

CHAPTER ONE

INTRODUCTION

1.1 NONLINEAR SOLID DYNAMICS

The nonlinear analysis of the mechanical behavior of solid continua can be categorized in a number of ways. Solids may exhibit material or geometric nonlinearity. In the former the constitutive behavior, that is, the stress-strain relationship will be nonlinear and in the latter geometric changes, such as large rotations, affect the behavior. In many situations, such as metal forming, both occur simultaneously. A further category is whether the response of the solid to loading, be it forces or temperature, is dynamic or static, or in other words time dependent or not, and to be more precise whether inertial forces are relevant or can be ignored. In a previous text entitled *Nonlinear Solid Mechanics for Finite Element Analysis: Statics* the authors covered the fundamental nonlinear continuum mechanics necessary for the development of the equilibrium equations and their eventual solution using finite element discretization. The present text extends that development into the nonlinear dynamic realm and, whereas it is reasonably self-contained, it is useful to have an awareness of the material in the companion Statics text.

The dynamic response of solids may be linear or nonlinear. Linear response is generally associated with small deformation vibration behavior about an equilibrium position where geometrically nonlinear effects are normally insignificant, but not always, as in the case of a vibrating string in tension. Examples of dynamic behavior that can be considered in the linear regime are the vibration of buildings under moderate earthquakes which do not take the structure anywhere near its possible failure. Nonlinear dynamic behavior is characterized primarily by the presence of large rotation in addition to possible large strain. For example, satellites exhibit large rotation small strain behavior, whereas the collapse of, say, a building due to an extreme earthquake or a high speed impact situation is the result of both large rotation and large strain.

As the title implies, the solid continua are modeled in a discrete manner by an assemblage of three-dimensional finite elements for which formulations are developed to represent the geometric and material behavior due to the motion of the solid with respect to time as a result of various types of loading. In general the spatial representation of the body in terms of finite elements is similar between dynamic and static applications. The main difference emerges in relation to the introduction of inertial forces due to the mass and acceleration of the body and the presence of time as a key independent variable.

Time being the essential difference between static and dynamic behavior means that procedures need to be devised in order to progress the motion of the solid as time progresses. These procedures are generally known as *time-stepping schemes* whereby the motion, that is, primarily the velocity and thence the position, are discretized in time. This means that the motion between discrete time steps is approximated in some way. A number of such time-stepping schemes are presented in this text, from the simple leap-frog scheme to more complex and general Newmark schemes.

Fundamental to the description of dynamic behavior, linear or nonlinear, is Newton's Second Law of Motion, giving a dynamic equilibrium equation relating force, mass, and acceleration. Motion is progressed by determining the acceleration and then using the time-stepping scheme to find the velocity and advance the position. In mathematical terms this implies the solution of a second-order equation in time by an appropriate time-stepping scheme.

An alternative approach presented in this text involves the reformulation of the dynamic equilibrium of a solid into a system of first-order conservation laws for the physical and geometric variables describing the solid and its motion. This process leads to a mixed set of unknowns incorporating both velocities and strains. Similar sets of conservation laws are used extensively in computational fluid dynamics and the discretization of such laws via the finite element method is well understood and routinely applied. In this text both the traditional displacement-based approach, leading to a second-order system of equations in time, and the first-order set of conservation laws will be presented to solve solid dynamics problems. In the final chapters a number of examples will be solved using both these approaches to demonstrate their validity and general applicability.

The general aim of this book is to provide the reader with a good understanding of the necessary continuum mechanics concepts and theory required to successfully model by finite elements the time-dependent large deformation of solids, including possible thermal effects. It will cover important continuum concepts such as virtual work, potential and kinetic energy, the Lagrangian, and Hamilton's principle, as well as thermodynamics and thermoelasticity. It will

also present time-discretization issues such as energy and momentum conserving schemes and symplectic integrators or advanced finite element discretization technologies such as Petrov–Galerkin methods for first-order conservation laws. Although the authors have made every effort to keep the text self-contained, the reader would benefit from some degree of familiarity with the contents of the companion statics volume: *Nonlinear Solid Mechanics for Finite Element Analysis: Statics*. This text will be referred to in the remainder of this book as the NL-Statics Volume.

The remainder of this chapter sets out to provide a gentle introduction to nonlinear dynamic behavior via simple one- or two-degrees-of-freedom examples. These examples are used to introduce simple time integration schemes such as the leap-frog method or the mid-point rule and discuss important issues associated with these schemes such as stability, or to compare implicit versus explicit methodologies.

1.2 ONE-DEGREE-OF-FREEDOM NONLINEAR DYNAMIC BEHAVIOR

In this section the torsion spring supported single rigid column discussed in Section 1.2.2 of NL-Statics is re-examined to demonstrate a simple example of nonlinear dynamic behavior. In order to introduce inertial forces, the column supports a mass which is restrained by a linear elastic torsion spring and viscous torsion damping at the bottom hinge, see Figure 1.1. The governing equations do not admit an analytical solution and consequently a simple numerical “leap-frog” integration in time will be introduced in order to obtain the behavior of the column with respect to time.

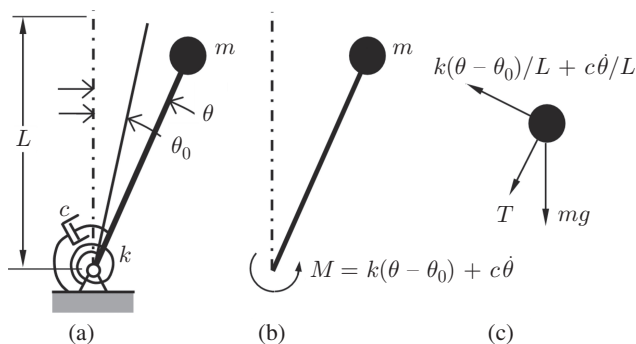


FIGURE 1.1 Simple column.

1.2.1 Equation of Motion

The column shown in Figure 1.1, where the angular motion of the mass m , from an initial angle of θ_0 , is constrained by a torsion spring having stiffness k and, in parallel, torsional viscous damping having a coefficient c . The effect of introducing viscosity into the system will be demonstrated below. The torsion spring produces a moment $k(\theta - \theta_0)$ which in turn is equivalent to a tangential force on the mass equal to $k(\theta - \theta_0)/L$. The viscous damping produces a similar force but now equal to $c\dot{\theta}/L$, where $\dot{\theta}$ is the angular velocity. Elementary kinematics gives the tangential and radial acceleration of the mass m in terms of $\dot{\theta}$ and the tangential acceleration $\ddot{\theta}$ as $L\ddot{\theta}$ and $L\dot{\theta}^2$, respectively.

Employing Newton's Second Law of Motion the dynamic equilibrium equations in the tangential and radial directions are

$$mL \frac{d^2\theta}{dt^2} = mg \sin \theta - \frac{c}{L} \frac{d\theta}{dt} - \frac{k}{L}(\theta - \theta_0), \quad (1.1a)$$

$$mL \left(\frac{d\theta}{dt} \right)^2 = T + mg \cos \theta. \quad (1.1b)$$

Observe that the first of these equations is sufficient to determine the motion of the column via the evaluation of $\theta(t)$, whereas the second equation enables the calculation of the tension, T , in the column once the angle $\theta(t)$ and angular velocity $\dot{\theta}(t)$ have been obtained. The geometric nonlinearity in, for example, the first of the above equations is enshrined in the tangential gravitational force term $mg \sin \theta$.

As mentioned above, unlike the static column example considered in NL-Statics Section 1.2.2, the above equations do not readily admit an analytical solution and have to be solved using a numerical time integration scheme which enables the angular velocity $\dot{\theta}(t)$ and hence the angular position $\theta(t)$ to be calculated after the acceleration $\ddot{\theta}(t)$ has been found from Equation (1.1a). Numerous time integration schemes exist and a number of these will be introduced in this book, but for now a simple robust scheme known as leap-frog time integration will be employed.

1.2.2 Leap-Frog Time Integration

The column dynamic tangential equilibrium Equation (1.1a) can be rewritten to give the tangential acceleration as

$$\mathbf{a} = \frac{g}{L} \sin(\mathbf{x}) - \frac{c}{mL^2} \mathbf{v} - \frac{k}{mL^2}(\mathbf{x} - \mathbf{X}); \quad \mathbf{a} = \frac{d^2\theta}{dt^2}; \quad \mathbf{v} = \frac{d\theta}{dt}; \quad \mathbf{x} = \theta, \quad (1.2)$$

where the notation for acceleration $\mathbf{a}$, velocity $\mathbf{v}$, and coordinate position $\mathbf{x}$ provide for a general description of the time-stepping scheme* that will be valid for multiple-degree-of-freedom problems.

* For the column problem $\mathbf{x} = \theta$, $\mathbf{v} = \dot{\theta}$, and $\mathbf{X} = \theta_0$.

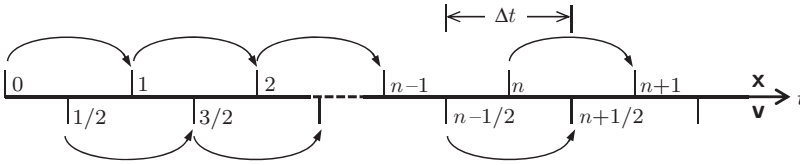


FIGURE 1.2 Leap-frog time integration.

The time during which the motion takes place is now divided into a number of equal time steps Δt , where a typical time is labeled n , see Figure 1.2. Using the above acceleration $\mathbf{a}$, assuming $\mathbf{v}$ and $\mathbf{x}$ to be known, an approximate integration over a time step Δt can be employed to calculate the velocity at the half-time-step $t_{n+1/2}$ in terms of the previous half time step velocity and the time step $\Delta t = t_{n+1/2} - t_{n-1/2}$ as

$$\mathbf{v}_{n+1/2} = \mathbf{v}_{n-1/2} + \int_{t_{n-1/2}}^{t_{n+1/2}} \mathbf{a} dt \approx \mathbf{v}_{n-1/2} + \mathbf{a}_n \Delta t. \quad (1.3)$$

Similarly the updated coordinates $\mathbf{x}_{n+1}$ are found using the velocity at the previous half time step above to yield

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \int_{t_n}^{t_{n+1}} \mathbf{v} dt \approx \mathbf{x}_n + \mathbf{v}_{n+1/2} \Delta t. \quad (1.4)$$

This process, illustrated in Figure 1.2, is an example of a staggered time-stepping scheme.

Notice in Figure 1.2 that at the first time step $n = 0$ the velocity at $\mathbf{v}_{1/2}$ is required to start the process. Given an initial velocity $\mathbf{v}_0$ and an initial acceleration $\mathbf{a}_0$ calculated from Equation (1.2), $\mathbf{v}_{1/2}$ can be found using the approximation

$$\mathbf{v}_{1/2} = \mathbf{v}_0 + \mathbf{a}_0 \frac{\Delta t}{2}. \quad (1.5)$$

Observe that, given the staggered nature of the scheme, unless the viscosity vanishes, that is $c = 0$, the velocity in Equation (1.2) is not generally known at time n and needs to be taken at $t_{n-1/2}$ in order to allow a direct or explicit evaluation of the acceleration, $\mathbf{a}_n$, at time n as

$$\mathbf{a}_n = \frac{g}{L} \sin(\mathbf{x}_n) - \frac{c}{mL^2} \mathbf{v}_{n-1/2} - \frac{k}{mL^2} (\mathbf{x}_n - \mathbf{X}). \quad (1.6)$$

The exception to this rule occurs at $n = 0$, when $\mathbf{v}_0$ is actually known from the initial conditions, and hence $\mathbf{a}_0$ can be evaluated as[†]

$$\mathbf{a}_0 = \frac{g}{L} \sin(\mathbf{x}_0) - \frac{c}{mL^2} \mathbf{v}_0 - \frac{k}{mL^2} (\mathbf{x}_0 - \mathbf{X}). \quad (1.7)$$

[†] $\mathbf{x}_0$ need not be equal to $\mathbf{X}$; see Exercise 5 of Chapter 2.

It is clear that the replacement of $\mathbf{v}_n$ by $\mathbf{v}_{n-1/2}$ in Equation (1.6) introduces an error in the calculation of $\mathbf{a}_n$. This can be avoided by using the correct $\mathbf{v}_n$ expressed as $\mathbf{v}_n = \mathbf{v}_{n-1/2} + (\Delta t/2)\mathbf{a}_n$ to give an expression for $\mathbf{a}_n$ as

$$\mathbf{a}_n = \frac{g}{L} \sin(\mathbf{x}_n) - \frac{c}{mL^2} \left(\mathbf{v}_{n-1/2} + \frac{\Delta t}{2} \mathbf{a}_n \right) - \frac{k}{mL^2} (\mathbf{x}_n - \mathbf{X}). \quad (1.8)$$

Consequently, $\mathbf{a}_n$ is now implicit in its own evaluation. For a one-degree-of-freedom problem this is not a problem as it would suffice to move all terms containing $\mathbf{a}_n$ to the left of the equation and divide the remaining right side by the accumulated coefficient of $\mathbf{a}_n$, to yield

$$\mathbf{a}_n = \left(\frac{g}{L} \sin(\mathbf{x}_n) - \frac{c}{mL^2} \mathbf{v}_{n-1/2} - \frac{k}{mL^2} (\mathbf{x}_n - \mathbf{X}) \right) / \left(1 + \frac{c}{mL^2} \frac{\Delta t}{2} \right). \quad (1.9)$$

Whereas Equation (1.6) is an explicit equation, Equation (1.8) is an implicit equation requiring a solution for $\mathbf{a}_n$. In the single-degree-of-freedom case, solving Equation (1.9) is trivial, but for realistic simulations containing large numbers of degrees of freedom k , c , and m are matrices, which leads to a system of equations requiring solution. Generally, time-stepping schemes in which the variables can be updated without solving systems of equations are known as *explicit*, whereas schemes that require the solution of a system of equations are known as *implicit*, and if this is required at every time step such schemes are computationally costly.

In comparison to an implicit scheme, the greater efficiency of an explicit scheme would seem preferable; however, it will be seen later that they suffer from severe limitations with respect to time-step size before producing grossly inaccurate solutions. These time-step limitations are known as stability restrictions and will be discussed in Section 1.3.3 below. Implicit time-stepping schemes are usually constructed in such a manner that avoids stability restrictions and consequently allow much larger time steps to be used, albeit at greater cost per step. The leap-frog algorithm is shown in Box 1.1. This explicit algorithm is of general applicability and as a consequence does not include the correction implied by Equation (1.9).

BOX 1.1: Leap-frog algorithm

- INPUT geometry, material properties, and solution parameters
- INITIALIZE $\mathbf{x}_0$, $\mathbf{v}_0$
- SET $\mathbf{a}_0$ (1.7) using initial values
- SET $\mathbf{v}_{1/2}$ (1.5)
- DO WHILE $t < t_{max}$ (time steps)
 - SET $\mathbf{x} = \mathbf{x} + \mathbf{v} \Delta t$ (1.4)
 - FIND $\mathbf{a}$ (1.6)

(continued)

Box 1.1: (*cont.*)

- SET $\mathbf{v} = \mathbf{v} + \mathbf{a} \Delta t$ (1.3)
- ENDDO

1.2.3 Column Examples

The following examples employ the leap-frog time integration discussed in Section 1.2.2. In Figure 1.3 the viscous coefficient is $c = 0$ and in Figure 1.4 the viscous coefficient is $c = 650$.[‡] Both examples have an initial angle of $\theta_0 = 45^\circ$, $L = 10$, $m = 100$, $g = 9.81$, $k = 1000$, $\Delta t = 0.0001$, and the initial tangential velocity is $\mathbf{v}_0 = 0$. Allowing the time to extend to $t = 200$ reveals that the damped solution converges to the static solution of $\theta \approx 167.42^\circ$.

1.3 TWO-DEGREES-OF-FREEDOM EXAMPLE

The spring-mass system shown in Figure 1.5 will be used to introduce the effects of geometric nonlinearity in a dynamic situation. The internal force in the spring is $\mathbf{T}$ and the spring has a stiffness k . At time $t = 0$ the orientation of the spring is

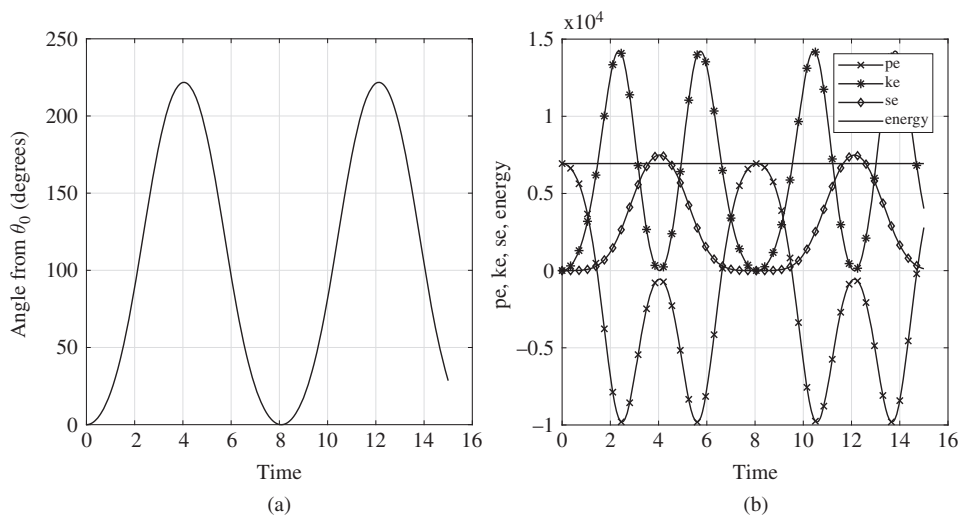


FIGURE 1.3 Column leap-frog with $\theta_0 = 45^\circ$, $c = 0$: (a) Angle from θ_0 –time; (b) Energy–time (pe = potential energy, se = elastic (strain) energy, ke = kinetic energy, total energy).

[‡] For this case the correction for a_n given by the last term in Equation (1.9) involving the viscosity c is negligible and is not employed in the solution.

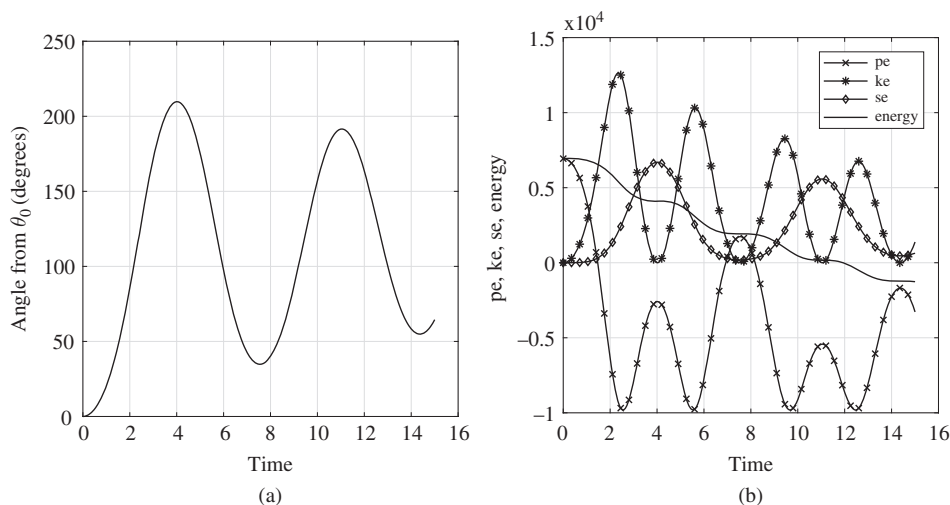


FIGURE 1.4 Column leap-frog with $\theta_0 = 45^\circ$, $c = 650$: (a) Angle from θ_0 -time; (b) Energy-time.

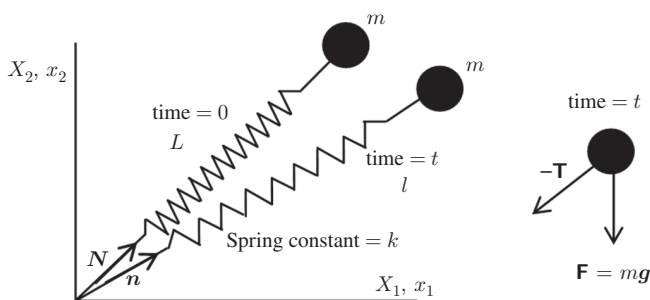


FIGURE 1.5 Two-degrees-of-freedom example.

given by the unit normal $\mathbf{N}$ and the length is L , while at time t the orientation is given by $\mathbf{n}(t)$ and the length is $l(t)$. The unit vectors $\mathbf{N}$ and $\mathbf{n}(t)$ at time $t = 0$ and t respectively are determined by the corresponding mass coordinates $\mathbf{X}$ and $\mathbf{x}(t)$ as

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}; \quad \mathbf{N} = \frac{1}{L} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}, \quad (1.10a,b)$$

$$\mathbf{x}(t) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}; \quad \mathbf{n}(t) = \frac{1}{l} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}; \quad l = \sqrt{\mathbf{x} \cdot \mathbf{x}}. \quad (1.10c,d,e)$$

Time derivatives of the coordinate $\mathbf{x}(t)$ give the velocity and acceleration as

$$\mathbf{v}(t) = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}; \quad \mathbf{v}(t) = \frac{d\mathbf{x}}{dt}; \quad \mathbf{a}(t) = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}; \quad \mathbf{a}(t) = \frac{d^2\mathbf{x}}{dt^2}. \quad (1.11a,b,c,d)$$

For simplicity the explicit dependency of quantities upon time will be removed, for example, $\mathbf{x}(t)$ becomes $\mathbf{x}$, etc. The internal spring force $\mathbf{T}$ and the vertical gravitational force $\mathbf{F}$ are

$$\mathbf{T}(\mathbf{x}) = T\mathbf{n}; \quad T = k(l - L); \quad \mathbf{F} = \begin{bmatrix} 0 \\ -mg \end{bmatrix}. \quad (1.12a,b)$$

Obviously, on Earth at least, $g = 9.81 \text{ m/s}^2$.

1.3.1 Equations of Motion

Employing Newton's Second Law of Motion applied to the mass m at time t , the equations of motion are written in terms of the acceleration $\mathbf{a}$, internal force $\mathbf{T}$, and external force $\mathbf{F}$ as

$$\begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 \\ -mg \end{bmatrix} - \frac{T}{l} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \text{or simply } \mathbf{M}\mathbf{a} = \mathbf{F} - \mathbf{T}(\mathbf{x}). \quad (1.13a,b)$$

Equation (1.12a,b) reveals that the internal force $\mathbf{T}$ at time t is a function of the length l and unit normal $\mathbf{n}$, both being functions of the current position $\mathbf{x}$ of the mass. Consequently, the equations of motion (1.13a,b) are geometrically nonlinear. Such equations do not admit an analytical solution and have to be solved numerically in the following section using the leap-frog scheme.

EXAMPLE 1.1: Energy conservation

The two-degrees-of-freedom spring-mass system described in this section is a convenient example with which to illustrate the conservation of physical quantities such as energy. The equilibrium Equation (1.13a,b) for the mass at time t and position $\mathbf{x}$ can be rewritten as

$$m\mathbf{a} + k(l - L)\mathbf{n} = m\mathbf{g},$$

where

$$\mathbf{a} = \frac{d\mathbf{v}}{dt}; \quad \mathbf{n} = \frac{\mathbf{x}}{l}; \quad l = \|\mathbf{x}\|; \quad \mathbf{g} = -g \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

(continued)

Example 1.1: (*cont.*)

Multiplying the equilibrium equation by $\mathbf{v}$ and rearranging gives

$$m\mathbf{a} \cdot \mathbf{v} + k(l - L)\mathbf{n} \cdot \mathbf{v} + mg v_2 = 0,$$

noting that

$$\mathbf{a} \cdot \mathbf{v} = \frac{1}{2} \frac{d}{dt}(\mathbf{v} \cdot \mathbf{v}) = \frac{1}{2} \frac{dv^2}{dt}; \quad v = \|\mathbf{v}\|; \quad v_2 = \frac{dx_2}{dt},$$

and

$$\mathbf{n} \cdot \mathbf{v} = \frac{\mathbf{x}}{l} \cdot \frac{d\mathbf{x}}{dt} = \frac{1}{2} \frac{1}{l} \frac{d}{dt}(\mathbf{x} \cdot \mathbf{x}) = \frac{1}{2} \frac{1}{l} \frac{dl^2}{dt} = \frac{dl}{dt}$$

gives, after some simple algebra,

$$\frac{d}{dt} \left[\frac{1}{2}mv^2 + \frac{1}{2}k(l - L)^2 + mgx_2 \right] = 0.$$

The term in the square bracket is the total energy of the mass m comprising the kinetic, elastic, and potential components, thus demonstrating the conservation of total energy.

1.3.2 Leap-Frog Examples

Using the two-degrees-of-freedom formulation, the column problem given in Section 1.2.3 can be rerun with a high linear spring constant of $k = 10^5$ to approximate a rigid column. This is equivalent to the column case shown in Figure 1.1 with a torsion spring constant of value zero. Figure 1.6 shows the results for the two-degrees-of-freedom simulation. When comparing with Figure 1.3, note that due to the presence of the torsion spring in Figure 1.3 the maximum angle is, as expected, less than that given in Figure 1.6 where the torsion spring stiffness is necessarily zero.

The next example is essentially a simple linear spring-mass problem with one degree of freedom. In order to achieve this, the mass is constrained to move along a fixed axis, as shown in Figure 1.7 (see also Exercise 2). This example will show that for a stable implementation of the leap-frog solution the value of the time step must be such that $\Delta t < 2/\omega$ where $\omega = (k/m)^{1/2}$. For a mass of unity and a spring stiffness $k = 1000$, Figure 1.8 shows the leap-frog solutions for time step $\Delta t = 0.01/\omega$ (solid plot) and $\Delta t = 2.01/\omega$ (dashed plot). For $\Delta t < 2/\omega$ the solution is identical to that given analytically in Exercise 2, having a maximum amplitude of $(1 - \cos(\omega t)) = 2$, whereas for $\Delta t > 2/\omega$ the numerical solution is clearly unstable. More interesting examples will be reserved for Section 1.3.6.

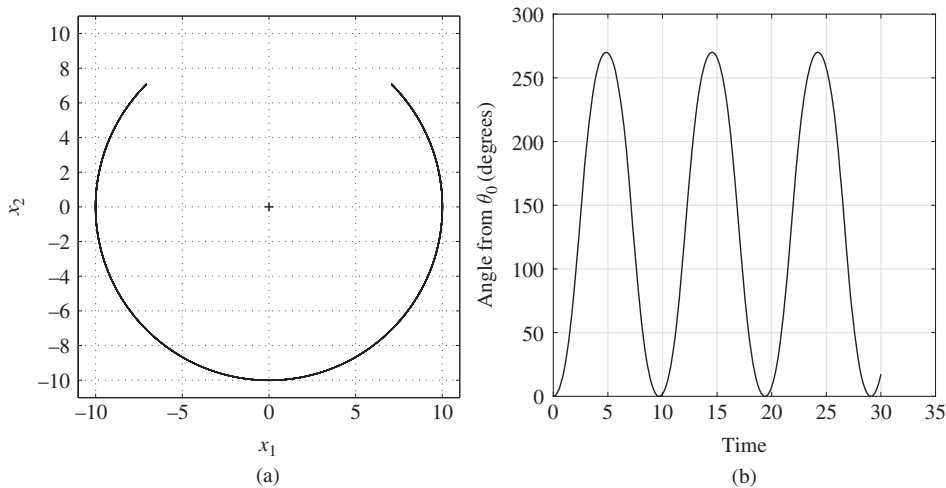


FIGURE 1.6 Column with torsion spring constant $k = 0$ as a 2 dof problem, linear spring constant $k = 10^5$, $\theta_0 = 45^\circ$. (a) Motion; (b) Angle from θ_0 -time.

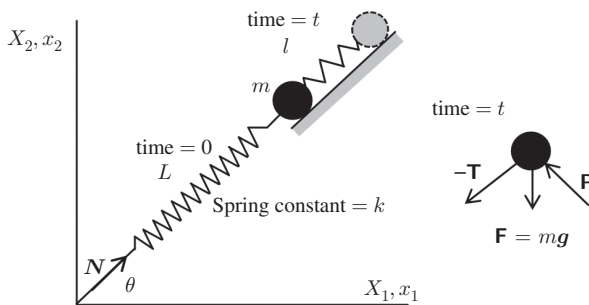


FIGURE 1.7 Constrained spring-mass system.

1.3.3 Time Integration – Stability

The preceding example demonstrated the problem of the numerical stability of the time integration scheme. This is generally complex to analyze for nonlinear systems with multiple degrees of freedom but can readily be predicted in the case of the single-degree-of-freedom system discussed above. In this case, when $\theta = 0$ (see Figure 1.7), gravity is not effective, and thus, in the absence of the forcing term, the dynamic equilibrium equation is simply

$$m \frac{d^2 x}{dt^2} + k x = 0; \quad x = x_1 - X_1, \quad (1.14a,b)$$

which has the analytical solution

$$x = \frac{v_0}{\omega} \sin(\omega t) + x_0 \cos(\omega t); \quad \omega^2 = \frac{k}{m}, \quad (1.15a,b)$$

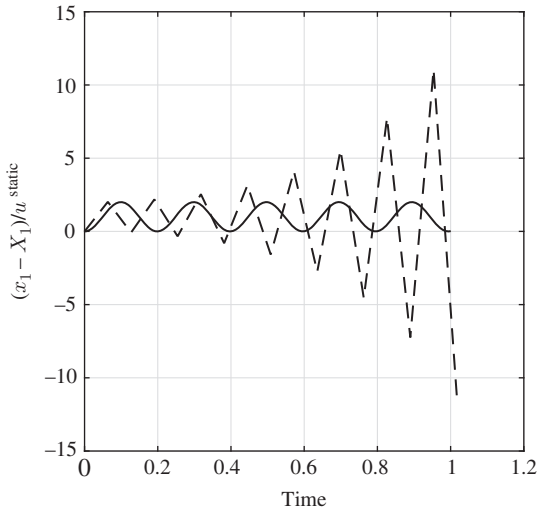


FIGURE 1.8 Constrained spring mass, $\theta_0 = 45^\circ$ numerical stability.

where x_0 and v_0 are the initial displacement and velocity respectively. Integration of Equation (1.14a,b) using the leap-frog method gives

$$v_{n+1/2} = v_{n-1/2} - \Delta t \frac{k}{m} x_n, \quad (1.16)$$

and

$$\begin{aligned} x_{n+1} &= x_n + \Delta t v_{n+1/2} \\ &= x_n + \Delta t \left(v_{n-1/2} - \Delta t \frac{k}{m} x_n \right). \end{aligned} \quad (1.17)$$

The intermediate velocity $v_{n-1/2}$ in the above equation can be eliminated by noting that $v_{n-1/2} = (x_n - x_{n-1})/\Delta t$ to give, after some simple algebra, a difference equation relating x_{n-1} , x_n , and x_{n+1} as

$$x_{n+1} - (2 - \omega^2 \Delta t^2) x_n + x_{n-1} = 0; \quad \omega^2 = \frac{k}{m}. \quad (1.18)$$

This is the discrete version of the continuous differential Equation (1.14a,b) that results from the use of the leap-frog integration scheme. The values of x_n for $n = 1, \dots, N$ can be predicted analytically by assuming a pattern of the type

$$x_{n+1} = A x_n, \quad x_n = A x_{n-1}, \dots, \quad (1.19a,b)$$

where A is the amplification factor between contiguous steps. Substituting this Equation (1.19a,b)_a for x_{n+1} into Equation (1.18) and using Equation (1.19a,b)_b to express x_{n-1} in terms of x_n gives, after dividing by x_n , an equation for A as

$$A^2 - (2 - \omega^2 \Delta t^2) A + 1 = 0, \quad (1.20)$$

which has solutions

$$A = \left(1 - \frac{\omega^2 \Delta t^2}{2}\right) \pm \sqrt{\left(1 - \frac{\omega^2 \Delta t^2}{2}\right)^2 - 1}. \quad (1.21)$$

Note that for values of A such that $|A| > 1$ the solution x_n will grow exponentially, very much unlike the analytical solution which has an oscillating nature. This is known as an unstable time integration process and must be avoided by adequate choice of Δt . A simple inspection of Equation (1.21) shows that for small enough values of Δt the discriminant is negative and the solution is therefore complex, that is,

$$A = \left(1 - \frac{\omega^2 \Delta t^2}{2}\right) \pm i \sqrt{1 - \left(1 - \frac{\omega^2 \Delta t^2}{2}\right)^2}. \quad (1.22)$$

In such cases it is easy to show that $|A| = 1$ and the solution for x_n exhibits a similar oscillating behavior as the exact solution. The boundary between $|A| = 1$ and $|A| > 1$ is defined by the expression

$$1 - \left(1 - \frac{\omega^2 \Delta t^2}{2}\right)^2 \geq 0, \quad (1.23)$$

which leads to the constraints

$$-1 \leq \left(1 - \frac{\omega^2 \Delta t^2}{2}\right) \leq 1. \quad (1.24)$$

Simple algebra now gives the stability limit for Δt as

$$0 \leq \Delta t \leq \frac{2}{\omega}. \quad (1.25)$$

Values of Δt between these boundaries ensure that the numerical solution to Equation (1.14a,b)_a is stable and “well behaved” in that it exhibits the oscillatory nature of the analytical solution. However, it is worth emphasizing that the critical value $\Delta t_{\text{crit}} = 2/\omega$ represents a rather large proportion of the natural period of vibration $T = 2\pi/\omega$ of the system. In particular,

$$\Delta t_{\text{crit}} = \frac{2}{\omega} = \frac{2}{2\pi/T} = \frac{T}{\pi}. \quad (1.26)$$

Usually, in order for the numerical integration to follow accurately a sinusoidal function in time, Δt should be in the region $T/10$ to $T/20$. Hence the critical time-step limit for stability is well beyond the value that would be necessary to use to achieve reasonable accuracy. This would seem to indicate that stability is not generally a problem. Recall, however, that the analysis carried out above only refers to one-degree-of-freedom problems. Real situations have multiple degrees of freedom and consequently assessing an adequate time step is much more complicated.

For example, in the two-degree-of freedom-problem described in Section 1.3, for high values of the spring stiffness k the motion will resemble the oscillation of a mass pendulum with a much larger period of vibration than that given a spring-mass system where the stiffness k is such that the spring can extend. In fact, even in this simple example that ratio between the period of the (high k) pendulum oscillation and the spring-mass system can be several orders of magnitude. Unfortunately the stability restriction will be constrained by the smaller of the two periods as

$$\Delta t_{\text{crit}} \leq \min \left\{ \frac{T_{\text{spring}}}{\pi}, \frac{T_{\text{pendulum}}}{\pi} \right\} = \frac{T_{\text{spring}}}{\pi}. \quad (1.27)$$

For a stiff spring with constant k this value can be many orders of magnitude smaller than the time step required to follow accurately the pendulum motion, which would be of the order of $T_{\text{pendulum}}/20$. Consequently, stability imposes very small time steps in relation to the actual motion being integrated. Problems of this nature, where the maximum and minimum natural vibration periods (or frequencies) are many orders of magnitude apart, are known as “stiff” systems (in the sense that the spring is very stiff in relation to the pendulum). However, this is not always the case; for instance, in the two-degree-of-freedom problem above, if the spring constant k is very low so that the ratio between the natural period of the spring and pendulum is small then the system would not be stiff. Using time-stepping schemes with stability restrictions of the type described in Equation (1.25) can result in the need to carry out a very large number of time steps in order to advance the solution over the time span required. It is therefore essential that each time step can be evaluated as quickly and efficiently as possible, ideally without the solution of a linear or nonlinear system of equations.

Given that the time-step constraints for stiff problems can be computationally onerous, the following section will present an example of an “implicit” time-stepping scheme that avoids such restrictions but requires the solution of a nonlinear system of equations at each time step.

1.3.4 Mid-Point Rule

In anticipation of the use of a Newton–Raphson solution technique, the equilibrium Equation (1.13a,b)_b is recast in terms of a residual force $\mathbf{R}(\mathbf{x}, \mathbf{a})$ as

$$\mathbf{R}(\mathbf{x}, \mathbf{a}) = \mathbf{M}(\mathbf{a}) + \mathbf{T}(\mathbf{x}) - \mathbf{F} = \mathbf{0}. \quad (1.28)$$

The mid-point rule scheme is an alternative time integration technique which, despite appearing similar to the leap-frog scheme, emerges as an implicit scheme. It is defined by the approximate integration of velocities and positions at time step $n + 1$ as

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \Delta t \mathbf{a}_{n+1/2}; \quad \mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t \mathbf{v}_{n+1/2}, \quad (1.29a,b)$$

where $\mathbf{a}_{n+1/2}$ is given from the equilibrium Equation (1.13a,b)_b or (1.28) by

$$\mathbf{M}\mathbf{a}_{n+1/2} = \mathbf{F} - \mathbf{T}(\mathbf{x}_{n+1/2}), \quad (1.30a,b)$$

and, by definition, $\mathbf{x}_{n+1/2}$ and $\mathbf{v}_{n+1/2}$ are

$$\mathbf{x}_{n+1/2} = \frac{1}{2}(\mathbf{x}_n + \mathbf{x}_{n+1}); \quad \mathbf{v}_{n+1/2} = \frac{1}{2}(\mathbf{v}_n + \mathbf{v}_{n+1}). \quad (1.31a,b)$$

Inspection of Equation (1.30a,b) reveals that $\mathbf{a}_{n+1/2}$ cannot be resolved explicitly since $\mathbf{x}_{n+1/2}$, which is a function of $\mathbf{x}_{n+1}$, is unknown. The terms $\mathbf{a}_{n+1/2}$ and $\mathbf{x}_{n+1/2}$ are linked by the time-stepping Equations (1.29a,b) and (1.31a,b), leading to a system of equations that have to be solved implicitly. There are a number of ways to approach this solution, but the most popular process splits the time step into a *predictor* phase followed by a *corrector* phase. The predictor phase is based on making a reasonable prediction of one of the two unknown variables, for instance $\mathbf{a}_{n+1/2}^{(0)} = \mathbf{a}_{n-1/2}$. This gives predictions for $\mathbf{x}$ and $\mathbf{v}$ as

$$\mathbf{v}_{n+1}^{(0)} = \mathbf{v}_n + \Delta t \mathbf{a}_{n+1/2}^{(0)}; \quad \mathbf{v}_{n+1/2}^{(0)} = \frac{1}{2}(\mathbf{v}_n + \mathbf{v}_{n+1}^{(0)}), \quad (1.32a,b)$$

$$\mathbf{x}_{n+1}^{(0)} = \mathbf{x}_n + \Delta t \mathbf{v}_{n+1/2}^{(0)}; \quad \mathbf{x}_{n+1/2}^{(0)} = \frac{1}{2}(\mathbf{x}_n + \mathbf{x}_{n+1}^{(0)}). \quad (1.32c,d)$$

A preliminary residual force $\mathbf{R}_{n+1/2}^{(0)}$ as a function of the predicted values $\mathbf{a}_{n+1/2}^{(0)}$ and $\mathbf{x}_{n+1/2}^{(0)}$ can be calculated from Equation (1.28) as

$$\mathbf{R}(\mathbf{x}_{n+1/2}^{(0)}, \mathbf{a}_{n+1/2}^{(0)}) = \mathbf{M}\mathbf{a}_{n+1/2}^{(0)} + \mathbf{T}(\mathbf{x}_{n+1/2}^{(0)}) - \mathbf{F}. \quad (1.33)$$

The corrector phase establishes a Newton–Raphson process to drive the above residual to zero by incrementing the position and acceleration at iteration step $(k-1)$ by $\Delta\mathbf{x}_{n+1/2}$ and $\Delta\mathbf{a}_{n+1/2}$ to give

$$\mathbf{x}_{n+1/2}^{(k)} = \mathbf{x}_{n+1/2}^{(k-1)} + \Delta\mathbf{x}_{n+1/2}; \quad \mathbf{a}_{n+1/2}^{(k)} = \mathbf{a}_{n+1/2}^{(k-1)} + \Delta\mathbf{a}_{n+1/2}. \quad (1.34a,b)$$

The corrector phase employs the iterative Newton–Raphson method which requires the linearization of Equation (1.28) at iteration step $(k-1)$ and time interval $n+1/2$ to give

$$\begin{aligned} \mathbf{R}(\mathbf{x}_{n+1/2}^{(k)}, \mathbf{a}_{n+1/2}^{(k)}) &\approx \mathbf{R}(\mathbf{x}_{n+1/2}^{(k-1)}, \mathbf{a}_{n+1/2}^{(k-1)}) + D(\mathbf{M}\mathbf{a}_{n+1/2}^{(k-1)})[\Delta\mathbf{a}_{n+1/2}] \\ &\quad + D\mathbf{T}(\mathbf{x}_{n+1/2}^{(k-1)})[\Delta\mathbf{x}_{n+1/2}] = \mathbf{0}. \end{aligned} \quad (1.35)$$

Since the mass matrix is constant the first directional derivative is simply $\mathbf{M}\Delta\mathbf{a}_{n+1/2}$. However, the second directional derivative involving the internal force $\mathbf{T}(\mathbf{x})$ is a function of the length l and the orientation given by the unit vector $\mathbf{n}$, both being a function of the position $\mathbf{x}$ at any iteration (k) time step n . This is the source of the geometrically nonlinear nature of the problem. Consideration of

this second directional derivative requires the increments in $\mathbf{a}_{n+1/2}$ and $\mathbf{x}_{n+1/2}$ to be linked through the mid-point rule given by Equations (1.29a,b) and (1.31a,b) to give

$$\mathbf{x}_{n+1/2} = \mathbf{x}_n + \frac{\Delta t}{2} \mathbf{v}_n + \frac{\Delta t^2}{4} \mathbf{a}_{n+1/2}; \quad (1.36)$$

consequently, the increment $\Delta \mathbf{x}_{n+1/2}$ is given by the directional derivative

$$\Delta \mathbf{x}_{n+1/2} = D(\mathbf{x}_{n+1/2})[\Delta \mathbf{a}_{n+1/2}] = \frac{\Delta t^2}{4} \Delta \mathbf{a}_{n+1/2}. \quad (1.37)$$

Equation (1.35) is now rewritten in terms of the mass matrix $\mathbf{M}$ and a tangent stiffness matrix $\mathbf{K}$ as

$$\begin{aligned} \mathbf{R}(\mathbf{x}_{n+1/2}^{(k)}, \mathbf{a}_{n+1/2}^{(k)}) &\approx \mathbf{R}(\mathbf{x}_{n+1/2}^{(k-1)}, \mathbf{a}_{n+1/2}^{(k-1)}) + \mathbf{M} \Delta \mathbf{a}_{n+1/2} \\ &+ \mathbf{K}(\mathbf{x}_{n+1/2}^{(k-1)}) \Delta \mathbf{x}_{n+1/2} = \mathbf{0}, \end{aligned} \quad (1.38a)$$

where the tangent stiffness matrix $\mathbf{K}$, which is considered in Section 1.3.5 below, is derived as

$$\mathbf{K}(\mathbf{x}_{n+1/2}^{(k-1)}) \Delta \mathbf{x}_{n+1/2} = D\mathbf{T}(\mathbf{x}_{n+1/2}^{(k-1)})[\Delta \mathbf{x}_{n+1/2}]. \quad (1.38b)$$

Substituting for $\Delta \mathbf{x}_{n+1/2}$ from Equation (1.37) into Equation (1.38b) enables the corrector phase of the mid-point algorithm to be established solely in terms of $\Delta \mathbf{a}_{n+1/2}$ to give

$$\left[\mathbf{M} + \frac{\Delta t^2}{4} \mathbf{K}(\mathbf{x}_{n+1/2}^{(k-1)}) \right] \Delta \mathbf{a}_{n+1/2} = -\mathbf{R}(\mathbf{x}_{n+1/2}^{(k-1)}, \mathbf{a}_{n+1/2}^{(k-1)}). \quad (1.39)$$

Solving the system of nonlinear Equations (1.39) enables the mid-point corrector phase to be completed to yield the position $\mathbf{x}_{n+1}$ and velocity $\mathbf{v}_{n+1}$ as

$$\mathbf{a}_{n+1/2}^{(k)} = \mathbf{a}_{n+1/2}^{(k-1)} + \Delta \mathbf{a}_{n+1/2}, \quad (1.40a)$$

$$\mathbf{x}_{n+1/2}^{(k)} = \mathbf{x}_{n+1/2}^{(k-1)} + \frac{\Delta t^2}{4} \Delta \mathbf{a}_{n+1/2}, \quad (1.40b)$$

$$\mathbf{x}_{n+1}^{(k)} = \mathbf{x}_{n+1}^{(k-1)} + \frac{\Delta t^2}{2} \Delta \mathbf{a}_{n+1/2}, \quad (1.40c)$$

$$\mathbf{v}_{n+1}^{(k)} = \mathbf{v}_{n+1}^{(k-1)} + \Delta t \Delta \mathbf{a}_{n+1/2}. \quad (1.40d)$$

The Newton–Raphson procedure involves repeated application of Equations (1.39) and (1.40) until the residual $\mathbf{R}(\mathbf{x}, \mathbf{a})$ is less than a given tolerance. Note that had the problem been geometrically linear then it would only be necessary

to apply these equations once. Once convergence is achieved the predictor phase, given by Equation (1.32a,b), can be initiated for the next time step.

EXAMPLE 1.2: Discrete conservation of energy

Example 1.1 demonstrated the principle of conservation of energy for the spring-mass system. It is possible to maintain the energy conservation principle despite the use of time discretization when using certain types of time integration such as the mid-point rule. However, this requires subtle modifications to the algorithm which are described below. To begin, using the mid-point rule the equilibrium Equation (1.13a,b) at time $t_{n+1/2}$ can be expressed as

$$m\mathbf{a}_{n+1/2} + \frac{k(l_{n+1/2} - L)}{l_{n+1/2}}\mathbf{x}_{n+1/2} = m\mathbf{g},$$

where

$$\mathbf{a}_{n+1/2} = \frac{\mathbf{v}_{n+1} - \mathbf{v}_n}{\Delta t}; \quad \mathbf{x}_{n+1/2} = \frac{1}{2}(\mathbf{x}_{n+1} + \mathbf{x}_n); \quad l_{n+1/2} = \|\mathbf{x}_{n+1/2}\|.$$

Multiplying the discrete “mid-point” equilibrium equation by $\mathbf{v}_{n+1/2} = \frac{1}{2}(\mathbf{v}_{n+1} + \mathbf{v}_n) = (\mathbf{x}_{n+1} - \mathbf{x}_n)/\Delta t$ gives

$$\begin{aligned} \frac{1}{2\Delta t}m(\mathbf{v}_{n+1} \cdot \mathbf{v}_{n+1} - \mathbf{v}_n \cdot \mathbf{v}_n) + k\frac{(l_{n+1/2} - L)}{l_{n+1/2}}\frac{1}{2\Delta t}(\mathbf{x}_{n+1} \cdot \mathbf{x}_{n+1} \\ - \mathbf{x}_n \cdot \mathbf{x}_n) + \frac{1}{\Delta t}mg(x_{2,n+1} - x_{2,n}) = 0. \end{aligned}$$

Multiplying the above equation by Δt , and after some simple algebra, gives

$$\begin{aligned} \left(\frac{1}{2}mv_{n+1}^2 - \frac{1}{2}mv_n^2\right) + \frac{1}{2}k\frac{(l_{n+1/2} - L)}{l_{n+1/2}}(l_{n+1} + l_n)(l_{n+1} - l_n) \\ + mg(x_{2,n+1} - x_{2,n}) = 0. \end{aligned}$$

The first and last terms in the preceding equation give the change in the kinetic and potential energies. In order to transform the center term into a change of elastic energy, it is necessary to evaluate $l_{n+1/2}$ as $(l_{n+1} + l_n)/2$ rather than $\|\mathbf{x}_{n+1/2}\|$. In this case the elastic term becomes

$$\begin{aligned} \frac{1}{2}k\frac{(l_{n+1/2} - L)}{l_{n+1/2}}(l_{n+1} + l_n)(l_{n+1} - l_n) \\ = \frac{1}{2}k\frac{((l_{n+1} + l_n)/2 - L)}{(l_{n+1} + l_n)/2}(l_{n+1} + l_n)(l_{n+1} - l_n) \end{aligned}$$

(continued)

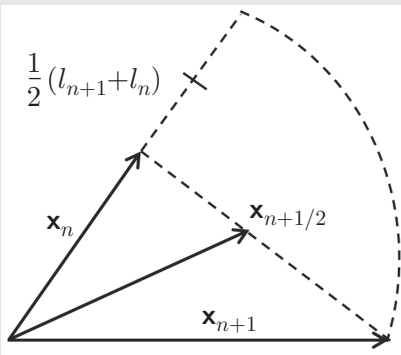
Example 1.2: (cont.)

$$\begin{aligned}
&= \frac{1}{2}k[(l_{n+1} - L) + (l_n - L)][(l_{n+1} - L) - (l_n - L)] \\
&= \frac{1}{2}k(l_{n+1} - L)^2 - \frac{1}{2}k(l_n - L)^2.
\end{aligned}$$

Hence the discrete equilibrium equation implies the conservation of total energy as

$$\begin{aligned}
&\frac{1}{2}mv_{n+1}^2 + \frac{1}{2}k(l_{n+1} - L)^2 + mgx_{2,n+1} \\
&= \frac{1}{2}mv_n^2 + \frac{1}{2}k(l_n - L)^2 + mgx_{2,n}.
\end{aligned}$$

Note that the evaluation of $l_{n+1/2}$ as the average between l_n and l_{n+1} will coincide with its evaluation as $\|\mathbf{x}_{n+1/2}\|$ when there is no rotation between $\mathbf{x}_n$ and $\mathbf{x}_{n+1}$. In more general cases, like the one shown in the figure below, this will not be exactly the case. For very small time steps, the differences, however, are minimal. Nevertheless, obtaining $l_{n+1/2}$ as the average ensures that energy is conserved and can have a beneficial effect in terms of the stability of the solution when large time steps are used. Since the average length is obtained under the assumption that the configurations at n and $n+1$ are aligned, this formulation is sometimes called “co-rotational” by some authors.

**1.3.5 Tangent Stiffness Matrix**

In order to determine the directional derivative in Equation (1.38b) it is necessary to find the linearization of a number of geometrical descriptors. For notational convenience and in recognition that the increment in mid-time-step acceleration given by Equation (1.37) represents a displacement, let $\Delta \mathbf{x}_{n+\frac{1}{2}} = \mathbf{u}$. Linearization is provided by the directional derivative for any position $\mathbf{x}$ as

$$D(\mathbf{x})[\mathbf{u}] = \left. \frac{d}{d\epsilon} \right|_{\epsilon=0} (\mathbf{x} + \epsilon \mathbf{u}) = \mathbf{u}; \quad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \quad (1.41)$$

A further example is the directional derivative of the length l^2 at time t , given as

$$\begin{aligned} D l^2(\mathbf{x})[\mathbf{u}] &= \left. \frac{d}{d\epsilon} \right|_{\epsilon=0} (\mathbf{x} + \epsilon \mathbf{u}) \cdot (\mathbf{x} + \epsilon \mathbf{u}) \\ &= \left. \frac{d}{d\epsilon} \right|_{\epsilon=0} \mathbf{x} \cdot \mathbf{x} + 2\epsilon \mathbf{u} \cdot \mathbf{x} + \epsilon^2 \cdot \mathbf{u} \\ &= 2\mathbf{u} \cdot \mathbf{x}. \end{aligned} \quad (1.42)$$

The above equation can now be used to find the directional derivative of l as follows:

$$D l^2(\mathbf{x})[\mathbf{u}] = 2l D l(\mathbf{x})[\mathbf{u}]. \quad (1.43)$$

Substituting Equation (1.42) into equation yields

$$D l(\mathbf{x})[\mathbf{u}] = \mathbf{n} \cdot \mathbf{u}. \quad (1.44)$$

The directional derivative $D(l^{-1})[\mathbf{u}]$ can now be found as

$$D l^{-1}(\mathbf{x})[\mathbf{u}] = -l^{-2} D l(\mathbf{x})[\mathbf{u}] = -l^{-2} \mathbf{n} \cdot \mathbf{u}. \quad (1.45)$$

Finally, the development of the tangent stiffness $\mathbf{K}$ requires the directional derivative of the unit normal $\mathbf{n}$ given in Equation (1.10c,d,e)_d. Observe that $\mathbf{n}$ is a function of l which itself is a function of $\mathbf{x}$. The directional derivative of $\mathbf{n}$ is found as

$$\begin{aligned} D \mathbf{n}(\mathbf{x})[\mathbf{u}] &= D \left(\frac{\mathbf{x}}{l} \right) [\mathbf{u}] \\ &= \mathbf{x} D l^{-1}(\mathbf{x})[\mathbf{u}] + \frac{1}{l} D \mathbf{x}[\mathbf{u}] \\ &= -\frac{1}{l^2} (\mathbf{n} \cdot \mathbf{u}) \mathbf{x} + \frac{\mathbf{u}}{l} \\ &= -\frac{1}{l} (\mathbf{n} \cdot \mathbf{u}) \mathbf{n} + \frac{\mathbf{u}}{l}. \end{aligned} \quad (1.46)$$

The tangent stiffness $\mathbf{K}(\mathbf{x})$ can now be found using Equations (1.12a,b)_a, (1.35), (1.44), and (1.46) as

$$\begin{aligned} D \mathbf{T}(\mathbf{x})[\mathbf{u}] &= D(T \mathbf{n}(\mathbf{x}))[\mathbf{u}] \\ &= k D(l - L)[\mathbf{u}] \mathbf{n} + T D(\mathbf{n}(\mathbf{x}))[\mathbf{u}] \\ &= k(\mathbf{n} \cdot \mathbf{u}) \mathbf{n} - T \frac{1}{l} (\mathbf{n} \cdot \mathbf{u}) \mathbf{n} + T \frac{\mathbf{u}}{l} \end{aligned}$$

$$\begin{aligned}
 &= \left[\left(k - \frac{T}{l} \right) (\mathbf{n} \otimes \mathbf{n})_{2 \times 2} + \frac{T}{l} \mathbf{I}_{2 \times 2} \right] \mathbf{u} \\
 &= \mathbf{K}(\mathbf{x}) \mathbf{u},
 \end{aligned} \tag{1.47}$$

where the tangent stiffness is

$$\mathbf{K}(\mathbf{x}) = \left[\left(k - \frac{T}{l} \right) (\mathbf{n} \otimes \mathbf{n})_{2 \times 2} + \frac{T}{l} \mathbf{I}_{2 \times 2} \right]. \tag{1.48}$$

1.3.6 Mid-Point Rule Examples

It is expected that the implicit mid-point time integration should be able to produce reliable results using a much larger time step than the leap-frog explicit scheme for the same accuracy. For $m = 10$, $\mathbf{x}^{(0)} = [5, 5]^T$, $c = 0$, and $k = 50\,000$ Figures 1.9 and 1.10 show that the mid-point scheme, with a time step $\Delta t = 0.1$, produces the same results as the leap-frog scheme, with a time step $\Delta t = 0.02$. In both cases the high spring stiffness results in a predictable circular motion.

Furthermore, a correct formulation of a Newton–Raphson iterative solution should result in quadratic convergence of the residual norm. For the two-degrees-of-freedom spring-pendulum system this is clearly demonstrated in Figure 1.11.

In Example 1.2 it was shown that, in order to conserve energy for the spring pendulum simulation, it was necessary to calculate the length of the spring as $l_{n+1/2} = (l_{n+1} + l_n)/2$ as opposed to the more obvious $\|\mathbf{x}_{n+1/2}\|$. Figure 1.12

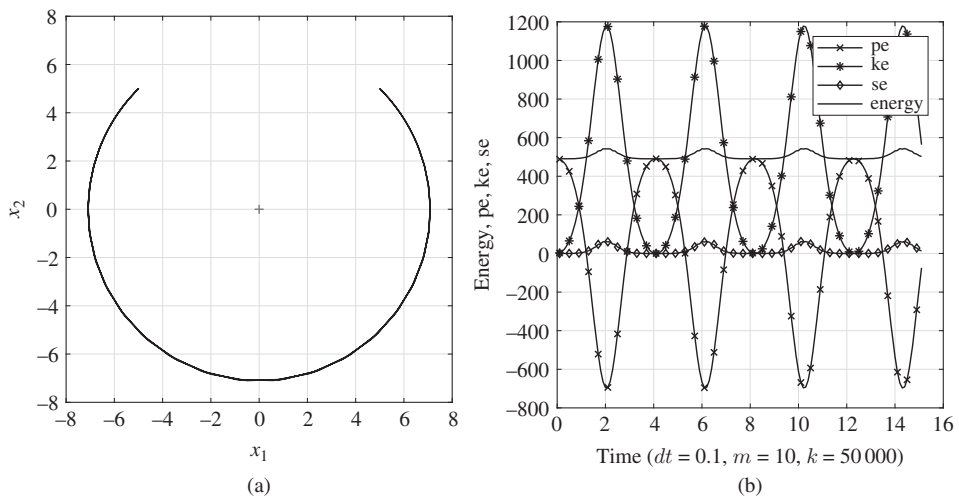


FIGURE 1.9 Spring-pendulum, mid-point, $\Delta t = 0.1$, $m = 10$, $k = 50\,000$: (a) Motion; (b) Energy–time.

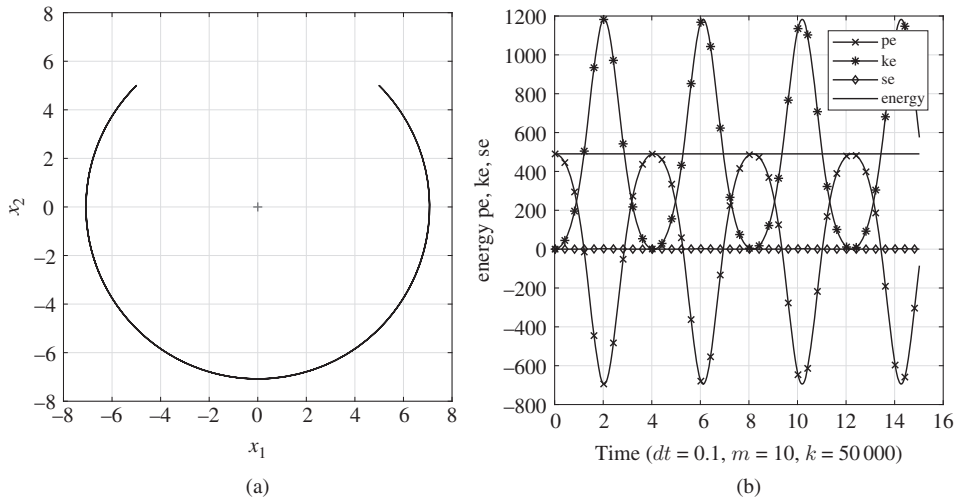


FIGURE 1.10 Spring-pendulum, leap-frog, $\Delta t = 0.02$, $m = 10$, $k = 50\,000$: (a) Motion; (b) Energy-time.

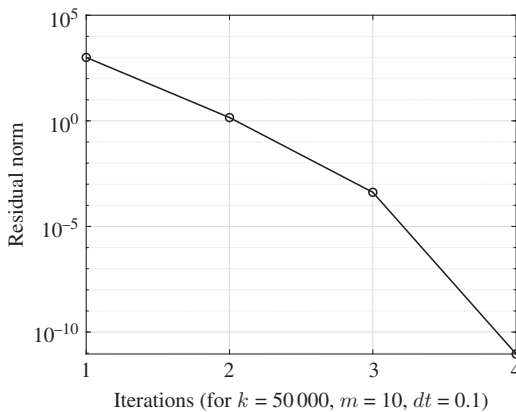


FIGURE 1.11 Mid-point integration: quadratic convergence for time $t = 15$.

shows the result of using the two alternatives, where it is clear that the average approach maintains the energy. It should be noted that, although the Newton–Raphson procedure converged, the algorithm used the tangent stiffness matrix given by Equation (1.48). However, this is not exactly correct since Equation (1.48) employed the directional derivative given by Equation (1.44), that is, $D(l)[u]$ and not $D((l_{n+1} + l_n)/2)[u]$.

Finally, Figure 1.13 shows a less predictable mid-point simulation with a time step of $\Delta t = 0.1$ and where the lower stiffness $k = 50$ results in a somewhat chaotic motion. The length calculation given in Example 1.2 is used, which makes the total energy remain constant. The mid-point algorithm is shown in Box 1.2.

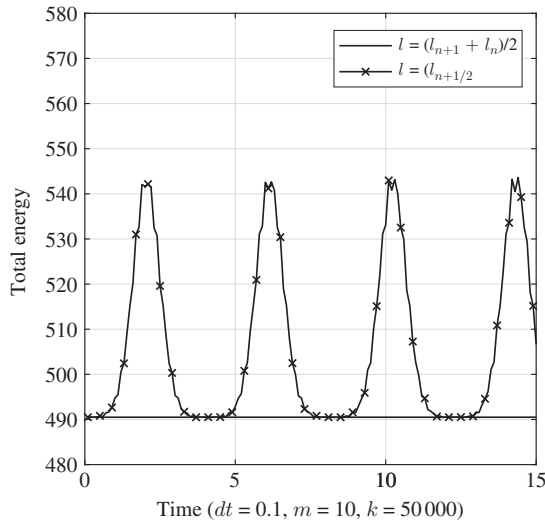


FIGURE 1.12 Spring-pendulum, mid-point, alternative length calculations: $\Delta t = 0.1$, $m = 10$, $k = 50\,000$.

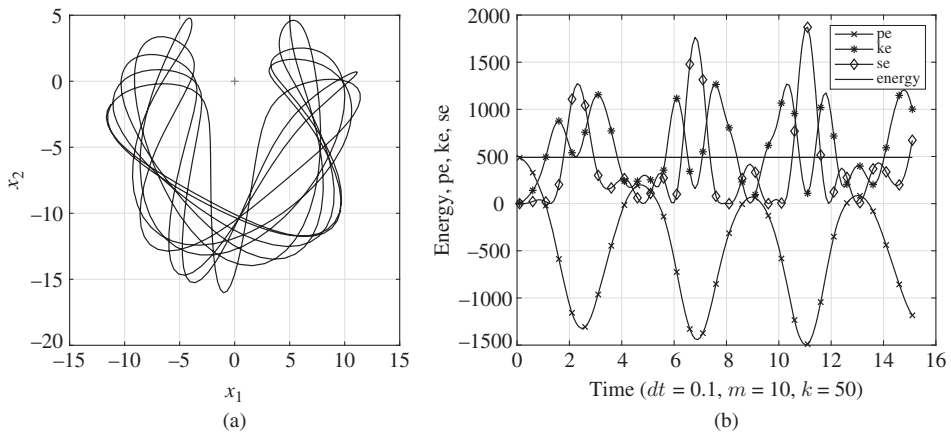


FIGURE 1.13 Spring-pendulum, mid-point (with energy conserving modified length formula $l_{n+1/2} = (l_n + l_{n+1})/2$, $\Delta t = 0.1$, $m = 10$, $k = 50$): (a) Motion ($t_{max} = 50$); (b) Energy-time ($t_{max} = 15$).

For the two-degrees-of-freedom spring-pendulum the leap-frog and mid-point rule MATLAB programs are presented at the end of the chapter in Boxes 1.3 and 1.4 respectively. Observe that in the leap-frog program energy calculation the velocity is adjusted as $v - a\Delta t/2$ to ensure that the displacement and velocity are synchronized to the same time step.

BOX 1.2: Mid-point algorithm

- INPUT geometry $\mathbf{x}$, initial velocity $\mathbf{v}$, material properties m and k , and solution parameters
- INITIALIZE $\mathbf{v}_{n+1/2} = \mathbf{v}$, $\mathbf{a}_{n+1/2} = \mathbf{0}$
- WHILE $t < t_{max}$ (time steps)
 - PREDICT $\mathbf{v}$, $\mathbf{v}_{n+1/2}$, $\mathbf{x}$, and $\mathbf{x}_{n+1/2}$ (1.32a–d)
 - DO WHILE ($\|\mathbf{R}\|/\|\mathbf{F}\| > \text{tolerance}$)
 - FIND $\mathbf{T}$ (1.12a,b)
 - FIND $\mathbf{R}$ (1.39)
 - SOLVE $(\mathbf{M} + \mathbf{K})\Delta\mathbf{a} = -\mathbf{R}$ (1.39)
 - CORRECT $\mathbf{a}_{n+1/2}^k$, $\mathbf{x}_{n+1/2}^k$, $\mathbf{x}_{n+1}^k$, $\mathbf{v}_{n+1}^k$ (1.40a–d)
 - SET $\mathbf{x}_{n+1}^k = \mathbf{x}_{n+1}^{k-1}$
 - (FIND kinetic, strain, and potential energies)
 - ENDDO
- ENDLOOP

Exercises

1. For the column problem given in Section 1.2.3, devise a numerical procedure to show that the static solution is $\theta \approx 167.42^\circ$.
2. For the constrained spring-mass system shown in Figure 1.7,
 - (a) show that the equations of motion are

$$ma_1 + k(x_1 - X_1) = -mg \cos \theta \sin \theta,$$

$$ma_2 + k(x_2 - X_2) = -mg \sin^2 \theta;$$

- (b) for the initial conditions $\mathbf{x} = \mathbf{X}$ and velocity $\mathbf{v} = \mathbf{0}$ show that the solution to the above equations, in terms of the angular frequency ω , is

$$x_1(t) = X_1 + u_1^{\text{static}}(1 - \cos(\omega t)),$$

$$x_2(t) = X_2 + u_2^{\text{static}}(1 - \cos(\omega t)),$$

where $\omega = (k/m)^{1/2}$ and the static displacements are

$$u_1^{\text{static}} = \frac{-mg \cos \theta \sin \theta}{k},$$

$$u_2^{\text{static}} = \frac{-mg \sin^2 \theta}{k}.$$

3. (a) Referring to Example 1.2, show that the tangent stiffness matrix for the case when the length is calculated at the average $l_{n+1/2} = (l_{n+1} + l_n)/2$ is

$$\bar{\mathbf{K}}(\mathbf{x}_{n+1/2}) = \left[\left(k - \frac{T}{l_{n+1/2}} \right) (\mathbf{n} \otimes \mathbf{m})_{2 \times 2} + \frac{T}{l_{n+1/2}} \mathbf{I}_{2 \times 2} \right],$$

where $T = k(l_{n+1/2} - L)$, $\mathbf{n} = \mathbf{x}_{n+1/2}/l_{n+1/2}$, and $\mathbf{m} = \mathbf{x}_{n+1}/l_{n+1}$.

- (b) Change the mid-point program to incorporate $l_{n+1/2} = (l_{n+1} + l_n)/2$ and $\bar{\mathbf{K}}(\mathbf{x})$, and confirm that the total energy is constant and that convergence is quadratic.

BOX 1.3: Leap-frog time integration

```
function SpringpendulumLeapFrog
%Leap Frog time integration
%-----
clear;clf;clc;cla;whitebg('white');
% mass coords
x=[5;5];L=norm(x);xprev=x;
% initial velocities and accelerations
v=[0;0];vprev=v;ahalf=[0;0];
% rod stiffness, mass,force
springK=50000;mass=10;g=[0;-9.81];
M=[mass,0;0,mass];F=M*g;
% control data
tmax=30;dt=0.02;t=0;count=0;
% start velocity
a=accel(x,L,F,M,springK);
v=v+a*dt/2;
%time loop
while t<tmax
    count=count+1;
% leap frog time integration
    x=x+v*dt;
    a=accel(x,L,F,M,springK);
    v=v+a*dt;
% update time, data for output
    t=t+dt;
    tt(count)=t;xx(count)=x(1);yy(count)=x(2);
% energy calculation
    l=norm(x);vel=norm(v-a*dt/2);
    ke(count)=0.5*mass*vel^2;
    se(count)=0.5*springK*(L-l)^2;
    pe(count)=mass*-x'*g;
    energy(count)=(ke(count)+se(count)+pe(count));
end
% Graphics
plot(0,0,'+r'); hold on
```

(continued)

Box 1.3: Leap-frog time integration (*cont.*)

```

    plot(xx,yy,'LineWidth',1,'Color','k');
    axis square;xlabel('x(1)');ylabel('x(2)');
    axis([-8, 8, -8, 8]);grid on
end
function a=accel(x,L,F,M,springK)
%geometry
l=norm(x);n=x/l;
Fint=springK*(l-L);
T=Fint*n;
%accelerations
a=M\ (F-T);
end

```

BOX 1.4: Mid-point time integration

```

% Mid-point time integration
%-----
clear;clf;clc;cla;whitebg('white');
% mass coords
x=[5;5];L=norm(x);xprev=x;
% initial velocities and accelerations
v=[0;0];vprev=v;ahalf=[0;0];
% rod stiffness, mass,force
springK=50;mass=10;g=[0;-9.81];
M=[mass,0;0,mass];F=M*g;
% control data
tmax=30;dt=0.1;t=0;count=0;
cnorm=1e-6;miter=50;
% time loop
while t<tmax
    t=t+dt;rnorm=10^6;
    count=count+1;niter=0;
% predictor
    v=vprev+dt*ahalf;vhalf=(vprev+v)/2;
    x=xprev+dt*vhalf;xhalf=(xprev+x)/2;
% corrector
    while ((rnorm>cnorm)&&(niter<miter))
        niter=niter+1;
% residual
% find length
        l=norm(xhalf);n=xhalf/l;
        Fint=springK*(l-L);T=Fint*n;
        resid=M*ahalf+T-F;rnorm=norm(resid);
% stiffness
        Ka=M+0.25*dt*dt*((springK-Fint/l)*(n*n')...
            + (Fint/l)*eye(2));
        da=-Ka\resid;
% update

```

(*continued*)

Box 1.4: Mid-point time integration (*cont.*)

```

    ahalf=ahalf+da;
    xhalf=xhalf+0.25*dt*dt*da;
    x=x+0.5*dt*dt*da;
    v=v+dt*da;
end
xprev=x;vprev=v;
tt(count)=t;xx(count)=x(1);yy(count)=x(2);
% energy calculation
l=norm(x);vel=norm(v);
ke(count)=0.5*mass*vel^2;
se(count)=0.5*springK*(L-l)^2;
pe(count)=mass*-x'*g;
energy(count)=(ke(count)+se(count)+pe(count));
end
% graphics
plot(0,0,'+r'); hold on
plot(xx,yy,'LineWidth',1,'Color','k');
axis square;xlabel('x(1)');ylabel('x(2)');
grid on

```