Sandia
National
Laboratories

Sierra/SolidMechanics 5.26 Theory Manual

Sierra Solid Mechanics Team
Computational Solid Mechanics and Structural Dynamics Department
Engineering Sciences Center

Abstract

Presented in this document are the theoretical aspects of capabilities contained in the Sierra/SM code. This manuscript serves as an ideal starting point for understanding the theoretical foundations of the code. For a comprehensive study of these capabilities, the reader is encouraged to explore the many references to scientific articles and textbooks contained in this manual. It is important to point out that some capabilities are still in development and may not be presented in this document. Further updates to this manuscript will be made as these capabilities come closer to production level.

Acknowledgements

Front Matter

List of Tables

List of Figures

This page left blank

1 Nonlinear Behavior

1.1 Introduction

We begin our study of nonlinear computational solid mechanics in this chapter by surveying some frequently encountered sources of nonlinearity in engineering mechanics. This will be done in a rather elementary way, by discussing perhaps the simplest structural idealization, the truss member, which is assumed to transmit loads in the axial direction only. By introducing various nonlinearities into this system one at a time, we will motivate the more general discussion of nonlinear continuum mechanics, constitutive modeling, and numerical treatments to follow. This model system will serve as a template throughout the text as new continuum mechanical and computational ideas are introduced.

Following this motivation will be an introduction to the prescription of initial/boundary value problems in solid mechanics. This introduction will be provided by discussing a completely linear system, namely linear elastic behavior in a continuum subject to infinitesimal displacements. This treatment will include presentation of the relevant field equations, boundary conditions, and initial conditions, encompassing both dynamic and quasistatic problems in the discussion. Also featured is a brief discussion of the *weak* or *integral* form of the governing equations, providing a starting point for application of the finite element method. Examination of these aspects of problem formulation in the comparatively simple setting of linear elasticity allows one to concentrate on the ideas and concepts involved in problem description without the need for an overly burdensome notational structure.

In anticipation of nonlinear solid mechanics applications, however, we will find it necessary to expand this notational framework so that large deformation of solids can be accommodated. Fortunately, provided certain interpretations are kept in mind, the form of the governing equations is largely unchanged by the generalization of the linear elastic system. This chapter therefore provides an introduction to how this generalization can be made. However, it will be seen that the continuum description and constitutive modeling of solids undergoing large deformations are complex topics that should be understood in detail before formulating and implementing numerical strategies. The closely related topics of nonlinear continuum mechanics and constitutive modeling will therefore be the subjects of subsequent chapters, followed by significant discussions of numerical strategies.

We conclude with a short list of references the reader may find useful as background material. Throughout the text, we assume little or no familiarity with either the finite element method or nonlinear solid mechanics, but we do assume a basic level of familiarity with the mechanics of materials, linear continuum mechanics, and linear elasticity. The last section of this chapter provides some basic references in these areas for those wishing to fill gaps in knowledge.

1.2 Linear Structural Component

We consider the simple axial (or in structural terms, truss) member shown schematically in Fig. 1.1. We can think of this member as a straight bar of material, whose transverse dimensions are small compared to its overall length, and which can only transmit loads in the axial direction. Real-world examples include taut cables in tension, truss members, and similar rod-like objects.
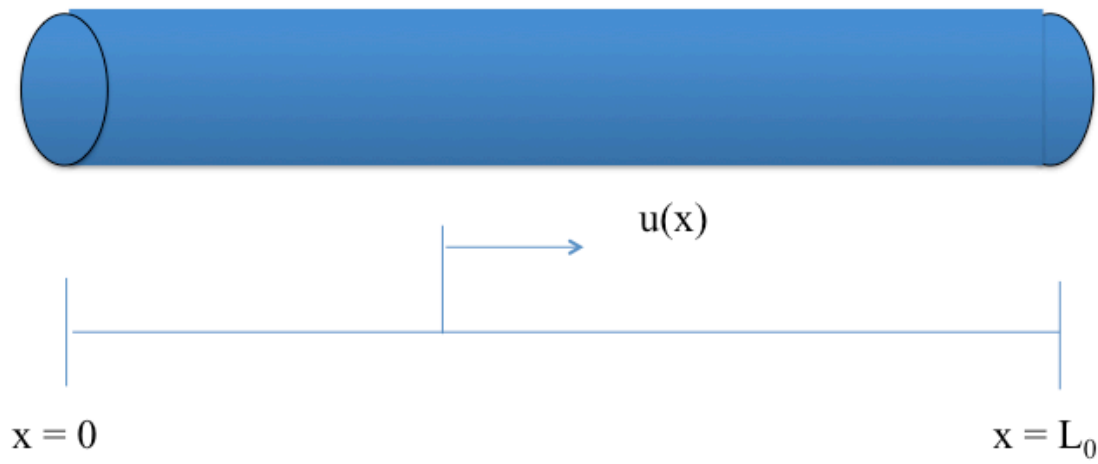
Fig. 1.1 Axial model problem: schematic and local coordinate system.

We index the material with coordinates $\mathbf{x}$ with values between 0 and L_0. Assuming that all displacement of the rod occurs in the axial direction, we write this displacement as $u(\mathbf{x}, t)$, with t signifying time. The **infinitesimal strain** or **engineering strain** at any point $\mathbf{x} \in (0, L_0)$ is given by

$$\epsilon_E(\mathbf{x}, t) = \frac{\partial}{\partial \mathbf{x}} u(\mathbf{x}, t). \tag{1.1}$$

The **true stress** σ_T at any point in the bar and at any instant is described via

$$\sigma_T(\mathbf{x}, t) = \frac{P(\mathbf{x}, t)}{A(\mathbf{x}, t)}, \tag{1.2}$$

where P is the total axial force acting at location $\mathbf{x}$ and A is the current cross-sectional area at that location. If the cross-sectional area does not change very much as a result of the deformation, it is appropriate to define the **nominal stress** or **engineering stress** as

$$\sigma_E = \frac{P(\mathbf{x}, t)}{A_0(\mathbf{x})}, \tag{1.3}$$

where $A_0(\mathbf{x})$ is the initial cross-sectional area at point $\mathbf{x}$. If the material behaves in a **linear elastic** manner then σ_E and ϵ_E are related via

$$\sigma_E = E\epsilon_E, \tag{1.4}$$

where E is the **elastic modulus**, or **Young's modulus**, for the material.

To begin we consider the case of **static equilibrium** where inertial effects are either negligible or nonexistent and the response is therefore independent of time. One can in this case suppress the time argument in (1.2) and (1.4). The balance of linear momentum for the static case is expressed at each point $\mathbf{x}$ by

$$\frac{d}{dx}\left(A_0(\mathbf{x})\sigma_E(\mathbf{x})\right) = f(\mathbf{x}), \tag{1.5}$$

where f is the **applied external body loading**, assumed to be axial, with units of force per unit length. Substitution of (1.4) into (1.5) gives the following ordinary differential equation for $u(\mathbf{x})$ on the domain $(0, L_0)$:

$$\frac{d}{dx}\left(EA_0\frac{d}{dx}(u)\right) = f(\mathbf{x}). \tag{1.6}$$

If we assume that the cross-section is uniform so that A_0 does not vary with $\mathbf{x}$, and that the material is **homogeneous** so that E does not vary throughout the rod, then

$$EA_0\frac{d^2}{dx^2}u(\mathbf{x}) = f(\mathbf{x}), \tag{1.7}$$

We note that (1.7) is a linear, second order differential equation for the unknown displacement field u. In order to pose a mathematical problem that can be uniquely solved it is necessary to pose two **boundary conditions** on the unknown u. We will be interested primarily in two types, corresponding to **prescribed displacement** and **prescribed force** (or stress) boundary conditions. An example of a displacement boundary condition would be

$$u(0) = \bar{u}, \tag{1.8}$$

while an example of a force boundary condition is

$$\sigma_E(L_0) = E\frac{du}{dx}(L_0) = \bar{\sigma}, \tag{1.9}$$

where $\bar{u}$ and $\bar{\sigma}$ are prescribed values for the displacement and axial stress at the left and right bar ends, respectively. In mathematics parlance, the boundary condition in (1.8) is called a **Dirichlet**

boundary condition while the boundary condition represented by (1.9) is a **Neumann** boundary condition. Dirichlet boundary conditions involve the unknown independent variable itself, while Neumann boundary conditions are expressed in terms of its derivatives.

Virtually any combination of such boundary conditions can be applied to our problem, but only one boundary condition (either a Neumann or Dirichlet condition) can be applied at each endpoint. In the case where Neumann (stress) conditions are applied at both ends of the bar, the solution $u(\mathbf{x})$ is only determinable up to an arbitrary constant (this fact can be verified by applying separation of variables to (1.7)).

We now consider a particular case of this linear problem that will be useful in considering some of the various nonlinearities to be discussed below. In particular, suppose $f = 0$ on the domain $(0, L_0)$, and furthermore consider the boundary conditions

$$u = 0 \text{ at } x = 0, \tag{1.10}$$

and

$$\sigma_E = \frac{F^{ext}}{A_0} \text{ at } x = L, \tag{1.11}$$

where F^{ext} is an applied force on the right end of the rod.

In this case, examination of (1.5) yields

$$A_0 \frac{\mathrm{d}}{\mathrm{d}x} \left(\sigma_E(\mathbf{x}) \right) = 0, \tag{1.12}$$

meaning that σ_E does not vary along the length of the rod. Since σ_E is proportional to ϵ_E (see (1.4)), the strain must also be a constant value along the rod length.

Finally, in view of (1.1) we conclude that $u(\mathbf{x})$ must vary linearly with $\mathbf{x}$. In other words, we know that the solution $u(\mathbf{x})$ must take the form

$$u(\mathbf{x}) = u(0) + \delta \left(\frac{x}{L} \right) = \delta \left(\frac{x}{L} \right), \tag{1.13}$$

where δ is the elongation, or difference between the left and right end displacement. The problem therefore reduces to finding the elongation produced by the applied force F^{ext}. This problem is trivially solved and leads to the familiar linear relationship between F^{ext} and δ:

$$\frac{EA_0}{L_0} \delta = F^{ext}; \tag{1.14}$$

in other words, we have a simple linear spring with stiffness EA_0/L_0. After solving for δ one may merely substitute (1.13) to obtain the desired expression for $u(\mathbf{x})$.

1.3 Material Nonlinearity

We examine the case of a **material nonlinearity** by replacing (1.4) with generic relationship between σ_E and ϵ_E,

$$\sigma_E = \hat{\sigma}(\epsilon_E), \qquad (1.15)$$

where $\hat{\sigma}$ is a smooth and generally nonlinear function, see Fig. 1.2.
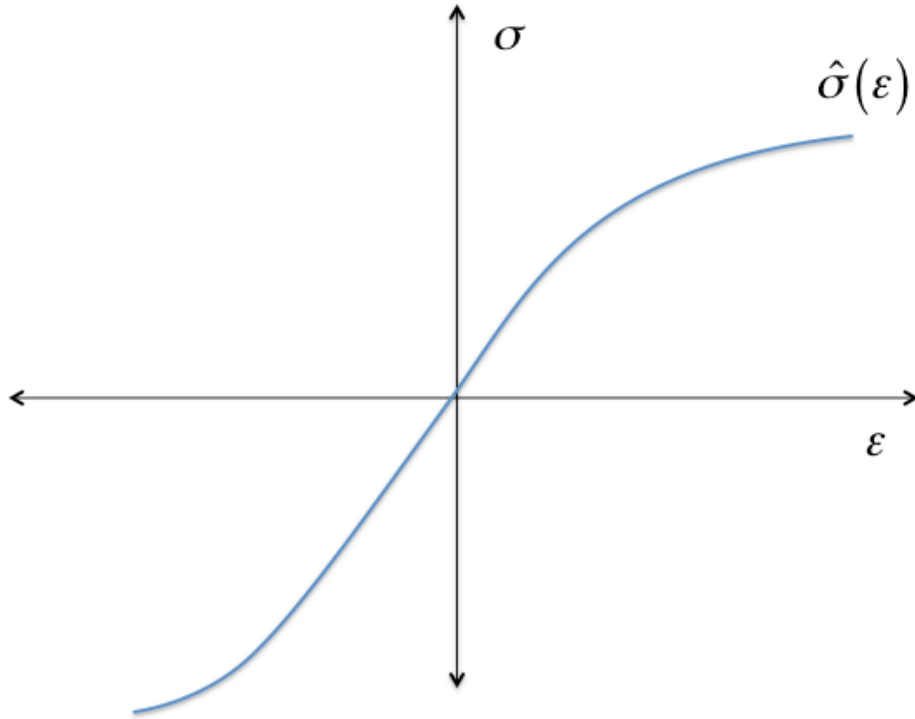
Fig. 1.2 Schematic of a nonlinear one-dimensional stress-strain relation.

We make few restrictions on the specific form of $\hat{\sigma}$, other than to assume that $\frac{\mathrm{d}}{\mathrm{d}\epsilon_E}\hat{\sigma} > 0$ for all values of ϵ_E. If we retain the assumption that $f = 0$ and impose boundary conditions (1.10) and (1.11) then (1.12) is still valid, i.e.,

$$\sigma_E = \frac{F^{ext}}{A_0} \qquad (1.16)$$

throughout the rod. Furthermore, since we assume that a one-to-one relation exists between σ_E and ϵ_E, we conclude that, just as in the linear material case, the strain is a constant value in the rod

given by

$$\epsilon_E = \frac{\delta}{L_0}.$$ (1.17)

We can solve the problem by finding δ as before, but now we must solve the nonlinear equation

$$A_0 \hat{\sigma} \left(\frac{\delta}{L_0} \right) = F^{ext}.$$ (1.18)

We can express (1.18) as an equation for the displacement at the right end which we denote as $d_L = u(L)$. We can write

$$N(d_0) = F^{ext},$$ (1.19)

where $N(d_0)$ is a nonlinear function of the unknown d_L defined in this case as

$$N(d_0) = A_0 \hat{\sigma} \left(\frac{d_L}{L_0} \right).$$ (1.20)

In general, (1.20) will not have a closed-form solution and some sort of iterative procedure is necessary. Nonlinear equation solving is discussed at length in Chapter Section 13. Here we resort to one of the more recognized and widely-used procedures, **Newton-Raphson iteration**. In this method one introduces a set of indices k corresponding to the iterations, and given a current iterate d_L^k, a first-order Taylor series expansion of (1.20) is utilized to generate the next iterate d_L^{k+1} via

$$0 = F^{ext} - N\left(d_L^{k+1}\right) \approx F^{ext} - \left(N\left(d_L^k\right) + \frac{\mathrm{d}}{\mathrm{d}d_L} N\left(d_L^k\right) \Delta d_L \right),$$ (1.21)

where

$$d_L^{k+1} = d_L^k + \Delta d_L.$$ (1.22)

(1.21) can be expressed more compactly as

$$K\left(d_L^k\right) \Delta d_L = R\left(d_L^k\right),$$ (1.23)

where $R\left(d_L^k\right)$, the **residual** or **out-of-balance force**, is given by

$$R\left(d_L^k\right) := F^{ext} - N\left(d_L^k\right),$$ (1.24)

and $K\left(d_L^k\right)$, the **incremental** or **tangent stiffness**, is written as

$$K\left(d_L^k\right) := \frac{\mathrm{d}}{\mathrm{d}d_L} N\left(d_L^k\right).$$ (1.25)

The Newton-Raphson procedure is then carried out by recursively solving (1.23) and (1.22).

1.4 Geometric Nonlinearity

Geometric nonlinearities are induced by nonlinearities in the kinematic description of the system at hand. We will identify and work with several nonlinearities of this general type in great detail in Section 4, Section 5, and Section 6, but to begin we consider two particular cases in the context of our simple model problem.

The first type of nonlinearity we consider is introduced by the use of nonlinear strain and stress measures in definition of the stress-strain relation. As an example, let us consider alternatives to (1.1) and (1.3), which defined the engineering stain ϵ_E and engineering stress σ_E that we have utilized to this point. When used in our model problem with $f = 0$ and boundary conditions (1.10) and (1.11), we have seen that the engineering strain does not vary over the rod's length, having a constant value δ/L_0. For this strain measure to be appropriate, the deformation δ should be infinitesimal. In the presence of larger deformations, the **true strain** or **logarithmic strain** is often used,

$$\epsilon_T = \int_{L_0}^{L} \frac{\mathrm{d}\gamma}{\gamma} = \log\left(\frac{L}{L_0}\right) = \log\left(1 + \epsilon_E\right). \tag{1.26}$$

Similarly, if the cross-sectional area A changes appreciably during the process, it is likely that the engineering stress σ_E should be replaced by the true stress σ_T defined in (1.2). In the case of our model problem, this would imply

$$\sigma_T = \frac{F^{ext}}{A}, \tag{1.27}$$

where A is to be interpreted as the cross-sectional area in the final (deformed) configuration.

Relating this area to the elongation δ requires a constitutive assumption to be made. For example, if we assume the rod consists of an elastic material, we could approximate this variation by considering the area to vary according to Poisson's effect. This would require that for each differential increment $\mathrm{d}\epsilon_T$ in the axial true strain, each lateral dimension should change by a factor of $(1 - \nu\mathrm{d}\epsilon_T)$, where ν is **Poisson's ratio** for the material.

At a given instant of the loading process, therefore, an incremental change in the area A can be approximated via

$$\begin{aligned}
A + \mathrm{d}A &= (1 - \nu\mathrm{d}\epsilon_T)^2 A \\
&\approx (1 - 2\nu\mathrm{d}\epsilon_T) A.
\end{aligned}$$

(1.28) implies that

$$\begin{aligned}
\frac{1}{A}\mathrm{d}A &= -2\nu\mathrm{d}\epsilon_T \\
&= -2\nu\frac{\mathrm{d}\epsilon_T}{\mathrm{d}L}\mathrm{d}L \\
&= -2\nu\left(\frac{1}{L}\right)\mathrm{d}L.
\end{aligned}$$

Integrating (1.28) between the initial and the final configurations gives

$$A = A_0 \left(\frac{L_0}{L} \right)^{2\nu} = A_0 \left(\frac{L_0}{L_0 + \delta} \right)^{2\nu} . \tag{1.28}$$

If we assume Hooke's Law,

$$\sigma_T = E \epsilon_T, \tag{1.29}$$

we can use (1.26), (1.27), and (1.28) to conclude that

$$E A_0 \log \left(\frac{L_0 + \delta}{L_0} \right) \left(\frac{L_0}{L_0 + \delta} \right)^{2\nu} = F^{ext}, \tag{1.30}$$

which is a nonlinear equation governing the elongation δ. Note that this nonlinearity is not caused by any sort of nonlinear stress-strain relation, but instead results from the observation that the amount of deformation may not be small, necessitating more general representations of stress and strain.

The second sort of nonlinearity we wish to consider is that caused by large superimposed rigid body rotations and translations that introduce nonlinearities into many problems even when the strains in the material are well-approximated by infinitesimal measures. Toward this end we refer to Fig. 1.3, in which we embed our one-dimensional truss element in a two-dimensional frame. We locate one end of the rod at the origin and consider this end to be pinned so that it is free to rotate but not translate. The other end of the rod, initially located at coordinates x_1^0, x_2^0, is subjected to a (vector valued) force $\mathbf{F}^{ext}$, which need not be directed along the axis of the rod.

We note that under the restriction of small motions this problem is ill-posed because the rod is incapable of transmitting anything but axial force ($\mathbf{F}^{ext}$ would need to act in the axial direction). However, in the current context we allow unlimited rotation with the result that the rod will rotate until it aligns with $\mathbf{F}^{ext}$ in its equilibrium condition. In fact this observation allows us to guess the solution to the problem. Since we assume that the axial response of the rod is completely linear, we may deduce that the final elongation is given by

$$\delta = \frac{L_0 \|\mathbf{F}^{ext}\|}{E A_0}, \tag{1.31}$$

where $\|\mathbf{F}^{ext}\|$ denotes the Euclidean length of the vector $\mathbf{F}^{ext}$. The final orientation of the rod must coincide with the direction $\mathbf{F}^{ext}$, so we can write the final position of the end of the rod, using the coordinates $\left(x_1^f, x_2^f \right)$, as

$$\begin{bmatrix} x_1^f \\ x_2^f \end{bmatrix} = \frac{L_0}{\|\mathbf{F}^{ext}\|} \left(1 + \frac{\|\mathbf{F}^{ext}\|}{E A_0} \right) \begin{bmatrix} F_1^{ext} \\ F_2^{ext} \end{bmatrix}, \tag{1.32}$$

or, writing the solution in terms of the rod end displacements d_1 and d_2,

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \frac{L_0}{\|\mathbf{F}^{ext}\|} \left(1 + \frac{\|\mathbf{F}^{ext}\|}{E A_0} \right) \begin{bmatrix} F_1^{ext} \\ F_2^{ext} \end{bmatrix} - \begin{bmatrix} x_1^0 \\ x_2^0 \end{bmatrix} . \tag{1.33}$$
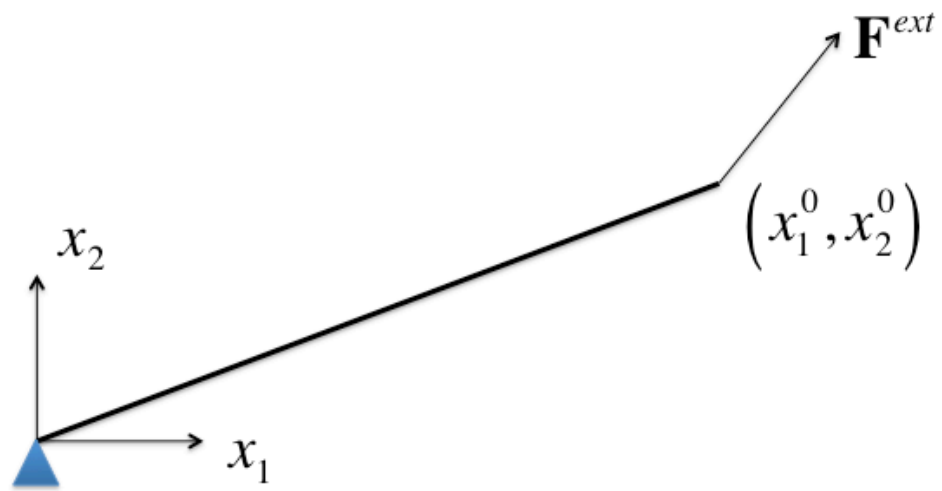
Fig. 1.3 Model problem with infinitesimal motions superposed on large rigid body motions.

It is instructive to proceed as though we do not know the solution summarized in (1.33) and formulate the equilibrium equations governing d_1 and d_2.

If we observe that the elongation δ of the rod can be written as

$$\delta = \sqrt{\left(d_1 + x_1^0\right)^2 + \left(d_2 + x_2^0\right)^2} - L_0, \tag{1.34}$$

then (1.31) gives the relationship between $\|\mathbf{F}^{ext}\|$ and the unknown displacements. Furthermore, as noted above, the direction of $\mathbf{F}^{ext}$ is given by

$$\frac{\mathbf{F}^{ext}}{\|\mathbf{F}^{ext}\|} = \frac{1}{\sqrt{\left(d_1 + x_1^0\right)^2 + \left(d_2 + x_2^0\right)^2}} \begin{bmatrix} d_1 + x_1^0 \\ d_2 + x_2^0 \end{bmatrix}. \tag{1.35}$$

Combining these facts gives the equation that governs d_1 and d_2,

$$\begin{bmatrix} F_1^{ext} \\ F_2^{ext} \end{bmatrix} = EA_0 \frac{\sqrt{\left(d_1 + x_1^0\right)^2 + \left(d_2 + x_2^0\right)^2} - L_0}{L_0 \sqrt{\left(d_1 + x_1^0\right)^2 + \left(d_2 + x_2^0\right)^2}} \begin{bmatrix} d_1 + x_1^0 \\ d_2 + x_2^0 \end{bmatrix}. \tag{1.36}$$

The reader may wish to verify this equation by substituting the solution (1.33) into (1.36).

(1.36) is a nonlinear, vector-valued equation for the unknowns d_1 and d_2. Recalling the generic form for nonlinear equations we introduced in the one dimensional case in (1.19), we could write this generically as

$$\mathbf{N}(\mathbf{d}) = \mathbf{F}^{ext}, \tag{1.37}$$

where

$$\mathbf{d} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \tag{1.38}$$

and

$$\mathbf{N}(\mathbf{d}) := EA_0 \frac{\sqrt{\left(d_1 + x_1^0\right)^2 + \left(d_2 + x_2^0\right)^2} - L_0}{L_0 \sqrt{\left(d_1 + x_1^0\right)^2 + \left(d_2 + x_2^0\right)^2}} \begin{bmatrix} d_1 + x_1^0 \\ d_2 + x_2^0 \end{bmatrix}. \tag{1.39}$$

Just as was done in the last section for the one degree of freedom case, we could introduce a Newton-Raphson strategy to solve (1.37) via

$$\mathbf{K}(\mathbf{d}^k)\Delta d = **R**(\mathbf{d}^k) = \mathbf{F}^{ext} - \mathbf{N}(\mathbf{d}^k), \tag{1.40}$$

and

$$\mathbf{d}^{k+1} = \mathbf{d}^k + \Delta\mathbf{d}, \tag{1.41}$$

where

$$\mathbf{K}(\mathbf{d}^k) := \frac{\partial \mathbf{N}}{\partial \mathbf{d}}(\mathbf{d}^k) = \left[\begin{array}{cc} \frac{\partial N_1}{\partial d_1} & \frac{\partial N_1}{\partial d_2} \\ \\ \frac{\partial N_2}{\partial d_1} & \frac{\partial N_2}{\partial d_2} \end{array} \right]_{\mathbf{d}=\mathbf{d}^k} \tag{1.42}$$

Carrying out the calculation of $\mathbf{K}(\mathbf{d}^k)$ for the specific $\mathbf{N}(\mathbf{d})$ at hand gives

$$\mathbf{K}(\mathbf{d}^k) = \mathbf{K}_{direct}(\mathbf{d}^k) + \mathbf{K}_{geom}(\mathbf{d}^k). \tag{1.43}$$

$\mathbf{K}_{direct}(\mathbf{d}^k)$ is given by

$$\mathbf{K}_{direct}(\mathbf{d}^k) := \frac{EA_0}{\sqrt{\left(d_1 + x_1^0\right)^2 + \left(d_2 + x_2^0\right)^2}} \left[\begin{array}{cc} \left(d_1^k + x_1^0\right)^2 & \left(d_1^k + x_1^0\right)\left(d_2^k + x_2^0\right) \\ \left(d_1^k + x_1^0\right)\left(d_2^k + x_2^0\right) & \left(d_2^k + x_2^0\right)^2 \end{array} \right] \tag{1.44}$$

and $\mathbf{K}_{geom}(\mathbf{d}^k)$ by

$$\mathbf{K}_{geom}(\mathbf{d}^k) := EA_0 \left(\frac{1}{L_0} - \frac{1}{\sqrt{\left(d_1 + x_1^0\right)^2 + \left(d_2 + x_2^0\right)^2}} \right) \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right]. \tag{1.45}$$

As the notation suggests, $\mathbf{K}_{direct}$ is sometimes referred to as the **direct stiffness**, or that part of the stiffness emanating directly from the material stiffness of the system at hand. $\mathbf{K}_{geom}$, on the other hand, is sometimes called the **geometric stiffness**, and arises not from inherent stiffness of the material but by virtue of the large motions in the problem.

To gain insight into these issues in the current context, consider the case where $\|\mathbf{d}^k\| \ll \|\mathbf{x}^0\|$, the case where the motions are small in comparison to the rod's length. In this case we find

$$\mathbf{K}_{geom}(\mathbf{d}^k) \to 0, \tag{1.46}$$

and

$$\mathbf{K}_{direct}(\mathbf{d}^k) \to \frac{EA_0}{L_0} \left[\begin{array}{cc} \cos\theta\cos\theta & \cos\theta\sin\theta \\ \cos\theta\sin\theta & \cos\theta\cos\theta \end{array} \right], \tag{1.47}$$

where $\theta = \arctan\left(\frac{x_2^0}{x_1^0}\right)$ is the angle between the original axis of the rod and the positive x-axis. In other words, when the motions become small, the geometric stiffness vanishes and the direct stiffness reduces to the familiar stiffness matrix associated with a two-dimensional truss member.

1.5 Contact Nonlinearity

A final type of nonlinearity we wish to consider is that created due to contact with another deformable or rigid body. As a simple model problem for this case we refer to Fig. 1.4, where we consider a prescribed motion $\bar{d}$ of the left end of our one-dimensional rod and solve for the static equilibrium of the unknown displacement d of the right end, subject to the constraint

$$g(d) = d - g_0 \leq 0, \tag{1.48}$$

where g_0 is the initial separation, or **gap**, between the right end of the rod and the rigid obstacle.
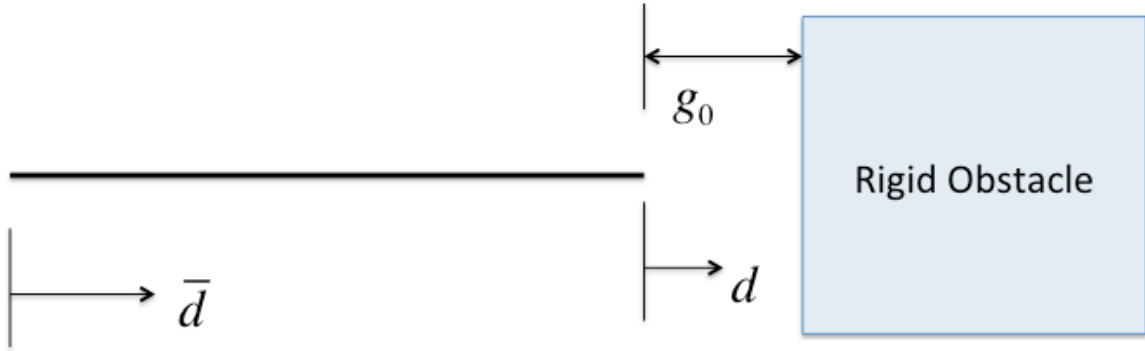
Fig. 1.4 Schematic of the rigid obstacle problem.

Even if we assume that the motions are small and the material response of the rod is elastic, the equations governing the response of our rod are nonlinear. To see this, let us choose d as our unknown and construct the following residual $R(d)$ for our system:

$$R(d) = \frac{EA_0}{L_0}(d - \bar{d}) + F_c, \tag{1.49}$$

22

Here F_c, the contact force between the obstacle and the rod (assumed positive in compression), is subject to the constraints

$$F_c \geq 0; \ g(d) \leq 0 \ \text{ and } F_c g(d) = 0 \,. \tag{1.50}$$

Equations (1.50) are called **Kuhn-Tucker complementary conditions** in mathematical parlance and physically require that the contact force be compressive, that the rod end not penetrate the obstacle, and that the contact force only be nonzero when $g = 0$, i.e. when contact between the rod and obstacle occurs. In fact F_c is a **Lagrange multiplier** in this problem, enforcing the kinematic constraint (1.48). We see that the condition operating on the right end of the bar is neither a Dirichlet nor a Neumann boundary condition; in fact, both the stress and the displacement at this point are unknown but are related to each other through the constraints expressed in Equations (1.50).

Plots of the residual defined by Equations (1.49) and (1.50) are given in Fig. 1.5 for the two distinct cases of interest: where contact does not occur (when $\bar{d} < g_0$) and where contact does occur (when $\bar{d} \geq g_0$). The solutions (i.e. the zeros of R) are readily apparent. When no contact occurs $d = \bar{d}$, while in the case of contact $d = g_0$. The internal stresses generated in the bar are then readily deduced.

One may note from Fig. 1.5 some important practical features of this problem. First, in both cases the admissible region for d is restricted to be less than g_0. Second, at the value $d = g_0$, each diagram shows the residual be multiple valued, which is a direct consequence of the fact that in this condition (i.e., where $g = 0$), F_c can be any positive number.

Finally, although the solution to our simple model problem is readily guessed, we can see from both cases that the plot of R versus d is only piecewise linear; the kink in each diagram indicates the fact that a finite tangent stiffness operates when contact is not active, changing to an infinite effective stiffness imposed by Equations (1.50) when contact between the rod and obstacle is detected. This contact detection therefore becomes an important feature in general strategies for contact problems, and introduces both nonlinearities and non smoothness into the global equations as this rather simple example demonstrates.

The books [15], [16], [26], [34], [38], [45], [48] are suggested for those readers wishing to reinforce their knowledge of linear elasticity, elementary continuum mechanics, and/or fundamentals of solid mechanics. They are presented in alphabetical order, with no other significance to be attached to the order of presentation.
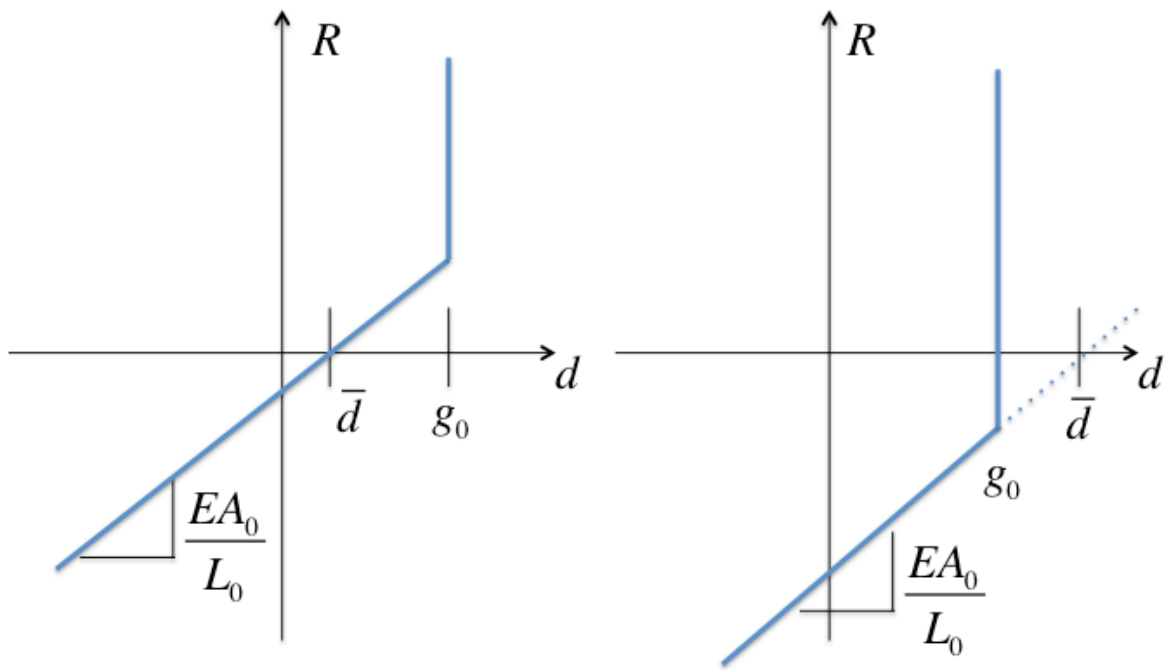
Fig. 1.5 Plots of residuals verses displacement for the rigid obstacle problem: (a) the case where $\bar{d} < g_0$ **(no contact); (b) the case where** $\bar{d} \geq g_0$**.**

2 Linear Elastic Initial/Boundary Value Problem

2.1 Basic Equations of Linear Elasticity

Having reviewed some relevant nonlinearities in the context of a simple structural element in Chapter Section 1, let us begin to generalize our problem description to encompass a larger group of continuous bodies. We begin this development by first reviewing the basic equations of *linear elasticity*, where we assume small motions and linear material behavior. This discussion will provide the basis for a more general notational framework in the next section, where we will remove the kinematic restriction to small motions and also allow the material to behave in an inelastic manner.

The notation we will use in this section is summarized in Fig. 2.1, where we have depicted a solid body positioned in the three dimensional Euclidean space, or $\mathbb{R}^3$. The set of spatial points $\mathbf{x}$ defining the body is denoted by Ω, and we consider the boundary $\partial\Omega$ to be subdivided into two regions Γ_u and Γ_σ, where Dirichlet and Neumann boundary conditions will be specified as discussed below. We assume that these regions obey the following:

$$\Gamma_u \cup \Gamma_\sigma = \partial\Omega$$
$$\Gamma_u \cap \Gamma_\sigma = \emptyset. \tag{2.1}$$

The unknown, or independent, variable in this problem is $\mathbf{u}$, the vector-valued displacement which in general depends upon $\mathbf{x} \in \Omega$ and time t.

2.2 Equations of Motion

At any point Ω the following statement of local linear momentum balance must hold:

$$\nabla \cdot \mathbf{T} + \mathbf{f} = \rho \frac{\partial^2 \mathbf{u}}{\partial t^2}. \tag{2.2}$$

Note that $\nabla \cdot \mathbf{T}$ denotes the divergence operator applied to $\mathbf{T}$, the Cauchy stress tensor. The vector $\mathbf{f}$ denotes the distributed body force in Ω, with units of force per volume, and ρ denotes the mass density, which need not be uniform. (2.2) represents the balance of linear momentum in direct notation. Balance of angular momentum is enforced within the domain by requiring that the Cauchy stress tensor is symmetric. We will frequently employ **index notation** in the work that follows. Toward that end, (2.2) can be expressed as

$$T_{ij,j} + f_i = \rho \frac{\partial^2 u_i}{\partial t^2}, \tag{2.3}$$

where indices i and j run between 1 and 3 (the spatial directions), and unless otherwise indicated, repeated indices within a term of an expression imply a summation over that index. For example,

$$T_{ij,j} = \sum_{j=1}^{3} \frac{\partial T_{ij}}{\partial x_j}. \tag{2.4}$$
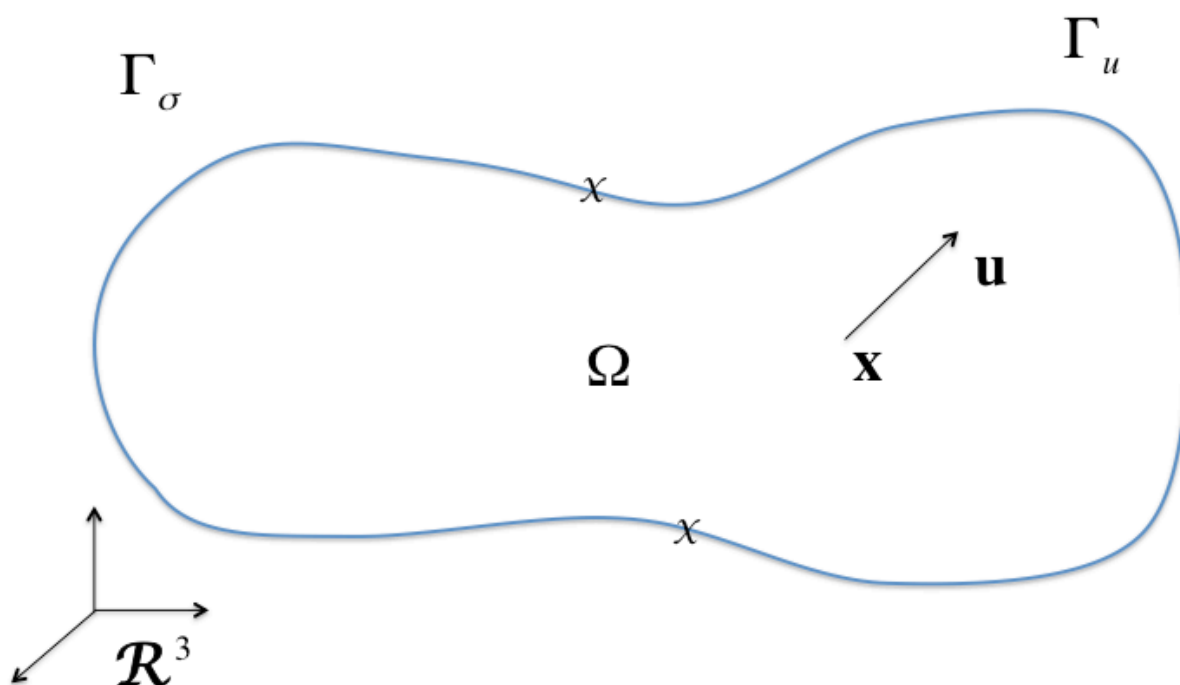
Fig. 2.1 Notation for the linear elastic initial/boundary value problem.

The notation $\beta_{,j}$ indicates partial differentiation with respect to x_j.

As indicated above the independent variables are u_i, so it is necessary to specify the relation between the displacements and the Cauchy stress. In linear elasticity this is accomplished by two additional equations. The first is the linear strain-displacement relation

$$\epsilon_{ij} = u_{(i,j)} = \frac{1}{2}(u_{i,j} + u_{j,i}), \tag{2.5}$$

where ϵ_{ij} is the infinitesimal strain equal to the symmetric part of the **displacement gradient** denoted by $u_{(i,j)}$. The second equation is the linear **constitutive relation** between T_{ij} and ϵ_{ij}, which is normally written

$$T_{ij} = C_{ijkl}\epsilon_{kl}. \tag{2.6}$$

Note that C_{ijkl} is the fourth-order elasticity tensor, to be discussed further below.

(2.5) and (2.6) can also be written in direct notation as

$$\boldsymbol{\epsilon} = \nabla_s \mathbf{u} = \frac{1}{2}\left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right), \tag{2.7}$$

where ∇_s denotes the symmetric gradient operator defined by $\nabla_s \square = 1/2\left(\nabla\square + \nabla\square^T\right)$, and

$$\mathbf{T} = \mathbf{C} : \boldsymbol{\epsilon}, \tag{2.8}$$

where the colon indicates double contraction of the fourth-order tensor $\mathbf{C}$ with the second-order tensor $\boldsymbol{\epsilon}$.

The fourth-order elasticity tensor $\mathbf{C}$ is ordinarily assumed to possess a number of symmetries, which greatly reduces the number of independent components that describe it. It possesses **major symmetry**, which means $C_{ijkl} = C_{klij}$, and it also possesses **minor symmetries**, meaning for example that $C_{ijkl} = C_{jikl} = C_{jilk} = C_{ijlk}$. Another important property of the elasticity tensor is **positive definiteness**, implying in this context that

$$A_{ij}C_{ijkl}A_{kl} > 0 \quad \text{for all symmetric tensors } A \tag{2.9}$$

$$\text{and } A_{ij}C_{ijkl}A_{kl} = 0 \quad \text{iff } A = 0. \tag{2.10}$$

In the most general case, assuming the aforementioned symmetries and no others, the elasticity tensor has 21 independent components. Various material symmetries reduce the number greatly, the most specific case being an **isotropic** material possessing rotational symmetry in all directions. In this case only two independent elastic constants are required to specify $\mathbf{C}$, which under these circumstances can be written as

$$C_{ijkl} = \lambda\delta_{ij}\delta_{kl} + \mu\left[\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right], \tag{2.11}$$

where δ_{ij}, the **Kronecker delta**, satisfies

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise,} \end{cases} \tag{2.12}$$

and λ and μ denote the **Lam'e parameters** for the material. These can be written in terms of the more familiar **Young's (i.e., elastic) modulus** and **Poisson's ratio** via

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \tag{2.13}$$

$$\mu = \frac{E}{2(1+\nu)}. \tag{2.14}$$

The quantity μ is also known as the **shear modulus** for the material.

Substitution of (2.7) and (2.8) into (2.2) gives a partial differential equation for the vector-valued unknown displacement field **u**. Full specification of the problem with suitable boundary and initial conditions is discussed next.

2.3 Boundary and Initial Conditions

Paralleling earlier discussion of the one-dimensional example, we will consider the possibility of two types of boundary conditions, Dirichlet and Neumann. Dirichlet boundary conditions will be imposed on the region Γ_u in Fig. 2.1 as

$$\mathbf{u}(\mathbf{x}, t) = \bar{\mathbf{u}}(\mathbf{x}, t) \quad \forall \mathbf{x} \in \Gamma_u, \quad t \in (0, T). \tag{2.15}$$

Note that $\bar{\mathbf{u}}(\mathbf{x}, t)$ denotes a prescribed displacement vector depending on spatial position and time. The simplest and perhaps most common example of such a boundary condition would be a fixed condition, which if imposed throughout the time interval of interest $(0, T)$ and for all of Γ_u would imply $\bar{\mathbf{u}}(\mathbf{x}, t) = \mathbf{0}$.

The other type of boundary condition is a Neumann, or traction, boundary condition. To write such a condition we must first define the concept of **traction** on a surface. If we use **n** to denote the outward normal to the surface Γ_σ at a point $\mathbf{x} \in \Gamma_\sigma$, the traction vector **t** at **x** is defined via

$$\mathbf{t} = \mathbf{T} \cdot \mathbf{n}, \tag{2.16}$$

or, in index notation,

$$T_i = T_{ij} n_j. \tag{2.17}$$

Physically this vector represents a force per unit area acting on the external surface at **x**. A Neumann boundary condition is then written in the current notation as

$$\mathbf{T}(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) = \bar{\mathbf{t}}(\mathbf{x}, t) \quad \forall \mathbf{x} \in \Gamma_\sigma, \quad t \in (0, T). \tag{2.18}$$

Note that $\bar{\mathbf{t}}(\mathbf{x}, t)$ is the prescribed traction vector field on Γ_σ throughout the time interval of interest $(0, T)$. One could identify several examples of such a boundary condition. An unfixed surface free of any external force would be described by $\bar{\mathbf{t}} = \mathbf{0}$. A surface subject to a uniform pressure loading, p, on the other hand, could be described by setting $\bar{\mathbf{t}}(\mathbf{x}, t) = -p\mathbf{n}(\mathbf{x})$, where we assume a compressive pressure to be positive.

With these definitions in hand, we recall the restrictions in (2.1) on Γ_u and Γ_σ and physically interpret them as follows: 1) one must specify either a traction or a displacement boundary condition at every point of $\partial\Omega$; and 2) at each point of $\partial\Omega$ one may not specify both the traction and the displacement but must specify one or the other.

In fact these conditions are slightly more stringent than required. The problem remains well-posed if, for each component direction i, we specify either the traction component $\bar{t}_i$ or the displacement component $\bar{u}_i$ at each point $\mathbf{x} \in \partial\Omega$, as long as for a given spatial direction we do not attempt to specify both. In other words, we may specify a displacement boundary condition in one direction at a point while specifying a traction boundary condition in the other. An example of such a case would be the common roller boundary condition, where a point is free to move in a traction-free manner to an interface (i.e., a traction boundary condition) while being constrained from movement in a direction normal to an interface (i.e., a displacement boundary condition). Of course a multitude of other boundary condition permutations could be identified. Thus, while we choose a rather simple boundary condition restriction (summarized by (2.1)) for notational simplicity, it is important to realize that many other possibilities exist and require only minor alterations of the methodology we will discuss.

The final important ingredient in our statement of the linear elastic problem is the specification of initial conditions. One may note that our partial differential equation ((2.2)) is second order in time; accordingly, two initial conditions are required. In the current context these are the initial conditions on the displacement $\mathbf{u}$ and the velocity $\dot{\mathbf{u}}$ and can be rather straightforwardly specified as

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) \quad \text{on } \Omega \tag{2.19}$$

$$\frac{\partial \mathbf{u}}{\partial t}(\mathbf{x}, 0) = \mathbf{v}_0(\mathbf{x}) \quad \text{on } \Omega, \tag{2.20}$$

where $\mathbf{u}_0$ and $\mathbf{v}_0$ are the prescribed initial displacement and velocity fields, respectively.

2.4 Problem Specification

We now collect the equations and conditions of the past two sections into a single problem statement for the linear elastic system shown in Fig. 2.1. For the elastodynamic case, this problem falls into the category of an **initial/boundary value problem**, since both types of conditions are included in its definition. Our problem is formally stated as follows:

Given the boundary conditions $**\bar{t}**$ on $\Gamma_\sigma \times (0, T)$ and $**\bar{u}**$ on $\Gamma_u \times (0, T)$, the initial conditions $\mathbf{u}_0$ and $\mathbf{v}_0$ on Ω, and the distributed body force $\mathbf{f}$ on $\Omega \times (0, T)$, find the displacement field $\mathbf{u}$ on $\Omega \times (0, T)$ such that

$$\nabla \cdot \mathbf{T} + \mathbf{f} = \rho \frac{\partial^2 \mathbf{u}}{\partial t^2} \quad \text{on } \Omega \times (0, T), \tag{2.21}$$

$$\mathbf{u}(\mathbf{x}, t) = \bar{\mathbf{u}}(\mathbf{x}, t) \quad \text{on } \Gamma_u \times (0, T), \tag{2.22}$$

$$\mathbf{t}(\mathbf{x}, t) = \bar{\mathbf{t}}(\mathbf{x}, t) \quad \text{on } \Gamma_\sigma \times (0, T), \tag{2.23}$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) \quad \text{on } \Omega, \tag{2.24}$$

$$\frac{\partial \mathbf{u}}{\partial t}(\mathbf{x}, 0) = \mathbf{v}_0(\mathbf{x}) \quad \text{on } \Omega, \tag{2.25}$$

where the Cauchy stress, $\mathbf{T}$, is given by

$$\mathbf{T} = \mathbf{C} : (\nabla_s \mathbf{u}). \tag{2.26}$$

Equations (2.21) through (2.26) constitute a linear hyperbolic initial/boundary value problem for the independent variable $\mathbf{u}$.

2.5 The Quasistatic Approximation

Before leaving the elastic problem, it is worthwhile to discuss how our problem specification will change if inertial effects are negligible in the equilibrium equations. This special case is often referred to as the **quasistatic assumption** and considerably simplifies specification of the problem.

Simply stated, the quasistatic assumption removes the second temporal derivative of $\mathbf{u}$, i.e., acceleration, from (2.21), thereby also eliminating the need for initial conditions (Equations (2.24) and (2.25)). Such an approximation is appropriate when the loadings do not vary with time or when they vary over time scales much longer than the periods associated with the fundamental structural modes of Ω.

It is convenient, however, to maintain time in our description of the problem for two reasons: 1) the loadings $\bar{\mathbf{t}}$ and $\mathbf{f}$ and the displacement condition $\bar{\mathbf{u}}$ may still vary with time; and 2) when we consider more general classes of constitutive equations, we may wish to allow time dependence in the stress/strain response, e.g., in creep plasticity. Accordingly, we state below a boundary value problem appropriate for quasistatic response of a linear elastic system.

Given the boundary conditions $\bar{\mathbf{t}}$ on $\Gamma_\sigma \times (0, T)$, $\bar{\mathbf{u}}$ on $\Gamma_u \times (0, T)$, and the distributed body force $\mathbf{f}$ on $\Omega \times (0, T)$, find the displacement field $\mathbf{u}$ on $\Omega \times (0, T)$ such that

$$\nabla \cdot \mathbf{T} + \mathbf{f} = 0 \quad \text{on } \Omega \times (0, T), \tag{2.27}$$

$$\mathbf{u}(\mathbf{x}, t) = \bar{\mathbf{u}}(\mathbf{x}, t) \quad \text{on } \Gamma_u \times (0, T), \tag{2.28}$$

$$\mathbf{t}(\mathbf{x}, t) = \bar{\mathbf{t}}(\mathbf{x}, t) \quad \text{on } \Gamma_\sigma \times (0, T), \tag{2.29}$$

where the Cauchy stress, $\mathbf{T}$, is given by

$$\mathbf{T} = \mathbf{C} : (\nabla_s \mathbf{u}). \tag{2.30}$$

We note in that given a time $t \in (0, T)$, Equations (2.27) through (2.30) constitute a linear elliptic boundary value problem governing the independent variable $\mathbf{u}$.

3 Weak Forms

3.1 Introduction

A key feature of the finite element method is the form of the boundary value problem (or initial/boundary value problem in the case of dynamics) that is discretized. More specifically, the finite element method is one of a large number of **variational methods** that rely on the approximation of integral forms of the governing equations. In this chapter we briefly examine how such integral (alternatively, weak or variational) forms are constructed for the linear elastic system we introduced in Chapter Section 2.

3.2 Quasistatic Case

Consider the quasistatic case first, we recall (2.27) – (2.30) and explore an alternative manner in which the conditions can be stated. We consider a collection of vector-valued functions $\mathbf{w}$, which we call weighting functions for reasons that will soon be clear. We require that these functions $\mathbf{w} : \bar{\Omega} \to \mathbb{R}^3$ satisfy

$$\mathbf{w} = \mathbf{0} \text{ on } \Gamma_u. \tag{3.1}$$

Furthermore it is assumed that these functions are sufficiently smooth that all necessary partial derivatives can be computed. Suppose we have the solution $\mathbf{u}$ of (2.27) and (2.28). We can then take any smooth function $\mathbf{w}$ satisfying (3.1) and compute its dot product with (2.27), which must produce

$$\mathbf{w} \cdot (\nabla \cdot \mathbf{T} + \mathbf{f}) = 0 \text{ on } \Omega \tag{3.2}$$

at each time $t \in (0, T)$. We can then integrate (3.2) over Ω to obtain

$$\int_\Omega \mathbf{w} \cdot (\nabla \cdot \mathbf{T} + \mathbf{f}) d\Omega = 0. \tag{3.3}$$

(3.3) can be manipulated further by noting that

$$\mathbf{w} \cdot (\nabla \cdot \mathbf{T}) = \nabla \cdot (\mathbf{Tw}) - (\nabla \mathbf{w}) : \mathbf{T} \tag{3.4}$$

(product rule of differentiation), and by also taking advantage of the divergence theorem from multivariate calculus:

$$\int_\Omega \nabla \cdot (\mathbf{Tw}) \, d\Omega = \int_{\partial\Omega} (\mathbf{n} \cdot \mathbf{Tw}) \, d\Gamma. \tag{3.5}$$

Note that $\mathbf{n}$ is the outward normal directed normal on $\partial\Omega$ and $d\Gamma$ is a differential area of this surface. Use of (3.4) and (3.5) in (3.3) and rearranging gives

$$\int_\Omega (\nabla \mathbf{w}) : \mathbf{T} \, d\Omega = \int_\Omega \mathbf{w} \cdot \mathbf{f} \, d\Omega + \int_{\partial\Omega} (\mathbf{n} \cdot \mathbf{Tw}) \, d\Gamma. \tag{3.6}$$

Now, taking advantage of the symmetry of $\mathbf{T}$ and noting, from (2.16), that the surface traction $\mathbf{t}$ equals $\mathbf{Tn}$, we can write

$$\int_{\partial\Omega} (\mathbf{n} \cdot \mathbf{Tw}) \, d\Gamma = \int_{\partial\Omega} (\mathbf{w} \cdot \mathbf{Tn}) \, d\Gamma = \int_{\partial\Omega} \mathbf{w} \cdot \mathbf{t} \, d\Gamma. \tag{3.7}$$

We now recall the restrictions in (2.1), which tell us that $\partial\Omega$ is the union of Γ_u and Γ_σ. Since by definition $\mathbf{w} = 0$ on Γ_u, we can write

$$\int_{\partial\Omega} \mathbf{w} \cdot \mathbf{t} \, d\Gamma = \int_{\Gamma_u} \mathbf{w} \cdot \mathbf{t} \, d\Gamma + \int_{\Gamma_\sigma} \mathbf{w} \cdot \mathbf{t} \, d\Gamma = \int_{\Gamma_\sigma} \mathbf{w} \cdot \bar{\mathbf{t}} \, d\Gamma \tag{3.8}$$

where the last equality incorporates the boundary condition $\mathbf{t} = \bar{\mathbf{t}}$ on Γ_σ.

We collect these calculations to conclude that

$$\int_\Omega (\nabla \mathbf{w}) : \mathbf{T} \, d\Omega = \int_\Omega \mathbf{w} \cdot \mathbf{f} \, d\Omega + \int_{\Gamma_\sigma} \mathbf{w} \cdot \bar{\mathbf{t}} \, d\Gamma, \tag{3.9}$$

which must hold for the solution $\mathbf{u}$ of (2.27) – (2.30) for any $\mathbf{w}$ satisfying condition (3.1).

To complete our alternative statement of the boundary value problem, the concepts of solution and variational spaces need to be introduced. We define the **solution space** $\mathbb{S}_t$ corresponding to time t via

$$\mathbb{S}_t = \{\mathbf{u} \,|\, \mathbf{u} = \bar{\mathbf{u}}(t) \quad \text{on } \Gamma_u, \quad \mathbf{u} \text{ is smooth}\} \tag{3.10}$$

and the **weighting space** $\mathbb{W}$ as

$$\mathbb{W} = \{\mathbf{w} \,|\, \mathbf{w} = 0 \quad \text{on } \Gamma_u, \quad \mathbf{w} \text{ is smooth}\}. \tag{3.11}$$

With these two collections of functions in hand, we consider the following alternative statement of the boundary value problem summarized by (2.27) – (2.30):

Given the boundary conditions $\bar{\mathbf{t}}$ on $\Gamma_\sigma \times (0, T)$, $**\bar{u}**$ on $\Gamma_u \times (0, T)$ and the distributed body force $\mathbf{f}$ on $\Omega \times (0, T)$, find the $\mathbf{u} \in \mathbb{S}_t$ for each time $t \in (0, T)$ such that

$$\int_\Omega (\nabla \mathbf{w}) : \mathbf{T} \, d\Omega = \int_\Omega \mathbf{w} \cdot \mathbf{f} \, d\Omega + \int_{\Gamma_\sigma} \mathbf{w} \cdot \bar{\mathbf{t}} \, d\Gamma \tag{3.12}$$

for all $\mathbf{w} \in \mathbb{W}$, where $\mathbb{S}_t$ is as defined in (3.10), $\mathbb{W}$ is as defined in (3.11), and the Cauchy stress, $\mathbf{T}$, is given by

$$\mathbf{T} = \mathbf{C} : (\nabla_s \mathbf{u}). \tag{3.13}$$

This statement of the boundary value problem is often referred to as a weak formulation, since it explicitly requires only a weighted integral of the governing partial differential equations to be zero, rather than the differential equation itself.

It should be clear, based upon the above derivation of the weak form, that the solution $\mathbf{u}$ of (2.27) – (2.30), sometimes referred to as the **strong form**, will satisfy our alternative statement

summarized by (3.12) and (3.13). Less clear is the fact that solutions of the weak form will satisfy the strong form whenever this formulation admits a solution. Since the continuity requirements for existence of a strong solution are more stringent than for the analogous weak formulation (hence the adjective strong), equivalence between these two forms is restricted to the case when both exist, i.e., whenever a solution of the strong form of the boundary value problem exists, then a weak solution also exists, and these solutions are identical.

It is important to note that the existence of a solution to the weak form of the boundary value problem does not necessarily imply existence of a solution to the strong form. The strong form's constraints upon solution smoothness imply that for some problems (e.g., point sources that induce jumps in derivative terms), a weak form might exist, but no strong form can be constructed without substantially revising some basic principles of differential calculus. So the existence of a weak solution does not necessarily imply that an identical strong solution exists: only that if a strong solution can be found, it will be identical to the weak solution.

In practice, the existence of a weak solution in these cases turns out to be one of the most important advantages of finite element techniques, because the integral formulations that form the mathematical foundation of finite element approximations permit accurate simulation of important problems that are not readily solved via competing differential techniques derived from strong formulations. Many of the most important problems of computational mechanics (e.g., contact, material discontinuity, structural failure) often admit only weak solutions, and that is one of the main reasons why weak formulations are important in practice.

So the equivalence between strong and weak forms is restricted to those cases where strong solutions exist, and in that case, the strong solution is identical to the analogous weak solution. Although not shown here this equivalence can be rigorously established; the interested reader should consult Reference [26] at the end of this chapter for details. We simply remark in the present discussion that the equivalent argument depends crucially on the satisfaction of (3.12) for all $\mathbf{w} \in \mathbb{W}$, with the arbitrariness of $\mathbf{w}$ rendering the two statements equivalent whenever the strong solution exists.

Given the requirement of efficient numerical implementation, we can also remark that approximate methods will in effect narrow our definitions of the solution and weighting spaces to finite-dimensional subspaces. Simply stated, this means that rather than including an infinite number of smooth $\mathbf{u}$ and $\mathbf{w}$ satisfying the requisite boundary conditions in our problem definition, we will restrict our attention to some finite number of functions comprising subsets of $\mathbb{S}_t$ and $\mathbb{W}$.

In so doing we introduce a difference between the solution of our (now approximate) weak form and the strong form, where the degree of approximation is directly related to the difference between the full solution and weighting spaces and the subsets of them used in the numerical procedure. In fact it is this difference that is at the heart of solution verification, an important activity to ensure that an appropriate subset of spaces (i.e., discretization or mesh refinement) is chosen. Solution verification as part of the broader question of verification is discussed in the Solid Mechanics Verification Manual.

Finally, it is worthwhile at this point to make a connection to so-called **virtual work** methods which may be more familiar to those versed in linear structural mechanics. In this derivation we will work in index notation so that the meaning of the direction notation used above can be

reinforced. Accordingly, for a possible solution u_i of the governing equations, we write the expression for the total potential energy of the system,

$$P(u_i) = \frac{1}{2} \int_\Omega u_{(i,j)} C_{ijkl} u_{(k,l)} \mathrm{d}\Omega - \left[\int_\Omega u_i f_i \, \mathrm{d}\Omega - \int_{\Gamma_\sigma} u_i \bar{t}_i \, \mathrm{d}\Gamma \right]. \tag{3.14}$$

Note that the first term on the right hand side represents the total **strain energy** associated with u_i and the last two terms represent the potential energy of the applied loadings f_i and $\bar{t}_i$. A **virtual work principle** for this system simply states that the potential energy defined in (3.14) should be minimized by the equilibrium solution. Accordingly, let u_i now represent the actual equilibrium solution. We can represent any other kinematically admissible displacement field via $u_i + \epsilon w_i$, where ϵ is a scalar parameter (not necessarily small) and w_i is a so-called virtual displacement, which we assume to obey the boundary conditions outlined in (3.1). This restriction on the w_i causes $u_i + \epsilon w_i$ to satisfy the Dirichlet boundary conditions (hence the term kinematically admissible) because the solution u_i does. We can write the total energy associated with any of these possible solutions via

$$P(u_i + \epsilon w_i) = \frac{1}{2} \int_\Omega \left(u_{(i,j)} + \epsilon w_{(i,j)} \right) C_{ijkl} \left(u_{(k,l)} + \epsilon w_{(k,l)} \right) \mathrm{d}\Omega$$
$$- \int_\Omega (u_i + \epsilon w_i) \, f_i \, \mathrm{d}\Omega - \int_{\Gamma_\sigma} (u_i + \epsilon w_i) \, \bar{t}_i \, \mathrm{d}\Gamma. \tag{3.15}$$

Note that if the potential energy associated with u_i is to be lower that that of any other possible solution $u_i + \epsilon w_i$, then the derivative of $P(u_i + \epsilon w_i)$ with respect to ϵ at $\epsilon = 0$ (i.e., at the solution u_i) should be zero for any w_i satisfying the conditions in (3.1), since u_i is an extremum point of the function P. Computing this derivative of (3.15), and setting the result equal to zero, yields

$$\frac{d}{d\epsilon}\bigg|_{\epsilon=0} P(u_i + \epsilon w_i) = \int_\Omega w_{(i,j)} C_{ijkl} u_{(k,l)} \mathrm{d}\Omega - \int_\Omega w_i f_i \, \mathrm{d}\Omega - \int_{\Gamma_\sigma} w_i \bar{t}_i \, \mathrm{d}\Gamma = 0 \tag{3.16}$$

which must hold for all w_i satisfying the boundary condition on Γ_u. (3.16) can be manipulated further by noting that

$$w_{(i,j)} C_{ijkl} u_{(k,l)} = w_{(i,j)} C_{ijkl} E_{kl} = w_{(i,j)} T_{ij} = w_{i,j} T_{ij}. \tag{3.17}$$

The last equality in (3.17), while perhaps not intuitively obvious, holds because of the symmetry of T_{ij}:

$$w_{(i,j)} T_{ij} = \frac{1}{2}(w_{i,j} + w_{j,i}) T_{ij} = \frac{1}{2}(w_{i,j} T_{ij} + w_{j,i} T_{ji}) = w_{i,j} T_{ij}. \tag{3.18}$$

Use of (3.17) in (3.16) yields

$$\int_\Omega w_{i,j} T_{ij} \mathrm{d}\Omega - \int_\Omega w_i f_i \, \mathrm{d}\Omega - \int_{\Gamma_\sigma} w_i \bar{t}_i \, \mathrm{d}\Gamma = 0, \tag{3.19}$$

which is simply the index notation counterpart of (2.27). Summarizing, we see that the weak or integral form of the governing equations developed previously can be interpreted as a statement of

the principle of minimum potential energy. This alternative viewpoint is the reason that the weighting functions w_i are sometimes called variations or virtual displacements, with the terminology used often depending upon the mathematical and physical arguments used to develop the weak form.

Despite the usefulness of this physical interpretation, it should be noted that the presence of an energy principle is somewhat specific to the case at hand and may be difficult or impossible to deduce for many of the nonlinear systems to be considered in our later study. For example, many systems are not conservative, including those featuring inelasticity, so at best our thermodynamic understanding must be expanded if we insist on formulating such problems in terms of energy principles. Thus, while the energy interpretation is enlightening for many systems, including those featuring elastic continuum and/or structural response, insistence on this approach for more general applications of variational methods can be quite limiting. Conversely, the derivation given in (3.2) – (3.9) does not depend on the system being conservative, nor even upon the form of the constitutive equation used. We will exploit the generality of this weighted residual derivation as we increase the level of nonlinearity and complexity in the chapters to come.

3.3 Fully Dynamic Case

Another advantage of the weighted residual approach is that it can be straightforwardly applied to dynamic problems. Before examining the dynamic case in detail, whose development parallels that of quasistatic problems, it is worthwhile to emphasize again the definitions of the weighting and solution spaces and to highlight the differences between them. Examining the definition of $\mathbb{S}_t$ in (3.10) and that of $\mathbb{W}$ in (3.11), we see that $\mathbb{S}_t$ depends on t through the boundary conditions on Γ_u, while $\mathbb{W}$ is independent of time. We retain these definitions in the current case and pose the following problem corresponding to the quasistatic system posed previously:

Given the boundary conditions $\bar{\mathbf{t}}$ on $\Gamma_\sigma \times (0, T)$ and $\bar{\mathbf{u}}$ on $\Gamma_u \times (0, T)$, the initial conditions $\mathbf{u}_0$ and $\mathbf{v}_0$ on Ω, and the distributed body force $\mathbf{f}$ on $\Omega \times (0, T)$, find the $\mathbf{u} \in \mathbb{S}_t$ for each time $t \in (0, T)$ such that

$$\int_\Omega \rho \mathbf{w} \cdot \frac{\partial^2 \mathbf{u}}{\partial t^2}\, d\Omega + \int_\Omega (\nabla \mathbf{w}) : \mathbf{T}\, d\Omega = \int_\Omega \mathbf{w} \cdot \mathbf{f}\, d\Omega + \int_{\Gamma_\sigma} \mathbf{w} \cdot \bar{\mathbf{t}}\, d\Gamma \tag{3.20}$$

for all $\mathbf{w} \in \mathbb{W}$, where $\mathbb{S}_t$ is as defined in (3.10), $\mathbb{W}$ is as defined in (3.11), and the Cauchy stress, $\mathbf{T}$, is given by

$$\mathbf{T} = \mathbf{C} : (\nabla_s \mathbf{u}). \tag{3.21}$$

In addition, the solution $\mathbf{u}$ is subject to the following conditions at $t = 0$:

$$\int_\Gamma \mathbf{w} \cdot (\mathbf{u}(0) - \mathbf{u}_0)\, d\Omega = 0 \tag{3.22}$$

and

$$\int_\Gamma \mathbf{w} \cdot \left(\frac{\partial \mathbf{u}}{\partial t}(0) - \mathbf{v}_0 \right) d\Omega = 0, \tag{3.23}$$

both of which must hold for all $\mathbf{w} \in \mathbb{W}$.

The integral form of the dynamic equations given in (3.20) is obtained, just as in the quasistatic case, by taking the dynamic governing partial differential equation, (2.21), multiplying it by a weighting function, integrating over the body, and applying integration by parts to the stress divergence term. The new ingredients in the current specification are the initial conditions summarized by (3.22) and (3.23), which are simple weighted residual expressions of the strong form of the initial conditions given in (2.25).

Before leaving this section, we reemphasize the fact that the weighting functions are time independent while the solution spaces remain time dependent. This fact will have important consequences later when numerical algorithms are discussed, because we wish to use the same classes of functions in our discrete representations of $\mathbb{W}$ and $\mathbb{S}_t$. These discretizations will involve spatial approximation, which in the case of $\mathbb{S}_t$ will leave the time variable continuous in the discrete unknowns of the system to be solved.

This **semi-discrete** approach to transient problems is pervasive in computational mechanics and has its origin in the difference between the weighting and solution spaces.

The reference for this chapter is [26].

4 Large Deformation Framework

4.1 Introduction

In this chapter and the next several chapters we extend our discussion of the linear elastic problem to accommodate two categories of important nonlinearities: potentially large motions and deformations, and nonlinear material response. We will do this by introducing a more general notational framework. While the equations governing large deformation initial/boundary value problems are similar in form to their counterparts from the small deformation theory just discussed, a rigorous prescription and understanding of large deformation problems can only be achieved through a careful examination of the concepts of nonlinear continuum mechanics, which will be the concern of the next several chapters.

The organization of this material is as follows. This chapter establishes a notational framework for the generic specification of a nonlinear solid mechanics problem. Section 5 and Section 6 discuss **large deformation kinematics** in a general context. Section 7 will then discuss the various measures of stress that are frequently encountered in large deformation analysis. Then, with these preliminaries in hand, we will be in a position to state relevant balance laws in notation appropriate for large deformation problems in Section 8. Finally, in Section 9, we will discuss the important concept of material frame indifference, which demands that material laws be unaltered by rigid body motions. We will see that this concept places important restrictions on the kinematic and stress measures that are suitable for prescription of constitutive laws, providing important background information for the chapter on material models.

4.2 Notational Framework

The system we wish to consider is depicted schematically in Fig. 4.1. We consider a body, initially in a location denoted by Ω, undergoing a time dependent motion φ that describes its trajectory through space (assumed here to be $\mathbb{R}^3$).

The set Ω is called the **reference configuration** and can be thought of as consisting of points $\mathbf{X}$ that serve as labels for the material points existing at their respective locations. For this reason, the coordinates $\mathbf{X}$ are often called **reference** or **material coordinates**.

We assume, as before, that the surface $\partial\Omega$ of Ω can be decomposed into subsets Γ_σ and Γ_u obeying restrictions in (2.1). The general interpretation of these surfaces remains the same. Traction boundary conditions will be imposed on Γ_σ and displacement boundary conditions will be imposed on Γ_u. Full specification of these boundary conditions must be deferred, however, until some continuum mechanical preliminaries are discussed.

We have mentioned that the motion φ is in general time dependent. In fact, we could write this fact in mathematical terms as $\varphi : \bar{\Omega} \times (0, T) \to \mathbb{R}^3$. If we *fix* the time argument of φ, we obtain a **configuration mapping** φ_t, summarized as $\varphi_t : \bar{\Omega} \to \mathbb{R}^3$, which gives us the location of the body
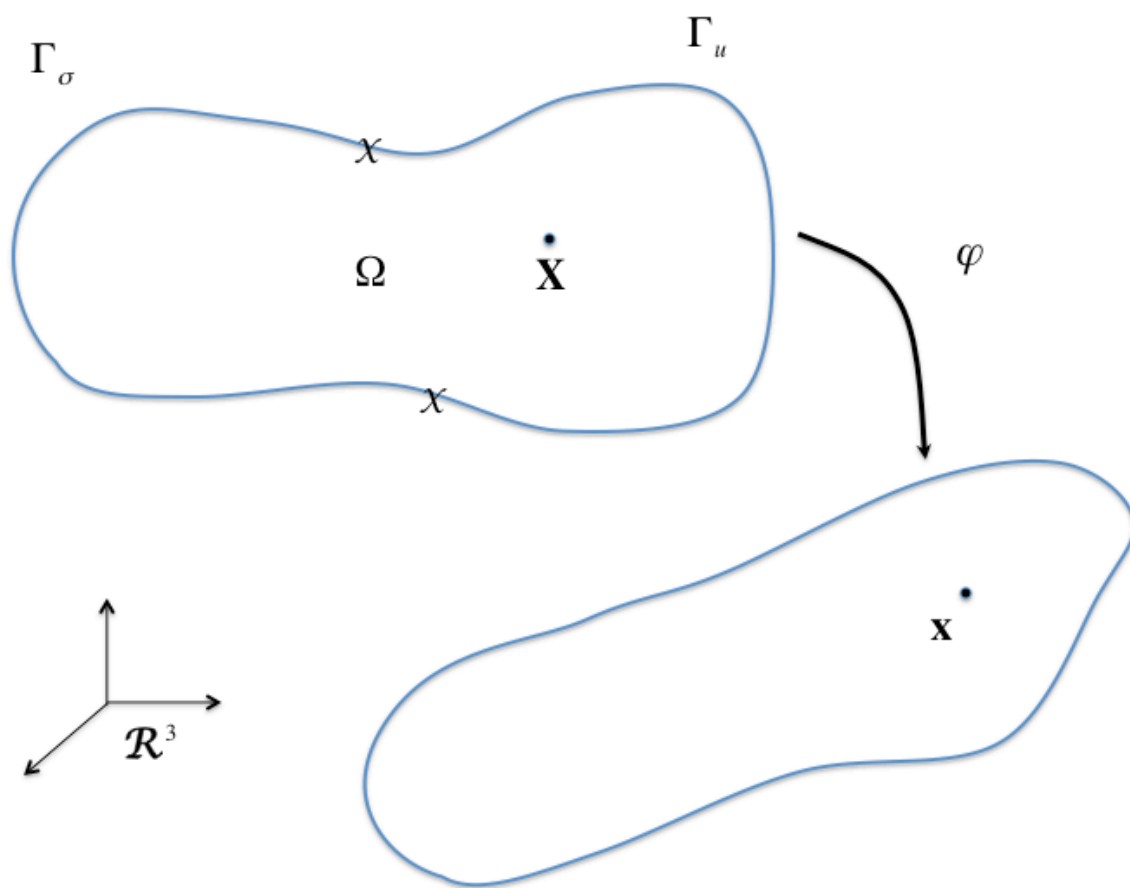
Fig. 4.1 Notation for large deformation initial/boundary value problems.

at time t given the reference configuration Ω. Coordinates in the current location $\varphi(\Omega)$ of the body will be denoted by $\mathbf{x}$.

The current location is often called the **spatial configuration** and the coordinates, **x spatial coordinates**. Given a material point $\mathbf{X} \in \Omega$ and a configuration mapping φ_t, we may write

$$\mathbf{x} = \varphi_t(\mathbf{X}). \tag{4.1}$$

A key decision in writing the equations of motion for this system is whether to express the equations in terms of $\mathbf{X} \in \Omega$ or $\mathbf{x} \in \varphi_t(\Omega)$.

4.3 Lagrangian and Eulerian Descriptions

The choice of whether to use the reference coordinates $\mathbf{X}$ or the spatial coordinates $\mathbf{x}$ in the problem description is generally highly dependent on the physical system to be studied.

For example, suppose we wish to write the equations of motion for a gas flowing through a duct, or for a fluid flowing through a nozzle. In these cases the physical region of interest (the control volume bounded by the duct or nozzle) is fixed, and does not depend on the solution or time. It could also be observed that identification of individual particle trajectories in such problems is probably not of primary interest, with such quantities as pressure, velocity, and temperature at particular locations in the flow field being more desirable. In such problems, it is generally most appropriate to associate field variables and equations with spatial points, or in the current notation, $\mathbf{x}$. A system described in this manner is said to be utilizing the **Eulerian description** and implicitly associates all field variables and equations with spatial points $\mathbf{x}$ without specific regard for the material points $\mathbf{X}$ involved in the flow of the problem. Most fluid and gas dynamics problems are written in this way, as are problems in **hydrodynamics** and some problems in solid mechanics involving fully developed plastic flow.

When thinking of Eulerian coordinate systems, it is sometimes useful to invoke the analogy of watching an event through a window; the window represents the Eulerian frame and has our coordinate system attached to it. Particles pass through our field of view, thereby defining a flow, but we describe this flow from the frame of reference of our window without specific reference to the particles undergoing the motion we observe.

In most solid mechanics applications, by contrast, the identity of specific material particles is of central interest in modeling a system. For example, the plastic response of metals is history dependent, meaning that the current relationship between stress and strain (the **material model**) at a point in the body depends on the deformation history associated with that material point. To construct and use such models effectively requires knowledge of the history of individual particles, or material points, throughout a *deformation process*. Furthermore, many physical processes we wish to describe do not lend themselves to an invariant Eulerian frame. In a forging process, for example, the metal at the end of the procedure occupies a very different region in space than it did at the outset. In addition, there may be periods of time over which boundary conditions are applied requiring precise knowledge of the boundary of the region of interest. For

these reasons, as well as others, the predominant approach to solid mechanics systems is to write all equations in terms of the material coordinates, or to use the **Lagrangian** frame of reference.

Returning to the notation summarized in Fig. 4.1, for a Lagrangian description we associate all field variables and equations with points $\mathbf{X} \in \Omega$, and keep track of these reference particles throughout the process. One may note in the last subsection a bias toward this approach already. We have written the primary unknown in the problem, φ, as a function of $\mathbf{X} \in \Omega$ and $t \in (0, T)$. Sierra/SM uses the **Lagrangian** frame of reference though as we will see next, the spatial frame is also of great interest to us.

4.4 Governing Equations in the Spatial Frame

We turn now to the equations governing the motion of a medium. If we adopt for the moment the spatial frame as our frame of reference, the form of these equations is largely unchanged from the linear elastic case presented previously (where we explicitly took advantage of the fact that for linear problems there is no difference between material and spatial descriptions). We fix our attention on some time $t \in (0, T)$ and consider the current (unknown) location of the body Ω. Over this region $\varphi_t(\Omega)$, the following conditions must hold:

$$\nabla \cdot \mathbf{T} + \mathbf{f} = \rho \mathbf{a} \quad \text{on } \varphi_t(\Omega), \tag{4.2}$$

$$\varphi_t = \bar{\varphi}_t \quad \text{on } \varphi_t(\Omega_u), \tag{4.3}$$

and

$$\mathbf{t} = \bar{\mathbf{t}} \quad \text{on } \varphi_t(\Omega_\sigma), \tag{4.4}$$

subject to initial conditions at $t = 0$. Some explanation of these equations is necessary. The operator ∇ in (4.2) is with respect to spatial coordinates $\mathbf{x}$. The acceleration $\mathbf{a}$ is the acceleration of the particle currently at $\mathbf{x}$ written with respect to spatial coordinates, and $\bar{\varphi}_t$ is the prescribed location for the particles on the Dirichlet boundary. We leave the constitutive law governing $\mathbf{T}$ unspecified at this point but remark that in general the stress must depend on φ_t through appropriate strain/displacement and stress/strain relations.

We see from (4.2) through (4.4) that the equations of motion are easily written in the form inherited from the kinematically linear case, but that the frame in which this is done, the spatial frame, is *not* independent of the unknown field φ_t but relies upon it for its own definition. Thus, although the equations we now consider are essentially identical in form to those from linear elasticity, they posses a considerably more complex relationship to the dependent variable. Rigorous specification of this general boundary value problem requires an in-depth treatment of the continuum mechanics of large deformation, as will be provided in the next chapters.

Before leaving this topic, we address an item which frequently causes confusion. Although we have written the governing equations in (4.2) through (4.4) in terms of the spatial domain, this does not imply an Eulerian statement of the problem. In fact, if we choose (as we have done) to consider our dependent variable (in this case φ_t) to be a function of reference coordinates, the

framework we have chosen is inherently Lagrangian. Another way of saying this is that (4.2) through (4.4) are the Lagrangian equations of motion which have been converted through a change-of-variables so that they are written in terms of **x**. In the remainder of this text, the reader should assume a Lagrangian framework unless otherwise noted.

This page left blank

5 Deformation Measures

5.1 Deformation Gradient

Furthering our discussion of large deformation solid mechanics, we continue to use the notation presented in Fig. 4.1. We restrict our attention to some time $t \in (0, T)$, and consider the corresponding configuration mapping φ_t, which can be mathematically represented via $\varphi_t : \bar{\Omega} \to \mathbb{R}_3$. The **deformation gradient F** is given by the gradient of this transformation,

$$\mathbf{F} = \frac{\partial \varphi_t}{\partial \mathbf{X}}, \tag{5.1}$$

or in index notation,

$$F_{iJ} = \frac{\partial \varphi_{ti}}{\partial X_J}. \tag{5.2}$$

In (5.2) and throughout this documented unless otherwise noted, lower case indices are associated with coordinates in the spatial frame and upper case indices with material coordinates. Repeated indices of either case imply summation.

The deformation gradient is the most basic object used to quantify the local deformation at a point in a solid. Most kinematic measures and concepts we will discuss rely on it explicitly for their definition. For example, elementary calculus provides a physical interpretation of the determinant of $\mathbf{F}$. Consider a cube of material in the reference configuration (see Fig. 5.1) whose sides are assumed to be aligned with the coordinate axes $X_I, I = 1, 2, 3$. The initial differential volume $\mathrm{d}V$ of this cube is given by

$$\mathrm{d}V = \mathrm{d}X_1 \mathrm{d}X_2 \mathrm{d}X_3. \tag{5.3}$$

If we now consider the condition of this cube of material after the deformation φ_t is applied, we notice that its volume in the current configuration $\mathrm{d}v$ is that of the parallelepiped spanned by the three vectors $\varphi_t(\overrightarrow{\mathrm{d}X_J})$, where the notation $\overrightarrow{\mathrm{d}X_J}$ is used to indicate a reference vector in coordinate direction J with magnitude $\mathrm{d}X_J$. This volume can be written in terms of the vector triple product,

$$\mathrm{d}v = \varphi_t(\overrightarrow{\mathrm{d}X_1}) \cdot \varphi_t(\overrightarrow{\mathrm{d}X_2}) \cdot \varphi_t(\overrightarrow{\mathrm{d}X_3}). \tag{5.4}$$

If we consider any differential vector $\overrightarrow{\mathrm{d}R}$ in the reference configuration, the calculus of differentials tells us that application of the mapping φ_t will produce a differential vector $\overrightarrow{\mathrm{d}r} = \varphi_t(\overrightarrow{\mathrm{d}R})$ whose coordinate are given by

$$(\overrightarrow{\mathrm{d}r})_i = \frac{\partial \varphi_{ti}}{\partial X_K}(\overrightarrow{\mathrm{d}R})_K. \tag{5.5}$$
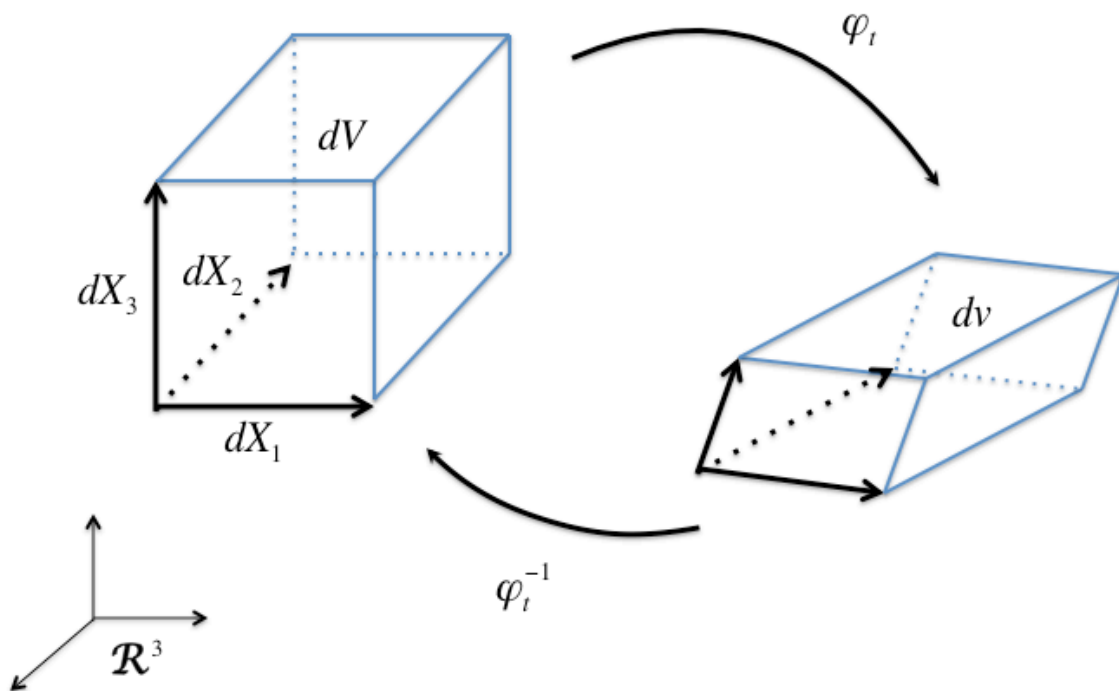
Fig. 5.1 Deformation of a volume element as described by the configuration mapping φ_t.

Application of this logic to the particular differential vectors $\overrightarrow{\mathrm{d}R_J}$ leads one to conclude that

$$(\varphi_t(\overrightarrow{\mathrm{d}X_J}))_i = \begin{cases} F_{i1}\mathrm{d}X_1, & J = 1, \\ F_{i2}\mathrm{d}X_2, & J = 2, \\ F_{i3}\mathrm{d}X_3, & J = 3. \end{cases} \tag{5.6}$$

We can write (5.4) in index notation by first noting that the cross product of two vectors **a** and **b** is written as

$$(\mathbf{a} \times \mathbf{b})_i = e_{ijk}a_j b_k, \tag{5.7}$$

where e_{ijk}, the **permutation symbol**, is defined as

$$e_{ijk} = \begin{cases} 1 & \text{if } (i, j, k) = (1, 2, 3) \text{ or } (2, 3, 1) \text{ or } (3, 1, 2), \\ -1 & \text{if } (i, j, k) = (3, 2, 1) \text{ or } (2, 1, 3) \text{ or } (1, 3, 2), \\ 0 & \text{otherwise.} \end{cases} \tag{5.8}$$

(5.4) can then expressed as

$$\begin{aligned} \mathrm{d}v &= F_{i1}\mathrm{d}X_1(e_{ijk}F_{j2}\mathrm{d}X_2 F_{k3}\mathrm{d}X_3) \\ &= e_{ijk}F_{i1}F_{j2}F_{k3}\mathrm{d}X_1\mathrm{d}X_2\mathrm{d}X_3 \\ &= \det(\mathbf{F})\mathrm{d}V, \end{aligned}$$

where we have used (5.3) and the fact that $\det(\mathbf{F}) = e_{ijk}F_{i1}F_{j2}F_{k3}$ (which can be verified through trial). Introducing the notation $J = \det(\mathbf{F})$, we conclude

$$\mathrm{d}v = J\mathrm{d}V. \tag{5.9}$$

(5.9) tells us that the deformation φ_t converts reference differential volumes $\mathrm{d}V$ to current volumes $\mathrm{d}v$ according to the determinant of the deformation gradient. For this mapping to make physical sense, the current volume $\mathrm{d}v$ should be positive which then places a physical restriction upon the deformation gradient **F** that must be obeyed point wise throughout the domain,

$$J = \det(\mathbf{F}) = \det\left(\frac{\partial\varphi}{\partial\mathbf{X}}\right) > 0. \tag{5.10}$$

This physical restriction has important mathematical consequences as well. According to the inverse function theorem of multivariate calculus, a smooth function whose gradient has a nonzero determinant possesses a smooth and differentiable inverse. Since we have assumed φ_t to be smooth and physical restrictions demand that $J > 0$, we can conclude that a function φ_t^{-1} exists and is differentiable; in fact, the gradient of this function is given by

$$\frac{\partial\varphi_t^{-1}}{\partial\mathbf{X}} = \mathbf{F}^{-1}. \tag{5.11}$$

We will assume throughout the remainder of our discussion that $J > 0$, so that such an inverse is guaranteed to exist.

5.2 Polar Decomposition

With the definition of $\mathbf{F}$ in hand, we turn our attention to the quantification of local deformation in a body. For any matrix such as $\mathbf{F}$, whose determinant is positive, the following decomposition can always be made:

$$\mathbf{F} = \mathbf{R}\mathbf{U} = \mathbf{V}\mathbf{R}. \tag{5.12}$$

In (5.12), $\mathbf{R}$ is a proper orthogonal tensor (right-handed rotation), while $\mathbf{U}$ and $\mathbf{V}$ are positive-definite, symmetric tensors. One can show that under the conditions stated, the decompositions in (5.12) can always be made and that they are unique. The interested reader should consult Reference [20] of Chapter 1 for details. The decompositions $\mathbf{R}\mathbf{U}$ and $\mathbf{V}\mathbf{R}$ in (5.12) are called right and left **polar decompositions** of $\mathbf{F}$, respectively. $\mathbf{R}$ is often called the rotation tensor, while $\mathbf{U}$ and $\mathbf{V}$ are sometimes referred to as the right and left stretches.

The significance of the polar decomposition is made more clear in Fig. 5.2, where we consider the deformation of a neighborhood of material surrounding a point $\mathbf{X} \in \Omega$. (5.5) shows that the full deformation gradient maps arbitrary reference differentials into their current positions at time t. By considering the polar decomposition, we see that the deformation of material neighborhoods of infinitesimal extent can always be conceptualized in two ways. In the right polar decomposition, $\mathbf{U}$ contains all information necessary to describe the distortion of a neighborhood of material, while $\mathbf{R}$ then maps this distorted neighborhood into the current configuration through pure (right-handed) rotation. On the other hand, in the left polar decomposition, the rotation $\mathbf{R}$ is considered first followed by the distortion $\mathbf{V}$. In developing measures of local deformation, we can thus focus on either $\mathbf{U}$ or $\mathbf{V}$. The choice of which decomposition to use is typically based on the coordinates in which we wish to write the strains. The right stretch $\mathbf{U}$ most naturally takes reference coordinates as arguments, while the left stretch $\mathbf{V}$ is ordinarily written in terms of spatial coordinates. This can be expressed as

$$\mathbf{F}(\mathbf{X}) = \mathbf{R}(\mathbf{X})\mathbf{U}(\mathbf{X}) = \mathbf{V}(\varphi(\mathbf{X}))\mathbf{R}(\mathbf{X}). \tag{5.13}$$

In characterizing large deformations, it is also convenient to define the right and left Cauchy-Green tensors via

$$\mathbf{C} = \mathbf{F}^T\mathbf{F} \tag{5.14}$$

and

$$\mathbf{B} = \mathbf{F}\mathbf{F}^T. \tag{5.15}$$

The right Cauchy-Green tensor is ordinarily considered to be a material object $\mathbf{C}(\mathbf{X})$, while the left Cauchy-Green tensor is a spatial object $\mathbf{B}(\varphi_t(\mathbf{X}))$. Since $\mathbf{R}$ is orthogonal, one can write

$$\mathbf{R}^T\mathbf{R} = \mathbf{R}\mathbf{R}^T = \mathbf{I}, \tag{5.16}$$

$$dx = F \cdot dX$$

$$F$$

$$dX$$

$$X$$

$$U$$

$$d\breve{x} = U \cdot dX$$

$$\breve{x}$$

$$dx$$

$$x = \varphi_t(X)$$

$$R$$

$$dx = R \cdot d\breve{x} = R \cdot U \cdot dX$$

$$\mathcal{R}^3$$
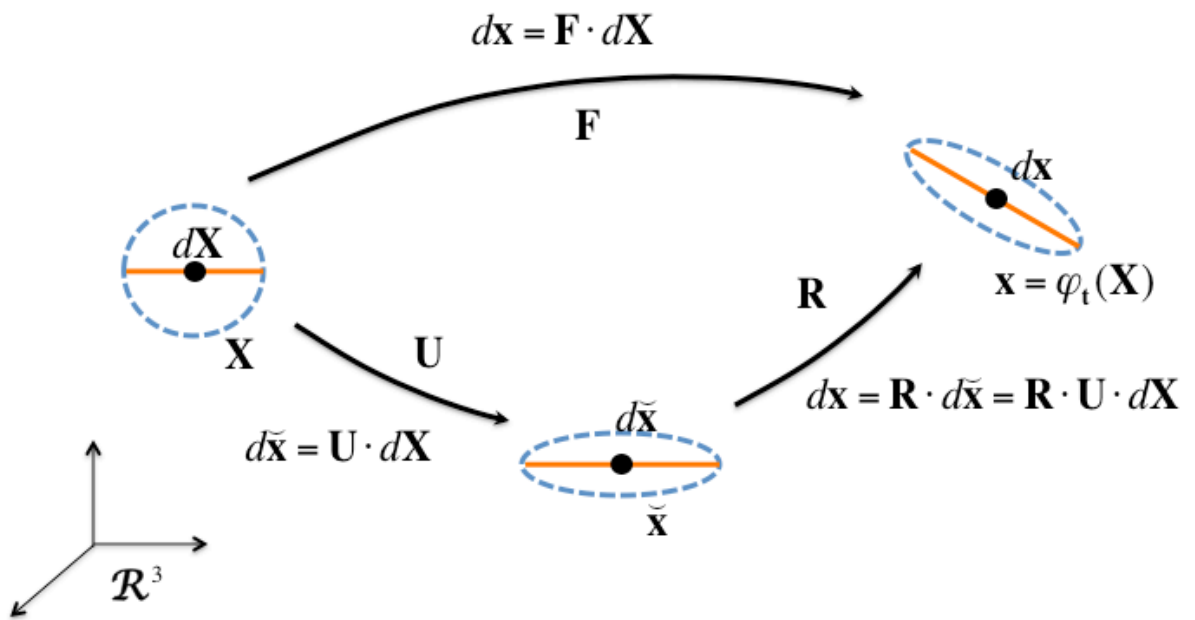
Fig. 5.2 Dotted outline indicates infinitesimal neighborhood of point X.)

where $\mathbf{I}$ is the 3×3 identity tensor. Manipulating (5.13) through (5.15) reveals that

$$\mathbf{U} = \mathbf{C}^{\frac{1}{2}} \tag{5.17}$$

and

$$\mathbf{V} = \mathbf{B}^{\frac{1}{2}}. \tag{5.18}$$

One can see the connection with small strain theory by considering the **Green strain tensor E** defined with respect to the reference configuration,

$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I}). \tag{5.19}$$

We define the reference configuration displacement field $\mathbf{u}$, such that

$$\mathbf{u}(\mathbf{X}) = \varphi(\mathbf{X}) - \mathbf{X}. \tag{5.20}$$

Working in index notation, we write $\mathbf{E}$ in terms of $\mathbf{u}$

$$\begin{aligned}
E_{IJ} &= \frac{1}{2}(C_{IJ} - \delta_{IJ}) = \frac{1}{2}\left(F_{iI}F_{iJ} - \delta_{IJ}\right) \\
&= \frac{1}{2}\left(\frac{\partial}{\partial X_I}(u_i + X_i)\frac{\partial}{\partial X_J}(u_i + X_i) - \delta_{IJ}\right) \\
&= \frac{1}{2}\left(\left(\frac{\partial u_i}{\partial X_I} + \delta_{iI}\right)\left(\frac{\partial u_i}{\partial X_J} + \delta_{iJ}\right) - \delta_{IJ}\right) \\
&= \frac{1}{2}\left(\delta_{iI}\frac{\partial}{\partial X_J}(u_i) + \delta_{iJ}\frac{\partial}{\partial X_I}(u_i) + \frac{\partial u_i}{\partial X_I}\frac{\partial u_j}{\partial X_J}\right)
\end{aligned} \tag{5.21}$$

In the case where the displacement gradients are small, i.e., $|\frac{\partial U_i}{\partial X_J}| \ll 1$, the quadratic term in (5.21) will be much smaller that the terms linear in the displacement gradients. If, in addition, the displacement components u_i are very small when compared with the size of the body, then the distinction between reference and spatial coordinates becomes unnecessary and (5.21) simplifies to

$$E_{IJ} \approx \frac{1}{2}\left(\frac{\partial u_I}{\partial X_J} + \frac{\partial u_J}{\partial X_I}\right), \tag{5.22}$$

which is identical to the infinitesimal case (cf. (2.5)).

The references for Chapter 5 are [15, 20, 39].

6 Rates of Deformation

The development of the last chapter fixed our attention on an instant $t \in (0, T)$, and proposed some measurements of material deformation in terms of the configuration mapping φ_t. We now allow time to vary and consider two questions:

- How are velocities and accelerations quantified in both the spatial and reference frames?

- How are time derivatives of deformation measures properly considered in a large deformation framework?

The former topic is obviously crucial in the formulation of dynamics problems, while the latter is necessary, for example, in rate-dependent materials where quantities such as strain rate must be quantified.

6.1 Material and Spatial Velocity and Acceleration

One obtains the **material velocity V** and the **material acceleration A** by fixing attention on a particular material particle (i.e., fixing the reference coordinate **X**), and then considering successive (partial) time derivatives of the motion $\varphi(\mathbf{X}, t)$. This can be written mathematically as

$$\mathbf{V}(\mathbf{X}, t) = \frac{\partial}{\partial t} \big(\varphi(\mathbf{X}, t) \big) \tag{6.1}$$

and

$$\mathbf{A}(\mathbf{X}, t) = \frac{\partial}{\partial t} \big(\mathbf{V}(\mathbf{X}, t) \big) = \frac{\partial^2}{\partial t^2} \big(\varphi(\mathbf{X}, t) \big). \tag{6.2}$$

Note in (6.1) and (6.2) that **V** and **A** take **X** as their first argument, hence their designation as material quantities. A Lagrangian description of motion, in which reference coordinates are the independent variables, would most naturally use these measures of velocity and acceleration.

An Eulerian description, on the other hand, generally requires measures written in terms of spatial points **x** without requiring explicit knowledge of material points **X**. The **spatial velocity v** and the **spatial acceleration a** are obtained from (6.1) and (6.2) through a change in variables:

$$\mathbf{v}(\mathbf{x}, t) = \mathbf{V}\left(\varphi_t^{-1}(\mathbf{x}), t \right) = \mathbf{V}_t \cdot \varphi_t^{-1}(\mathbf{x}) \tag{6.3}$$

and

$$\mathbf{a}(\mathbf{x}, t) = \mathbf{A}\left(\varphi_t^{-1}(\mathbf{x}), t \right) = \mathbf{A}_t \cdot \varphi_t^{-1}(\mathbf{x}). \tag{6.4}$$

The expression given in (6.4) for the spatial acceleration may be unfamiliar to those versed in fluid mechanics who may be more accustomed to thinking of acceleration as the **total time derivative** of the spatial velocity **v**. We reconcile these different viewpoints here through the introduction of

the equivalent concept of the **material time derivative**, defined, in general, as the time derivative of any object, spatial or material, taken so that the identity of the material particle is held fixed. Applying this concept to the spatial velocity gives

$$
\begin{aligned}
\mathbf{a}(\mathbf{x}, t) &= \dot{\mathbf{v}}(\mathbf{x}, t)|_{\mathbf{x} = \varphi(\mathbf{X}, t)} \\
&= \left. \frac{d}{dt} \right|_{\mathbf{X} \text{ fixed}} \mathbf{v}\left(\varphi(\mathbf{X}, t), t\right) \\
&= \frac{\partial \mathbf{v}}{\partial \mathbf{x}}(\mathbf{x}, t) \cdot \frac{\partial \varphi}{\partial t}\left(\varphi_t^{-1}(\mathbf{x}), t\right) + \frac{\partial \mathbf{v}}{\partial t}\left(\varphi_t^{-1}(\mathbf{x}), t\right) \\
&= \frac{\partial \mathbf{v}}{\partial t} + \nabla \mathbf{v} \cdot \mathbf{v}.
\end{aligned}
\tag{6.5}
$$

This may be recognized as the so-called total time derivative of the spatial velocity $\mathbf{v}$. Exercising the concept of a material time derivative a little further, we can see from (6.1) that the material velocity is the material time derivative of the motion, so that

$$
\mathbf{V} = \dot{\varphi}.
\tag{6.6}
$$

Comparing (6.2) and (6.5), we conclude that $\mathbf{A}$ and $\mathbf{a}$ are, in fact, the same physical entity expressed in different coordinates. The former is most naturally written in terms of $\mathbf{V}$, while the latter is conveniently expressed in terms of $\mathbf{v}$.

(6.5) uses the superposed dot notation for the time derivative of $\mathbf{v}$. Such superposed dots will always imply a material time derivative in this document, whether applied to material or spatial quantities. Furthermore, the gradient $\nabla \mathbf{v}$ is taken with respect to spatial coordinates and is called the **spatial velocity gradient**. It is used often enough to warrant the special symbol $\mathbf{L}$:

$$
\mathbf{L} := \nabla \mathbf{v}.
\tag{6.7}
$$

6.2 Rate of Deformation Tensors

From the spatial velocity gradient $\mathbf{L}$ defined in (6.7), we define two spatial tensors $\mathbf{D}$ and $\mathbf{W}$, known respectively as the **spatial rate of deformation tensor** and the **spatial spin tensor**:

$$
\mathbf{D} := \nabla_s \mathbf{v} = \frac{1}{2}[\mathbf{L} + \mathbf{L}^T],
\tag{6.8}
$$

and

$$
\mathbf{W} := \nabla_a \mathbf{v} = \frac{1}{2}[\mathbf{L} - \mathbf{L}^T].
\tag{6.9}
$$

It is clear that $\mathbf{D}$ is merely the symmetric part of the velocity gradient, while $\mathbf{W}$ is the antisymmetric, or skew, portion.

The quantities $\mathbf{D}$ and $\mathbf{W}$ are called spatial measures of deformation. $\mathbf{D}$ is effectively a measure of strain rate suitable for large deformations, while $\mathbf{W}$ provides a local measure of the rate of rotation

of the material. In fact, in small deformations it is readily verified that (6.8) amounts to nothing more than the time derivative of the infinitesimal strain tensor defined in (2.5). It is of interest at this point to discuss whether appropriate material counterparts of these objects exist. Toward this end, we calculate the material time derivative of the **deformation gradient F**. If **F** is an analytic function, the order of partial differentiation can be reversed:

$$\dot{\mathbf{F}} = \frac{\partial}{\partial t}\left[\frac{\partial}{\partial \mathbf{X}}\varphi(\mathbf{X},t)\right] = \frac{\partial}{\partial \mathbf{X}}\left[\frac{\partial}{\partial t}\varphi(\mathbf{X},t)\right] = \frac{\partial \mathbf{V}}{\partial \mathbf{X}}. \tag{6.10}$$

From (6.10), we conclude that the material time derivative $\dot{\mathbf{F}}$ is nothing more than the material velocity gradient. Manipulating this quantity further gives

$$\frac{\partial \mathbf{V}}{\partial \mathbf{X}} = \frac{\partial}{\partial \mathbf{X}}\left(\mathbf{v}\circ\varphi_t\right) = \nabla\mathbf{v}\left(\varphi_t(\mathbf{X})\right)\frac{\partial}{\partial \mathbf{X}}\left(\varphi_t(\mathbf{X})\right) = \mathbf{L}\left(\varphi_t(\mathbf{X})\right)\mathbf{F}(\mathbf{X}). \tag{6.11}$$

Examination of (6.10) and (6.11) reveals that

$$\mathbf{L} = \left(\dot{\mathbf{F}}\cdot\varphi_t^{-1}\right)\mathbf{F}^{-1}. \tag{6.12}$$

Recalling the definition for the right Cauchy-Green strain tensor **C** in (5.14) Section 5, we compute its material time derivative via

$$\dot{\mathbf{C}} = \frac{\partial}{\partial t}[\mathbf{F}^T\mathbf{F}] = \dot{\mathbf{F}}^T\mathbf{F} + \mathbf{F}^T\dot{\mathbf{F}} = (\mathbf{L}\mathbf{F})^T\mathbf{F} + \mathbf{F}^T(\mathbf{L}\mathbf{F}) = \mathbf{F}^T(\mathbf{L} + \mathbf{L}^T)\mathbf{F}. \tag{6.13}$$

which, in view of (6.8), leads us to conclude

$$\dot{\mathbf{C}}(\mathbf{X},t) = 2\mathbf{F}^T(\mathbf{X},t)\mathbf{D}(\varphi(\mathbf{X}),t)\mathbf{F}(\mathbf{X},t). \tag{6.14}$$

(6.14) reveals why $\frac{1}{2}\dot{\mathbf{C}}$ is sometimes called the **material rate of deformation tensor**. Noting that **F** is the Jacobian of the transformation φ_t, readers with a background in differential geometry will recognize $\frac{1}{2}\dot{\mathbf{C}}$ as the pull-back of the spatial tensor field **D** defined on $\varphi_t(\Omega)$. Conversely, **D** is the push-forward of the material tensor field $\frac{1}{2}\dot{\mathbf{C}}$ defined on Ω. The concepts of pull-back and push-forward are outside the scope of this document, but the physical principle they embody in the current context is perhaps useful. Loosely speaking, the push forward (or pull-back) of a tensor with respect to a given transformation produces a tensor in the new frame of reference that we, as observers, would observe as identical to the original tensor if we were embedded in the material during the transformation. Thus, the same physical principle is represented by both $\frac{1}{2}\dot{\mathbf{C}}$ and **D**, but they are very different objects mathematically since the transformation that interrelates them is the deformation itself. Recalling the definition of Green's strain **E** given in (5.19), we can easily see that

$$\dot{\mathbf{E}} = \frac{1}{2}\dot{\mathbf{C}} = \mathbf{F}^T\mathbf{D}\mathbf{F}. \tag{6.15}$$

This further substantiates the interpretation of **D** as a strain rate.

We have thus far developed measures of strain and strain rate appropriate for both the spatial and reference configurations. Now we consider appropriate definitions of these quantities for the

rotated configuration defined according to the polar decomposition and depicted schematically in Fig. 4.1. This can be done by applying the linear transformation $\mathbf{R}$ relating the rotated configuration to the spatial one.

The **rotated rate of deformation tensor D** is thus defined via

$$
\begin{aligned}
D(\mathbf{X}, t) &= \mathbf{R}^T(\mathbf{X}, t) \cdot \mathbf{D}(\varphi(\mathbf{X}, t), t) \cdot \mathbf{R}(\mathbf{X}, t) \\
&= \mathbf{R}^T(\mathbf{D} \circ \varphi)\mathbf{R}.
\end{aligned} \tag{6.16}
$$

Noting that

$$
\dot{\mathbf{C}} = 2\mathbf{F}^T(\mathbf{D} \circ \varphi)\mathbf{F} = 2\mathbf{U}^T\mathbf{R}^T\mathbf{D} \circ \varphi)\mathbf{R}\mathbf{U} = 2\mathbf{U}^T D\mathbf{U}, \tag{6.17}
$$

we find

$$
D = \frac{1}{2}\mathbf{U}^{-1}\dot{\mathbf{C}}\mathbf{U}^{-1} = \frac{1}{2}\mathbf{C}^{-1/2}\dot{\mathbf{C}}\mathbf{C}^{-1/2}. \tag{6.18}
$$

In connection with the rotated reference, another tensor, L, is sometimes introduced:

$$
L = \dot{\mathbf{R}}\mathbf{R}^T. \tag{6.19}
$$

Note that L is skew:

$$
L + L^T = \dot{\mathbf{R}}\mathbf{R}^T + \mathbf{R}\dot{\mathbf{R}}^T = \frac{\partial}{\partial t}(\mathbf{R}\mathbf{R}^T) = \frac{\partial \mathbf{I}}{\partial t} = 0. \tag{6.20}
$$

We will return later in this document to the various measures associated with the rotated configuration. They have particular importance in the study of material frame indifference.

7 Stress Measures

7.1 Cauchy Stress

In this chapter we discuss the quantification of force intensity, or stress, within a body undergoing potentially large amounts of deformation. We begin with the **Cauchy stress** tensor $\mathbf{T}$ and note that, provided we associate this object with the spatial configuration, this object can be interpreted exactly as in the infinitesimal case outlined in Section 2. In the current notational framework, we interpret the components of $\mathbf{T}$, denoted as T_{ij}, which represent forces per unit areas in the spatial configuration at a given spatial point $\mathbf{x} \in \varphi_t(\Omega)$.

It will be necessary in our description to consider related measures of stress defined in terms the reference and rotated configurations. To motivate this discussion, we reconsider the concept of traction discussed previously in the context of the infinitesimal elastic system. Recall that given a plane passing through the point of interest $\mathbf{x}$, the traction, or force per unit area acting on this plane, is given by the formula

$$t_i = T_{ij}n_j, \tag{7.1}$$

where n_j is the unit normal vector to the plane in question.

7.2 Nanson's Formula

We consider two differential vectors, $d\mathbf{r}_1$ and $d\mathbf{r}_2$, as illustrated in Fig. 7.1. We assume that $d\mathbf{r}_1$ and $d\mathbf{r}_2$ are linearly independent and that both have spatial point $\mathbf{x}$ as their base point. We further assume that their orientations are such that the following relation from basic geometry holds:

$$d\mathbf{r}_1 \times d\mathbf{r}_2 = \mathbf{n}da, \tag{7.2}$$

where da is the (differential) area of the parallelogram encompassed by $d\mathbf{r}_1$ and $d\mathbf{r}_2$.

As discussed in Section 5 (see (5.5)), we can think of the differential vectors $d\mathbf{r}_1$ and $d\mathbf{r}_2$ as the current positions of reference differential vectors $d\mathbf{R}_1$ and $d\mathbf{R}_2$, which are based at $\mathbf{X} = \varphi_t^{-1}(\mathbf{x})$. In index notation, we can relate these two sets of differential vectors using the deformation gradient via

$$(d\mathbf{r}_1)_i = F_{iI}(d\mathbf{R}_1)_I, \tag{7.3}$$

and

$$(d\mathbf{r}_2)_i = F_{iI}(d\mathbf{R}_2)_I. \tag{7.4}$$

We now seek to re-express (7.2) in terms of reference quantities. Working in index notation, we write

$$\begin{aligned}
n_i da &= e_{ijk}F_{jJ}(d\mathbf{R}_1)_J F_{kK}(d\mathbf{R}_2)_K \\
&= e_{ljk}\delta_{li}F_{jJ}(d\mathbf{R}_1)_J F_{kK}(d\mathbf{R}_2)_K \\
&= e_{ljk}F_{lL}F_{Li}^{-1}F_{jJ}(d\mathbf{R}_1)_J F_{kK}(d\mathbf{R}_2)_K
\end{aligned} \tag{7.5}$$
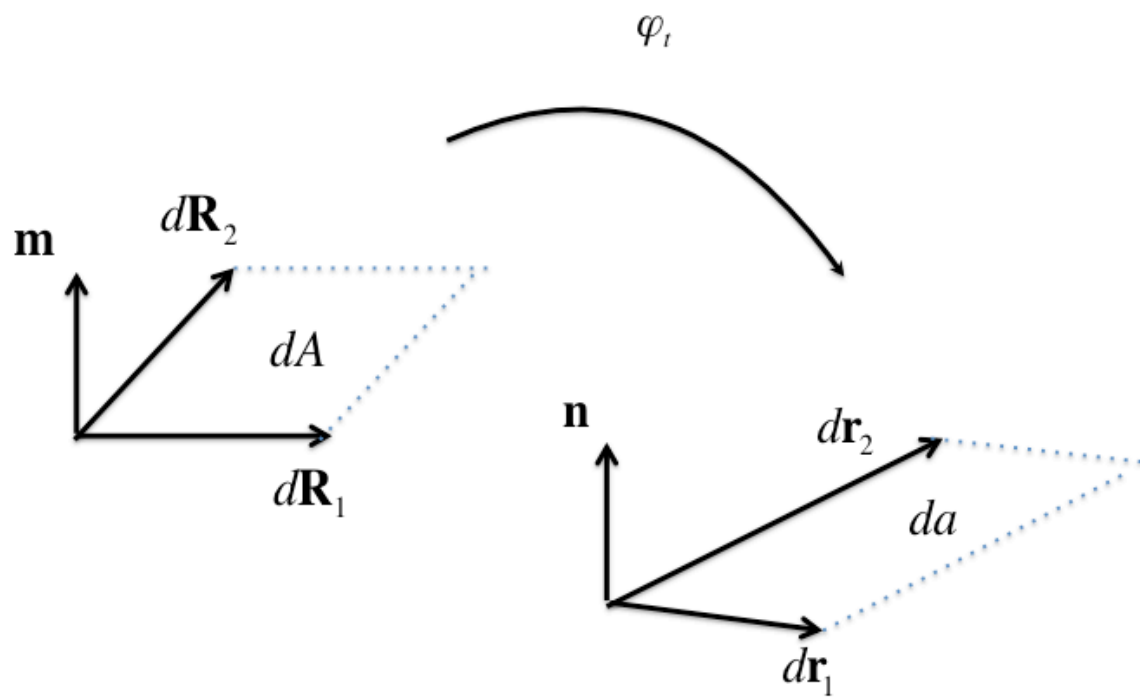
Fig. 7.1 Notation for derivation of Nanson's formula.

We extract and manipulate a particular product in the last line of (7.5), namely $e_{ljk}F_{lL}F_{jJ}F_{kK}$. One can show by a case-by-case examination that the following relation holds:

$$e_{ljk}F_{lL}F_{jJ}F_{kK} = e_{LJK}e_{ljk}F_{l1}F_{j2}F_{k3}.$$ (7.6)

Recall from Section 5, (5.10) that $J = \det(\mathbf{F})$ has the following representation in index notation:

$$J = \det(\mathbf{F}) = e_{ljk}F_{l1}F_{j2}F_{k3}$$ (7.7)

Combination of (7.5), (7.6), and (7.7) yields the following result:

$$\begin{aligned} n_i \mathrm{d}a &= Je_{LJK}F_{Li}^{-1}(\mathrm{d}\mathbf{R}_1)_J(\mathrm{d}\mathbf{R}_2)_K \\ &= JF_{Li}^{-1}m_L\mathrm{d}A. \end{aligned}$$ (7.8)

In (7.8), $\mathrm{d}A$ is the differential reference area spanned by $\mathrm{d}\mathbf{R}_1$ and $\mathrm{d}\mathbf{R}_2$, and $\mathbf{m}$ is the reference unit normal to this area.

In direct notation, we express this result as

$$\mathbf{n}\mathrm{d}a = J\mathbf{F}^{-T}\mathbf{m}\mathrm{d}A.$$ (7.9)

(7.9) is referred to as Nanson's formula and it is important, among other reasons, because it provides the appropriate change-of-variables formula for surface integrals in the reference and current configurations.

7.3 First and Second Piola-Kirchhoff Stress

We want to compute a differential force, which is the product of the traction acting on our plane at $\mathbf{x}$ and the differential area under consideration. Denoting this differential force by $\mathrm{d}\mathbf{f}$, we write

$$\mathrm{d}\mathbf{f} = \mathbf{t}\mathrm{d}a = \mathbf{T}\mathbf{n}\mathrm{d}a = J\mathbf{T}\mathbf{F}^{-T}\mathbf{m}\mathrm{d}A.$$ (7.10)

In examining (7.10), we find that the following definition is useful:

$$\mathbf{P}(\mathbf{X}) = J(\mathbf{X})\mathbf{T}\left(\varphi_t(\mathbf{X})\right)\mathbf{F}^{-T}\left(\varphi_t(\mathbf{X})\right).$$ (7.11)

This allows us to write

$$\mathrm{d}\mathbf{f} = \mathbf{P}\mathbf{m}\mathrm{d}A.$$ (7.12)

In (7.12), the product $\mathbf{P}\mathbf{m}$ represents a traction, the current force, $\mathrm{d}\mathbf{f}$, divided by the reference area, $\mathrm{d}A$. The tensor $\mathbf{P}$ is called the (**First**) **Piola-Kirchhoff Stress** and $\mathbf{P}\mathbf{m}$ is called the Piola Traction. Similar to the Piola Traction, the First Piola-Kirchhoff Stress measures stress in terms of forces in the current configuration and areas in the reference configuration. The one-dimensional manifestation of this stress measure is the engineering stress, σ_E, originally defined in (1.3).

It is worthy to note that $\mathbf{P}$ is neither a pure spatial nor a pure reference object. A reference object for stress can be constructed by performing a pull-back of the spatial Cauchy stress tensor $\mathbf{T}$ to the reference configuration:

$$\begin{aligned}
\mathbf{S}(\mathbf{X}) \quad &= J\mathbf{F}^{-1}\left(\varphi_t(\mathbf{X})\right)\mathbf{T}\left(\varphi_t(\mathbf{X})\right)\mathbf{F}^{-T}\left(\varphi_t(\mathbf{X})\right) \\
&= \mathbf{F}^{-1}\left(\varphi_t(\mathbf{X})\right)\mathbf{P}(\mathbf{X}).
\end{aligned} \tag{7.13}$$

$\mathbf{S}$ is called the **Second Piola-Kirchhoff Stress** and it is purely a reference object. We note, in particular, that $\mathbf{S}$ is a symmetric tensor, while $\mathbf{P}$ is not symmetric in general.

This same concept of pull-back can be employed to define a stress tensor in the rotated configuration, which we shall denote as $\boldsymbol{T}$. This rotated tensor is defined as

$$\boldsymbol{T} = \mathbf{R}^T\left(\varphi_t(\mathbf{X})\right)\mathbf{T}\left(\varphi_t(\mathbf{X})\right)\mathbf{R}\left(\varphi_t(\mathbf{X})\right). \tag{7.14}$$

As was the case with the rotated configuration quantities introduced in Section 6, this definition will be of particular importance in the later examination of frame indifference.

8 Balance Laws

In this chapter, we examine the local forms of the various conservation laws as expressed in the various reference frames we have introduced (spatial, reference, and rotated). To expedite our development, we first discuss how integral representations of balances can be converted to point wise conservation principles, a process known as localization.

8.1 Localization

Suppose we consider an arbitrary volume of material in the reference configuration, $V \subset \Omega$, of a solid body as depicted in Fig. 8.1. Suppose further that we can establish the following generic integral relation over this volume:

$$\int_V f(\mathbf{X}) \mathrm{d}V = 0, \tag{8.1}$$

where f is some reference function, be it scalar-, vector-, or tensor-valued, defined over all of Ω. If (8.1) holds true for each and every subvolume V of Ω, then the localization theorem states that

$$f = 0 \text{ pointwise in } \Omega. \tag{8.2}$$

The interested reader should consult reference [20] for elaboration on this principle.

It should be noted that the same procedure can be applied spatially. In other words, if we are working with a spatial object, we might consider arbitrary volumes v in the spatial domain, and if the following holds for a spatial object g for all v:

$$\int_v g(\mathbf{x}) \mathrm{d}v = 0, \tag{8.3}$$

then $g(\mathbf{x}) = 0$ throughout $\varphi_t(\Omega)$.

Our primary interest in these localization principles will be to take the well known conservation laws for control volumes and convert them to their local counterparts valid point wise throughout the domain.

8.2 Conservation of Mass

For conservation of mass, we consider a fixed control volume, v, in the spatial domain, completely filled with our solid body at the instant in question as the body moves through it. We can write a conservation of mass for this control volume via

$$-\int_{\partial v} \rho \mathbf{v} \cdot \mathbf{n} \mathrm{d}a = \int_v \frac{\partial \rho}{\partial t} \mathrm{d}v, \tag{8.4}$$
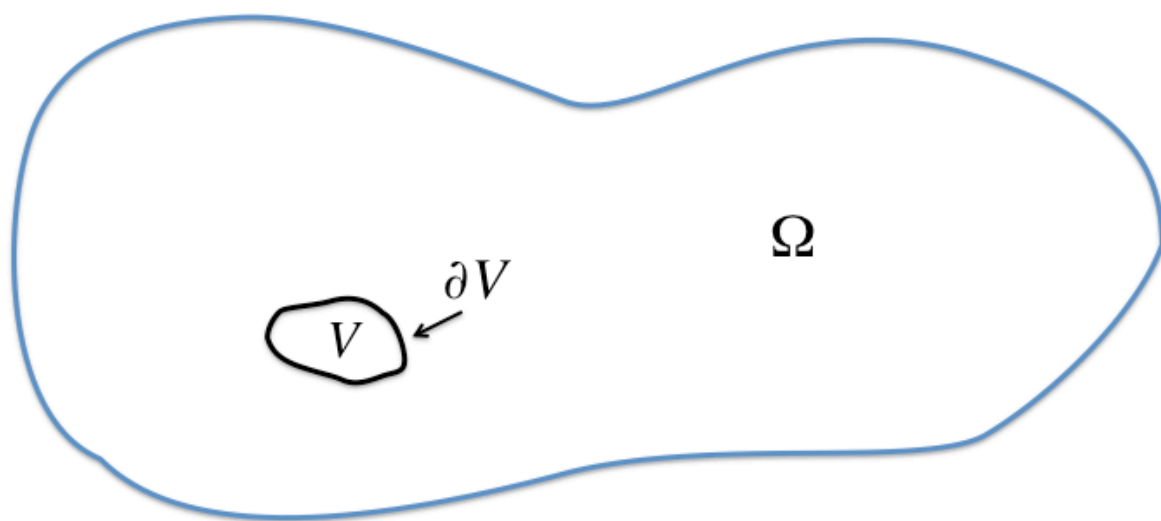
Fig. 8.1 Notation for the localization concept.

where the left-hand side can be interpreted as the net mass influx to the control volume, and the right-hand side is the rate of mass accumulation inside the control volume. Applying the divergence theorem to the left-hand side gives

$$- \int_v \nabla \cdot (\rho \mathbf{v}) \, \mathrm{d}v = \int_v \frac{\partial \rho}{\partial t} \mathrm{d}v. \tag{8.5}$$

This can be further rearranged to yield

$$\int_v \left(\frac{\partial \rho}{\partial t} + \nabla \rho \cdot \mathbf{v} + \rho (\nabla \cdot \mathbf{v}) \right) \mathrm{d}v = 0, \tag{8.6}$$

which can be established for any arbitrary spatial volume v. Applying the localization theorem gives the local expression of continuity, which may be familiar to those versed in fluid mechanics:

$$\frac{\partial \rho}{\partial t} + \nabla \rho \cdot \mathbf{v} + \rho (\nabla \cdot \mathbf{v}) = \dot{\rho} + \rho (\nabla \cdot \mathbf{v}) = 0, \tag{8.7}$$

where the concept of the material time derivative has been employed (cf. (6.5)).

A reference configuration representation of continuity is desirable for the study of solid mechanics. Therefore we convert (8.6) to a reference configuration integral to obtain:

$$\int_{V=\varphi_t^{-1}(v)} (\dot{\rho} + \rho \dot{\mathbf{F}} : \mathbf{F}^{-T}) J \mathrm{d}V = 0, \tag{8.8}$$

where the transformation between $\mathrm{d}v$ and $\mathrm{d}V$ is accomplished using (5.9) and the chain rule is used to convert $\nabla \cdot \mathbf{v}$ via

$$\begin{aligned} v_{i,i}(\mathbf{x}) &= \frac{\partial}{\partial x_i} V_i \left(\varphi_t^{-1}(\mathbf{x}) \right) \\ &= \frac{\partial}{\partial X_I} V_i \left(\varphi_t^{-1}(\mathbf{x}) \right) \frac{\partial X_I}{\partial x_i} \left(\varphi_t^{-1}(\mathbf{x}) \right) \\ &= \dot{F}_{iI} \left(\varphi_t^{-1}(\mathbf{x}) \right) F_{Ii}^{-1} \left(\varphi_t^{-1}(\mathbf{x}) \right), \end{aligned} \tag{8.9}$$

which is the index notation form of $\dot{\mathbf{F}} : \mathbf{F}^{-T}$. Applying the localization theorem in the reference configuration gives

$$\dot{\rho} J + \rho J \dot{\mathbf{F}} : \mathbf{F}^{-T} = 0, \tag{8.10}$$

which holds point wise in Ω.

Working in index notation, we can further simplify (8.10) by concentrating on the term $J\dot{\mathbf{F}} : \mathbf{F}^{-T}$. We compute the material time derivative of J as

$$\dot{J} = \frac{\partial J}{\partial F_{mM}} \dot{F}_{mM}, \tag{8.11}$$

where

$$
\begin{aligned}
\frac{\partial J}{\partial F_{mM}} &= \frac{\partial}{\partial F_{mM}}(e_{ijk}F_{i1}F_{j2}F_{k3}) \\
&= e_{ijk}\delta_{im}\delta_{M1}F_{j2}F_{k3} + e_{ijk}\delta_{jm}\delta_{M2}F_{i1}F_{k3} + e_{ijk}\delta_{km}\delta_{M3}F_{i1}F_{j2} \\
&= e_{ijk}F_{iN}F_{Nm}^{-1}\delta_{M1}F_{j2}F_{k3} + e_{ijk}F_{jN}F_{Nm}^{-1}\delta_{M2}F_{i1}F_{k3} + e_{ijk}F_{kN}F_{Nm}^{-1}\delta_{M3}F_{i1}F_{j2},
\end{aligned} \tag{8.12}
$$

which simplifies to

$$
\begin{aligned}
\frac{\partial J}{\partial F_{mM}} &= JF_{1m}^{-1}\delta_{M1} + JF_{2m}^{-1}\delta_{M2} + JF_{3m}^{-1}\delta_{M3} \\
&= JF_{Im}^{-1}\delta_{MI} = JF_{Mm}^{-1}.
\end{aligned} \tag{8.13}
$$

Substitution of (8.13) into (8.11) gives

$$
\dot{J} = JF_{Mm}^{-1}\dot{F}_{mM}, \tag{8.14}
$$

which is the index notation form of

$$
\dot{J} = J\mathbf{F}^{-1} : \dot{\mathbf{F}}. \tag{8.15}
$$

Finally, substitution of (8.15) into (8.10) gives

$$
\dot{\rho}J + \rho\dot{J} = \frac{\mathrm{d}}{\mathrm{d}t}(\rho J) = 0. \tag{8.16}
$$

(8.16) is the reference configuration version of the continuity equation, which tells us that the product of the density and deformation gradient determinant must be invariant with time for all material points. This is commonly enforced in practice by assigning a reference density ρ_0 to all material points. If the current density ρ is computed via

$$
\rho = \frac{1}{J}\rho_0, \tag{8.17}
$$

then (8.16) is automatically satisfied (recall that the Jacobian is unity in the reference configuration).

8.3 Conservation of Linear Momentum

Considering once more a fixed control volume v, the control volume balance of linear momentum can be expressed as

$$
\int_{\partial v}(\rho\mathbf{v})\mathbf{v}\cdot\mathbf{n}\mathrm{d}a + \int_v \frac{\partial}{\partial t}(\rho\mathbf{v})\mathrm{d}v = \int_v \mathbf{f}\mathrm{d}v + \int_{\partial v}\mathbf{t}\mathrm{d}a. \tag{8.18}
$$

On the left-hand side, the first term expresses the momentum out flux and the second term represents the rate of accumulation inside the control volume. This net change of momentum is

produced by the total resultant force on the system, i.e., the right-hand side of the equation, which is equal to the sum effect of the body forces $\mathbf{f}$ and the surface tractions $\mathbf{t}$.

Applying the divergence theorem to both surface integrals, we find that

$$\int_{\partial v} (\rho \mathbf{v}) \mathbf{v} \cdot \mathbf{n} da = \int_v \left[\nabla \cdot (\rho \mathbf{v}) \mathbf{v} + \rho (\nabla \mathbf{v}) \mathbf{v} \right] dv, \tag{8.19}$$

and

$$\int_{\partial v} \mathbf{t} da = \int_{\partial v} \mathbf{T} \mathbf{n} da = \int_v \nabla \cdot \mathbf{T} dv. \tag{8.20}$$

Substituting (8.19) and (8.20) into (8.18), and rearranging, gives

$$\int_v \left[\nabla \cdot \mathbf{T} + \mathbf{f} - \rho \frac{\partial \mathbf{v}}{\partial t} - \rho (\nabla \mathbf{v}) \mathbf{v} - \frac{\partial \rho}{\partial t} \mathbf{v} - (\nabla \rho \cdot \mathbf{v}) \mathbf{v} - \rho (\nabla \cdot \mathbf{v}) \mathbf{v} \right] dv = 0. \tag{8.21}$$

Employing the spatial form of the continuity (8.6) and recalling the formula for the material time derivative (6.5) gives

$$\int_v \left[\nabla \cdot \mathbf{T} + \mathbf{f} - \rho \dot{\mathbf{v}} \right] dv = 0. \tag{8.22}$$

The localization theorem then implies

$$\nabla \cdot \mathbf{T} + \mathbf{f} = \rho \dot{\mathbf{v}} \tag{8.23}$$

point wise, which is recognized as the same statement of linear momentum balance utilized in our earlier treatment of linear elasticity, (2.2).

In large deformation problems it is desirable to also have a reference configuration form of (8.23). Converting (8.22) to its index form, we have

$$\int_v \left[T_{ij,j} + f_i - \rho \dot{v}_i \right] dv = 0. \tag{8.24}$$

Working with the stress divergence term first, we write

$$T_{ij,j} = \frac{\partial T_{ij}}{\partial X_J} \frac{\partial X_J}{\partial x_j} = \frac{\partial T_{ij}}{\partial X_J} F_{Jj}^{-1}. \tag{8.25}$$

Using (7.11), we can write

$$\frac{\partial T_{ij}}{\partial X_J} = \frac{\partial}{\partial X_j} \left(\frac{1}{J} P_{iI} F_{jI} \right) = \frac{-1}{J^2} \frac{\partial J}{\partial F_{kK}} \frac{\partial F_{kK}}{\partial X_j} P_{iI} F_{jI} + \frac{1}{J} \frac{\partial}{\partial X_J} (P_{iI} F_{jI}). \tag{8.26}$$

Now using (8.13), we can simplify (8.26) and post multiply F_{Jj}^{-1} to obtain:

$$\frac{\partial T_{ij}}{\partial X_J} F_{Jj}^{-1} = \frac{-1}{J} F_{Kk}^{-1} \frac{\partial F_{kK}}{\partial X_J} P_{iJ} + \frac{1}{J} \frac{\partial P_{iI}}{\partial X_I} + \frac{1}{J} F_{Jj}^{-1} \frac{\partial F_{jI}}{\partial X_J} P_{iI} \tag{8.27}$$

The first and last terms on the right-hand side of (8.27) cancel each other due to the fact that $\frac{\partial F_{jI}}{\partial X_J} = \frac{\partial F_{jJ}}{\partial X_I}$. Therefore we have

$$\frac{\partial T_{ij}}{\partial X_J} F_{Jj}^{-1} = \frac{1}{J}\frac{\partial P_{iI}}{\partial X_I}.\tag{8.28}$$

Combining this result with (8.25) and (8.24), and applying a change of variables, gives

$$\int_V (P_{iI,I} + b_i - \rho_0 \dot{V}_i)\mathrm{d}V = 0,\tag{8.29}$$

where $b_i = Jf_i$, the prescribed body force per unit reference volume. Employing the localization theorem gives

$$\nabla_0 \cdot \mathbf{P} + \mathbf{b} = \rho_0 \dot{V}\tag{8.30}$$

point wise in Ω, which expresses the balance of linear momentum in terms of reference coordinates. In (8.30), ∇_0 is the gradient operator with respect to the reference configuration.

8.4 Conservation of Angular Momentum

Again considering an arbitrary control volume in the spatial frame, we write its balance of angular momentum via

$$\int_{\partial v} (\mathbf{x} \times \rho\mathbf{v})\mathbf{v} \cdot \mathbf{n}\mathrm{d}a + \int_v \frac{\partial}{\partial t}(\mathbf{x} \times \rho\mathbf{v})\mathrm{d}v = \int_v (\mathbf{x} \times \mathbf{f})\mathrm{d}v + \int_{\partial v} \mathbf{x} \times \mathbf{t}\mathrm{d}a,\tag{8.31}$$

where the terms on the left-hand side are the out flux and accumulations terms, and the terms on the right-hand side represent the total resultant torque.

Working this time in index notation, we apply the divergence theorem to the surface integrals as follows:

$$\int_{\partial v} e_{ijk}\rho x_j v_k v_l n_l \mathrm{d}a = \int_v \left(\rho_{,l} e_{ijk} x_j v_k v_l + e_{ijk}\rho\delta_{jl}v_k v_l + e_{ijk}\rho x_j v_{k,l}v_l + e_{ijk}\rho x_j v_k v_{l,l}\right)\mathrm{d}v,\tag{8.32}$$

and

$$\int_{\partial v} e_{ijk}x_j T_{kl} n_l \mathrm{d}a = \int_v \left(e_{ijk}x_j T_{kl,l} + e_{ijk}T_{kj}\right)\mathrm{d}v.\tag{8.33}$$

Substituting (8.32) and (8.33) into (8.31), and rearranging terms, reveals that

$$\int_v \Bigg(e_{ijk}x_j \left(T_{kl,l} + f_k - \rho\frac{\partial v_k}{\partial t} - \rho\frac{\partial v_k}{\partial x_l}v_l \right)$$
$$- e_{ijk}x_j v_k \left(\frac{\partial \rho}{\partial t} + \frac{\partial \rho}{\partial x_l}v_l + \rho v_{l,l} \right)\tag{8.34}$$
$$+ e_{ijk}T_{kl} - \rho e_{ijk}\rho v_j v_k \Bigg)\mathrm{d}v = 0.$$

Using (8.24) and (8.7), and noting that the cross product of a vector with itself is zero, we can simplify (8.34) and apply the localization theorem to conclude

$$e_{ijk} T_{kl} = 0,$$ (8.35)

which, in turn, implies the following three equations:

$$T_{23} = T_{32}, \; T_{13} = T_{31}, \; T_{21} = T_{12}.$$ (8.36)

In other words, the symmetry of the Cauchy stress tensor is a direct consequence of the conservation of angular momentum. Use of (7.13) and (7.14), respectively, reveals that the Second Piola-Kirchhoff stress $\mathbf{S}$ and the rotated stress tensor $\boldsymbol{T}$ are likewise symmetric. The First Piola-Kirchhoff stress is not symmetric and is not, in fact, a tensor in the purest sense because it does not fully live in either the spatial or reference frame.

8.5 Stress Power

We examine the consequences of a control volume expression of energy balance. We assume herein a purely mechanical description and, to begin, that there is no mechanical dissipation, so that the system we consider conserves energy exactly. In other words, all work put into the system through the applied loads goes either into stored internal elastic energy or into kinetic energy.

With this in mind, the conservation of energy for a spatial control volume is written as

$$\int_{\partial v} \left(e + \frac{1}{2}\rho \mathbf{v} \cdot \mathbf{v} \right) \mathbf{v} \cdot \mathbf{n} \mathrm{d}a + \int_v \frac{\partial}{\partial t} \left(e + \frac{1}{2}\rho \mathbf{v} \cdot \mathbf{v} \right) \mathrm{d}v = \int_v \mathbf{f} \cdot \mathbf{v} \mathrm{d}v + \int_{\partial v} \mathbf{t} \cdot \mathbf{v} \mathrm{d}a,$$ (8.37)

where e is the internal stored energy (i.e., elastic energy) per unit spatial volume.

As we have done previously, we apply the divergence theorem to the surface integrals:

$$\int_{\partial v} \left(e + \frac{1}{2}\rho \mathbf{v} \cdot \mathbf{v} \right) \mathbf{v} \cdot \mathbf{n} \mathrm{d}a = \int_v \Bigg[\nabla \cdot \mathbf{v} \left(e + \frac{1}{2}\rho \mathbf{v} \cdot \mathbf{v} \right) + \nabla e \cdot \mathbf{v}$$
$$+ \frac{1}{2}\nabla \rho \cdot \mathbf{v}(\mathbf{v} \cdot \mathbf{v}) + \rho \mathbf{v} \cdot (\nabla \mathbf{v})\mathbf{v} \Bigg] \mathrm{d}v,$$ (8.38)

and

$$\int_{\partial v} \mathbf{t} \cdot \mathbf{v} \mathrm{d}a = \int_v [\mathbf{T} : \nabla \mathbf{v} + (\nabla \cdot \mathbf{T}) \cdot \mathbf{v}] \, \mathrm{d}v.$$ (8.39)

Substituting (8.38) and (8.39) into (8.37), and rearranging, gives

$$0 = \int_v \Bigg[\left(\nabla \cdot \mathbf{T} + \mathbf{f} - \rho \frac{\partial \mathbf{v}}{\partial t} - (\rho \, \nabla \mathbf{v}) \, \mathbf{v} \right) \cdot \mathbf{v}$$
$$- \frac{1}{2}\mathbf{v} \cdot \mathbf{v} \left(\frac{\partial \rho}{\partial t} + (\nabla \cdot \mathbf{v})\rho + \nabla \rho \cdot \mathbf{v} \right)$$ (8.40)
$$+ \mathbf{T} : \nabla \mathbf{v} - (\nabla \cdot \mathbf{v})e - \dot{e} \Bigg] \, \mathrm{d}v.$$

Using (8.24) and (8.7), we find

$$0 = \int_v [\mathbf{T} : \nabla \mathbf{v} - (\nabla \cdot \mathbf{v})e - \dot{e}] \, dv. \tag{8.41}$$

Splitting (8.41) into two integrals, we have

$$0 = \int_v \mathbf{T} : \nabla \mathbf{v} dv - \int_v ((\nabla \cdot \mathbf{v})e + \dot{e}) \, dv. \tag{8.42}$$

We now convert (8.42) to the reference configuration and apply localizations. In so doing, we recognize that the second integral in (8.42) can be treated directly analogous to that of (8.6), with the density ρ in (8.6) replaced by the energy e in the current case. The result of this manipulation will be analogous to (8.16) with e substituted for ρ. In other words, we have

$$\int_v ((\nabla \cdot \mathbf{v})e + \dot{e}) \, dv = \int_V \frac{d}{dt}(eJ) dV. \tag{8.43}$$

Concentrating on the first integral and using (6.12) and (8.11) to aid in the calculation, we find

$$\int_v \mathbf{T} : \nabla \mathbf{v} dv = \int_V (\mathbf{T} \circ \varphi^{-1}) : (\mathbf{L} \circ \varphi^{-1}) J dV$$

$$= \int_V (\mathbf{T} \circ \varphi^{-1}) : (\dot{\mathbf{F}} \mathbf{F}^{-1}) J dV = \int_V \mathbf{P} : \dot{\mathbf{F}} dV. \tag{8.44}$$

Plugging the results of (8.43) and (8.44) into (8.42) and employing the localization theorem, we determine that

$$\frac{d}{dt}(eJ) = \dot{E} = \mathbf{P} : \dot{\mathbf{F}} \tag{8.45}$$

point wise in Ω, where E is the stored elastic energy per unit reference volume. Therefore, $\mathbf{P} : \dot{\mathbf{F}}$ represents the rate of energy input into the material by the stress (per unit volume), commonly known as the **stress power**. Taking into account the various measures of stress and deformation rate we have considered, it can be shown that for a given material point, the stress power can be written in the following alternative forms:

$$\textbf{stress power} = \mathbf{P} : \dot{\mathbf{F}} = \frac{1}{2}\mathbf{S}\dot{\mathbf{C}} = J\mathbf{T} : \mathbf{D} = J\boldsymbol{T} : \boldsymbol{D}. \tag{8.46}$$

It should be noted that this definition can be used also for dissipative (i.e., non-conservative) materials but the interpretation changes. The stress power in that case is the sum of the rate of increase of stored energy and the rate of energy dissipated by the solid.

8.6 Thermodynamics

Finally we discuss the application of the laws of thermodynamics to large deformation Lagrangian mechanics. Recalling the notation from Section 4, we consider the first and second laws applied to a body Ω in the reference (i.e., material) configuration.

First Law

The first law states that the change in internal energy, change in kinetic energy, external power, and heat flux over the body must be balanced:

$$\dot{\mathcal{E}} + \dot{\mathcal{K}} = \mathcal{W} + Q$$

where

- $\mathcal{E}$, the total internal energy, is given by

$$\mathcal{E} = \int_{\Omega} \rho_0 w \, dV,$$

  where w is the specific internal energy (both elastic and dissipated),

- $\mathcal{K}$, the kinetic energy, is given by

$$\mathcal{K} = \int_{\Omega} \frac{1}{2} \rho_0 \, |\mathbf{v}|^2 \, dV,$$

- $\mathcal{W}$, the conventional external power, is given by

$$\mathcal{W} = \int_{\partial\Omega} \mathbf{Tn} \cdot \mathbf{v} \, d\Gamma + \int_{\Omega} \mathbf{b} \, dV,$$

- Q, the heat flux, is given by

$$Q = -\int_{\partial\Omega} \mathbf{q} \cdot \mathbf{n} \, d\Gamma + \int_{\Omega} q \, dV,$$

  where q is the scalar heat supply, and $\mathbf{q}$ is the heat flux vector.

The corresponding local energy balance can be readily derived by applying the divergence theorem and power balance on sub-regions:

$$\rho_0 \dot{w} = \mathbf{T} : \mathbf{D} - \nabla_0 \cdot \mathbf{q} + q, \tag{8.47}$$

where $\nabla_0 \cdot$ is the divergence operator in the reference configuration.

Second Law

The second law of thermodynamics states that the entropy of an isolated system can not decrease. The global inequality over Ω corresponding to this statement is

$$\int_{\Omega} \rho_0 \dot{\eta} \, dV \geq -\int_{\partial\Omega} \frac{\mathbf{q}}{\theta} \, d\Gamma + \int_{\Omega} \frac{q}{\theta} \, dV,$$

where η is the specific entropy, θ is the absolute temperature, $\frac{\mathbf{q}}{\theta}$ is the entropy flux, and $\frac{q}{\theta}$ is the entropy supply. The corresponding local entropy imbalance can be derived using the divergence theorem and localization to sub-regions:

$$\rho_0 \dot{\eta} \geq -\nabla_0 \cdot \left(\frac{\mathbf{q}}{\theta}\right) + \frac{q}{\theta}.$$

Using (8.47) and some manipulation this can be rewritten

$$\mathbf{T} : \mathbf{D} - \rho_0 \left(\dot{e} + \eta \dot{\theta} \right) - \frac{1}{\theta} \mathbf{q} \cdot \nabla_0 \theta \geq 0, \tag{8.48}$$

where e, the specific free-energy (i.e., the elastic energy), is given by

$$e = w - \theta \eta.$$

In the absence of thermal effects, (8.48) reduces to

$$\mathbf{T} : \mathbf{D} \geq \rho_0 \, \dot{e},$$

which states that the internal power expenditure (stress power) must exceed the rate of increase of stored energy. More simply put, the dissipated power in the body must always be positive.

The references for chapter 8 are [15, 20, 21, 39].

9 Frame Indifference

An important concept in the formulation of constitutive theories in large deformations is frame indifference, alternatively referred to as objectivity. Although somewhat mathematically involved, the concept of objectivity is fairly simple to understand physically.

When we write constitutive laws in their most general form, we seek to express tensorial quantities, such as stress and stress rate, in terms of kinematic tensorial quantities, most commonly strain and strain rate. The basic physical idea behind frame indifference is that this constitutive relationship should be unaffected by any rigid body motions of the material. Mathematically, we evaluate frame indifference by defining an alternative reference frame that is rotating and translating with respect to the coordinate system in which we pose the problem. For our constitutive description to make sense, the tensorial quantities we use (stress, stress rate, strain, and strain rate) should transform according to the laws of tensor calculus when subjected to a change in reference frame. If a given quantity does this, we say it is material frame indifferent, and if it does not, we say it is not properly invariant.

9.1 Objective Strain and Strain Rate Measures

Consider a motion, $\varphi(\mathbf{X}, t)$. We imagine ourselves to be viewing this motion from another reference frame, denoted in the following by $*$, which is related to the original spatial frame via

$$\mathbf{x}^* = \mathbf{c}(t) + \mathbf{Q}(t)\mathbf{x}, \tag{9.1}$$

where $\mathbf{x} = \varphi(\mathbf{X}, t)$. In (9.1), $\mathbf{c}(t)$ and $\mathbf{Q}(t)$ are rigid body translation and rotation, respectively, between the original frame and observer $*$. To observer $*$, the motion appears as defined by

$$\mathbf{x}^* = \varphi^*(\mathbf{X}, t) = \mathbf{c}(t) + \mathbf{Q}(t)\varphi(\mathbf{X}, t). \tag{9.2}$$

The time derivative of this motion equation gives the relationship between the deformation gradients in the two frames:

$$\mathbf{F}^* = \frac{\partial}{\partial \mathbf{X}}\varphi_t^* = \mathbf{Q}\frac{\partial}{\partial \mathbf{X}}\varphi_t(\mathbf{X}) = \mathbf{QF}. \tag{9.3}$$

The spatial velocity gradient $\mathbf{L}^*$ is then

$$\mathbf{L}^* = \nabla^*\mathbf{v}^* = \dot{\mathbf{F}}^*(\mathbf{F}^*)^{-1} = \frac{d}{dt}(\mathbf{QF})(\mathbf{QF})^{-1} = \left(\mathbf{Q}\dot{\mathbf{F}}\mathbf{F}^{-1}\mathbf{Q}^T + \dot{\mathbf{Q}}\mathbf{F}\mathbf{F}^{-1}\mathbf{Q}^T\right), \tag{9.4}$$

which simplifies to

$$\mathbf{L}^* = \mathbf{QLQ}^T + \dot{\mathbf{Q}}\mathbf{Q}^T. \tag{9.5}$$

For $\mathbf{L} = \nabla\mathbf{v}$ to be objective, it would transform according to the laws of tensor transformation between the two frames, i.e., only the first term on the right-hand side of (9.5) would be present. Clearly, $\mathbf{L} = \nabla\mathbf{v}$ is *not* objective.

Examining the rate of deformation tensor $\mathbf{D}^*$, on the other hand, one finds:

$$\mathbf{D}^* = \frac{1}{2}\left(\mathbf{L}^* + (\mathbf{L}^*)^T\right) = \frac{1}{2}\left[\mathbf{Q}\mathbf{L}\mathbf{Q}^T + \dot{\mathbf{Q}}\mathbf{Q}^T + \mathbf{Q}\mathbf{L}^T\mathbf{Q}^T + \mathbf{Q}\dot{\mathbf{Q}}^T\right],\tag{9.6}$$

where

$$\dot{\mathbf{Q}}\mathbf{Q}^T + \mathbf{Q}\dot{\mathbf{Q}}^T = \frac{d}{dt}\left[\mathbf{Q}\mathbf{Q}^T\right] = \frac{d}{dt}\left[\mathbf{I}\right] = 0.\tag{9.7}$$

Hence, (9.6) simplifies to

$$\mathbf{D}^* = \frac{1}{2}\mathbf{Q}\left[\mathbf{L} + \mathbf{L}^T\right]\mathbf{Q}^T = \mathbf{Q}\mathbf{D}\mathbf{Q}^T,\tag{9.8}$$

which shows us that $\mathbf{D}$ is objective.

Therefore we have a tensorial quantity for the spatial rate-of-strain that is objective. The question arises whether corresponding reference measures of rate are objective. It turns out that such material rates are automatically objective, since they do not change when superimposed rotations occur spatially. Consider, for example, the right Cauchy-Green tensor $\mathbf{C}$:

$$\mathbf{C}^* = (\mathbf{F}^*)^T(\mathbf{F}^*) = \mathbf{F}^T\mathbf{Q}^T\mathbf{Q}\mathbf{F} = \mathbf{C}.\tag{9.9}$$

Similarly, the time derivative of (9.9) simplifies to

$$\dot{\mathbf{C}}^* = \dot{\mathbf{C}}.\tag{9.10}$$

9.2 Stress Rates

Turning our attention to stress rates, examine the material time derivative of the Cauchy stress $\mathbf{T}$:

$$\dot{\mathbf{T}} = \left[\frac{d}{dt}(\mathbf{T}\circ\varphi_t)\right]\bullet\varphi_t^{-1} = \left(\frac{\partial\mathbf{T}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{T}\right).\tag{9.11}$$

$\mathbf{T}$ is itself objective by its very definition as a tensorial quantity. Thus, we can write

$$\mathbf{T}^* = \mathbf{Q}\mathbf{T}\mathbf{Q}^T.\tag{9.12}$$

Computing the material time derivative of (9.12) gives

$$\dot{\mathbf{T}}^* = \dot{\mathbf{Q}}\mathbf{T}\mathbf{Q}^T + \mathbf{Q}\dot{\mathbf{T}}\mathbf{Q}^T + \mathbf{Q}\mathbf{T}\dot{\mathbf{Q}}^T.\tag{9.13}$$

Since the first and third terms on the right-hand side of (9.13) do not, in general, cancel, we see that the material time derivative of the Cauchy stress $\mathbf{T}$ is *not* objective.

It therefore becomes critical to consider a frame indifferent measure of stress rate when formulating a constitutive description that requires a stress rate. A multitude of such rates have

been contrived; the interested reader is encouraged to consult Reference [39] for a highly theoretical treatment. For our discussion here, we consider two such rates especially prevalent in the literature: the Jaumann rate and the Green-Naghdi rate. Both rates rely on roughly the same physical idea. Rather than taking the derivative of the Cauchy stress itself, we rotate the object from the spatial frame before taking the time derivative, so that the reference frame in which the time derivative is taken is the same for all frames related by the transformation in (9.1).

For example, we consider the Jaumann rate of stress, which we denote here as $\hat{\mathbf{T}}$. Its definition is given as

$$\hat{\mathbf{T}} = \dot{\mathbf{T}} - \mathbf{WT} + \mathbf{TW}, \tag{9.14}$$

where $\mathbf{W} = \mathbf{L} - \mathbf{D}$. We can verify that this rate of stress is truly objective by considering the object as it would appear to observer $*$:

$$\hat{\mathbf{T}}^* = \dot{\mathbf{T}}^* - \mathbf{W}^*\mathbf{T}^* + \mathbf{T}^*\mathbf{W}^*. \tag{9.15}$$

The quantity $\dot{\mathbf{T}}^*$ is given by (9.13), $\mathbf{T}^*$ is given by (9.12), and $\mathbf{W}^*$ $is computed with the aid of : eq : 'frame_ndifference : eq : 05$ and (9.8):

$$\mathbf{W}^* = \mathbf{L}^* - \mathbf{D}^* = \mathbf{QLQ}^T + \dot{\mathbf{Q}}\mathbf{Q}^T - \mathbf{QDQ}^T. \tag{9.16}$$

Substituting these quantities into (9.15), we find

$$\begin{aligned} \hat{\mathbf{T}}^* = {} & \dot{\mathbf{Q}}\mathbf{T}\mathbf{Q}^T + \mathbf{Q}\dot{\mathbf{T}}\mathbf{Q}^T - \mathbf{Q}\mathbf{T}\dot{\mathbf{Q}}^T \\ & - \left(\mathbf{QLQ}^T + \dot{\mathbf{Q}}\mathbf{Q}^T - \mathbf{QDQ}^T\right)\mathbf{Q}\mathbf{T}\mathbf{Q}^T \\ & + \mathbf{Q}\mathbf{T}\mathbf{Q}^T\left(\mathbf{QLQ}^T + \dot{\mathbf{Q}}\mathbf{Q}^T - \mathbf{QDQ}^T\right). \end{aligned} \tag{9.17}$$

Canceling terms and using the fact that $\dot{\mathbf{Q}}\mathbf{Q}^T = -\mathbf{Q}\dot{\mathbf{Q}}^T$, we can simplify (9.17) to

$$\hat{\mathbf{T}}^* = \mathbf{Q}\left[\dot{\mathbf{T}} - \mathbf{WT} + \mathbf{TW}\right]\mathbf{Q}^T = \mathbf{Q}\hat{\mathbf{T}}\mathbf{Q}^T, \tag{9.18}$$

which ensures us that, indeed, $\hat{\mathbf{T}}$ is objective.

By considering the Green-Naghdi rate we can gain more insight into how objective rates are defined. The Green-Naghdi rate of Cauchy stress is defined via

$$\tilde{\mathbf{T}} = \mathbf{R}\dot{\mathbf{T}}\mathbf{R}^T, \tag{9.19}$$

where $\mathbf{R}$ is the rotation tensor from the polar decomposition of $\mathbf{F}$, and $\mathbf{T}$ is the rotated Cauchy stress defined in (7.14).

We examine how the rotation tensor $\mathbf{R}$ transforms. Utilizing (9.3) and the polar decomposition, we get

$$\mathbf{F}^* = \mathbf{R}^*\mathbf{U}^* = \mathbf{QF} = \mathbf{QRU}. \tag{9.20}$$

We now note two things: first, the product **QR** is itself a proper orthogonal tensor; and second, the polar decomposition is unique for a given deformation gradient. Therefore, comparing the second and fourth terms of (9.20), we must conclude

$$\mathbf{U}^* = \mathbf{U}, \tag{9.21}$$

and

$$\mathbf{R}^* = \mathbf{QR}. \tag{9.22}$$

Using (9.22) and (9.19), we can compute

$$\tilde{\mathbf{T}}^* = \mathbf{R}^* \dot{T}^* \mathbf{R}^{*T} = \mathbf{QR} \dot{T}^* \mathbf{R}^T \mathbf{Q}^T. \tag{9.23}$$

Returning to the definition of T in (7.14) and incorporating (9.12) and (9.22), we can write

$$T^* = \mathbf{R}^{*T} \mathbf{T}^* \mathbf{R}^* = \mathbf{R}^T \mathbf{Q}^T (\mathbf{QTQ}^T) \mathbf{QR} = \mathbf{R}^T \mathbf{TR} = T. \tag{9.24}$$

Therefore, the rotated stress tensor appears exactly the same in both frames of reference. It follows that

$$\dot{T}^* = \dot{T}, \tag{9.25}$$

which, when substituted into (9.23), gives

$$\tilde{\mathbf{T}}^* = \mathbf{QR} \dot{T} \mathbf{R}^T \mathbf{Q}^T = \mathbf{Q} \tilde{\mathbf{T}} \mathbf{Q}^T. \tag{9.26}$$

This is recognized as the properly objective transformation of $\tilde{\mathbf{T}}$.

One may note that this result gives considerable insight into how objective rates can be constructed. In the current case, we transform the stress into the rotated configuration before computing its time derivative, and then we transform the result back to the spatial configuration. Since the rotated stress is exactly the same for all reference frames, related by (7.1), taking the time derivative of it and then transforming produces an objective tensor. This idea can be generalized as follows: construction of an objective rate of stress is achieved by considering the time derivative of a stress measure defined in a coordinate system that is rotating about some set of axes. In fact, one can show that the Jaumann stress rate can be similarly interpreted.

Finally, the Green-Naghdi rate can be manipulated further to a form more closely resembling the form for the Jaumann rate ((9.14)). That is, we can write

$$\begin{aligned} \tilde{\mathbf{T}} &= \mathbf{R} \frac{d}{dt} (\mathbf{R}^T \mathbf{TR}) \mathbf{R}^T \\ &= \mathbf{R} \dot{\mathbf{R}}^T \mathbf{T} + \dot{\mathbf{T}} + \mathbf{T} \dot{\mathbf{R}} \mathbf{R}^T \\ &= \dot{\mathbf{T}} + L^T \mathbf{T} + \mathbf{T} L \\ &= \dot{\mathbf{T}} + \mathbf{T} L - L \mathbf{T}, \end{aligned} \tag{9.27}$$

where (6.19) is used to define L, recalling also that this object is skew.

The reference for Chapter 9 is [39]

10 Discretization

10.1 Weak Form for Large Deformation Problems

We begin by reviewing the field equations to be considered. The reference for this chapter is [39]. The problem to be solved is shown schematically in Fig. 10.1, in which we want to the compute finite deformation response of a body Ω in its reference configuration.
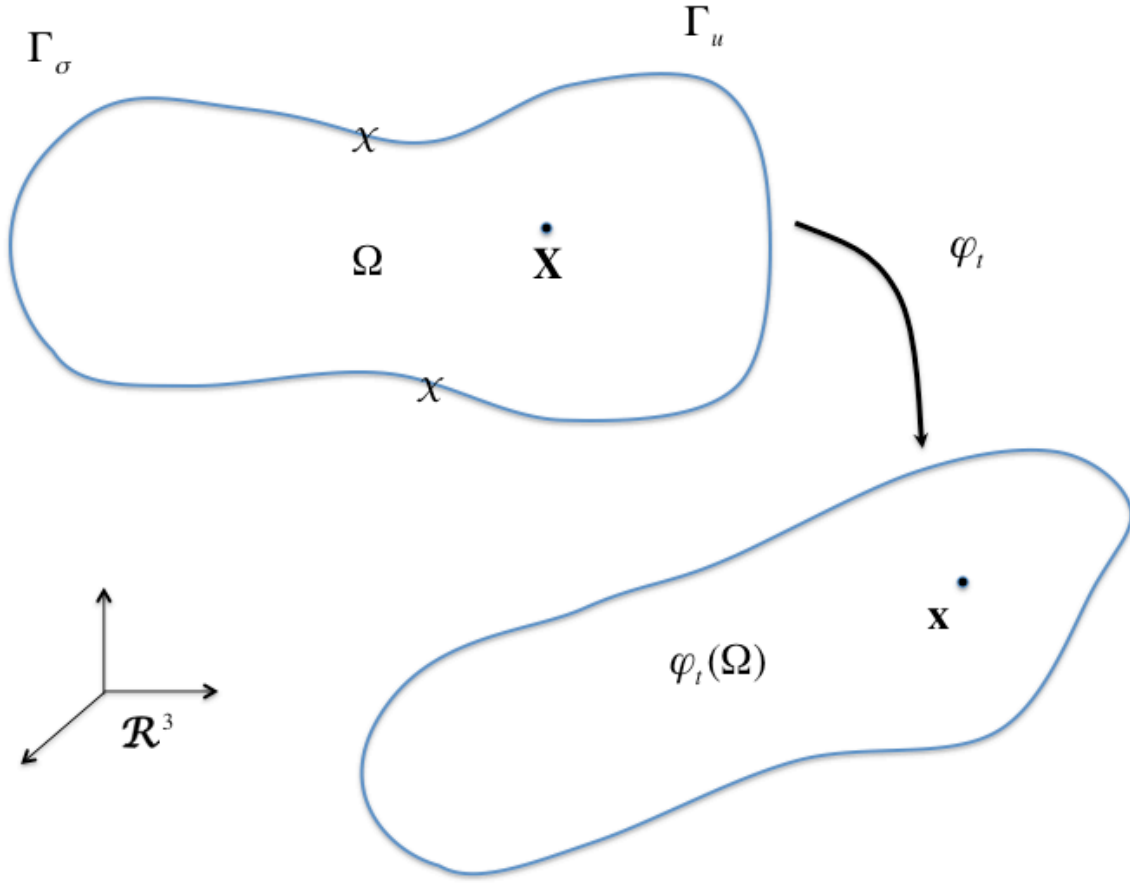
Fig. 10.1 Large deformation initial/boundary value problem

Assuming that this time dependent configuration mapping is denoted by φ_t, the following problem is solved for each time, t, in the time interval of interest:

$$\nabla \cdot \mathbf{T} + \mathbf{f} = \rho \mathbf{a} \text{ on } \varphi_t(\Omega), \tag{10.1}$$

$$\varphi_t = \bar{\varphi}_t \text{ on } \varphi_t(\Gamma_u), \tag{10.2}$$

and

$$\mathbf{t} = \bar{\mathbf{t}} \text{ on } \varphi_t(\Gamma_\sigma), \tag{10.3}$$

where all notations are as discussed in Section 4. 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 $\mathbf{t} = \mathbf{Tn}$, where $\mathbf{n}$ is the outward unit normal to the spatial surface $\varphi_t(\Gamma_\sigma)$.

The problem is also subject to initial conditions of the form

$$\varphi(\mathbf{X}, 0) = \varphi_0(\mathbf{X}) \text{ on } \Omega, \tag{10.4}$$

and

$$\frac{\partial \varphi}{\partial t}(\mathbf{X}, 0) = \mathbf{X}_0(\mathbf{X}) \text{ on } \Omega. \tag{10.5}$$

Recall that (10.1) through (10.3) are written in the so-called spatial configuration, but we still consider ourselves working in a Lagrangian framework where all quantities are ultimately indexed to material points through the mapping $\mathbf{x} = \varphi_t(\mathbf{X})$ (see Lagrangian and Eulerian Descriptions in Section 4).

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 Section 3 by considering weighting functions φ^* defined over Ω which satisfy the following condition:

$$\varphi^* = \mathbf{0} \text{ on } \Gamma_u, \tag{10.6}$$

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 by $\mathbf{w}$ and note that they may be obtained via

$$\mathbf{w}(\mathbf{x}) = \varphi^* \left(\varphi_t^{-1}(\mathbf{x}) \right) \tag{10.7}$$

for any $\mathbf{x} \in \varphi_t(\mathbf{X})$. (10.6) means

$$\mathbf{w} = \mathbf{0} \text{ on } \varphi_t(\Gamma_u). \tag{10.8}$$

Assuming the configuration mapping φ_t is smooth, all required partial derivatives of $\mathbf{w}$ can be computed.

With these definitions, the development in Section 3 can be reproduced in the current context to provide the following spatial representation of the *variational form for large deformations*:

Given the boundary conditions $\bar{\mathbf{t}}$ 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 for $\mathbf{f}$ on $\varphi_t(\Omega)$, find $\varphi_t \in S_t$ for each time $t \in (0, T)$ such that:

$$\int_{\varphi_t(\Omega)} \rho \mathbf{w} \cdot \mathbf{a} dv + \int_{\varphi_t(\Omega)} \nabla \mathbf{w} : \mathbf{T} dv = \int_{\varphi_t(\Omega)} \mathbf{w} \cdot \mathbf{f} dv + \int_{\varphi_t(\Gamma_\sigma)} \mathbf{w} \cdot \bar{\mathbf{t}} da \tag{10.9}$$

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

$$S_t = \{\varphi_t | \varphi_t = \bar{\varphi}(t) \text{ on } \Gamma_u , \varphi_t \text{ is smooth}\} \tag{10.10}$$

and where admissible $\mathbf{w}$ are related in a one-to-one manner via (10.7) to the material variations $\varphi^* \in W$ with the definition of W being

$$W = \{\varphi^* | \varphi^* = \mathbf{0} \text{ on } \Gamma_u , \varphi^* \text{ is smooth}\} . \tag{10.11}$$

Note that in contrast to the previous development, the constitutive relation governing $\mathbf{T}$ is left unspecified, but it can in general be subject to both geometric and material nonlinearities. Furthermore, it should be implied that geometric nonlinearities include consideration of large deformation kinematics discussed in Section 5, Section 6, and Section 9. The notation $\mathbf{a}$ for the acceleration is to be understood as the material acceleration as defined by (6.4).

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

$$\int_\Omega \varphi^* \cdot (\varphi|_{t=0} - \varphi_0) \mathrm{d}\Omega = 0 \tag{10.12}$$

and

$$\int_\Omega \varphi^* \cdot \left(\frac{\partial \varphi}{\partial t} \bigg|_{t=0} - \mathbf{V}_0 \right) \mathrm{d}\Omega = 0, \tag{10.13}$$

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

10.2 Finite Element Discretization

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). The variational form to be considered here is that just summarized in the previous section. We now refer to Fig. 10.2 which gives the general notation to be used in the description of the discretization process.

Referring to Fig. 10.2, the reference domain Ω is subdivided into a number of element subdomains Ω^e. The superscript e is an index to a specific element, running between 1 and the total number of elements in the discretization, n_{el}, of the domain Ω. We assume in the figure and throughout the ensuing discussion that Ω is a subset of $\mathbb{R}^3$.

Note that a number of nodal points are indicated by the dots in Fig. 10.2. We assume that all degrees of freedom in the discrete system to be proposed will be associated with these nodes. 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, etc. having values running between 1 and n_{np}, the total number of nodal points in the problem discretization.
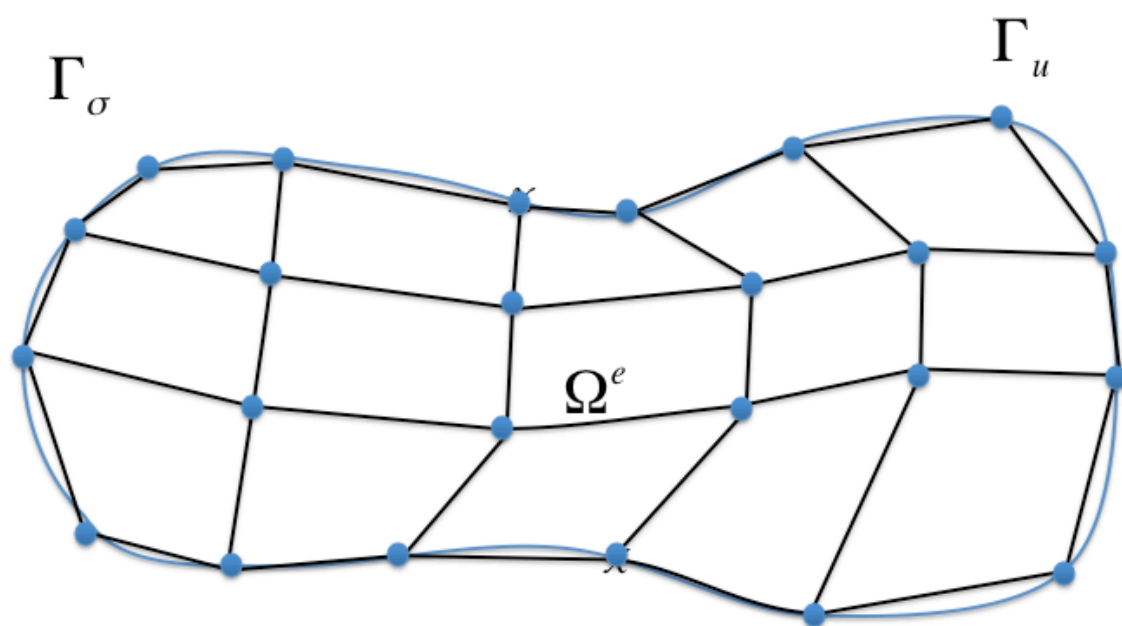
Fig. 10.2 General notation for finite element discretization of the reference domain.

10.3 Galerkin Finite Element Methods

The essence of any finite element method lies in the discretization of the variational form. This discretization process involves approximation of a typical member of both the solution space S_t and 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 the space.

Roughly speaking, the idea of discretization is as follows. We know from earlier chapters that if the variational equation is enforced considering all $\varphi_t \in S_t$ 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 φ^* and the very general definitions for S_t and W. If we restrict our attention only to some subset of the above spaces, we inherently incur some error with the solution of our approximated weak form in that it no longer is 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 arbitrary 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).

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

$$N_A : \bar{\Omega} \to \mathbb{R}. \tag{10.14}$$

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

$$\varphi_t^h = \sum_{B=1}^{n_{np}} N_B \mathbf{d}_B(t), \tag{10.15}$$

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

$$S_t^h = \left\{ \varphi_t^h = \sum_{B=1}^{n_{np}} N_B \mathbf{d}_B(t) \middle| \varphi_t^h \approx \tilde{\varphi}_t(\mathbf{X} \text{ for all } \mathbf{X} \in \Gamma_u \right\}. \tag{10.16}$$

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_t^h to interpolate the nodal values of $\bar{\varphi}_t$ on Γ_u with the N_B serving as the interpolation functions. Note that Γ_u itself is typically geometrically approximated by the finite element discretization, also contributing to the approximation.

This notationally defines the discretization procedure for φ_t^h. 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_t. Accordingly, we define a member of this space, $(\varphi^*)^h$, via

$$(\varphi^*)^h = \sum_{A=1}^{n_{np}} N_A \mathbf{c}_A, \tag{10.17}$$

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

$$W^h = \left\{ (\varphi^*)^h = \sum_{A=1}^{n_{np}} N_A \mathbf{c}_A \middle| (\varphi^*)^h = 0 \text{ for all } \mathbf{X} \in \Gamma_u \right\}. \tag{10.18}$$

Analogous to the situation for S_t^h, (10.18) features a discrete version of the boundary condition on Γ_u. In other words, W^h consists of all functions of the form (10.17) resulting in satisfaction of this condition. Note that the only restriction on $\mathbf{c}_A$ is that they result in satisfaction of the homogeneous boundary condition on Γ_u.

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

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

$$\int_{\varphi_t^h(\Omega)} \rho \mathbf{w}^h \cdot \mathbf{a}^h \mathrm{d}v + \int_{\varphi_t^h(\Omega)} \nabla \mathbf{w}^h : \mathbf{T}^h \mathrm{d}v = \int_{\varphi_t^h(\Omega)} \mathbf{w}^h \cdot \mathbf{f} \mathrm{d}v + \int_{\varphi_t^h(\Gamma_\sigma)} \mathbf{w}^h \cdot \bar{\mathbf{t}} \mathrm{d}a \tag{10.19}$$

for all admissible $\mathbf{w}^h$, where S_t is defined in (10.16) and where admissible $\mathbf{w}^h$ are related to the material variations $(\varphi^*)^h \in W^h$ via

$$\mathbf{w}^h(\mathbf{x}) = (\varphi^*)^h \in \left(\varphi_t^h \right)^{-1} (\mathbf{x}). \tag{10.20}$$

In (10.19), $\mathbf{T}^h$ 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 read

$$\mathbf{d}_B(0) = \bar{\varphi}_0(\mathbf{X}_B) \tag{10.21}$$

and

$$\dot{\mathbf{d}}_B(0) = \mathbf{V}_0(\mathbf{X}_B), \tag{10.22}$$

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

section{Notation for Discrete Problem}

In preparation for generating vector/matrix equations for the discrete system, it will be helpful to be explicit with our notation. We therefore express the nodal vectors $\mathbf{c}_A$ and $\mathbf{d}_B$ in terms of their components via

$$\mathbf{c}_A = \{c_{iA}\}, \; i = 1, 2, 3 \tag{10.23}$$

and

$$\mathbf{d}_B = \{d_{jB}\}, \; j = 1, 2, 3. \tag{10.24}$$

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. Thus, we define for notational convenience the concept of an ID array that is set up as follows:

$$ID(i, A) = P \text{ (global degree of freedom number).} \tag{10.25}$$

In other words, the ID array takes the spatial direction index and nodal point number as arguments and assigns a global degree of freedom number to the corresponding unknown. For three-dimensional deformation problems, the number of degrees of freedom n_{dof} is

$$n_{dof} = 3 \times n_{np}. \tag{10.26}$$

With this notation, the equation numbers P and Q corresponding to the degrees of freedom are defined as

$$P = ID(i, A) \tag{10.27}$$

and

$$Q = ID(j, B). \tag{10.28}$$

10.4 Discrete Equations

We now generate the discrete equations by substitution of (10.15) and (10.17) into (10.19), causing the variational equation to read

$$\int_{\varphi_t^h(\Omega)} \rho \left(\sum_{A=1}^{n_{np}} N_A \left(\varphi_t^{-1}(\mathbf{x}) \right) c_A \right) \cdot \left(\sum_{B=1}^{n_{np}} N_B \left(\varphi_t^{-1}(\mathbf{x}) \right) \ddot{\mathbf{d}}_B(t) \right) dv$$

$$+ \int_{\varphi_t^h(\Omega)} \left(\sum_{A=1}^{n_{np}} \nabla N_A \left(\varphi_t^{-1}(\mathbf{x}) \right) \otimes c_A \right) : \mathbf{T}^h dv \tag{10.29}$$

$$+ \int_{\varphi_t^h(\Omega)} \left(\sum_{A=1}^{n_{np}} N_A \left(\varphi_t^{-1}(\mathbf{x}) \right) c_A \right) \cdot \mathbf{f} dv + \int_{\varphi_t^h(\Gamma_\sigma)} \left(\sum_{A=1}^{n_{np}} N_A \left(\varphi_t^{-1}(\mathbf{x}) \right) c_A \right) \cdot \bar{\mathbf{t}} da$$

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

The inertial term in (10.29) can be expanded as

$$
\int_{\varphi^h_t(\Omega)} \rho \left(\sum_{A=1}^{n_{np}} N_A \left(\varphi_t^{-1}(\mathbf{x}) \right) c_A \right) \cdot \left(\sum_{B=1}^{n_{np}} N_B \left(\varphi_t^{-1}(\mathbf{x}) \right) \ddot{\mathbf{d}}_B(t) \right) \mathrm{d}v
$$

$$
= \sum_{A=1}^{n_{np}} \sum_{i=1}^{3} c_{iA} \int_{\varphi^h_t(\Omega)} \left(\rho N_A \left(\varphi_t^{-1}(\mathbf{x}) \right) c_{iA} \left(\sum_{B=1}^{n_{np}} N_B \left(\varphi_t^{-1}(\mathbf{x}) \right) \ddot{d}_{iB} \right) \mathrm{d}v \right)
$$

$$
= \sum_{A=1}^{n_{np}} \sum_{i=1}^{3} c_{iA} \left[\sum_{B=1}^{n_{np}} \sum_{i=1}^{3} \int_{\varphi^h_t(\Omega)} \rho N_A \left(\varphi_t^{-1}(\mathbf{x}) \right) \delta_{ij} N_B \left(\varphi_t^{-1}(\mathbf{x}) \right) \mathrm{d}v \, \ddot{d}_{iB} \right]
$$

$$
= \sum_{P=1}^{n_{dof}} c_P \left(\sum_{Q=1}^{n_{dof}} M_{PQ} \ddot{d}_Q \right)
$$

(10.30)

where M_{PQ} is defined as

$$
M_{PQ} = \int_{\varphi^h_t(\Omega)} \rho N_A \left(\varphi_t^{-1}(\mathbf{x}) \right) \delta_{ij} N_B \left(\varphi_t^{-1}(\mathbf{x}) \right) \mathrm{d}v.
$$

(10.31)

The second term of (10.29) can be simplified via

$$
\int_{\varphi^h(\Omega)} \left(\sum_{A=1}^{n_{np}} \nabla N_A \left(\varphi_t^{-1}(\mathbf{x}) \right) \otimes \mathbf{c}_A \right) : \mathbf{T}^h \mathrm{d}v
$$

$$
= \int_{\varphi^h(\Omega)} \left(\sum_{A=1}^{n_{np}} \sum_{i=1}^{3} \sum_{j=1}^{3} N_{A,j} \left(\varphi_t^{-1}(\mathbf{x}) \right) c_{iA} T_{ij}^h \right) \mathrm{d}v = \sum_{P=1}^{n_{dof}} c_P F_P^{\mathrm{int}}
$$

(10.32)

where

$$
F_P^{\mathrm{int}} = \int_{\varphi^h_t(\Omega)} \left[\sum_{j=1}^{3} N_{A,j} \left(\varphi_t^{-1}(\mathbf{x}) \right) T_{ij} N_B \right] \mathrm{d}v
$$

(10.33)

Finally, the last two terms of (10.29) can be treated as

$$
\int_{\varphi^h_t(\Omega)} \left(\sum_{A=1}^{n_{np}} N_A \left(\varphi_t^{-1}(\mathbf{x}) \right) \mathbf{c}_A \right) \cdot \mathbf{f} \mathrm{d}v + \int_{\varphi^h_t(\Gamma_\sigma)} \left(\sum_{A=1}^{n_{np}} N_A \left(\varphi_t^{-1}(\mathbf{x}) \right) \mathbf{c}_A \right) \cdot \bar{\mathbf{t}} \mathrm{d}a = \sum_{P=1}^{n_{dof}} c_P F_P^{\mathrm{ext}} \quad (10.34)
$$

where

$$
F_P^{\mathrm{ext}} = \int_{\varphi^h_t(\Omega)} N_A \left(\varphi_t^{-1}(\mathbf{x}) \right) f_i \mathrm{d}v + \int_{\varphi^h_t(\Gamma_\sigma)} N_A \left(\varphi_t^{-1}(\mathbf{x}) \right) \cdot \bar{t}_i \mathrm{d}a.
$$

(10.35)

10.5 Generation of Vector/Matrix Equations

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

$$
\begin{aligned}
\mathbf{c} &= \{c_P\} \\
\mathbf{d}(t) &= \{d_Q(t)\} \\
\mathbf{F}^{\text{int}}(\mathbf{d}(t)) &= \{F_P^{\text{int}}\} \\
\mathbf{F}^{\text{ext}} &= \{F_P^{\text{ext}}\} \\
\mathbf{M} &= [M_{PQ}]
\end{aligned}
\tag{10.36}
$$

The results of (10.30)–(10.35) can now be summarized as

$$
\mathbf{c}^T \left[\mathbf{M}\ddot{\mathbf{d}}(t) + \mathbf{F}^{\text{int}}(\mathbf{d}(t)) - \mathbf{F}^{\text{ext}} \right] = 0,
\tag{10.37}
$$

which must hold for all n_{dof}-vectors $\mathbf{c}$ that result in satisfaction of the homogeneous boundary condition imposed on W (i.e., (10.18)).

Finally we observe that not all of the members of $\mathbf{d}(t)$ are unknown; for 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 (10.37) corresponding to free degrees of freedom must be identically zero, so that (10.37) will hold for arbitrary combinations of the c_P. Thus we can write the nonlinear equation that expresses the discrete equations of motion:

$$
\mathbf{M}\ddot{\mathbf{d}}(t) + \mathbf{F}^{\text{int}}(\mathbf{d}(t)) = \mathbf{F}^{\text{ext}}.
\tag{10.38}
$$

Here we employ a slight abuse of notation because we have asserted in (10.36) that all vectors and matrices have dimension n_{dof}, yet we only enforce (10.38) for free degrees of freedom. Denoting the number of free degrees of freedom as n_{eq}, on can account for this difference in practice by calculating the vector and matrix entries for all degrees of freedom and then merely disregarding the $n_{dof} - n_{eq}$ 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 (10.38). It should simply be remembered that only n_{eq} members of $\mathbf{d}(t)$ are, in fact, unknown. We will have an opportunity to visit the general topic of constraint enforcement in greater detail when discussing solutions to these nonlinear equations (see Section 13).

10.6 Localization and Assembly

The description 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 detail of shape function construction will be discussed in Section 14 in the context of element programming. However it is useful to discuss here the basic character of finite element approximation functions to give general insight into the structure of the method. We refer to Fig. 10.3 which depicts a node A in Ω, along with the 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 sub-portion 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.
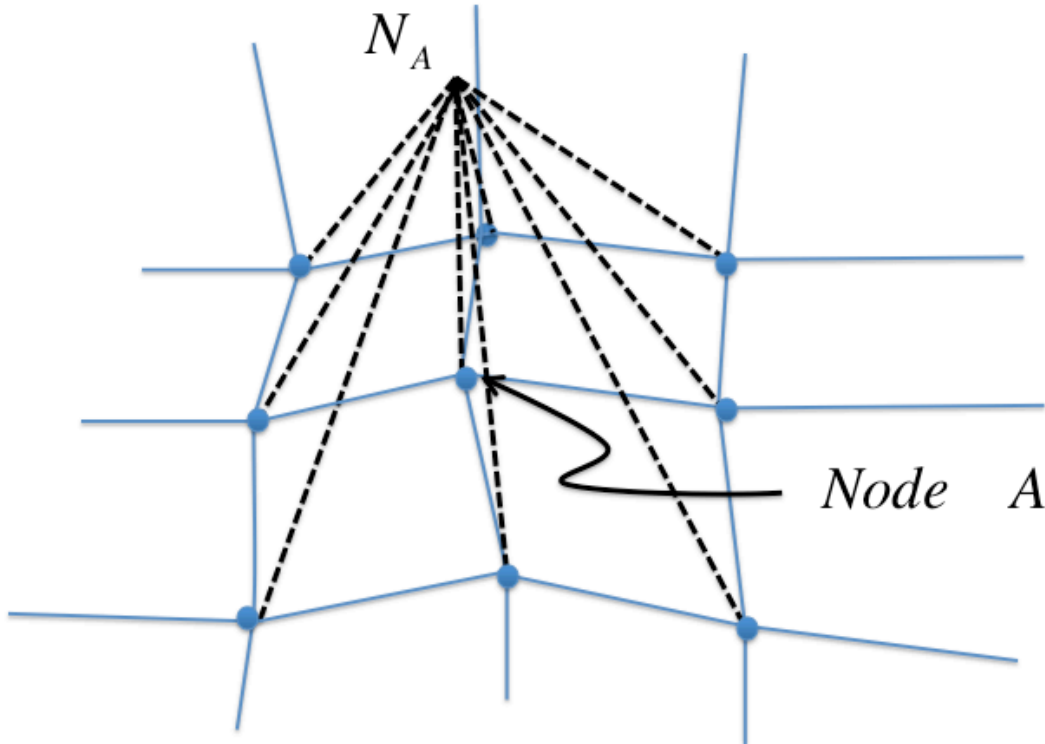
Fig. 10.3 Local support of finite element interpolation functions. The region of support for N_A shown as shaded.

To gain insight into the effect of this property, we examine the expression given in (10.31) for an element of the mass matrix M_{PQ}. We note in particular that the integrand of (10.31) will 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 conclude that very few Nodes B will produce nonzero column entries in $\mathbf{M}$. This matrix is therefore sparse, and it would be a tremendous waste of time 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 (10.38) are more typically computed using two important concepts: localization and assembly. Still considering the matrix **M** as an example, we note that by the elementary properties of integration, we can write

$$
\begin{aligned}
M_{PQ} &= \int_{\varphi_t^h(\Omega)} \rho N_A \left(\varphi_t^{-1}(\mathbf{x}) \right) \delta_{ij} N_B \left(\varphi_t^{-1}(\mathbf{x}) \right) dv \\
&= \sum_{e=1}^{n_{el}} \int_{\varphi_t^h(\Omega^e)} \rho N_A \left(\varphi_t^{-1}(\mathbf{x}) \right) \delta_{ij} N_B \left(\varphi_t^{-1}(\mathbf{x}) \right) dv \\
&= \sum_{e=1}^{n_{el}} M_{PQ}^e,
\end{aligned}
\tag{10.39}
$$

where

$$
M_{PQ}^e = \int_{\varphi_t^h(\Omega^e)} \rho N_A \left(\varphi_t^{-1}(\mathbf{x}) \right) \delta_{ij} N_B \left(\varphi_t^{-1}(\mathbf{x}) \right) dv.
\tag{10.40}
$$

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 degrees of freedom indices p and q, as indicated in Fig. 10.4. Assuming that p and q can take on values between 1 and n_{edof}, where n_{edof} is the number of degrees of freedom associated with the element, an $n_{edof} \times n_{edof}$ matrix $\mathbf{m}$ is constructed as

$$
\mathbf{m}^e = \left[m_{pq}^e \right].
\tag{10.41}
$$

The m_{pq}^e can be specified by introducing the concept of a local node number a or b as shown in Fig. 10.4. With these definitions we can write

$$
m_{pq}^e = \int_{\varphi_t^h(\Omega^e)} \rho N_a \left(\varphi_t^{-1}(\mathbf{x}) \right) \delta_{ij} N_b \left(\varphi_t^{-1}(\mathbf{x}) \right) dv
\tag{10.42}
$$

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

$$
p = (a - 1) \times 2 + i
\tag{10.43}
$$

(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 the local element entities, such as $\mathbf{m}^e$, turns out to be highly modular procedure whose form remains essentially unchanged for any element in a mesh. Detailed discussion of this calculation is deferred until Section 14.
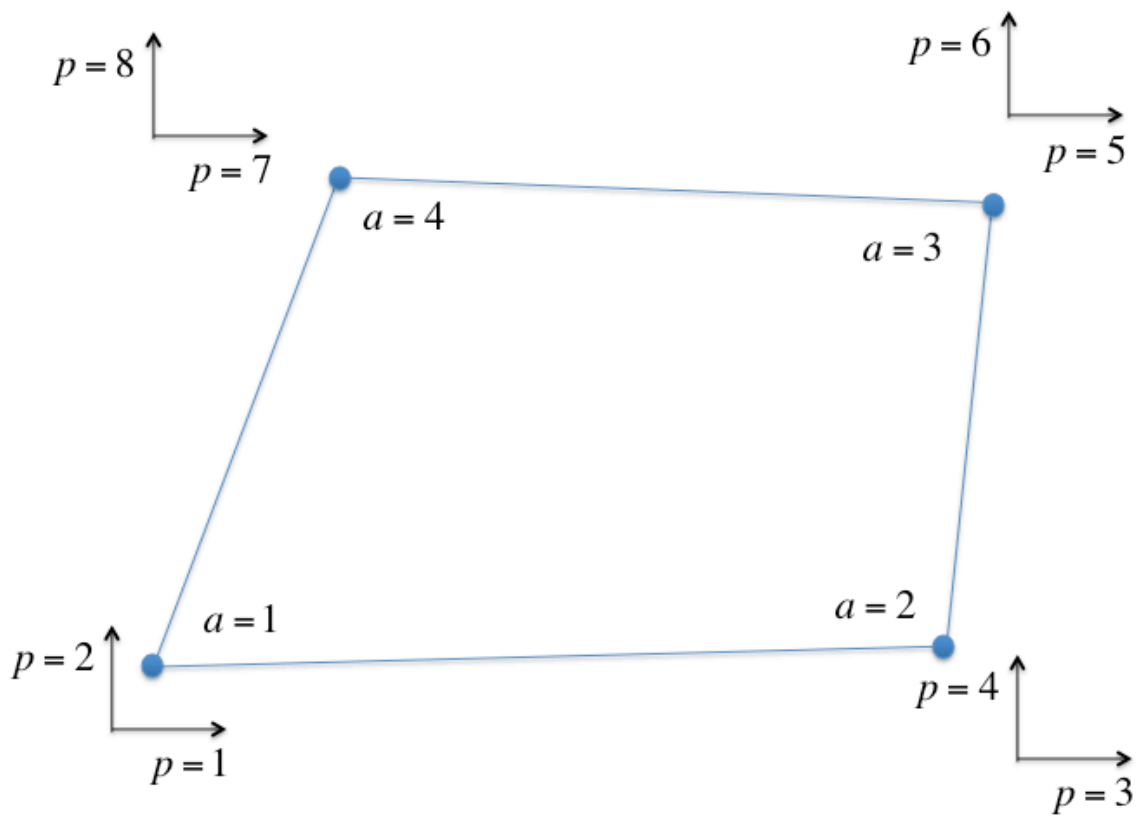
Fig. 10.4 Element (local) degrees of freedom for a sample finite element.

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 $\mathbf{F}^{\text{int}}$:

- Zero out $\mathbf{M}$, $\mathbf{F}^{\text{int}}$.

- For each element e, $e = 1, \ldots, n_{el}$:

  – Prepare local data necessary for element calculations - e.g., $\mathbf{X}^e$ (n_{edof} - vector of element nodal coordinates), $\mathbf{d}^e$ (n_{edof}-vector of element nodal configuration mappings), etc.

  – Calculate element internal force vector $\mathbf{f}^{\text{int},e} = \left\{ f_p^{\text{int},e} \right\}$ and element mass matrix $\mathbf{m}^e = \left[m_{pq}^e \right]$ via

$$f_p^{\text{int},e} = \int_{\varphi_t^h(\Omega^e)} \left[\sum_{j=1}^{3} N_{a,j} \left(\varphi_t^{-1}(\mathbf{x}) \right) T_{ij}^h \right] \mathrm{d}v \qquad (10.44)$$

  and (10.42).

  – 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:

$$M_{PQ} = M_{PQ} + m_{pq}^e \qquad (10.45)$$

  and

$$F_P^{\text{int}} = F_P^{\text{int}} + f_p^{\text{int},e}, \qquad (10.46)$$

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

$$P = LM(p, e) \qquad (10.47)$$

  and

$$Q = LM(q, e). \qquad (10.48)$$

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 calculations will be discussed in detail in Section 14. Step 2c) is the process known as assembly and takes the data produced by the element level calculations and assembles 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 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 chapters on the classes of problems and global equation-solving strategies to be utilized.

11 Quasistatics

11.1 Quasistatic Assumption

As discussed previously in the context of a Linear Elastic IBVP, the **quasistatic approximation** is appropriate when inertial forces are negligible compared to the internal and applied forces in a system. The question of what is negligible generally relies on intuition, and numerical experimentation is one way to gain this intuition.

Omission of the inertial term in the discrete equations of motion, (10.38), yields a quasistatic problem of the form

$$\mathbf{F}^{\text{int}}(\mathbf{d}(t)) = \mathbf{F}^{\text{ext}} \tag{11.1}$$

subject to only one initial condition of the form

$$\mathbf{d}(0) = \mathbf{d}_0. \tag{11.2}$$

Note that the time variable, t 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 t to be taken as a generic parameterization for the applied loading on the system as discussed below.

11.2 Internal Force Vector

The quantity $\mathbf{F}^{\text{int}}(\mathbf{d}(t))$ 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

$$F_P^{\text{int}} = \int_{\varphi_t^h(\Omega)} \left(\sum_{j=1}^{3} N_{A,j} \left(\varphi_t^{-1}(\mathbf{x}) \right) T_{ij}^h \right) dv. \tag{11.3}$$

This vector-valued operator is generally a nonlinear function of the unknown solution vector $\mathbf{d}(t)$ due to the possible material nonlinearity and/or geometric nonlinearity inherent in the definition of the Cauchy stress T_{ij}^h in (11.3). As implied by our notation, we assume the solution vector $\mathbf{d}$ to be smoothly parameterized by t which may represent time or some other loading parameter.

11.3 External Force Vector

The external load vector $\mathbf{F}^{\text{ext}}(t)$ must equilibrate the internal force vector, as is clear from (11.1). As presented in the previous chapter, the expression of an element F_P^{ext} of $\mathbf{F}^{\text{ext}}(t)$ is

$$F_P^{\text{ext}} = \int_{\varphi_t^h(\Omega)} N_A \left(\varphi_t^{-1}(\mathbf{x}) \right) f_i(t) dv + \int_{\varphi_t^h(\Gamma_\sigma)} N_A \left(\varphi_t^{-1}(\mathbf{x}) \right) \cdot \bar{t}_i(t) da, \tag{11.4}$$

where the explicit dependence of f_i and $\bar{t}_i$ upon t has been indicated and where $P = ID(i, a)$ as given in (10.27). In other words, we assume that the prescribed external force loadings f_i and prescribed surface tractions $\bar{t}_i$ are given functions of t.

(11.4) implies no dependence of either $\bar{t}_i$ or f_i upon $\varphi_t(\mathbf{x})$ (and thus $\mathbf{d}$). Provided no such dependence exists, the external force is completely parameterized by t, and the sole dependence of the equilibrium equations on $\mathbf{d}$ occurs through $\mathbf{F}^{\text{int}}$. However , it is important to realize that some important loading cases are precluded by this assumption. Perhaps the most important being the case of pressure loading, where the direction of applied traction is opposite to the surface normal, which in large deformation problems depends upon $\varphi_t(\mathbf{x})$. Such a load is sometimes called a follower force and will, in general, contribute additional nonlinearities. Such nonlinearities are handled notationally, simply by recognizing that the traction $\bar{t}_i$ now depends on $\varphi_t(\mathbf{x})$, i.e.,

$$F_P^{\text{ext}} = \int_{\varphi_t^h(\Omega)} N_A \left(\varphi_t^{-1}(\mathbf{x}) \right) f_i dv + \int_{\varphi_t^h(\Gamma_\sigma)} N_A \left(\varphi_t^{-1}(\mathbf{x}) \right) \cdot \bar{t}_i(t, \varphi_t(\mathbf{x})) da. \tag{11.5}$$

11.4 Incremental Load Approach

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

$$t \in [0, \text{T}] \tag{11.6}$$

We subdivide this interval of interest into a set of sub-intervals via

$$[0, \text{T}] = \bigcup_{n=0}^{N-1} [t_n, t_{n+1}], \tag{11.7}$$

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 $t_N = \text{T}$, but we do not, in general, assume that all time intervals $[t_n, t_{n+1}]$ have the same width.

With this notation, 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:

$$\mathbf{F}^{\text{int}}(\mathbf{d}_{n+1}) = \mathbf{F}^{\text{ext}}(\mathbf{d}_{n+1}). \tag{11.8}$$

where we have included an assumed dependence of the external loading on deformation $\varphi_t(\mathbf{x})$.

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

$$\mathbf{r}\left(\mathbf{d}_{n+1}\right) = \mathbf{F}^{\text{ext}}\left(\mathbf{d}_{n+1}\right) - \mathbf{F}^{\text{int}}\left(\mathbf{d}_{n+1}\right). \qquad (11.9)$$

Solution of (11.8), therefore, amounts to finding the root of the equation

$$\mathbf{r}\left(\mathbf{d}_{n+1}\right) = 0. \qquad (11.10)$$

The importance of stating equilibrium in this manner will be made much clearer in the Chapter discussing nonlinear equation solving, (chapter Section 13). For the moment, the physical meaning of this approach is depicted graphically in Fig. 11.1. Starting with an initial equilibrium state t_n, so that $\mathbf{r}(\mathbf{d}_n) = 0$, we introduce an increment in the prescribed load and attempt to find that displacement increment, $\mathbf{d}_{n+1} - \mathbf{d}_n$, that will restore equilibrium (i.e., result in satisfaction of (11.10)). This will require a nonlinear equation solving technique for determination of $\mathbf{d}_{n+1}$, a topic that will be discussed further in Section 13.
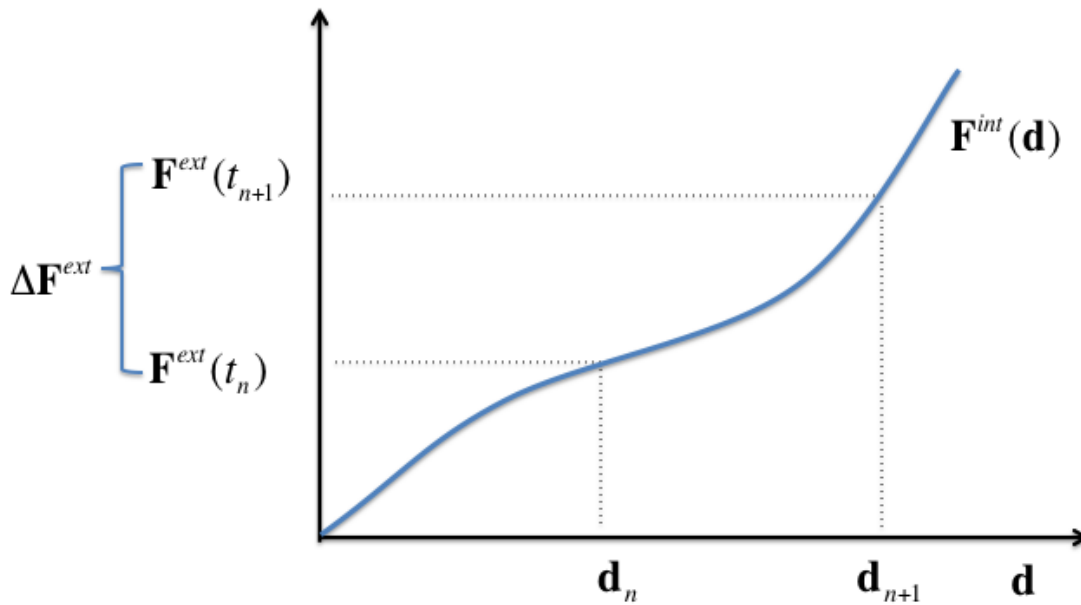
Fig. 11.1 Simple illustration of the incremental load approach to quasistatics problems

This page left blank

12 Dynamics

12.1 Semi-Discrete Approach

We now include the inertial terms in the discrete equation system and consider solving

$$\mathbf{M}\ddot{\mathbf{d}}(t) + \mathbf{F}^{\text{int}}(\mathbf{d}(t)) = \mathbf{F}^{\text{ext}}(\mathbf{d}(t)) \tag{12.1}$$

for $t \in [0, \text{T}]$ subject to the initial conditions

$$\mathbf{d}(0) = \mathbf{d}_0 \tag{12.2}$$

and

$$\dot{\mathbf{d}}(0) = \mathbf{v}_0. \tag{12.3}$$

Note that in (12.1) time remains continuous, whereas spatial discretization has already been achieved by the finite element interpolations summarized in Section 10. This type of finite element approach to transient problems is sometimes referred to as the semi-discrete 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 next.

12.2 Time-Stepping Procedures

As discussed in Section 11, we subdivide the time interval of interest $[0, \text{T}]$ via

$$[0, \text{T}] = \bigcup_{n=0}^{N-1} [t_n, t_{n+1}] \tag{12.4}$$

and consider the problem:

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

A thoroughly studied topic in dynamics is the construction of effective time integrators for application to the semi-discrete equations of motion. An ideal approach possesses minimal dispersion and dissipation. As shown in Fig. 12.1, a measure of **numerical dispersion** is period error $(\bar{T} - T)$, and a measure of **numerical dissipation** is amplitude decay $(\bar{A} - A)$. Fig. 12.1 depicts a single wave with amplitude and period A and T that generically is the exact solution to the wave equation (subject to the proper initial conditions and/or external force). Numerical dispersion by the time integrator causes a wave's frequency to decrease, thus dispersing its energy

A measure of
numerical dispersion
is period error

d

A

$\bar{A}$

T $\bar{T}$

A measure of
numerical dissipation
is amplitude decay

time

exact integration of wave propagation
numerical integration
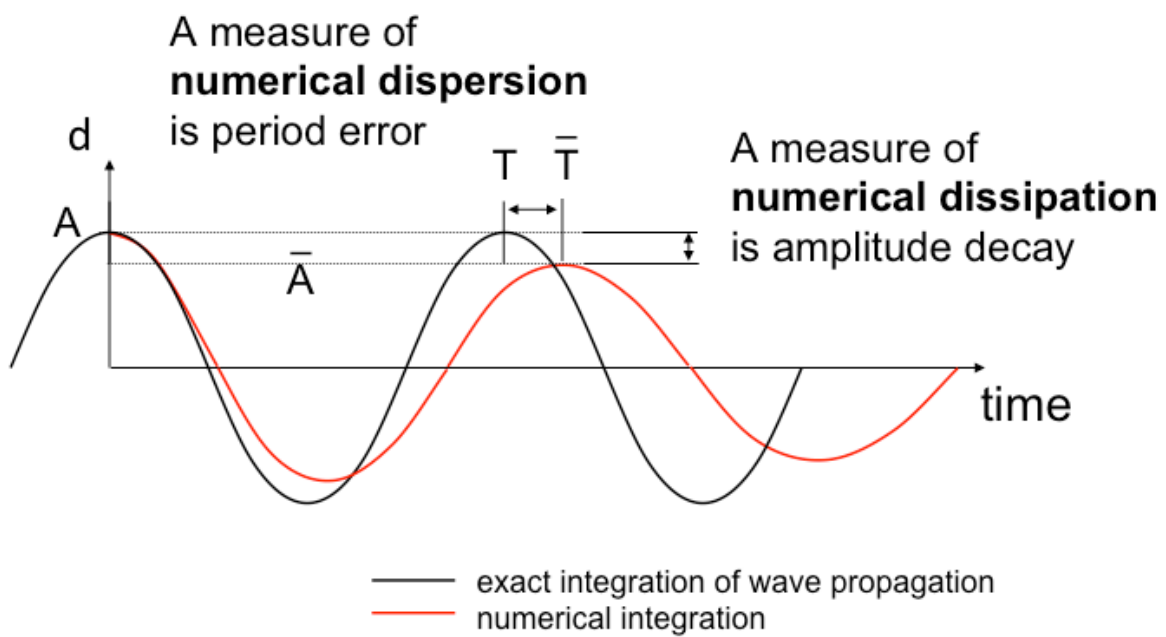
Fig. 12.1 Simple illustration of approximation error in transient time integrators.

to the lower frequencies. Numerical dissipation by the time integrator causes the wave's energy to decrease and therefore is said to dissipate its energy.

The time integrators we consider here can all be described by a 3-parameter method called the α-method of time integrators. It is also referred to as the Hilber-Hughes-Taylor Method, or HHT method, as described in Reference [25]], which is a generalization of the well-known and pervasive Newmark family of temporal integrators (Reference [[40]). The Newmark algorithm can be summarized in a time step $[t_n, t_{n+1}]$ as follows:

$$\mathbf{M}\mathbf{a}_{n+1} + \mathbf{F}^{\text{int}}(\mathbf{d}_{n+1}) = \mathbf{F}^{\text{ext}}(\mathbf{d}_{n+1})$$

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \Delta t \mathbf{v}_n + \frac{\Delta t^2}{2}\left[(1 - 2\beta)\mathbf{a}_n + 2\beta\mathbf{a}_{n+1}\right] \tag{12.5}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \Delta t \left[(1 - \gamma)\mathbf{a}_n + \gamma\mathbf{a}_{n+1}\right],$$

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

The extension of the Newmark family of integrators to the α-method of integrators is accomplished with the addition of the parameter, α:

$$\mathbf{M}\mathbf{a}_{n+1} + (1 + \alpha)\mathbf{F}^{\text{int}}(\mathbf{d}_{n+1}) - \alpha\mathbf{F}^{\text{int}}(\mathbf{d}_n) = (1 + \alpha)\mathbf{F}^{\text{ext}}(\mathbf{d}_{n+1}) - \alpha\mathbf{F}^{\text{ext}}(\mathbf{d}_n)$$

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \Delta t \mathbf{v}_n + \frac{\Delta t^2}{2}\left[(1 - 2\beta)\mathbf{a}_n + 2\beta\mathbf{a}_{n+1}\right] \tag{12.6}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \Delta t \left[(1 - \gamma)\mathbf{a}_n + \gamma\mathbf{a}_{n+1}\right],$$

where, as expected, setting α to zero reduces the HHT integrator to Newmark's method. Although a wide range of algorithms exist corresponding to the different available choices of β and γ, two algorithms in particular are significant:

- Central Differences ($\alpha = 0$, $\beta = 0$, $\gamma = 1/2$). This integrator is second-order accurate in time and only *conditionally stable*, meaning that the linearized stability is only retained when Δt is less than some critical value. This algorithm is an example of an **explicit** finite element integrator discussed in Section 12.3.

- Trapezoid rule ($\alpha = 0$, $\beta = 1/4$, $\gamma = 1/2$). This integrator is also second-order accurate but *unconditionally stable* for linear problems, meaning that the spectral radii of the integrator remains less than one in modulus for any time step Δt (in linear problems). This algorithm is an example of an **implicit** finite element integrator discussed in Section 12.4

12.3 Explicit Finite Element Methods

Examining the central differences algorithm, we substitute $\beta = 0$, $\gamma = 1/2$ into (12.6) to obtain

$$\mathbf{a}_{n+1} = \mathbf{M}^{-1}\left(\mathbf{F}^{\text{ext}}(\mathbf{d}_{n+1}) - \mathbf{F}^{\text{int}}(\mathbf{d}_{n+1})\right)$$

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \Delta t \mathbf{v}_n + \frac{\Delta t^2}{2}\mathbf{a}_n \tag{12.7}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \frac{\Delta t}{2}\left[\mathbf{a}_n + \mathbf{a}_{n+1}\right],$$

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

(12.7) can be used to explain why this formulation is termed explicit. 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 *provided the mass matrix* $\mathbf{M}$ *is a diagonal matrix.*

It is important to note approximation properties of the explicit time integrator (see Reference [33]). By itself, the explicit time integrator causes the period to be shortened. However, a *lumped* or diagonalized mass matrix as opposed to a consistent mass matrix causes the period to be elongated. For one-dimensional problems with uniform meshes the period error cancels exactly. In the words of Reference [33], these compensating errors generally produce a matched approach. Thus a lumped mass matrix gives rise to the fully explicit algorithm, requiring only an inverse of a diagonal matrix.

Although this form of the central difference formulation ((12.7)) is readily obtained from the Newmark formulas, it does not give insight into the source of the **central difference** terminology and, in fact, does not represent the (historical) manner in which the integrator is ordinarily developed or implemented. To see the usual form, one starts with the **difference formulas** for acceleration and velocity (see e.g., The Difference Calculus, Chapter 9 in Reference [22]):

$$\mathbf{a}_n = \frac{\mathbf{v}_{n+1/2} - \mathbf{v}_{n-1/2}}{t_{n+1/2} - t_{n-1/2}}, \tag{12.8}$$

and

$$\mathbf{v}_{n+1/2} = \frac{\mathbf{d}_{n+1} - \mathbf{d}_n}{t_{n+1} - t_n}, \tag{12.9}$$

where, as shown in Fig. 12.2, the time axis is discretized with notions of whole step configurations at times t_{n-1}, t_n, t_{n+1} and half-step configurations at times $t_{n-1/2}, t_{n+1/2}, \ldots$

Rearranging, these difference formulas ((12.8) and (12.9)) can be converted into integration formulas:

$$\begin{aligned} \mathbf{v}_{n+1/2} &= \mathbf{v}_{n-1/2} + \frac{1}{2} \left(\Delta t_{n-1/2} + \Delta t_{n+1/2} \right) \mathbf{a}_n \\ \mathbf{d}_{n+1} &= \mathbf{d}_n + \Delta t_{n+1/2} \mathbf{v}_{n+1/2} \end{aligned} \tag{12.10}$$

Combining these integration formulas with the equilibrium equation evaluated at t_n, we can express the algorithm as

$$\begin{aligned} \mathbf{a}_n &= \mathbf{M}^{-1} \left[\mathbf{F}^{\text{ext}}(\mathbf{d}_n) - \mathbf{F}^{\text{int}}(\mathbf{d}_n) \right] \\ \mathbf{v}_{n+1/2} &= \mathbf{v}_{n-1/2} + \frac{1}{2} \left(\Delta t_{n-1/2} + \Delta t_{n+1/2} \right) \mathbf{a}_n \\ \mathbf{d}_{n+1} &= \mathbf{d}_n + \Delta t_{n+1/2} \mathbf{v}_{n+1/2} \end{aligned} \tag{12.11}$$

The velocity and displacement updates emanate from the central difference approximations to the acceleration $\mathbf{a}_n$ and velocity $\mathbf{v}_{n+1/2}$, respectively, giving the algorithm its name. The velocity measures that are utilized by the algorithm are shifted by a half step (said to be centered at the
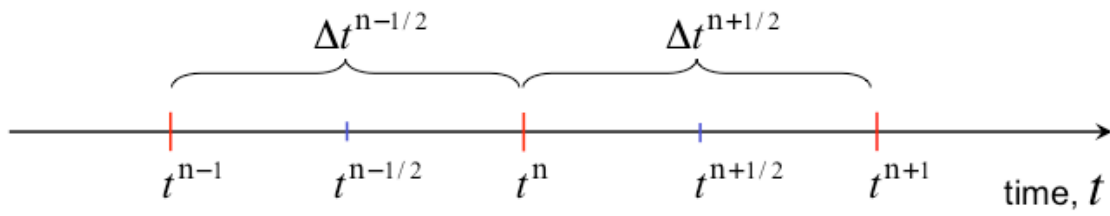
Fig. 12.2 Graphical construction of the central difference time integrator.
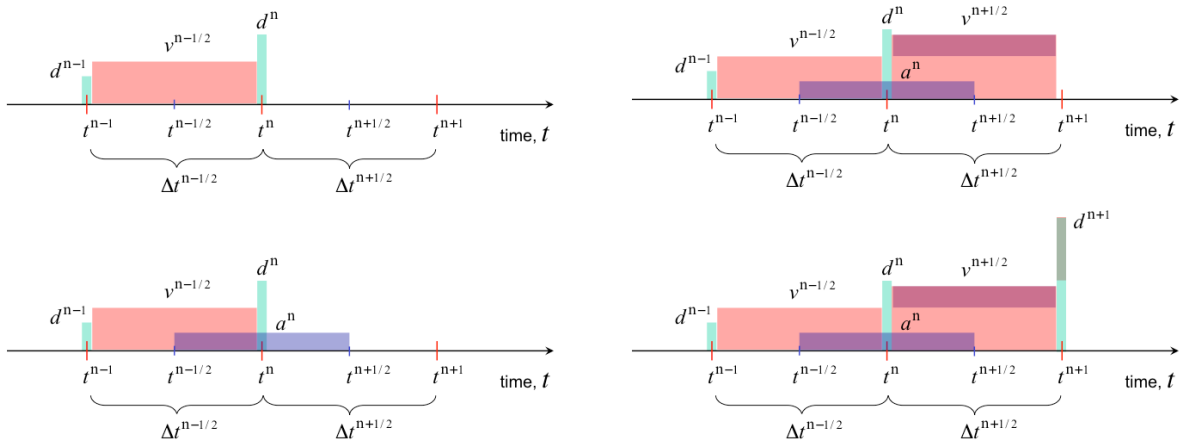
Fig. 12.3 Graphical representation of the central difference time integrator.

half-step), whereas accelerations and displacements are centered at the whole step. Fig. 12.3 graphically reveals the simplicity of the explicit time integration scheme.

As already mentioned, 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 (see Reference [5]), can be shown to be as follows

$$\Delta t \leq \frac{2}{\omega}, \tag{12.12}$$

where ω is the highest natural frequency in the mesh. An important necessary step in the central difference explicit time integrator is the estimation of this highest natural frequency in the discretized problem. Explicit dynamics problems frequently involve large deformations with potentially significant geometric, material, and contact nonlinearities, all of which can cause significant changes in the critical time step. Therefore, estimation of the critical time step must be made repeatedly throughout the problem simulation. It is thus important that the this calculation be as accurate and efficient as possible to make the most of the explicit method.

12.3.1 Element-based Critical Time Step Estimate

Stable time step estimates for explicit finite element methods are traditionally based on the conservative estimate of the frequency:

$$\omega = 2 \left(\frac{c}{h} \right)_{\text{max}}, \tag{12.13}$$

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 (12.12) and (12.13) we find that

$$\Delta t^{\text{max}} = \left(\frac{h}{c} \right)_{\text{min}}. \tag{12.14}$$

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. Such an estimation of the critical time step is based solely on element level calculations and is, in fact, part of the element internal force calculation. This is due in large part to the estimate of the sound speed of the material, which is of a dilatational wave. The accuracy of directly applying this condition is limited in practice due to the arbitrary finite element geometries in a typical mesh because the definition of characteristic length is somewhat of an art for distorted elements. Alternatively, the stability limit as reported in Reference [33] is related to the maximum global eigenvalue, λ_{max}:

$$\Delta t^2 = \frac{4}{\lambda_{\text{max}}}. \tag{12.15}$$

Because the maximum element eigenvalue is an upper bound on the maximum global eigenvalue (Reference [12]), we can compute an element-based stable time step estimate using

$$\Delta t^E = \frac{2}{\sqrt{\lambda}\Big|_{\text{max over } e}}. \tag{12.16}$$

Details of how this element-based time step is calculated for different elements are covered in the chapter on element formulations.

12.3.2 Nodal-based Critical Time Step Estimate

A method is now described in which the maximum element modal stiffness are used to estimate a maximum nodal stiffness which, when combined with the lumped nodal mass, gives a sharper upper bound on the maximum global eigenvalue.

Let $\lambda_{\max}$ denote the largest eigenvalue of the generalized problem

$$(\mathbf{K} - \lambda\mathbf{M})\,\mathbf{u} = 0 \tag{12.17}$$

and $\mathbf{u}_{\max}$ the eigenvalue corresponding to $\lambda_{\max}$. In (12.17), $\mathbf{K}$ is the stiffness matrix and $\mathbf{M}$ the diagonal, lumped mass matrix. The Rayleigh quotient for the maximum eigenvalue is

$$\lambda_{\max} = \frac{\mathbf{u}_{\max}^T\mathbf{K}\mathbf{u}_{\max}}{\mathbf{u}_{\max}^T\mathbf{M}\mathbf{u}_{\max}}. \tag{12.18}$$

Noting that the numerator of (12.18) is twice the strain energy S of the system when deformed into the mode shape $\mathbf{u}_{\max}$, we can write

$$2S = \mathbf{u}_{\max}^T\mathbf{K}\mathbf{u}_{\max} = \sum_{e=1}^{n_e} \left(\mathbf{u}_{\max}^e\right)^T \mathbf{K}^e \left(\mathbf{u}_{\max}^e\right). \tag{12.19}$$

We observe that the eigenvalue problem for the element stiffness matrix $\mathbf{K}^e$ may be stated as

$$\mathbf{K}^e \phi^e = k^e \phi^e. \tag{12.20}$$

Consequently,

$$(\mathbf{u}^e)^T\mathbf{K}^e\mathbf{u}^e \le k_{\max}^e(\mathbf{u}^e)^T\mathbf{u}^e \tag{12.21}$$

for all $\mathbf{u}^e$ where $k_{\max}^e$ is the maximum eigenvalue (so called modal stiffness) of the element stiffness matrix. From this result, we define a global stiffness matrix $\hat{\mathbf{K}}$ assembled from the element stiffness matrices $\hat{\mathbf{K}}^e$ defined as

$$\hat{\mathbf{K}}^e = k_{\max}^e\mathbf{I}^e \tag{12.22}$$

where $\mathbf{I}^e$ is an *ndofe* by *ndofe* identity matrix (*ndofe* is the number of degrees of freedom in the element). Based on (12.18), (12.20) and (12.21),

$$2S \le \sum_{e=1}^{n_e} \left(\mathbf{u}_{\max}^e\right)^T \hat{\mathbf{K}}^e \left(\mathbf{u}_{\max}^e\right) \tag{12.23}$$

leading to

$$\lambda_{\max} = \frac{\mathbf{u}_{\max}^T\mathbf{K}\mathbf{u}_{\max}}{\mathbf{u}_{\max}^T\mathbf{M}\mathbf{u}_{\max}} \le \frac{\mathbf{u}_{\max}^T\hat{\mathbf{K}}\mathbf{u}_{\max}}{\mathbf{u}_{\max}^T\mathbf{M}\mathbf{u}_{\max}} = \hat{\lambda}_{\max}. \tag{12.24}$$

Given the mode shape $\mathbf{u}_{\max}$, the expression for $\hat{\lambda}_{\max}$ is easily evaluated since both $\hat{\mathbf{K}}$ and $\mathbf{M}$ are diagonal. Methods for predicting this mode shape have been developed for specific 'template' geometries (Reference [17], but for general finite element geometries this remains impractical.

Rather than directly calculating $\hat{\lambda}_{\max}$, we seek an upper bound. To this end, we define the ratio for every node I as

$$\hat{\lambda}^I = \frac{\hat{K}^I}{M^I}, \tag{12.25}$$

where $\hat{K}^I$ and M^I are the diagonal elements in the Ith row of $\hat{\mathbf{K}}$ and $\mathbf{M}$, respectively. Without loss of generality, the ratios are ordered such that $\hat{\lambda}^m \geq \hat{\lambda}^{m-1} \geq \cdots \geq \hat{\lambda}^1$, in which case (12.24) can be written as

$$\hat{\lambda}_{\max} = \frac{\sum_I \mathbf{u}_{\max I}^T \hat{\mathbf{K}}^I}{\sum_I \mathbf{u}_{\max I}^T \mathbf{M}^I} = \hat{\lambda}^m \left\{ \frac{1 + \left[\left((u_{\max}^{m-1})^2 M^{m-1} \right) / \left((u_{\max}^m)^2 M^m \right) \right] \frac{\hat{\lambda}^{m-1}}{\hat{\lambda}^m} + \cdots}{1 + \left[\left((u_{\max}^{m-1})^2 M^{m-1} \right) / \left((u_{\max}^m)^2 M^m \right) \right] + \cdots} \right\} \tag{12.26}$$

Since all the ratios $\hat{\lambda}^{m-1}/\hat{\lambda}^m$ are less than or equal to one, it follows immediately that

$$\hat{\lambda}_{\max} \leq \hat{\lambda}^m = \left. \frac{\hat{K}^I}{M^I} \right|_{\text{max over } I}, \tag{12.27}$$

in which M^I is the lumped mass at node I, and $\hat{K}^I$ is the assembly of the maximum element modal stiffness at node I, that is

$$\hat{K}^I = \sum_{e \in e^I} k_{\max}^e, \tag{12.28}$$

where e^I is the set of elements that are connected to node I.

(12.15), (12.24), and (12.27) lead to a nodal-based stable time step estimate:

$$\Delta t^N = \frac{2}{\sqrt{\left. \frac{\hat{K}^I}{M^I} \right|_{\text{max over } e}}}. \tag{12.29}$$

Now we show that the nodal-based stable time step estimate is always greater than or equal to the element-based estimate. Following a similar procedure outlined in (12.26), we can write

$$\hat{\lambda}^I = \frac{\hat{K}^I}{M^I} = \frac{\sum_{e \in e^I} k_{\max}^e}{\sum_{e \in e^I} m^e} = \frac{\frac{k_{\max}^1}{m^1} + (m^2/m^1)\frac{k_{\max}^2}{m^2} + \cdots}{1 + (m^2/m^1) + \cdots} \leq \frac{k_{\max}^1}{m^1}, \tag{12.30}$$

where the element eigenvalues $k_{\max}^e/m^e$ are arranged in descending order, $k_{\max}^1/m^1 \geq k_{\max}^2/m^2 \geq \ldots$. Thus the nodal-based estimate of the maximum eigenvalue at node I is bounded by the largest of all element eigenvalues connected to node I. It follows from (12.30) that

$$\hat{\lambda}^m = \left. \frac{\hat{K}^I}{M^I} \right|_{\text{max over } I} \leq \left. \left. \frac{k_{\max}^e}{m^e} \right|_{\text{max over } e \in e^I} \right|_{\text{max over } I} = \left. \frac{k_{\max}^e}{m^e} \right|_{\text{max over } e} = \left. \lambda_{\max}^e \right|_{\text{max over } e} \tag{12.31}$$

Since $\hat{\lambda}^m \leq \lambda^e_{\max}\big|_{\text{max over }e}$, it follows directly from (12.15) that the nodal-based estimate is always greater than or equal to the element-based estimate.

The cost of the nodal-based estimate calculation includes the element eigenvalue analysis (which must be done in the case of the element based calculation) plus the cost of an assembly procedure every time step. (12.29) must be evaluated at each node as opposed to evaluating (12.16) for every element.

12.3.3 Lanczos-based Critical Time Step Estimate

The paper [32], which is reproduced here, demonstrates the cost-effective use of the Lanczos method for estimating the critical time step in an explicit, transient dynamics code. The Lanczos method can give a significantly larger estimate for the critical time-step than an element-based method (the typical scheme). However, the Lanczos method represents a more expensive method for calculating a critical time-step than element-based methods. Our paper shows how the additional cost of the Lanczos method can be amortized over a number of time steps and lead to an overall decrease in run-time for an explicit, transient dynamics code. We present an adaptive hybrid scheme that synthesizes the Lanczos-based and element-based estimates and allows us to run near the critical time-step estimate provided by the Lanczos method.

12.3.3.1 Introduction

Codes using explicit time integration techniques are important for simulating transient dynamics problems involving large deformation of solids with various nonlinear effects (contact, nonlinear materials, element death, etc.). The second order central difference operator used in explicit codes is stable if the time step is no larger than the critical time step. For most problems in solid mechanics, the critical time step is extremely small and the number of time steps required for a typical analysis is quite large. Therefore, the accurate, efficient, and reliable calculation of the critical time step is of fundamental importance.

The element-based method [13] is an efficient method for producing a critical time step estimate at every time step. However, it can produce a conservative estimate for the critical time step in many cases. The Lanczos [35] method is a reliable procedure for producing a time step that is the theoretical maximum value for a structure and is usually much better than the element-based estimate. The cost of obtaining a Lanczos based estimate will not offset the cost benefit of the increased value for the critical time step. Therefore, it is not feasible to call the Lanczos method at every explicit dynamics time step. In this paper we outline a cost-effective method for utilizing the Lanczos method (together with an element-based scheme) for the critical time step estimation.

Benson [4] investigates estimating the critical time step by using the power iteration. Parlett [44] presents analysis comparing the Lanczos method and power iteration. The Lanczos method provides a more rapid approximation, in terms of matrix-vector products, relative to the power iteration for approximating the largest eigenvalue as the relative separation of the largest eigenvalue decreases. Hence, we can expect the Lanczos method to require less matrix-vector

products to approximate the critical time step to a specified tolerance. We also remark that in contrast to our paper, Benson [4] does not present a scheme that addresses two crucial issues when using the power iteration (or Lanczos method) for estimating the critical time step.

Two crucial issues must be addressed when using the Lanczos method to estimate the critical time step. First, the Lanczos-based time step estimate must be used for two to three times the number of explicit time integration steps required to recover the cost of the Lanczos method if we are to see a noticeable reduction in overall computation times for a problem. (We explore the cost of the Lanczos method in terms of internal force calculations in later sections.) Second, the Lanczos method provides an overestimate of the critical time step, and so we need an effective scheme to scale back the Lanczos-based critical time step estimate. We address both these issues and present an adaptive hybrid scheme that synthesizes the Lanczos-based and element-based estimates and allows us to run near the critical time-step estimate provided by the Lanczos method.

We also remark that in addition to the increased efficiency that can be achieved with the Lanczos-based time step, we also have the added benefit of increased accuracy. For explicit transient dynamic codes, using a time step as close as possible to the critical time step [33] gives the most accurate answer. Reducing the time step in an explicit transient dynamics code actually increases the error.

Our paper is organized as follows. Section Section 12.3.3.2 discusses the critical time step and motivates a Lanczos-based estimate. The Lanczos iteration and method are briefly introduced in section Section 12.3.3.3. A cost benefit analysis of the element-based and Lanczos-based approximations to the critical time is considered in section Section 12.3.3.4. A practical implementation within an explicit dynamics code is the subject of section Section 12.3.3.5. Several numerical examples are presented in section Section 12.3.3.6, and we provide our conclusions in section Section 12.3.3.7.

12.3.3.2 Critical time step

Let $\mathbf{K}$ and $\mathbf{M}$ be the stiffness and mass matrices arising in an explicit dynamics simulation so that $\mathbf{M}$ is a diagonal matrix due to mass lumping. The critical time step for second order central time differencing is bounded from above by $2\omega_{\max}^{-2}$ where $\omega_{\max}^2$ is the largest eigenvalue of the generalized eigenvalue problem

$$\mathbf{K}\mathbf{u} = \mathbf{M}\mathbf{u}\omega_{\max}^2, \quad \left(\mathbf{K}, \mathbf{M} \in \mathbb{R}^{n \times n}\right), \tag{12.32}$$

where we assume that $\omega_{\max}^2$ is positive. An inexpensive [28] upper bound to ω^2 'is given by the maximum element eigenvalue $: math :omega^2\_\{max,e\}$' over all the element eigenvalue problems

$$\mathbf{K}^e\mathbf{u}^e = \mathbf{M}^e\mathbf{u}^e\omega_e^2, \quad \left(\mathbf{K}^e, \mathbf{M}^e \in \mathbb{R}^{n^e \times n^e}\right), \tag{12.33}$$

where $n^e \ll n$. Therefore, $\omega_{\max,e}^{-2} \le \omega_{\max}^{-2}$ and we have a lower bound for the critical time step. The maximal element eigenvalue is typically computed analytically [13] for the finite elements that are typically used in transient dynamics.

The Lanczos method rapidly provides a lower bound $\omega^2_{\max,L}$ to $\omega^2_{\max}$ so that

$$\omega^{-2}_{\max,e} \leq \omega^{-2}_{\max} \leq \omega^{-2}_{\max,L}. \tag{12.34}$$

In fact, the Lanczos iteration is sharp so that $\omega^{-2}_{\max} \lesssim \omega^{-2}_{\max,L}$ so that with care, an excellent approximation to the critical time step is computed for a modest cost. This approximation may be dramatically superior to the standard element based estimate. The details of a careful use of the Lanczos-based estimate is the subject of section .

12.3.3.3 Lanczos iteration

The Lanczos reduction rapidly provides approximations to the maximum and minimum eigenvalues of a symmetric $\mathbf{A} \in \mathbb{R}^{n \times n}$, in particular the largest in magnitude eigenvalue. Suppose that

$$\mathbf{A}\mathbf{Q}_j = \mathbf{Q}_j\mathbf{T}_j + \mathbf{f}_j\mathbf{e}_j^T, \tag{12.35}$$

is a Lanczos reduction of length j where $\mathbf{f}_j \in \mathbb{R}^n$, and $\mathbf{e}_j \in \mathbb{R}^j$ contains column j of the identity matrix $\mathbf{I}_n \in \mathbb{R}^{n \times n}$. If we denote

$$\mathbf{T}_j = \begin{pmatrix} \alpha_1 & \beta_2 & \cdots & 0 \\ \beta_2 & \alpha_2 & \cdots & 0 \\ \vdots & & \ddots & \beta_j \\ 0 & \cdots & \beta_j & \alpha_j \end{pmatrix}, \quad \alpha_i, \beta_i \in \mathbb{R}$$

and

$$\mathbf{Q}_j = \begin{pmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_j \end{pmatrix}, \quad \mathbf{q}_i \in \mathbb{R}^n$$

then the familiar Lanczos three-term recurrence is recovered by equating column j of ((12.35)) to obtain

$$\mathbf{f}_j = \mathbf{A}\mathbf{q}_j - \mathbf{q}_j\alpha_j - \mathbf{q}_{j-1}\beta_{j-1}^T. \tag{12.36}$$

Furthermore, because of the orthonormality of $\mathbf{Q}_j$, we have

$$\alpha_j = \mathbf{q}_j^T\mathbf{A}\mathbf{q}_j,$$
$$\mathbf{q}_{j+1}\beta_{j+1} = \mathbf{f}_j,$$
$$\mathbf{q}_i^T\mathbf{f}_j = 0, \quad i = 1, \ldots, j$$

and so $\mathbf{q}_{j+1} = \mathbf{f}_j\beta_{j+1}^{-1}$, where we assume that β_{j+1} is non-zero. We define a *Lanczos iteration* to be that computing $\mathbf{A}\mathbf{q}_j, \alpha_j, \beta_{j+1}$, and $\mathbf{f}_j$. We define the *Lanczos method* that of computing m iterations and computing the largest in magnitude eigenvalue of $\mathbf{T}_m$.

The largest eigenvalue of the symmetric tridiagonal matrix $\mathbf{T}_j$ approximates the largest in magnitude eigenvalue of $\mathbf{A}$. We can determine the quality of the approximation produced by an eigenpair of $\mathbf{T}_j$. If we post multiply ((12.35)) by $\mathbf{s}$ where $\mathbf{T}_j\mathbf{s} = \mathbf{s}\theta$ (and $\|\mathbf{s}\| = 1$), then

$$\mathbf{A}(\mathbf{Q}_j\mathbf{s}) - (\mathbf{Q}_j\mathbf{s})\theta = \mathbf{f}_j(\mathbf{e}_j^T\mathbf{s}). \tag{12.37}$$

In words, the residual of the approximate eigenpair $(\mathbf{Q}_j\mathbf{s}, \theta)$ is proportional to $\mathbf{f}_j$ (note that $\mathbf{e}_j^T\mathbf{s}$ is notation for the last component of $\mathbf{s}$). The implication is that we can easily monitor the quality of the approximation produced by the Lanczos method. If θ is the largest in magnitude eigenvalue of $\mathbf{T}_j$, then $\theta \leq \omega_{\max}^2 \leq \|\mathbf{f}_j\|_2 |\mathbf{e}_j^T\mathbf{s}| + \theta$ (see [44] for a discussion). Hence,

$$\frac{1}{\|\mathbf{f}_j\|_2 |\mathbf{e}_j^T\mathbf{s}| + \theta} \leq \omega_{\max}^{-2} \leq \frac{1}{\theta}. \tag{12.38}$$

We also remark that the norm of the residual is a non-increasing function of j; again see [44].

The Lanczos iteration is adapted for computing the largest eigenvalues of ((12.32)) by replacing $\mathbf{A}$ with $\mathbf{M}^{-1}\mathbf{K}$ and computing an $\mathbf{M}$-orthonormal $\mathbf{Q}_j$. This orthonormality is needed so that $\mathbf{M}^{-1}\mathbf{K}$ is symmetric in the inner product induced by $\mathbf{M}$. See [42], [44] for further discussion and implementations.

The cost of a careful implementation of a Lanczos iteration, $j>1$, is one matrix-vector product with $\mathbf{K}$ and $\mathbf{M}^{-1}$, and two vector products and vector subtractions. Within an explicit dynamics code, the cost of computing a Lanczos vector is approximately the cost of an internal force calculation, represented by the matrix-vector product $\mathbf{K}\mathbf{q}_j$. Therefore, we approximate the cost of computing the Lanczos-based time step estimate as

$$m\tau \tag{12.39}$$

where m denotes the number of Lanczos iterations and τ represents the CPU (central processor unit) time needed for an element-based explicit dynamics time integration step.

The Lanczos method only requires knowledge of $\mathbf{K}$ via its application on a vector. If internal force calculations are used for the needed matrix-vector products, the Lanczos vectors $\mathbf{q}_j$ are scaled so that they represent velocities associated with small strain. When these scaled vectors are sent to the internal force calculation, the internal force calculation becomes a matrix-vector product with a (constant) tangent stiffness matrix $\mathbf{K}_T$.

12.3.3.4 Cost-Benefit Analysis

This section provides a simple model for assessing the cost of using the Lanczos method for computing an estimate of the critical time step. We assume that Lanczos-based time step is valid for n_L time integration steps. We address the important issue of the adapting the time step when we present the details for practical use of the Lanczos method in a subsequent section.

Denote by Δt_L and Δt_e the time steps estimate of the critical time step computed by the Lanczos and element-based methods, where the ratio ρ of Δt_L to Δt_e is at least as large as one because of

((12.34)). After n_L time steps, the dynamics simulation is advanced in time $n_L \Delta t_L$. Let n_e be the number of element-based time steps so that $n_e \Delta t_e \leq n_L \Delta t_L < (n_e + 1)\Delta t_e$. In terms of ρ, we have the relationship

$$n_e \leq \rho n_L < n_e + 1, \tag{12.40}$$

so bounding the number of Lanczos-based explicit integration steps in terms of ρ and the number of element-based integration steps.

Let us examine the computational costs in terms of CPU time in performing the above n_L and n_e integration steps. Denote by τ the CPU time for an element-based time integration step and assume that it is dominated by the cost of an internal force calculation. Using ((12.40)), the CPU time of n_L time integration steps is

$$(n_L + m)\tau, \tag{12.41}$$

and the CPU time of n_e time integration steps is $n_e \tau$. Equating these two CPU times, determines when the cost of both approaches is equivalent and results in the relationship

$$\hat{n}_e = m + \hat{n}_L. \tag{12.42}$$

Using ((12.42)) within ((12.40)) gives

$$\frac{m}{\rho - 1} \leq \hat{n}_L < \frac{m + 1}{\rho - 1} \tag{12.43}$$

so bounding the minimum number of Lanczos-based time integration steps in terms of the number of Lanczos iterations and ρ so that the cost of the computing the Lanczos-based time step is amortized.

Our cost benefit analysis provides theb reak-even point at which the Lanczos method becomes cost-effective by overcoming the associated overhead. For example, let $\rho = 1.25$ and $m = 20$ so that $\hat{n}_L$ is bounded from below by 80, and by ((12.41)) $\hat{n}_e = 100$. Hence, the time integration with the Lanczos-based and element-based estimates of the critical time step give the same simulation time for the same CPU time. If we use the Lanczos-based time step Δt_L for more than 80 time integration steps, then the Lanczos-based approach is cost-effective.

A Lanczos-based critical time estimate is cost effective if m is small and ρ is not close to one. The size of m is dependent upon the ability of the Lanczos method to rapidly provide an accurate approximation to $\omega_{\max}^2$. If ρ approaches one, then the Lanczos-based critical time step approaches the element-based critical time step, implying that $\hat{n}_L$ must increase to offset the cost of the m Lanczos iterations. Section Section 12.3.3.6 demonstrate that m is small and that ρ is not close to one for realistic problems.

Our section ends by considering the additional cost involved with contact. The addition of contact to an analysis can add computational costs to a time step that are as large as or larger than the internal force calculations. Therefore, for an analysis with contact, running at a larger time step than the element-based estimate can have an even greater impact on reducing CPU time for an analysis.

The above analysis is easily extended to the case where we have contact. If the CPU time of contact over a time step is some multiple γ of τ, then in analogy to ((12.42)) and ((12.43)), we have

$$(1 + \gamma)\hat{n}_e = m + (1 + \gamma)\hat{n}_L,$$

and

$$\frac{m}{(\rho - 1)(1 + \gamma)} \leq \hat{n}_L < \frac{m + 1 + \gamma}{(\rho - 1)(1 + \gamma)}$$

Again, for example, let $\rho = 1.25$ and $m = 20$ and assume the computational cost of contact calculations is the same as an internal force calculation so that $\gamma = 1$. Hence, the break-even point is $\hat{n}_L = 40$ and $\hat{n}_e = 50$. The additional cost of the contact calculations within the time integration reduces the break-even point over that with no contact ($\gamma = 0$).

12.3.3.5 Using the Lanczos-based estimate

The previous section shows how the repeated use of a Lanczos-based time step estimate could be cost-effective within an explicit transient dynamics simulation. This section presents an adaptive scheme that combines the Lanczos-based estimate with an element-based estimate of the critical times-step over a number of explicit time integration steps.

Section 12.3.3.2 explained that the Lanczos method provides an approximation to the maximum eigenvalue of (12.32) from below so overestimating the critical time step. Therefore, we scale back the Lanczos-based time. The scheme to determine a scaled-back value employs the element-based time step estimate. Again, let Δt_L and Δt_e be the time steps computed by the Lanczos and element-based methods. The scaled back estimate for the critical time step, Δt_s, is computed from the equation

$$\Delta t_s = \Delta t_e + f_s(\Delta t_L - \Delta t_e),$$

where f_s is a scale factor. (The value for f_s ranges from 0.9 to 0.95 for our problems—a rigorous estimate can be made by using ((12.38)).) This value of f_s results in Δt_s close to and slightly less than the critical time step. Once Δt_s is determined, the ratio

$$t_r = \frac{\Delta t_s}{\Delta t_e}$$

is computed. This ratio is then used to scale subsequent element-based estimates for the critical time step. If $\Delta t_{e(n)}$ is the nth element-based time step after the time step where the Lanczos method is computed, then the nth time step computed is

$$\Delta t_{(n)} = t_r \Delta t_{e(n)}.$$

The ratio t_r is used until the next call to the Lanczos method. The next call to the Lanczos method is controlled by one of two mechanisms. First, the user can set the frequency with which the

Lanczos method is called. The user can set a parameter so that the Lanczos method is called only once every n time steps. This number remains fixed throughout an analysis. Second, the user can control when the Lanczos method is called based on changes in the element-based time step. For this second method, the change in the element-based critical time step estimate is tracked. At the nth step after the call to the Lanczos iteration, the element-based time step is $\Delta t_{e(n)}$. If the value

$$\frac{|\Delta t_{e(n)} - \Delta t_e|}{\Delta t_e}$$

is greater than some limit set by the user, then the Lanczos method is called. If there is a small, monotonic change in the element-based time step over a large number of time integration steps, this second mechanism will result in the Lanczos method being computed. If there is a large, monotonic change in the element-based critical time step over a few time steps, the Lanczos method will also be called.

These two mechanisms for calling the Lanczos method may be combined resulting in an adaptive scheme for estimating the critical time step during an explicit transient dynamics simulation. For example, suppose the second mechanism, the mechanism based on a change in the element-based time step, results in a call to the Lanczos method. This resets the counter for the first mechanism, the mechanism using a set number of time steps between calls to the Lanczos iteration.

12.3.3.6 Numerical experiments

This method for reusing a Lanczos-based time step estimate has been implemented in Presto [31], and employed within a number of explicit dynamics simulations. We discuss several of these examples.

Example one: The Lanczos method has been used to obtain a critical time step estimate for a cubic block consisting solely of cubic elements—a $10 \times 10 \times 10$ mesh of eight-node hexahedral elements. We know that, for a cubic eight-node hexahedral element, the element-based estimate is conservative by a factor of $1/\sqrt{3}$. The Lanczos method yields a critical time estimate for this mesh that is $\rho = \sqrt{3}$ (approximately 1.732) times larger than the element-based estimate. This is done by using 20 Lanczos vectors.

Example two: Critical time step estimates were made for two mechanical systems. The systems consisted of cylindrical metal cans containing a variety of components. Some of these components have relatively simple geometries, while other components have complex shapes. A number of the components with complex shapes are a foam material used to absorb impact loads. One component was modeled with approximately 250,000 degrees of freedom, and the other one was modeled with approximately 350,000 degrees of freedom. For both of these models, a good estimate for the maximum eigenvalue was obtained with the Lanczos method by computing only twenty Lanczos vectors. For the model with 250,000 degrees of freedom, an actual analysis was run. The value for ρ for this problem was 1.83. The break-even point for this case ($n_L = 20$ and $\rho = 1.83$) is $n_e = 45$. It was possible to use the same scale factor for 1700 time steps for this analysis, which is well above the break-even point. The extended use of the Lanczos based estimate reduced the computation cost by over 56%.

Example three: A study of a large-scale model involving 1.7 million nodes (5.1 million degrees of freedom) showed that only 45 Lanczos vectors were required to obtain a good estimate of the maximum eigenvalue. The value of ρ for this problems was 1.2. Use of this Lanczos based estimated for this problem would be extremely cost-effective.

12.3.3.7 Conclusions

The Lanczos method is cost-effective for estimating the critical time step in an explicit, transient dynamics code. The Lanczos method can give a significantly larger estimate for the critical time-step than an element-based method (the typical scheme). The adaptive hybrid scheme synthesizes the Lanczos-based and element-based estimates and allows us to run near the critical time-step estimate provided by the Lanczos method.

Not all problems will lend themselves reuse of one Lanczos-based estimate for thousands of time steps. However, if it is possible to use the Lanczos-based estimate for two to three times the number of time steps required for break-even, we begin to see a noticeable reduction in the total CPU time required for a problem.

In addition, to the increased efficiency we can achieve with the Lanczos iteration, we also have the added benefit of increased accuracy. For explicit transient dynamic codes, using a time step as close as possible to the critical time gives the most accurate answer. Reducing the time step in an explicit transient dynamics code actually increases the error.

12.4 Implicit Finite Element Methods

To introduce the concept of an implicit time finite element method, we examine the trapezoidal rule, which is simply the member of the Newmark family obtained by setting $\alpha = 0$, $\beta = 1/4$, and $\gamma = 1/2$. Substitution of these values into (12.6) yields

$$\mathbf{M}\mathbf{a}_{n+1} + \mathbf{F}^{\text{int}}(\mathbf{d}_{n+1}) = \mathbf{F}^{\text{ext}}(\mathbf{d}_{n+1})$$

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \Delta t \mathbf{v}_n + \frac{\Delta t^2}{4} \left[\mathbf{a}_n + \mathbf{a}_{n+1} \right] \tag{12.44}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \frac{\Delta t}{2} \left[\mathbf{a}_n + \mathbf{a}_{n+1} \right].$$

Insight into this method can be obtained by combining the first two equations in (12.44) and solving for $\mathbf{d}_{n+1}$ to get

$$\frac{4}{\Delta t^2}\mathbf{M}\mathbf{d}_{n+1} + \mathbf{F}^{\text{int}}(\mathbf{d}_{n+1}) = \mathbf{F}^{\text{ext}}(\mathbf{d}_{n+1}) + \mathbf{M}\left(\mathbf{a}_n + \Delta t \mathbf{v}_n + \frac{4}{\Delta t^2}\mathbf{d}_n \right)$$

$$\mathbf{a}_{n+1} = \frac{4}{\Delta t^2}\left(\mathbf{d}_{n+1} - \mathbf{d}_n \right) - \frac{4}{\Delta t}\mathbf{v}_n - \mathbf{a}_n \tag{12.45}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \frac{\Delta t}{2}\left[\mathbf{a}_n + \mathbf{a}_{n+1} \right]$$

Solving the first equation 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 non-linear in general due to the internal force vector.

Note that we can write the first equation of (12.45) in terms of a dynamic incremental residual $\mathbf{r}_{n+1}$ via

$$\mathbf{r}(\mathbf{d}_{n+1}) = \left[\mathbf{F}^{\text{ext}}(\mathbf{d}_{n+1}) + \mathbf{M}\left(\mathbf{a}_n + \Delta t \mathbf{v}_n + \frac{4}{\Delta t^2}\mathbf{d}_n\right) - \left(\frac{4}{\Delta t^2}\mathbf{M}\mathbf{d}_{n+1} + \mathbf{F}^{\text{int}}(\mathbf{d}_{n+1})\right) \right] = 0 \qquad (12.46)$$

This system has the same form as (11.10), which suggests that the same sort of nonlinear solution strategies are needed for implicit dynamic calculations as in quasistatics (Section 11). Equation solving is the topic of the next chapter, where we will discuss at some length the techniques used to solve (11.9) and (12.46) in Sierra/SolidMechanics, particularly for parallel computing.

This page left blank

13 Nonlinear Equation Solving

13.1 Introduction

This chapter discusses non-linear equation solving methods, specifically the use of iterative algorithms for problems in solid mechanics. Although some of this work has taken place over many years at Sandia National Labs and elsewhere, recent efforts have significantly added to the functionality and robustness of these algorithms. This chapter primarily documents these recent efforts. Some historical development is covered for context and completeness, hopefully showing a complete picture of the current status of iterative solution algorithms for nonlinear solid mechanics in Sierra/SolidMechanics.

Iterative algorithms have seen somewhat of a resurgent interest, possibly due to the advancement of parallel computing platforms. Increases in computational speed and available memory have raised expectations on model fidelity and problem size. Increased problem size has sparked interest in iterative solvers because the direct solution strategy becomes increasingly inefficient as problem size grows. A traditional implicit global solution strategy is typically based on Newton's method, generating fully coupled linearized equations that are often solved using a direct method. In many applications in solid mechanics this procedure poses no particular problem for modern computing platforms with sufficient memory. However, for large three-dimensional models of interest, the cost of direct equation solving becomes prohibitive on any computer, except for the largest supercomputers. This motivates the use of iterative solution strategies that do not require the direct solution of linearized global equations.

Application of purely iterative solvers to the broad, general area of nonlinear finite element solid mechanics problems has seen only modest success. Certain classes of problems have remained notoriously difficult to solve. Examples of these include problems that are strongly geometrically nonlinear, problems with nearly incompressible material response, and problems with frictional sliding. Thus, much of this chapter is devoted to examining and discussing an implementation of a multi-level solution strategy, where the nonlinear iterative solver is asked to solve simplified **model problems** from which the real solution to these difficult problems is accumulated. This strategy has greatly contributed to the functionality and robustness of the nonlinear iterative solver.

13.2 The Residual

$$\mathbf{r}\left(\mathbf{d}_{n+1}\right) = \mathbf{F}^{\text{ext}}\left(\mathbf{d}_{n+1}\right) - \mathbf{F}^{\text{int}}\left(\mathbf{d}_{n+1}\right) = \mathbf{0} \tag{13.1}$$

and the implicit dynamics problem using the trapezoidal time integration rule, (12.46), is written as

$$\mathbf{r}(\mathbf{d}_{n+1}) = \left[\mathbf{F}^{\text{ext}}(\mathbf{d}_{n+1}) + \mathbf{M}\left(\mathbf{a}_n + \Delta t \ \mathbf{v}_n + \frac{4}{\Delta t^2}\mathbf{d}_n\right) - \left(\frac{4}{\Delta t^2}\mathbf{M}\mathbf{d}_{n+1} + \mathbf{F}^{\text{int}}(\mathbf{d}_{n+1})\right)\right] = \mathbf{0}. \tag{13.2}$$

In either case, the equation to be solved takes the form

$$\mathbf{r}\left(\mathbf{d}_{n+1}\right) = \mathbf{0}, \tag{13.3}$$

where the residual $\mathbf{r}\left(\mathbf{d}_{n+1}\right)$ is, in general, a nonlinear function of the solution vector $\mathbf{d}_{n+1}$. This form allows us to consider the topic of nonlinear equation solving in its most general form, with the introduction of iterations, $j = 0, 1, 2, ...$, as

$$\mathbf{r}\left(\left(\mathbf{d}_{n+1}\right)_j\right) = \mathbf{0}, \tag{13.4}$$

or simply

$$\mathbf{r}_j = \mathbf{r}\left(\mathbf{d}_j\right) = \mathbf{0}. \tag{13.5}$$

For implicit dynamic Sierra/SM simulations, each load step from time n to $n + 1$ requires a new nonlinear solve with sub-iterations $j = 0, 1, 2, ...$ Here we have omitted the references to the load step, yet it is understood that, e.g., $\mathbf{d}_j$ is at $n + 1$.

We can rewrite (13.1) and (13.2) as

$$\mathbf{r}_j = \mathbf{F}_j^{\text{ext}} - \mathbf{F}_j^{\text{int}} = \mathbf{0} \tag{13.6}$$

and

$$\mathbf{r}_j = \mathbf{F}_j^{\text{ext}} - \frac{4}{\Delta t^2}\mathbf{M}\mathbf{d}_j - \mathbf{F}_j^{\text{int}} + \tilde{\mathbf{F}} = \mathbf{0} \tag{13.7}$$

where $\mathbf{F}_j^{\text{int}} = \mathbf{F}^{\text{int}}\left(\mathbf{d}_j\right) = \mathbf{F}^{\text{int}}\left(\left(\mathbf{d}_{n+1}\right)_j\right)$ and $\tilde{\mathbf{F}}$ is the constant portion of the residual, defined as

$$\tilde{\mathbf{F}} = \mathbf{M}\left(\mathbf{a}_n + \Delta t\ \mathbf{v}_n + \frac{4}{\Delta t^2}\mathbf{d}_n\right).$$

The task for any nonlinear equation solution technique is to improve the iterate (or guess) for the solution vector $\mathbf{d}_j$ such that the residual $\mathbf{r}_j$ is close enough to $\mathbf{0}$. How that is done depends on the method employed.

Fig. 13.1 depicts a generalized nonlinear loadstep solution with solution iterates $j = 0, 1, 2, ..., j^*$, where the iterates converge when $\left\|\mathbf{r}_{j^*}\right\| \approx 0$ at iteration j^*.

In (a) of Fig. 13.1, the solution starts with iterate $\mathbf{d}_0$ taken as the solution of the prior load step from $n - 1$ to n. (Note that the zero iterate is not always taken to be the prior solution. See Section 13.6.4 on predictors for more details.) Iterate $\mathbf{d}_0$ results in a residual of $\mathbf{r}_0$, which then informs the next iterate $\mathbf{d}_1$ such that $\|\mathbf{r}_1\| < \|\mathbf{r}_0\|$. For details on how $\mathbf{d}_1$ is formed, see the following Sections Section 13.3 through Section 13.8. In (b) and (c) this procedure from iteration j to $j + 1$ is depicted, and in (d) the solution procedure has converged at iteration j^* with iterate $\mathbf{d}_{j^*}$. The load step from n to $n + 1$ is then solved, and the solution procedure for the next load step from $n + 1$ to $n + 2$ starts over in (a).

Fig. 13.1 Graphical depiction of nonlinear iterations.

In the figure above:

Panel (b) Initial iteration $j=0$:

$$F_0^{ext} \equiv F_{j=0}^{ext}\Big|_{n+1} = F^{ext}\Big|_n + \Delta F^{ext}\Big|_{n+1}$$

finish

$$r_0 = F_0^{ext} - F_0^{int}$$

start

$$F_0^{int} \equiv F_{j=0}^{int}\Big|_{n+1} = F^{int}\Big|_n$$

$$d_0 = d_{j=0}^{n+1} = d^n \qquad d_{j^*} = d_{j=j^*}^{n+1}$$

Panel (b) iteration j:

F_0^{ext} $\qquad F_j^{ext}$

F_j^{int}

$$r_j = F_j^{ext} - F_j^{int}$$

F_0^{int}

$$d_0 = d_{j=0}^{n+1} \qquad d_j \qquad d_{j^*} = d_{j=j^*}^{n+1}$$

Panel (c) iteration $j+1$:

F_j^{ext} $\quad F_{j+1}^{ext}$

r_{j+1}

F_j^{int} $\quad F_{j+1}^{int}$

$$F_0^{int} \equiv F_{j=0}^{int}\Big|_{n+1} = F^{int}\Big|_n$$

$$d_0 = d_{j=0}^{n+1} \qquad d_j \; d_{j+1} \qquad d_{j^*} = d_{j=j^*}^{n+1}$$
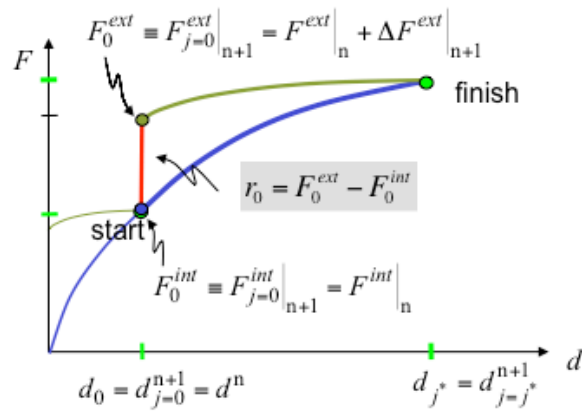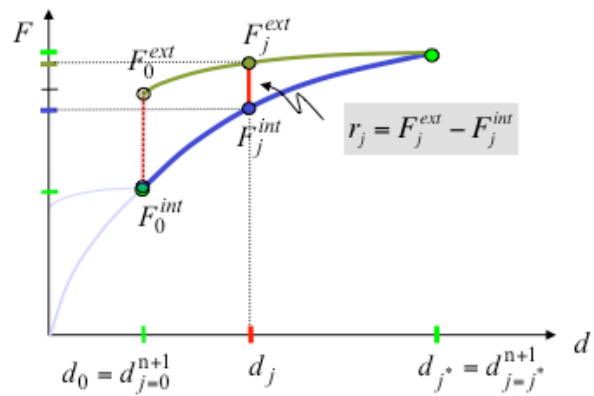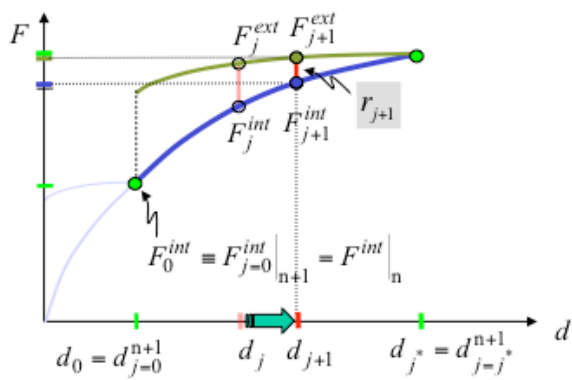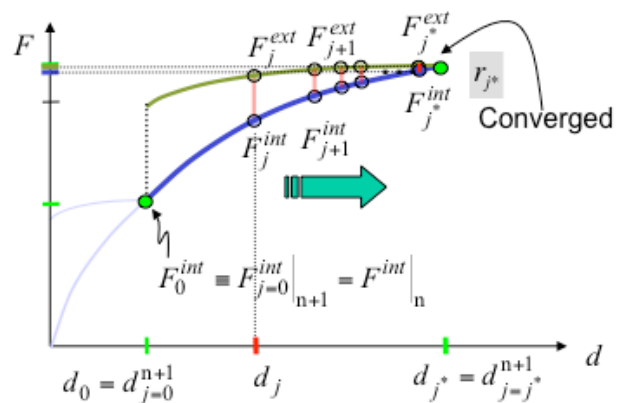
Panel (d) Convergence at iteration j^*:

F_j^{ext} $\; F_{j+1}^{ext}$ $\; F_{*}^{ext}$

r_{j^*}

$F_{j^*}^{int}$

Converged

F_j^{int} $\; F_{j+1}^{int}$

$$F_0^{int} \equiv F_{j=0}^{int}\Big|_{n+1} = F^{int}\Big|_n$$

$$d_0 = d_{j=0}^{n+1} \qquad d_j \qquad d_{j^*} = d_{j=j^*}^{n+1}$$

13.3 Gradient Property of the Residual

The residual has the very important property that it points in the steepest descent or gradient direction of the function f:

$$f\left(\mathbf{d}_j\right) = \frac{1}{2}\left(\mathbf{d}_j - \mathbf{d}^*\right)^T \mathbf{r}(\mathbf{d}_j), \tag{13.8}$$

which is the **energy error of the residual**. Solving for $\mathbf{d}_j = \mathbf{d}^*$ is equivalent to minimizing the energy error of the residual, $f\left(\mathbf{d}_j\right)$.

The importance of this property can not be overemphasized. Any iterative solver makes use of it in some way or another. Even though the solution $\mathbf{d}^*$ is not known, a non-zero residual points the way to improving the guess. Mathematically, our nonlinear solid mechanics problem looks like a minimization problem discussed at length in the optimization literature, see e.g. [36]. It is from this viewpoint that the remainder of the nonlinear solution methods will be discussed. The concept of the energy error of the residual reveals important physical insights into how iterative algorithms are expected to perform on particular classes of problems.

An example of the energy error of the residual providing physical insight into a problem is demonstrated in Fig. 13.2.

Two beams, one thick and one thin, are subjected to a uniform pressure load causing a downward deflection to the equilibrium point (d_1, d_2) indicated by the blue dot. If we think of *modes of deformation* rather than the nodal degrees of freedom (d_1, d_2), two modes of deformation come to mind: a bending mode and an axial mode.

For the thick beam in Fig. 13.3, the red dashed line is the locus of points (d_1, d_2) that induce only bending stresses in the beam and is therefore called a bending mode. In contrast, the blue dashed line is the locus of points (d_1, d_2) that induce only axial stresses in the beam and is therefore called an axial mode. These bending and axial modes are characterized by the eigenvectors q_b and q_a, respectively.

Eigenvectors are typically written as linear combinations of the nodal degrees of freedom. The bending modes, for example, can be written as $q_b = a_1 d_1 + a_2 d_2$. However, since we are dealing with a nonlinear problem in our simple example (and in general), the coefficients a_1 and a_2 vary with the deformation of the beam - which is precisely why the dashed red line is curved. The energy error contours can thus be displayed, as shown in Fig. 13.4. Any displacement away from the equilibrium point $(d_1, d_2)^*$ produces a nonzero residual and consequently requires work.

Now we compare moving the tip of the beam along the red dashed line, which invokes a bending mode of deformation of the beam versus moving the tip along the blue dashed line, which invokes an axial mode of deformation. The larger modal stiffness (eigenvalue) corresponding to the axial mode induces a greater energy penalty for a given amount of displacement compared to the bending mode. This produces the stretched energy contours shown. Since the ratio of stiffness between the axial and bending modes is much larger for the thin beam than the thick beam, the stretching of the energy error contours is more pronounced for the thin beam. Mathematically, these contours are a graphical representation of the gradient of the residual, $\nabla \mathbf{r}(\mathbf{d})$.

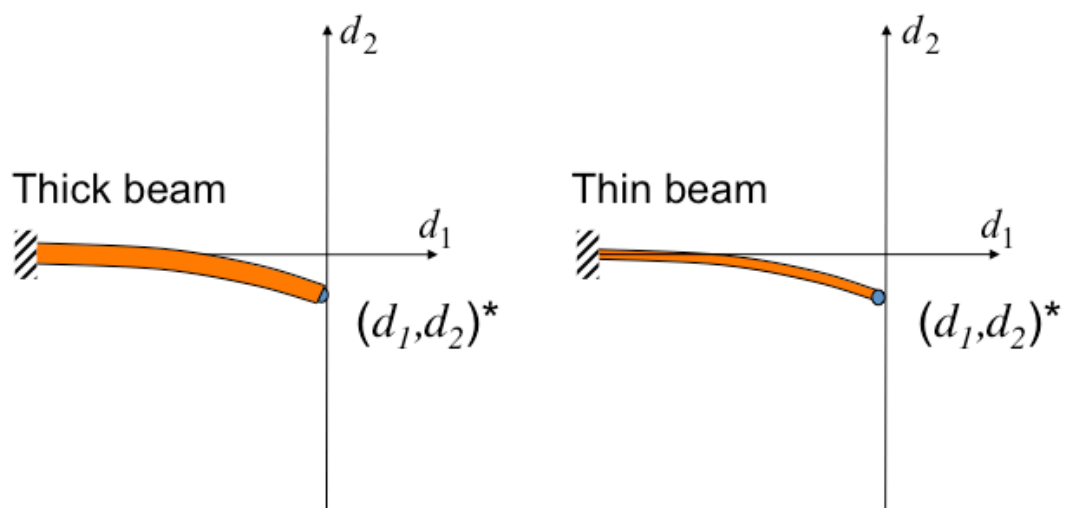**Fig. 13.2 Energy error example: two beams with large and small x-sectional moments of inertia.**
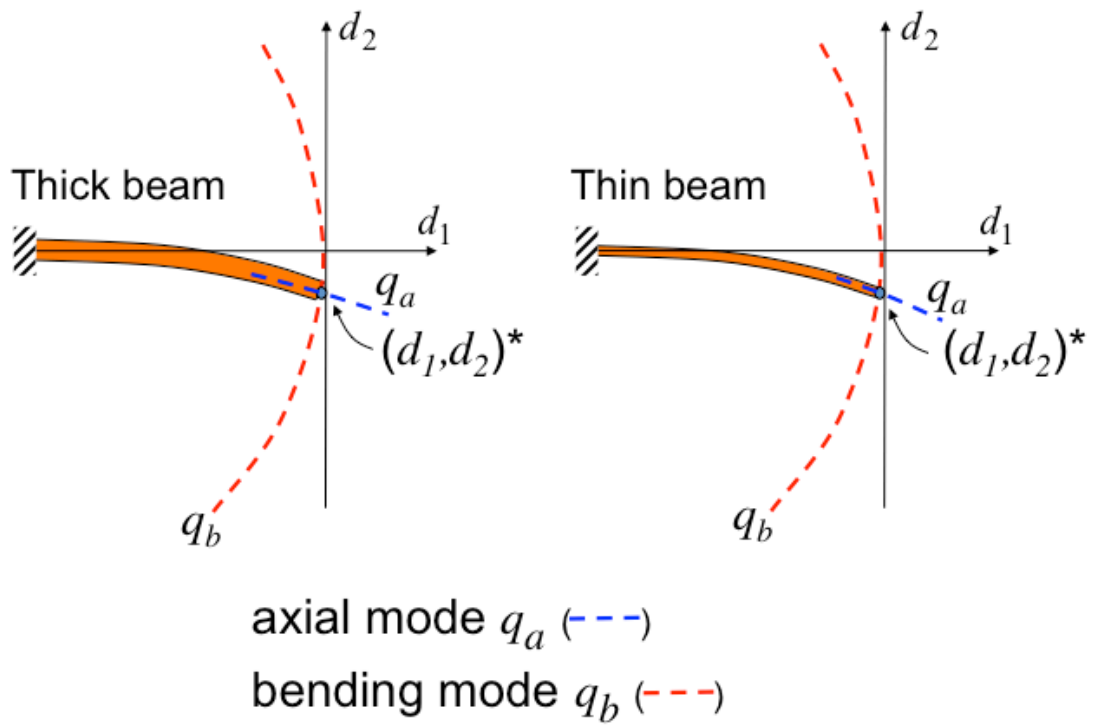
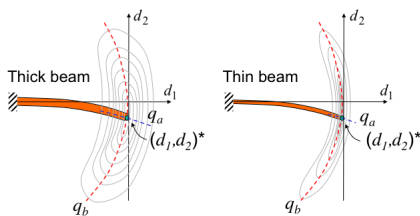**Fig. 13.3 Energy error example: modes of deformation for two beams.**

Fig. 13.4 Energy error example: energy error contours for two beams.

The beam example is chosen for its simplicity, however it also poses a non-trivial nonlinear problem. Experience has shown that the thinner the beam becomes the more difficult it is to solve. In fact, convergence investigations reveal that it is the ratio of maximum to minimum eigenvalue of $\nabla \mathbf{r}(\mathbf{d})$ that is critical to the performance of iterative methods.

13.4 Newton's Method for Solving Nonlinear Equations

In this context, the idea embodied in classical Newton's method is simple. Substituting the nonlinear residual $\mathbf{r}(\mathbf{d}_j)$ with the local tangent approximation $\mathbf{y}(\mathbf{d})$ gives

$$\mathbf{y}(\mathbf{d}) = \mathbf{r}(\mathbf{d}_j) + \nabla \mathbf{r}(\mathbf{d}_j)(\mathbf{d} - \mathbf{d}_j), \tag{13.9}$$

which is linear in the vector of unknowns ($\mathbf{d}$). Solving (13.9) (solving for $\mathbf{y}(\mathbf{d}) = \mathbf{y}(\mathbf{d}_{j+1}) = \mathbf{0}$) gives the iterative update for Newton's method,

$$\mathbf{d}_{j+1} = \mathbf{d}_j - \nabla \mathbf{r}^{-1}(\mathbf{d}_j)\mathbf{r}(\mathbf{d}_j). \tag{13.10}$$

The structural mechanics community commonly refers to the **tangent stiffness matrix** in the context of geometrically nonlinear problems. Based on (13.9), the tangent stiffness matrix arises from the tangent approximation of $f(\mathbf{d}_j)$:

$$\mathbf{K}_T = \nabla \mathbf{r}(\mathbf{d}_j). \tag{13.11}$$

Then (13.10) can be written as

$$\mathbf{d}_{j+1} = \mathbf{d}_j - [\mathbf{K}_T]^{-1}\, \mathbf{r}(\mathbf{d}_j), \tag{13.12}$$

the solution of which requires the inverse of $\mathbf{K}_T$.

A conceptual view of Newton's method applied to our two beam example is shown in Fig. 13.5.

Newton's method is generally considered to be the most robust of the nonlinear equation solution techniques, albeit at the cost of generating the tangent stiffness matrix (13.11) and solving the *linear* system of equations with n_{dof} unknowns:

$$[\mathbf{K}_T]\left(\mathbf{d}_{j+1} - \mathbf{d}_j\right) = -\mathbf{r}(\mathbf{d}_j). \tag{13.13}$$

There are a number of linear equation solution techniques available, and Sierra/SolidMechanicshas the ability to apply a linear equation solution approach available in the FETI library (discussed briefly in section Section 13.7).

As mentioned, Newton's method relies on computing the tangent stiffness matrix which, by examination of (13.14), requires the partial derivatives (with respect to the unknowns) of the external and internal force vector,

$$\mathbf{K}_T = \frac{\partial}{\partial \mathbf{d}}\left[\mathbf{F}_j^{\text{ext}} - \mathbf{F}_j^{\text{int}}\right]. \tag{13.14}$$
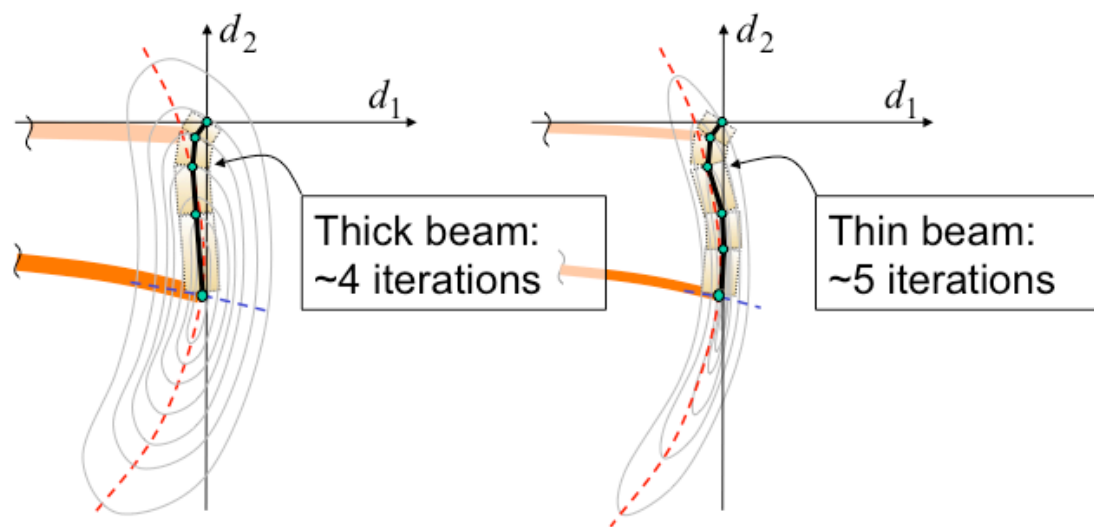
Fig. 13.5 Energy error example: Newton's method applied to two beams.

In practice, for all but the simplest of material models, the exact tangent cannot be computed. Thus Sierra/SolidMechanics computes a secant approximation with the property

$$\mathbf{K}_{\tilde{T}} \cdot (\delta \mathbf{d}) = \mathbf{r}(\mathbf{d}_j + \delta \mathbf{d}) - \mathbf{r}(\mathbf{d}_j) \tag{13.15}$$

by simply probing the nonlinear system via perturbations $\delta \mathbf{d}$. In (13.15), the notation $\mathbf{K}_{\tilde{T}}$ is used to indicate that the probed tangent is an approximation of the exact tangent.

13.5 Steepest Descent Method

As mentioned in section Section 13.3, the steepest descent iteration takes steps in the direction of the gradient of the energy error of the residual. On its own, it would not be considered a viable solver for solid mechanics because of its general lack of performance compared to Newton-based methods. However, there are algorithmic elements of this method that are conceptually important for understanding nonlinear iterative solver such as the method of conjugate gradients, and in fact are used in their construction.

The idea behind the steepest descent method is to construct a sequence of search directions, $\mathbf{s}_j$,

$$\mathbf{s}_j = \mathbf{M}^{-1}\mathbf{g}_j = -\mathbf{M}^{-1}\mathbf{r}(\mathbf{d}_j), \tag{13.16}$$

in which the energy decreases, thus producing a new guess of the solution vector

$$\mathbf{d}_{j+1} = \mathbf{d}_j + \alpha \mathbf{s}_j. \tag{13.17}$$

The minimization is accomplished by taking a step of length α along $\mathbf{s}_j$, where α is called the line search parameter:

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} f(\mathbf{d}_j + \alpha \mathbf{s}_j) \approx \left[\mathbf{r}(\mathbf{d}_j)\right]^T \mathbf{s}_j + \alpha \left[\mathbf{r}(\mathbf{d}_j + \mathbf{s}_j) - \mathbf{r}(\mathbf{d}_j)\right]^T \mathbf{s}_j = 0, \tag{13.18}$$

which, after simplification, gives

$$\alpha = \frac{\left[\mathbf{r}(\mathbf{d}_j)\right]^T \mathbf{s}_j}{\left[\mathbf{r}(\mathbf{d}_j + \mathbf{s}_j) - \mathbf{r}(\mathbf{d}_j)\right]^T \mathbf{s}_j}. \tag{13.19}$$

The **preconditioner** matrix $\mathbf{M}$ is included in (13.16) to accelerate the convergence rate of the steepest descent method. Note that in this case $\mathbf{M}$ is not meant to be the mass matrix.

Fig. 13.6 through Fig. 13.8 all show high aspect ratio ellipses. It turns out that the ideal preconditioner would transform the ellipses to circles; this, in turn, would be $\mathbf{M} = \mathbf{K}_T$. As expected, the ideal steepest descent method is Newton's method. However, the steepest descent framework gives us a way to use approximations of $\mathbf{K}_T$.

A conceptual view of the steepest descent method applied to our two beam example is shown in Fig. 13.6. As indicated in the figure, the thinner beam would require more steepest descent iterations to obtain convergence compared to the thicker beam.
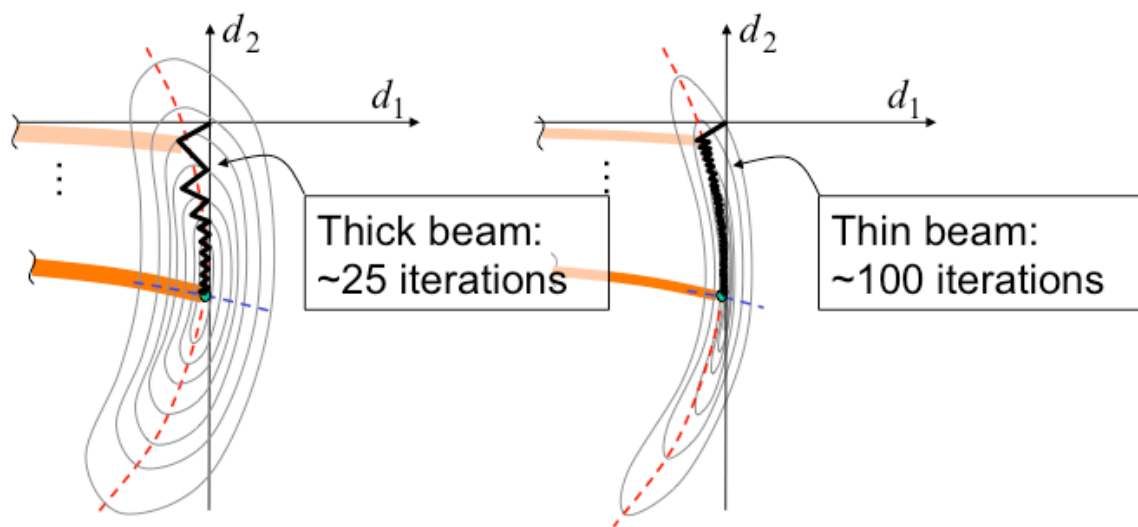
Fig. 13.6 Energy error example: Steepest descent method applied to two beams.

It is instructive to consider whether or not the large number of iterations are due to the nonlinearities in this model problem. For this purpose, we construct the two beam model problem in linearized form. Fig. 13.7 shows the first iteration of the steepest descent method for the linearized problem. The immediate difference seen between the linearized version and the nonlinear problem is in the elliptic form of the energy error contours. However, the contours are still stretched reflecting the relative modal stiffness of the axial and bending modes. Thus, from the same starting point, $d_1 = d_2 = 0$, the initial search direction is composed of different amounts of d_1 and d_2. This is also apparent in all subsequent iterations. Fig. 13.8 shows the completed iterations for both thick and thin beams.
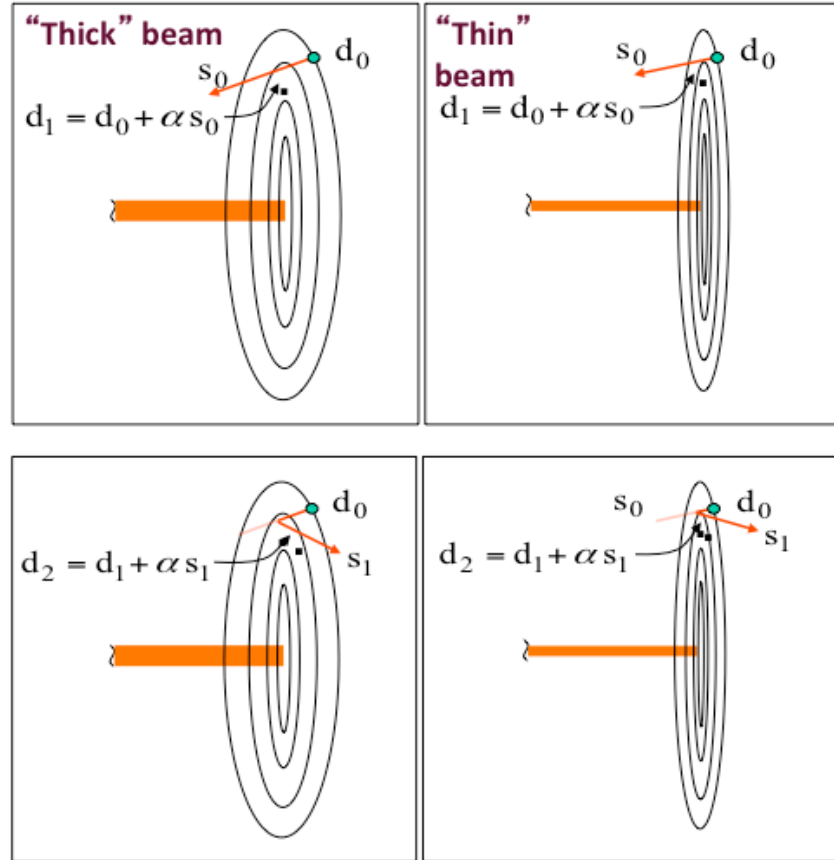
Fig. 13.7 Energy error example: First two iterations of the steepest descent method applied to linearized version of the two beam problem.

Even for the linearized problem, there is a large difference in the number of iterations required for the steepest descent method to converge for the two beams. We can see this because the slope of the search directions is smaller for the thin beam. Therefore, each iteration makes less progress to the solution.

In general, the convergence rate of the steepest descent method is directly related to the spread of the eigenvalues in the problem. In our conceptual beam example, the ratio $\lambda_{max}/\lambda_{min} = \lambda_a/\lambda_b$ (often called the condition number) is larger for the thin beam. It can be shown that in the worst

Fig. 13.8 Energy error example: Steepest descent method applied to linearized version of the two beam problem.

case, the steepest descent iterations reduce the energy error of the residual according to
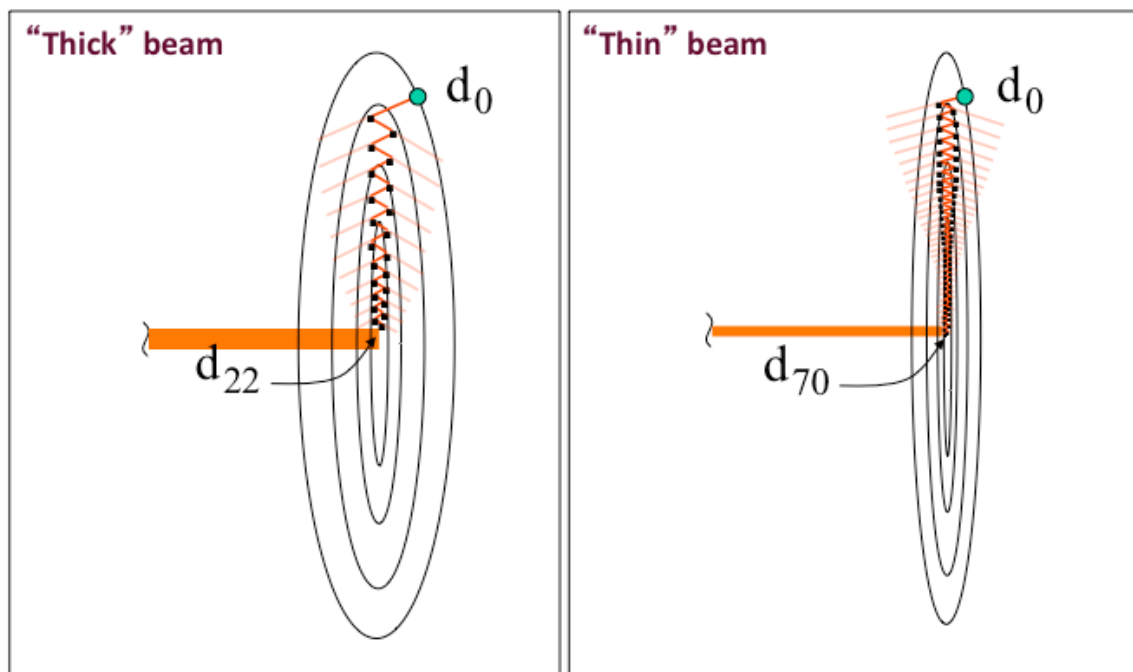
$$f(\mathbf{d}_{j+1}) = \left(\frac{\lambda_{\max}/\lambda_{\min} - 1}{\lambda_{\max}/\lambda_{\min} + 1} \right)^2 f(\mathbf{d}_j). \tag{13.20}$$

13.6 Method of Conjugate Gradients

With the foundation provided by the steepest descent method, application of a conjugate gradient algorithm to (13.6) or (13.7) follows in a straightforward manner. Like the steepest descent method, the important feature the conjugate gradient (or CG) algorithm is that it only needs to compute the nodal residual vectors element by element, and as a result, does not need the large amount of storage required for Newton's method.

The method of conjugate gradients is a well-developed algorithm for solving linear equations. Much of the original work can be found in the articles [7, 8, 14] and the books [23, 41]. A convergence proof of CG with inexact line searches can be found in [19], and a well-presented tutorial of linear CG can be found in [46]. The goal here is to review the method of conjugate gradients to understand the benefits and potential difficulties encountered when applying it to the solution of the nonlinear equations in solid mechanics problems.

13.6.1 Linear CG

The CG algorithm also uses the gradient, $\mathbf{g}_j$, to generate a sequence of search directions $\mathbf{s}_j$ for iterations $j = 1, 2, ...$:

$$\mathbf{s}_j = -\mathbf{M}^{-1}\mathbf{r}(\mathbf{d}_j) + \beta_j \mathbf{s}_{j-1}. \tag{13.21}$$

Note the additional (rightmost) term in (13.21) relative to the steepest descent algorithm of (13.16). The scalar β_j is chosen such that $\mathbf{s}_j$ and $\mathbf{s}_{j-1}$ are $\mathbf{K}$-conjugate; this property is key to the success of the CG algorithm. Vectors $\mathbf{s}_j$ and $\mathbf{s}_{j-1}$ are $\mathbf{K}$-conjugate if

$$\mathbf{s}_j^T \mathbf{K} \mathbf{s}_{j-1} = 0, \tag{13.22}$$

where $\mathbf{K}$ is the stiffness matrix. For a linear problem, $\mathbf{K}$ is a constant positive definite matrix (assuming the internal force of (13.14) is linear). Combining (13.21) and (13.22) gives the following expression for the search direction:

$$\mathbf{s}_j = \frac{\mathbf{g}_j^T \mathbf{K} \mathbf{s}_{j-1}}{\mathbf{s}_{j-1}^T \mathbf{K} \mathbf{s}_{j-1}}. \tag{13.23}$$

Effective progress toward the solution requires minimizing the energy error of the residual along proposed search directions. As with the steepest descent method, the line search performs this

function. Minimizing the energy error of the residual along the search direction occurs where the inner product of the gradient and the search direction is zero:

$$
\begin{aligned}
\mathbf{g}_j^T(\Delta\mathbf{d}_j + \alpha\mathbf{s}_j)\mathbf{s}_j &= \left[\Delta\mathbf{F}^{\text{ext}}(t) - \mathbf{K}\cdot(\Delta\mathbf{d}_j + \alpha\mathbf{s}_j)\right]^T \mathbf{M}^{-1}\mathbf{s}_j \\
&= \left[(\Delta\mathbf{F}^{\text{ext}}(t) - \mathbf{K}\cdot(\Delta\mathbf{d}_j)^T - (\mathbf{K}\cdot\alpha\mathbf{s}_j)^T\right] \mathbf{M}^{-1}\mathbf{s}_j \\
&= \mathbf{g}_j^T\mathbf{s}_j - \alpha_j\mathbf{s}_j^T\mathbf{K}^T\mathbf{M}^{-1}\mathbf{s}_j \\
&= 0.
\end{aligned}
\tag{13.24}
$$

Solving (13.24) gives an exact expression for the line search parameter α,

$$
\alpha_j = \frac{\mathbf{g}_j^T\mathbf{s}_j}{\mathbf{s}_j^T\mathbf{M}^{-1}\mathbf{K}\mathbf{s}_j},
\tag{13.25}
$$

due to the inherent symmetry of $\mathbf{K}$.

The essential feature of the method of conjugate gradients is that once a search direction contributes to the solution, it need never be considered again. As a result, the inner product of the error $\mathbf{e}$ changes from iteration to iteration in the following manner

$$
\mathbf{e}_{j+1}\mathbf{K}\mathbf{e}_{j+1} - \mathbf{e}_j\mathbf{K}\mathbf{e}_j =
$$

$$
\left[\sum_{i=j+1}^{n-1}\delta_i\mathbf{s}_i\right]^T \mathbf{K}\left[\sum_{i=j+1}^{n-1}\delta_i\mathbf{s}_i\right] - \left[\delta_j\mathbf{s}_j + \sum_{i=j+1}^{n-1}\delta_i\mathbf{s}_i\right]^T \mathbf{K}\left[\delta_j\mathbf{s}_j + \sum_{i=j+1}^{n-1}\delta_i\mathbf{s}_i\right]
\tag{13.26}
$$

$$
= -\left[\delta_j\mathbf{s}_j\right]^T \mathbf{K}\left[\delta_j\mathbf{s}_j\right].
$$

Since $\mathbf{K}$ is constant and positive definite, the energy error of the residual decreases monotonically as the iterations proceed. Choosing β_j such that the property in (13.22) holds gives the important result that the sequence of search directions $\mathbf{s}_1, \mathbf{s}_2, \ldots$ spans the solution space in at most n_{eq} iterations. Furthermore, (13.26) reveals that the search directions $\mathbf{s}_1, \mathbf{s}_2, \ldots$ reduce the error in the highest eigenvalue mode shapes first and progressively move to lower ones.

An additional important numerical property of CG is that it can tolerate some inexactness in the line search as discussed in [18] and still maintain its convergence properties.

Applying linear CG to our simple *linearized* beam model problem would generate the comparison depicted in Fig. 13.9. The fact that the linear CG algorithm precisely converges in two iterations demonstrates the significance of the orthogonalization with the previous search direction.

13.6.2 Nonlinear CG

For fully nonlinear problems, where the kinematics of the system are not confined to small strains, the material response is potentially nonlinear and inelastic, and the contact interactions feature potentially large relative motions between surfaces with frictional response, the residual is a function of the unknown configuration at $(n + 1)$, as indicated in (13.1) and (13.2).
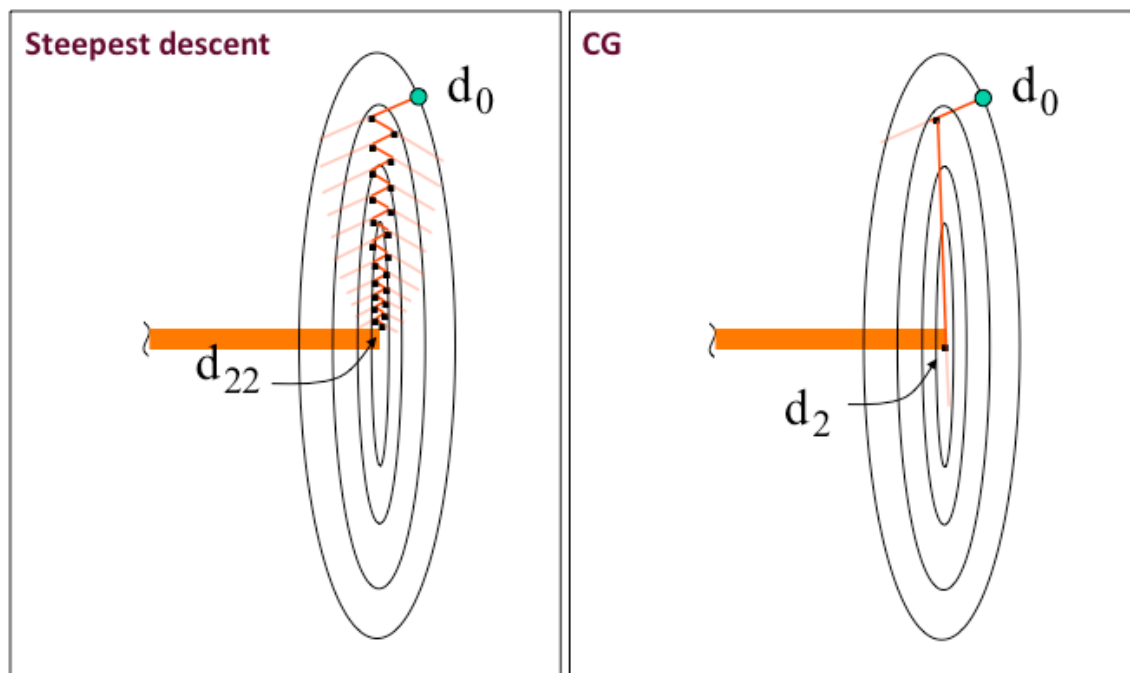
Fig. 13.9 A comparison of steepest descent and linear CG methods applied to the linearized beam example.

Nonetheless, in application of linear CG concepts, it is typical to proceed with the requirement that the new search direction satisfy

$$\mathbf{s}_j^T(\mathbf{g}_j - \mathbf{g}_{j-1}) = 0. \tag{13.27}$$

A comparison of (13.27) and (13.22) reveals that $(\mathbf{g}_j - \mathbf{g}_{j-1})$ can be interpreted to mean the instantaneous representation of $\mathbf{K}^{NL}\mathbf{s}_{j-1}$ to the extent that the incremental solution is known and therefore, how it influences the residual.

Combining (13.21) and (13.27) gives the following result for the search direction

$$\mathbf{s}_j = \beta_j \mathbf{s}_{j-1} - \mathbf{g}_j = \left(\frac{\mathbf{g}_j^T(\mathbf{g}_j - \mathbf{g}_{j-1})}{\mathbf{s}_{j-1}^T(\mathbf{g}_j - \mathbf{g}_{j-1})} \right) \mathbf{s}_{j-1} - \mathbf{g}_j. \tag{13.28}$$

Use of β_j as implied in (13.28) is proposed in the nonlinear CG algorithm in [24]. Alternatives to β_j have also been proposed. For example, simplification of (13.28) is possible if it can be assumed that previous line searches were exact, in which case

$$\mathbf{s}_{j-1}^T\mathbf{g}_j = \mathbf{s}_{j-2}^T\mathbf{g}_{j-1} = 0. \tag{13.29}$$

The orthogonality implied in (13.29) allows the following simplification to the expression for the search direction:

$$\mathbf{s}_j = \left(\frac{\mathbf{g}_j^T(\mathbf{g}_j - \mathbf{g}_{j-1})}{-\mathbf{s}_{j-1}^T\mathbf{g}_{j-1}} \right) = \left(\frac{\mathbf{g}_j^T(\mathbf{g}_j - \mathbf{g}_{j-1})}{-(\beta_{j-1}\mathbf{s}_{j-2} - \mathbf{g}_{j-1})^T\mathbf{g}_{j-1}} \right) = \left(\frac{\mathbf{g}_j^T(\mathbf{g}_j - \mathbf{g}_{j-1})}{\mathbf{g}_{j-1}^T\mathbf{g}_{j-1}} \right). \tag{13.30}$$

Use of the result in (13.30) to define the search directions is recommended in the nonlinear CG algorithm in [19]. The Solid Mechanics code adopts this approach because it has performed better overall. There are, however, instances when the condition implied in (13.29) is not satisfied (due to either highly nonlinear response or significantly approximate line searches).

The orthogonality ratio is computed every iteration to determine the nonlinearity of the problem and/or the inexactness of the *previous* line search. When the orthogonality ratio exceeds a nominal value (default is 0.1), the nonlinear CG algorithm is **reset** by setting

$$\mathbf{s}_j = \mathbf{g}_j. \tag{13.31}$$

We recognize that the line search must be more general to account for potential nonlinearities. Minimizing the gradient $\mathbf{g}^T(\Delta\mathbf{d}_j + \alpha_j\mathbf{s}_j)$ along the search direction $\mathbf{s}_j$ still occurs where their inner product is zero, but an exact expression for α_j can no longer be obtained. A secant method for estimating the rate of change of the gradient along $\mathbf{s}_j$ is employed. Setting the expression to zero will yield the value of α_j that ensures the gradient is orthogonal to the search direction:

$$\frac{d}{d\alpha}\left[\mathbf{g}^T(\mathbf{d}_j + \alpha\mathbf{s}_j)\right]_{\alpha=0}\mathbf{s}_j \approx \mathbf{g}^T(\mathbf{d}_j)\mathbf{s}_j + \alpha_j\mathbf{s}_j^T\left[\frac{d}{d\alpha}[\mathbf{g}^T(\mathbf{d}_j + \alpha\mathbf{s}_j)]_{\alpha=0}\right]\mathbf{s}_j = 0, \tag{13.32}$$

where $\left[\frac{d}{d\alpha}[\mathbf{g}^T(\mathbf{d}_j + \alpha\mathbf{s}_j)]_{\alpha=0}\right]$ is the instantaneous representation of the tangent stiffness matrix. In order to preserve the memory efficient attribute of nonlinear CG, a secant approximation of the tangent stiffness is obtained by evaluating the gradient at distinct points $\alpha = 0$ and $\alpha = \epsilon$,

$$\left[\frac{d}{d\alpha}[\mathbf{g}^T(\mathbf{d}_j + \alpha\mathbf{s}_j)]_{\alpha=0}^{\epsilon}\right] = \frac{1}{\epsilon}\left[\mathbf{g}^T(\mathbf{d}_j + \alpha\mathbf{s}_j)\right]_{\alpha=\epsilon} - \left[\mathbf{g}^T(\mathbf{d}_j + \alpha\mathbf{s}_j)\right]_{\alpha=0}. \tag{13.33}$$

Substituting (13.33) into (13.32) and taking $\epsilon = 1$ yields the following result for the value of the line search parameter α_j:

$$\alpha_j = \frac{-\mathbf{g}^T(\Delta\mathbf{d}_j)\mathbf{s}_j}{\mathbf{g}_T(\Delta\mathbf{d}_j + \mathbf{s}_j) - \mathbf{g}^T(\Delta\mathbf{d}_j)\mathbf{s}_j}. \tag{13.34}$$

Applying nonlinear CG to our simple beam model problem would conceptually generate the iterations depicted in Fig. 13.10.
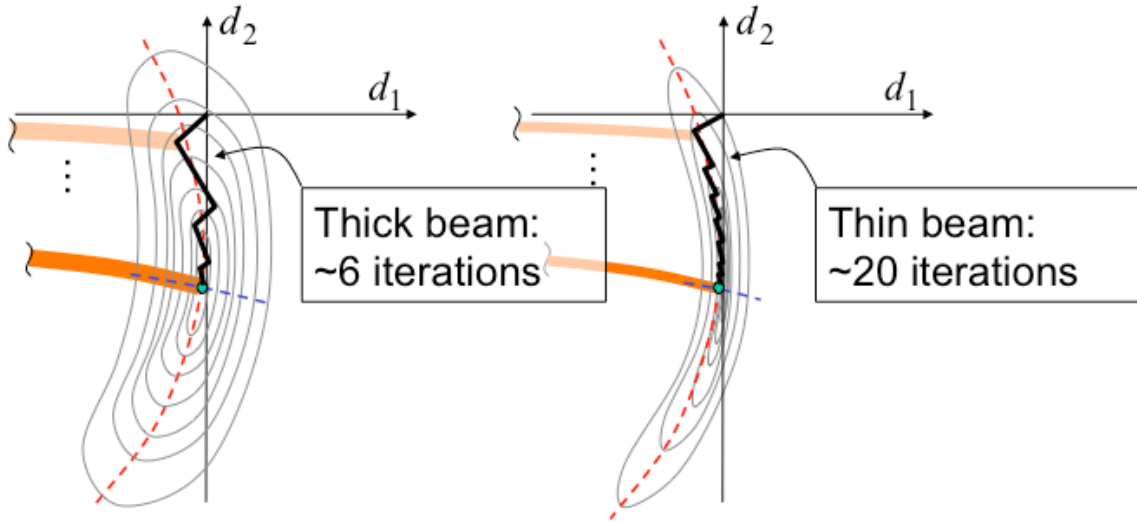

Fig. 13.10 Nonlinear conjugate gradient method applied to the two beam problem.

123

13.6.3 Convergence Properties of CG

It is well known that the convergence rates of iterative, matrix-free solution algorithms such as CG are highly dependent on the eigenvalue spectrum of the underlying equations. In the case of linear systems of equations, where the gradient direction varies linearly with the solution error, the number of iterations required for convergence is bounded by the number of degrees of freedom. Unfortunately, no such guarantee exists for nonlinear equations. In practice, it is observed that convergence is unpredictable. Depending on the nonlinearities, a solution may be obtained in surprisingly few iterations, or the solution may be intractable even with innumerable iterations and the reset strategy of (13.31), where the search direction is reset to the steepest descent (current gradient) direction.

As a practical matter, for all but the smallest problem there is an expectation that convergence will be obtained in fewer iterations. In order to understand the conditions under which this is even possible, we summarize here an analysis (which can be found in many texts) of the convergence rate of the method of conjugate gradients.

$$f(\mathbf{d}_j) = 2 \left(\frac{\sqrt{\lambda_{\max}/\lambda_{\min}} - 1}{\sqrt{\lambda_{\max}/\lambda_{\min}} + 1} \right)^j f(\mathbf{d}_0). \tag{13.35}$$

Several important conclusions can be drawn from this analysis. First, convergence of linear CG as a whole is only as fast as the worst eigenmode. Second, it is not only the spread between the maximum and minimum eigenvalues that is important but also the number of distinct eigenvalues in the spectrum. Finally, the starting value of the residual can influence the convergence path to the solution.

These conclusions hold for the case of CG applied to the linear equations, yet they remain an important reminder of what should be expected in the nonlinear case. They can provide guidance when the convergence behavior deteriorates.

13.6.4 Predictors

One of the most beneficial capabilities added to the nonlinear preconditioned CG iterative solver (nlPCG) is the ability to generate a good starting vector. Algorithmically, good starting vector is simply

$$\mathbf{d}_0^{*pred*} = \mathbf{d}_0 + \Delta \mathbf{d}^{*pred*}, \tag{13.36}$$

where $\Delta \mathbf{d}^{*pred*}$ is called the predictor.

This can dramatically improve the convergence rate. A perfect predictor would give a configuration that has no inherent error, and thus no iterations would be required to improve the solution.

Any other predicted configuration, of course, has error associated with it. This error can be expressed as a linear combination of distinct eigenvectors. Theoretically, CG will iterate at most

to the same number as there are distinct eigenvectors. The goal is to generate a predictor with less computational work than that required to iterate to the same configuration.

Computing the incremental solution from the previous step to the current one, and using this increment to extrapolate a guess to the next is a cost effective predictor. Not only is it trivially computed, but it also contains modes shapes that are actively participating in the solution. That is,

$$\Delta \mathbf{d}^{*pred*} = \mathbf{d}_{j*}^{n-1} - \mathbf{d}_0^{n-1} \tag{13.37}$$

and therefore,

$$\mathbf{d}_0^{n,*pred*} = \mathbf{d}_0^n + \Delta \mathbf{d}^{*pred*}. \tag{13.38}$$

In (13.37), $(n-1)$ refers to the previous load step, as we explicitly write the **predicted configuration** $\mathbf{d}_0^{n,*pred*}$ for load step n in (13.38).

When the solution path is smooth and gradually varying, this predictor is extremely effective. A slight improvement can be made by performing a line search along the predictor in which case it is more appropriately named a starting search direction. The effect of a simple linear predictor on our simple beam model problem is depicted in Fig. 13.11.

13.6.5 Preconditioned CG

We have mentioned the preconditioner $\mathbf{M}$ without any specifics on how it is formed. Preconditioning is essential for good performance of the CG solver. Sierra/SolidMechanics offers two forms of preconditioning, the **nodal** preconditioner and the **full tangent** preconditioner. The nodal preconditioner is constructed by simply computing and assembling the 3×3 block diagonal entry of the gradient of the residual, (13.11). In this most general case, the precondition will contain contributions from both the internal force and external force. Sierra/SolidMechanics at this point only includes the contribution to the nodal preconditioner from the internal force:

$$\left[\mathbf{M}_I^{nPC} \right] = \left[\int_{\varphi_t^h(\Omega)} \left[N_{I,i} \left(\varphi_t^{-1}(\mathbf{x}) \right) \mathbf{C} N_{I,j} \left(\varphi_t^{-1}(\mathbf{x}) \right) \right] dv \right], \tag{13.39}$$

where the term $\mathbf{C}$ in (13.39) is the instantaneous tangent material properties describing the material. For the many nonlinear material models supported by the Solid Mechanics module, exact material tangents would be onerous. A simple but effective alternative is to assume an equivalent hypo-elastic material response *for every material model* where the hypo-elastic bulk and shear moduli are conservatively set to the largest values that the material model may obtain. The formation of the nodal preconditioner is therefore simple, and need only be performed once per load step.

The full tangent preconditioner is constructed by computing the tangent stiffness matrix. As mentioned in section Section 13.4, the tangent stiffness is obtained via probing (13.15), the nonlinear system of equations.
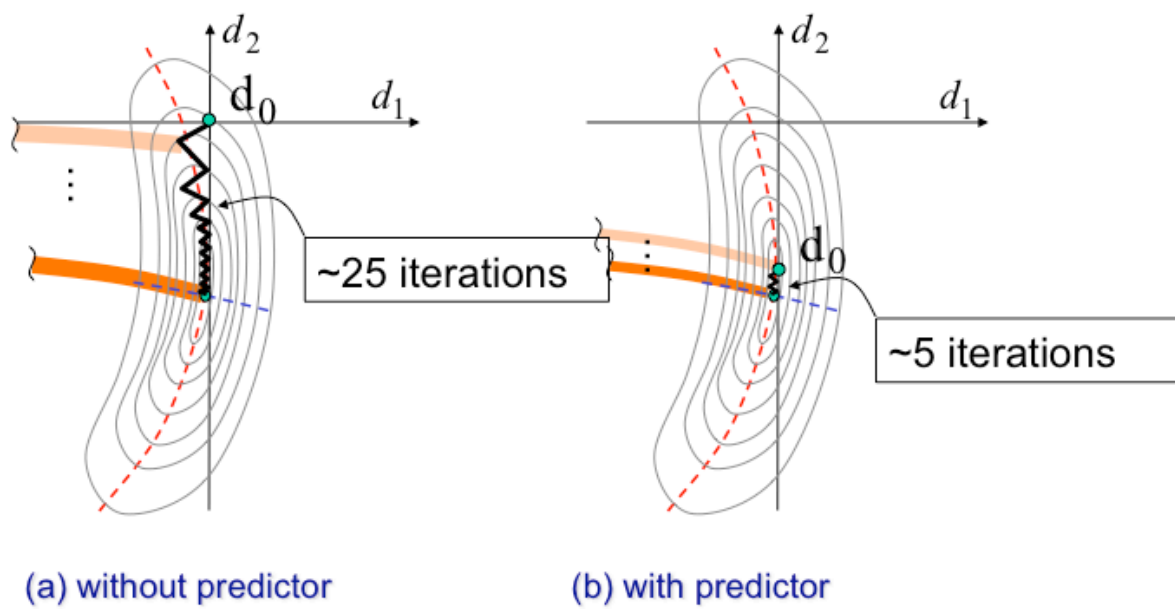
Fig. 13.11 A linear predictor applied to the beam problem can produce a good starting point.

13.7 Parallel Linear Equation Solving

FETI is now a well established approach for solving a linear system of equations on parallel MPI-based computer architectures. Its inception and early development is described in [49]. Prevalent in the literature is a description of the FETI algorithm as the foundation for a parallel implementation of Newton's method and its typical requirement for direct equation solving capability. The dual-primal unified FETI method which forms the basis of the Sierra/SM's FETI solver was introduced in [9, 10].

The Sierra/SM module generalizes the use of FETI to include not just a means to provide Newton's method, but also as a preconditioner for nonlinear preconditioned conjugate gradient. The basic notion of FETI is embodied in its name, Finite Element Tearing and Interconnecting, resulting in a separability of the linear system of equations to sub-problems, one for each processor.

13.8 Enforcing Constraints within Solvers

Theoretically, constraint enforcement is reasonably straightforward. However, performance and/or robustness difficulties reveal themselves in the practical use of solvers where there are many constraints and/or a changing active constraint set. It is in the application of the methods for treating constraints within the solver where difficulties start. Mathematically, there are two broad categories of constraints, **equality constraints** and **inequality constraints**. Again, with the aid of the simple beam example we have used throughout this chapter, Fig. 13.12 shows where one would encounter such constraints in practice.

At the fixed end of the cantilever beam, where the displacements are required to be zero, we pose an equality constraint,

$$\mathbf{h}(\mathbf{d}) = 0. \tag{13.40}$$

(13.40) is written in matrix notation and can alternatively be written in index notation as

$$h_L(d_i) = 0 \ , \ L = 1, n_{\text{con}} \ , \ i = 1, n_{\text{dofpn}}, \tag{13.41}$$

where $\mathbf{h}$ is the **constraint operator**. The constraint operator is simply the collection by row of all the equality constraints (in this case, $n_{\text{con}} = 2$). Notice that for the fixed end of the beam, the constraint operator is very simple. All of the constraints are linear with respect to the displacements $d_1^{I=1}$ and $d_2^{I=1}$. The form of the equality constraint operator may be linear, $\alpha_i d_i = 0$, or nonlinear. However, the essential feature is that the unknowns can be written on the left-hand side of the equation.

Returning to our simple beam example, the ellipse presents itself as an obstacle to the motion of the tip of the beam. It constrains node 2 to be *outside* the ellipse that has major axis a, minor axis b, is centered at $(0, c)$ and is rotated by angle α with respect to the horizontal axis. Given these
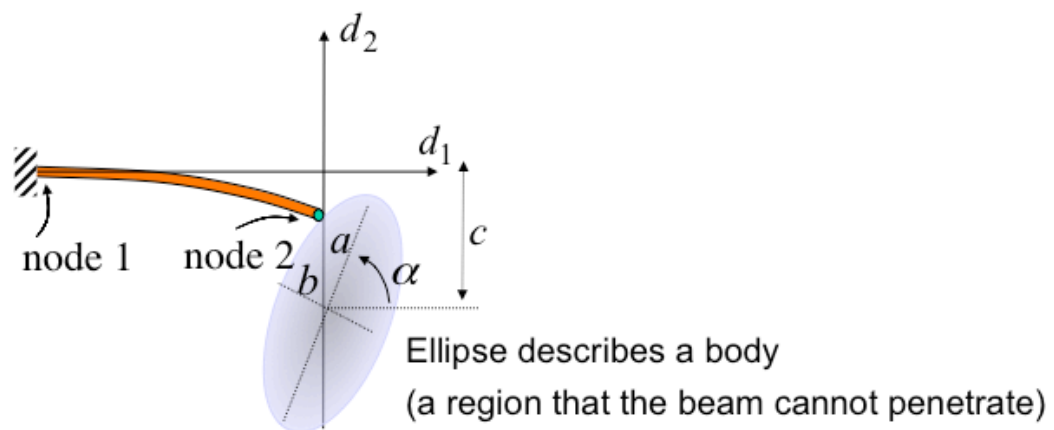
Fig. 13.12 Simple beam example with constraints.

specifications for the location and orientation of the obstacle, we write the following inequality constraint

$$\mathbf{g(d)} \geq 0, \tag{13.42}$$

in matrix notation, and alternatively in index notation as

$$t(d_i) \geq 0 \,,\, L = 1, n_{\text{con}} \,,\, i = 1, n_{\text{dofpn}}. \tag{13.43}$$

Fig. 13.13 (a) and (b) is a graphical depiction of the energy error contours as they are modified when using a Lagrange multiplier method and a penalty method, respectively.

(a) Energy error contours for Lagrange multiplier

(b) Energy error contours for penalty formulation
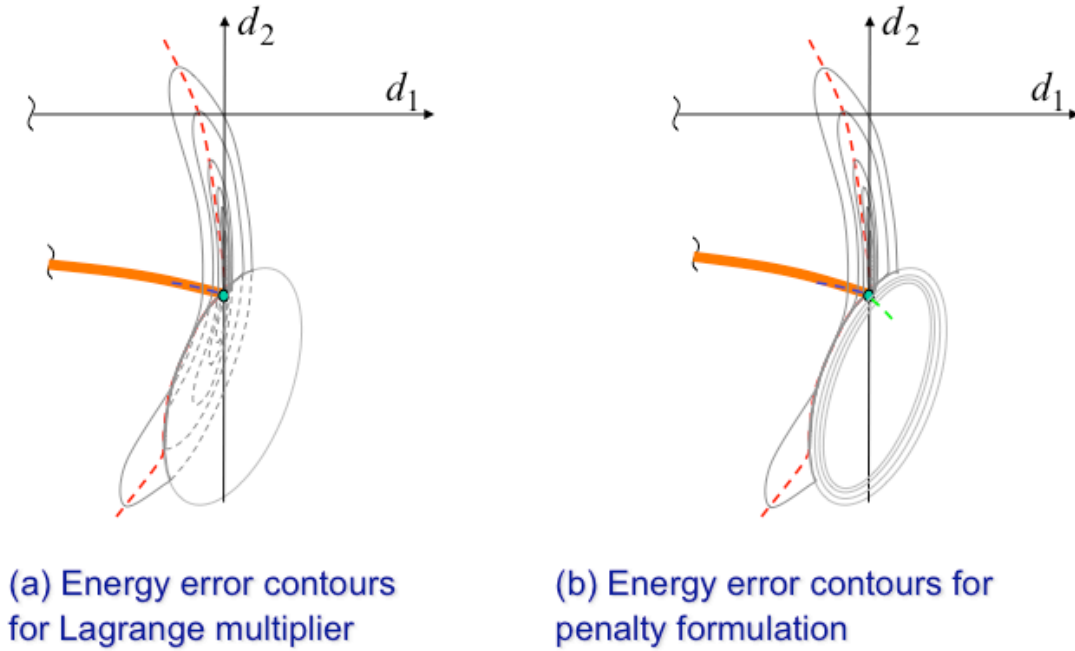
Fig. 13.13 Energy error contours for simple beam example with constraints.

Fig. 13.14 is a graphical depiction of the energy error contours as they are modified when using an augmented Lagrangian (mixed Lagrangian, penalty) method. As the tip of the beam is penetrating the ellipse (violating the kinematic constraint), a penalty force is generated according to

$$f\left(d_{k+j//j^*}\right) = \frac{1}{2}\left(d_{k+j//j^*} - d^*\right)^T r\left(d_{k+j//j^*}\right) + \lambda_k^T H\left(d_{k+j//j^*}\right) + \frac{1}{2}\varepsilon_g g^T\left(d_{k+j//j^*}\right) g\left(d_{k+j//j^*}\right),$$
$$\tag{13.44}$$

in which it is apparent that an augmented Lagrange method is a combination of a Lagrange multiplier method and a penalty method. The advantage of this approach is that the penalty ε_g can be soft, thus avoiding the ill-conditioning associated with penalty methods that must rely on overly stiff penalty parameters for acceptable constraint enforcement.
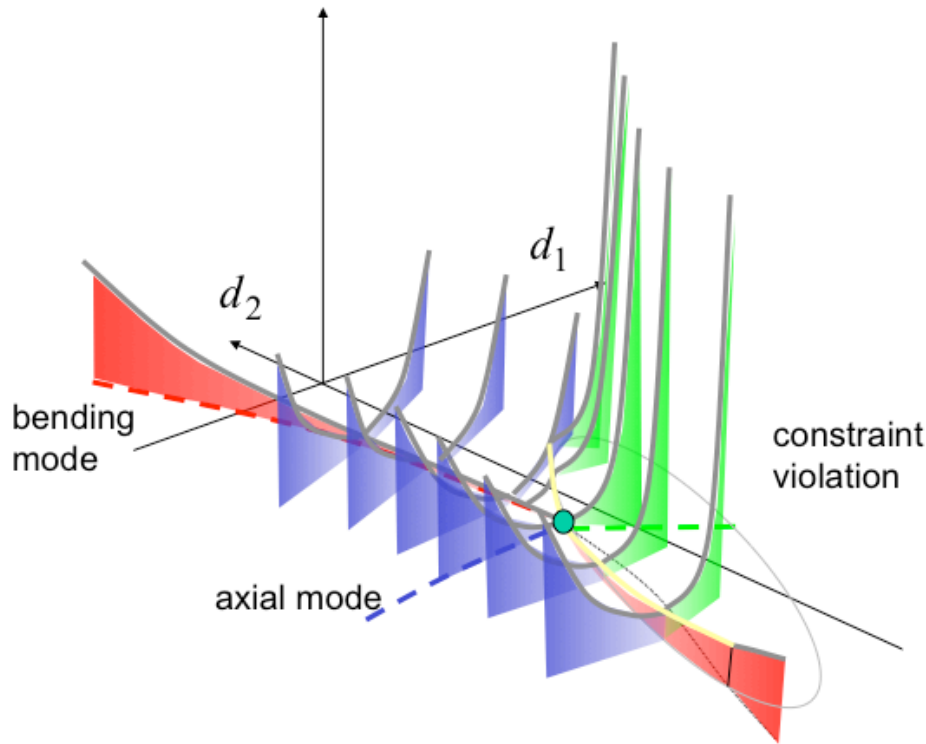
Fig. 13.14 Energy error contours for simple beam example with constraints.

The soft penalty parameter is indicated by the energy error contours increasing only moderately. The iteration counter j refers to the nonlinear CG iteration. It proceeds from $j = 1, 2, \ldots$ to j^*, where the well-conditioned model problem is converged. However, because of the soft penalty parameter, there is a significant constraint violation. Introducing an outer loop and the concept of **nested iterations**, repeated solutions of the well-conditioned problem are solved while the multiplier, λ_k , is updated in each of the outer iterations, $k = 1, 2, \ldots$.

Fig. 13.15 shows a graphical depiction of the updates of the Lagrange multiplier. The iteration counter k refers to the outer Lagrange multiplier update. Although not immediate obvious, once the multiplier is updated, dis-equilibrium is introduced (especially in the early updates) and a new model problem must be solved. Eventually, as the multiplier converges, the constraint error tends to zero as well as the corresponding dis-equilibrium.
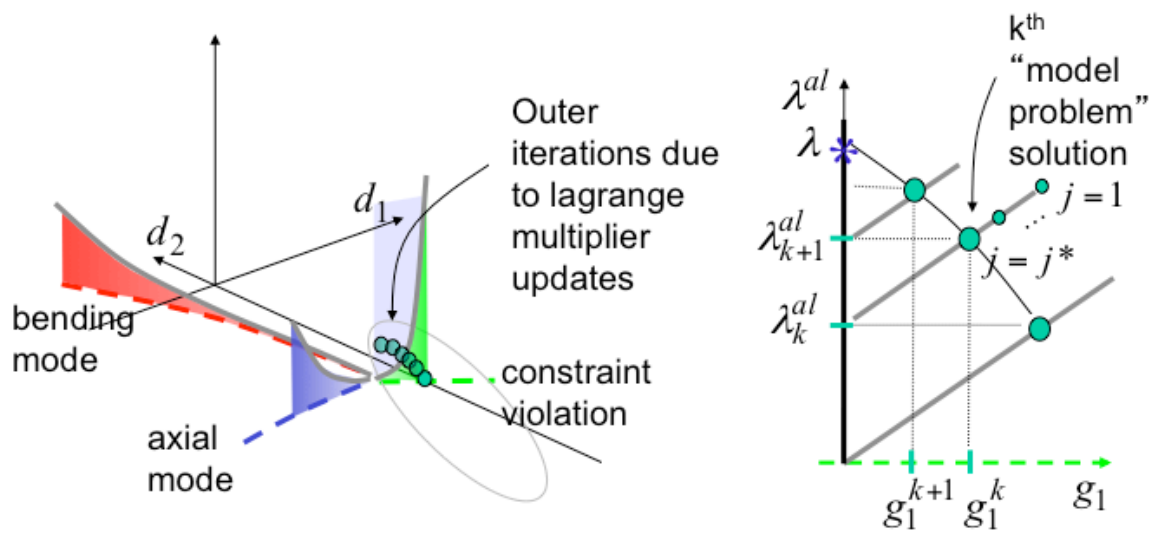
Fig. 13.15 Energy error contours for simple beam example with constraints.

13.9 Multi-Level Iterative Solver

The multi-level solver concept is based on a strategy where an attribute and/or nonlinearity is controlled within the nonlinear solver. It is important to recognize that complete linearization (as in a Newton Raphson approach) is not necessary and in many cases not optimal. Furthermore, there are several cases where nonlinearities are not even the source of the poor convergence behavior. The essential concept of the strategy is to identify the feature that makes convergence difficult to achieve and to control it in a manner that encourages the nonlinear core solver to converge to the greatest extent possible.

The control is accomplished by holding fixed a variable that would ordinarily be free to change during the iteration, by reducing the stiffness of dilatational modes of deformation, or by restricting the search directions to span only a selected sub-space. The core CG solver is used to solve a **model problem** - a problem where the control is active. When the core CG solver is converged, an update on the controlled variable is performed, the residual is recalculated, and a new model problem is solved. The approach has similarities to a Newton Raphson algorithm, as shown in Fig. 13.16.
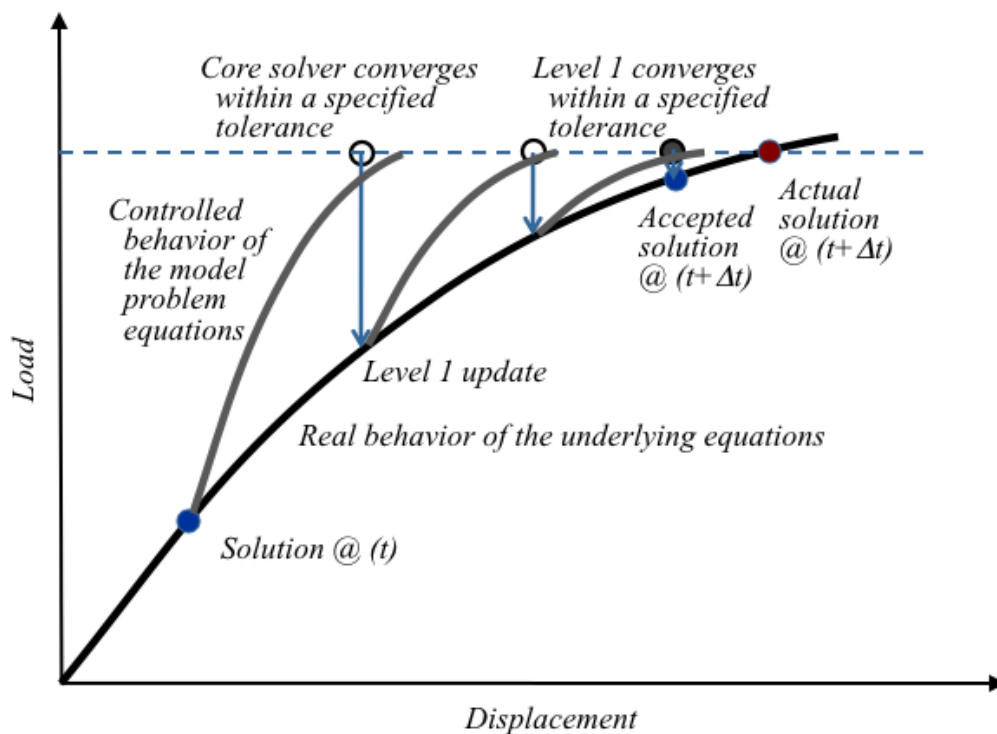
Fig. 13.16 A schematic of a single-level multi-level solver.

The generality of the multi-level solver is apparent in the case where multiple controls are active.

Multiple controls can occur at a single-level or be nested at different levels - hence the name multi-level solver. Fig. 13.17 depicts a 2-level multi-level solver.
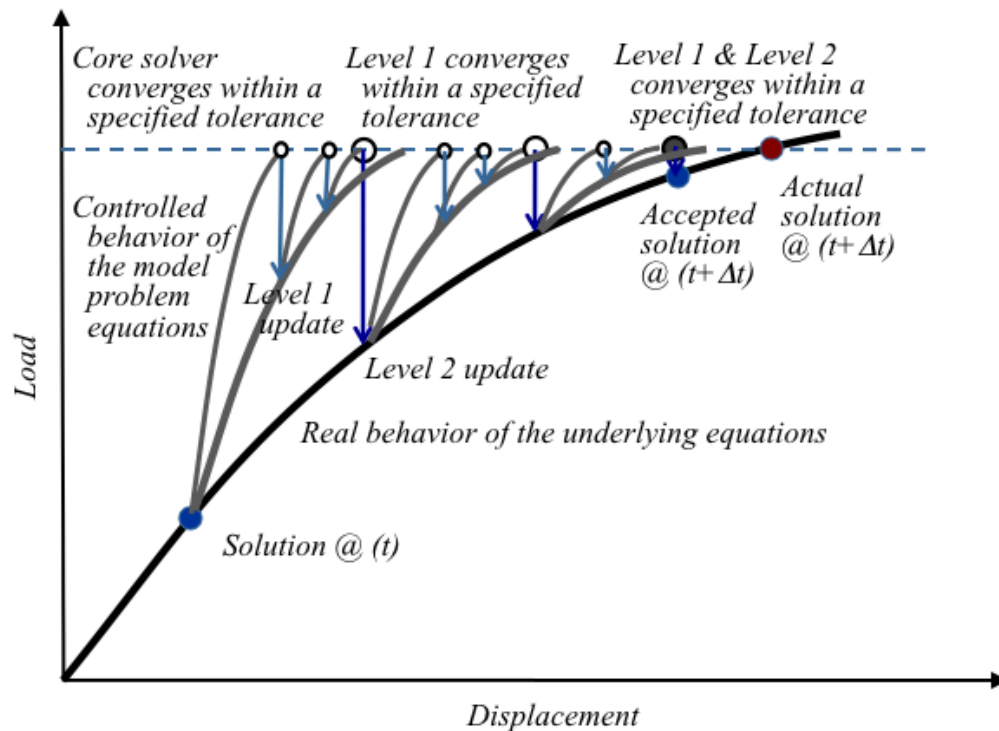
Fig. 13.17 A schematic of a two-level multi-level solver.

As depicted in Fig. 13.16 and Fig. 13.17, the iterative solver by its nature solves the model problem and/or the nested problem within some specified tolerance (as opposed to nearly exact solutions obtained by a direct solver). The inexactness of these solves is most often not an issue, however there are some cases where a certain amount of precision is required.

133

This page left blank

14 Element Basics

14.1 Properties of Shape Functions

In this chapter 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 $\mathbf{m}^e$, $\mathbf{f}_e^{\text{int}}$, and $\mathbf{k}^e$, that are needed for assembly and solution of the global equations of motion. We concentrate in this chapter 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.

To start, we discuss in general terms the 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 discuss.

The first condition relates to convergence of the finite element method in general, and the implication on properties of shape functions for elements. We begin by defining m, which will denote the highest order shape function spatial derivative present in the expression for the stiffness matrix. For the class of problems we have considered so far, we find from Section 13 that the element stiffness takes the form

$$k_{pq}^e\left(\mathbf{d}_{n+1}^{e^i}\right) = \frac{\partial f_p^{\text{int}}}{\partial d_q^e}\left(\mathbf{d}_{n+1}^{e^i}\right) \tag{14.1}$$

The internal force vector required in (14.1) was given generically in Section 10, equations (10.43) and (10.44), as:

$$f_p^{\text{int},e} = \int_{\varphi_t^h(\Omega^e)}\left[\sum_{j=1}^{3} N_{a,j}\left(\varphi_t^{-1}(\mathbf{x})\right) T_{ij}^h\right] \mathrm{d}v \tag{14.2}$$

Performing the differentiation indicated in (14.1) will produce no higher than first-order derivatives of the shape functions; therefore $m = 1$.

The three general convergence requirements we need to mention are as follows:

- The global shape function N_J 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 $\{N_J\}$) should be C^m on the element interiors.

- The elemental shape functions $\{N_J\}$ should be complete.

The first two of these requirements are fairly simple to understand. The first, 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_J should be C^0 continuous. 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 that 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 expressions are 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, as we discuss later.

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 φ_t^h to be interpolated according to the same polynomial point wise in 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 global solutions representable by polynomials, up to and including order 1, should be exactly representable by the element. It is worthwhile to consider an example of this point. Suppose we are in three dimensions and set element degrees of freedom via

$$\mathbf{d}_a^e = c_0 + c_1 X_a^e \mathbf{e}_x + c_2 Y_a^e \mathbf{e}_y + c_3 Z_a^e \mathbf{e}_z, \tag{14.3}$$

where c_0, c_1, c_2, 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

$$\varphi_t^h(\mathbf{X}^e) = \sum_{a=1}^{nen} N_a(\mathbf{X}^e)\mathbf{d}_a^e = \left(c_0 + c_1 X^e \mathbf{e}_x + c_2 Y^e \mathbf{e}_y + c_3 Z^e \mathbf{e}_z\right) \tag{14.4}$$

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

14.1.1 Element patch test

As mentioned above, the completeness 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 global solutions that are 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 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.

A particularly useful instantiation of the patch test is to prescribe a combined rigid body rotation and stretch, making use of all of the constants c_0, c_1, ..., as depicted graphically in Fig. 14.1. Here a piecewise combination of global linear function are specified.
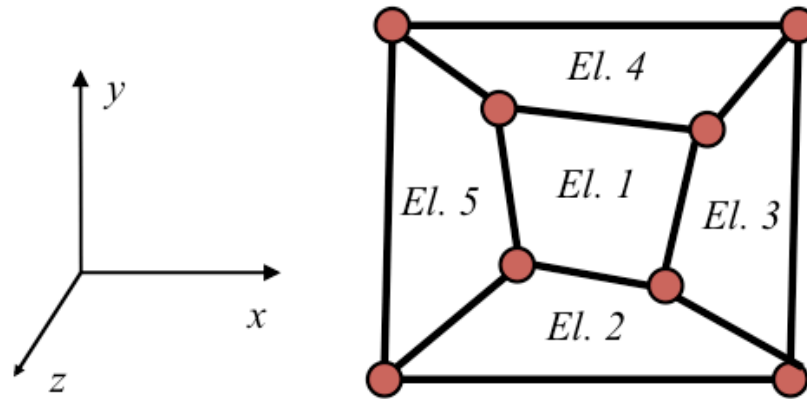
Fig. 14.1 Element Patch test in 2D.

14.2 Parameterization

With these 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, which contributes in great part to the modularity we will desire for element level operations.

We will denote a vector of these local variables by **r**, with **r** being a 2-vector in two dimensions

and a 3-vector in three dimensions. Specifically, we define $\mathbf{r}$ as

$$\mathbf{r} = \begin{bmatrix} r \\ s \end{bmatrix} \text{ two dimensions,} \qquad \begin{bmatrix} r \\ s \\ t \end{bmatrix} \text{ three dimensions} \tag{14.5}$$

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 of the same type in a given problem. The domain of $\mathbf{r}$ is often referred to as the parent domain. As shown in Fig. 14.2, the two dimension parent domain is a bi-unit square, and in three dimensions a bi-unit cube.
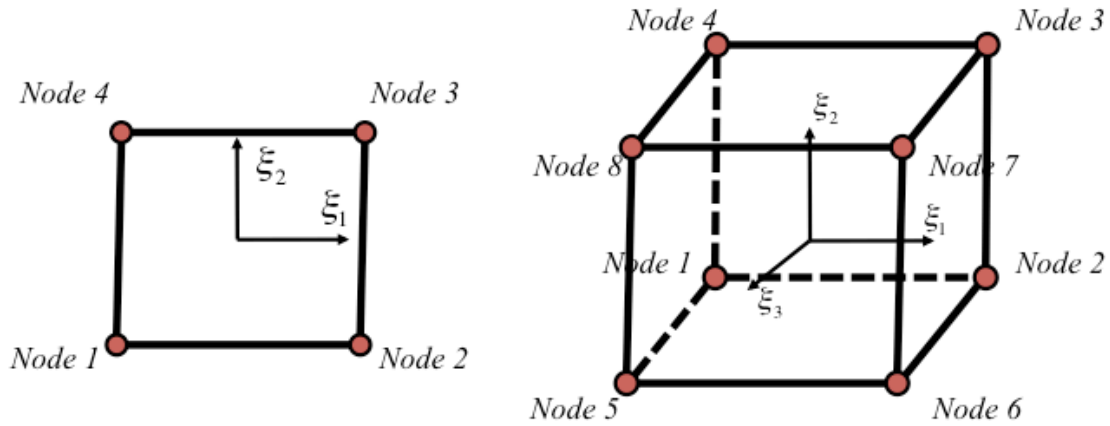
Fig. 14.2 Local parameterization and coordinate mappings in two and three dimensions.

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

$$\mathbf{X}^e(\mathbf{r}) = \sum_{a=1}^{nen} \tilde{N}_a(\mathbf{r}) \mathbf{X}_a^e, \tag{14.6}$$

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 in (14.6) that 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 a 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 (14.6). The most important generic class of finite elements is comprised of **isoparametric elements**. Such elements are defined by utilizing the same shape functions for definition of deformation $\varphi_t^h(\mathbf{X}^e)$ as for the element coordinates $\mathbf{X}^e$. One can show that, providing all element shape functions sum to one at any point in the element, an isoparametric element automatically satisfies the completeness condition. Furthermore, provided the shape functions are also suitably smooth on the element interior and match neighboring element descriptions on element boundaries, all three of the conditions required for convergence are met by isoparametric shape functions.

There are important implications of the isoparametric approach for the Lagrangian description of large deformation solid mechanics. The implications are related to the restrictions imposed on the mapping from the parent domain to the physical domain. So that we may distinguish carefully between mappings taking $\mathbf{r}$ as an argument and those taking $\mathbf{X}$, we will use the superposed tildes for the former, as in (14.6). If an element is isoparametric, then by definition the configuration mapping over an element is given by

$$\tilde{\varphi}_t^h(\mathbf{r}) = \sum_{I=1}^{nen} \tilde{N}_I(\mathbf{r})\mathbf{d}_I^e, \tag{14.7}$$

where the shape functions $\tilde{N}_I(\mathbf{r})$ are exactly the same as in (14.6). However, it should also be the case that the function $\tilde{\varphi}_t^h(\mathbf{r})$ should be obtainable from the composition of $\tilde{\varphi}_t^h(\mathbf{X}^e)$ defined according to (14.4) with $\mathbf{X}^e(\mathbf{r})$ defined according to (14.6). Thus we can write:

$$\sum_{I=1}^{nen} \tilde{N}_I(\mathbf{r})\mathbf{d}_I^e = \tilde{\varphi}_t^h(\mathbf{r}) = \sum_{I=1}^{nen} N_I(\mathbf{X}^e(\mathbf{r}))\,\mathbf{d}_I^e. \tag{14.8}$$

Comparing the leftmost and rightmost expressions of (14.8) and realizing that the equality must hold for any given combination of the element degrees of freedom $\mathbf{d}_I^e$, we are led to conclude that the alternative shape function expressions $\tilde{N}_I(\mathbf{r})$ and $N_I(\mathbf{X}^e)$ must be related by composition via

$$\tilde{N}_I = N_I \circ \mathbf{X}^e. \tag{14.9}$$

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 $\tilde{N}_I$ and then derive the implied expression for N_I according to

$$N_I = \tilde{N}_I \circ (\mathbf{X}^e)^{-1}. \tag{14.10}$$

(14.10) reveals the important implications as a practical condition on the inverse mapping $(\mathbf{X}^e)^{-1}$ of $\mathbf{X}^e$. It must be well behaved for the shape function N_I to make sense.

Fortunately, according to the implicit function theorem, the inverse function to (14.6) 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, e.g., the implication is that all interior angles in each 4-noded element must be less than 180 degrees.

Finally, let us consider shape functions that take the current coordinates, $\mathbf{x}^e = \varphi_t^h(\mathbf{X}^e)$. Such an expression is needed in (14.2) where the spatial derivatives in the current configuration are needed:

$$\hat{N}_{I,i} = \frac{\partial}{\partial \mathbf{x}_i^e} \hat{N}_I \tag{14.11}$$

where we have temporarily introduced the additional notation $\hat{N}$ to indicate that the shape function takes the current coordinate.

Following similar reasoning as above, one can conclude that the functions $\hat{N}$ must obey

$$N_I = \hat{N}_I \circ \left(\tilde{\varphi}_t^h\right)^{-1} \tag{14.12}$$

Again for the needed function $\left(\tilde{\varphi}_t^h\right)^{-1}$ to be well-behaved, the Jacobian of the transformation ((14.7) must be non-zero. This amounts to:

$$\det\left[\frac{\partial \tilde{\varphi}_t^h}{\partial \mathbf{r}}\right] = \det\left[\frac{\partial \tilde{\varphi}_t^h}{\partial \mathbf{X}^e}\right] \det\left[\frac{\partial \mathbf{X}^e}{\partial \mathbf{r}}\right] \neq 0 \tag{14.13}$$

Provided the original element definitions are not overly distorted, the second term on the right hand side of (14.13) will be non-zero. Thus the well-posedness of the spatial shape functions $\tilde{N}_I$ requires that $\det\left[\frac{\partial \tilde{\varphi}_t^h}{\partial \mathbf{X}^e}\right]$ be non-zero. Notice, though, that this is an approximation of the determinant, J, of the deformation gradient, as defined in Section 5. According to (5.10), J must be positive point wise for the concept of volume change to have any physical meaning. Thus, provided the approximated deformation mapping remains kinematically admissible (i.e., $J > 0$), the spatially defined shape functions are guaranteed to be well-behaved.

With this discussion as background, we now turn our attention to definition of the shape functions according to the parent domain. To keep the notation complexity to a minimum, we will drop the explicit distinction between N_I, $\tilde{N}_I$, and $\hat{N}_I$, referring to all these objects as simply N_I.

15 Element Formulations

This chapter covers the various elements that the Solid Mechanics module has available. We first discuss all of the **solid elements**, then the **shell elements**, and finally the **beam elements**.

15.1 Uniform Gradient Hex8 Solid Element

This element has proven to be the workhorse solid element for Solid Mechanics. The hex8 is an eight-node hexahedron element with a topology and a node numbering convention shown in Fig. 15.1. It also is referred to as the **mean-quadrature Hex8** because of the particular manner in which it generates a mean-quadrature representation of the gradient (and divergence) operator. The approach adapted for developing a mean strain rate quadrature for the eight-node hexahedron is that given by Reference [11]. While the approach and notation is cumbersome, it provides the structure needed to achieve a closed-form solution for the integration of an arbitrary hexahedron and an explicit and unambiguous identification of the orthogonal hourglass modes.
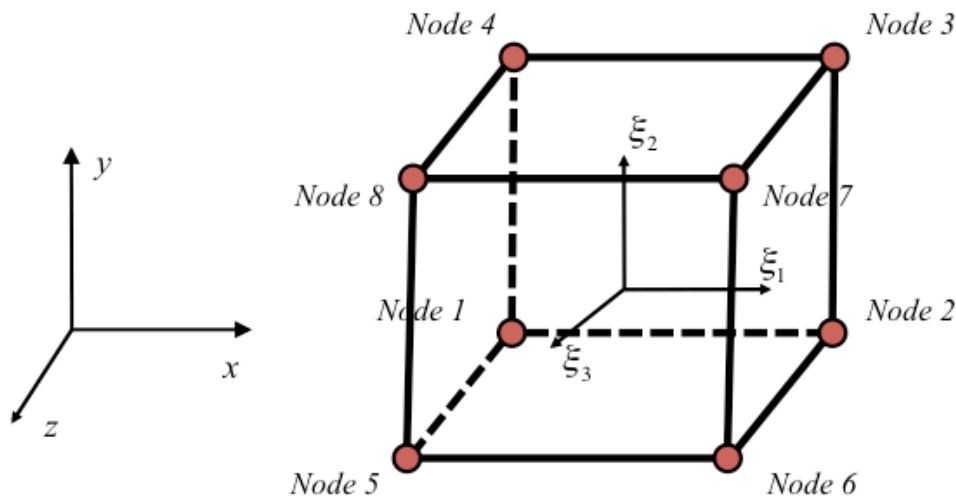
Fig. 15.1 Isoparametric coordinate representation of the eight-noded hex element.

15.1.1 Kinematics

The eight-node solid hexahedron element relates the spatial coordinates x^i to the nodal coordinates x_I^i through the isoparametric shape functions N_I as follows:

$$x^i = x_I^i N^I(\xi^i). \tag{15.1}$$

In accordance with index notation convention, repeated subscripts imply summation over the range of that subscript. The lower case subscripts have a range of three, representing the spatial coordinate directions. Upper case subscripts have a range of eight, corresponding to element nodes.

The same shape functions are used to define the element displacement field in terms of the nodal displacements u_{iI}:

$$u_i = u_{iI} N^I(\xi^i). \tag{15.2}$$

Since these shape functions apply to both spatial coordinates and displacement, their material derivative (represented by a superposed dot) must vanish. Hence, the velocity field is given by:

$$v_i = v_{iI} N^I(\xi^i). \tag{15.3}$$

The velocity gradient $v_{i,j}$ is defined as follows:

$$v_{i,j} = v_{iI} N_{,j}^I. \tag{15.4}$$

By convention, a comma preceding a lower case subscript denotes differentiation with respect to the spatial coordinates, hence $v_{i,j}$ denotes $\partial v_i / \partial x^j$.

The shape functions N^I map a unit cube in the isoparametric coordinates ξ^i to a general hexahedron in the spatial coordinates x^i. The unit cube is centered at the origin in ξ^i-space so that the shape functions may be conveniently expanded in terms of an orthogonal set of base vectors, given in Table 15.1, as follows:

$$N^I(\xi^i) = \frac{1}{8}\Sigma^I + \frac{1}{4}\xi^i \Lambda_i^I + \frac{1}{2}\xi^2\xi^3\Gamma_1^I + \frac{1}{2}\xi^1\xi^3\Gamma_2^I + \frac{1}{2}\xi^1\xi^2\Gamma_3^I + \xi^1\xi^2\xi^3\Gamma_4^I. \tag{15.5}$$

Table 15.1 Deformation modes of the eight-noded hex element

	ξ^1	ξ^2	ξ^3	Σ^I	Λ_1^I	Λ_2^I	Λ_3^I	Γ_1^I	Γ_2^I	Γ_3^I	Γ_4^I
1	$-\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	1	-1	-1	-1	1	1	1	-1
2	$\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	1	1	-1	-1	1	-1	-1	1
3	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	1	1	1	-1	-1	-1	1	-1
4	$-\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	1	-1	1	-1	-1	1	-1	1
5	$-\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	1	-1	-1	1	-1	-1	1	1
6	$\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	1	1	-1	1	-1	1	-1	-1
7	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	1	1	1	1	1	1
8	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	-1	1	1	1	-1	-1	-1

The above vectors represent the deformation modes of a unit cube, as shown in Fig. 15.2. The first vector, Σ^I accounts for rigid body translation. The linear base vectors Λ_i^I may be readily

combined to define three uniform normal strain rate modes, three uniform shear strain rate modes, and three rigid body rotation rates for the unit cube. The last four vectors Γ_α^I (Greek subscripts have a range of four) give rise to modes with linear strain variations which are neglected by mean strain quadrature. These vectors define the hourglass patterns for a unit cube. Hence the modes Γ_α^I are referred to as the hourglass base vectors.
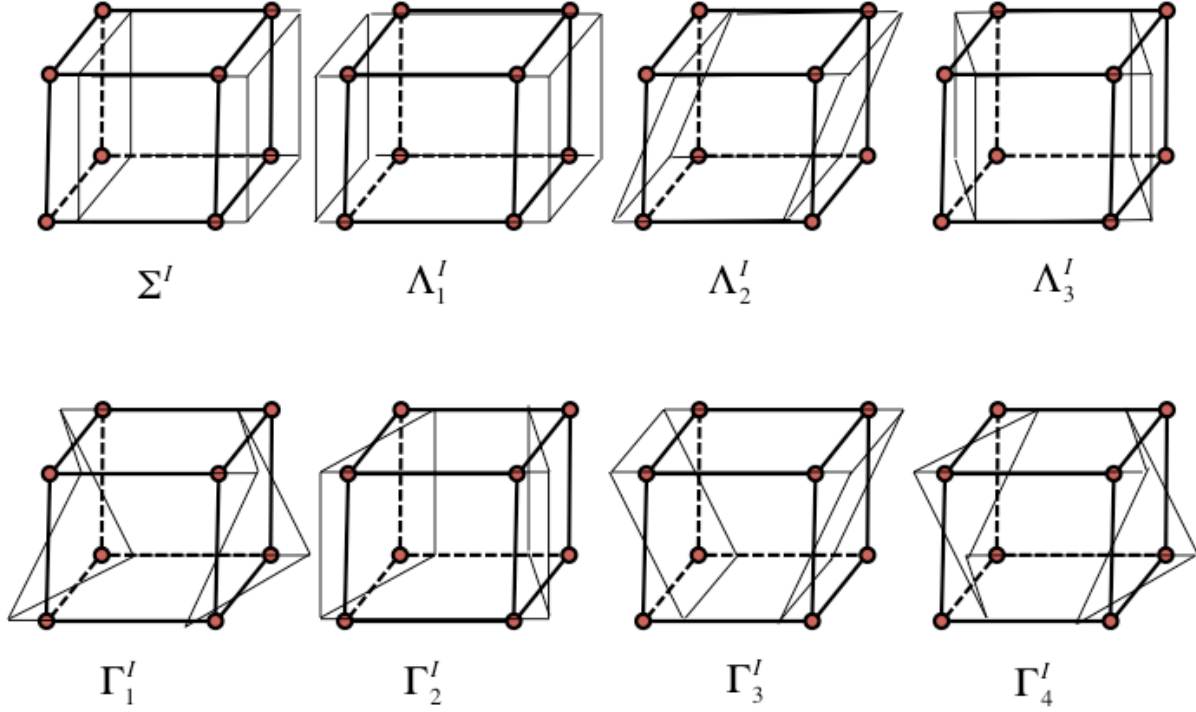
Fig. 15.2 Deformation modes of the eight-noded hex element.

15.1.2 Mean Quadrature

The variational statement gives the following relationship for the element nodal forces f^{iI} due to the divergence of the stress field,

$$v_{iI} f^{iI} = \int_V t^{ij} d_{ij} \mathrm{d}v. \tag{15.6}$$

The integral in (15.6) is evaluated using a constant stress, thereby considering only a mean strain rate within the element:

$$v_{iI} f^{iI} = \int_V t^{ij} d_{ij} \mathrm{d}v = V \bar{t}^{ij} \bar{v}_{i,j}. \tag{15.7}$$

The assumed constant stress field is represented by $\bar{t}^{ij}$, which will be referred to as the **mean stress tensor**. It is assumed that the mean stress depends only on the mean strain. Mean kinematic quantities are defined by integrating over the element as follows:

$$\bar{v}_{i,j} = \frac{1}{V} \int_V v_{i,j} \mathrm{d}v. \tag{15.8}$$

The gradient operator B_i^I is defined by

$$B_i^I = \int_V N_{,j}^I \mathrm{d}v. \tag{15.9}$$

The mean velocity gradient, applying (15.9) is then given by

$$\bar{v}_{i,j} = \frac{1}{V} v_{iI} B_j^I. \tag{15.10}$$

The nodal forces are then given by

$$f^{iI} = \bar{t}^{ij} B_j^I. \tag{15.11}$$

Computing nodal forces by this integration scheme requires evaluation of the gradient operator B_j^I and volume. These two tasks can be linked together by using the relationship $x_{,j}^i = \delta_j^i$. Therefore (15.9) yields

$$x_I^i B_j^I = \int_V \left(x_I^i N^I \right)_{,j} \mathrm{d}v = V \delta_j^i. \tag{15.12}$$

Consequently, the gradient operator B_i^I may alternatively be expressed by

$$B_j^I = \frac{\partial V}{\partial x_I^i}. \tag{15.13}$$

To integrate the element volume in closed form, the Jacobian of the isoparametric transformation is used transform the integral over the unit cube,

$$V = \int_V \mathrm{d}V = \int_{-1/2}^{1/2} \int_{-1/2}^{1/2} \int_{-1/2}^{1/2} J \mathrm{d}\xi^1 \mathrm{d}\xi^2 \mathrm{d}\xi^3. \tag{15.14}$$

The Jacobian J is the determinant of the transformation operator $\partial x^i / \partial \xi^j$ and may be expressed as

$$J = e^{ijk} \frac{\partial x^1}{\partial \xi^i} \frac{\partial x^2}{\partial \xi^j} \frac{\partial x^3}{\partial \xi^k}. \tag{15.15}$$

Using (15.1), (15.14), and (15.15), the element volume may be expressed in the following form:

$$V = x_I^1 x_J^2 x_K^3 D^{IJK}, \tag{15.16}$$

where

$$D^{IJK} = e^{ijk} \int\limits_{-1/2}^{1/2} \int\limits_{-1/2}^{1/2} \int\limits_{-1/2}^{1/2} J \frac{\partial N^I}{\partial \xi^i} \frac{\partial N^J}{\partial \xi^j} \frac{\partial N^K}{\partial \xi^k} \mathrm{d}\xi^1 \mathrm{d}\xi^2 \mathrm{d}\xi^3. \qquad (15.17)$$

Observe that the coefficient array D^{IJK} is identical for all hexahedra. Furthermore, it possesses the alternator properties given by

$$D^{IJK} = D^{JKI} = D^{KIJ} = -D^{IKJ} = -D^{JIK} = -D^{KJI}. \qquad (15.18)$$

Therefore, applying (15.13) and (15.14) to the expression (15.16) yields the following closed-form expression for evaluation the components of the gradient operator, B_i^I:

$$\begin{bmatrix} B_1^I \\ B_2^I \\ B_3^I \end{bmatrix} = \begin{bmatrix} x_J^2 x_K^3 \\ x_J^3 x_K^1 \\ x_J^1 x_K^2 \end{bmatrix} D^{IJK}. \qquad (15.19)$$

Looking at the form of (15.19), it is evident that evaluating each component of D^{IJK} involves integrating a polynomial which is at most bi-quadratic. However, since the integration is 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 D^{IJK} is

$$D^{IJK} = \frac{1}{192} e^{ijk} \left(3\Lambda_i^I \Lambda_j^J \Lambda_k^K + \Lambda_i^I \Gamma_k^J \Gamma_j^K + \Gamma_k^I \Lambda_j^J \Gamma_i^K + \Gamma_j^I \Gamma_i^J \Lambda_k^K \right). \qquad (15.20)$$

The expression in (15.20) is evaluated using Table 15.1, after which practical formula for computing the gradient operator B_i^I and volume are developed.

The gradient operator component B_x^1 is given explicitly by

$$B_x^1 = [y_2(z_{63} - z_{45}) + y_3 z_{24} + y_4(z_{38} - z_{52}) + y_5(z_{86} - z_{24}) + y_6 z_{52} + y_8 z_{45}] /12, \qquad (15.21)$$

where $\{x_I^i\} = \{x_I, y_I, z_I\}$ and $z_{IJ} = z_I - z_J$. To obtain the balance of the gradient operator components B_x^I, the nodal index permutations contained in Table 15.2 are used. To obtain the components B_y^I and B_z^I, the coordinate permutations contained in Table 15.3 are used.

Table 15.2 Permutation of Nodal Indices

B_i^1	2	3	4	5	6	7	8
B_i^2	3	4	1	6	7	8	5
B_i^3	4	1	2	7	8	5	6
B_i^4	1	2	3	8	5	6	7
B_i^5	8	7	6	1	4	3	2
B_i^6	5	8	7	2	1	4	3
B_i^7	6	5	8	3	2	1	4
B_i^8	7	6	5	4	3	2	1

Table 15.3 Permutation of Nodal Coordinates

B_x^I	y	z
B_y^I	z	x
B_z^I	z	y

It is worth noting at this point the difference between the mean quadrature (alt. mean strain rate, mean stress) approach and one-point Gauss quadrature. The latter method would effectively neglect the last three terms of (15.21). In a parallelepiped, the nodal coordinates contain no component of the hourglass base vectors, consequently, only the first term of (15.21) is necessary to compute the gradient operator and volume. In such a case, one-point quadrature is equivalent to the mean quadrature formula. However, for a general hexahedron shape, one-point quadrature does not correctly assess a state of uniform stress and strain, thus, may not be convergent [Zienkiewicz, 1977]. In view of the requirements of the Iron's patch test, it is likely that (15.20) is unique.

15.1.3 Orthogonal Hourglass Control

The mean stress - mean strain rate formulation considers only the linear part of the velocity field. The remaining portion of the velocity field is the so-called hourglass field. Excitation of these modes may lead to sever, unresisted mesh distortion. A method for isolating the hourglass modes so that they may be treated independently of the rigid body and uniform strain modes is required. This is accomplished by developing an **hourglass gradient operator** connected with hourglass restoring forces. The *linear* velocity field on which the mean strain rates are based is given by

$$v_i^{\text{LIN}} = v_{iI} \left(\frac{1}{8}\Sigma^I + \frac{1}{V}(x^j - \frac{1}{8}x_J^j\Sigma^J)B_j^I \right). \tag{15.22}$$

The hourglass velocity field v_i^{HG} may be defined by removing the linear portion of the velocity field. Thus,

$$v_i^{\text{HG}} = v_i - v_i^{\text{LIN}}, \tag{15.23}$$

or in terms of the nodal velocities,

$$v_i^{\text{HG}} = v_{iI} - v_{i0}\Sigma_I - \frac{1}{V}\left(x_I^j - x_0^j\Sigma_I\right)v_{iJ}B_j^J, \tag{15.24}$$

where $v_{i0} = \frac{1}{8}v_{iI}\Sigma_I$ and $x_0^i = \frac{1}{8}x_I^i\Sigma_I$.

The hourglass velocity field, (15.24), is in the improper null space of the gradient operator B_i^I. The linear velocity field, (15.22), spans 12 degrees of freedom: 3 rates of rigid body translation, 3 rates of rigid body rotation, and 6 uniform strain rates, which means that the hourglass subspace is remaining 12 degrees of freedom.

An hourglass gradient operator is constructed from the hourglass basis vectors Λ_α^I as follows:

$$G_\alpha^I = \frac{V}{\delta}\left[\Lambda_1^I, \ \Lambda_2^I, \ \Lambda_3^I, \ \Lambda_4^I\right], \tag{15.25}$$

where δ is a generalized element dimension developed below. This scaling provides the hourglass gradient operator with the same dimensional characteristics as the uniform gradient operator. While G_α^I is orthogonal to B_i^I, the following property:

$$B_i^I G_\alpha^I \neq 0,\tag{15.26}$$

means that G_α^I used with the full velocity field v_{iI} will couple the hourglass behavior to the uniform strain rate behavior. Thus, hourglass strain rates $\dot{q}_{i\alpha}$ are developed with G_α^I operating on only the hourglass velocities v_{iI}^{HG},

$$\dot{q}_{i\alpha} = \frac{1}{V} v_{iI}^{\mathrm{HG}} G_\alpha^I.\tag{15.27}$$

Alternatively, an unrestricted operator may be developed by requiring is to satisfy the following condition:

$$v_{iI} \gamma_\alpha^I = v_{iI}^{\mathrm{HG}} G_\alpha^I.\tag{15.28}$$

Using the hourglass velocity, (15.24), provides

$$v_{iI} \gamma_\alpha^I = \left[v_{iI} - v_{i0} \Sigma^I - \frac{1}{V} \left(x_I^j - x_0^j \Sigma_I \right) v_{iJ} B_j^J \right] G_\alpha^I,\tag{15.29}$$

which, when rearranged and using the orthogonality of the mode shapes Σ^I and Γ_α^I (i.e., $\Sigma^I \Gamma_\alpha^I = 0$) gives

$$v_{iI} \gamma_\alpha^I = v_{iI} \left(G_\alpha^I - \frac{1}{V} x_J^j G_\alpha^J B_j^I \right).\tag{15.30}$$

The condition for the unrestricted operator is satisfied if the hourglass operator γ_α^I is defined as

$$\gamma_\alpha^I = \frac{V}{\delta} \left(\Gamma_\alpha^I - \frac{1}{V} x_J^j \Gamma_\alpha^J B_j^I \right),\tag{15.31}$$

and the hourglass strain rates are defined as

$$\dot{q}_{i\alpha} = \frac{1}{V} v_{iI} \gamma_\alpha^I.\tag{15.32}$$

To control the hourglass modes, generalized forces $Q^{i\alpha}$ are defined which are conjugate to $\dot{q}_{i\alpha}$, so that the work rate is given by

$$v_{iI} f_{\mathrm{HG}}^{iI} = V Q^{i\alpha} \dot{q}_{i\alpha}.\tag{15.33}$$

Utilizing (15.31), the contribution to the nodal forces due to hourglass resistance is given be

$$f_{\mathrm{HG}}^{iI} = Q^{i\alpha} \gamma_\alpha^I.\tag{15.34}$$

The hourglass restoring forces are calculated from

$$\check{Q}^{i\alpha} = \epsilon \, 2\mu_{tan} \delta^{ij} \delta^{\alpha\beta} \dot{q}_{j\beta},\tag{15.35}$$

where $2\mu_{tan}$ is the tangent shear stiffness obtained from the deviatoric constitutive behavior of the mean stress and mean strain rate in the element, and ϵ is a scaling parameter. The scaling ϵ assures the level of the hourglass restoring forces remains below that of the mean deviatoric stress state.

The deviatoric behavior is used since the hourglass modes are constant volume, higher order straining modes of the element. The tangent modulus assures that the evolution of the hourglass restoring forces *parallels* that of the mean deviatoric stress state.

The invariant time derivative of the generalized forces $Q^{i\alpha}$ accounts for the finite rotations expected in use of the element in applications. The derivative is given by

$$\check{Q}^{i\alpha} = \dot{Q}^{i\alpha} - \omega_{ij}Q^{j\alpha}, \tag{15.36}$$

where ω_{ij} is the spin tensor.

The hourglass restoring forces are added to those obtained from the divergence of the mean stress state so that the complete result is

$$f^{iI} = \left(\bar{t}^{ij}B_j^I + Q^{i\alpha}\gamma_\alpha^I\right). \tag{15.37}$$

15.1.4 Linear Hyperelastic Hourglass Control

The traditional hourglass formulation is in rate form, and as such is not guaranteed to be energetically reversible and may not result in symmetric contributions to the finite element stiffness matrix. Here we describe a hyperelastic hourglass formulation that overcomes these limitations and also allows us to define Lagrangian hourglass strains which are valuable for extending the formulation to nonlinear hourglass response, as discussed in the next section.

Consider an element's total hourglass energy defined as follows:

$$\psi_e(\mathbf{x}) = V\epsilon\mu_{tan}\left(\sum_{\alpha=1}^{4} \varepsilon_\alpha^2\right), \tag{15.38}$$

where V is the reference volume of the element, $\mathbf{x}$ is the element's current nodal coordinates, and ε_α is a measure of the hourglass strain for hourglass mode α. The hourglass strain here is based on an hourglass operator defined in the model/reference coordinates (this is in contrast to the total and incremental hourglass formulations which use hourglass operators in the current configuration). The hourglass strains are given by

$$\varepsilon_\alpha = \sqrt{\sum_{i=1}^{3} \epsilon_{i\alpha}\epsilon_{i\alpha}} \quad \text{(no sum over } \alpha),$$

with hourglass strain vector ϵ:

$$\epsilon_{i\alpha} = \sum_{I=1}^{8} H_\alpha^I x_i^I,$$

where H_α^I is the unrestricted hourglass operator defined in the reference configuration:

$$H_\alpha^I = \frac{1}{\delta}\left[\Gamma_\alpha^I - \frac{1}{V}X_j^J\Gamma_\alpha^J B_j^I\right],$$

where B is the gradient operator with respect to the reference configuration, and $\mathbf{X}$ is the element's reference coordinates. Note that this definition of the unrestricted hourglass is analogous to γ_α^I from Section 15.1.3, but includes a factor of $\frac{1}{V}$ to simplify notation. The value of δ, a characteristic element length scale, is chosen so that the hyperelastic hourglass forces match the classical formulation at small stains. The proposed hourglass energy is objective due to the definition of the hourglass strain, which is invariant to rigid body rotations of the current coordinates, $\mathbf{x}$. The resulting hourglass forces follow from work conjugacy:

$$f_{\text{HG}}^{iI} = -\frac{\partial \psi_e}{\partial x_i^I}.$$

Being derived from an objective potential energy, these forces are objective, path independent. The resulting stiffness matrix is guaranteed to be symmetric and hourglass deformations are elastically reversible. Because we are assuming an energy which is quadratic in the hourglass strain, the resulting forces are linear with hourglass displacement. In addition, the unrestricted hourglass operator H has both rigid body modes and affine deformation modes in its null-space, meaning the hourglass forces will be orthogonal to affine motions of the element.

An extension to a non-linear hyperelastic hourglass formulation is described below.

15.1.5 Nonlinear Hyperelastic Hourglass Control

An additional limitation of the traditional hourglass control is that the hourglass resistance is typically formulated to be (incrementally) linearly proportional to hourglass deformation increments. A hyperelastic hourglass control formulation can overcome this limitation by using a nonlinear hyperelastic energy which is a function of the four hourglass strains. We use a generalized definition of the energy function from (15.38):

$$\hat{\psi}_e(\mathbf{x}) = V\epsilon\mu_{tan}\varepsilon_0^2\left(\sum_{\alpha=1}^{4}\left[e_m\left(\frac{\varepsilon_\alpha}{\varepsilon_0}\right)\right]^2\right),$$

where ε_0 is called the transition strain, and $e_m(\cdot)$ is a function which takes a strain and returns an alternative *Seth-Hill* strain measure. In particular, it is defined by

$$e_m(\varepsilon) = \frac{1}{m}\left((\varepsilon+1)^m - 1\right),$$

if ε is a strain, $e_m(\varepsilon)$ satisfies all the requirements of a strain measure. In particular, it satisfies $e_m(0) = 0$, $e_m'(0) = 1$. As a result, at small strains, $e_m \approx \varepsilon$ for any m. A value of $m = 2$ corresponds to a Green-Lagrange strain measure, while $m = 1$ is an identity map. The variable ε_0 is called the transition strain because it sets the strain level at which the nonlinearity of the strain

measure begins to become dominant. For a very large transition strain, the hourglass force response will remain linear up to large hourglass strains as the term $\frac{\varepsilon_\alpha}{\varepsilon_0}$ remains small. For small transition strains, the nonlinearity become noticeable earlier. The hourglass energy for this model scales as ε^{2m} as strains get large, meaning the hourglass force scales as the hourglass displacement to the $2m - 1$ power at large deformations.

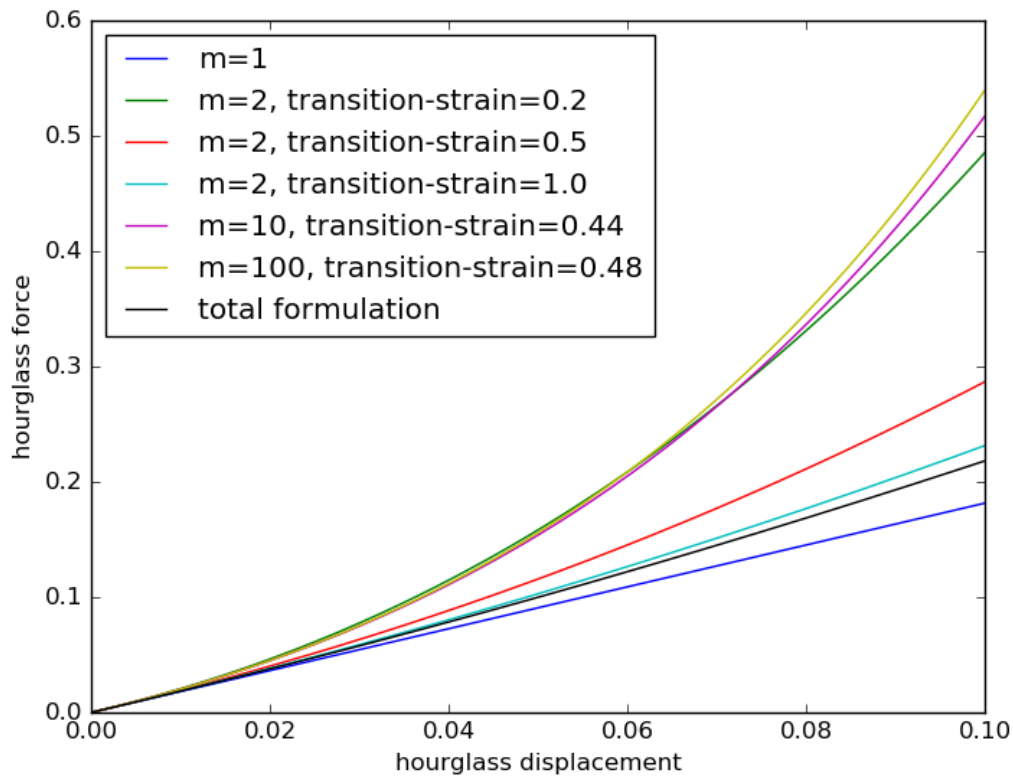Fig. 15.3 shows the hourglass resistance force vs displacement for varying transition strains and Seth-Hill exponent m.

Fig. 15.3 Nonlinear hourglass force versus displacement.

15.2 Tet4 Solid Element

This element is the standard 4-noded tetrahedral element. It is notoriously stiff and prone to locking, but included for completeness. More information on this element can be found in [27].

15.3 Tet10 Solid Element

The default 10-noded tetrahedral solid element is the composite tetrahedron, as given in [43], which is an extension of the composite tetrahedron and triangle formulations in [47]] and [[50]. This 10-noded tetrahedron consists of 12 linear, 4-noded sub-tetrahedra. The nodal fields, including displacement, are linear within each sub-tetrahedron and, therefore, piecewise linear within the parent 10-noded tetrahedron. The deformation gradient and stress fields are formulated to be linear over this 10-noded tetrahedron; the gradient operator projects the piecewise discontinuous gradients among the 12 sub-tetrahedra into a linear basis on the parent tetrahedron.

As stated in [43]], [[47]], and [[50], there are several advantages of this composite tetrahedron formulation over commonly used alternatives. Tetrahedral elements provide generally more robust and efficient finite element meshing than hexahedral elements. As opposed to the traditional formulation of the quadratic 10-noded tetrahedron, this formulation has a well-balanced mass lumping to all 10 nodes lending to improved performance in explicit transient dynamics, contact, and other solid mechanics capabilities that rely upon the nodal mass distribution. Volumetric locking and unrealistic pressure oscillations are still possible for this element when modeling isochoric deformation (plasticity) and nearly incompressible materials, but this behavior can be alleviated further by volume-averaging the dilation over the element [43]. In Sierra/SM, this option is called `VOLUME AVERAGE J = ON`, which is a default setting for the composite tetrahedron.

A 10-noded, quadratic, fully-integrated tetrahedral element is also available in Sierra/SM. For more information on this element, refer to [27].

15.4 Belytschko-Tsay Shell Element

The 4-noded Belytschko-Tsay shell (or BT-shell4) is the simplest of the shell elements offered. The original reference can be found in [3]. It should be considered as the minimal 5-parameter Mindlin-type formulation that includes a constant transverse shear contribution.

15.5 Key-Hoff Shell Element

The 4-noded Key-Hoff shell (or KH-shell4) is slightly more involved than the BT-shell4, in that it includes a term for a linear-varying transverse shear in its formulation. The inclusion of this term is an improvement on the BT-shell4 because it properly models warped shell geometry - albeit in a low-order way. The original reference for this element can be found in [30].

15.6 Belytschko-Leviathan Shell Element

The 4-noded Belytschko-Leviathan shell (or BL-shell4) is slightly more involved than the KH-shell4, in that it includes additional shear terms as well as additional hourglass controls. The inclusion of the hourglass terms (also known as the physical stabilization parameter) is an improvement on the KH-shell in that it eliminates some of the over-stiffness sometimes observed in the KH-shell4. The BL-shell4 also includes a projection of the angular velocities and the internal forces. The original references for this element can be found in [1]] and [[2].

15.7 Shear Correction for Layered Shell Elements

For sandwich composite plates with a low modulus core, the effects of transverse shear deformation can be significant. Thus, the results of first-order shear deformation theory, as applied for layered shell elements in Sierra/SM, are affected by the choice of shear correction factor (K). In reference [37], an expression is derived for the variation of transverse shear through the thickness of a laminated plate. The expression given for the shear correction factor is:

$$K = \left[\left(A_{44} - \frac{A_{45}^2}{A_{55}} \right) \int_{-h/2}^{h/2} \frac{\left[\int_{-h/2}^{z} (\bar{Q}_{1i}\beta_{1i} + z\bar{Q}_{1i}\delta_{1i})dz \right]^2}{\left[\bar{q}_{44} - \frac{\bar{Q}_{45}^2}{\bar{Q}_{55}} \right]} \right]^{-1} , \quad \text{for } i = 1, 2, 6. \qquad (15.39)$$

Here, the A_{ij} are the corresponding terms in the laminate extensional stiffness matrix, the $\bar{Q}_{ij}$ are the terms of the reduced stiffness matrix, β_{ij} and δ_{ij} are terms of the compliance sub-matrices, and h is the thickness of the section. When expressed in algebraic form, for a laminate of N layers and a coordinate system centered at the centroidal axis of the section, the shear correction factor can be expressed as

$$K = \frac{\left[A_{44} - \frac{A_{45}^2}{A_{55}} \right]^{-1}}{\sum_{k=1}^{N} \frac{1}{\left(\bar{Q}_{44}^k - \frac{\bar{Q}_{45}^{k2}}{\bar{Q}_{55}^k} \right)} \left[P_k(z_{k+1} - z_k) + \frac{R_k}{2}(z_{k+1}^2 - z_k^2) + \frac{V_k}{3}(z_{k+1}^3 - z_k^3) + \frac{W_k}{4}(z_{k+1}^4 - z_k^4) + \frac{X_k}{5}(z_{k+1}^5 - z_k^5) \right]},$$

$$(15.40)$$

where

$$P_k = T_k^2 + H_k^2 z_k^2 - 2T_k H_k z_k + U_k^2 + \frac{J_k^2 z_k^4}{4} - U_k J_k z_k^2 + 2T_k U_k - T_k J_k z_k^2 - 2H_k U_k z_k + H_k J_k z_k^3,$$

$$R_k = 2T_k H_k - 2H_k^2 z_k + 2H_k U_k - H_k J_k z_k^2,$$

$$V_k = H_k^2 - \frac{J_k^2 z_k^2}{2} + U_k J_k + T_k J_k - z_k H_k J_k,$$

$$W_k = H_k J_k,$$

$$X_k = \frac{J_k^2}{4},$$

$$T_k = \sum_{m=1}^{k-1} H_m (z_{m+1} - z_m),$$

$$U_k = \sum_{m=1}^{k-1} \frac{J_m}{2} (z_{m+1}^2 - z_m^2),$$

$$H_k = \bar{Q}_{1i}^k \beta_{1i} , \text{ for } i = 1, 2, 6,$$

$$(15.41)$$

and

$$J_k = \bar{Q}_{1i}^k \delta_{1i} , \text{ for } i = 1, 2, 6.$$

For the laminate cross-section geometry, see Figure 2 in reference [37]. This correction factor has been coded into a subroutine and is used for the layered shell formulation in Sierra/SM.

15.8 3D Beam Element

The two-noded beam in Sierra/SM is based on conventional Timoshenko beam theory in which the functional form of the deformation is made explicit on a cross section normal to the reference axis. Thus, the deformation is described in terms of kinematic variables that depend on the coordinate along the reference axis. As shown in Fig. 15.4, the axis connecting *node 1* and *node 2* labeled ξ_1 is this reference axis. The beam is defined by a cross-section of fixed-shape existing uniformly along the reference axis and is formulated using isoparametric coordinates.

As will be apparent, the assumptions about the deformation of the beam are those of a Timoshenko beam theory. In particular, the transverse shear deformation is modeled. Planar cross-sections originally perpendicular to the reference axis remain flat and undeformed though not necessarily remain perpendicular to the reference axis under deformation.

When initially curved beams are modeled with straight beam segments, the global curvature properties are represented by the change in orientation from one beam element to the next. In effect, the smooth variation in curvature of the original reference axis is approximated by discrete changes in orientation occurring at the element ends; the elements are chord approximations to the original curved beam, much like linear shell elements when modeling a curved structure. This

approximation is the same order as the constant membrane and bending stress approximations introduced in the element integration.
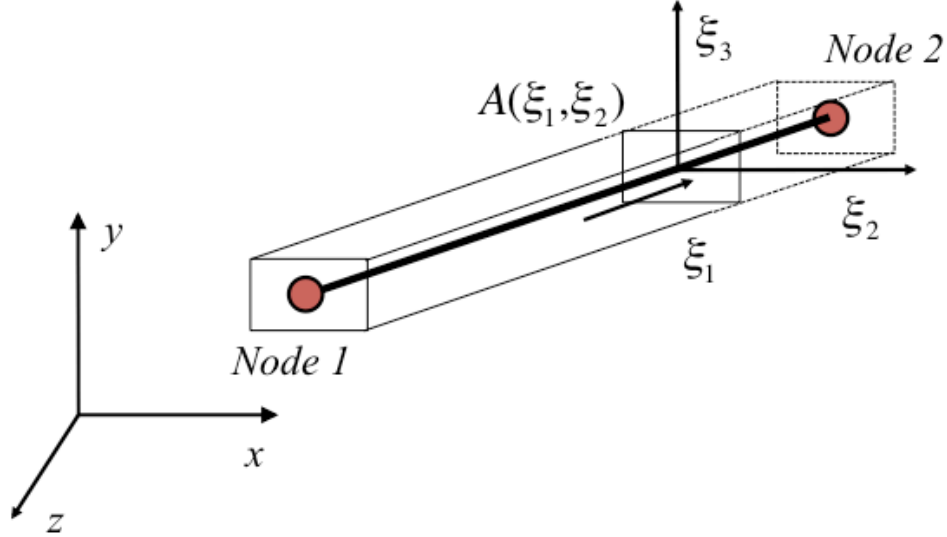

Fig. 15.4 Isoparametric coordinate representation of the two-noded beam element.

15.8.1 Kinematics

The motion of the beam is defined in terms of the velocity of the reference axis and the additional rotation of the region within the cross-section defined by $A(\xi_2, \xi_3)$,

$$v_i(x^j) = v_i(\xi_1) - \varepsilon_{imn}\rho^m\omega^n(\xi_1). \tag{15.42}$$

Here, ρ^m is the position vector from the reference axis to a point in the cross-section $A(\xi_2, \xi_3)$. The position vector is perpendicular to the reference axis and has the units of length.

Based on (15.42), the spatial gradient of the velocity is given by

$$v_{i,j}(x^k) = v_{i,j}(\xi_1) - \varepsilon_{imn}\rho^m\omega^n_{,j}(\xi_1). \tag{15.43}$$

[Note: In the special case when the isoparametric coordinates ξ_i coincide with the spatial coordinates x_i, the velocity of the beam is given by:

$$\{v_i\} = \{(v_x + z\omega_y - y\omega_z), \ (v_y - z\omega_x), \ (v_z + y\omega_x)\}. \tag{15.44}$$

The stretching (symmetric part of the velocity gradient) is then given by

$$
\begin{aligned}
d_{xx} &= v_{x,x} + z\omega_{y,z} - y\omega z, x, \\
d_{yy} &= 0, \\
d_{zz} &= 0, \\
2d_{xy} &= -\omega_z + v_{y,x} - z\omega x, x, \\
2d_{xz} &= \omega_y + v_{z,x} - y\omega x, x, \\
2d_{yz} &= 0,
\end{aligned}
\tag{15.45}
$$

and the spin (skew-symmetric part of the velocity gradient) is given by

$$
\begin{aligned}
2\omega_{xy} &= -\omega_z - v_{y,x} + z\omega x, x, \\
2\omega_{xz} &= \omega_y - v_{z,x} - y\omega x, x, \\
2\omega_{yz} &= -2\omega_x,
\end{aligned}
\tag{15.46}
$$

where it is now apparent that Timoshenko beam theory allowing transverse shear deformation is considered, see Reference [6]. Using Timoshenko beam theory allows the rotation rates ω_y and ω_z to be described separately, rather than defined by $-v_{z,x}$ and $v_{y,x}$, respectively. Consequently, the (separate) finite element assumptions on the velocity and rotation rates are required to be no more than continuous represented. In the event that the slender beam limit of vanishing transverse shear strains holds, classical beam theory is recovered, though special considerations in the element formulation (introduced below) are needed to prevent shear locking.]

Returning to our description of the more general case, the two-noded beam relates the spatial coordinates x_{iI} through the isoparametric shape functions $N_I, \ I = 1, 2$ as follows:

$$x_i = x_{iI} N_I(\xi_1). \tag{15.47}$$

The shape functions map a unit interval in the isoparametric coordinate ξ_i to a general beam segment in the spatial coordinates, x_i. The unit interval is centered at the origin in the ξ_1-space so that the shape functions may be conveniently expanded in terms of an orthogonal set of base vectors:

$$N_I(\xi_1) = \frac{1}{2}\Sigma_I + \xi_1\Lambda_I \tag{15.48}$$

where at *node 1*: $\xi_1 = -\frac{1}{2}$, $\Sigma_1 = 1$, $\Lambda_1 = -1$, and at *node 2*: $\xi_1 = +\frac{1}{2}$, $\Sigma_2 = 1$, $\Lambda_2 = 1$. As shown in Fig. 15.5, these two modes represent the deformation modes of a unit interval $-\frac{1}{2} \leq \xi_1 \leq \frac{1}{2}$.

Although the velocity gradient of the two-noded beam is quite complex in description when using a Timoshenko beam theory (15.43), the modes Σ_I and Λ_I combine to represent rates of rigid body translation and rotation, and the uniform strain rates, with no hourglass mode of deformation.
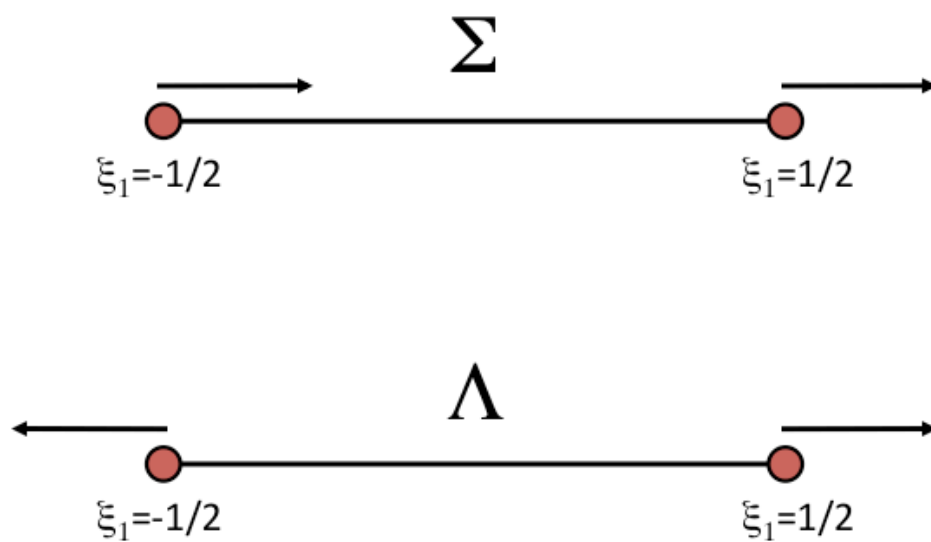
Fig. 15.5 Deformation modes of a unit interval.

The same shape functions are used to define the reference axis displacement in terms of the nodal displacements, u_{iI}:

$$u_i = u_{iI}N_I(\xi_1). \tag{15.49}$$

Since these shape functions apply to spatial coordinates and displacements, their material derivatives must vanish. Hence, the velocity field and rotational rate are given by

$$v_i = v_{iI}N_I(\xi_1),$$
$$\omega_i = \omega_{iI}N_I.(\xi_1) \tag{15.50}$$

The velocity gradient and the gradient of the rotational rate are defined as follows:

$$v_{i,j} = v_{iI}N_{I,j},$$
$$\omega_{i,j} = \omega_{iI}N_{I,j}. \tag{15.51}$$

15.8.2 Mean Quadrature

In order to introduce the concept of a mean (constant) strain and stress in the beam, we need to deal with the explicit dependence of the velocity on the coordinates ξ_2 and ξ_3 normal to the reference axis. The divergence of the stress field in the variational statement is expanded for the beam as follows:

$$\int_V t^{ij}d_{ij}dv \int_{-1/2}^{+1/2}\int_A t^{ij}v_{i,j}(\xi_1)l\,da\,d\xi_1$$
$$-\int_{-1/2}^{+1/2}\int_A t^{ij}\left[\varepsilon_{imn}\rho^m_{,j}\omega^n(\xi_1) + \varepsilon_{imn}\rho^m\omega^n_{,j}(\xi_1)\right]l\,da\,d\xi_1. \tag{15.52}$$

The dependence on ξ_2 and ξ_3 is explicit since J specializes to

$$J = A\varepsilon_{rst}\frac{\partial x_r}{\partial \xi_1}m_sn_t = Al, \tag{15.53}$$

where l is the length of the beam, A its (constant) cross-sectional area, and m_s and n_t are the unit vectors along the ξ_2 and ξ_3 axes, respectively.

At this point, we can write the classical force and bending stress resultants N^{ij} and M^i_j as:

$$N^{ij} = \int_A t^{ij}da,$$
$$M^i_j = \int_A t^{in}\varepsilon_{nmj}\rho^m da. \tag{15.54}$$

Now, we introduce the **average stresses** $\tau^{i}j$ and **average bending stresses** μ^{i}_{j} as:

$$
\begin{aligned}
\tau^{ij} &= \frac{1}{A} \mathcal{N}^{ij}, \\
\mu^{I}_{j} &= \frac{1}{A} \mathcal{M}^{i}_{j}.
\end{aligned}
\tag{15.55}
$$

Combining (15.52) through (15.55) yields a reduced divergence of the stress field:

$$
\begin{aligned}
\int_{V} t^{ij} d_{ij} \mathrm{d}v & \int\limits_{-1/2}^{+1/2} \tau^{ij} v_{i,j}(\xi_1) l A \mathrm{d}\xi_1 \\
& - \int\limits_{-1/2}^{+1/2} \left[\tau^{ij} \varepsilon_{imn} \rho^{m}_{,j} \omega^{n}(\xi_1) + \mu^{j}_{n} \omega^{n}_{,j}(\xi_1) \right] l A \mathrm{d}\xi_1.
\end{aligned}
\tag{15.56}
$$

The integrals in the reduced divergence of the stress field in the element are evaluated using a mean stress, thereby considering only a state of constant axial, bending, and torsional stress within the element. Expressing (15.56) explicitly with mean kinematic quantities $\bar{v}_{i,j}$, $\bar{\omega}^{n}$ and $\bar{\omega}^{n}_{,j}$, and mean stresses $\bar{\tau}^{ij}$ and $\bar{\mu}^{j}_{n}$, yields:

$$
\int_{V} t^{ij} d_{ij} \mathrm{d}v = V \left(\bar{\tau}^{ij} \bar{v}_{i,j} + \bar{\tau}^{ij} \varepsilon_{imn} \rho^{m}_{,j} \bar{\omega}^{n} + \bar{\mu}^{j}_{n} \bar{\omega}^{n}_{,j} \right),
\tag{15.57}
$$

where the mean kinematic quantities are defined by integrating over the element as follows:

$$
\begin{aligned}
\bar{v}_{i,j} &= \frac{1}{V} \int_{V} v_{i,j} \mathrm{d}v, \\
\bar{\omega}^{n} &= \frac{1}{V} \int_{V} \omega^{n} \mathrm{d}v, \\
\bar{\omega}^{n}_{,j} &= \frac{1}{V} \int_{V} \omega^{n}_{,j} \mathrm{d}v.
\end{aligned}
\tag{15.58}
$$

The gradient operator is defined by

$$
: label : element_{f}ormulations : eq : b18
$$

$$
B_{iI} = \int\limits_{-1/2}^{+1/2} N_{I,j} J \mathrm{d}\xi_1,
$$

and an averaging operator is defined by

$$
: label : element_{f}ormulations : eq : b19
$$

$$
A_{I} = \int\limits_{-1/2}^{+1/2} N_{I} J \mathrm{d}\xi_1.
$$

With these definitions, the mean velocity gradient, mean rotational rate, and mean rotational rate gradient can be expressed in the more convenient form:

$$
\begin{aligned}
\bar{v}_{i,j} &= \frac{1}{V} v_{iI} B_{jI}, \\
\bar{\omega}^n &= \frac{1}{V} \omega_I^n A_I, \\
\bar{\omega}^n_{,j} &= \frac{1}{V} \omega_I^n B_{jI}.
\end{aligned}
\tag{15.59}
$$

Thus, the divergence of the stress field becomes:

$$
\int_V t^{ij} d_{ij} \mathrm{d}v = v_{iI} \left(\bar{\tau}^{ij} B_{jI} \right) - \varepsilon_{imn} \rho^m_{,j} \omega_I^n \left(\bar{\tau}^{ij} A_I \right) - \omega_I^n \left(\bar{\mu}_n^j B_{jI} \right),
\tag{15.60}
$$

where, evident by inspection of (15.60), the nodal forces due to the divergence of the stress field are then given by:

$$
f_{iI} = \bar{\tau}^{ij} B_{jI},
\tag{15.61}
$$

and the nodal torques by:

$$
m_{nI} = -\varepsilon_{imn} \rho^m_{,j} \bar{\tau}^{ij} A_I - \bar{\mu}_n^j B_{jI}.
\tag{15.62}
$$

15.8.3 Evaluation of Stress Resultants

The constant axial, bending, and torsional stress resultant assumptions result in a mean gradient operator and an averaging operator that select mean strain rates linear over the cross-section of the beam. Material non-linearities, though, will result in the stress distribution over the cross to be anything but linear, e.g., in the case of plasticity. As a consequence, the integrals for the force and bending stress resultants are implemented using numerical quadrature over the cross-section. The location of these integration points for the different cross-sections supported are shown in the Theusersguide.

At each integration point, the strain rate is computed from the nodal velocities and rotation rates. The material constitutive behavior is also incrementally evaluated. With a weighting factor and distance from the reference axis for each integration point, the stress resultant integrals are computed simply as a weighted-sum over all integration points. Finally, the stress resultant integrals include the optional offset of the reference axis from the geometric centroid of the cross-section. Details of how the cross-section is specified and how the reference axis is offset are discussed in the Sierra/SM User Manual.

15.8.4 Bending Performance

A correction of the strain energy in the bending of thick beams is necessary due to the overestimation of the transverse shear contributions. This correction of $\frac{4}{5}$ (Reference [6]) is related to the discrepancy between the constant distributions of transverse shear strains implied by the displacement assumptions of the beam and the parabolic through distribution.

In the limit of reducing cross-sectional area, a beam theory with transverse shear becomes arbitrarily stiff in transverse shear response and the transverse shear strains should vanish. Without any correction, the result is a $\frac{1}{h^2}$ (and $\frac{1}{w^2}$) growth in the transverse shear strain energy, known as shear locking. If l is the length of the beam element, transverse shear strain energy scale factors of $\frac{6h^2}{l^2}$ (and $\frac{6w^2}{l^2}$) provide, in the limiting case of slender beam behavior, quadratic displacement convergence to the Kirchhoff bending result without the shear locking in the element. Implementation of the shear locking correction factors is done by considering the minimum of $\frac{4}{5}$ and $\frac{6h^2}{l^2}$ (and $\frac{4}{5}$ and $\frac{6w^2}{l^2}$), thus allowing a transition from the transverse shear corrected thick beam to the vanishing transverse shear strains d_{xz}, d_{yz} (implying $v_{z,x} = \omega_y$ and $-v_{y,x} = \omega_x$) required for the thin slender behavior.

15.9 3D Spring Element

The 2-noded 3D spring element is a simple beam formulation that includes concepts embodied in Timoshenko beam theory.

15.10 Superelement

The superelement formulation in Sierra/SM conforms to the Craig-Bampton reduction capability in Sierra/SD. An option in Sierra/SM is a corotational formulation, which uses the Kabsch algorithm [29] to minimize the root mean square deviations between model coordinates and current coordinates of the superelement connection nodes. Additionally, this superelement formulation supports uniform gravity load and uniform initial velocity, with the latter satisfying a zero modal velocity condition, $v_i = 0$.

16 Contact

section{Contact virtual work}

As a starting point for the treatment of contact, its contribution to the virtual work expression can be stated as:

$$\int_{S^3} \left(-t_N \delta g_N + t_{T_\alpha} \delta g_T^\alpha \right) \mathrm{d}a, \tag{16.1}$$

where S^3 is the common surface between two continua, t_N is the contact normal traction (positive in compression) t_{T_α} is the contact tangential traction in one of two local (tangent plane) directions α, and $delta g\_N:math:$ and *delta g_T^{alpha}:math:* are the directional derivatives of the contact normal gap *g_N$ and tangential slip :math:'g_T^{alpha}* in the direction of $\dot\varphi$, i.e.:

$$
\begin{aligned}
\delta g_N &:= \left. \frac{\mathrm{d}}{\mathrm{d}\beta} \right|_{\beta=0} \left[g_N \left(\varphi + \beta\dot\varphi \right) \right] \\
, \delta g_T^\alpha &:= \left. \frac{\mathrm{d}}{\mathrm{d}\beta} \right|_{\beta=0} \left[g_T^\alpha \left(\varphi + \beta\dot\varphi \right) \right].
\end{aligned}
\tag{16.2}
$$

In (16.1), the deformation is subject to the following constraints, referred to as the **Kuhn-Tucker conditions**. The Kuhn-Tucker conditions are a set of constraints to be considered representative of the mechanical contact problem in continuum mechanics, and can be written as:

$$
\begin{aligned}
\delta g_N &\geq 0 \text{ (a) impenetrability constraint,} \\
t_N &\geq 0 \text{ (b) no adhesion condition,} \\
t_N g_n &\geq 0 \text{ (c) complementary condition,} \\
t_N \dot g_N &\geq 0 \text{ (d) persistency condition,}
\end{aligned}
\tag{16.3}
$$

for frictionless response and

$$
\begin{aligned}
\Phi &:= \|t_T\| - \mu t_N \leq 0 \text{ (a) slip function,} \\
L_v g_T - \varsigma \frac{t_T}{\|t_T\|} &= 0 \text{ (b) slip rule,} \\
\varsigma &\geq 0 \text{ (c) consistency parameter,} \\
\Phi\varsigma &= 0 \text{ (d) complementary condition,}
\end{aligned}
\tag{16.4}
$$

for the prescription of a Coulomb friction (where μ is the friction coefficient). In (16.3), the gap g_N is defined with respect to all material points $\mathbf{Y} \in S^3$ as:

$$g_N(\mathbf{X}, t) = \min_{\mathbf{Y} \in S^3} \|\varphi(\mathbf{X}) - \varphi(\mathbf{Y})\| * sign * (g_N), \tag{16.5}$$

where

$$\mathrm{sign}(g_N) = \begin{cases} -1 & \text{if } \varphi(\mathbf{X}) \text{ lies on the interior of the contacted body,} \\ 1 & \text{otherwise.} \end{cases} \tag{16.6}$$

The tangential gap rate $L_v g_T$ in (16.4) is defined as follows:

$$L_v g_T = \left(\varphi\left(\mathbf{X}\right) - \varphi\left(\bar{\mathbf{Y}}(\mathbf{X})\right) \right) \cdot \left(\mathbf{p}^\alpha \otimes \mathbf{p}_\alpha \right), \tag{16.7}$$

where $\mathbf{p}^\alpha$ and $\mathbf{p}_\alpha$ are base vectors associated with any appropriate surface coordinate system used to describe S^3, with these base vectors being evaluated at the *current* contact point $(\bar{\mathbf{Y}}(\mathbf{X}))$ that satisfies the minimization of (16.3). Use of the notation L_v is meant to imply a Lie derivative, which can be understood to be the time derivative of an object as viewed from an embedded reference frame, in this case the convected frame $\mathbf{p}_\alpha$ frame, that moves along with the point.

16.1 Discretized forms of contact constraints

The question is then, how to represent these conditions in discretized form suitable for FE solution methods. A simple example, shown in Fig. 16.1, serves to demonstrate the concern that this question embodies. Two discretizations for the interface are evident and, as this simple example indicates, leads to an ambiguous definition of the interface.
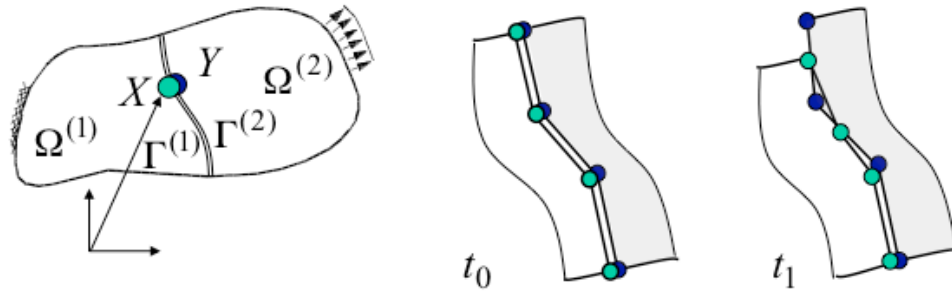

Fig. 16.1 Concerns in constraints choices for contact problems.

A historical treatment of contact has focused on applying the Kuhn-Tucker condition directly to the discretized form, leading to what we will be referring to here as a node-face treatment of contact, or **node-face contact**. As we will review here in this chapter, this treatment of contact is relatively straightforward from a conceptual standpoint, however it does have several issues - even to the point of the overall approach being pathological in some applications.

Alternatively, more recent investigations have focused on addressing these issues, leading to what we will be referring to here as a face-face treatment of contact, or **face-face contact**. These methods consider the weak form more directly, thus leading to a variationally consistent approach (e.g., mortar methods are an example of this approach and are, at the moment, prevalent in the literature).

16.1.1 Node-Face contact

For node-face contact the Kuhn-Tucker conditions are assumed to apply to one side of the contacting surfaces.Thus the gap g_N is defined with respect to all **nodal points Y_I** as:

$$g_N(\mathbf{X}_I, t) = \min_{\mathbf{Y}_I \in S^3} \|\varphi(\mathbf{X}_I) - \varphi(\mathbf{Y})\| * sign * (g_N), \tag{16.8}$$

where I refers to a nodal point on one side of the interface, whose coordinates are $\mathbf{X}_I$, t at time t of interest. The right-hand side of (16.8) is the discrete form of (16.5) but is more commonly called the **closest-point projection**, which will be discussed in some detail in Section 16.2.3.

As mentioned, there are issues associated with node-face constraints. They stem from the application of contact constraints directly to the discretized problem. As shown in Fig. 16.1, the potential to over constrain the interface is avoided by applying the impenetrability constraint only at selected points along the interface. In the Solid Mechanics module these points coincide with the nodes, as it makes it convenient to obtain contact results (normal and tangential tractions, stick/slip results, etc.) and interpret them in post-processing.

However, this approach does not truly alleviate over-constraining. This is easily demonstrated with an enlightening example (we will make use of this example for the discussion of node-face contact and face-face contact, so making a proper introduction is worthwhile). Fig. 16.2, shows a beam bending problem that is being modeled with continuum elements, in this case hex8 elements. The beam is cantilevered at its left end appropriately, i.e., fixed at the neutral axis and constrained from motion only in the x-direction elsewhere.

The analytic solution to this problem is one where the neutral axis should take the displacement corresponding to an arc of a circle. When the moment is prescribed to be M^* the beam should deform into a perfect circle.

When the beam is meshed with either an all coarse mesh (4 elements through its thickness) or an all fine mesh (16 elements through its thickness), the Finite Element results appear to be quite acceptable, producing the pure bending solution.

However, lets now combine coarse and fine discretizations to solve the problem. In this case a mesh tying constraint is required to obtain the solution, *which is seen to be fundamentally a*
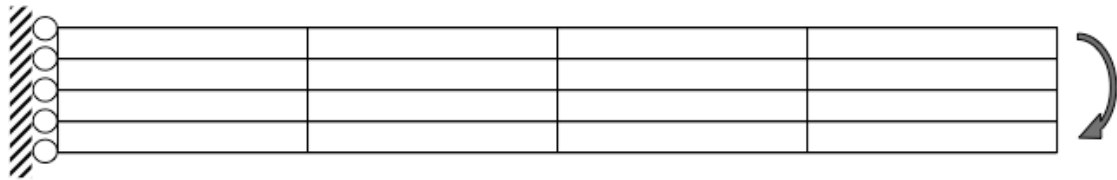
Fig. 16.2 A continuum beam subjected to pure bending.

contact problem with adhesion and infinite frictional capacity. The combining of coarse and fine discretizations can be done is a couple of canonical ways, as shown in Fig. 16.3; one where the interface between the discretizations is vertical and the other where is along the neutral axis.
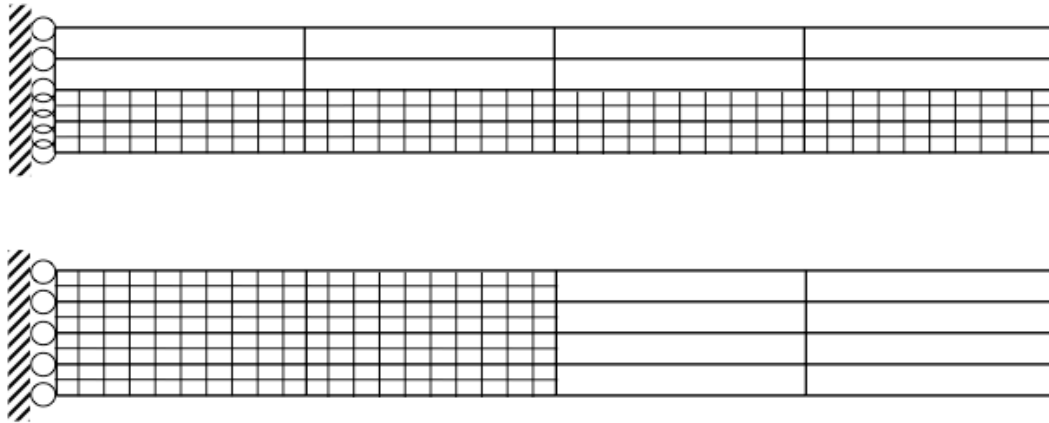

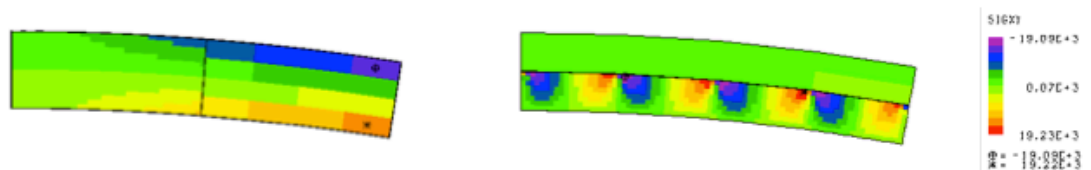**Fig. 16.3 A continuum beam that includes mesh tying subjected to pure bending.**

In both cases, we apply the standard rule of thumb: given the same material on both sides of the interface, apply the contact constraints on the finer discretization. Subjecting the beam to the prescribed moment reveals at once the issue: kinematically enforcing a zero gap condition at each node is exactly correct in one case, where the interface is through the depth of the beam, and severely over constraining in the other, where the interface is along the neutral axis. As Fig. 16.4 shows, the over constraint can be severe and may produce spurious stress distributions in the fine mesh, particularly near the neutral axis.
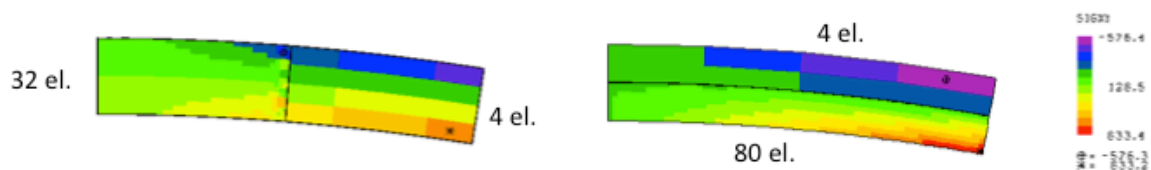
Consistent mesh tying: A constant stress field is accurately transmitted across the interface, but higher order modes are left stress-free (analogy: mean strain modes , hourglass modes)
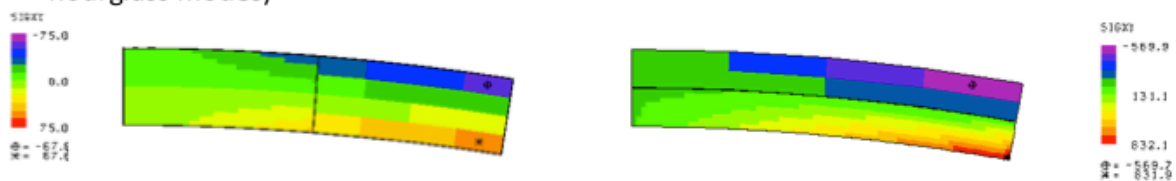
Mortar method mesh tying: linear interpolation of interface tractions, no modes are unresisted but cannot pass patch test for general curved interfaces

MPC mesh tying: no modes are unresisted but can produce unrealistic results due to overconstraining

Fig. 16.4 Results for a continuum beam that includes mesh tying subjected to pure bending.

16.2 Contact Search

The contact search algorithm is a logical component of the overall contact capability. Much of the reason to consider search as a separate component is due to a need to revisit and replace algorithms as they demonstrate better performance. As problem sizes grow there is an increasing computational cost of this aspect of computational solid mechanics, particularly on distributed memory (parallel) computers.

As a way of introducing the concepts inherent in contact search algorithms, we recognize the similarity of contact search to many other other problems in the simulation domain (e.g., the video gaming industry). In this more abstract sense, a significant part of the contact search algorithm is a proximity determination of one object with respect to another. Collision detection, as it is also referred to, is computationally intensive but also studied thoroughly to obtain the best performance possible. Thus the contact algorithm in Sierra/SolidMechanics is comprised of the more general proximity search followed by the much more specific detailed detection of contact in the context of a discrete Finite Element method.

16.2.1 Proximity search algorithms

Although various proximity search algorithms have been developed over the years, those that have been used in Sierra/SolidMechanics are discussed. Proximity search algorithms deal with bounding boxes. Construction of bounding boxes are straightforwardly computed as the vector of min/max coordinates of the volume swept by the predicted motion of either nodes or faces over a time step. Specifically, over the time step $(t \rightarrow t + \Delta t)$, the axis-aligned bounding box for node I is computed as follows:

$$aabb(N_I) = \min(\varphi(\mathbf{X}_I, t), \varphi(\mathbf{X}_I, t + \Delta t)) , \ \max(\varphi(\mathbf{X}_I, t), \varphi(\mathbf{X}_I, t + \Delta t)). \tag{16.9}$$

The resulting data of an axis aligned bounding box calculation is the absolute minimum amount of data (min and max x, y, z coordinates) representing a contact entity. The example expressed in (16.9) is for a node; with the extension to a face being straightforwardly computed as:

$$aabb(F_M) = \min(aabb(N_J), \ J = \text{nodes of face } M) , \ \max(aabb(N_J), \ J = \text{nodes of face } M). \tag{16.10}$$

This allows a simplification or reduction of the contact problem to the more general proximity detection based solely on axis-aligned bounding boxes of contact entities. Many of the structural modeling capabilities within Sierra/SM are converted to these primitives (i.e., nodes and faces). Beams and shells, for example, do not explicitly represent volume, but a volume is inferred with ancillary data such as thickness. These elements are converted to contact primitives by lofting the finite element geometry explicitly to volumetric discretizations.

16.2.2 Parallel search algorithms

The strategy for computational simulation of contact on distributed memory architectures (parallel computing) is to decompose the contact problem in a distributed manner among the compute nodes. Typically known as domain decomposition methods, there are several that are directly applicable to the contact problem. Inertial decomposition and recursive coordinate bisection (RCB) decomposition are two geometric-based algorithms that are examples. This geometry-based decomposition approach is depicted graphically in Fig. 16.5.
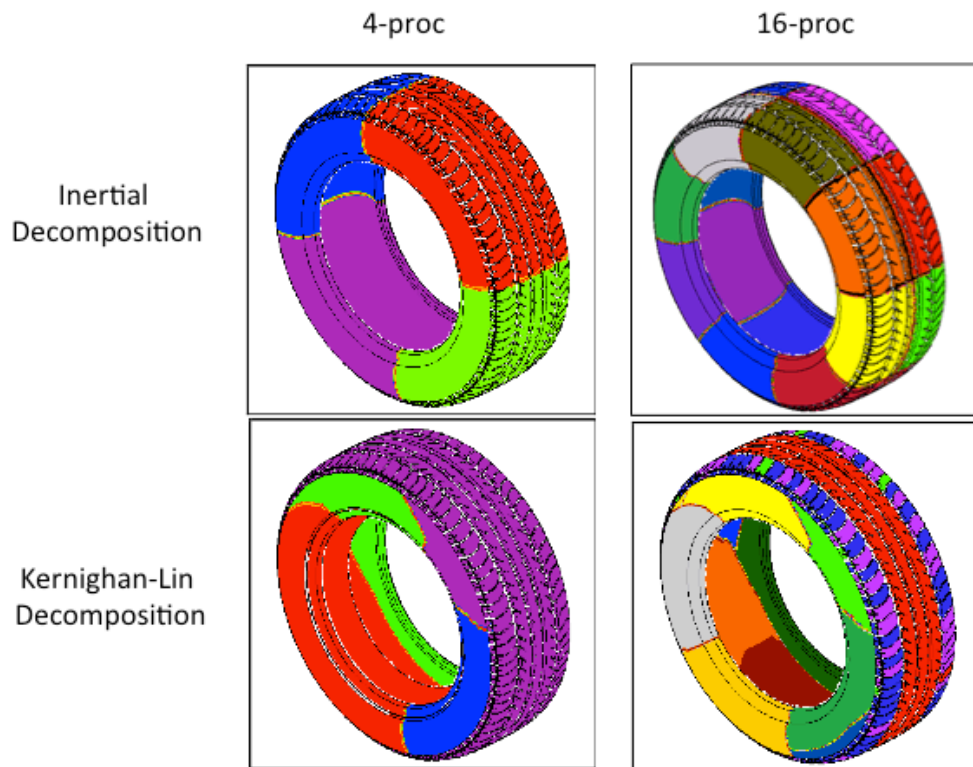

Fig. 16.5 Simple illustration of the domain decomposition for contact problems.

The proximity search algorithm thus performs with parallel scalability and serial efficiently on each processor.

16.2.3 Contact kinematics

Recall that the output of the proximity search is a collection of bounding box pairs whose volumes overlap. The contact entities associated with the bounding boxes are then considered for contact in what we call a detailed search. Detailed searching is a term applied to computing the contact kinematic quantities associated with either the node-face or face-face algorithm.

Closest point projection for node-face contact

The underpinning of the node-face contact approach is the choice of a set of points at which to apply the Kuhn-Tucker conditions. Once this choice is made, it follows that a contact point must be determined for each node. The closest point projection is the name given to this calculation, which is simply the point on the opposing surface that minimizes the gap, i.e.,

$$g_N(\mathbf{X_I}, t) = \min_{\mathbf{Y} \in \{*facetsonsideAsurface*\}} \|\varphi(\mathbf{X_I}) - \varphi(\mathbf{Y})\| * sign * (g_N). \qquad (16.11)$$

This can be seen simply as the discrete form of the gap function expressed in (16.3). Immediately with the discrete surface *not* possessing C_1 continuity, the closest point projection is no longer unique. At and around edges and corners are the regions on the side A surface where the issues arise. Fig. 16.6 depicts a simple illustration of the non-uniqueness encountered.
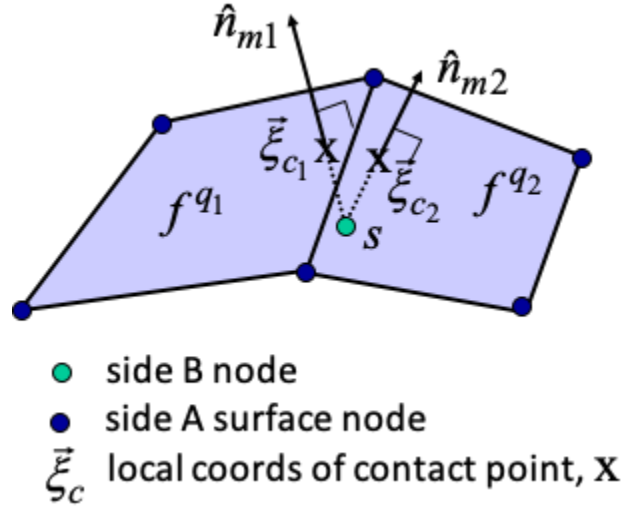

Fig. 16.6 Simple illustration of the non-uniqueness in the closest point projection.

Minimum volume overlap for face-face contact

The face-face contact approach seeks to avoid these issues by considering the volume overlap between the discrete sides of the interface.

This page left blank

17 Boundary Conditions

This chapter describes the theoretical and mathematical basis for some common boundary conditions.

17.1 Distributed Force and Moment

17.1.1 Boundary Condition Purpose

The purpose of the distributed force boundary condition is to distribute a known set of forces and moments onto a meshed body of N nodes in a smooth manner. The force distribution is formulated to have the following properties:

- The provided force distribution exactly reproduces three XYZ net target translational forces and three XYZ net target moments.

- The distribution avoids concentrated forces that may cause high local deformation.

- The force distribution uses translational forces only. Net moments are applied via translational force couples.

There are likely infinitely many force distributions that meet the above properties. The distributed force BC aims to find and apply at least one reasonable such distribution.

17.1.2 Boundary Condition Implementation

The distributed force boundary condition applies nodal forces constructed by a linear combination of six assumed distributions. Each of these force distributions provide a contribution predominately aligned with each of the net forces and net moments. The distributions are essentially a weight for dividing a net global force over the N-node set.

Three translational force distributions $\mathbf{D}_x$, $\mathbf{D}_y$ and $\mathbf{D}_z$ are given in (17.1). The x, y, and z subscripts denote the force direction, and $\hat{\mathbf{x}} = (1, 0, 0)$, $\hat{\mathbf{y}} = (0, 1, 0)$, and $\hat{\mathbf{z}} = (0, 0, 1)$. The Ith subscript denotes the Ith node in the N-node set. m_I is the mass at node I. The translational force distributions are unitless.

$$\mathbf{D}_{xI} = \left(\frac{m_I}{\sum_{I=1}^{N} m_I} \right) \hat{\mathbf{x}}$$

$$\mathbf{D}_{yI} = \left(\frac{m_I}{\sum_{I=1}^{N} m_I} \right) \hat{\mathbf{y}} \qquad (17.1)$$

$$\mathbf{D}_{zI} = \left(\frac{m_I}{\sum_{I=1}^{N} m_I} \right) \hat{\mathbf{z}}$$

Note, the translational distributions have an identical shape to a gravity load. This choice to weight the nodal forces by mass is somewhat arbitrary, but does a good job of minimizing artificial force concentration. Also, note that a gravity load applies no net moment about the center of mass of the node set.

The moment distributions apply a net moment about the node set center of mass. The center of mass of the node set $\mathbf{C}$ is calculated by (17.2). $\mathbf{p}_I$ represents the coordinates of node I. The three trial moment distributions $\mathbf{D}'_{rx}, \mathbf{D}'_{ry}, \mathbf{D}'_{rz}$ are given by (17.3). The rx, ry, and rz subscripts denote the torques about $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{z}}$.

$$\mathbf{C} = \frac{\sum_{I=1}^{N} m_I \mathbf{p}_I}{\sum_{I=1}^{N} m_I} \tag{17.2}$$

$$\begin{aligned}
\mathbf{D}'_{rxI} &= m_I(\hat{\mathbf{x}} \times (\mathbf{p}_I - \mathbf{C})) \\
\mathbf{D}'_{ryI} &= m_I(\hat{\mathbf{y}} \times (\mathbf{p}_I - \mathbf{C})) \\
\mathbf{D}'_{rzI} &= m_I(\hat{\mathbf{z}} \times (\mathbf{p}_I - \mathbf{C}))
\end{aligned} \tag{17.3}$$

The constructed trial moment distributions may produce a net translational force. This is corrected by first computing the net translational force produced by each trial moment distribution, and subtracting off a scaled translational force distribution. The corrected pure moment distributions are given in (17.4).

$$\begin{aligned}
\mathbf{D}_{rxI} &= \mathbf{D}'_{rxI} - \left(\sum_{I=1}^{N} \mathbf{D}'_{rxI} \cdot \hat{\mathbf{x}}\right)\mathbf{D}_{xI} - \left(\sum_{I=1}^{N} \mathbf{D}'_{rxI} \cdot \hat{\mathbf{y}}\right)\mathbf{D}_{yI} - \left(\sum_{I=1}^{N} \mathbf{D}'_{rxI} \cdot \hat{\mathbf{z}}\right)\mathbf{D}_{zI} \\
\mathbf{D}_{ryI} &= \mathbf{D}'_{ryI} - \left(\sum_{I=1}^{N} \mathbf{D}'_{ryI} \cdot \hat{\mathbf{x}}\right)\mathbf{D}_{xI} - \left(\sum_{I=1}^{N} \mathbf{D}'_{ryI} \cdot \hat{\mathbf{y}}\right)\mathbf{D}_{yI} - \left(\sum_{I=1}^{N} \mathbf{D}'_{ryI} \cdot \hat{\mathbf{z}}\right)\mathbf{D}_{zI} \\
\mathbf{D}_{rzI} &= \mathbf{D}'_{rzI} - \left(\sum_{I=1}^{N} \mathbf{D}'_{rzI} \cdot \hat{\mathbf{x}}\right)\mathbf{D}_{xI} - \left(\sum_{I=1}^{N} \mathbf{D}'_{rzI} \cdot \hat{\mathbf{y}}\right)\mathbf{D}_{yI} - \left(\sum_{I=1}^{N} \mathbf{D}'_{rzI} \cdot \hat{\mathbf{z}}\right)\mathbf{D}_{zI}
\end{aligned} \tag{17.4}$$

The total forces to be applied are a weighted sum of the six force distributions $\mathbf{D}_{rx}, \mathbf{D}_{ry}, \mathbf{D}_{rz}, \mathbf{D}_x$, $\mathbf{D}_y$, and $\mathbf{D}_z$. Note the moment force $\mathbf{D}_{rx}, \mathbf{D}_{ry}$, and $\mathbf{D}_{rz}$ distribution values have units of mass times length while the translational force distributions $\mathbf{D}_x, \mathbf{D}_y, \mathbf{D}_z$ are unitless.

The translational force distributions apply no moment. The corrected moment force distributions apply no net translational force. However, a moment distribution applied in one direction may cause a secondary moment in different direction. These moment coupling terms are computed in (17.5). The M_{xy} term, as an example, represents the net moment generated about $\hat{\mathbf{y}}$, given that a

nodal force distribution $\mathbf{D}_{rx}$ is applied.

$$M_{xx} = \sum_{I=1}^{N} (((\mathbf{p}_I - \mathbf{C}) \times \mathbf{D}_{rxI}) \cdot \hat{\mathbf{x}})$$

$$M_{xy} = \sum_{I=1}^{N} (((\mathbf{p}_I - \mathbf{C}) \times \mathbf{D}_{rxI}) \cdot \hat{\mathbf{y}})$$

$$M_{xz} = \sum_{I=1}^{N} (((\mathbf{p}_I - \mathbf{C}) \times \mathbf{D}_{rxI}) \cdot \hat{\mathbf{z}})$$

$$M_{yx} = \sum_{I=1}^{N} (((\mathbf{p}_I - \mathbf{C}) \times \mathbf{D}_{ryI}) \cdot \hat{\mathbf{x}})$$

$$M_{yy} = \sum_{I=1}^{N} (((\mathbf{p}_I - \mathbf{C}) \times \mathbf{D}_{ryI}) \cdot \hat{\mathbf{y}}) \qquad (17.5)$$

$$M_{yz} = \sum_{I=1}^{N} (((\mathbf{p}_I - \mathbf{C}) \times \mathbf{D}_{ryI}) \cdot \hat{\mathbf{z}})$$

$$M_{zx} = \sum_{I=1}^{N} (((\mathbf{p}_I - \mathbf{C}) \times \mathbf{D}_{rzI}) \cdot \hat{\mathbf{x}})$$

$$M_{zy} = \sum_{I=1}^{N} (((\mathbf{p}_I - \mathbf{C}) \times \mathbf{D}_{rzI}) \cdot \hat{\mathbf{y}})$$

$$M_{zz} = \sum_{I=1}^{N} (((\mathbf{p}_I - \mathbf{C}) \times \mathbf{D}_{rzI}) \cdot \hat{\mathbf{z}})$$

It is observed (but not proven) that the 3×3 moment coupling matrix $\mathbf{M}$ is symmetric. Approximate symmetry of the moment coupling matrix is assumed during the solution process. If the moment coupling matrix is not symmetric, then the net moments applied by the distributed force and moment boundary condition may be off in an amount proportional to the lack of symmetry.

To achieve the target net forces and moments $\mathbf{b}$, (17.6) is solved to find the force distribution multipliers $\mathbf{w}$. b_x, b_y, and b_z have units of force and b_{rx}, b_{ry}, and b_{rz} have units of moment. For the units to work out, w_x, w_y, and w_z have units of force, while w_{rx}, w_{ry}, and w_{rz} have units of one over time squared.

$$
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & M_{xx} & 0.5*(M_{xy}+M_{yx}) & 0.5*(M_{xz}+M_{zx}) \\
0 & 0 & 0 & 0.5*(M_{xy}+M_{yx}) & M_{yy} & 0.5*(M_{yz}+M_{zy}) \\
0 & 0 & 0 & 0.5*(M_{xz}+M_{zx}) & 0.5*(M_{yz}+M_{zy}) & M_{zz}
\end{bmatrix}
\begin{bmatrix}
w_x \\ w_y \\ w_z \\ w_{rx} \\ w_{ry} \\ w_{rz}
\end{bmatrix}
=
\begin{bmatrix}
b_x \\ b_y \\ b_z \\ b_{rx} \\ b_{ry} \\ b_{rz}
\end{bmatrix}
\qquad (17.6)
$$

The actual final force $\mathbf{F}$ to apply to each node I is given by (17.7).

$$\mathbf{F}_I = w_x\mathbf{D}_{xI} + w_y\mathbf{D}_{yI} + w_z\mathbf{D}_{zI} + w_{rx}\mathbf{D}_{rxI} + w_{ry}\mathbf{D}_{ryI} + w_{rz}\mathbf{D}_{rzI} \tag{17.7}$$

17.1.3 Limitations and Special Cases

The distributed force and moment boundary conditions apply moments via translational force couples to a set of nodes. Special cases of node sets exist such that the application of distributed moments is not well-posed.

One example of these use cases is if the node set contains a single node, or a set of nodes in the same exact position. In this scenario, no force applied to the nodes will result in any moments. This manifests as a zero matrix for the moment coupling terms given by (17.5). The resulting zero sub-matrix in (17.6) renders its solution impossible. Such a case should be avoided. However, if encountered in the code, the distributed moments will be ignored.

A second pathological node arrangement is a collinear set of nodes. No set of forces on a collinear node set can produce a torque around the collinear axis. Such a case will manifest as a singular system in (17.6). This node configuration should be avoided. However, if detected, the target torque around the collinear node axis will be ignored and the other two orthogonal moments returned correctly.

17.2 Inertia Relief

The inertia relief boundary condition is used to balance the free body diagram of forces acting on a body such that the net force external force acting on the body is zero. The inertia relief boundary condition heavily leverages the distributed force and moment capability Section 17.1.

Inertia relief computes the net external forces $\mathbf{F}_{\text{sum}}$ and moments $\mathbf{M}_{\text{sum}}$ acting on a body. These net external forces include forces from pressures, tractions, gravity, and other boundary conditions. $\mathbf{F}_{\text{sum}}$ and $\mathbf{M}_{\text{sum}}$ are computed using Equations (17.8) and (17.9). The I index is the Ith node in the set. $\mathbf{F}_{ext}$ is the translation external force acting on a node, $\mathbf{M}_{ext}$ is the external moment acting on a node, and $\mathbf{p}$ represents the coordinates of the node. The moments on the body are computed around the body center of mass C as computed in (17.2).

$$\mathbf{F}_{\text{sum}} = \sum \mathbf{F}_{ext_I} \tag{17.8}$$

$$\mathbf{M}_{\text{sum}} = \sum ((\mathbf{p}_I - \mathbf{C}) \times \mathbf{F}_{ext_I} + \mathbf{M}_{ext_I}) \tag{17.9}$$

In order to compute the inertia relief forces, the distributed force boundary condition is leveraged,

ultimately solving (17.6) for the **b** given in (17.10).

$$\mathbf{b} = \begin{bmatrix} -F_{\text{sum}_x} \\ -F_{\text{sum}_y} \\ -F_{\text{sum}_z} \\ -M_{\text{sum}_x} \\ -M_{\text{sum}_y} \\ -M_{\text{sum}_z} \end{bmatrix} \qquad (17.10)$$

17.3 Viscous Damping

17.3.1 Rigid Body Invariant Damping

The rigid body invariant damping option heavily leverages the inertia relief boundary condition Section 17.2. Rigid body invariant damping automatically applies an inertia relief boundary condition that counterbalances just the damping forces being applied by the viscous damping BC. This counterbalancing force ensures the total applied damping has no effect on the rigid body motion of parts and thus only effects the vibration models of the part.

This page left blank

A. Known Issues

- Many more references could be given in Section 1.1.

- Many, if not all, Epic materials models have no theory documentation. We recommend searching the open literature.

- Particle methods, such as Smooth Particle Hydrodynamics, are undocumented.

- Representative volume elements are undocumented. However a reference paper does exist.

This page left blank

References

[1] T. Belytschko and I. Leviathan. Physical stabilization of the 4-node shell element with one point quadrature. *Computer Methods in Applied Mechanics and Engineering*, 113:321–350, 1994.

[2] T. Belytschko and I. Leviathan. Projection schemes for one-point quadrature shell elements. *Computer Methods in Applied Mechanics and Engineering*, 115:277–286, 1994.

[3] T. Belytschko and C.-S Tsay. Explicit algorithms for nonlinear dynamics of shells. *Nonlinear finite element analysis of plates and shells*, pages 209–231, 1981.

[4] D.J. Benson. Stable time step estimation for multi-material eulerian hydrocodes. *Computer Methods in Applied Mechanics and Engineering*, 167:191–205, 1998.

[5] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen differenzengleichungen der mathematischen physik. *Mathematische Annalen*, 100:32–74, 1928.

[6] G.R. Cowper. The shear coefficient in Timoshenko's beam theory. *Journal of Applied Mechanics*, 33:335–340, 1966.

[7] J.W. Daniel. Convergence of the conjugate gradient method with computationally convenient modifications. *Numerische Mathematik*, 10:125–131, 1967.

[8] J.W. Daniel. The conjugate gradient method for linear and nonlinear operator equations. *SIAM Journal on Numerical Analysis*, 4:10–26, 1967.

[9] C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen. FETI-DP: a dual-primal unified FETI method – part i: a faster alternative to the two-level FETI method. *International Journal for Numerical Methods in Engineering*, 50:1523–1544, 2001.

[10] C. Farhat, M. Lesoinne, and K. Pierson. A scalable dual-primal domain decomposition method. *Numerical Linear Algebra with Applications*, 7:687–714, 2000.

[11] D.P. Flanagan and T. Belytschko. A uniform strain hexahedron and quadrilateral with orthogonal hourglass control. *International Journal for Numerical Methods in Engineering*, 17:679–706, 1981. doi:10.1002/nme.1620170504.

[12] D.P. Flanagan and T. Belytschko. Simultaneous relaxation in structural dynamics. *Journal of the Engineering Mechanics Division, ASCE*, 107:1039–1055, 1981.

[13] D.P. Flanagan and T. Belytschko. Eigenvalues and stable time steps for the uniform strain hexahedron and quadrilateral. *ASME Journal of Applied Mechanics*, 51:35–40, 1984.

[14] R. Fletcher and C.M. Reeves. Function minimization by conjugate gradients. *The Computer Journal*, 7:149–153, 1964.

[15] Y.C. Fung. *Foundations of Solid Mechanics*. Prentice-Hall, Inc, 1965.

[16] Y.C. Fung. *A First Course in Continuum Mechanics*. Prentice-Hall, Inc, 2nd edition, 1977.

[17] G.L. Goudreau. Evaluation of discrete methods for the linear dynamic response of elastic and viscoelastic solids. Technical Report 69–15, U.C. Berkeley, Department of Civil Engineering, 1970.

[18] A. Greenbaum. Behavior of slightly perturbed Lanczos and conjugate-gradient recurrences. *Linear Algebra Appl.*, 113:7–63, 1989.

[19] L. Grippo and S. Lucidi. A globally convergent version of the Polak-Ribière conjugate gradient method. *Math. Program.*, 78:375–391, 1997.

[20] M.E. Gurtin. *An Introduction to Continuum Mechanics*. Mathematics in Science and Engineering. Academic Press, Inc., 1982.

[21] M.E. Gurtin, E. Fried, and L. Anand. *The Mechanics and Thermodynamics of Continua*. Cambridge University Press, 2010.

[22] R.W. Hamming. *Numerical Methods for Scientists and Engineers*. Dover, 1973.

[23] M.R. Hestenes. *Conjugate Direction Methods in Optimization*. Springer-Verlag, Berlin, 1980.

[24] M.R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand.*, 49:409–436, 1952.

[25] H.M. Hilber, T.J.R. Hughes, and R.L. Taylor. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering and Structural Dynamics*, 5:283–292, 1977.

[26] T.J.R. Hughes. *The Finite Element Method–Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, Inc, Englewood Cliffs, NJ, 1987.

[27] T.J.R. Hughes. *The Finite Element Method: Linear static and dynamic finite element analysis*. Dover, 2000. Reprint of "The Finite Element Method", Prentice-Hall, 1987.

[28] B.M. Irons. Applications of a theorem on eigenvalues to finite element problems. Technical Report CR/132/70, University of Wales, Department of Civil Engineering, Swansea, U.K., 1970.

[29] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, Sep 1976. doi:10.1107/S0567739476001873.

[30] S.W. Key and C.C. Hoff. An improved constant membrane and bending stress shell element for explicit transient dynamics. *Computer Methods in Applied Mechanics and Engineering*, 124(1–2):33–47, 1995. doi:10.1016/0045-7825(95)00785-Y.

[31] J.R. Koteras and A. S. Gullerud. Presto user's guide version 1.05. Technical Report SAND2003-1089, Sandia National Laboratories, Albuquerque, NM, April 2003.

[32] J.R. Koteras and R.B. Lehoucq. Estimating the critical time-step in explicit dynamics using the Lanczos method. *International Journal for Numerical Methods in Engineering*, 69(13):2780–2788, 2006.

[33] R.D. Krieg and S.W. Key. Transient shell response by numerical time integration. *International Journal for Numerical Methods in Engineering*, 7:273–286, 1973.

[34] W.M. Lai, D. Rubin, and E. Krempl. *Introduction to Continuum Mechanics*. Pergamon Press, 3rd edition, 1993.

[35] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Research of the National Bureau of Standards*, 45(4):255–282, October 1950. Research Paper 2133.

[36] D.G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. International Series in Operations Research & Management Science. Springer, 2008.

[37] P. Madabhusi-Raman and J.F. Davalos. Static shear correction factor for laminated rectangular beams. *Composites Part B: Engineering*, 27(3):285–293, 1996. doi:10.1016/1359-8368(95)00014-3.

[38] L.E. Malvern. *Introduction to the Mechanics of a Continuous Medium*. Prentice-Hall, Inc, 1969. pages 226-228.

[39] J.E. Marsden and T.J.R. Hughes. *Mathematical Foundations of Elasticity*. Dover, 1983.

[40] N.M. Newmark. A method of computation for structural dynamics. *Journal of Engineering Mechanics, ASCE*, 85(EM3):67–94, 1959.

[41] J. Nocedal. *Theory of Algorithms for Unconstrained Optimization*, pages 199–242. Springer, 1991.

[42] B. Nour-Omid. The Lanczos algorithm for solution of large generalized eigenproblems. In *The Finite Element Method: Linear static and dynamic finite element analysis,Thomas J.R. Hughes*, pages 582–600. Dover, 2000.

[43] J.T. Ostien, J.W. Foulk, A. Mota, and M.G. Veilleux. A 10-node composite tetrahedral finite element for solid mechanics. *International Journal for Numerical Methods in Engineering*, 2016.

[44] B.N. Parlett. *The symmetric eigenvalue problem*. Number 20 in Classic in Applied Mathematics. SIAM, 1998. Reprint of "The symmetric eigenvalue problem", Prentice-Hall, 1980.

[45] W.D. Pilkey and W. Wunderlich. *Mechanics of Structures: Variational and Computational Methods*. CRC Press, 1994.

[46] J.R. Shewchuck. An introduction to the conjugate gradient algorithm without all of the agonizing pain. Technical Report, Carnegie Mellon University, Pittsburg, PA, 1994.

[47] P. Thoutireddy, J.F. Molinari, E.A. Repetto, and M. Ortiz. Tetrahedral composite finite elements. *International Journal for Numerical Methods in Engineering*, 53(6):1337–1351, 2002.

[48] S. Timoshenko and J.N. Goodier. *Theory of Elasticity*. McGraw Hill Book Company, 3rd edition, 1970.

[49] A. Toselli and O. Widlund. *Domain Decomposition Methods – Algorithms and Theory*. Springer Series in Computational Mathematics. Springer, 2005.

[50] Y.Guo, M. Ortiz, T. Belytschko, and E.A. Repetto. Triangular composite finite elements. *International Journal for Numerical Methods in Engineering*, 47(1-3):287–316, 2000.