CHAPTER 6
SOLUTION METHODS AND STABILITY

by Ted Belytschko and Brian Moran
Northwestern University
Copyright 1996

6.1 INTRODUCTION

This Chapter describes solution procedures for nonlinear finite element discretizations. In addition, methods for examining the physical stability of solutions and the stability of solution procedures are described.

The first part of the chapter describes time integration, the procedures used for integrating the discrete momentum equation and other time dependent equations in the system, such as the constitutive equation. We begin with the simplest of methods, the central difference method for explicit time integration. Next the family of Newmark $\beta$-methods, which encompass both explicit and implicit methods, are described. Explicit and implicit methods are compared and their relative advantages described. As part of implicit methods, the solution of equilibrium equations is also examined.

A critical step in the solution of implicit systems and equilibrium problems is the linearization of the governing equations. Linearization procedures for the equations of motion, and as a special case, the equilibrium equations are described.

6.2 EXPLICIT METHODS

In this Section the major features of explicit and implicit time integration methods for the discretized momentum equation and solution methods for the discrete equilibrium equations are described. The methods are described in the context of Lagrangian meshes, but can be extended to Eulerian and ALE meshes with some techniques described in Chapter 7. The description of the solution procedures of equilibrium problems is combined with the description of implicit procedures for dynamic problems, because, as we show later, the methodologies are almost identical; the solution of a static problem by an implicit method only requires that the inertial term be dropped.

To illustrate the major features of explicit and implicit methods for time integration, the solution of the equations of motion is first considered for rate-independent materials. In this class of equations, we can avoid some of the complications that arise in the treatment of rate-dependent materials but still illustrate the most important properties of explicit and implicit methods. We will first describe explicit and implicit methods using only a single time integration formula: the central difference method for explicit time integration and the Newmark $\beta$-methods for implicit integration. In Section X, other time integration formulas are considered.
6.2.1. Central Difference Method. The central difference method is among the most popular of the explicit methods in computational mechanics and physics. It has already been discussed in Chapter 2, where it was chosen to demonstrate some nonlinear solutions in one dimension. The central difference method is developed from central difference formulas for the velocity and acceleration. We consider here its application to Lagrangian meshes with rate-independent materials. Geometric and material nonlinearities are included, and in fact have little effect on the time integration algorithm.

For the purpose of developing this and other time integrators we will use the following notation. Let the time of the simulation \( 0 \leq t \leq t_E \) be subdivided into time intervals, or time steps, \( \Delta t^n \), \( n = 1 \) to \( n_{TS} \) where \( n_{TS} \) is the number of time steps and \( t_E \) is the end-time of the simulation; \( \Delta t^n \) is also called the \( n \)th time increment. The variables at any time step are indicated by a superscript; thus \( t^n \) is the time at time step \( n \), \( t^0 = 0 \) is the beginning of the simulation and \( d^n = d(t^n) \) is the matrix of nodal displacements at time step \( n \).

We consider here an algorithm where the time step varies with time. This is necessary in most practical calculations since the stable time step will change as the mesh deforms. For this purpose, we define time increments \( n \) and \( n + 1/2 \) by

\[
\Delta t^n = t^n - t^{n-1} \quad \Delta t^{n+1/2} = \frac{1}{2}(\Delta t^n + \Delta t^{n+1})
\]  

(6.2.1)

The time step \( n + 1/2 \) is often called the midpoint time step.

The central difference formula for the velocity is

\[
\dot{d}^{n+1/2} = \frac{1}{\Delta t^{n+1/2}}(d^{n+1} - d^n), \quad d^{n+1} = d^n + \Delta t^{n+1/2}v^{n+1/2} \tag{6.2.2a,b}
\]

where the second equation gives the corresponding integration equation which is obtained by a rearrangement of the first. The acceleration is given by

\[
\ddot{d}^n \equiv a^n = \frac{1}{\Delta t^n} \left(v^{n+1/2} - v^{n-1/2}\right) \quad v^{n+1/2} = v^{n-1/2} + \Delta t^n a^n \tag{6.2.3a}
\]

As can be seen from the above, the velocities are defined at the midpoints of the time intervals, or at half-steps using the midpoint time step. By substituting (6.2.2a) and its counterpart for the previous time step into (6.2.3), the acceleration can be expressed directly in terms of the displacements

\[
\dot{d}^n = \frac{\Delta t^{n-1/2}(d^{n+1} - d^n) - \Delta t^{n+1/2}(d^n - d^{n-1})}{\Delta t^n \Delta t^{n-1/2} \Delta t^{n+1/2}} \tag{6.2.3b}
\]

For the case of equal time steps the above reduces to
\[ \ddot{\mathbf{d}}^n \equiv \mathbf{a}^n = \frac{\left( \mathbf{d}^{n+1} - 2\mathbf{d}^n + \mathbf{d}^{n-1} \right)}{(\Delta t^n)^2} \] 

(6.2.3c)

This is the well known central difference formula for the second derivative of a function.

We now consider the time integration of the undamped equations of motion for rate-independent materials, Eq. (4.x.x.), which at time step \( n \) are given by

\[ \mathbf{M} \mathbf{a}^n = \mathbf{f}^n = \mathbf{f}^{ext}(\mathbf{d}^n, \mathbf{r}^n) - \mathbf{f}^{int}(\mathbf{d}^n) \] 

(6.2.4a)

subject to \( g_I(\mathbf{d}^n) = 0, \ I = 1 \) to \( n_c \)

(6.2.4b)

where (6.2.4b) is a generalized representation of the \( n_c \) displacement boundary conditions and other constraints on the model. These constraints are linear or nonlinear algebraic function of the nodal displacements. If the constraint involves integral or differential relationships, such as a dependence on the velocities, it can be put in the above form by using difference equations or a numerical approximation of the integral. The mass matrix in this expression is considered constant because as noted in Section X, it is time independent for a Lagrangian mesh. Methods for Eulerian and ALE meshes are discussed in Chapter 7. The internal and external nodal forces are functions of the nodal displacements and the time. The external loads are usually prescribed as functions of time; they may also be functions of the nodal displacements because they may depend on the configuration of the structure, as when pressures are applied to the surfaces which undergo large deformations. The dependence of the internal nodal forces on displacements is quite obvious: the nodal displacements determine the strains, which in turn determine the stresses and hence the nodal internal forces by Eq. (4.4.11). Internal nodal forces are generally not directly dependent on time, but there are situations of engineering relevance when this is the case; for example, when the temperature is prescribed as a function of time, the stresses and hence the internal nodal forces depend directly on time, but we have not indicated this time dependence in (6.2.4a).

The equations for updating the nodal velocities and displacements are obtained as follows. Substituting (6.2.4a) into (6.2.3b) gives

\[ \mathbf{v}^{n+\frac{1}{2}} = \Delta t^n \mathbf{M}^{-1} \mathbf{f}^n + \mathbf{v}^{n-\frac{1}{2}} \] 

(6.2.5)

At any time step \( n \), the displacements \( \mathbf{d}^n \) will be known. The nodal forces \( \mathbf{f}^n \) can be determined by using in sequence the strain-displacement equations, the constitutive equation and the relation for the nodal internal forces. Thus the entire right hand side of (6.2.5) can be evaluated, and (6.2.5) can be used to obtain \( \mathbf{v}^{n+\frac{1}{2}} \). The displacements \( \mathbf{d}^{n+1} \) can be then be determined by (6.2.2b).
The update of the nodal velocities and nodal displacements can be accomplished without solving any system equations provided that the mass matrix $M$ is diagonal. This is the salient characteristic of an explicit method: *in an explicit method, the time integration of the discrete momentum equations for a finite element model does not require the solution of any equations.*

In numerical analysis, integration methods are classified according to the structure of the time difference equation. The difference equations for first and second derivatives are written in the general forms

$$
\sum_{n=0}^{m} (\alpha_n d^n - \Delta t \beta_n d^n) = 0 \quad \sum_{n=0}^{n_S} (\alpha_n d^{n_S-n} - \Delta t^2 \beta_n d^n) = 0
$$

(6.2.6)

where $m$ is the number of steps in the difference equation. The difference formula for the first or second derivatives is called explicit if $\beta_m = 0$ or $\beta_m = 0$, respectively. From (6.2.3c) it can be seen that $\beta_0 = 0, \beta_1 = 1, \beta_2 = 0$, so the formula is explicit. Thus the difference formula is called explicit if the equation for the function at time step $n$ only involves the derivatives at previous time steps. Difference equations which are explicit according to this classification generally lead to solution schemes which require no solution of equations. In most cases there is no benefit in using explicit schemes which involve the solution of equations, so the use of such explicit schemes is rare. There are a few exceptions. For example, if the consistent mass is used with the central difference method, even though the difference equation is classified as explicit, a system of equations still need to be solved in the update.

### 6.2.2. Implementation.

A flow chart for explicit time integration of a finite element model with rate-independent materials is shown in Box 6.1. This flowchart generalizes the flowchart given in Chapter 2 by considering nonzero initial conditions, a variable time step and including elements which require more than one-point quadrature. The implemetation of the velocity update is broken into two substeps (6.2.3a) by

$$
v^n = v^{n-\frac{1}{2}} + \frac{1}{2} \Delta t^n a^n \quad v^{n+\frac{1}{2}} = v^n + \frac{1}{2} \Delta t^n a^n
$$

(6.2.6b)

This enables the energy balance to be checked at integer time steps.

The primary dependent variables in this flowchart are the velocities and the Cauchy stresses. Initial conditions must be given for the velocities, the Cauchy stresses, and all state variables of the materials in the model. The displacements are initially considered to vanish.
set $v^0, \sigma^0$, and initial values of other material state variables;

$d^0 = 0, n = 0, t = 0; \text{compute } M$

2. getf $(f^n, \Delta t^n)$

3. compute accelerations $a^n = M^{-1}f^n$

4. compute kinetic energy and check energy balance, see Section ??

5. update nodal velocities: $v^{n+\frac{1}{2}} = v^n + \frac{1}{2} \Delta t^n a^n$

6. enforce velocity boundary conditions:

   if node $I$ on $\Gamma_{v_i}, \dot{v}_{ii}^{n+\frac{1}{2}} = \bar{v}_i \left( x_i, t^{n+\frac{1}{2}} \right)$

7. update nodal displacements: $d^{n+1} = d^n + \Delta t^{n+\frac{1}{2}} v^{n+\frac{1}{2}}$

8. update nodal velocities: $v^{n+1} = v^{n+\frac{1}{2}} + \frac{1}{2} \Delta t^n a^n$

9. update counter and time: $n \leftarrow n + 1, t \leftarrow t + \Delta t$

10. output, if simulation not complete, go to 2

Subroutine getf $(f^n, \Delta t^n)$

0. initialization: $f^n = 0, \Delta t_{crit} = \infty$

1. compute external nodal forces $f^{ext,n}$ which are global

2. loop over elements $e$

   i. GATHER element nodal displacements and velocities

   ii. $f^{int,n}_e = 0$

   iii. loop over quadrature points $\xi_Q$

      1. if $n=0$, go to 4

      2. compute measures of deformation: $D^{n+\frac{1}{2}}(\xi_Q), F^n(\xi_Q), E^n(\xi_Q)$

      3. compute stress $\sigma^n(\xi_Q)$ by constitutive equation

      4. $f^{int,n}_e \leftarrow f^{int,n}_e + B^T \sigma^n \bar{w}_Q J_{\xi_Q}$

         END quadrature point loop

   iv. compute external nodal forces on element, $f^{ext,n}_e$

   v. $f^n_e = f^{ext,n}_e - f^{int,n}_e$

   vi. compute $\Delta t^{e}_{crit},$ if $\Delta t^{e}_{crit} < \Delta t_{crit}$ then $\Delta t_{crit} = \Delta t^{e}_{crit}$

   vii. SCATTER $f^n_e$ to global $f^n$

3. END loop over elements

4. $\Delta t^n = \Delta t_{crit}$
In this algorithm, the accelerations are first integrated to obtain the velocities. The integration of the velocities is broken into two half-steps so that the velocities are available at an integer step in the computation of the energy balance. The displacements are computed in each time step by integrating the velocities.

The main part of the procedure is the calculation of the nodal forces from the nodal displacements at a given time step, which is performed in `getf`. In this subroutine, the equations governing a continuum are used along with the gather/scatter procedures:

1. the nodal displacements of the element are extracted from the global matrix of nodal displacements by the “gather” operation;
2. the strain measures are computed at each quadrature point of the element;
3. the stresses are computed by the constitutive equation at each quadrature point;
4. the internal nodal forces are computed by integrating the product of the $\mathbf{B}$ matrix and the stresses over the domain of the element with the Cauchy stress;
5. the nodal forces of the element are scattered into the global array.

In the first time step, the strain measures and the stress are not computed. Instead, as shown in the flowchart, the initial stresses are used to obtain the internal nodal forces.

The flowchart shows the algorithm with the matrix form of the internal force computation, in which the stress tensor is stored as a square matrix and the $\mathbf{B}$ matrix is used. The change to the Voigt form only requires the use of a column matrix for the stresses and the $\mathbf{B}$ matrix, (4.5.14). Similarly, the internal force computation can be changed to the total Lagrangian format by replacing the discrete values of the integrand in step 10 by the integrands of (B4.8.2).

Most essential boundary conditions are easily handled in explicit methods. For example, if the velocities or displacements are prescribed as functions of time along any boundary, then the velocity/displacement boundary conditions can be enforced by setting the nodal velocities according to the data:

$$v_i^n = \bar{v}_i^n(x_i, t^n) \quad (6.2.7)$$

If the data is not available on the nodes, the least square procedure given in Section 2.4.5 can be used to fit the nodal values.

The velocity boundary conditions can also be enforced in local coordinate systems as shown in the Box 6.1. In that case, the equations of motion at these nodes must be expressed in the local coordinate system, so the nodal force components must be expressed in the local coordinate systems before assembly and time integration. The boundary condition is also enforced in the local coordinate system. The orientation of the local coordinate system may vary with time but the time integration formulas must then be modified to account for the additional terms in the equations of motion.
When essential boundary conditions are given as linear or nonlinear algebraic equations relating the displacements, the implementation is more complicated. One approach is to use a linearization of the constraint. Consider for example the nonlinear constraint

\[ g(d(t)) = 0 \]  \hspace{1cm} (6.2.8)

where \( g = \{g_I\} \).

As can be seen from the flowchart, an explicit method is easily implemented. Furthermore, explicit time integration is very robust, by which we mean that the explicit procedure seldom aborts due to failure of the numerical algorithm. The salient disadvantage of explicit integration, the price you pay for the simplicity of the method and its avoidance of the solution of equations, is the \textit{conditional stability of explicit methods}. If the time step exceeds a critical value \( \Delta t_{\text{crit}} \), the solution may grow unboundedly and will in any case be erroneous.

The \textit{critical time step} is also called the \textit{stable time step}. The critical time step for a model depends on the mesh and the material properties. For low order elements, we will show in Section X that the critical time step for linear response is given by

\[ \Delta t_{\text{crit}} = \min \frac{l_e}{c_e} \]  \hspace{1cm} (6.2.10)

where \( l_e \) is a characteristic length of element \( e \) and \( c_e \) the wavespeed within element \( e \). Thus the critical time step decreases with mesh refinement and increasing stiffness of the material. The cost of an explicit simulation is independent of the frequency content which is of interest and depends only on the size of the model and the time of the simulation relative to the critical time step given by (6.2.10).

The time step is calculated in the flowchart on an element basis. For each element, a critical time step is calculated, and if it is smaller than that calculated for all previous elements in that time step, it is reset. The theoretical justification for setting the critical time step on an element basis and other approaches are described in Section 6.??

For nonlinear problems, the wavespeed is based on the tangent moduli.

\section*{6.3 EQUILIBRIUM SOLUTIONS AND IMPLICIT TIME INTEGRATION.}

\subsection*{6.3.1. Equilibrium and Transient Problems.} We will combine the description of the solution of the equilibrium equations with time integration by implicit methods because they share many common features. To begin, we write the discrete momentum equation at time step \( n+1 \) in a form applicable to both equilibrium and dynamic problems:

\[ \text{(equation content)} \]
0 = \mathbf{r}(d^{n+1}, t^{n+1}) = s_D M \ddot{d}^{n+1} - f^{n+1} = s_D M \ddot{a}^{n+1} - f^{\text{ext}}(d^{n+1}, t^{n+1}) + f^{\text{int}}(d^{n+1}) \quad (6.3.1)

where \( s_D \) is a switch which is set by:

\[
s_D = \begin{cases} 
0 & \text{for a static (equilibrium) problem} \\
1 & \text{for a dynamic (transient) problem} 
\end{cases} \quad (6.3.2)
\]

The column matrix \( \mathbf{r}(d^{n+1}, t^{n+1}) \) is called a residual. In addition, the displacement boundary conditions must be met; these can be written as a set of \( n_c \) nonlinear algebraic equations

\[
g_I(d^{n+1}) = 0, \quad I = 1 \text{ to } n_c \quad (6.3.2b)
\]

Differential and integral constraints are put in discrete form by using discretizations of the derivatives and integrals, respectively. In most cases the displacement boundary conditions are linear algebraic equations, but we have written the general form (6.3.2b) because complex boundary conditions are often needed in nonlinear problems.

When the accelerations vanish or are negligible, a system is in equilibrium and the solution of (6.3.1) is called an \textit{equilibrium solution}. The equilibrium equations are given by (6.3.1) with \( s_D = 0 \):

\[
0 = \mathbf{r}(d^{n+1}, t^{n+1}) = f^{\text{int}}(d^{n+1}) - f^{\text{ext}}(d^{n+1}, t^{n+1}) \quad (6.3.3)
\]

In equilibrium problems, the residuals correspond to the out-of-balance forces; problems in which the accelerations can be neglected are called static problems.

The governing equations for both the implicit update of the equations of motion and the equilibrium equations are a set of nonlinear algebraic equations in the nodal displacements, \( d^{n+1} \). In equilibrium problems with rate-independent materials, \( t \) need not be the real time. Instead it can be any monotonically increasing parameter which describes the changing load. If the constitutive equation is a differential or integral equation, it must also be discretized in time to obtain a set of algebraic equations for the system.

\subsection*{6.3.2a. Newmark \( \beta \)-equations.} We will now show that the discrete equations obtained with an implicit time integrator applied to the dynamic equations are nonlinear algebraic equations in the unknowns \( d^{n+1} \). For this purpose we consider a popular class of time integrators called the Newmark \( \beta \)-method. In this time integration formula, the updated displacements and velocities are given by

\[
\begin{align*}
\mathbf{d}^{n+1} &= \ddot{d}^{n+1} + \beta \Delta t^2 \mathbf{a}^{n+1} \\
\ddot{d}^{n+1} &= \mathbf{d}^n + \Delta \mathbf{v}^n + \frac{\Delta t^2}{2} (1 - 2\beta) \mathbf{a}^n
\end{align*} \quad (6.3.4)\quad (6.3.5)
\]
Here $\beta$ and $\gamma$ are parameters whose useful values are summarized in Box 6.2. In writing the time integration formulas, we have segregated the historical values of the nodal variables, i.e. those pertaining to time step $n$, in $\mathbf{v}^n$ and $d^n$. The resulting formulas correspond to the predictor-corrector form given by Hughes and Liu( ). This segregation of the historical terms is convenient for the algebraic operations which follow and for the construction of explicit-implicit time integration procedures.

Equation (6.3.4) can be solved for the updated accelerations for $\beta > 0$, giving

$$a^{n+1} = \frac{1}{\beta \Delta t^2} \left( d^{n+1} - \tilde{d}^{n+1} \right)$$  \hfill (6.3.8)

Substituting (6.3.8 ) into (6.3.1) gives

$$0 = \mathbf{r} = \frac{s_D}{\beta \Delta t^2} \mathbf{M}(d^{n+1} - \tilde{d}^{n+1}) - \mathbf{f}^{ext}(d^{n+1}, t^{n+1}) + \mathbf{f}^{int}(d^{n+1})$$ \hfill (6.3.9)

which is a set of nonlinear algebraic equations in the nodal displacements $d^{n+1}$. Eq.(6.3.9) applies to both the static and dynamic problems. Therefore, in both cases the discrete problem is

$$\text{find } d^{n+1} \text{ so that } \mathbf{r}(d^{n+1}, t^{n+1}) = 0 \text{ subject to } \mathbf{g}(d^{n+1}) = 0$$ \hfill (6.3.10)

where $\mathbf{r}(d^{n+1}, t^{n+1})$ is given by Eq. (6.3.9).

6.3.3. **Newton’s Method.** The most widely used and most robust method for the solution of the nonlinear algebraic equations (6.3.9) is Newton’s method. The method is often called the Newton-Raphson method in computational mechanics. It is identical to the Newton method taught in introductory calculus courses.

We first illustrate the Newton method for one equation in one unknown $d$ without a displacement boundary condition. It is then generalized to an arbitrary number of unknowns. For the case of one unknown, (6.3.9) reduces to a single nonlinear algebraic equation

$$r(d^{n+1}, t^{n+1}) = \frac{s_D}{\beta \Delta t^2} \mathbf{M}(d^{n+1} - \tilde{d}^{n+1}) - f(d^{n+1}, t^{n+1}) = 0$$ \hfill (6.3.11)

The solution of (6.3.11) by Newton’s method is an iterative procedure. The iteration number is indicated by Greek subscript: $d_v$ is the $v$th iteration at time step $n+1$; the time step number $n+1$ will be omitted.
To begin the iterative procedure, a starting value for the unknown must be chosen; usually the value of the solution $d^n$ from the last time step is used, so $d_0 \equiv d^n$. Taking a Taylor expansion of the residual about the current value of the nodal displacement, $d_v$, and setting the resulting residual equal to zero:

$$0 = r(d_{v+1}, t^{n+1}) = r(d_v, t^n) + \frac{\partial r(d_v, t^n)}{\partial d} \Delta d + O(\Delta d^2)$$  \hspace{1cm} (6.3.12)

where

$$\Delta d = d_{v+1} - d_v,$$  \hspace{1cm} (6.3.12b)

$$r(d_v, t^n) = \frac{SD}{\beta \Delta t^2} M(d_v - \tilde{d}^{n+1}) + f^{\text{int}}(d_v) - f^{\text{ext}}(d_v, t^n)$$  \hspace{1cm} (6.3.13)

If the terms which are higher order in $\Delta d$ than linear are dropped, then (6.3.12) gives a linear equation for $\Delta d$:

$$0 = r(d_v, t^n) + \frac{\partial r(d_v, t^n)}{\partial d} \Delta d$$  \hspace{1cm} (6.3.14)

Note that in the Taylor expansion, the residual is written in terms of the time $t^{n+1}$. The time-dependence of the residual at constant nodal displacements is usually known. For example, if the tractions and body forces are given as functions of time, then the time dependent part of the nodal forces is known at time $t^{n+1}$ at the beginning of the iterations. Therefore the residual is always computed using the external forces at time $t^{n+1}$. The above is called a linear model or linearized model of the nonlinear equations. The linear model is the tangent to the nonlinear residual function; the process of obtaining the linear model is called **linearization**.

Equation (6.3.14) is often called a linear model of the nonlinear equations, Dennis and Schnabel (1983). Solving this linear model for the incremental displacements gives

$$\Delta d = -\left( \frac{\partial r(d_v)}{\partial d} \right)^{-1} r(d_v)$$  \hspace{1cm} (6.3.15)

In the Newton procedure, the solution to the nonlinear equation is obtained by iteratively solving a sequence of linear models (6.3.15). The new value for the unknown in each step of the iteration is obtained by rewriting Eq. (6.3.12b) as

$$d_{v+1} = d_v + \Delta d$$  \hspace{1cm} (6.3.16)

The procedure is illustrated in Fig. 6.1. The process is continued until the solution is obtained with the desired level of accuracy.
6.3.4. **Newton’s Method for n Unknowns.** The generalization of this procedure to \( n_{DOF} \) unknowns is accomplished by replacing the above scalar equations by matrix equations. The counterpart of Eq. (6.3.12) becomes

\[
r(d) + \frac{\partial r(d)}{\partial d} \Delta d + O(\Delta d^2) = 0
\]

or

\[
r_a(d) + \sum_{b=1}^{n_{DOF}} \frac{\partial r_a(d)}{\partial d_b} \Delta d_b + O(\Delta d_b)^2 = 0
\]

(6.3.17)

The matrix \( \frac{\partial r}{\partial d} \) is called the Jacobian matrix and will be denoted by \( A \):

\[
A = \frac{\partial r}{\partial d}, \quad \text{or} \quad A_{ab} = \frac{\partial r_a}{\partial d_b}
\]

(6.3.18)

Using (6.3.17) and dropping terms of higher order than linear in \( \Delta d \), (6.3.16) can be rewritten as

\[
r + A \Delta d = 0
\]

(6.3.19)
which is the linear model of the nonlinear equations. The linear model is difficult to picture for problems with more than one unknown, since $\mathbf{r}(\mathbf{d})$ maps $\mathbb{R}^n$ to $\mathbb{R}^n$. Figure 6.2 shows the first component of the residual for a function of two unknowns. The linear model is a plane tangent to the nonlinear function $r_1(d_1, d_2)$. The other residual component is another nonlinear function $r_2(d_1, d_2)$, which is not drawn.

![Figure 6.2. Depiction of a residual component $r_1$ as a function of $d_1$ and $d_2$ and the tangent plane.](image)

The increment in the nodal displacements in the Newton iterative procedure is obtained by solving (6.3.18), which gives

$$\Delta \mathbf{d} = -\mathbf{A}^{-1}\mathbf{r}(\mathbf{d}_u, t^{n+1})$$

(6.3.20)

The increment in the nodal displacements is obtained from this system of linear algebraic equations. The solution of these equations is discussed in Section X. Once the increments in nodal displacements have been obtained, the new values of the nodal displacements are obtained by
The new displacement is checked for convergence, see Section 6.3.7. If the convergence criterion is not met, a new linear model is constructed and used to find another increment in the nodal displacements. The procedure is repeated until the convergence criterion is satisfied.

In computational mechanics, the Jacobian is called the effective tangent stiffness matrix and the contributions of the inertial, internal and external nodal forces are linearized separately. From (6.3.9) we can write

\[
A = \frac{\partial r}{\partial d} = \frac{s_D}{\beta \Delta t^2} M + \frac{\partial f^{\text{int}}}{\partial d} - \frac{\partial f^{\text{ext}}}{\partial d} \tag{6.3.22}
\]

where we have used the fact that the mass matrix in a Lagrangian mesh is constant in time and (6.3.4). The Jacobian of the internal nodal forces is called the tangent stiffness matrix and will be denoted by \( K^{\text{int}} \):

\[
K^{\text{int}}_{ab} = \frac{\partial f^{\text{int}}}{\partial d_b}, \quad K^{\text{int}}_{ij} = \frac{\partial f^{\text{int}}}{\partial u_{ij}} \quad K^{\text{int}} = \frac{\partial f^{\text{int}}}{\partial d} \tag{6.3.23}
\]

and it is shown above in three forms.

The Jacobian matrix of the external nodal forces is called the load stiffness matrix and given by

\[
K^{\text{ext}}_{ab} = \frac{\partial f^{\text{ext}}}{\partial d_b}, \quad K^{\text{ext}}_{ij} = \frac{\partial f^{\text{ext}}}{\partial u_{ij}} \quad K^{\text{ext}} = \frac{\partial f^{\text{ext}}}{\partial d} \tag{6.3.24}
\]

The development of these matrices is called linearization and is treated in Sections 6.4 and 6.5. Using (6.3.23-24), the Jacobian matrix (6.3.22) can be written as

\[
A = \frac{s_D}{\beta \Delta t^2} M + K^{\text{int}} - K^{\text{ext}} \tag{6.3.25}
\]

This Jacobian matrix applies to both dynamic and equilibrium problems with the dynamic switch \( s_D \) set by (6.3.2).

The Jacobians in (6.3.23-24) can be used to relate differentials of the nodal forces to differentials of the nodal displacements by

\[
df^{\text{int}} = K^{\text{int}} dd \quad df^{\text{ext}} = K^{\text{ext}} dd \quad dr = A dd \tag{6.3.26}
\]

The matrices which relate finite increments of nodal displacements to increments of nodal forces differ from the above. We will use a...
\[
\Delta f^{\text{int}} = K^{\text{int}}_\Delta \Delta d \\
\Delta f^{\text{ext}} = K^{\text{ext}}_\Delta \Delta d \\
\Delta r = A_\Delta \Delta d \\
\text{(6.3.27)}
\]

The matrix $K^{\text{int}}_\Delta$ is called a secant stiffness and $A_\Delta$ the secant Jacobian. The secant stiffness and secant Jacobian depend on the magnitude and direction of $\Delta d$. This can easily be seen in one dimension as illustrated in Fig. 6.3, which shows secants for various stepsizes in two directions. The tangent and secant Jacobians are identical only in the limit as $\Delta d \to 0$; for finite increments, the secant stiffness in (6.3.27) differs from the tangent stiffness in (6.3.23).

![Figure 6.3. Secant Jacobians for various step sizes and directions; step in secant is indicated by superscript.](image)

**Conservative Problems (Stationary Points).** It is useful at this point to examine the discrete problem corresponding to the stationary principle described in Section 4.9.3. This stationary principle only applies to conservative equilibrium problems, but it is nevertheless provides insight into the character of nonlinear problems. An equilibrium solution is a stationary point of the potential, so by enforcing the conditions that the derivative of the potential vanish and using (4.9.29-30) and the definition of the residual (6.3.3) we have

\[
0 = r = -\frac{\partial W}{\partial d} = \frac{\partial W^{\text{int}}}{\partial d} - \frac{\partial W^{\text{ext}}}{\partial d} = f^{\text{int}} - f^{\text{ext}} \\
\text{(6.3.28)}
\]

A solution is a stable equilibrium solution if it corresponds to a minimum of the potential energy. Thus stable equilibrium solutions can be found by minimizing the potential $W$. The situation is depicted in Fig. 6.4, which shows the local behavior of a potential of two generalized displacements and the contours for this potential. The residual is the negative of the gradient of the potential (note the sign in the above.)

The linear model for (6.3.28) is (see 6.3.17-18)
\[-r_v = \frac{\partial r(d_v)}{\partial d} \Delta d = -\frac{\partial^2 W(d_v)}{\partial d\partial d} \Delta d = A\Delta d \quad (6.3.29a)\]

where

\[A_{ab} = \frac{\partial^2 W}{\partial d_a \partial d_b} \quad \text{or} \quad A = \frac{\partial^2 W}{\partial d\partial d} \quad (6.3.29b)\]

The matrix A when defined by the second derivatives of a potential is called a Hessian matrix. For a conservative system, it is identical to the Jacobian, so

\[A = K^\text{int} - K^\text{ext} \quad (6.3.30)\]

The linearized equations for a conservative system are

\[\left(K^\text{int} - K^\text{ext}\right)\Delta d = -r\]

The above are identical to (6.3.19) except that the mass matrix is omitted, since dynamic effects cannot be included in a conservative problem. Finding a satisfable solution of a conservative problem can also be posed as a minimization problem. Then many techniques not directly applicable to more general nonlinear problems, such as the method of steepest descent, can be used. Viewing the solution of the residual equations as a minimization problem is helpful in many cases.

6.3.5. Implementation of Newton Method. Flowcharts for implicit integration and equilibrium solutions are given in Boxes 6.3 and 6.4. Both the dynamic problem and the equilibrium problem are solved by time-stepping: the external loads and other conditions are described as functions of time, which is incremented over the range of interest. In equilibrium problems, the time is often replaced by a monotonically increasing parameter. Solutions of equilibrium processes obtained in this manner are called incremental solutions.

The flowchart shows a procedure often called a full Newton algorithm, where the Jacobian matrix is evaluated and inverted in every iteration of the procedure. Many programs use a modified Newton algorithm, in which the Jacobian is only triangulated at the beginning of the iterations or intermittently during the iteration. For, example, in a modified Newton procedure the Jacobian may be triangulated only when the iterative procedure does not seem to be converging well. These modified schemes are faster but less robust.

The flowcharts begin with the imposition of the initial conditions. The initial conditions can be handled exactly as in explicit methods. The initial displacements are considered to be zero. The initial accelerations are computed as shown in steps 2 and 3.

The displacements \(d^{n+1}\) for each time step are obtained by the iterative Newton procedure. To begin the iterative procedure, a starting value of \(d\) is needed; usually the solution from the preceding step is used. The residual is then calculated for this starting value. In an equilibrium solution, the residual depends
only on the internal and external nodal forces, and is obtained in the module \textit{getf}. This module, \textit{getf}, is the same as in the explicit procedure, Box 6.1, except that the calculation of the stable time step is omitted, so it is not repeated. In transient implicit solutions, the residuals also depend on the accelerations.

**Box 6.3**

**Flowchart for Implicit Time Integration**

1. Initial conditions & initialization of parameters:
   
   set $\mathbf{v}^0, \mathbf{\sigma}^0, \mathbf{d}^0 = \mathbf{0}$, $n = 0, t = 0$; compute $\mathbf{M}$

2. get $\mathbf{f}^0 = f(\mathbf{d}^0, 0)$

3. compute initial accelerations $\mathbf{a}^n = \mathbf{M}^{-1}\mathbf{f}^n$

4. estimate next solution $\mathbf{d} = \mathbf{d}^n$

5. Newton iterations for time step $n+1$
   
   a. get $f$ computes $f(\mathbf{d}, t^{n+1})$

   b. $\mathbf{a}^{n+1} = \frac{1}{\beta \Delta t} (\mathbf{d} - \tilde{\mathbf{d}}^{n+1})$, $\mathbf{v}^{n+1} = \tilde{\mathbf{v}}^{n+1} + \gamma \varepsilon \mathbf{a}^{n+1}$, see Eqs. (6.3.4 - 6.3.7)

   c. $\mathbf{r} = \mathbf{Ma}^{n+1} - \mathbf{f}$

   d. compute Jacobian $\mathbf{A}(\mathbf{d})$

   e. modify $\mathbf{A}(\mathbf{d})$ for essential boundary conditions

   f. solve linear equations $\Delta \mathbf{d} = \mathbf{A}^{-1}\mathbf{r}$

   g. $\mathbf{d} \leftarrow \mathbf{d} + \Delta \mathbf{d}$

   h. check error criterion; if not met, go to step 5a

6. update displacements, counter and time: $\mathbf{d}^{n+1} = \mathbf{d}^n, n \leftarrow n + 1, t \leftarrow t + \Delta t$

7. check energy balance

8. output, if simulation not complete, go to 3

**Box 6.4**

**Flowchart for Equilibrium Solution**

1. Initial conditions and initialization: set $\mathbf{d}^0 = \mathbf{0}$; $\mathbf{\sigma}^0; n = 0; t = 0$

2. Newton iterations for load increment $n+1$
   
   a. get $f$ computes $f(\mathbf{d}, t^{n+1})$; $\mathbf{r} = f(\mathbf{d}, t^{n+1})$

   b. compute $\mathbf{A}(\mathbf{d})$

   c. modify $\mathbf{A}(\mathbf{d})$ for essential boundary conditions

   d. solve linear equations $\Delta \mathbf{d} = -\mathbf{A}^{-1}\mathbf{r}$

   e. $\mathbf{d} \leftarrow \mathbf{d} + \Delta \mathbf{d}$

   f. check error criterion; if not met, go to 2a

3. update displacements, step count and time: $\mathbf{d}^{n+1} = \mathbf{d}^n, n \leftarrow n + 1, t \leftarrow t + \Delta t$

4. output, if simulation not complete, go to 2
The Jacobian matrix in this algorithm is then calculated based on the latest state of the body. In some algorithms, the Jacobian for the last converged solution is used for all the iterations or the Jacobian is only recomputed intermittently during the iterations; these are known as modified Newton methods.

Simple essential boundary conditions, such as homogeneous displacement conditions, can be enforced by modifying the Jacobian matrix. The equation corresponding to the vanishing displacement component is either omitted or replaced by a dummy equation which states that the component vanishes. This can be done by zeroing the corresponding row and column and putting a positive constant on the diagonal of the Jacobian. For more complex algebraic constraints, Lagrange multipliers methods or penalty methods are used; these are described in Section 6.7.

### 6.3.8 Convergence Criteria

The termination of the iterative procedure in implicit and equilibrium solutions by the Newton method is determined by convergence criteria. These criteria pertain to the convergence of the discrete solution to the equations \( r(d^n, r^n) = 0 \), not the convergence of the discrete solution to the solution of the partial differential equations. Three types of convergence criteria are used to control the iterations:

1. criteria based on the magnitude of the residual \( r \);
2. criteria based on the magnitude of the displacement increments \( \Delta d \);
3. energy error criteria.

Usually an \( \ell_2 \) norm of the vectors is used for the first two criteria. The criteria then are:

**Residual error criterion:**

\[
\| r \|_{\ell_2} = \left( \sum_{a=1}^{n_{DOF}} r_a^2 \right)^{\frac{1}{2}} \leq \varepsilon \max \left( \| r^{ext} \|_{\ell_2}, \| r^{int} \|_{\ell_2}, \| Ma \|_{\ell_2} \right) 
\]  

(6.3.28)

**Displacement increment error criterion:**

\[
\| \Delta d \|_{\ell_2} = \left( \sum_{a=1}^{n_{DOF}} (\Delta d_a)^2 \right)^{\frac{1}{2}} \leq \varepsilon \| d \|_{\ell_2} 
\]  

(6.3.29)

The \( \ell_2 \) norm, which has been indicated in the above, is the probably most suitable when the mean error over all degrees of freedom is to be controlled, but a maximum norm can also be used. A maximum norm would limit the maximum error at any node. The terms on the right-hand side of Eqs. (6.3.28) and (6.3.29) are scaling factors. Without these, the criterion would depend on the parameters of the problem. The error tolerance \( \varepsilon \) determines the precision with which the displacements are calculated before terminating the iterative procedure; when \( \varepsilon = 10^{-3} \), the mean accuracy of the nodal displacements is in the third significant digit when the \( \ell_2 \) norm is used. The convergence tolerance determines the speed and accuracy of a calculation. If the criterion is too coarse, the solution may be
quite inaccurate. On the other hand, a criterion which is too tight results in unnecessary computations.

The energy convergence criterion measures the energy flow to the system resulting from the residual, which is like an error in energy. It is given by

$$\max \left| \Delta d^T r \right| = \max \left| \Delta d_a r_a \right| \leq \varepsilon \max \left(W_{\text{ext}}, W_{\text{int}}, W_{\text{kin}} \right)$$ (6.3.30)

where the computation of the energies used for scaling the criterion is described in Section 6.7. The left hand side in the above represents an error in the energy, since a nonzero residual is an error in the forces on the system.

6.3.9. Convergence and Robustness of Newton Iteration. The rate of the convergence of the iterations in the Newton method is quadratic when the Jacobian matrix $A$ satisfies certain conditions. These conditions may roughly be described as follows:

1. the Jacobian $A$ should be a sufficiently smooth function of $d$;

2. the Jacobian $A$ should be regular (invertable) and well-conditioned in the entire domain in the displacement space that the iterative procedure traverses.

Quadratic convergence means that the $\ell_2$ norm of the difference between the solution and the iterate $d_{u_0}$ decreases quadratically in each iteration:

$$\|d_{u+1} - d_u\| \leq c \|d_{u} - d\|^2$$ (6.3.31)

where $c$ is a constant that depends on the nonlinearity of the problem and $d$ is the solution to the nonlinear algebraic equations. Thus the convergence of the Newton algorithm is quite rapid when $A$ meets the above conditions. The above gives the requirements for convergence only in broad terms and convergence has been proven for various conditions on $A$. One set of conditions for quadratic convergence are: the residual must be continuously differentiable and the inverse of the Jacobian matrix must exist and be uniformly bounded in the neighborhood of the solution, Dennis and Schnabel (1983, p 90).

These conditions are usually not satisfied by nonlinear finite element problems. For example, in an elastic-plastic material, the residual is not a continuously differentiable function of the nodal displacements when an element quadrature point changes from elastic to plastic or vice versa; therefore, the Jacobian is discontinuous. In the solution of contact-impact problems with Lagrange multiplier methods, the residual often lacks smoothness, as illustrated by Chapter 10. Thus the conditions for quadratic convergence of the Newton method are often not satisfied in engineering problems. Yet, Newton’s method is remarkably effective in engineering problems, although the rate of convergence often deteriorates. At this time, more robust methods are not available. In many problems, the conditions for quadratic convergence are satisfied; for example, the above conditions are satisfied in the response of a model with a Mooney-Rivlin material when the load is small enough so that the equilibrium solutions are stable.
Newton’s method fails particularly often when applied to equilibrium problems. Since (6.3.3) are nonlinear algebraic equations, they can have multiple solutions and solutions which are unstable. When the equilibrium path is unstable, the Jacobian matrix is no longer regular at all points and the proof of quadratic convergence does not apply. The convergence of the Newton method often fails in the vicinity of unstable states. These types of problems are considered in the next Section.

In summary, Newton’s method sometimes lacks robustness when applied to engineering problems. The robustness decreases as we increase the time step and appears more often in equilibrium solutions, since in the latter we lose the effect of the mass matrix. The mass matrix improves the conditioning of the Jacobian matrix because it is always positive definite, see Exercise X. As the time step increases, the beneficial effects of the mass matrix decrease since the coefficient of the mass matrix is inversely proportional to the square of the time step, as can be seen from Eq. (6.3.9). For many problems, a straightforward application of the Newton method will sometimes fail completely, and enhancements of the Newton method such as the arc length method, line search, and augmented Lagrangian, which are described in Section ?, are needed to solve the nonlinear algebraic equations.

6.3.10. Line Search. An effective way to increase the robustness of Newton methods when convergence is slow is to use the line search technique. The rationale behind line search is that the direction $\Delta d$ found by the Newton method is often a good direction, but the step size is not optimal. It is cheaper to find the best point along this direction by several computations of the residual than to get a new direction by using a new Jacobian. Therefore, before proceeding to the next direction, a measure of the residual is minimized along the line $d_{old} + \xi \Delta d$ where $d_{old}$ is the last iterate and $\xi > 0$ is a parameter. In other words, we find the parameter $\xi$ so that $d_{old} + \xi \Delta d$ minimizes some measure of the residual. We can use as a measure of the measure of the residual its $\ell_2$ norm, as defined in Eq. (6.3.28), the maximum norm, i.e. the maximum absolute value of any component of the residual, or some other measure. Line search then involves the calculation of two or more residuals along the line, an interpolation of a measure of the residual, and finding the point along the line where the measure of the residual is a minimum.

A measure for the residual which is frequently used in line search is based on the existence of a potential for the problem, i.e. on the solution by the stationary energy principle, Sections 4.9.3 and 6.3.6. For a conservative problem, the minimizer of the potential $W(d)$, along the direction $\Delta d$ is the point where the gradient of the function is orthogonal to the $\Delta d$. The residual is given in terms of a potential by

$$\frac{\partial W}{\partial d} = \frac{\partial W^{\text{int}}}{\partial d} - \frac{\partial W^{\text{ext}}}{\partial d} = f^{\text{int}} - f^{\text{ext}} = r$$

(6.3.32)

where the above follows from (4.9.34) and (6.3.3). When the residual is orthogonal to the incremental displacement
the potential must be minimum (or be stationary) at that point. This is illustrated in Fig. 6.??, which shows the contours of the potential energy for a two degree-of-freedom system and the residual of the nodal forces for several points along the line $d_{old} + \xi \Delta d$. As can be seen, the potential is minimum when the residual, i.e. the gradient of the potential, is normal to the line. The line search can then be conducted by minimizing $|\Delta d^T r|$.

This criterion can also be used for systems that are not conservative since $\Delta d^T r$ can be evaluated in the absence of a potential. Note that this measure of the residual is equivalent to the criterion for error in energy, (6.3.30).

Equation (6.3.33a) can also be derived directly by using the chain rule to expand the potential energy in the parameter $\xi$. This gives

$$\frac{dW(\xi)}{d\xi} = \frac{\partial W}{\partial d} \cdot d_d \frac{d}{d\xi} = 0 \Rightarrow r^T \Delta d = 0$$ (6.3.34)

where we have set the derivative of the potential energy with respect to the parameter $\xi$ equal to zero, since we are looking for the minimum of the potential along the line $\Delta d$ parametrized by $\xi$. The second equation follows from (6.3.32) and

$$\frac{d}{d\xi} = \frac{d(d_{old} + \xi \Delta d)}{d\xi} = \Delta d$$ (6.3.35)

Once a measure of the residual has been chosen, the line search can be made with any of the methods for minimizing a function of a single parameter. The method of bisection or searches based on interpolation or combinations thereof can be used. Once the residual has been evaluated at two points, the residual measure can be interpolated by a quadratic function, since its value at $\xi = 0$ is known to vanish. This quadratic interpolate can then be used to estimate the position of the minimum. The iteration along the line is terminated when the measure has been minimized to a suitable precision. Note that when the orthogonality condition (6.3.29) is used, it should be normalized like the error energy criterion in (6.3.26).

6.3.11. Stability of Implicit methods. The advantage of an implicit method over an explicit method is that for linear transient problems, suitable implicit integrators are unconditionally stable. The unconditional stability of implicit integrators has not been proven for all nonlinear systems, although results which deal with specific situations indicate that unconditional stability holds at least for certain nonlinear systems. In any case, experience indicates that the time
steps which can be used with implicit integrators are much larger than those for explicit integration in many problems.

The major restrictions on the size of time steps in implicit methods arise from accuracy requirements and the decreasing robustness of the Newton procedure as the time step increases. The latter is particularly pronounced in problems with very rough response, such as contact-impact. With a large time step, the starting iterate may be far from the solution, so the possibility of failure of the Newton method to converge increases. Therefore small time steps are often used to improve the robustness of the Newton algorithm.

In return for their enhanced stability, implicit methods exact a significant price: implicit methods require the solution of nonlinear algebraic equations in each time step. The construction of the linearized algebraic equations for the Newton procedure is often quite involved. Furthermore, the storage of these equations requires significant amounts of memory. The memory requirements can be reduced substantially by iterative linear equation solvers (an iterative method within an iterative Newton method). In recent research, iterative solvers have improved dramatically, so implicit solutions are feasible in many problems where they were prohibitive before, see Section ?. The robustness and speed of Newton methods has increased markedly over the past two decades, and we are certain that further improvements are imminent. Nevertheless, high cost and lack of robustness still plague many implicit and equilibrium solution procedures.

6.4 LINEARIZATION

Linearization is carried out in two ways:
1. Linearization is carried out before the stress-update algorithm (integration algorithm for the constitutive equation) is introduced; this gives rise to the so-called continuum tangent moduli which will be discussed below.
2. Linearization is carried out after the stress-update algorithm is introduced; this gives rise to the so-called algorithmic moduli or secant moduli.

The choice of which approach to use rests on practical considerations related to ease of implementation and on the smoothness of the problem. The first approach, based on the continuum tangent modulus, is straightforward to implement. However, it can run into convergence difficulties, when the derivatives of the constitutive equation are discontinuous, as at the yield point of elastic-plastic materials.

The second approach, based on the algorithmic moduli, exhibits better convergence because, through linearization of the stress-update algorithm, it accounts for the finite increment of strain. One drawback of the method is that it is not always possible to derive explicit forms for the algorithmic moduli for complex constitutive relations. Numerical differentiation schemes are sometimes used to obtain the algorithmic moduli, and they introduce additional inaccuracies.
We first consider linearization of the discrete equations based on the tangent moduli, which relate a stress rate to a strain rate. The resulting material tangent stiffness matrix is called the *tangent stiffness matrix*.

Next a linearization based on directional derivatives is presented. The resulting expressions are equivalent to those obtained by using the procedure based on the material time derivative but they involve directional derivatives. The linearization procedure based on the directional derivative is then used to develop the linearized equations for the second approach discussed above, i.e., linearization of the weak form after introduction of the stress-update algorithm. This tangent stiffness is called the algorithmic stiffness (sometimes referred to as the consistent tangent modulus because of the consistent linearization of the weak form and the stress-update algorithm). Examples of the algorithmic stiffness for the 2-node bar element and the 3-node triangle are also given.

### 6.4.1 Linearization of the Discrete Equations

In the following, we derive expressions for the continuum tangent stiffness matrix $\mathbf{K}^{int}$. As will be seen, part of the expression can be derived independently of the material response. These expressions are completed upon introduction of the constitutive relation. The continuum rate form of the constitutive relation will be used, i.e., linearization is carried out prior to introduction of the stress-update algorithm. Specific examples for the tangent matrices for hyperelastic materials and elastic-plastic materials are presented in Section 6.4.2.

For notational convenience, we will develop the tangential stiffness matrix by relating rates of the internal nodal forces $f^{int}$ to the nodal velocities $\dot{d}$. Thus the stiffness matrices $\mathbf{K}^{int}$ can be derived by taking the material time-derivative of the nodal internal forces. The procedure is identical to relating an infinitesimal increment of nodal forces $d\mathbf{f}^{int}$ to an infinitesimal increment of nodal displacements $d\mathbf{d}$, and we will occasionally recast the equations in that form; the superposed dot notation is chosen for convenience. The results are exact for any continuously differentiable residual; for rougher residuals, directional derivatives are needed and are described later.

By (4.9.10-11), the internal nodal forces in the total Lagrangian form are given by,

$$ f^{int} = \int_{\Omega_0} \mathbf{B}_0^T \mathbf{P} d\Omega_0, \quad f^{int}_{il} = \int_{\Omega_0} \frac{\partial N_i}{\partial X_j} P_{ji} d\Omega_0 \quad (6.4.1) $$

where $\mathbf{P}$ is the nominal stress tensor with components $P_{ji}, N_i$ are the nodal shape functions and $\mathbf{B}_{0lj} = \frac{\partial N_i}{\partial X_j}$. We have chosen the total Lagrangian form because this leads to the simplest derivation. In the total Lagrangian form, (6.4.1), the only variable which is a function of time is the nominal stress, i.e. it is the only variable which varies with deformation. In the updated Lagrangian form, (4.5.5) the domain of the element (or body), the spatial derivatives $\frac{\partial N_i}{\partial X_j}$ and the Cauchy stress depend on the deformation, and hence on time.

Taking the material time-derivative of (6.4.1) gives
\[
\dot{f}^{\text{int}} = \int_{\Omega_0} \mathbf{B}_0^T \dot{\mathbf{P}} d\Omega_0, \quad \dot{f}_{li}^{\text{int}} = \int_{\Omega_0} \frac{\partial N_l}{\partial X_j} \dot{P}_{lj} d\Omega_0
\]  
(6.4.2)

since \( \mathbf{B}_0 \) and \( d\Omega_0 \) are independent of the deformation or time. To obtain the stiffness matrix \( \mathbf{K}^{\text{int}} \) it is now necessary to express the stress rate \( \dot{\mathbf{P}} \) in terms of nodal velocities. However, constitutive equations are not expressed in terms of \( \dot{\mathbf{P}} \) because this stress rate is not objective. So we work in terms of the material time derivative of the PK2 stress, which we have seen is objective.

The material time derivative of the PK2 stress is then related to the material time derivative of the nominal stress by Box 3.2, which gives \( \dot{\mathbf{P}} = \dot{\mathbf{S}} \cdot \mathbf{F}^T \), so

\[
\dot{\mathbf{P}} = \dot{\mathbf{S}} \cdot \mathbf{F}^T + \mathbf{S} \cdot \dot{\mathbf{F}}^T \quad \text{or} \quad \dot{P}_{ij} = \dot{S}_{ir} F_{rj}^T + S_{ir} \dot{F}_{rj}^T
\]  
(6.4.3)

Substituting (6.4.3) into (6.4.2) yields

\[
\dot{f}_{li}^{\text{int}} = \int_{\Omega_0} \frac{\partial N_l}{\partial X_j} (\dot{S}_{ir} F_{rj}^T + S_{ir} \dot{F}_{rj}) d\Omega_0 \quad \text{or} \quad \dot{f}_{il} = \int_{\Omega_0} \frac{\partial N_l}{\partial X_j} (dS_{ir} F_{rj} + S_{ir} dF_{rj}) d\Omega_0
\]  
(6.4.4)

The above shows that the rate (or increment) of the internal nodal forces consists of two distinct parts:

1. The first term involves the rate of stress (\( \dot{\mathbf{S}} \)) and thus depends on the material response and leads to what is called the material tangent stiffness matrix which we denote by \( \mathbf{K}^{\text{mat}} \).
2. The second term involves the current state of stress, \( \mathbf{S} \) and accounts for rotation of the stress with the motion. This term is called the geometric stiffness because it represents for geometric nonlinearities associated with rotation of the stress. It is also called the initial stress matrix to indicate the role of the existing state of stress. It is denoted by \( \mathbf{K}^{\text{geo}} \).

Therefore we write (6.4.4) as

\[
\dot{f}^{\text{int}} = \dot{f}^{\text{mat}} + \dot{f}^{\text{geo}} \quad \text{or} \quad \dot{f}_{il} = \dot{f}_{il}^{\text{mat}} + \dot{f}_{il}^{\text{geo}}
\]  
(6.4.5)

where

\[
\dot{f}_{il}^{\text{mat}} = \int_{\Omega_0} \frac{\partial N_l}{\partial X_j} \dot{S}_{ir} F_{rj} d\Omega_0, \quad \dot{f}_{il}^{\text{geo}} = \int_{\Omega_0} \frac{\partial N_l}{\partial X_j} S_{ir} \dot{F}_{rj} d\Omega_0
\]  
(6.4.6a,b)

To simplify the remaining development, we put the above expression into Voigt form. Voigt form is convenient in developing the material stiffness matrices because the tensor of material coefficients, \( C_{ijkl} \), which relates the stress
rate to the strain rate is a fourth order tensor; this tensor cannot be handled by readily standard matrix operations.

We first consider the material effects on the nodal force rates. Recalling (4.9.20)

\[ B_{ij}^0 = \text{sym}_{(j,r)} \left( \frac{\partial N_I}{\partial X_j} F_{ir} \right) \] (6.4.7)

we can rewrite the material rate of the internal nodal forces, (6.4.5), in Voigt notation as

\[ \dot{f}_{\text{mat}}^\text{int} = \int_{\Omega_0} B_0^T [\dot{S}] d\Omega_0 \] (6.4.8)

where \([\dot{S}]\) is the rate of the PK2 stress in a column matrix arranged according to the Voigt kinetic rule, also see Appendix A. It should be stressed that (6.4.8) is identical to Eq. (6.4.6a). We now consider the constitutive equation in the following rate form

\[ \dot{S}_{ij} = C_{ijkl}^S E_{kl} \] (6.4.9)

Recall (4.9.27), which gives the following relation in Voigt notation

\[ \dot{\mathbf{E}} = B_0 d \] (6.4.10)

Substituting (6.4.9) and (6.4.10) into (6.4.8) gives

\[ \dot{f}_{\text{mat}}^\text{int} = \int_{\Omega_0} B_0^T [C^S] B_0 d\Omega_0 \dot{d} \] or \[ \dot{f}_{\text{mat}}^\text{int} = \int_{\Omega_0} B_0^T [C^S] B_0 d\Omega_0 d\dot{d} \] (6.4.11)

So the material tangent stiffness matrix is given by

\[ K_{\text{mat}} = \int_{\Omega_0} B_0^T [C^S] B_0 d\Omega_0 \] or \[ K_{IJ}^\text{mat} = \int_{\Omega_0} B_0^T [C^S] B_{0J} d\Omega_0 \] (6.4.12)

The material tangent stiffness relates the increment (or rate) in internal nodal forces to the increment (or rate) of displacement due to material response, which is reflected in the material response matrix \( C^S \).

The geometric effect on the nodal forces is obtained as follows. From the definition \( B_{ij}^0 = \frac{\partial N_I}{\partial X_j} \) and (6.4.4), we can write
where in the second step we have used (4.9.7), \( \hat{\mathbf{F}}_{\text{tr}} = \mathbf{B}_{\text{tr}}^0 \hat{\mathbf{u}}_{\text{tr}} \), and in the third step we have added a dummy unit matrix so that the component indices in \( \mathbf{j}_{\text{geo}}^\text{geo} \) and \( \mathbf{u}_{kJ} \) are not the same. Writing the resulting expression for the geometric stiffness in matrix form gives

\[
\mathbf{f}_l = \mathbf{K}_{lj}^\text{geo} \mathbf{u}_J \quad \text{where} \quad \mathbf{K}_{lj}^\text{geo} = \int \mathbf{B}_{0j}^T \mathbf{B}_{0j} d\Omega_0 \mathbf{I} \tag{6.4.15}
\]

Note that the PK2 stress in the above is a square matrix. Each submatrix of the geometric stiffness matrix is a unit matrix. Therefore the geometric stiffness matrix is invariant with rotation, i.e.

\[
\mathbf{K}_{lj}^\text{geo} = \mathbf{K}_{lj}^\text{geo} \tag{6.4.16}
\]

where \( \mathbf{K}_{lj}^\text{geo} \) is expressed in any alternate set of Cartesian coordinates.

To summarize

\[
d\mathbf{f}_l^\text{int} = \mathbf{K}_l^\text{int} d\mathbf{d} \quad \text{or} \quad \mathbf{i}_l^\text{int} = \mathbf{K}_l^\text{int} d\mathbf{d} \quad \text{where} \quad \mathbf{K}_l^\text{int} = \mathbf{K}_l^\text{mat} + \mathbf{K}_l^\text{geo} \tag{6.4.17}
\]

where the material tangent stiffness and the geometric stiffness are given by (6.4.12) and 6.4.15), respectively. The material tangent stiffness reflects the effect on the nodal internal forces of the deformation of the material. The geometric stiffness reflects the effects of the rotation and deformation on the current state of stress.

The above forms are easily converted to updated Lagrangian forms by letting the current configuration be a reference configuration, as in Section 4.9.2, (4.9.28-29). From (4.9.29), we recall that taking the current configuration as the reference configuration gives

\[
\mathbf{B}_0 = \mathbf{B} \quad \mathbf{B}_0 = \mathbf{B} \quad \mathbf{S} = \mathbf{\sigma} \quad d\Omega_0 = d\Omega \tag{6.4.18}
\]

i.e. the \( \mathbf{B}_0 \) becomes \( \mathbf{B} \) when the reference configuration corresponds to the current configuration. Also, we note that when a current configuration is the reference configuration, then

\[
\mathbf{F} = \mathbf{I} \tag{6.4.19}
\]

\[
\mathbf{C}^{SE} = \mathbf{C}^{\sigma_T} \tag{6.4.20}
\]
where (6.4.20) is shown in Section 5.?? Thus, (6.4.13) and (6.4.16) become

\[
\mathbf{K}_{IJ}^{\text{mat}} = \int_{\Omega} \mathbf{B}_{I}^{T} \mathbf{C}^{\sigma_{T}} \mathbf{B}_{J} d\Omega \quad \mathbf{K}_{IJ}^{\text{geo}} = \int_{\Omega} \mathbf{B}_{I}^{T} \mathbf{S} \mathbf{B}_{J} d\Omega
\]

(6.4.21)

The integrand in the geometric stiffness is a scalar for given values of \( I \) and \( J \), so (6.4.21) can be written as

\[
\mathbf{K}_{IJ}^{\text{geo}} = \mathbf{I} \int_{\Omega} \mathbf{B}_{I}^{T} \mathbf{S} \mathbf{B}_{J} d\Omega
\]

(6.4.22)

These forms are generally easier to use than the total Lagrangian forms, since \( \mathbf{B} \) is more easily constructed than \( \mathbf{B}_{0} \) and many material laws are developed in terms of Cauchy stress. Note that either the material or geometric stiffness can be used in total Lagrangian form with the other in updated Lagrangian form. The numerical values of the matrices in total and updated Lagrangian form are identical, and the choice which to use is a matter of convenience.

**External Load Stiffness.** An important class of loads are follower loads, which change with the configuration of the body. Examples of follower forces are shown in Figure ?? Pressure loading is a common example of a follower load. Since a pressure loading is always normal to the surface, as the surface moves, the nodal external forces change even if the pressure is constant. These effects are accounted for in the Jacobian matrix \( \mathbf{K}^{\text{ext}} \), which is also called the load stiffness.

The load stiffness \( \mathbf{K}^{\text{ext}} \) is obtained by relating the time derivative (or increment) of the external nodal forces to the time derivative (or increment) of nodal displacements. We first consider loading by pressure, \( p(\mathbf{x}, t) \). The external nodal forces on a surface of element \( e \) are given by letting \( t = -p \mathbf{n} \) in (4.9.13):

\[
f_{I}^{\text{ext}} = - \int_{\Gamma} N_{I} p \mathbf{n} d\Gamma
\]

(6.4.61)

Let the surface \( \Gamma \) be described in terms of two variables \( \xi \) and \( \eta \). For a quadrilateral surface element, these independent variables are the parent element coordinates on the biunit square. As in (E4.3.1b), since \( \mathbf{n} d\Gamma = x_{,\xi} \times x_{,\eta} d\xi d\eta \) becomes

\[
f_{I}^{\text{ext}} = - \int_{-1}^{1} \int_{-1}^{1} p N_{I} x_{,\xi} \times x_{,\eta} d\xi d\eta
\]

(6.4.62)

Taking the time derivative of the above gives
\[ f^\text{ext}_I = - \int_{-1}^{1} \int_{-1}^{1} N_I \left( \hat{p} \mathbf{x}_{\xi} \times \mathbf{x}_{\eta} + p \mathbf{v}_{\xi} \times \mathbf{x}_{\eta} + p \mathbf{x}_{\xi} \times \mathbf{v}_{\eta} \right) d\xi d\eta \quad (6.4.63) \]

The first term is the rate of change of the external forces due to the rate of change of the pressure. In many problems the rate of change of pressure is prescribed as part of the problem definition. In other problems, such as in fluid-structure interaction problems, the pressure may arise from changes of the geometry; these effects must then be linearized and added to the load stiffness. The second two terms represent the changes in the external nodal forces due to the change in the direction of the surface and the area of the surface. These are the terms which are reflected in the external load stiffness, so

\[ K^\text{ext}_{ik} \mathbf{v}_K = - \int_{-1}^{1} \int_{-1}^{1} pN_I \left( \mathbf{v}_{\xi} \times \mathbf{x}_{\eta} + \mathbf{x}_{\xi} \times \mathbf{v}_{\eta} \right) d\xi d\eta \quad (6.4.64) \]

At this point, it is convenient to switch partially to indicial notation. Taking the dot product of the above with the unit vector \( \mathbf{e}_i \) gives

\[ \mathbf{e}_i \cdot K^\text{ext}_{ik} \mathbf{v}_K \equiv K^\text{ext}_{iklj} \mathbf{v}_{kj} \]

\[ = - \int_{-1}^{1} \int_{-1}^{1} pN_I \left[ N_{j, \xi} \mathbf{e}_i \cdot (\mathbf{e}_k \times \mathbf{x}_{\eta}) + N_{j, \eta} \mathbf{e}_i \cdot (\mathbf{x}_{\xi} \times \mathbf{e}_k) \right] d\xi d\eta \mathbf{v}_{kj} \quad (6.4.65) \]

where we have expanded the velocity field in terms of the shape functions by \( \mathbf{v}_{\xi} = \mathbf{v}_K N_{K, \xi} \). We now define

\[ H_{ik}^\eta = e_{ikt} x_t, \eta \quad H_{ik}^\xi = e_{ikt} x_t, \xi \quad (6.4.66) \]

Using these definitions and (6.4.65), we obtain

\[ K^\text{ext}_{ijkl} = - \int_{-1}^{1} \int_{-1}^{1} pN_I \left( N_{j, \xi} H_{ik}^\eta - N_{j, \eta} H_{ik}^\xi \right) d\xi d\eta \quad (6.4.67) \]

or

\[ K^\text{ext}_{ij} = - \int_{-1}^{1} \int_{-1}^{1} pN_I \left( N_{j, \xi} H_{ik}^\eta - N_{j, \eta} H_{ik}^\xi \right) d\xi d\eta \]

If we write out the matrices \( \mathbf{H}^\xi \) and \( \mathbf{H}^\eta \) we have
which is the load stiffness of any surface which is generated from a biunit square in the parent element loaded by a pressure $p$. The load stiffness matrix for a surface with a triangular parent element can be similarly expressed in terms of the area coordinates, although the limits of integration need to be changed. This load stiffness reflects the effect of the change in geometry on the external nodal forces: both alterations in the direction of the loaded surfaces and size of the surface will cause change the nodal forces. It is immediately apparent from (6.4.68) that the submatrices of the load stiffness matrix are not symmetric. However, it can be shown that for a closed structure in a constant pressure field, the assembled external load stiffness is symmetric.


We consider the three-node triangle in two dimensions as in Example 4.1. The element in the $x$-$y$ plane in a state of plane strain deformation. The only velocity components are $v_x$ and $v_y$, and derivatives with respect to $z$ vanish. The tangent stiffness matrix is derived and explicit forms for hyperelastic and rate-independent hypoelastic-plastic materials are given. The geometric tangent stiffness matrix, which is independent of material response, is then derived.

Material Tangent Stiffness Matrix. The material tangent stiffness matrix for a rate-independent material given by (6.4.21):

$$K^{tan} = \int_A B^T C^{\sigma,T} B dA \quad (E6.1.1)$$

where $A$ is the current area of the element and we have assumed a unit thickness (see Eq. (4.??)).

$$C^{\sigma,T}_{ab} = \begin{bmatrix} C_{1111} & C_{1122} & C_{1112} \\ C_{2211} & C_{2222} & C_{2212} \\ C_{1211} & C_{1222} & C_{1212} \end{bmatrix} \quad (E6.1.2)$$

The $B$ matrix is given by (E4.1.14):
\[
B = \frac{1}{2A} \begin{bmatrix}
y_{23} & 0 & y_{31} & 0 & y_{12} & 0 \\
0 & y_{32} & 0 & y_{13} & 0 & y_{21} \\
x_{32} & x_{23} & x_{13} & x_{31} & x_{21} & y_{12}
\end{bmatrix}
\] (E6.1.3)

The material tangent stiffness matrix, (6.4.81), is rewritten, using Eqs. (6.4.82-84) as

\[
K^{\text{tan}} = \int_A \left( \frac{1}{2A} \right)^2 \begin{bmatrix}
y_{23} & 0 & y_{32} \\
0 & y_{31} & 0 & y_{12} & 0 \\
x_{32} & x_{23} & x_{13} & x_{31} & x_{21} \\
y_{12} & 0 & x_{21} & y_{12} \\
0 & x_{32} & 0 & x_{13} & 0 & x_{21}
\end{bmatrix} \begin{bmatrix}
C^{\sigma_T} \\
C^{\sigma_T} \\
C^{\sigma_T} \\
C^{\sigma_T} \\
C^{\sigma_T}
\end{bmatrix} \begin{bmatrix}
C^{\sigma_T} \\
C^{\sigma_T} \\
C^{\sigma_T} \\
C^{\sigma_T} \\
C^{\sigma_T}
\end{bmatrix} dA
\] (E6.1.4)

Assuming the integrand to be constant, we obtain, by multiplying the integrand by the element area \(A\) (note that a unit thickness is assumed):

\[
K_{AB}^{\text{tan}} = \frac{1}{4A} \begin{bmatrix}
y_{23} & 0 & y_{32} \\
0 & y_{31} & 0 & y_{12} & 0 \\
x_{32} & x_{23} & x_{13} & x_{31} & x_{21} \\
y_{12} & 0 & x_{21} & y_{12} \\
0 & x_{32} & 0 & x_{13} & 0 & x_{21}
\end{bmatrix} \begin{bmatrix}
C^{\sigma_T} \\
C^{\sigma_T} \\
C^{\sigma_T} \\
C^{\sigma_T} \\
C^{\sigma_T}
\end{bmatrix} \begin{bmatrix}
C^{\sigma_T} \\
C^{\sigma_T} \\
C^{\sigma_T} \\
C^{\sigma_T} \\
C^{\sigma_T}
\end{bmatrix} \begin{bmatrix}
C^{\sigma_T} \\
C^{\sigma_T} \\
C^{\sigma_T} \\
C^{\sigma_T} \\
C^{\sigma_T}
\end{bmatrix} \begin{bmatrix}
C^{\sigma_T} \\
C^{\sigma_T} \\
C^{\sigma_T} \\
C^{\sigma_T} \\
C^{\sigma_T}
\end{bmatrix} dA
\] (E6.1.5)

Neo-Hookean Material. For a Neo-Hookean material (see Section 5.7.?),

\[
C^{\sigma_T}_{ijkl} = J^{-1} C^{\sigma}_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu(J)(\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk})
\] (E6.1.6)

where

\[
J = \det \mathbf{F}, \quad \mu(J) = \mu_0 - \lambda \log J,
\] (E6.1.7)

Using the fact that \(J = A/A_0\) the material response matrix is written in Voigt notation as
(E6.1.8)

Thus, for a Neo-Hookean material, the material tangent stiffness matrix has the same form as the stiffness matrix for small strain linear elasticity except for the dependence of the moduli on the deformation (through (E5.1.7)) and the geometry factor $A_0/A$.

**Rate-Independent Elastoplasticity.** For a rate-independent elastic-plastic model in terms of the Kirchoff stress, with associated plastic flow and a von-Mises yield condition, the tangent modulus is given by Eq. (5.??)

$$C^\tau_{ijkl} = C^\varepsilon_{ijkl} - \frac{1}{2} \left( \delta_{il} \tau_{jk} + \tau_{ik} \delta_{jl} + \delta_{ik} \tau_{jl} + \tau_{il} \delta_{jk} \right)$$  \hspace{1cm} (E6.1.9)

The elastoplastic tangent modulus is given by

$$C^\varepsilon_{ijkl} = C_{ijkl} - \frac{C_{ijmn} P_{mn} C_{ikrs} P_{rs}}{h + p_{nm} C_{mars} P_{rs}}$$  \hspace{1cm} (E6.1.10)

where $h$ is the plastic modulus, $p_{ij} = \frac{3\tau_{ij}'}{2\bar{\sigma}}$ is the plastic flow direction, $\tau_{ij}'$ is the deviatoric part of the Kirchoff stress and $\bar{\sigma}$ is the effective stress defined by (??). Assuming constant isotropic elastic moduli, (6.4.92) is written as

$$C^\varepsilon_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu \left( \delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk} \right) - \frac{4\mu^2}{h + 3\mu} p_{ij} p_{kl}$$  \hspace{1cm} (E6.1.11)

Using Voigt notation and letting $\gamma = \frac{2\mu}{(h + 3\mu)}$, $p_1 = p_{11}$, $p_2 = p_{22}$, $p_3 = p_{12}$ and $\tau_1 = \tau_{11}$, $\tau_2 = \tau_{22}$, $\tau_3 = \tau_{12}$, the tangent stiffness matrix is obtained as

$$C^\varepsilon_{ab} = \begin{bmatrix} \lambda + 2\mu & \lambda & 0 \\ 0 & \lambda & \lambda + 2\mu \\ 0 & 0 & \mu \end{bmatrix}$$

$$-2\mu \gamma \begin{bmatrix} p_1^2 & p_1 p_2 & p_1 p_3 \\ p_2 p_1 & p_2^2 & p_2 p_3 \\ p_3 p_1 & p_3 p_2 & p_3^2 \end{bmatrix} - \frac{1}{2} \begin{bmatrix} 4\tau_1 & 0 & 2\tau_3 \\ 0 & 4\tau_2 & 2\tau_3 \\ 2\tau_3 & 2\tau_3 & \tau_1 + \tau_2 \end{bmatrix}$$  \hspace{1cm} (E6.1.12)

**Geometric Stiffness Matrix.** The geometric stiffness matrix is given by (6.4.21):
\[ \mathbf{K}_{IJ}^{geo} = \mathbf{I}_{2 \times 2} \int_{A} \mathcal{B}_I^T \mathbf{\sigma} \mathcal{B}_J dA = \mathbf{I}_{2 \times 2} H_{IJ} \]  

(E6.1.13)

From (E4.1.18)

\[ \mathcal{B} = \frac{1}{2A} \begin{bmatrix} y_{23} & y_{31} & y_{12} \\ x_{32} & x_{13} & x_{21} \end{bmatrix} \]  

(E6.1.15)

Substituting (E6.1.15) into (E6.1.13) gives

\[ \mathbf{H} = \frac{1}{2A} \begin{bmatrix} y_{23} & x_{32} & \sigma_{xx} & \sigma_{xy} \\ y_{31} & x_{13} & \sigma_{xy} & \sigma_{yy} \\ y_{12} & x_{21} & \sigma_{yx} & \sigma_{yy} \end{bmatrix} \begin{bmatrix} 1 \\ \frac{1}{2A} \\ x_{32} & x_{13} & x_{21} \end{bmatrix} \]  

(E6.1.18)

Assuming the integrand to be constant, the geometric stiffness matrix is obtained by multiplying the integrand in (E6.1.13) by \( A \) to give

\[ \mathbf{K}_{IJ}^{geo} = \frac{1}{4A} H_{IJ} \mathbf{I}_{2 \times 2} \begin{bmatrix} H_{11} & 0 & H_{12} & 0 & H_{13} & 0 \\ 0 & H_{11} & 0 & H_{12} & 0 & H_{13} \\ H_{21} & 0 & H_{22} & 0 & H_{23} & 0 \\ 0 & H_{21} & 0 & H_{22} & 0 & H_{23} \\ H_{31} & 0 & H_{32} & 0 & H_{33} & 0 \\ 0 & H_{31} & 0 & H_{32} & 0 & H_{33} \end{bmatrix} \]  

(E6.1.19)

The geometric stiffness matrix is independent of material response and as can be seen from (E6.1.18-19) depends only on the current stress state and the geometry of the element.

**Example 6.2. Two-Node Rod Element.**

We now consider the two-node rod element under a state of uniaxial stress. The rod is assumed to lie along the \( \mathbf{x} \)-axis. The only non-zero Cauchy stress component is \( \dot{\sigma}_{11} \equiv \dot{\sigma}_x \). The tangent stiffness and the external load matrices are derived in the updated Lagrangian form, i.e. in the current configuration. We first reconsider the constitutive relation for the special case of uniaxial stress. The superscript hats are dropped in the following for convenience.

The Truesdell rate of the Cauchy stress is assumed to be given by Eq. (6.3.??)

\[ \dot{\sigma}_{ij} = C_{ijk \ell} \dot{\sigma}_{k \ell} \]  

(E6.2.1)
For the case of uniaxial stress, the only non-zero components of the rate of deformation tensor are $D_{11}$, $D_{22}$, and $D_{33}$.

The uniaxial stress rate is therefore given by

$$
\sigma_{11} = C_{1111}^{\sigma} D_{11} + C_{1122}^{\sigma} D_{22} + C_{1133}^{\sigma} D_{33}
$$

(E6.2.2)

The traction-free condition on the surface of the rod can be stated as

$$
\sigma_{22} = C_{2211}^{\sigma} D_{11} + C_{2222}^{\sigma} D_{22} + C_{2233}^{\sigma} D_{33} = 0
$$

$$
\sigma_{33} = C_{3311}^{\sigma} D_{11} + C_{3333}^{\sigma} D_{22} + C_{3333}^{\sigma} D_{33} = 0
$$

(E6.2.3)

If the rod is initially transversely isotropic (with the axis of symmetry coincident with the $x_1$-axis) the tangent moduli are related by $C_{1133}^{\sigma} = C_{1122}^{\sigma}$ and $C_{2222}^{\sigma} = C_{3333}^{\sigma}$. Furthermore, uniaxial stressing in the direction of the axis of isotropy preserves the transverse isotropy and these relations hold throughout the deformation. Solving Eq. (6.2.3), with these assumptions we obtain

$$
D_{22} = D_{33}, \quad D_{22} = -\frac{C_{2211}^{\sigma}}{C_{2222}^{\sigma} + C_{2233}^{\sigma}} D_{11}
$$

(E6.2.4)

Using Eq. (6.2.4) for $D_{22}$ and $D_{23}$ in Eq. (E6.2.3) gives the uniaxial relation

$$
\sigma_{11} = E^{\sigma} D_{11} \quad \text{or} \quad \begin{bmatrix} C^{\sigma \sigma} \end{bmatrix} = \begin{bmatrix} E^{\sigma} \end{bmatrix}
$$

(E6.2.5)

where $E^{\text{tang}}$ is the tangent modulus associated with the state of uniaxial stress and is given by

$$
E^{\sigma \sigma} = C_{1111}^{\sigma \sigma} - \frac{2C_{2211}^{\sigma \sigma} C_{1122}^{\sigma \sigma}}{C_{2222}^{\sigma \sigma} + C_{2233}^{\sigma \sigma}}
$$

(E6.2.6)

**Material Tangent Stiffness Matrix.** The tangent stiffness matrix for a rate-independent material is given by (6.4.21) in the current configuration which we write in the local coordinate system as

$$
\dot{K}^{\text{mat}} = \int_{\Omega} \dot{B}^{\sigma} \dot{C}^{\sigma \sigma} \dot{B} d\Omega
$$

(E6.2.7)

Using the $B$ matrix from (E4.6.3) and $C^{\sigma \sigma}$ as given by (E6.2.5), we obtain

$$
\dot{K}^{\text{mat}} = \int_{\xi} \left[ \begin{array}{ccc} -1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{array} \right] \left[ E^{\sigma \sigma} \right]^{-1} \left[ A \right] d\xi
$$

(E6.2.8)
Here, the $B$ matrix has been expanded to a $4 \times 1$ matrix by adding zeros to reflect that the $\dot{x}$-component of the rate-of-deformation is independent of the transverse velocities. If we assume $E^{\sigma_T}$ is constant in the element, then

$$
\dot{\mathbf{K}}_{\text{mat}} = \frac{AE^{\sigma_T}}{\ell} \begin{bmatrix}
+1 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 \\
-1 & 0 & +1 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
$$

(E6.2.9)

This is identical to the linear stiffness matrix for a rod if $E^{\sigma_T}$ is replaced by Young's modulus $E$. The material tangent stiffness in terms of global components is given by (4.5.42):

$$
\mathbf{K}_{\text{mat}} = \mathbf{T}^T \mathbf{K}_{\text{mat}} \mathbf{T}
$$

(E6.2.10a)

where $\mathbf{T}$ is given by

$$
\mathbf{T} = \begin{bmatrix}
cos \theta & sin \theta & 0 & 0 \\
-sin \theta & cos \theta & 0 & 0 \\
0 & 0 & cos \theta & sin \theta \\
0 & 0 & -sin \theta & cos \theta
\end{bmatrix}
$$

(E6.2.10b)

so

$$
\mathbf{K}_{\text{mat}} = \frac{AE^{\sigma_T}}{\ell} \begin{bmatrix}
\cos^2 \theta & \cos \theta \sin \theta & -\cos^2 \theta & -\cos\theta \sin \theta \\
\sin^2 \theta & -\cos\theta \sin \theta & -\sin^2 \theta & -\sin \theta \cos \theta \\
\cos^2 \theta & \cos \theta \sin \theta & -\sin^2 \theta & -\sin \theta \cos \theta \\
symmetric
\end{bmatrix}
$$

(E6.2.11)

where the material constant $E^{\sigma_T}$ relates the Truesdell rate of the Cauchy stress to the rate-of-deformation in a uniaxial state of stress.

**Geometric Stiffness Matrix.** The geometric stiffness is developed in a coordinate system that at time $t$ coincides with the axis of the bar but is fixed in time. Note that since the coordinate system is fixed in the orientation shown in Fig. ??, it is not a true corotational coordinate system, so the rotation corrections of an objective rate must be considered. We will use the Truesdell rate. We could also consider the $\dot{x}$, $\dot{y}$ coordinate system corotational and derive the geometric stiffness by accounting for the change of the transformation matrix $\mathbf{T}$ in (E4.6.11). Such derivations are given in Crisfield. The result should be identical, since the same mechanical effect is represented, but the derivation is generally more difficult. The geometric stiffness matrix is given by Eq. (6.4.21):.
\[
\dot{K}_{ij} = \dot{H}_{ij}, \quad \dot{H} = \int_{\Omega} \mathbf{B}^T \sigma \mathbf{B} d\Omega
\]  
(E6.2.12)

where the geometric stiffness has been expressed in the local coordinate system for simplicity. Using the \( \mathbf{B} \) matrix from Eq. (4.6.3), it follows that

\[
\dot{H} = \int_{\Omega} \mathbf{B}^T \sigma \mathbf{B} d\Omega
\]  
(E6.2.13)

Assuming that the stress is constant gives

\[
\dot{H} = \frac{\dot{\sigma}_x A}{\ell} \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix}
\]  
(E6.2.14)

Expanding the above, we obtain the geometric stiffness as given by (E6.2.12)

\[
\dot{K}_{\text{geo}} = \frac{A \dot{\sigma}_x}{\ell} \begin{bmatrix} +1 & 0 & -1 & 0 \\ 0 & +1 & 0 & -1 \\ -1 & 0 & +1 & 0 \\ 0 & -1 & 0 & +1 \end{bmatrix}
\]  
(E6.2.15)

Use of the transformation formula, Eq. (4.5.42), shows that the geometric stiffness is independent of the orientation of the beam.

\[
\dot{K}_{\text{geo}} = T^T \dot{K}_{\text{geo}} T = \dot{K}_{\text{geo}}
\]  
(E6.2.16)

The total tangent stiffness is then given by the sum of the material and geometric stiffnesses

\[
\dot{K} = \dot{K}_{\text{mat}} + \dot{K}_{\text{geo}}
\]  
(E6.2.17)

The matrix is symmetric, which is a consequence of choosing a constitutive equation in terms of the Truesdell rate of the convected rate of the Kirchhoff stress. The matrix is positive definite as long as the initial stress is small compared to the tangent modulus.

**Material Tangent Stiffness Matrix in Total Lagrangian Form.**

The material tangent stiffness matrix for a rate-independent material is given by Eq. (6.4.18) in the reference configuration

\[
\dot{K}_{\text{mat}} = \int_{\Omega_0} \mathbf{B}^T C^{SE} \mathbf{B} d\Omega_0
\]  
(E6.2.25)

Using the \( \mathbf{B} \) matrix from Eq. (E4.7??) and \( C^{SE} \) as given by Eq. (E6.2.5), we obtain
If we assume \( E^{SE} \) is constant in the element, then

\[
K_{mat} = \frac{1}{\ell_0} \begin{pmatrix}
-\cos \theta \\
-\sin \theta \\
\cos \theta \\
\sin \theta
\end{pmatrix}
E^{SE} \begin{pmatrix}
1 & -\cos \theta & -\sin \theta & \cos \theta & \sin \theta
\end{pmatrix} A_0 \ell_0 d\xi
\]  
(E6.2.26)

where the material constant \( E^{SE} \) relates the Truesdell rate of the Cauchy stress to the rate-of-deformation in a uniaxial state of stress. It can easily be shown via (???) that the above is identical to (E6.2.12).

**Geometric Stiffness Matrix in Total Lagrangian Form.** The geometric stiffness is developed from (6.4.15):

\[
K_{IJ} = H_{IJ} I \\
H = \int_{\Omega_0} \mathcal{B}_0^T \mathcal{S} \mathcal{B}_0 d\Omega_0
\]  
(E6.2.28)

where the \( \mathcal{B}_0 \) matrix is given in (4.6.3), so

\[
H = \int_{\Omega_0} \frac{1}{\ell_0} \begin{bmatrix}
-1 \\
+1
\end{bmatrix} [S_{11}] \frac{1}{\ell_0} \begin{bmatrix}
-1 \\
+1
\end{bmatrix} d\Omega_0
\]  
(E6.2.29)

Assuming that the stress is constant gives

\[
H = \frac{S_{11} A_0}{\ell_0} \begin{bmatrix}
+1 & -1 \\
-1 & +1
\end{bmatrix}
\]  
(E6.2.30)

Expanding the above, we obtain the geometric stiffness

\[
K_{geo} = \frac{A_0 S_{11}}{\ell_0} \begin{bmatrix}
+1 & 0 & -1 & 0 \\
0 & +1 & 0 & -1 \\
-1 & 0 & +1 & 0 \\
0 & -1 & 0 & +1
\end{bmatrix}
\]  
(E6.2.31)

The total tangent stiffness is then given by the sum of the material and geometric stiffnesses, (E6.2.17).
6.5. STABILITY: CONTINUATION AND ARCLENGTH METHODS

Stability. In nonlinear problems, stability of solutions is of considerable interest. There are many definitions of stability: stability is a concept that depends on the beholder and his objectives. However, some general definitions are widely accepted. We will here describe a theory of stability that originates from Liapunov(??) and is widely used throughout mathematical analysis, see Saybol(??) for a very lucid account of its computational application to a variety of problems. We will focus on its application to finite element methods.

We will first give a definition of stability and explore its implications. Consider a process that is governed by an evolution equation such as the equation of motion or the heat conduction equations. Let the solution for the initial conditions \( d_A(0) = d_A^0 \) be denoted by \( d_A(t) \). Now consider additional solutions for initial conditions \( d_B(0) = d_B^0 \), where \( d_B^0 \) are small perturbation of \( d_A^0 \). This means that \( d_B^0 \) is close to \( d_A^0 \) in some norm, i.e.

\[
\|d_A^0 - d_B^0\| \leq \varepsilon \tag{6.5.1}
\]

A solution is stable when for all initial conditions that satisfy (6.5.1), the solution satisfies

\[
\|d_A(t) - d_B(t)\| \leq C\varepsilon \quad \forall t > 0 \tag{6.5.2}
\]

To explain this definition, we specify the norm to be the \( \ell_2 \) norm. Note that all initial conditions which satisfy (6.5.1) lie in a neighborhood of \( d_A^0 \). It is often said that the initial conditions lie in a ball around \( d_A^0 \). The definition then states that for all initial conditions which lie in the ball around \( d_A^0 \), the solutions \( d_B(t) \) must lie in a ball around the solution \( d_A(t) \) for all time. This definition is illustrated for a system with two dependent variables in Fig. 6.7. The right-hand side shows the behavior of a stable system. Here we have only shown two solutions resulting from perturbations of the initial data, since it is impossible to show an infinite number of solutions. The left-hand side shows an unstable system. It suffices for a single solution starting in the ball about \( d_A^0 \) to diverge to indicate an unstable solution.

The applicability of this definition to processes we intuitively consider stable and unstable can be seen by the following examples. Consider a beam a beam loaded axially by a vertical load as shown in Fig. 6.8. We first consider the numerical response when the beam is perfectly straight. The lateral response in
this case is the path is denoted by \( d_A(t) \), and as can be seen, the lateral displacement is zero even though the load eventually exceeds the buckling load. If you don't believe this, try it. The beam will usually not buckle in an incremental solution or a dynamic solution with explicit or implicit integration. Only if roundoff error introduces a "numerical imperfection" will the perfectly straight beam buckle. We then plot the lateral displacement of the beam as we perturb the location of node 2, which can be considered an initial condition on the displacement of that node. The resulting paths are also shown in Fig. 6.8. It can be seen that when the load is below the buckling load, the paths for different initial conditions remain in a ball about the \( d_A(t) \). However, when the load exceeds the buckling load, the solutions for different initial conditions in the location of point A diverge drastically. Therefore any process in which the load exceeds the buckling load is unstable. Note that the direction of the divergence depends on the sign of the initial imperfection.

Another example is the flow of a liquid in a pipe. When the flow velocity is below a critical Reynold’s number, the flow is stable. A perturbation of the state leads to small changes in the evolution of the system. On the other hand, when the flow is above the critical Reynold’s number, a small perturbation leads to large changes because the flow changes from laminar to turbulent.

Stability is usually ascribed to a state, rather than a process. The definition is then identical: a state is stable if a small perturbation of that state results in a small differences for all time. When perturbations lead to large differences in the subsequent states of the system, the state is unstable. This concept fits within the framework of the definition of stability given by Eq. (6.5.1) with the state considered as the initial condition.

A common example of stable and unstable states often given in introductory dynamics texts is shown in Fig. 6.9. It is clear that state A is stable, since small perturbations of the position of the ball will not significantly change the evolution of the system. State B is unstable, small perturbations will lead to large changes: the ball can roll either to the right or to the left. State C is often called neutral stability in introductory texts. According to the definition of Eq. (6.5.1), state C is an unstable state, since small changes in the velocity will lead to large changes in the position as large times. Thus the definition of stability given in introductory texts does not completely conform to the one given here.

**Stability of Equilibrium Solutions.** To obtain a good understanding of the behavior of a system, its equilibrium paths, or branches, and their stability must be determined. It is often argued among structural mechanicians that the difficulties associated with unstable behavior can be circumvented by simply obtaining a dynamic solution. When a structure is loaded above its limit point or a bifurcation point in a dynamic simulation, the structure passes dynamically to the vicinity of the next stable branch and the instability is not apparent except for the onset of a different mode of deformation, such as the lateral deformation in a beam. However, to understand the behavior of a structure or process thoroughly, its static equilibrium behavior should be carefully examined. Many vagaries of structural behavior may be hidden by dynamic simulations. For example, when the fundamental path bifurcates with an asymmetric branch as shown in Fig. 6.10, the structure becomes very sensitive to imperfections. The theoretical
bifurcation point is not a realistic measure of the strength; an actual structure will buckle at a much lower load than the theoretical value because imperfections are unavoidable. A single numerical simulation could miss this sensitivity completely. This sensitivity to imperfections for cylindrical shells was analyzed by Koiter(??) and is a classical example of imperfection sensitivity.

As a first step in studying the equilibrium behavior of a system, the load and any other parameters of interest, such as the temperature or an active control, must be parametrized. Up to now we have parametrized the load by the time \( t \), which is convenient in many practical problems. However, a single parameter does not always suffice in the study of equilibrium problems. We will now parametrize the load by \( n_{\gamma} \) parameters \( \gamma_{a} \), so the load is then given by \( \gamma_{a}q_{a} \), where \( q_{a} \) represent a distributed loading such as a pressure or concentrated loads. We use the convention that repeated indices are summed over the range, in this case \( n_{\gamma} \). For distributed loads, the parameter \( \gamma_{a} \) should not be applied directly to the external nodal forces, since the external nodal forces will depend on the nodal displacements. The discrete loads can be parametrized by \( \gamma_{a}f_{a}^{ext} \), where \( f_{a}^{ext} \) are column matrices of nodal external forces associated with a loading mode \( a \).

Our intention is to trace the equilibrium behavior of the model as a function of the parameters \( \gamma_{a} \). The problem then is then is to find \( d(\gamma_{a}) \) such that

\[
\mathbf{r}(d(\gamma_{a})) = 0
\]  
(6.5.2b)

For purposes of characterizing the nonlinear system, the solutions are usually grouped into branches, which are continuous lines describing the response for one change of one parameter. Branches along which the solution is in equilibrium, i.e. satisfies Eq(6.5.2b), are called equilibrium branches, regardless of whether they are stable or unstable.

Nonlinear systems exhibit three types of branching behavior:

1. turning points, usually called limit points in structural analysis, in which the slope of the branch changes sign;
2. stationary bifurcations, often called simply bifurcations, in which two equilibrium branches intersect.
3. Hopf bifurcations, in which an equilibrium branch intersects with a branch on which there is periodic motion.

The behavior of the shallow truss exhibits a limit (or turning) point, as can be seen from Fig. 6.11. Subsequent to a turning point, a branch can be either stable or stable. In this case, as shown in the analysis of the problem in Example 6.4, the branch after the first limit point, point \( a \), is unstable, while the branch after the second limit point, point \( b \), is stable.

The beam problem shown in Fig. 6.12 is a classical example of a bifurcation. The point \( b \) where the two branches intersect is the point of bifurcation. Subsequent to the bifurcation point, the continuation of the
fundamental branch \( ab \), becomes unstable. Point \( b \), the bifurcation point, corresponds to the classical buckling load of the Euler beam. This type of branching is often called a pitchfork, (do you see the hay on the end?)

Hopf bifurcations are quite uncommon in passive structures. They are found in general nonlinear behavior and can be seen in structures under active control. In a Hopf bifurcation, stable equilibrium solutions become impossible at the end of a branch. Instead, there are two branches with periodic solutions. An example of a Hopf bifurcation is given in Example ??.

Methods of continuation and arclength methods. The tracing of branches is called a continuation method. The tracing of equilibrium branches is often quite difficult and robust, automatic procedures for continuation are not available. In the following, we describe continuation methods base on parametrization, such as the arc length method. In the arc length method, the arc length along the equilibrium path replaces the load as the incremental parameter. It enables branches to be followed around turning points, which is critical to the successful continuation of equilibrium branches.

We first consider continuation with the arc length method for the case of a single load parameter. In tracing the branches in a model with a single load parameter, the load parameter is usually started at zero and incremented. For each load increment, an equilibrium solution is computed, i.e. we find \( d^{n+1} \), a solution to

\[
\mathbf{r}(d^{n+1}, \gamma^{n+1}) = 0 \quad \text{or} \quad f^{\text{int}}(d^{n+1}) - \gamma^{n+1}f^{\text{ext}} = 0 \tag{6.5.3}
\]

where \( n \) is the step number and \( f^{\text{ext}} \) is the load distribution chosen for tracing the branch. We assume that the distribution of nodal external forces does not change with the deformation of the model; for situations where the load multiplier is on a distributed load such as a pressure, the methods must be changed. The inertial term is not included in the above because continuation methods are applicable only to equilibrium problems.

One of the most widely used continuation methods in structural mechanics is the arc length method. Instead of incrementing the load parameter \( \gamma \) to trace the branch, a measure of the arclength is incremented. This is accomplished by adding a the following equation to the equilibrium equations

\[
p(d, \gamma) = 0 \tag{6.5.4}
\]

In the arclength method the parametrization equation is

\[
p(d, \gamma) = (d - d^n)^T(d - d^n) + \alpha \Delta \gamma^2 - \Delta s^2 = 0 \quad \Delta \gamma = \gamma^{n+1} - \gamma^n \tag{6.5.4b}
\]

where \( \Delta s \) is an approximate value of the arclength along the force-displacement path to be traversed during the step and \( \alpha \) is a scaling factor. Many other types of parametrization equations can be devised, and some are described later.
The total system of equations then consists of the equilibrium equations and the parametrization equation, so we have

\[
\begin{align*}
\mathbf{r}(\mathbf{d}, \gamma) &= \mathbf{0} \\
\mathbf{p}(\mathbf{d}, \gamma) &= \mathbf{0}
\end{align*}
\] (6.5.5)

The load parameter \( \gamma \) is treated as an additional unknown and the arclength \( s \) is now incremented instead of the parameter \( \gamma \).

This procedure is most easily explained for a one degree-of-freedom problem such as the shallow truss shown in Fig. 6.13. The fundamental branch is shown in Fig. 6.13 and we assume that a solution has been obtained at point \( n \). The arclength equation (6.5.4b) when viewed in the \( \gamma, d_y \) plane is the circle about point \( n \); in a problem with two displaced degrees of freedom, it would be a sphere about the point. In solving the parametrized equations, (6.5.5), we seek a solution which is the intersection of the equilibrium branch with the circle about the last solution point, which gives the solution shown as point \( n+1 \) in Fig. 6.13. Thus, while incrementing the load parameter would be fruitless at point \( n \), the problem has been restated in terms of the arclength along the branch so that a solution with a lower load can be found.

The parametrized equations for the symmetric truss can then be posed as follows: find a solution to

\[
\begin{align*}
r_i(d_1, \gamma) &= -f_i(d_1, \gamma) = 0 \\
\text{subject to } (\gamma(s) - \gamma^n)^2 + (d_1(s) - d_1^n)^2 - \Delta s^2 &= 0
\end{align*}
\] (6.5.6)

Alternatively, we can write the above in terms of increments in the displacements and the load parameters as: find a solution to

\[
\begin{align*}
f_i &= 0 \quad \text{subject to } \Delta \gamma^2 + \left(\Delta d_1\right)^2 = \Delta s^2
\end{align*}
\] (6.5.7)

Thus the original set of discrete equations in one unknown is augmented by a second equation and a second unknown \( \gamma \) is added. The load need not increase in the step, and may in fact decrease. It is only necessary for the arclength parameter to increase, which is a perfectly natural way of tracing the branch.

The resulting equations can be solved by the standard Newton methods we have described. Let

\[
\delta \mathbf{d} = \mathbf{d}_{v+1}^{n+1} - \mathbf{d}_v^{n+1} = \mathbf{d}_{\text{new}} - \mathbf{d}_{\text{old}} \quad \delta \gamma = \gamma_{v+1}^{n+1} - \gamma_v^{n+1} = \gamma_{\text{new}} - \gamma_{\text{old}}
\]

Note that \( \delta \) is used as a change of a variable during an iteration and does not refer to a variation here. The linearized equations for the Newton method are given by
Using the definitions

\[
\begin{bmatrix}
\frac{\partial F}{\partial d} & \frac{\partial F}{\partial \gamma} \\
\frac{\partial \mathcal{P}}{\partial d} & \frac{\partial \mathcal{P}}{\partial \gamma}
\end{bmatrix}
\begin{bmatrix}
\delta d \\
\delta \gamma
\end{bmatrix} =
\begin{bmatrix}
-r_v \\
-p_v
\end{bmatrix}
\] (6.5.8)

where the Jacobians of the nodal forces have been expressed in terms of the internal tangent stiffness and the load stiffness on the LHS. As can be seen, the above equations are not symmetric, so in larger problems considerable advantage can be gained by solving the parametrization equation separately. Difficulties can also arise because the augmented system has two solutions, and the solution which retraces the path is not wanted.