-valued strain measure (in this case the deformation gradient). It should be noted that (3.178) is given for illustrative purposes only; the stress and strain measures conveniently utilized in the linearization vary widely, depending on the constitutive relation used. It should be noted, however, that provided the moduli are symmetric (in the major sense), then the element stiffness matrix \mathbf{k}^e will be as well.

Advanced Element Design Issues

Introduction

In this section we discuss some advanced element design issues having particular relevance to large deformation problems featuring inelastic response. We begin the discussion with a specific example of how the standard element formulations discussed in Basics of Element Design can have difficulty in problems featuring near or complete incompressibility, as is common, for example, in computational plasticity. Some basic remedies for this situation are then discussed.

Constrained Media and Locking

The incompressibility dilemma can be motivated fairly simply by considering linear elastic, isotropic behavior. Returning once more to the discussion in Linear Elastic IBVP, we consider a stress strain relation of the form

$$T_{ij} = C_{ijkl} E_{kl}, \quad (3.179)$$

where the material moduli C_{ijkl} are of the form

$$C_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu [\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}]. \quad (3.180)$$

Plugging (3.180) into (3.179), one obtains

$$T_{ij} = 2\mu u_{(i,j)} + \lambda \delta_{ij} u_{k,k}. \quad (3.181)$$

We are most interested in the volumetric response of this material; accordingly, let us define the hydrostatic pressure p as

$$-p = \frac{1}{3} T_{k,k} \quad (3.182)$$

and the dilatation (volume change) Θ as

$$\Theta = u_{k,k}. \quad (3.183)$$

Computing according to (3.181), we find that for an isotropic material,

$$-p = \frac{1}{3} (2\mu + 3\lambda) \Theta. \quad (3.184)$$

The coefficient relating $-p$ and Θ is ordinarily called the bulk modulus K :

$$\kappa = 2\mu + 3\lambda \quad (3.185)$$

and corresponds physically to the volumetric stiffness of the material. Recalling that μ is the shear modulus, representing the resistance of the material to shearing motions, it proves useful to examine the ratio of κ to μ as an indicator of the degree of incompressibility of the material. Using the relationships between the Lamé parameters and the more familiar elastic modulus and Poisson's ratio (see (1.64) and (1.65)), we find this ratio can be written solely in terms of the Poisson's ratio:

$$\frac{\kappa}{\mu} = \frac{2(1 + \nu)}{3(1 - 2\nu)}. \quad (3.186)$$

Recalling that the thermodynamically admissible values for ν range between -1 and $\frac{1}{2}$, we see

that the case where ν approaches $\frac{1}{2}$ from below causes (3.186) to grow without bound, so that

the bulk modulus becomes infinitely large when $\nu = \frac{1}{2}$. In this case the volume change Θ is constrained to be zero pointwise in the medium, and the material is said to be incompressible.

Let us now consider the behavior of a finite element discretization of the linear boundary value problem described in Linear Elastic IBVP, where $\nu = \frac{1}{2}$. We consider the mesh shown in Figure

3.13, comprised of linear triangles. This is an element not discussed to this point but can be obtained formally by consideration of the four-noded quadrilateral discussed in Basics of Element Design, with two of the nodal coordinates set to be the same. Thus the displacement field is linearly interpolated with the result, in the case of triangles, that the strains are constant throughout the element.

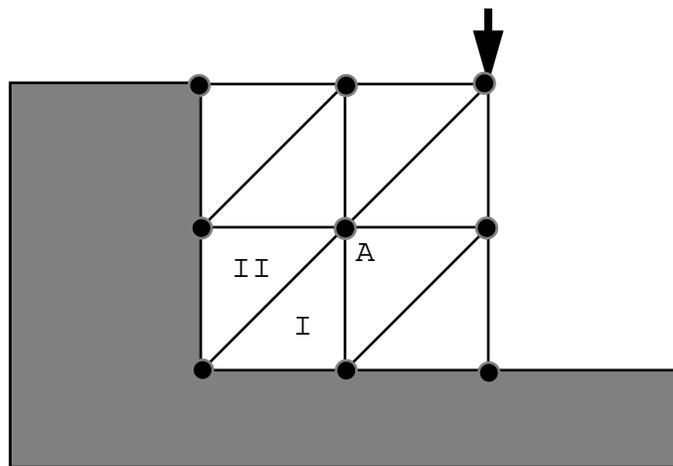


Figure 3.13 Sample mesh illustrating mesh locking for the incompressible case.

Since the strains are constant in the element, the requirement that

$$\mathbf{u}_{k,k}^h = 0 \quad (3.187)$$

pointwise causes the total volume change in the element to be zero also:

$$\frac{dA}{A} = \int_A \mathbf{u}_{k,k}^h dA = 0, \quad (3.188)$$

meaning simply that each element in the mesh may not change area due to the incompressibility constraint.

Examining now the behavior of Element I in Figure 3.13, the constant area constraint implies that Node A can only move in the horizontal direction, since the two lower nodes of Element I are fixed. However, Element II places the restriction that Node A can only move vertically. Taken together, the isochoric constraint in each element prevents Node A from moving at all, in any direction. This argument can be repeated throughout the mesh, to conclude that no node can move at all, so that $\mathbf{u}^h = 0$ is the solution to the discrete problem. However, it is clear that in the physical situation, the fact that the material is incompressible does not preclude all deformation. Thus the finite element solution produces a solution that is nearly nonphysical because of the fact that the numerical approximation of the incompressibility condition overconstrains the numerical representation of the physical system.

The phenomenon described for this admittedly specialized system is referred to, generally speaking, as “mesh locking”. It will not, of course, always be the case that $\mathbf{u}^h = 0$ for every boundary value problem, but it does turn out that fully integrated elements of the type discussed in Basics of Element Design will, in general, produce excessively stiff solutions when the material is either nearly or completely incompressible. We are thus led to consider techniques where the amount of constraint placed by the approximation of the volumetric material response can be relaxed when appropriate.

Selective/Fully Reduced Integration

One of the simplest techniques used to eliminate element locking is to deliberately underintegrate the internal force vector \mathbf{f}^{inte} (and the element stiffness \mathbf{k}^e in the case of a quasistatic or implicit dynamic calculation). Selective reduced integration means that only the troublesome volumetric terms are underintegrated, whereas fully reduced integration means that all terms are underintegrated. The latter option is particularly attractive in explicit dynamic and matrix-free quasistatic calculations, since the element level calculations comprise a large proportion of total solution costs in these cases. Since the cost of element calculations is directly proportional to the number of quadrature points, fully reduced integration becomes very attractive when speed is of special concern.

Let us consider the case of the eight-noded hexahedron in three dimensions as an example, and apply fully reduced integration in the calculation of $\mathbf{f}^{\text{int}^e}$. The ordinary quadrature rule for this element would be eight-point Gauss (two points in each direction), but it turns out that this element locks. The reduced quadrature rule would then be one-point Gauss, which leads to the following expression for $\mathbf{f}^{\text{int}^e}$:

$$\mathbf{f}^{\text{int}^e}|_p \approx 8N_{a,j}(\mathbf{0})T_{ij}^h(\mathbf{0})j_0(\mathbf{0}), \quad (3.189)$$

where, as indicated, all quantities are evaluated at the origin $\mathbf{0}$ of the parent coordinates.

Schemes such as this have the advantages of being cheap and of eliminating locking but come at a cost. They do not accurately integrate those parts of the integrand of $\mathbf{f}^{\text{int}^e}|_p$ that come from deformation varying spatially in a superlinear fashion.

Hourglass Control

Arguably the most important work done in this area was published by [Flanagan, D.P. and Belytschko, T., 1981]. The development in this section closely follows their original presentation. To understand more clearly the possible spurious behaviors enabled by reduced integration, we first note that the shape functions N_a can be written in terms of some standardized element deformation modes as follows:

$$\begin{aligned} N_a = & \frac{1}{8}\Sigma_a + \frac{1}{4}r\Lambda_{1a} + \frac{1}{4}s\Lambda_{2a} + \frac{1}{4}t\Lambda_{3a} \\ & + \frac{1}{2}st\Gamma_{1a} + \frac{1}{2}rt\Gamma_{2a} + \frac{1}{2}rs\Gamma_{3a} + \frac{1}{2}rst\Gamma_{4a} \end{aligned} \quad (3.190)$$

These modes are depicted schematically in Figure 3.14. As can be seen therein, Σ_a represents a rigid body translation, the Λ_{ia} represent constant strain deformation modes, and the $\Gamma_{\alpha a}$ are referred to as hourglass modes.

If we consider the velocity field representable by this element, we find that it can be written via

$$\mathbf{v}^e = V_i^e \mathbf{e}_i, V_i^e = \sum_{a=1}^{\text{nen}} V_{ia} N_a. \quad (3.191)$$

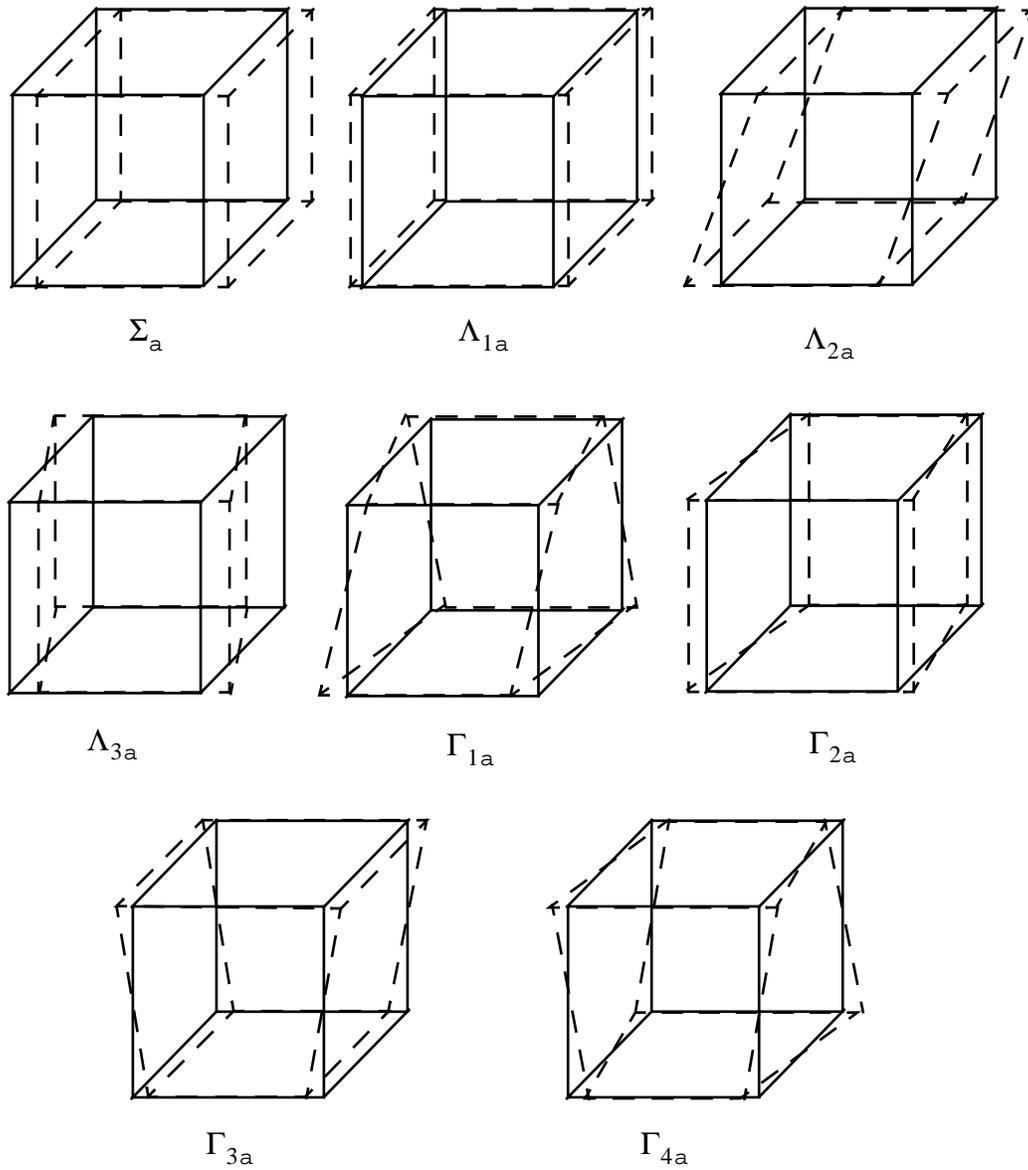


Figure 3.14 Mode shapes for the eight-noded hexahedron element.

The fully linear portion of the velocity field is made up of the Σ_a and Λ_{ia} modes, so that the hourglass portion of the velocity field can be written as

$$\mathbf{v}_{ia}^{hg} = \mathbf{v}_{ia} - \mathbf{v}_{ia}^{lin} = \frac{1}{\sqrt{8}} \mathbf{q}_{i\alpha} \dot{\Gamma}_{\alpha a}, \quad (3.192)$$

where $\frac{1}{\sqrt{8}}$ is a normalizing factor, and $\mathbf{q}_{i\alpha}$ are the hourglass normal velocities. It turns out that the hourglass velocities are orthogonal to the element's other modes in that

$$V_{ia}^{hg} \Sigma_a = 0 \quad (3.193)$$

and

$$V_{ia}^{hg} \Lambda_{ia} = 0. \quad (3.194)$$

Basically the one-point integration scheme discussed in the last section fully controls the linear modes of the system but provides no resistance at all to the hourglass velocities V_{ia}^{hg} . The objective of hourglass control therefore is to restore such control even in the context of one-point integration.

Flanagan and Belytschko wrote the hourglass nodal velocities in terms of the hourglass shape vector $\gamma_{\alpha a}$ via

$$\dot{q}_{i\alpha} = \frac{1}{\sqrt{8}} V_{ia} \gamma_{\alpha a}, \quad (3.195)$$

with the shape vector $\gamma_{\alpha a}$ found to be

$$\gamma_{\alpha a} = \Gamma_{\alpha a} - \frac{1}{V} \frac{\partial V}{\partial X_{ia}^e} X_{ib}^e \Gamma_{\alpha b}. \quad (3.196)$$

Hourglass forces f_{ia}^{hg} are applied in these directions, so as to be orthogonal to the physical modes of the system. One choice is

$$f_{ia}^{hg} = \frac{1}{2} Q_{i\alpha} \gamma_{\alpha a}, \quad (3.197)$$

where the generalized forces $Q_{i\alpha}$ are given via

$$Q_{i\alpha} = \frac{\hat{\mu}}{50V} \frac{\partial V}{\partial X_{ja}^e} \frac{\partial V}{\partial X_{ja}^e} \dot{q}_{i\alpha}, \quad (3.198)$$

with $\hat{\mu}$ being an effective shear modulus. The interested reader should refer to Hourglass Control Algorithm in Eight-Node Uniform Strain Element for the details of a specific implementation of hourglass control.

Eight-Node Uniform Strain Element

Introduction

The eight-node, three-dimensional isoparametric element is widely used in computational mechanics. The determination of optimal integration schemes for this element, however, presents a difficult dilemma. A one-point integration of the element underintegrates the element, resulting in a rank deficiency that manifests itself in spurious zero energy modes, commonly referred to as hourglass modes (see Hourglass Control). A two-by-two-by-two integration of the element, by contrast, overintegrates the element and can lead to serious problems of element locking in fully plastic and incompressible problems (see Constrained Media and Locking). The eight-point integration also carries a tremendous computational penalty compared to the one-point rule. Particularly in explicit dynamic applications (see Explicit Finite Element Methods), this added expense is extremely undesirable.

In this section we present an element that is widely used in explicit analyses, wherein one-point integration is utilized in combination with an hourglass control scheme that controls the spurious modes. The implementation presented below follows directly from [Flanagan, D.P. and Belytschko, T., 1981]. In particular the aspects of this element pertaining to hourglass control have already been discussed in Hourglass Control.

The hexahedral element relates the spatial element coordinates x_i^e to the nodal coordinates d_{ia} through the isoparametric shape functions N_a as follows

$$x_i^e = N_a(\xi, \eta, \zeta)d_{ia}, \quad (3.199)$$

where for convenience the summation convention on nodal indices a has been adopted. Subscripts i have a range of three, corresponding to the spatial coordinate directions, and subscripts a have a range of eight.

Our discussion here will focus primarily on explicit dynamics applications, where the argument for the use of this element is most compelling. As such it is necessary to have expressions for the elemental velocity field

$$v_i = N_a v_{ia}, \quad (3.200)$$

and for the acceleration field

$$a_i = N_a a_{ia}. \quad (3.201)$$

In (3.200) and (3.201) the nodal velocities v_{ia} and nodal accelerations a_{ia} are the localized global velocities and accelerations, which for central differences are produced by the updates (3.75) given in Explicit Finite Element Methods.

The velocity gradient tensor \mathbf{L} has been discussed in Rates of Deformation and was specified in Eq. (2.30). It can be written in the domain of an element as

$$L_{ij} = v_{i,j} = v_{ia}N_{a,j}. \quad (3.202)$$

By convention a comma preceding a lowercase subscript denotes differentiation with respect to the spatial coordinates (e.g., $v_{i,j}$ denotes $\frac{\partial v_i}{\partial x_j}$).

The three-dimensional isoparametric shape functions map the unit cube in \mathbf{x} -space (\mathbf{x} is written explicitly as (r, s, t)) to a general hexahedron in \mathbf{x} -space, as depicted for the element reference configuration in Figure 3.6. The trilinear shape functions defined over this domain, as summarized in Shape Functions, can be conveniently expanded in terms of an orthogonal set of base vectors, as was mentioned previously in Hourglass Control:

$$N_a = \frac{1}{8}\Sigma_a + \frac{1}{4}r\Lambda_{1a} + \frac{1}{4}s\Lambda_{2a} + \frac{1}{4}t\Lambda_{3a} + \frac{1}{2}st\Gamma_{1a} + \frac{1}{2}rt\Gamma_{2a} + \frac{1}{2}rs\Gamma_{3a} + \frac{1}{2}rst\Gamma_{4a}. \quad (3.203)$$

The basis vectors represent the displacement modes of a unit cube, as was also discussed in Hourglass Control. The first vector, Σ_a , accounts for rigid body translation. The vectors Λ_{ia} may be readily combined to define three uniform normal strains and three rigid body rotation modes for the unit cube. We refer to Λ_{ia} as the volumetric base vectors since, as we will illustrate below, they are the only base vectors which appear in the element volume expression. The last four vectors, $\Gamma_{\alpha a}$, where Greek subscripts have a range of four, give rise to linear strain modes which are neglected by uniform strain integration (i.e., the one-point quadrature rule summarized in (3.189)). These vectors define the hourglass patterns for a unit cube, so that we refer to $\Gamma_{\alpha a}$ as the hourglass base vectors. The displacement modes represented by all these vectors are shown in Figure 3.14.

Element Force Vector

Recalling the development of Basics of Element Design, the generic expression for the element internal force vector (see (3.142)) is

$$\mathbf{f}^{\text{int}}|_{ia}^e = \int_{\phi_t^h(\Omega^e)} \mathbf{N}_{a,j} \mathbf{T}_{ij} d\mathbf{v}. \quad (3.204)$$

In the element we consider, the one-point integration scheme neglects the nonlinear portion of the element displacement field, thereby considering a state of uniform strain and stress. The preceding expression is approximated by

$$\mathbf{f}^{\text{int}}|_{ia}^e \approx \bar{\mathbf{T}}_{ij} \int_{\phi_t^h(\Omega^e)} \mathbf{N}_{a,j} d\mathbf{v}, \quad (3.205)$$

where $\bar{\mathbf{T}}_{ij}$, the mean stress tensor, represents the assumed uniform stress field. By neglecting the nonlinear displacements, we have assumed that the mean stresses depend only on the mean strains. Mean kinematic quantities are defined by integrating over the element as follows:

$$\bar{\mathbf{v}}_{i,j} = \frac{1}{V} \int_{\phi_t^h(\Omega^e)} v_{i,j} d\mathbf{v}. \quad (3.206)$$

We now define the discrete gradient operator as

$$\mathbf{B}_{ia} = \int_{\phi_t^h(\Omega^e)} \mathbf{N}_{a,i} d\mathbf{v}. \quad (3.207)$$

The mean velocity gradient, applying Eq. (3.202), is given by

$$\bar{\mathbf{v}}_{i,j} = \frac{1}{V} v_{ia} \mathbf{B}_{ja}. \quad (3.208)$$

Combining Equations (3.205) and (3.207), we may express the nodal forces by

$$\mathbf{f}^{\text{int}}|_{ia}^e = \bar{\mathbf{T}}_{ij} \mathbf{B}_{ja}. \quad (3.209)$$

Computing nodal forces with this integration scheme requires evaluation of the gradient operator and the element volume. These two tasks are linked since

$$\mathbf{x}_{i,j} = \delta_{ij}, \quad (3.210)$$

where δ_{ij} is the Kronecker delta. Equations (3.199), (3.207), and (3.210) yield

$$\mathbf{x}_{ia} \mathbf{B}_{ja} = \int_{\mathbf{v}^e} (\mathbf{x}_{ia} \mathbf{N}_a)_{,j} d\mathbf{v} = \mathbf{v} \delta_{ij}. \quad (3.211)$$

Consequently, the gradient operator may be expressed by

$$B_{ia} = \frac{\partial v}{\partial x_{ia}}. \quad (3.212)$$

To integrate the element area in closed form, we use the Jacobian of the isoparametric transformation to transform to an integral over the biunit cube:

$$v = \int_v dv = \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} j dr ds dt. \quad (3.213)$$

The Jacobian is given in terms of the permutation symbol e_{ijk} as

$$J = e_{ijk} \frac{\partial x}{\partial r_i} \frac{\partial y}{\partial r_j} \frac{\partial z}{\partial r_k}. \quad (3.214)$$

Therefore, Equation (3.213) can be written as

$$v = x_a y_b z_c C_{abc}, \quad (3.215)$$

where

$$C_{abc} = e_{ijk} \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} \frac{\partial N_a}{\partial r_i} \frac{\partial N_b}{\partial r_j} \frac{\partial N_c}{\partial r_k} dr_1 dr_2 dr_3. \quad (3.216)$$

Observe that the coefficient array C_{abc} is identical for all hexahedrons. Furthermore, it possesses the following alternator properties:

$$C_{abc} = C_{bca} = C_{cab} = -C_{acb} = -C_{bac} - C_{cba}. \quad (3.217)$$

Therefore, applying Equations (3.212) and (3.217) to (3.215) yields the following form for evaluating the **B**-matrix:

$$B_{ia} = \begin{bmatrix} Y_b Z_c \\ Z_b X_c \\ X_b Y_c \end{bmatrix} C_{abc}. \quad (3.218)$$

In light of Equation (3.203), it is evident that evaluating each component of C_{abc} involves integrating a polynomial that is at most biquadratic. However, since we are integrating over a symmetric region, any term with a linear dependence will vanish. The only terms which survive the integration will be the constant, square, double square, and triple square terms. Furthermore, the alternator properties cause half of these remaining terms to drop out. The resulting expression for C_{abc} is

$$C_{abc} = \frac{1}{192} e_{ijk} (3\Lambda_{ia}\Lambda_{jb}\Lambda_{kc} + \Lambda_{ia}\Gamma_{kb}\Gamma_{jc} + \Gamma_{ka}\Lambda_{jb}\Gamma_{ic} + \Gamma_{ja}\Gamma_{ib}\Lambda_{kc}) \quad (3.219)$$

Since C_{abc} has the alternator properties given in Equation (3.217), only 56 (the combination of eight nodes taken three at a time) distinct nonzero terms are possible. However, the volume must be independent of the selection of Node 1, which implies that C_{abc} must be invariant under all node numberings, preserving the relative orientation of the element. Consequently, only 21 (the combination of seven nodes taken two at a time) terms may be independent. Furthermore, once Node 1 is selected, three orientations of the node numbering system are possible, so that only seven terms of C_{abc} need be evaluated.

The seven sets of triples (a, b, c) giving rise to independent terms of C_{abc} are: (1, 2, 3); (1, 2, 5); (1, 2, 6); (1, 2, 7); (1, 2, 8); (1, 3, 5); 1, 3, 6. Of these, only the first three terms do not vanish. All other nonzero terms of C_{abc} are found by permutations and use of the alternator properties summarized by Eq. (3.217).

With the C_{abc} in hand, the first term of B_{ia} is expressed using (3.218) as

$$B_{11} = \frac{1}{2} [Y_2((z_6 - z_3) - (z_4 - z_5)) + Y_3(z_2 - z_4) + Y_4((z_3 - z_8) - (z_5 - z_2)) + Y_5((z_8 - z_6) - (z_2 - z_4)) + Y_6(z_5 - z_2) + Y_8(z_4 - z_5)] \quad (3.220)$$

After permuting the nodes according to the transformations described above, other terms of B_{ia} are also evaluated using (3.218). The element volume is most easily computed by contracting the \mathbf{B} -matrix and nodal coordinates as per Eq. (3.211).

Lumped Mass Matrix

In order to reap the benefits of an explicit architecture, we must diagonalize the mass matrix (see Explicit Finite Element Methods). We do this by integrating the inertial terms as

$$(\mathbf{m}^e \mathbf{a}^e)_{ia} = a_{ib} m_{ab}^e, \quad (3.221)$$

where

$$m_{ab}^e = \rho v \delta_{ab}, \quad (3.222)$$

and δ_{ab} is the Kronecker delta. Clearly the assembly process for the global mass matrix from the individual element matrices results in a global mass matrix that is diagonal and can be expressed as a vector, M_p , if desired.

Finite Rotation Algorithm

As discussed in more detail in Frame Indifference, an important factor in proper formulation of large deformation problems is the assurance of material objectivity. In the element we now consider, this is achieved by formulation of the constitutive updates in the *rotated configuration* depicted in Figure 2.2 and introduced more thoroughly in Rate of Deformation Tensors. Of particular interest in constitutive modeling are quantities like the rotated rate of deformation tensor \mathbf{D} (see Eq. (2.39)), the rotated Cauchy stress \mathbf{T} in Eq. (2.57), and the Green-Naghdi rate of Cauchy stress $\tilde{\mathbf{T}}$ defined in (2.122).

Notably all of the above objects require the determination of \mathbf{R} , the rotation tensor defined by the polar decomposition summarized by Eq. (2.13). Here we describe an incremental algorithm for determination of this tensor with emphasis on computational efficiency and numerical accuracy. We begin by considering (2.42) as a first-order differential equation in \mathbf{R}

$$\dot{\mathbf{R}} = \mathbf{L}\mathbf{R}. \quad (3.223)$$

The crux of integrating Eq. (3.223) for \mathbf{R} is to maintain the orthogonality of \mathbf{R} . Unfortunately, if one merely applies a forward difference scheme, the orthogonality of \mathbf{R} degenerates rapidly no matter how fine the time increments. Instead the algorithm of Hughes and Winget ([Hughes, T.J.R. and Winget, J., 1980]) for integrating incremental rotations can be adopted as follows.

A rigid body rotation over a time increment Δt may be represented by

$$\mathbf{x}_{t+\Delta t} = \mathbf{Q}_{\Delta t}\mathbf{x}_t, \quad (3.224)$$

where $\mathbf{Q}_{\Delta t}$ is a proper orthogonal tensor with the same rate of rotation as \mathbf{R} , as given by Eq. (3.223). The total rotation \mathbf{R} is updated via

$$\mathbf{R}_{t+\Delta t} = \mathbf{Q}_{\Delta t}\mathbf{R}_t. \quad (3.225)$$

For a constant rate of rotation, the midpoint velocity and the midpoint coordinates are related by

$$\frac{1}{\Delta t}(\mathbf{x}_{t+\Delta t} - \mathbf{x}_t) = \frac{1}{2}\mathbf{L}(\mathbf{x}_{t+\Delta t} + \mathbf{x}_t). \quad (3.226)$$

Combining Eqs. (3.224) and (3.226) yields

$$(\mathbf{Q}_{\Delta t} - \mathbf{I})\mathbf{x}_t = \frac{\Delta t}{2}\mathbf{L}(\mathbf{Q}_{\Delta t} + \mathbf{I})\mathbf{x}_t. \quad (3.227)$$

Since \mathbf{x}_t is arbitrary in Equation (3.227), it may be eliminated. We then solve for $\mathbf{Q}_{\Delta t}$, which gives

$$\mathbf{Q}_{\Delta t} = \left(\mathbf{I} - \frac{\Delta t}{2}\mathbf{L}\right)^{-1}\left(\mathbf{I} + \frac{\Delta t}{2}\mathbf{L}\right). \quad (3.228)$$

The accuracy of this integration scheme is dependent upon the accuracy of the midpoint relationship of Equation (3.226). The rate of rotation must not vary significantly over the time increment. Furthermore, [Hughes, T.J.R. and Winget, J., 1980] showed that the conditioning of Equation (3.228) degenerates as $\Delta t \mathbf{L}$ grows.

Our complete numerical algorithm for a single time step can be summarized as below.

- Calculate the rate of deformation tensor \mathbf{D} and the spin tensor \mathbf{W} (see Eqs. (2.31) and (2.32))
- Determine \mathbf{L} from \mathbf{W} and \mathbf{V} (the left stretch, see Eq. (2.13)), using the following algorithm due to [Dienes, J.K., 1979]. Compute

$$z_i = e_{ijk}V_{jm}D_{mk} \quad (3.229)$$

$$\mathbf{l} = \mathbf{w} - 2(\mathbf{V} - \mathbf{I}\text{tr}(\mathbf{V}))^{-1}\mathbf{z} \quad (3.230)$$

$$L_{ij} = \frac{1}{2}e_{ijk}l_k, \quad (3.231)$$

where

$$w_i = e_{ijk}W_{jk}. \quad (3.232)$$

- Solve

$$\left(\mathbf{I} - \frac{\Delta t}{2}\mathbf{L}\right)\mathbf{R}_{t+\Delta t} = \left(\mathbf{I} + \frac{\Delta t}{2}\mathbf{L}\right)\mathbf{R}_t. \quad (3.233)$$

- Calculate

$$\dot{\mathbf{V}} = (\mathbf{D} + \mathbf{W})\mathbf{V} - \mathbf{V}\mathbf{L}. \quad (3.234)$$

- Update

$$\mathbf{V}_{t+\Delta t} = \mathbf{V}_t + \Delta t \dot{\mathbf{V}}. \quad (3.235)$$

- Compute the rotated rate of deformation (see (2.39))

$$\mathbf{D} = \mathbf{R}^T \mathbf{D} \mathbf{R}. \quad (3.236)$$

- Integrate the constitutive equations in the rotated frame of reference

$$\dot{\mathbf{T}} = \mathbf{f}(\mathbf{D}, \mathbf{T}). \quad (3.237)$$

- Compute the Cauchy stress in the spatial configuration

$$\mathbf{T} = \mathbf{R}\mathbf{T}\mathbf{R}^T. \quad (3.238)$$

Note that this algorithm requires that the tensors \mathbf{V} and \mathbf{R} be stored in memory for each element.

Determination of Effective Moduli

Algorithms for calculating the stable time increment require effective moduli for each element (see (3.73) in Explicit Finite Element Methods). Such calculations of dilatational and shear moduli are also necessary for hourglass control, bulk viscosity, and nonreflecting boundaries. Here we present a procedure for adaptively determining the effective dilatational and shear moduli of the material.

In an explicit integration algorithm, the constitutive response over a time step can be recast a posteriori as a hypoelastic relationship. We approximate this relationship as isotropic. This defines effective Lamé parameters, $\hat{\lambda}$ and $\hat{\mu}$, in terms of the hypoelastic stress increment and strain increment (in the rotated frame of reference) as follows:

$$\Delta T_{ij} = \Delta t(\hat{\lambda}D_{kk}\delta_{ij} + 2\hat{\mu}D_{ij}). \quad (3.239)$$

Equation (3.239) can be rewritten in terms of volumetric and deviatoric parts as

$$\Delta T_{kk} = \Delta t(3\hat{\lambda} + 2\hat{\mu})D_{kk}, \quad (3.240)$$

and

$$\mathbf{s}_{ij} = \Delta t 2\hat{\mu} \mathbf{e}_{ij}, \quad (3.241)$$

where

$$\mathbf{s}_{ij} = \Delta T_{ij} - \frac{1}{3}\Delta T_{kk}\delta_{ij} \quad (3.242)$$

and

$$\mathbf{e}_{ij} = D_{ij} - \frac{1}{3}D_{kk}\delta_{ij}. \quad (3.243)$$

The effective bulk modulus \hat{K} follows directly from Equation (3.240) as

$$3\hat{\kappa} = 3\hat{\lambda} + 2\hat{\mu} = \frac{\Delta T_{kk}}{\Delta t D_{kk}}. \quad (3.244)$$

Taking the inner product of Equation (3.241) with the deviatoric strain rate and solving for the effective shear modulus $2\hat{\mu}$, gives

$$2\hat{\mu} = \frac{s_{ij}e_{ij}}{\Delta t e_{mn}e_{mn}}. \quad (3.245)$$

Using the result of Equation (3.244) with Equation (3.245), we can calculate the effective dilatational modulus $\hat{\lambda} + 2\hat{\mu}$:

$$\hat{\lambda} + 2\hat{\mu} = \frac{1}{3}(3\hat{\kappa} + 2 \cdot (2\hat{\mu})). \quad (3.246)$$

If the strain increments are insignificant, Equations (3.244) and (3.245) will not yield numerically meaningful results. In this circumstance the dilatational modulus can be set to an initial estimate, $\lambda_0 + 2\mu_0$. An initial estimate of the dilatational modulus is, therefore, the only parameter which every constitutive model is required to provide to the time step control algorithm.

In a case where the volumetric strain increment is significant but the deviatoric increment is not, the effective shear modulus can be estimated by rearranging Equation (3.246) as follows:

$$2\hat{\mu} = \frac{1}{2}(3(\lambda_0 + 2\mu_0) - 3\hat{\kappa}). \quad (3.247)$$

If neither strain increment is significant, the effective shear modulus can be set equal to the initial dilatational modulus.

Determination of the Stable Time Increment

Flanagan and Belytschko [Flanagan, D.P. and Belytschko, T., 1984] provided eigenvalue estimates for the uniform strain hexahedron described in this section. They showed that the maximum eigenvalue is bounded by

$$8 \frac{\lambda + 2\mu}{\rho} \frac{B_{ia}B_{ia}}{v^2} \geq \omega_{\max}^2 \geq \frac{8}{3} \frac{\lambda + 2\mu}{\rho} \frac{B_{ia}B_{ia}}{v^2}. \quad (3.248)$$

Using the effective dilatational modulus from Determination of Effective Moduli with the eigenvalue estimates of Equation (3.248) allows us to write the stability criteria of Eq. (3.73) as

$$\Delta \hat{t}^2 \leq \frac{1}{2} \frac{(\rho_0 V_0)}{(\hat{\lambda} + 2\hat{\mu})B_{iI}B_{iI}}. \quad (3.249)$$

The stable time increment is determined from Equation (3.249) as the minimum over all elements.

Equation (3.249) is numerically invalid if the effective dilatational modulus is less than or equal to zero. A negative modulus indicates a strain softening situation that renders the central difference operator unconditionally unstable. In practice, however, strain softening is generally short lived, so that the calculations can continue in a stable manner once the softening energy has been dissipated. To aid the user in controlling an unstable strain-softening situation, the effective dilatational modulus can be adjusted with a strain-softening scale factor, *ssft*, as follows

$$\text{If } \hat{\lambda} + 2\hat{\mu} < 0 \text{ then } \hat{\lambda} + 2\hat{\mu} = \frac{\lambda_0 + 2\mu_0}{(\text{ssft})^2}. \quad (3.250)$$

To avoid dividing by zero in Equation (3.249), one can enforce the following condition:

$$\hat{\lambda} + 2\hat{\mu} \geq (\lambda_0 + 2\mu_0) \cdot 10^{-6}. \quad (3.251)$$

The estimate of the critical time increment given in Equation (3.249) is for the case where there is no damping in the system. If we define ε as the fraction of critical damping in the highest element mode, the stability criteria of Eq. (3.249) becomes

$$\Delta t \leq \Delta \hat{t} (\sqrt{1 + \varepsilon^2} - \varepsilon). \quad (3.252)$$

Conventional estimates of the critical time increment size have been based on the transit time of a dilatational wave over the shortest dimension l of an element or zone. For the undamped case this gives

$$\Delta t \leq \frac{l}{c}, \quad (3.253)$$

where c is the dilatational wave speed.

There are two fundamental and important differences between the time increment limits given by Equations (3.249) and (3.253). First, our time increment limit is dependent on a characteristic element dimension, which is based on the finite element gradient operator and does not require an ad hoc guess of this dimension. This characteristic element dimension, l , is defined by inspection of Equation (3.249) as

$$l = \frac{1}{2} \frac{v}{\sqrt{B_{iI}B_{iI}}}. \quad (3.254)$$

Second, the sound speed used in the estimate is based on the current response of the material and not on the original elastic sound speed. For materials that experience a reduction in stiffness due to plastic flow, this can result in significant increases in the critical time increment.

It should be noted that the stability analysis performed at each time step predicts the critical time increment for the next step. Our assumption is that the conservativeness of this estimate compensates for any reduction in the stable time increment over a single time step.

Hourglass Control Algorithm

The mean stress-strain formulation of the uniform strain element considers only a fully linear velocity field. The remaining portion of the nodal velocity field is the so-called hourglass field. Excitation of these modes may lead to severe, unresisted mesh distortion. The hourglass control algorithm described here is taken directly from [Flanagan, D.P. and Belytschko, T., 1981]. The method isolates the hourglass modes so that they may be treated independently of the rigid body and uniform strain modes.

A fully linear velocity field for the hexahedron can be described by

$$\mathbf{v}_i^{\text{LIN}} = \bar{\mathbf{v}}_i + \bar{\mathbf{v}}_{i,j}(\mathbf{x}_j - \bar{\mathbf{x}}_j). \quad (3.255)$$

The mean coordinates $\bar{\mathbf{x}}_i$ correspond to the center of the element and are defined as

$$\bar{\mathbf{x}}_i = \frac{1}{8} \mathbf{x}_{ia} \Sigma_a. \quad (3.256)$$

The mean translational velocity is similarly defined by

$$\bar{\mathbf{v}}_i = \frac{1}{8} \mathbf{v}_{ia} \Sigma_a. \quad (3.257)$$

The linear portion of the nodal velocity field may be expressed by specializing Eq. (3.255) to the nodes as follows:

$$\mathbf{v}_{ia}^{\text{LIN}} = \mathbf{v}_i \Sigma_a + \bar{\mathbf{v}}_{i,j}(\mathbf{x}_{ja} - \bar{\mathbf{x}}_j \Sigma_a), \quad (3.258)$$

where Σ_a is used to maintain consistent index notation and indicates that $\bar{\mathbf{v}}_i$ and $\bar{\mathbf{x}}_j$ are independent of position within the element. From Equations (3.211) and (3.258) and the orthogonality of the base vectors, it follows that

$$\mathbf{v}_{ia} \Sigma_a = \mathbf{v}_{ia}^{\text{LIN}} \Sigma_a = 8 \bar{\mathbf{v}}_i, \quad (3.259)$$

and

$$\mathbf{v}_{ia} \mathbf{B}_{ja} = \mathbf{v}_{ia}^{\text{LIN}} \mathbf{B}_{ja} = \mathbf{v} \bar{\mathbf{v}}_{i,j}. \quad (3.260)$$

The hourglass field $\mathbf{v}_{ia}^{\text{HG}}$ may now be defined by removing the linear portion of the nodal velocity field:

$$\mathbf{v}_{ia}^{\text{HG}} = \mathbf{v}_{ia} - \mathbf{v}_{ia}^{\text{LIN}}. \quad (3.261)$$

Equations (3.259) through (3.261) prove that Σ_a and \mathbf{B}_{ja} are orthogonal to the hourglass field:

$$\mathbf{v}_{ia}^{\text{HG}} \Sigma_a = 0 \quad (3.262)$$

$$\mathbf{v}_{ia}^{\text{HG}} \mathbf{B}_{ja} = 0. \quad (3.263)$$

Furthermore, it can be shown that the B matrix is a linear combination of the volumetric base vectors Λ_{ia} , so Eq. (3.263) can be written as

$$\mathbf{v}_{ia}^{\text{HG}} \Lambda_{ia} = 0. \quad (3.264)$$

Equations (3.262) and (3.264) show that the hourglass field is orthogonal to all the base vectors depicted in Figure 3.14 except the hourglass base vectors. Therefore, $\mathbf{v}_{ia}^{\text{HG}}$ may be expanded as a linear combination of the hourglass base vectors as follows:

$$\mathbf{v}_{ia}^{\text{HG}} = \frac{1}{\sqrt{8}} \dot{\mathbf{q}}_{i\alpha} \Gamma_{\alpha a}. \quad (3.265)$$

The hourglass nodal velocities are represented by $\dot{\mathbf{q}}_{i\alpha}$ above (the leading constant is added to normalize $\Gamma_{\alpha a}$). We now define the hourglass shape vector $\gamma_{\alpha a}$ such that

$$\dot{\mathbf{q}}_i = \frac{1}{\sqrt{8}} \dot{u}_{ia} \gamma_{\alpha a}. \quad (3.266)$$

By substituting Equations (3.258), (3.261), and (3.266) into (3.265), then multiplying by $\Gamma_{\alpha a}$ and using the orthogonality of the base vectors, we obtain the following:

$$\dot{u}_{iI} \Gamma_{\alpha I} - \dot{u}_{i,j} \mathbf{x}_{jI} \Gamma_{\alpha I} = \dot{u}_{iI} \gamma_{\alpha I}. \quad (3.267)$$

With the definition of the mean velocity gradient, Equation (3.208), we can eliminate the nodal velocities above. As a result, we can compute $\gamma_{\alpha I}$ from the following expression:

$$\gamma_{\alpha a} = \Gamma_{\alpha a} - \frac{1}{V} \mathbf{B}_{ia} \mathbf{x}_{ib} \Gamma_{\alpha b}. \quad (3.268)$$

The difference between the hourglass base vectors $\Gamma_{\alpha a}$ and the hourglass shape vectors $\gamma_{\alpha a}$ is very important. They are identical if and only if the hexahedron is a right-parallelepiped. For a general shape $\Gamma_{\alpha a}$ is orthogonal to B_{ja} , whereas $\gamma_{\alpha a}$ is orthogonal to the linear velocity field v_{ia}^{LIN} . $\Gamma_{\alpha a}$ defines the hourglass pattern, and $\gamma_{\alpha a}$ is necessary to accurately detect hourglassing.

For the purpose of controlling the hourglass modes, we define generalized forces $Q_{i\alpha}$, which are conjugate to $\dot{q}_{i\alpha}$ so that the rate of work is

$$v_{ia} f_{ia}^{HG} = \frac{1}{2} Q_{i\alpha} \dot{q}_{i\alpha} \quad (3.269)$$

for arbitrary \dot{u}_{iI} . Using Equation (3.266) it follows that the contribution of the hourglass resistance to the nodal forces is given by

$$f_{ia}^{HG} = \frac{1}{2} Q_{i\alpha} \gamma_{\alpha a}. \quad (3.270)$$

Two types of hourglass resistance are possible: artificial stiffness and artificial damping. Considering the stiffness type as an example, we can define a tuneable hourglass stiffness, κ , and express the resistance by

$$Q_{i\alpha} = \frac{\kappa}{2} 2\hat{\mu} \frac{B_{jb} B_{jb}}{v} \dot{q}_{i\alpha}. \quad (3.271)$$

Note that the stiffness expression must be integrated, which further requires that this resistance be stored in a global array.

Observe that the nodal antihourglass forces of Equation (3.270) have the shape of $\gamma_{\alpha a}$ rather than $\Gamma_{\alpha a}$. This fact is essential since the antihourglass forces should be orthogonal to the linear velocity field, so that no energy is transferred to or from the rigid body and uniform strain modes by the antihourglassing scheme.

Artificial Bulk Viscosity

Artificial viscosity may be desirable in numerical calculations for two reasons. First, high-velocity gradients can collapse an element before it has a chance to respond if no viscosity is employed. Second, viscosity is often useful in quieting truncation frequency “ringing”.

Ideally one would like to add viscosity only to the highest mode of the element, but isolating this mode is impractical. The standard technique is to simply add viscosity to the volumetric or “bulk” response. This generates a viscous pressure in terms of the volume strain rate as follows:

$$q = b_1 \rho c \frac{\dot{V}}{V} - \rho \left(b_2 \lambda \frac{\dot{V}}{V} \right)^2, \quad (3.272)$$

where b_1 and b_2 are coefficients for the linear and quadratic terms, respectively. The quadratic term in Equation (3.272) is more important and is designed to “smear” a shock front across several elements. This term yields a jump in energy as a smeared shock passes, which simulates the shock heating. As a result, the smeared shock front can be propagated as a steady wave.

The linear term is intended to dissipate truncation frequency oscillations. The quadratic term is only applied to compressive strain rates, since an element cannot collapse in expansion.

The preceding expression is simplified if we use the undamped stable time increment defined by Equation (3.249) and write

$$\Delta \hat{t} = \frac{1}{c} = \sqrt{\frac{V^2}{2B_{ia}B_{ia}} \cdot \frac{\rho}{\lambda + 2\mu}}, \quad (3.273)$$

or

$$\Delta \hat{t} = \sqrt{\frac{m}{\lambda + 2\mu} \cdot \frac{V}{2B_{ia}B_{ia}}}, \quad (3.274)$$

where m is the element mass. We now define the factor ε such that the quadratic viscosity term vanishes in expansion

$$\varepsilon = b_1 - b_2^2 \Delta \hat{t} \min(0, D_{kk}). \quad (3.275)$$

This quantity is required for the damped stability criteria of Equation (3.252). Note that the condition imposed by (3.251) prevents Equation (3.275) from yielding so large a value of ε that Equation (3.252) would numerically yield a zero value.

We will show below that ε can be used to estimate the fraction of critical damping in the highest element mode. Using Equation (3.275) in Equation (3.274) allows us to write the viscous pressure as

$$q = (b_1 - b_2^2 \Delta \hat{t} D_{kk})(\lambda + 2\mu) \Delta \hat{t} D_{kk}. \quad (3.276)$$

The bulk viscosity pressure is appended to the stresses during the internal force calculations to yield the following forces:

$$f_{ia} = q B_{ia}. \quad (3.277)$$

The above expression can be expanded using Equations (3.274) and (3.275) to yield

$$f_{ia} = \varepsilon \rho c \frac{1}{V} B_{jb} B_{ia} \dot{u}_{jb}. \quad (3.278)$$

This form indicates that if B_{i_a} is an eigenvector, the modal damping is

$$\varepsilon \rho \frac{cV}{1} . \quad (3.279)$$

The critical damping estimate of the maximum element frequency is

$$2m\omega = 2 \frac{\rho V}{8} \frac{2c}{1} = \rho \frac{cV}{21} . \quad (3.280)$$

The two expressions above show that ε is half the fraction of critical damping in the highest mode.

Four-Node Corotational Shell

Introduction

In this section we discuss in detail some of the implementational issues associated with a frequently utilized structural element in nonlinear mechanics: the four-noded corotational shell element depicted in Figure 3.15. In so doing, we will add some important detail to the very conceptual discussion of shells and other structural entities given in Structural Components.

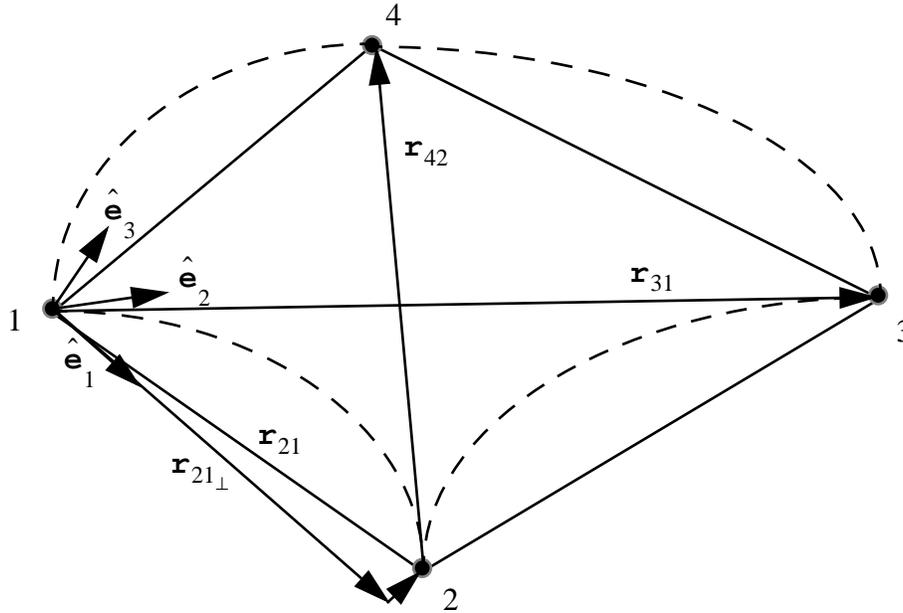


Figure 3.15 Four-noded, corotational shell element also showing the element coordinate system.

Although much of the discussion is equally applicable to matrix-free quasistatic solution strategies (see Conjugate Gradient Methods), We target our discussion here primarily to explicit dynamic calculations (see Explicit Finite Element Methods). In such settings the equations of motion of the deformable body thus become a system of uncoupled equations governing the nodal motions in the discretized system. For continuum elements each equation in the system is the equation for three-dimensional motion of a particle,

$$a_{iA} = (F_{iA}^{EXT} - F_{iA}^{INT}) / M_A, \quad (3.281)$$

where a_{iA} is the acceleration, F_{iA}^{EXT} and F_{iA}^{INT} are the external and internal forces, and M_A is the lumped mass, all associated with global Node A. For a continuum element with displacement degrees of freedom d_{iA} , a localized version of Eq. (3.281) completely describes the motion of the nodes.

By contrast, a shell element requires rotational degrees of freedom in addition to the displacement degrees of freedom. The additional equations governing these degrees of freedom are Euler's equations for the rotation of a rigid body about the principal axes, written here for an individual shell element

$$\begin{aligned}\alpha_{1b} &= [(m_{1b}^{\text{EXT}} - m_{1b}^{\text{INT}}) - (I_{3b} - I_{2b})\omega_{3b}\omega_{2b}] / I_{1b} \\ \alpha_{2b} &= [(m_{2b}^{\text{EXT}} - m_{2b}^{\text{INT}}) - (I_{1b} - I_{3b})\omega_{1b}\omega_{3b}] / I_{2b} \cdot \\ \alpha_{3b} &= [(m_{3b}^{\text{EXT}} - m_{3b}^{\text{INT}}) - (I_{2b} - I_{1b})\omega_{2b}\omega_{1b}] / I_{3b}\end{aligned}\quad (3.282)$$

In these equations α_{1b} is the angular acceleration for local Node b, ω_{1b} is angular velocity, and I_{1b} is the mass moment of inertia associated with local Node b in principal Direction 1. Similarly numerical subscripts 2 and 3 designate the other principal directions. A coordinate system is used at each node to track rotation of the principal axis system.

Shell Kinematics

In Mindlin shell theory (see [Mindlin, R.D., 1951]), the shell normals are assumed to remain straight, although they are allowed to rotate. Rotation of the normals allows the element to model transverse shear strains. Because displacements are assumed to vary linearly through the shell thickness, the velocity at any point can be expressed as

$$\mathbf{v}^* = \mathbf{v} + \hat{z}\hat{\mathbf{e}}_3 \times \hat{\mathbf{q}}, \quad (3.283)$$

in which \mathbf{v}^* is the velocity of a point in the shell body, \mathbf{v} is the velocity of the point on the midsurface lying on the same normal, and $\hat{\mathbf{q}}$ is the rotational velocity of the normal (see Figure 3.16). \hat{z} is the coordinate through the shell thickness along the unit normal vector, $\hat{\mathbf{e}}_3$. Vector components in the corotational coordinate system are indicated by the $\hat{\cdot}$ symbol.

The components of velocity strain in the corotational coordinate system are

$$\hat{d}_{ij} = \frac{1}{2} \left(\frac{\partial \hat{v}_i}{\partial \hat{x}_j} + \frac{\partial \hat{v}_j}{\partial \hat{x}_i} \right). \quad (3.284)$$

Strain-displacement relationships for the shell are obtained by substitution of Eq. (3.283) into Eq. (3.284), giving

Constitutive Assumptions

If the velocity strain components in the corotational system are arranged in a column matrix, \mathbf{d} , as

$$\mathbf{d} = [\hat{d}_{11}, \hat{d}_{22}, 2\hat{d}_{12}, 2\hat{d}_{23}, 2\hat{d}_{13}]^T, \quad (3.286)$$

then the conjugate Cauchy stress components in the corotated coordinate system can be written as

$$\mathbf{T} = [\hat{\sigma}_{11}, \hat{\sigma}_{22}, \hat{\sigma}_{12}, \hat{\sigma}_{23}, \hat{\sigma}_{13}]^T. \quad (3.287)$$

Furthermore, the shell is assumed to be in a state of plane stress, so

$$\hat{\sigma}_{33} = 0. \quad (3.288)$$

Note also the omission of \hat{d}_{33} , the rate of through-the-thickness thinning, from (3.285) and (3.286).

The representations of deformation and stress given in (3.286) and (3.287) are conjugate, in the sense that the stress power P , discussed in general continuum mechanical terms in Stress Power, can be expressed as

$$P = \mathbf{d}^T \hat{\mathbf{T}}. \quad (3.289)$$

Shell Element Coordinate Systems

We consider now the element formulation of a four-node, quadrilateral shell (see Figure 3.15), where the midsurface velocities \hat{v}_i and rotation rates \hat{c}_i are interpolated using the bilinear shape functions described in Shape Functions. Thickness of the shell is handled as an element attribute. The element uses reduced integration with hourglass control and is based on a corotational formulation (see [Belytschko, T., Lin, J.I. and Tsay, C.S., 1984] and [Belytschko, T., Ong, J.S.-J., Liu, W.K. and Kennedy, J.M., 1984])

Three coordinate systems are used in the shell element formulation. The translational equations of motion (Eq. (3.281)) are written in the global system with coordinates x_i and basis vectors \mathbf{e}_i . Strain-displacement relationships and constitutive equations are enforced in a local element coordinate system that rotates and translates with each element. This corotational coordinate system has orthonormal basis vectors denoted by $\hat{\mathbf{e}}_i$ and coordinates designated by \hat{x}_i . The

internal element force ($\hat{\mathbf{f}}$) and moment ($\hat{\mathbf{m}}$) resultants are also computed in this element coordinate system. New element coordinate systems are computed at each time step using current nodal coordinates. The equations governing the rotational motion are written in a local coordinate system at each node. These nodal coordinate systems are assumed to rotate with the principal axes at each node with the motion governed by Eqs. (3.282). The nodal systems have orthonormal basis vectors, $\bar{\mathbf{e}}_i$, with coordinates designated by x_i . The nodal coordinate systems are updated at each time step.

The element (corotational) normal vector $\hat{\mathbf{e}}_3$ is approximated by the normalized cross product of the element diagonals (see Figure 3.15)

$$\hat{\mathbf{e}}_3 = \frac{\mathbf{r}_{31} \times \mathbf{r}_{42}}{\|\mathbf{r}_{31} \times \mathbf{r}_{42}\|} \quad (3.290)$$

In (3.290) \mathbf{r}_{ab} is the position vector of element Node a relative to element Node b in the current configuration. The direction of the normal vector is thus determined by the element-nodal connectivity, and the positive side of the shell is the one for which the nodes are numbered counterclockwise. Next the direction of $\hat{\mathbf{e}}_1$ is taken as the portion of the vector connecting nNodes 1 and 2 that is orthogonal to $\hat{\mathbf{e}}_3$.

$$\mathbf{r}_{21\perp} = \mathbf{r}_{21} - (\mathbf{r}_{21} \cdot \hat{\mathbf{e}}_3) \hat{\mathbf{e}}_3 \quad (3.291)$$

$$\hat{\mathbf{e}}_1 = \mathbf{r}_{21\perp} / \|\mathbf{r}_{21\perp}\| \quad (3.292)$$

The final basis vector for the element coordinate system is obtained from the cross product of $\hat{\mathbf{e}}_3$ and $\hat{\mathbf{e}}_1$,

$$\hat{\mathbf{e}}_2 = \hat{\mathbf{e}}_3 \times \hat{\mathbf{e}}_1. \quad (3.293)$$

Internal force resultants at the nodes are computed from the stress gradient in the local element coordinate system (see (3.142) in Basics of Element Design); they are then transformed to the global coordinate system via

$$\begin{Bmatrix} \mathbf{f}_x^{int^e} \\ \mathbf{f}_y^{int^e} \\ \mathbf{f}_z^{int^e} \end{Bmatrix} = \begin{bmatrix} \hat{\mathbf{e}}_{1x} & \hat{\mathbf{e}}_{2x} & \hat{\mathbf{e}}_{3x} \\ \hat{\mathbf{e}}_{1y} & \hat{\mathbf{e}}_{2y} & \hat{\mathbf{e}}_{3y} \\ \hat{\mathbf{e}}_{1z} & \hat{\mathbf{e}}_{2z} & \hat{\mathbf{e}}_{3z} \end{bmatrix} \begin{Bmatrix} \hat{\mathbf{f}}_1^{int^e} \\ \hat{\mathbf{f}}_2^{int^e} \\ \hat{\mathbf{f}}_3^{int^e} \end{Bmatrix}. \quad (3.294)$$

Internal moment resultants are also computed in local element coordinates and transformed to global coordinates using Equation (3.294). However, Euler's equations for the rotational accelerations (see (3.282)) are written in nodal coordinates that rotate and translate with the node

in question (denoted here by superposed bars). The transformation from global to nodal coordinates is accomplished by

$$\begin{Bmatrix} \bar{m}_1 \\ \bar{m}_2 \\ \bar{m}_3 \end{Bmatrix} = \begin{bmatrix} \bar{e}_{1x} & \bar{e}_{1y} & \bar{e}_{1z} \\ \bar{e}_{2x} & \bar{e}_{2y} & \bar{e}_{2z} \\ \bar{e}_{3x} & \bar{e}_{3y} & \bar{e}_{3z} \end{bmatrix} \begin{Bmatrix} m_x \\ m_y \\ m_z \end{Bmatrix}. \quad (3.295)$$

Therefore, the complete transformation for moments from the element to nodal coordinates is

$$\{\bar{m}\} = [\bar{\mathbf{e}}]^T [\hat{\mathbf{e}}] \{\hat{m}\}. \quad (3.296)$$

Element Equations

The shell element is based on a four-node quadrilateral with bilinear interpolations of midsurface coordinates and of both translational and rotational velocities. Coordinates in the midsurface of the shell are approximated as

$$\mathbf{x}_i = \mathbf{x}_{ia} N_a(\mathbf{r}, \mathbf{s}), \quad (3.297)$$

where N_a are the shape functions, and \mathbf{r} and \mathbf{s} are the parent domain coordinates. Repeated indices, a , indicate summation over the four nodes of the element. Similarly the velocity of the midsurface and the angular velocity of the normal are interpolated as

$$\mathbf{v} = \mathbf{v}_a N_a(\mathbf{r}, \mathbf{s}), \quad (3.298)$$

and

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_a N_a(\mathbf{r}, \mathbf{s}), \quad (3.299)$$

using the same shape functions used for the midsurface coordinates.

In a similar fashion as described for the three-dimensional, constant stress element (see Eight-Node Uniform Strain Element), the shape functions can be expanded in terms of an orthogonal set of basis vectors as

$$\phi_a = \Sigma_a + \frac{1}{2} r \Lambda_{1a} + \frac{1}{2} s \Lambda_{2a} + r s \Gamma_a. \quad (3.300)$$

These basis vectors represent deformation modes of a unit square, analogous to their three-dimensional counterparts shown in Figure 3.14. Rigid body motion is represented by the first vector Σ_a . The volumetric basis vectors, Λ_{1a} and Λ_{2a} , can be combined to represent the normal and shear strain in an element. With the reduced integration formulation, the element area involves only these two vectors. Since the last vector, Γ_a , is neglected in the uniform strain

formulation, it represents spurious energy, or hourglass, modes for the element. Substitution of Eqs. (3.298) and (3.299) into Eqs. (3.285) yields the discretized strain-displacement relationships:

$$\begin{aligned}
\hat{d}_{11} &= \frac{1}{A} [B_{1a} \hat{v}_{1a} + \hat{z} B_{1a} \dot{q}_{2a}] \\
\hat{d}_{22} &= \frac{1}{A} [B_{2a} \hat{v}_{2a} - \hat{z} B_{2a} \dot{q}_{1a}] \\
2\hat{d}_{12} &= \frac{1}{A} [B_{2a} \hat{v}_{1a} + B_{1a} \hat{v}_{2a} + \hat{z} (B_{2a} \dot{q}_{2a} - B_{1a} \dot{q}_{1a})] \\
2\hat{d}_{23} &= \frac{1}{A} [B_{2a} \hat{v}_{3a} - N_a \dot{q}_{1a}] \\
2\hat{d}_{13} &= \frac{1}{A} [B_{1a} \hat{v}_{3a} + N_a \dot{q}_{2a}]
\end{aligned} \tag{3.301}$$

The gradient operator, $B_{\alpha a}$, is defined as

$$B_{\alpha a} = \int_{A_e} \frac{\partial \phi_a}{\partial x_\alpha} dA, \tag{3.302}$$

where Greek subscripts indicate the 1 and 2 coordinates in the plane of the element. For the reduced integration element, the gradient operator is needed only at the point $r = 0$, $s = 0$ and can be expressed in closed form in terms of the corotational coordinates of the element nodes (see [Belytschko, T., Lin, J.I. and Tsay, C.S., 1984]) as

$$[B_{\alpha a}] = \frac{1}{2} \begin{bmatrix} \hat{y}_2 - \hat{y}_4 & \hat{y}_3 - \hat{y}_1 & \hat{y}_4 - \hat{y}_2 & \hat{y}_1 - \hat{y}_3 \\ \hat{x}_4 - \hat{x}_2 & \hat{x}_1 - \hat{x}_3 & \hat{x}_2 - \hat{x}_4 & \hat{x}_3 - \hat{x}_1 \end{bmatrix}. \tag{3.303}$$

Because the element is a quadrilateral, its area, A , can also be expressed in terms of nodal coordinates:

$$A = \frac{1}{2} [(\hat{x}_3 - \hat{x}_1)(\hat{y}_4 - \hat{y}_2) + (\hat{x}_2 - \hat{x}_4)(\hat{y}_3 - \hat{y}_1)]. \tag{3.304}$$

Computation of Internal Force and Moment Resultants

Representation of the rate of internal energy for a shell element in the global variational principle requires derivation of the internal force and moment resultants. Let the velocity vector for an element node be defined as

$$\hat{\mathbf{v}}_a = \begin{Bmatrix} \hat{v}_{1a} \\ \hat{v}_{2a} \\ \hat{v}_{3a} \\ \hat{q}_{1a} \\ \hat{q}_{2a} \end{Bmatrix} \quad (3.305)$$

in corotational coordinates of the element. Then the corresponding internal element force and moment vector is

$$\hat{\mathbf{f}}_a = \begin{Bmatrix} \hat{f}_{1a} \\ \hat{f}_{2a} \\ \hat{f}_{3a} \\ \hat{m}_{1a} \\ \hat{m}_{2a} \end{Bmatrix}. \quad (3.306)$$

The concept of stress power, discussed for a general continuum in Stress Power, can be utilized here to define the vector $\hat{\mathbf{f}}_{ia}$ via

$$\text{Element stress power} = \int_{V_e} T_{ij} v_{i,j} dV = \int_{V_e} \hat{\mathbf{d}}^T \hat{\mathbf{T}} dV. \quad (3.307)$$

Substitution of Eqs. (3.285) into (3.307) and using one-point quadrature leads to the following expressions for the element internal forces and moments:

$$\begin{aligned} \hat{f}_{1a} &= B_{1a} \hat{f}_{11} + B_{2a} \hat{f}_{12} \\ \hat{f}_{2a} &= B_{2a} \hat{f}_{22} + B_{1a} \hat{f}_{12} \\ \hat{f}_{3a} &= B_{1a} \hat{f}_{13} + B_{2a} \hat{f}_{23} \\ \hat{m}_{1a} &= -B_{2a} \hat{m}_{22} - B_{1a} \hat{m}_{12} - \frac{A}{4} \hat{f}_{yz} \\ \hat{m}_{2a} &= B_{1a} \hat{m}_{11} + B_{2a} \hat{m}_{12} + \frac{A}{4} \hat{f}_{xz} \\ \hat{m}_{3a} &= 0. \end{aligned} \quad (3.308)$$

with

$$\begin{aligned}
\hat{f}_{\alpha\beta} &= \int \hat{T}_{\alpha\beta} d\hat{z} \\
\hat{f}_{\alpha z} &= \kappa \int \hat{T}_{\alpha z} d\hat{z}, \\
\hat{m}_{\alpha\beta} &= \int \hat{z} \hat{T}_{\alpha\beta} d\hat{z}
\end{aligned} \tag{3.309}$$

where $\kappa = 5/6$ is the shear factor from classical plate theory. The integrals of stress over the shell thickness are computed analytically for linear elastic materials. For nonlinear materials the internal force and moment resultants are computed by numerical integration through the element thickness (h) using the trapezoid rule. Currently the user may use either three or five integration points; the first point is at $\hat{z} = -h/2$, the middle point is at $\hat{z} = 0$ (the midsurface), the last point is at $\hat{z} = +h/2$.

Hourglass Control

Reduced integration of the stress divergence leads to spurious zero-energy modes (hourglass modes) in the element, as discussed, in general, in Hourglass Control. The hourglass control algorithm implemented for the shell element is one developed by [Flanagan, D.P. and Belytschko, T., 1981] and by [Belytschko, T., Ong, J.S.-J., Liu, W.K. and Kennedy, J.M., 1984]. Removal of the linear portion of the nodal velocity results in the definition of hourglass shape vectors γ_a :

$$\gamma_a = \Gamma_a - \frac{1}{A} B_{\alpha a} x_{\alpha b} \Gamma_b, \tag{3.310}$$

and the corresponding hourglass forces and moments

$$\begin{aligned}
\hat{f}_{\alpha a}^{\text{HG}} &= Q_\alpha \gamma_a \\
\hat{f}_{3a}^{\text{HG}} &= Q_3 \gamma_a \\
\hat{m}_{\alpha a}^{\text{HG}} &= P_\alpha \gamma_a
\end{aligned} \tag{3.311}$$

with the generalized hourglass stresses given by

$$\begin{aligned}
Q_\alpha &= C_1 \gamma_b \hat{v}_{\alpha b} \\
Q_3 &= C_2 \gamma_b \hat{v}_{3b}, \\
P_\alpha &= C_3 \gamma_b \dot{q}_{\alpha b}
\end{aligned} \tag{3.312}$$

and the hourglass stiffnesses defined via

$$\begin{aligned}
C_1 &= r_m \frac{Eh}{8A} B_{\alpha a} B_{\alpha a} \\
C_2 &= r_z \frac{Gh^3}{12A^2} B_{\alpha a} B_{\alpha a}, \\
C_3 &= r_\theta \frac{Eh^3}{192A} B_{\alpha a} B_{\alpha a}
\end{aligned} \tag{3.313}$$

where E and G are Young's modulus and the shear modulus, respectively; and h is the element thickness. In the preceding equations the $\hat{f}_{\alpha a}^{HG}$ represent the nodal hourglass membrane forces associated with in-plane velocities \hat{v}_α ; \hat{f}_{3a}^{HG} , the hourglass bending forces associated with out-of-plane velocity \hat{v}_3 ; and $\hat{m}_{\alpha a}^{HG}$, the hourglass bending moments associated with rotational velocities $\dot{\alpha}_\alpha$. The corresponding hourglass parameters r_m , r_z , and r_θ are usually assigned values ranging from 0.01 to 0.05.

Calculation of the Stable Time Increment

The central difference operator is conditionally stable with the stability limit for a system with no damping given by

$$\Delta t \leq \frac{2}{\omega_{\max}}, \tag{3.314}$$

where ω_{\max} is the maximum frequency of the system (see also Explicit Finite Element Methods). The maximum frequency in the system can be bounded by the maximum element frequency, so the stability limit becomes

$$\Delta t \leq \frac{2}{\text{MAXIMUM}(\omega_{\max}^e)}, \tag{3.315}$$

where $\text{MAXIMUM}(\omega_{\max}^e)$ is the maximum element frequency of all the elements in the system.

Conservative estimates of the maximum frequency for quadrilateral shell elements were developed by Belytschko and Lin [Belytschko, T. and Lin, J.I., 1985] and are given for membrane, bending, and shear deformation as

$$\begin{aligned}\omega_{\max, m}^2 &= \frac{12D}{MAh^2}(R_1 + \sqrt{R_1^2 - 16(1 - \nu^2)A^2}) \\ \omega_{\max, z}^2 &= \frac{h^2}{12\alpha} \omega_{\max, m}^2 \\ \omega_{\max, \theta}^2 &= \frac{4}{M} \left(\frac{c_s \alpha}{A} + \frac{c_s A}{4\alpha} \right)\end{aligned}\tag{3.316}$$

with

$$\begin{aligned}c_s &= \kappa Gh \\ R_1 &= 2B_{\alpha a} B_{\alpha a} \\ R_2 &= \sqrt{R_1^2 - 16A^2} \\ \alpha_1 &= (R_1 + R_2)/4 \\ D &= \frac{Eh^3}{12(1 - \nu^2)}\end{aligned}\tag{3.317}$$

where E is Young's modulus, G is the shear modulus, ν is Poisson's Ratio, h is the element thickness, M is the element mass, and κ is the shear correction factor.

In the preceding expressions for maximum element frequency, α is a rotational inertia scaling factor assumed to be

$$\alpha = \frac{I}{A_c},\tag{3.318}$$

where I and A_c are the mass moment of inertia and area, respectively, of the cross section. α is approximated using the value for a flat, rectangular element:

$$\alpha = \frac{h^2 + A}{12}.\tag{3.319}$$

The maximum stable time step is computed using the maximum frequency over all shell and brick elements.

Constitutive Models

Two plane stress constitutive models that are widely implemented for the shell element described here are described next: an elastic model and an elastic-plastic model with combined linear hardening. In corotational coordinates the stress rate follows directly from the velocity strain.

Since all stress and strain quantities are computed in corotational coordinates, the $\hat{\cdot}$ notation has been dropped.

For a plane stress, linear elastic model, the stress rate is computed from the velocity strain as

$$\dot{\mathbf{T}} = \lambda'(\text{tr}\dot{\mathbf{d}})\mathbf{I} + 2\mu\dot{\mathbf{d}}, \quad (3.320)$$

where μ is the shear modulus.

$$\mu = G = \frac{E}{2(1 + \nu)}, \quad (3.321)$$

λ' is the Lamé constant for plane stress

$$\lambda' = \frac{E\nu}{(1 - \nu^2)} \quad (3.322)$$

$$(\text{tr}\dot{\mathbf{d}}) = \dot{d}_{ii}, \quad (3.323)$$

and \mathbf{I} is the second-order identity tensor.

Material properties required as input are Young's modulus E and Poisson's ratio ν , from which the above Lamé parameters can be calculated. There are no internal state variables.

In treatment of elastic-plastic materials, on the other hand, it should be first noted that internal force and moment resultants for nonlinear materials are computed by numerical integration through the element thickness, so the constitutive model must be evaluated at each integration point. The general theory for elastic-plastic materials, with combined isotropic and kinematic hardening, is discussed in detail in Constitutive Modeling.

Here we concentrate on the implementational details of plane stress radial return, as they differ from the fully three-dimensional situation. For plane stress, d_{33} must be computed from the constraint on the constitutive model rather than directly from the finite element equations. This constraint requires iterations on the radial return algorithm. The secant approach, presented by [Hallquist, J.O. and Benson, D.J., 1986] and assessed for accuracy and efficiency by [Whirley, R.G., Hallquist, J.O. and Goudreau, G.L., 1988], is described below.

Because the plane stress assumption only affects the volumetric strains, the trial shear stresses can be computed outside the iteration loop. The trial shear stresses are

$$T_{12}^T = T_{12} + 2\mu\Delta t d_{12} \quad (3.324)$$

$$T_{23}^T = T_{23} + 2\mu\Delta t d_{23} \quad (3.325)$$

$$T_{23}^T = T_{13} + 2\mu\Delta t d_{13}, \quad (3.326)$$

where \bar{d}_{ij} is the rate of deformation at the integration point computed from Eq. (3.285). The stress difference tensor ξ_{ij} is defined as the difference between the deviatoric stress tensor S_{ij} and the back stress tensor α_{ij} ,

$$\xi_{ij} = S_{ij} - \alpha_{ij} \quad (3.327)$$

$$S_{ij} = T_{ij} + p\delta_{ij} \quad (3.328)$$

$$p = -\frac{1}{3}T_{ii} \quad (3.329)$$

The shear stress differences are

$$\xi_{12}^T = T_{12}^T - \alpha_{12} \quad (3.330)$$

$$\xi_{23}^T = T_{23}^T - \alpha_{23} \quad (3.331)$$

$$\xi_{13}^T = T_{13}^T - \alpha_{13}. \quad (3.332)$$

The iteration loop is entered for computation of the volumetric stress difference and the equivalent plastic strain. For iteration i the volumetric velocity strain is

$$(\text{tr}\mathbf{d})^{(i)} = d_{11} + d_{22} + d_{33}^{(i)}. \quad (3.333)$$

The trial normal stresses and pressure then follow as:

$$T_{11}^T = T_{11} + \lambda\Delta t(\text{tr}\mathbf{d})^{(i)} + 2\mu\Delta t d_{11} \quad (3.334)$$

$$T_{22}^T = T_{22} + \lambda\Delta t(\text{tr}\mathbf{d})^{(i)} + 2\mu\Delta t d_{22} \quad (3.335)$$

$$T_{33}^T = \lambda\Delta t(\text{tr}\mathbf{d})^{(i)} + 2\mu\Delta t d_{33} \quad (3.336)$$

$$p^T = -\frac{1}{3}(T_{11}^T + T_{22}^T + T_{33}^T). \quad (3.337)$$

The first two normal components of the stress difference computed from Eq. (3.327) are:

$$\xi_{11}^T = T_{11}^T + p^T - \alpha_{11} \quad (3.338)$$

$$\xi_{22}^T = T_{22}^T + p^T - \alpha_{22}. \quad (3.339)$$

Because the stress difference tensor is deviatoric, the third normal component is given by

$$\xi_{33}^T = -(\xi_{11}^T + \xi_{22}^T). \quad (3.340)$$

The increment in equivalent plastic strain for the i^{th} iteration is

$$\Delta\gamma^{(i)} = \frac{1}{2\mu\left(1 + \frac{H'}{3\mu}\right)} (\sqrt{\xi_{ij}^T \xi_{ij}^T} - R), \quad (3.341)$$

where R is the radius of the yield surface from the previous time step,

$$R = \sqrt{\xi_{ij} \xi_{ij}}. \quad (3.342)$$

The hardening slope H' depends on Young's modulus and the plastic modulus E_p ,

$$H' = \frac{E_p E}{E - E_p}. \quad (3.343)$$

The normal stress difference in the thickness direction follows from the radial return algorithm as

$$T_{33}^{(i)} = T_{33}^T - 2\mu\Delta\gamma\beta\xi_{33}^T / \sqrt{\xi_{ij}^T \xi_{ij}^T}, \quad (3.344)$$

where β is a scalar parameter ranging from 0 to 1 that determines the relative amounts of isotropic and kinematic hardening. For $\beta = 1$ all hardening is assumed to be isotropic. At the other extreme $\beta = 0$ means only kinematic hardening is present. Finally, an estimate for $d_{33}^{(i+1)}$ is obtained from a secant update,

$$d_{33}^{(i+1)} = d_{33}^{(i-1)} + T_{33}^{(i)}(d_{33}^{(i)} - d_{33}^{(i-1)}) / (T_{33}^{(i)} - T_{33}^{(i-1)}). \quad (3.345)$$

The $(i+1)$ iteration proceeds by substituting the new value of d_{33} into (3.333) and repeating Eqs. (3.334) through (3.344) until σ_{33} has converged to zero.

Because the plane stress algorithm is based on a secant update, two starting values are required for d_{33} . The two starting values are

$$d_{33}^{(0)} = -\frac{\nu}{(1-\nu)}(d_{11} + d_{22}), \quad (3.346)$$

which assumes a completely elastic step, and

$$d_{33}^{(1)} = -(d_{11} + d_{22}) \quad (3.347)$$

based on a completely plastic (incompressible) step.

Once the algorithm has converged, the yield surface radius, equivalent plastic strain, and back stresses are updated as

$$\mathbf{R}^{t+\Delta t} = \mathbf{R}^t + \frac{2}{3}\beta\mathbf{H}'\Delta\boldsymbol{\gamma} \quad (3.348)$$

$$\bar{\mathbf{d}}^{p1\ t+\Delta t} = \bar{\mathbf{d}}^{p1\ t} + \sqrt{\frac{2}{3}}\Delta\boldsymbol{\gamma} \quad (3.349)$$

$$\boldsymbol{\alpha}_{ij}^{t+\Delta t} = \boldsymbol{\alpha}_{ij}^t + \frac{2}{3}(1-\beta)\mathbf{H}'\boldsymbol{\gamma}\xi_{ij}. \quad (3.350)$$

Finally, the stresses are updated using radial return

$$\boldsymbol{\alpha}_{ij}^{t+\Delta t} = \mathbf{T}_{ij}^T - 2\mu\Delta\boldsymbol{\gamma}\beta\xi_{ij}^T / \sqrt{\xi_{k1}^T \xi_{k1}^T}. \quad (3.351)$$

Time-Stepping Algorithm

In a central difference algorithm to integrate the equations of motion, such as is used in explicit dynamics, the translational variables are handled just as they would be for an ordinary continuum. Once the nodal accelerations at time t are solved from Eq. (3.281), the velocities and displacements in global coordinates follow as

$$\begin{aligned} \mathbf{v}^{t+\frac{\Delta t}{2}} &= \mathbf{v}^{t-\frac{\Delta t}{2}} + \Delta t \mathbf{a}^t \\ \mathbf{d}^{t+\Delta t} &= \mathbf{d}^t + \Delta t \mathbf{v}^{t+\frac{\Delta t}{2}} \end{aligned} \quad (3.352)$$

Angular accelerations at time t for nodes connected to shell elements are computed in coordinate systems that are local to each node. Because these coordinate systems rotate with the nodes, the angular accelerations cannot be integrated directly for updating the configuration. Instead the procedure of Belytschko and coworkers (see [Belytschko, T., Lin, J.I. and Tsay, C.S., 1984]) is implemented.

The nodal rotations are updated from the angular accelerations in the same manner as the translational velocities,

$$\boldsymbol{\omega}^{t+\frac{\Delta t}{2}} = \boldsymbol{\omega}^{t-\frac{\Delta t}{2}} + \Delta t \boldsymbol{\alpha}^t. \quad (3.353)$$

The updated nodal basis vectors are then

$$\begin{aligned} \bar{\mathbf{e}}_i^{t+\Delta t} &= \bar{\mathbf{e}}_i^t + \Delta t \dot{\bar{\mathbf{e}}}_i \\ &= \bar{\mathbf{e}}_i^t + \Delta t \left(\boldsymbol{\omega}^{t+\frac{\Delta t}{2}} \times \bar{\mathbf{e}}_i^t \right). \end{aligned} \quad (3.354)$$

For $\bar{\mathbf{e}}_3^{t+\Delta t}$, the preceding equation becomes

$$\bar{\mathbf{e}}_3^{t+\Delta t} = \bar{\mathbf{e}}_3^t + \Delta t \left(\omega_{\bar{y}}^{t+\frac{\Delta t}{2}} \bar{\mathbf{e}}_1^t - \omega_{\bar{x}}^{t+\frac{\Delta t}{2}} \bar{\mathbf{e}}_2^t \right). \quad (3.355)$$

The scalar product of $\bar{\mathbf{e}}_3^{t+\Delta t}$ and $\bar{\mathbf{e}}_1^t$ gives

$$\bar{e}_{3_{\bar{x}}}^{t+\Delta t} = \Delta t \omega_{\bar{y}}^{t+\frac{\Delta t}{2}}. \quad (3.356)$$

Similarly the scalar product of $\bar{\mathbf{e}}_3^{t+\Delta t}$ and $\bar{\mathbf{e}}_2^t$ gives the \bar{y} component of $\bar{\mathbf{e}}_3^{t+\Delta t}$.

$$\bar{e}_{3_{\bar{y}}}^{t+\Delta t} = -\Delta t \omega_{\bar{x}}^{t+\frac{\Delta t}{2}}. \quad (3.357)$$

The third component of $\bar{\mathbf{e}}_3^{t+\Delta t}$ is found by normality. The rotation over a single time step is assumed to be small so second-order terms are dropped; and the \bar{z} component of the updated vector is

$$\bar{e}_{3_{\bar{z}}}^{t+\Delta t} = \sqrt{1 - (\bar{e}_{3_{\bar{x}}}^{t+\Delta t})^2 - (\bar{e}_{3_{\bar{y}}}^{t+\Delta t})^2}. \quad (3.358)$$

Next $\bar{\mathbf{e}}_1$ is updated. From Equation (3.354)

$$\bar{\mathbf{e}}_1^{t+\Delta t} = \bar{\mathbf{e}}_1^t + \Delta t \left(\omega_{\bar{z}}^{t+\frac{\Delta t}{2}} \bar{\mathbf{e}}_2^t - \omega_{\bar{y}}^{t+\frac{\Delta t}{2}} \bar{\mathbf{e}}_3^t \right). \quad (3.359)$$

The scalar product of Equation (3.359), with $\bar{\mathbf{e}}_2^t$, gives the \bar{y} component of $\bar{\mathbf{e}}_1$ as

$$\bar{e}_{1_{\bar{y}}}^{t+\Delta t} = \Delta t \omega_{\bar{z}}^{t+\frac{\Delta t}{2}}. \quad (3.360)$$

If it is assumed that $\bar{e}_{1_{\bar{x}}}^{t+\Delta t}$ is approximately one (i.e., small rotations over the time step)

orthogonality of $\bar{\mathbf{e}}_1^{t+\Delta t}$ and $\bar{\mathbf{e}}_3^{t+\Delta t}$ yields

$$\bar{e}_{1_{\bar{z}}}^{t+\Delta t} = -\frac{(\bar{e}_{3_{\bar{x}}}^{t+\Delta t} + \bar{e}_{1_{\bar{y}}}^{t+\Delta t} \bar{e}_{3_{\bar{y}}}^{t+\Delta t})}{\bar{e}_{3_{\bar{z}}}^{t+\Delta t}}, \quad (3.361)$$

again neglecting the second-order terms. The third component of $\bar{\mathbf{e}}_1^{t+\Delta t}$ is then found by imposing unit length on the vector, so

$$\bar{e}_{1\bar{x}}^{t+\Delta t} = \sqrt{1 - (\bar{e}_{1\bar{y}}^{t+\Delta t})^2 - (\bar{e}_{1\bar{z}}^{t+\Delta t})^2}. \quad (3.362)$$

The cross product of $\bar{\mathbf{e}}_3^{t+\Delta t}$ with $\bar{\mathbf{e}}_1^{t+\Delta t}$ gives $\bar{\mathbf{e}}_2^{t+\Delta t}$ to complete the update of the basis vectors for the nodal coordinate system.

Bibliography

- Adley, M.D. and Moxley, R.E., 1996 *PENCURV/ABAQUS: A Simply Coupled Penetration Trajectory/Structural Dynamics Model for Deformable Projectiles Impacting Complex Curvilinear Targets*, Technical Report SL-96-6, U.S. Army Engineer Waterways Experiment Station, Vicksburg, MS.
- American National Standards Institute, 1978 *American National Standards Programming Language FORTRAN ANSI X3.9-1978*, American National Standards Institute, New York.
- American Society for Testing and Materials, 19xx *Annual Book of ASTM Standards: Section 3 - Metals Test Methods and Analytical Procedures*, American Society for Testing and Materials, Philadelphia, Pennsylvania.
- Attaway, S.W., 1990 *Update of PRONTO 2D and PRONTO 3D Transient Solid Dynamics Program*, SAND90-0102, Sandia National Laboratories, Albuquerque, NM.
- Balmer, H.A. and Witmer, E.A., 1964 *Theoretical-Experimental Correlation of Large Dynamic and Permanent Deformation of Impulsively Loaded Simple Structures*, FDP-TDR-64-108, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, OH.
- Bartels, R. and Daniel, J.W., 1973 "A conjugate gradient approach to nonlinear elliptic boundary value problems in irregular regions," in *Lecture Notes in Mathematics*, (A. Dold and B. Eckmann, eds.), Springer-Verlag, New York, pp. 1-11.
- Bathe, K.J. and Wilson, E.L., 1976 *Numerical Methods in Finite Element Analysis*, Prentice-Hall, Inc., New Jersey.
- Beisinger, Z.E., 1984 *SEACO: Sandia Engineering Analysis Department Code Output Data Base*, SAND84-2004, Sandia National Laboratories, Albuquerque, NM.
- Belytschko, T. and Bindelman, L.P., 1993 "Assumed strain stabilization of the eight node hexahedral element," *Computer Methods in Applied Mechanics and Engineering*, Vol. 105, pp. 225-260.
- Belytschko, T. and Lin, J.I., 1985 "Eigenvalues and Stable Time Step for the Bilinear Mindlin Plate Element," *International Journal for Numerical Methods in Engineering*, Vol. 21, pp. 1729-1745.
- Belytschko, T. and Lin, J.I., 1987 "A Three-Dimensional Impact-Penetration Algorithm with Erosion," *Computers and Structures*, Vol. 25, No. 1, pp. 95-104.
- Belytschko, T., Lin, J.I. and Tsay, C.S., 1984 "Explicit Algorithms for the Nonlinear Dynamics of Shells," *Computer Methods in Applied Mechanics and Engineering*, Vol. 42, pp. 225-251.

- Belytschko, T. and Marchertas, A.H., 1974 “Nonlinear Finite Element Method for Plates and its Application to the Dynamic Response of Reactor Fuel Subassemblies,” *Journal of Pressure Vessel Technology*, ASME, pp. 251-157.
- Belytschko, T. and Neal, M.O., 1991 “Contact-Impact by the Pinball Algorithm with Penalty and Lagrangian Methods,” *International Journal Numerical Methods of Engineering*, Vol. 31, pp. 547-572.
- Belytschko, T., Ong, J.S.-J., Liu, W.K. and Kennedy, J.M., 1984 “Hourglass Control in Linear and Nonlinear Problems,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 43, pp. 251-276.
- Belytschko, T., Schwer, L. and Klein, M.J., 1977 “Large Displacement Transient Analysis of Space Frames,” *International Journal for Numerical Methods in Engineering*, Vol. 11, pp. 65-84.
- Benson, D.J. and Hallquist, J.O., 1990 “A Single Surface Contact Algorithm for the Post-Buckling Analysis of Structures,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 78, pp. 141-163.
- Benz, W., 1988 “Applications of Smooth Particle Hydrodynamics (SPH) to Astrophysical Problems,” *Computational Physics Communications*, Vol. 48, pp. 97-105.
- Benz, W., 1990 “Smooth Particle Hydrodynamics: A Review,” in *The Numerical Modeling of Stellar Pulsation*, (J.R. Buchler, ed.), Kluwer, Dordrecht, Netherlands, p. 269.
- Bergmann, V.L., 1991 *Transient Dynamic Analysis of Plates and Shells with PRONTO 3D*, SAND91-1182, Sandia National Laboratories, Albuquerque, NM.
- Bertsekas, D.P., 1982 *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York.
- Biffle, J.H., 1981 *JAC--A Two-Dimensional Finite Element Computer Program for the Nonlinear Response of Solids with the Conjugate Gradient Method*, SAND81-0998, Sandia National Laboratories, Albuquerque, NM.
- Biffle, J.H., 1984 *JAC - A Two-Dimensional Finite Element Computer Program for the Nonlinear Quasi-Static Response of Solids with the Conjugate Gradient Method*, SAND81-0998, Sandia National Laboratories, Albuquerque, NM.
- Biffle, J.H., 1993 *JAC3D - A Three-Dimensional Finite Element Computer Program for the Nonlinear Quasi-Static Response of Solids with the Conjugate Gradient Method*, SAND87-1305, Sandia National Laboratories, Albuquerque, NM.

- Biffle, J.H. and Gubbels, M.H., 1976 *WULFF - A Set of Computer Programs for the Large Displacement Dynamic Response of Three Dimensional Solids*, SAND76-0096, Sandia National Laboratories, Albuquerque, NM.
- Biffle, J.H. and Stone, C.M., 1989 "Personal communications," Applied Mechanics Division, Sandia National Laboratories, Albuquerque, NM.
- Bishop, R.F., Hill, R. and Mott, N.F., 1945 "The theory of indentation and hardness", *Proceedings of the Royal Society*, Vol. 57 (3), pp. 147-159.
- Blacker, T.D., 1988 *FASTQ Users Manual Version 1.2*, SAND88-1326, Sandia National Laboratories, Albuquerque, NM.
- Boaz, M.L., 1966 *Mathematical Methods in the Physical Sciences*, John Wiley and Sons, New York, NY.
- Budiansky, B. and O'Connell, R.J., 1976 "Elastic moduli of a cracked solid," *Computer Methods in Applied Mechanics and Engineering*, Vol. 12, pp. 81-97.
- CRAY Research, Inc., 1989 *Volume 3: UNICOS Math and Scientific Library Reference Manual*, SR-2081 5.0.
- Campbell, P.M., 1988 *Some New Algorithms for Boundary Values Problems in Smooth Particle Hydrodynamics*, DNA-TR-88-286.
- Carpenter, N.J., Taylor, R.L. and Katona, M.G., 1991 "Lagrange Constraints for Transient Finite Element Surface Contact," *International Journal for Numerical Methods in Engineering*, Vol. 32, 103-128.
- Chait, R., 1972 "Factors influencing the strength differential of high strength steels", *Metalurgical Transactions A*, Vol. 3, pp. 365-371.
- Chakarabarty, J., 1987 *Theory of Plasticity*, McGraw-Hill Book Company, New York, pp. 306-315, 342-350.
- Chaudhary, A.B. and Bathe, K.J., 1986 "A Solution Method for Static and Dynamic Analysis of Three-Dimensional Contact Problems with Friction," *Computers and Structures*, Vol. 24, No. 6, pp. 855-873.
- Cheung, C.Y. and Cebon, D., 1997 "Experimental Study of Pure Bitumens in Tension, Compression, and Shear," *Journal of Rheology*, Vol. 41, No. 1, pp. 45-73.
- Chou, T.S., 1989 *The Dynamic Response at High-Explosive Inert-Solid Interface as Predicted by Finite Difference/Element Computer Programs*, Tech. Rep. MLM-3562, EG&G Mound Applied Technologies, Miamisburg, OH.

- Cloutman, L.D., 1990(a) *Basics of Smoothed Particle Hydrodynamics*, Lawrence Livermore National Laboratory, Livermore, CA, report UCRL-ID-103698.
- Cloutman, L.D., 1990(b) "An Evaluation of Smoothed Particle Hydrodynamics," *Proceedings of The NEXT Free-Lagrange Conference*, Jackson Lake Lodge, Moran, WY, June 3-7.
- Cohen, M. and Jennings, P.C., 1983 "Silent Boundary Methods," in *Computational Methods for Transient Analysis*, (Belytschko, T. and Hughes, T.J.R., eds.), North-Holland.
- Cole, R.H., 1965 *UnderWater Explosions*, Dover Publications.
- Concus, P., Golub, G.H. and O'Leary, D.P., 1976 "A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations," in *Sparse Matrix Computations*, (J.R. Bunch and D.J. Rose, eds.), Academic Press, New York, pp. 309-332.
- Courant, R., Friedrichs, K.O. and Lewy, H., 1928 *Mathematical Annotations*, Vol. 100, p. 32.
- Crismann, J.M. and Zapas, L.J., 1979 "Creep Failure and Fracture of Polyethylene in Uniaxial Extension," *Polymer Engineering Science*, Vol. 19, pp. 99-103.
- Curnier, A. and Alart, P., 1988 "A Generalized Newton Method for Contact Problems with Friction," *Journal de Mecanique Theorique et Appliquee*, Vol. 7, 67-82.
- Daniel, J.W., 1967 "The conjugate gradient method for linear and nonlinear operator equations," *SIAM Journal of Numerical Analysis*, Vol. 4, no. 1, pp. 10-26.
- Dennis, J.E. and Schnabel, R.B., 1996 *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Society for Industrial and Applied Mathematics, Philadelphia.
- Dienes, J.K., 1979 "On the analysis of rotation and stress rate in deforming bodies," *Acta Mechanica*, Vol. 32, pp. 217-232.
- Dobratz, B.M., 1981 *LLNL Explosives Handbook, Properties of Chemical Explosives and Explosive Simulants*, UCRL-52997, Lawrence Livermore National Laboratory, Livermore, CA.
- Duffey, T.A. and Macek, R.W., 1997 "Non-normal impact of earth penetrators", *Proceedings of the International Symposium on Penetration and Impact Problems (ICES'97)*, San Jose, Costa Rica.
- Engleman, R. and Jaeger, Z., 1987 *Theoretical Aids for Improvement of Blasting Efficiencies in Oil Shale and Rocks*, Tech. Rep AP-TR-12/87, Soreq Nuclear Research Center, Yavne, Israel.
- Ferry, 1950 ???

- Flanagan, D.P. and Belytschko, T., 1981 "A Uniform Strain Hexahedron and Quadrilateral with Orthogonal Hourglass Control," *International Journal for Numerical Methods in Engineering*, Vol. 17, pp. 679-607.
- Flanagan, D.P. and Belytschko, T., 1984 "Eigenvalues and Stable Time Steps for the Uniform Hexahedron and Quadrilateral," *Journal of Applied Mechanics*, Vol. 51, pp. 35-40.
- Flanagan, D.P., Mills-Curran, W.C. and Taylor, L.M., 1986 *SUPES - A Software Utilities Package for the Engineering Sciences*, SAND86-0911, Sandia National Laboratories, Albuquerque, NM.
- Flanagan, D.P. and Taylor, L.M., 1987 "An accurate numerical algorithm for stress integration with finite rotations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 62, pp. 305-320.
- Flanagan, D.P. and Taylor, L.M., 1987 "On the Analysis of Rotation and Stress Rate in Deforming Bodies," *Computer Methods in Applied Mechanics and Engineering*, Vol. 62, pp. 305-320.
- Fletcher, R. and Reeves, C.M., 1964 "Function minimization by conjugate gradients," *The Computer Journal*, Vol. 7, pp. 149-154.
- Forrestal, M.J., Altman, B.S., Cargile, J.D. and Hanchak, S.J., 1994 "An empirical equation for penetration depth of ogive-nose projectiles into concrete targets", *International Journal of Impact Engineering*, Vol. 15, pp. 395-405.
- Forrestal, M.J., Brar, N.S. and Luk, V.K., 1991 "Penetration of strain-hardening targets with rigid spherical-nose rods", *ASME Journal of Applied Mechanics*, Vol. 58, pp. 7-10.
- Forrestal, M.J. and Luk, V.K., 1991 "Penetration into soil targets", *International Journal of Impact Engineering*, Vol. 12, pp. 427-444.
- Forrestal, M.J., Okajima, K. and Luk, V.K., 1988 "Penetration of 6061-T651 aluminum targets with rigid long rods", *ASME Journal of Applied Mechanics*, Vol. 55, pp. 755-760.
- Forrestal, M.J. and Tzou, D.Y., 1996 "A spherical cavity-expansion penetration model for concrete targets", *International Journal of Solids Structures* (accepted).
- Forrestal, M.J., Tzou, D.Y., Askari, E. and Longcope, D.B., 1995 "Penetration into ductile metal targets with rigid spherical-nose rods", *International Journal of Impact Engineering*, Vol. 16, pp. 699-710.
- Freudenthal, A.M., 1963 ???
- Frew, D.J., Hanchak, S.J., Green, M.L. and Forrestal, M.J., 1997 "Penetration of concrete targets with ogive-nose steel rods", *International Journal of Impact Engineering*, (submitted).

- Frost and Ashby, 1977 *Deformation Mechanisms Maps*, ???.
- Fung, Y.C., 1965 *Foundations of Solid Mechanics*, Prentice-Hall, Englewood Cliffs, New Jersey.
- Fung, Y.C., 1977 *A First Course in Continuum Mechanics*, 2nd edition, Prentice-Hall, Englewood Cliffs, New Jersey.
- Gallego, F.J. and Anza, J.J., 1989 "A Mixed Finite Element Model for the Elastic Contact Problem," *International Journal for Numerical Methods in Engineering*, Vol. 28, 1249-1264.
- Giannakopoulos, A.E., 1989 "The Return Mapping Method for the Integration of Friction Constitutive Relations," *Computers and Structures*, Vol. 32, 157-167.
- Gilkey, A.P., 1988 *ALGEBRA - A Program that Algebraically Manipulates the Output of a Finite Element Analysis (EXODUS Version)*, SAND88-1431, Sandia National Laboratories, Albuquerque, NM.
- Gilkey, A.P. and Glick, J.H., 1989 *BLOT - A Mesh and Curve Plot Program for the Output of a Finite Element Analysis*, SAND88-1432, Sandia National Laboratories, Albuquerque, NM.
- Gilkey, A.P. and Sjaardema, G.D., 1989 *GEN3D: A GENESIS Database 2D to 3D Transformation Program*, SAND89-0485, Sandia National Laboratories, Albuquerque, NM.
- Gingold, R.A. and Monaghan, J.J., 1982 "Kernel Estimates as a Basis for General Particle Methods in Hydrodynamics," *Journal Computational Physics*, Vol. 46, pp. 429-453.
- Glowinski, R. and LeTallec, P., 1989 *Augmented Lagrangian and Operator Splitting Methods in Nonlinear Mechanics*, SIAM, Philadelphia.
- Goodier, J.N. 1965 "On the mechanics of indentation and cratering in the solid targets of strain-hardening metal by impact of hard and soft spheres", *Proceedings of the 7th Symposium on Hypervelocity Impact III*, pp. 215-259.
- Grady, D., 1983 "The Mechanics of Fracture Under High-rate Stress Loading," in *William Prager Symposium on Mechanics of Geomaterials: Rocks, Concretes and Soils*, (Bazant, Z.P., ed.).
- Guenther, C., Hicks, D.L. and Swegle, J.W., 1994 *Conservative Smoothing versus Artificial Viscosity*, SAND94-1853, Sandia National Laboratories, Albuquerque, NM.
- Gurtin, M.E., 1981 *An Introduction to Continuum Mechanics*, Academic Press, Inc.
- Hallquist, J.O., 1981 *User's Manual for DYNA3D and DYNAP*, Lawrence Livermore National Laboratory, Livermore, CA.

- Hallquist, J.O., 1982 *User's Manual for DYNA2D - An Explicit Two-Dimensional Hydrodynamic Finite Element Code with Interactive Rezoning*, Lawrence Livermore National Laboratory, Livermore, CA.
- Hallquist, J.O., 1984 *NIKE3D: An Implicit, Finite Deformation, Finite Element Code for Analyzing the Static and Dynamic Response of Three Dimensional Solids*, UCID-18822, Lawrence Livermore National Laboratory, Livermore, CA.
- Hallquist, J.O., 1984 *User's Manual for DYNA2D: An Explicit Two-Dimensional Hydrodynamic Finite Element Code with Interactive Rezoning*, Rev. 2, UCID-18756, Lawrence Livermore National Laboratory, Livermore, CA.
- Hallquist, J.O., 1986 *NIKE2D: A Vectorized Implicit, Finite Deformation Finite Element Code for Analyzing the Static and Dynamic Response of 2-D Solids with Interactive Rezoning and Graphics*, UCID-19677, Lawrence Livermore National Laboratory, Livermore, CA.
- Hallquist, J.O., Goudreau, G.L. and Benson, D.J., 1985 "Sliding Interfaces with Contact-Impact in Large-Scale Lagrangian Computations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 51, 107-137.
- Hallquist, J.O. and Benson, D.J., 1986 "A comparison of an Implicit and Explicit Implementation of the Hughes-Liu Shell," in *Finite Element Methods for Plate and Shell Structures, Volume 1: Element Technology*, (Hughes, T.J.R. and Hinton, E., eds.), Pineridge Press International, Swansea, U.K., pp. 394-430.
- Hallquist, J.O. and Benson, D.J., 1987 *User's Manual for DYNA3D: Nonlinear Dynamic Analysis of Structures*, Rev. 3, UCID-19592, Lawrence Livermore National Laboratory, Livermore, CA.
- Harding, D.C., Attaway, S.W., Neilsen, J., Blacker, T.D. and Pierce, J., 1992 *Evaluation of Four Multiple Package Crush Environment to the Common Package, Model 1, Plutonium Air Transport Container*, SAND92-0278, Sandia National Laboratories, Albuquerque, NM.
- Harlow, F.H., 1988 "PIC and its Progeny," *Computational Physics Communications*, Vol. 48, pp. 1-10.
- Hearmon, R. F. S., 1961 *An Introduction to Applied Anisotropic Elasticity*, Oxford University Press, Oxford, England.
- Herrman and Peterson, 1968 ???
- Heinstein, M.W., Attaway, S.W., Mellow, F.J. and Swegle, J.W., 1993 *A General-Purpose Contact Detection Algorithm for Nonlinear Structural Analysis Codes*, SAND92-2141, Sandia National Laboratories, Albuquerque, NM.

- Hestenes, M.R. and Stiefel, E., 1952 "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, Vol. 49, pp. 409-436.
- Hibbitt, Karlsson and Sorensen, Inc., 1989 *ABAQUS Users Manual, Version 4-8*, Hibbitt, Karlsson and Sorensen, Inc., Providence, Rhode Island.
- Hibbitt, Karlsson and Sorensen, Inc., 1992 *Contact Calculations with ABAQUS - ABAQUS Explicit Users Manual*, Hibbitt, Karlsson and Sorensen, Inc., Providence, Rhode Island.
- Hicks, D.L., 1978 "Stability Analysis of WONDY (A Hydrocode Based on the Artificial Viscosity Method of von Neumann and Richtmyer) for a Special Case of Maxwell's Law," *Mathematics of Computation*, Vol. 32, pp. 1123-1130.
- Hilber, Hughes and Taylor, 1977 "Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics," *Earthquake Engineering and Structural Dynamics*, Vol. 5, pp.283-292.
- Hill, R., 1948 *A Theory of Earth Movement Near a Deep Underground Explosion*, Memo No. 21-48, Armament Research Establishment, Fort Halstead, Kent, UK.
- Hill, R., 1950 *The Mathematical Theory of Plasticity*, Oxford University Press, London.
- Holcomb, D.J., 1985 "Results of the 55 Day Consolidation Test," Internal Memorandum to J. Stormont, 6332, Sandia National Labs, Albuquerque, NM.
- Holcomb, D.L and Hannum, D.W., 1982 *Consolidation of Crushed Salt Backfill Under Conditions Applicable to the WIPP Facility*, SAND82-0630, Sandia National Labs, Albuquerque, NM.
- Holcomb, D.J. and Shields, M.F., 1987 *Hydrostatic Consolidation of Crushed Salt with Added Water*, SAND87-1990, Sandia National Labs, Albuquerque, NM.
- Holden, J.T., 1972 "On the finite deflections of thin beams," *International Journal of Solids and Structures*, Vol. 8, pp. 1051-1055.
- Hopkins, H.G., 1960 "Dynamic expansion of spherical cavities in metals", *Progress in Solid Mechanics*, Vol. 1, (Editors I. Sneddon and R. Hill), North Holland, New York, pp. 85-164.
- Huang, J., 1969 "Transient Interaction of Plane Acoustic Waves with a Spherical Elastic Shell," *Journal Acoustical Society of America*, Vol. 45, pp. 661-670.
- Hughes, T.J.R., 1987 *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

- Hughes, T.J.R. and Winget, J., 1980 "Finite rotation effects in numerical integration of rate constitutive equations arising in large-deformation analysis," *International Journal for Numerical Methods in Engineering*, Vol. 15, No. 12, pp. 1862-1867.
- Ide, Y. and White, J.L., 1977 "Investigation of Failure During Elongational Flow of Polymer Melts," *Journal of Non-Newtonian Fluid Mechanics*, Vol. 2, pp. 281-298.
- Ide, Y. and White, J.L., 1979 "Experimental Study of Elongational Flow and Failure of Polymer Melts," *Journal of Applied Polymer Science*, Vol. 22, pp. 1061-1079.
- International Nickel Company, Inc., The, 1964 *18% Nickel Maraging Steels*, New York, NY.
- Irons, B. and Elsawaf, A., 1977 "The conjugate Newton algorithm for solving finite element equations," *Formulations and Computation Algorithms in Finite Element Analysis*, (K.J. Bathe and W. Wunderlich, eds.), MIT Press, pp. 655-672.
- Johnson, G.C. and Bammann, D.J., 1984 "A discussion of stress rates in finite deformation bodies," *International Journal of Solids and Structures*, Vol. 20, no. 8, pp. 725-737.
- Johnson, G.C. and Bammann, D.J., 1984 "On the Analysis of Rotation and Stress Rate in Deforming Bodies," *International Journal of Solids and Structures*, Vol. 20, No. 8, pp. 725-737.
- Johnson, G.R., 1983 *Development of Strength and Fracture Models for Computations Involving Severe Dynamic Loading, Vol. I: Strength and Fracture Models*, Tech. Rep. AFATL-TR-83-05, Air Force Armament Laboratory.
- Johnson, G.R., 1988 "Implementation of simplified constitutive models in large computer codes," in *Dynamic Constitutive/Failure Models*, (A. M. Rajendran and T. Nichols, eds.), pp. 409-426, Dayton, OH.
- Johnson, G.R. and Holmquist, T.J., 1989 *Test Data and Computational Strength and Fracture Model Constants for 23 Materials Subjected to Large Strains, High Strain Rates, and High Temperatures*, Tech. Rep. LA-11463-MS, Los Alamos National Laboratory, Los Alamos, NM.
- Johnson, G.R., Stryk, R.A., Holmquist, T.J. and Beissel, S.R., 1996 *User Instructions for the 1996 version of the EPIC Code*, Alliant Techsystems Inc., Hopkins, MN.
- Johnson, G.R., Stryk, R.A., Holmquist, T.J. and Beissel, S.R., 1997 *EPIC 97* (software to be released), Alliant Techsystems Inc., Hopkins, MN.
- Johnson, J.B., 1989 "Personal communication," USACRREL, Building 4070, Ft. Wainwright, Alaska.

- Ju, J.-W. and Taylor, R.L., 1988 “A Perturbed Lagrangian Formulation for the Finite Element Solution of Nonlinear Frictional Contact Problems,” *Journal of Theoretical and Applied Mechanics*, Vol. 7, 1-14.
- Kamei, E. and Onogi, 1975 “Extensional and Fractural Properties of Monodisperse Polystyrenes at Elevated Temperatures,” *Applied Polymer Symposium*, Vol 27, pp. 19-46.
- Kaplan, W., 1959 *Advanced Calculus*, Addison-Wesley Publishing, Reading MA.
- Kelley, C.T., 1995 *Iterative Methods for Linear and Nonlinear Equations*, Society for Industrial and Applied Mathematics, Philadelphia.
- Kerighan, B.W. and Ritchie, D.M., 1978 *The C Programming Language*, Prentice-Hall, Inc., New Jersey.
- Key, S.W., Beisinger, Z.E. and Kreig, R.D., 1978 *HONDO II -A Finite Element Computer Program for the Large Deformation Dynamics Response of Axisymmetric Solids*, SAND78-0422, Sandia National Laboratories, Albuquerque, NM.
- Kikuchi, N. and Song, Y.J., 1981 “Penalty/Finite Element Approximations of a Class of Unilateral Problems in Linear Elasticity,” *Quarterly of Applied Mathematics*, Vol. 39, 1-22.
- Kikuchi, N. and Oden, J.T., 1988 *Contact Problems in Elasticity: A Study of Variational Inequalities and Finite Element Methods*, SIAM, Philadelphia.
- Kipp, M.E. and Grady, D.E., 1978 *Numerical Studies of Rock Fragmentations*, SAND79-1582, Sandia National Laboratories, Albuquerque, NM.
- Kipp, M.E., Grady, D.E. and Chen, E.P., 1980 “Strain-rate Dependent Fracture Initiation,” *International Journal of Fracture*, Vol. 16, pp. 471-478.
- Kipp, M.E. and Lawrence, R.J., 1982 *WONDY V - A One-Dimensional Finite-Difference Wave Propagation Code*, SAND81-0930, Sandia National Laboratories, Albuquerque, NM.
- Krieg, R.D., 1978 *A Simple Constitutive Description for Soils and Crushable Foams*, SC-DR-72-0883, Sandia National Laboratories, Albuquerque, NM.
- Krieg, R.D. and Key, S.W., 1976 “Implementation of a Time Dependent Plasticity Theory into Structural Computer Programs,” *Constitutive Equations in Viscoplasticity: Computational and Engineering Aspects*, Vol. 20, ASME, New York, p. 125.
- Kreig, R.D. and Krieg, D.B., 1977 “Accuracies of numerical solution methods for the elastic-perfectly plastic model,” *ASME Journal of Pressure Vessel Technology*, Vol. 99, pp. 510-515.

- Kuszmaul, J.S., 1987 "A New Constitutive Model for Fragmentation of Rock Under Dynamic Loading," in *Proceedings of the Second International Symposium on Fragmentation by Blasting*, pp. 412-423, Keystone, CO.
- Kuszmaul, J.S., 1987 "A Technique for Predicting Fragmentation and Fragment Sizes Resulting from Rock Blasting," in *Proceedings of the 28th U.S., Symposium on Rock Mechanics*, Tucson, AZ.
- Lai, W.M, Rubin, D. and Krempl, E., 1993 *Introduction to Continuum Mechanics*, 3rd edition, Pergamon Press.
- Laursen, T.A., 1994 "The Convected Description in Large Deformation Frictional Contact Problems," *International Journal of Solids and Structures*, Vol. 31, pp. 669-681.
- Laursen, T.A. and Simo, J.C., 1993 "A Continuum-Based Finite Element Formulation for the Implicit Solution of Multibody, Large Deformation Frictional Contact Problems," *International Journal for Numerical Methods in Engineering*, Vol. 36, 3451-3485.
- Laursen, T.A. and Simo, J.C., 1993 "Algorithmic Symmetrization of Coulomb Frictional Problems Using Augmented Lagrangians," *Computer Methods in Applied Mechanics and Engineering*, Vol. 108, 133-146.
- Leaderman, 1943 ???
- Lee, E., Finger, M. and Collins, W., 1973 *JWL Equation of State Coefficients for High Explosives*, UCID-16189, Lawrence Livermore National Laboratory, Livermore, CA.
- Lee, Radok and Woodward, 1959 ???
- Lenard, M.L., 1976 "Convergence conditions for restarted conjugate gradient methods with inaccurate line searches," *Mathematical Programming*, Vol. 10, pp. 32-51.
- Libersky, L. and Petschek, A.G., 1990 "Smooth Particle Hydrodynamics with Strength of Materials," *Proceedings of The NEXT Free-Lagrange Conference*, Jackson Lake Lodge, Moran, WY, June 3-7.
- Longcope, D.B., 1991 *Coupled Bending/Lateral Load Modeling of Earth Penetrators*, SAND90-0789, Sandia National Laboratories, Albuquerque, NM.
- Longcope, D.B., 1996 *Oblique Penetration Modeling and Correlation with Field Tests into a Soil Target*, SAND96-2239, Sandia National Laboratories, Albuquerque, NM.
- Longcope, D.B. and Tabbara, M.R., 1998 **Modeling of Oblique Earth Penetration Using Cavity Expansion Loading with Surface Effects**, SAND98-xxxx, Sandia National Laboratories, Albuquerque, NM, in preparation.

- Lovejoy, S.C. and Whirley, R.G., 1990 *DYNA3D Example Problem Manual*, UCRL-MA-105259, Lawrence Livermore National Laboratory, Livermore, CA.
- Luenberger, D.G., 1984 *Linear and Nonlinear Programming*, 2nd ed., Addison-Wesley, Reading, MA.
- Lucy, L.B., 1977 "A Numerical Approach to the Testing of the Fission Hypothesis," *Astrophysics Journal*, Vol. 82, pp. 1013-1024.
- Luk, V.K. and Piekutowski, A.J., 1991 "An analytical model on penetration of eroding long rods into metallic targets," *International Journal of Impact Engineering*, Vol. 11, pp. 323-340.
- Lysmer, J. and Kuhlemeyer, R.L., 1979 "Finite Dynamic Model for Infinite Media," *Journal of the Engineering Mechanics Division of ASCE*, pp. 859-877.
- Malkin, A. Y. and Petrie, C.J.S., 1997 "Some Conditions for Rupture of Polymer Liquids in Extension," *Journal of Rheology*, Vol. 41 (1), pp. 1-25.
- Malvern, L.E., 1969 *Introduction to the Mechanics of a Continuous Medium*, Prentice-Hall, Inc., New Jersey, pp. 226-228.
- Marsden, J.E. and Hughes, T.J.R., 1983 *Mathematical Foundations of Elasticity*, Prentice-Hall, Englewood Cliffs, New Jersey.
- Matthies, H. and Strang, G., 1979 "The Solution of Nonlinear Finite Element Equations," *International Journal for Numerical Methods in Engineering*, Vol. 14, 1613-1626.
- McClelland, 19?? ???
- McClure, C., 1993 *Preliminary Report on Explosive Field Tests in Support of the Hull Deformation/Rupture Study*, NSWC Report NSWCDD/TN-93/94.
- Mendelson, A., 1968 *Plasticity: Theory and Application*, The Macmillan Company, New York, pp. 138-156.
- Metzinger, K., Attaway, S. and Mello, F., 1991 "Bobbin Stresses Generated by Wire Winding," *First International Conference of Web Handling*, Oklahoma State University, Stillwater, OK.
- Michalowski, R. and Mroz, Z., 1978 "Associated and Non-Associated Sliding Rules in Contact Friction Problems," *Archives of Mechanics*, Vol. 30, 259-276.
- Mills-Curran, W.C., 1988 *EXODUS: A Finite Element File Format for Pre- and Post-Processing*, SAND87-2997, Sandia National Laboratories, Albuquerque, NM.
- Mindlin, R.D., 1951 "Influence of Rotatory Inertia and Shear on Flexural Motions of Isotropic, Elastic Plates," *Journal of Applied Mechanics*, Vol. 18, pp. 31-38.

- Monaghan, J.J., 1982 "Why Particle Methods Work," *SIAM Journal Scientific Statistical Computing*, Vol. 3, pp. 422-433.
- Monaghan, J.J., 1985 "Particle Methods for Hydrodynamics," *Computational Physics Reports*, Vol. 3, pp. 71-124.
- Monaghan, J.J., 1988 "An Introduction to SPH," *Computational Physics Communications*, Vol. 48, pp. 89-96.
- Monaghan, J.J. and Gingold, R.A., 1983 "Shock Simulation by the Particle Method SPH," *Journal of Computational Physics*, Vol. 52, pp. 374-389.
- Morland and Lee, 1960 ???
- Mungiza, A., Owen, D.R.J. and Bicanic, N., 1995 "A Combined Finite-Discrete Element Method in Transient Dynamics of Fracturing Solids," *Engineering Computations*, Vol. 12, pp. 145-174.
- Muki and Sternberg, 1961 ???
- Neilsen, M.K., Morgan, H.S. and Krieg, R.D., 1986 *A Phenomenological Constitutive Model for Low Density Polyurethane Foams*, SAND86-2927, Sandia National Laboratories, Albuquerque, NM.
- Nemat-Nasser, S., Chung, D. and Taylor, L.M., 1989 "Phenomenological modeling of rate-dependent plasticity for high strain rate problems," *Mechanics of Materials*, Vol. 7, No. 4, pp. 319-344.
- Newmark, N.M., 1959 "A Method of Computation for Structural Dynamics," *Journal of the Engineering Mechanics Division*, ASCE, 67-94.
- Ogden, R.W., 1987 "Recent advances in the phenomenological theory of rubber elasticity," *Rubber Chemistry and Technology*, Vol. 59, pp. 361-383.
- Parisch, H., 1989 "A Consistent Tangent Stiffness Matrix for Three Dimensional Non-Linear Contact Analysis," *International Journal for Numerical Methods in Engineering*, Vol. 28, 1803-1812.
- Patel, N.R. and Finnie, I., 1969 ???, Report UCRL-13420, Lawrence Livermore Laboratory, Livermore, CA.
- Pearson, G.H. and Connelly, R.W., 1982 "The Use of Extensional Rheometry to Establish Operating Parameters for Stretching Processes," *Journal of Applied Polymer Science*, Vol. 27, pp. 969-981.

- Perzyna, P., 1966 "Fundamental Problems in Viscoplasticity," in *Recent Advances in Applied Mechanics*, Academic Press, New York, pp. 243-377.
- Pfeiffle, T.W. and Senseny, P.E., 1985 *Permeability and Consolidation of Crushed Salt from the WIPP Site*, Topical Report RSI-0278, RE/SPEC Inc., Rapid City, SD.
- Pilkey, W.D. and Wunderlich, W., 1994 *Mechanics of Structures: Variational and Computational Methods*, CRC Press.
- Plimpton, S.J., 1990 "Molecular Dynamics Simulations of Short-Range Force Systems on 1024-Node Hypercubes," *Proceedings of the Fifth Distributed Memory Computing Conference*, Charleston, SC.
- Powell, M.J.D., 1977 "Restart procedures for the conjugate gradient method," *Mathematical Programming*, Vol. 12, pp. 242-254.
- Ralston, A. and Wilf, H.S., 1960 *Mathematical Methods of Digital Computers*, John Wiley and Sons Inc., New York, pp. 62-72.
- Ratigan and Wagner, 19?? ???
- Rebello, N., Nagtegaal, J.C. and Hibbitt, H.D., 1990 "Finite Element Analysis of Sheet Forming Processes," *International Journal for Numerical Methods in Engineering*, Vol. 30, 1739-1758.
- Reedy, J.E.D., 1990 "Memo: Code corrections to pronto's soils and crushable foams model," Applied Mechanics Division, Sandia National Laboratories, Albuquerque, NM.
- Richtmyer, R.D. and Morton, K.W., 1967 *Difference Methods for Initial Value Problems*, Interscience, New York.
- Rivlin, R.S., 1948 "???", *Philosophical Transactions of the Royal Society of London*, A, pp. 459-490.
- Schoof, L.A. and Yarberry, V.R., 1994 *EXODUS II: A Finite Element Data Model*, SAND92-2137, Sandia National Laboratories, Albuquerque, NM.
- Schreyer, H.L., Kulak, R.F. and Kramer, J.M., 1979 "Accurate numerical solutions for elastic-plastic models," *ASME Journal of Pressure Vessel Technology*, Vol. 101, pp. 226-234.
- Schwarzl and Staverman, 1952 ???
- Silling, S. A., 1991 *CTH Reference Manual: Viscoplastic Models*, SAND91-0292, Sandia National Laboratories, Albuquerque, NM.

- Simo, J.C., 1992 “Algorithms for Static and Dynamic Multiplicative Plasticity that Preserve the Classical Return Mapping Schemes of the Infinitesimal Theory,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 99, 61-112.
- Simo, J.C., Marsden, J.E. and Krishnaprasad, P.S., 1988 “The Hamiltonian Structure of Nonlinear Elasticity: The Material and Convective Representations of Solids, Rods and Plates,” *Archive for Rational Mechanics*, Vol. 104, 125-183.
- Simo, J.C. and Miehe, C., 1992 “Associative Coupled Thermoplasticity at Finite Strains: Formulation, Numerical Analysis and Implementation,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 98, 41-104.
- Simo, J.C. and Taylor, R.L., 1985 “Consistent Tangent Operators for Rate-Independent Elastoplasticity,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 48, 101-118.
- Simo, J.C., Taylor, R.L. and Wriggers, P., 1991 “A Note on Finite Element Implementation of Pressure Boundary Loading,” *Communications in Applied Numerical Methods*, Vol. 50, 163-180.
- Sjaardema, G.D., 1989 *NUMBERS: A Collection of Utilities for Pre- and Post-Processing Two- and Three-Dimensional EXODUS Finite Element Models*, SAND88-0737, Sandia National Laboratories, Albuquerque, NM.
- Sjaardema, G.D., 1992 *GJOIN: A Program for Merging Two or More GENESIS Databases Version 1.4*, SAND92-2290, Sandia National Laboratories, Albuquerque, NM.
- Sjaardema, G.D., 1993 *GENSHELL: A GENESIS Database Shell Transformation Program*, Sandia National Laboratories, Albuquerque, NM.
- Sjaardema, G.D., 1993 *Overview of the Sandia National Laboratories Engineering Analysis Code Access System*, SAND92-2292, Sandia National Laboratories, Albuquerque, NM.
- Sjaardema, G.D. and Krieg, R.D., 1987 *A Constitutive Model for the Consolidation of WIPP Crushed Salt and Its Use in Analyses of Backfilled Shaft and Drift Configurations*, SAND87-1977*UC-70, Sandia National Laboratories, Albuquerque, NM.
- Stone, C.M., 1989 “Personal communication,” Applied Mechanics Division, Sandia National Laboratories, Albuquerque, NM.
- Stone, C.M., 1995 *SANTOS: A Two-Dimensional Finite Element Program for the Quasistatic Large Deformation, Inelastic Response of Solids*, SAND90-0543, Sandia National Laboratories, Albuquerque, NM.

- Stone, C.M., Krieg, R.D. and Beisinger, Z.E., 1988 *SANCHO - A Finite Element Computer Program for the Quasistatic, Large Deformation, Inelastic Response of Two-Dimensional Solids*, SAND84-2618, Sandia National Laboratories, Albuquerque, NM.
- Stone, C. M. and Wellman, G. W., 1993 “Implementation of Ductile Failure in PRONTO2D and PRONTO3D”, Memo to Distribution, Sandia National Laboratories, Albuquerque, NM.
- Stone, C.M., Wellman, G.W. and Krieg, R.D., 1990 *A Vectorized Elastic/Plastic Power Law Hardening Material Model Including Luders Strain*, SAND90-0153, Sandia National Laboratories, Albuquerque, NM.
- Strang, G. and Fix, G.J., 1973 *An Analysis of the Finite Element Method*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Swegle, J.W., 1978 *TOODY IV - A Computer Program for Two-Dimensional Wave Propagation*, SAND78-0552, Sandia National Laboratories, Albuquerque, NM.
- Swegle, J.W., 1992 “Search Algorithm,” Memo to Distribution, Sandia National Laboratories, Albuquerque, NM.
- Swegle, J.W., Attaway, S.W., Heinstejn, M.W., Mello, F.J. and Hicks, D.L., 1993 *An Analysis of Smoothed Particle Hydrodynamics*, SAND93-2513, Sandia National Laboratories, Albuquerque, NM.
- Swenson, D.V. and Taylor, L.M., 1983 “A Finite Element Model for the Analysis of Tailored Pulse Stimulation of Boreholes,” *International Journal for Numerical and Analytical Methods in Geomechanics*, Vol. 7, pp. 469-484.
- Takaki, T. and Bogue, D.C., 1975 “The Extensional and Failure Properties of Polymer Melts,” Vol. 19, pp. 419-433.
- Taylor, L.M. and Becker, E.B., 1983 “Some Computational Aspects of Large Deformation, Rate-Dependent Plasticity Problems,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 41, No. 3, pp. 251-277.
- Taylor, L.M., Chen, E.P. and Kuszmaul, J.S., 1986 “Microcrack-Induced Damage Accumulation in Brittle Rock Under Dynamic Loading,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 55, No. 3, pp. 301-320.
- Taylor, L.M., Flanagan, D.P. and Mills-Curran, W.C., 1986 *The GENESIS Finite Element Mesh File Format*, SAND86-0910, Sandia National Laboratories, Albuquerque, NM.
- Taylor, L.M. and Flanagan, D.P., 1987 *PRONTO 2D: A Two Dimensional Transient Solid Dynamics Program*, SAND86-0594, Sandia National Laboratories, Albuquerque, NM.

- Taylor, L.M. and Flanagan, D.P., 1989 ***PRONTO 3D: A Three Dimensional Transient Solid Dynamics Program***, SAND 87-1912, Sandia National Laboratories, Albuquerque, NM.
- Thompson, S.L., 1977 ***CSQII - An Eulerian Finite Difference Program for Two-Dimensional Material Response - Part I, Material Selections***, SAND77-1339, Sandia National Laboratories, Albuquerque, NM.
- Thorne, B.J., 1990 ***A Damage Model for Rock Fragmentation and Comparison of Calculations with Blasting Experiments in Granite***, SAND90-1389, Sandia National Laboratories, Albuquerque, NM.
- Thorne, B.J. and Preece, D.S., 1989 “Personal communications,” Geoenergy Technology Department, Sandia National Laboratories, Albuquerque, NM.
- Thrun, R., Goertner, J.F. and Harris, G.S., 1993 ***Underwater Explosion Bubble Collapse Against a Flat Plate***, Seneca Lake Test Series Data Report, NSWC Report, NSWCDD/TR-92/482.
- Timoshenko, S. and MacCollough, 1958 ???
- Timoshenko, S. and Goodier, J.N., 1970 ***Theory of Elasticity***, 3rd ed., McGraw-Hill, New York.
- Treloar, 1994 ???
- Truesdell, C., 1966 ***The Elements of Continuum Mechanics***, Springer Verlag, New York.
- Truesdell, C., 1977 ***A First Course in Rational Continuum Mechanics, Vol. 1, General Concepts***, Academic Press, Inc., New York, p. 162.
- Truesdell, C. and Noll, W., 1965 “Non-Linear Field Theories”, in the ***Handbook of Physics*** by Flugge, Springer-Verlag, Berlin.
- Valanis, K. and Lnadell, R.F., 1967 “The strain energy function of a hyperelastic material in terms of the extension ratios,” ***Journal of Applied Physics***, Vol. 38.
- Von Neumann, J. and Richtyter, R.D., 1950 “A Method for the Numerical Calculation of Hydrodynamic Shocks,” ***Journal of Applied Physics***, Vol. 21, p. 232.
- Warren, T.L. and Forrestal, M.J., 1997 “Effects of strain hardening and strain-rate sensitivity on the penetration of aluminum targets with spherical-nosed rods”, ***International Journal of Solids Structures*** (submitted).
- Warren, T.L. and Tabbara, M.R., 1997 ***Spherical Cavity-Expansion Forcing Function in PRONTO3D for Application to Penetration Problems***, SAND97-1174, Sandia National Laboratories, Albuquerque, NM.

- Weaver, Jr., W. and Johnson, P. R., 1984 *Finite Elements for Structural Analysis*, Prentice-Hall, Englewood Cliffs, New Jersey.
- Wellman, G. W., 1993 “Investigation of Mesh Dependencies in Ductile Failure for Transient Dynamics”, Memo to Distribution, Sandia National Laboratories, Albuquerque, NM.
- Wen, Y., Hicks, D. L. and Swegle, J. W., 1994 *Stabilizing S.P.H. with Conservative Smoothing*, SAND94-1932, Sandia National Laboratories, Albuquerque, NM.
- Whirley, R. G., Engelmann, B. E. and Hallquist, J. O., 1991 *DYNA3D Users Manual*, Lawrence Livermore Laboratory, Livermore, CA.
- Whirley, R.G., Hallquist, J.O. and Goudreau, G.L., 1988 *An Assessment of Numerical Algorithms for Plane Stress and Shell Elastoplasticity on Supercomputers*, UCRL-99690, Lawrence Livermore National Laboratory, Livermore, CA.
- Wilkins, M.L. and Guinan, M.W., 1973 “Impact of CYlinders on a Rigid Boundary”, *Journal of Applied Physics*, Vol. 44.
- Williams, Landell and Ferry, 1955 ???
- Wriggers, P. and Simo, J.C., 1985 “A Note on Tangent Stiffness for Fully Nonlinear Contact Problems,” *Communications in Applied Numerical Methods*, Vol. 1, 199-203.
- Wriggers, P., Vu Van, T. and Stein, E., 1990 “Finite Element Formulation of Large Deformation Impact-Contact Problems with Friction,” *Computers and Structures*, Vol. 37, 319-331.
- Zeuch, Holcomb and Lauson, 19?? ???
- Zhang, P. and Geers, T. L., 1993 “Excitation of a fluid-filled, submerged spherical shell by a transient acoustic wave,” *Journal Acoustical Society of America*, Vol. 93, pp. 696-705.
- Zhong, Z.H. and Nilsson, L., 1990 “A Contact Searching Algorithm for General 3D Contact-Impact Problems,” *Computers and Structures*, Vol. 34, No. 2, pp. 327-335.

Distribution:

MS9018 Central Technical Files, 8940-2

MS0899 Technical Library, 4916 (2)

MS0619 Review & Approval Desk, 15102 for DOE/OSTI (2)

MS0443 S.W. Attaway, 9117 (20)