

Chapter 2

Linear Equations

Aims In this chapter we look at ways of solving systems of linear equations, sometimes known as *simultaneous* linear equations. We investigate

- Gaussian elimination and its variants, to obtain a solution to a system.
- partial pivoting *within* Gaussian elimination, to improve the accuracy of the results.
- iterative refinement *after* Gaussian elimination, to improve the accuracy of a first solution.

In some circumstances the use of Gaussian elimination is not recommended and other techniques based on iteration are more appropriate including

- Gauss–Seidel iteration.

We look at the method in detail and compare this iterative technique with direct solvers based on elimination.

Overview We begin by examining what is meant by a linear relationship, before considering how to handle the (linear) equations which represent such relationships. We describe methods for finding a solution, which simultaneously satisfies several such linear equations. In so doing we identify situations for which we can guarantee the existence of one, and only one, solution; situations for which more than one solution is possible; and situations for which there may be no solution at all.

As already indicated, the methods to be considered neatly fall into two categories; direct methods based on elimination; and iterative techniques. We look at each type of method in detail. For direct methods we need to bear in mind the limitations of computer arithmetic, and hence we consider how best to implement the methods so that as the algorithm proceeds the accumulation of error may be kept to a minimum. For iterative methods the concern is more related to ensuring that convergence to a specified accuracy is achieved in as few iterations as possible.

Acquired Skills After reading this chapter you will appreciate that solving linear equations is not as straightforward a matter as might appear. No matter which

method you use accumulated computer errors may make the true solution unattainable. However you will be equipped with techniques to identify the presence of significant errors and to take avoiding action. You will be aware of the differences between a variety of direct and indirect methods and the circumstances in which the use of a particular method is appropriate.

2.1 Introduction

One of the simplest ways in which a physical entity may be seen to behave is to vary in a linear manner with respect to some external influence. For example, the length (L) of the mercury column in a thermometer is accepted as being directly related to ambient temperature (T); an increase in temperature is shown by a corresponding increase in the length of the column. This linear relationship may be expressed as

$$L = kT + c.$$

Here k and c are constants whose values depend on the units in which temperature (degrees Celsius, Fahrenheit, etc.) and length (millimetres, inches, etc.) are measured, and on the bore of the tube. k represents the change in length relative to a rise in temperature, whilst c corresponds to the length of the column at zero degrees. We refer to T as the **independent variable** and L as the **dependent variable**. If we were to plot values of L against corresponding values of T we would have a straight line with slope k and intercept on the L axis equal to c .

A further example of a linear relationship is Hooke's Law (2.1) which relates the tension in a spring to the length by which it has been extended. If T is the tension, x is the extension of the spring, a is the unextended length and λ is the modulus of elasticity, then we have

$$T = \frac{\lambda}{a}x. \quad (2.1)$$

Typical units are Newtons (for T and λ) and metres (for x and a). A plot of T against x would reveal a straight line passing through the origin with slope λ/a .

An example which involves more than two variables is supplied by Kirchoff's Law which states that the algebraic sum of currents meeting at a point must be zero. Hence, if i_1 , i_2 and i_3 represent three such currents, we must have

$$i_1 + i_2 + i_3 = 0. \quad (2.2)$$

An increase in the value of one of the variables (say i_1) must result in a corresponding decrease (to the same combined value) in one or more of the other two variables (i_2 and/or i_3) and in this sense the relationship is linear. For a relationship to be linear, the variables (or constant multiples of the variables) are combined by additions and subtractions.

2.2 Linear Systems

In more complicated modelling situations it may be that there are many linear relationships, each involving a large number of variables. For example, in the stress analysis of a structure such as a bridge or an aeroplane, many hundreds (or indeed thousands) of linear relationships may be involved. Typically, we are interested in determining values for the dependent variables using these relationships so that we can answer the following questions:

- For a given extension of a spring, what is the tension?
- If one current has a specific value, what values for the remaining currents ensure that Kirchoff's Law is preserved?
- Is a bridge able to withstand a given level of traffic?

As an example of a linear system consider a sporting event where spectators were charged for admission at the rate of £15.50 for adults and £5.00 for concessions. It is known that the total takings for the event was £3312 and that a total of 234 spectators were admitted to the event. The problem is to determine how many adults watched the event.

If we let x represent the number of adults and y the number of concessions then we know that the sum of x and y must equal 234, the total number of spectators. Further, the sum of $15.5x$ and $5y$ must equal 3312, the total takings. Expressed in mathematical notation, we have

$$x + y = 234 \quad (2.3)$$

$$15.5x + 5y = 3312. \quad (2.4)$$

Sets of equations such as (2.3) and (2.4) are known as systems of linear equations or **linear systems**. It is an important feature of this system, and other systems we consider in this chapter, that the number of equations is equal to the number of unknowns to be found.

When the number of equations (and consequently the number of variables in those equations) becomes large it can be tedious to write the system out in full. In any case, we need some notation which will allow us to specify an arbitrary linear system in compact form so that we may conveniently examine ways of solving such a system. Hence we write linear systems using matrix and vector notation as

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (2.5)$$

where \mathbf{A} is a matrix, known as the **coefficient matrix** (or **matrix of coefficients**), \mathbf{b} is the vector of right-hand sides and \mathbf{x} is the vector of unknowns to be determined. Assuming n (the length of \mathbf{x}) unknowns to be determined, \mathbf{b} must be an n -column vector and \mathbf{A} an $m \times n$ square matrix, although for the time being we only consider systems with the same number of equations as unknowns, ($m = n$).

Problem

Write (2.3), (2.4) in matrix–vector form.

Solution

We have

$$\mathbf{Ax} = \mathbf{b}, \quad \text{where } \mathbf{A} = \begin{pmatrix} 1 & 1 \\ 15.5 & 5 \end{pmatrix}, \quad \mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 234 \\ 3312 \end{bmatrix}.$$

A great number of electronic, mechanical, economic, manufacturing and natural processes may be modelled using linear systems in which, as in the above simple example, the number of unknowns matches the number of equations. If, as sometimes happens, the modelling process results in nonlinear equations (equations involving perhaps squares, exponentials or products of the unknowns) it is often the case that the iterative techniques involved to solve these systems reduce to finding the solution of a linear system at each iteration. Hence linear equation solvers have a wider applicability than might at first appear to be the case.

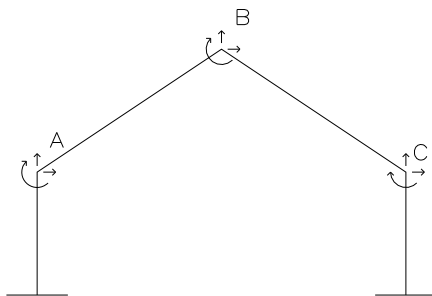
Since the modelling process either directly or indirectly yields a system of linear equations, the ability to solve such systems is fundamental. Our interest is in methods which can be implemented in a computer program and so we must bear in mind the limitations of computer arithmetic.

Problem

The structure shown in Fig. 2.1 shows a pitched portal frame. Using standard engineering analysis and assuming certain restrictions on the dimensions of the structure (the details of which need not concern us) it can be shown that the resulting displacements and rotations at the points A , B and C may be related to the acting forces and moments by a set of nine linear equations in nine unknowns. The equations involve constants reflecting the nature of the material of the structure.

Writing the displacements and rotations in the ascending sequence x_1, x_2, \dots, x_9 and assigning values to the constants and the external loads (the forces and the

Fig. 2.1 Pitched portal frame



moments) we might find that in a particular instance the equations take the following form:

$$x_1 = 0 \quad (2.6)$$

$$x_2 = 10 \quad (2.7)$$

$$2x_3 - 10x_5 + 3x_6 = 5 \quad (2.8)$$

$$3x_4 - 2x_5 + 2x_6 - 3x_7 = 2 \quad (2.9)$$

$$x_5 + 5x_6 + x_7 = 14 \quad (2.10)$$

$$3x_6 - 4x_7 = 17 \quad (2.11)$$

$$2x_7 = -4 \quad (2.12)$$

$$2x_8 = 10 \quad (2.13)$$

$$x_9 = 1. \quad (2.14)$$

Solution

Although this is quite a large system we can take advantage of its special structure to obtain a solution fairly easily. From (2.14) that $x_9 = 1$, from (2.13) that $x_8 = 5$, and from (2.12) that $x_7 = -2$.

To obtain the rest of the solution values we make use of the values that we already know. Substituting the value of x_7 into (2.11) we have $3x_6 + 8 = 17$ and so $x_6 = 3$. Now that we know x_6 and x_7 we can reduce (2.10) to an equation involving x_5 only, to give $x_5 + 15 - 2 = 14$, and so $x_5 = 1$. Similarly we can determine x_4 from (2.9) using $3x_4 - 2 + 6 + 6 = 2$, to give $x_4 = -2\frac{2}{3}$. Finally, to complete the solution we substitute the known values for x_5 and x_6 into (2.8) to obtain $2x_3 - 10 + 9 = 5$ and conclude that $x_3 = 3$. From (2.7) and (2.6) we have $x_2 = 10$ and $x_1 = 0$ to complete the solution.

To summarise, we have $x_1 = 0$, $x_2 = 10$, $x_3 = 3$, $x_4 = -2\frac{2}{3}$, $x_5 = 1$, $x_6 = 3$, $x_7 = -2$, $x_8 = 5$ and $x_9 = 1$.

Discussion

The system of equations we have just solved is an example of an **upper triangular system**. If we were to write the system in the matrix and vector form $\mathbf{Ax} = \mathbf{b}$, \mathbf{A} would be an **upper triangular matrix**, that is a matrix with zero entries below the diagonal. Mathematically the matrix \mathbf{A} is upper triangular if for all elements a_{ij}

$$a_{ij} = 0, \quad i > j.$$

For systems in which the coefficient matrix is upper triangular the last equation in the system is easily solved and this value can then be used to solve the penultimate equation. The results from these last two equations may be used to solve the previous equation, and so on. In this manner the whole system may be solved. The process of working backwards to complete the solution is known as **backward substitution**.

2.3 Gaussian Elimination

As part of the process we consider how general linear systems may be solved by reduction to upper triangular form.

Problem

Find the numbers of adults and concessions at the sporting event (2.3), (2.4) by solving the system of two equations in two unknowns.

Solution

In subtracting 15.5 times (2.3) from (2.4) we have $1.5y = 91.5$, and combining this with the original first equation we have

$$x + y = 234 \quad (2.15)$$

$$-10.5y = -315 \quad (2.16)$$

which is in upper triangular form. Solving (2.16) gives $y = 30$ (concessions) and substituting this value into (2.15) gives $x = 234 - 30 = 204$ (adults).

Discussion

The example illustrates an important mechanism for determining the solution to a system of equations in two unknowns, which may readily be extended to larger systems. It has two components:

1. Transform the system of equations into an equivalent one (that is, the solution to the transformed system is mathematically identical to the solution to the original system) in which the coefficient matrix is upper triangular.
2. Solve the transformed problem using backward substitution.

A linear system may be reduced to upper triangular form by means of successively subtracting multiples of the first equation from the second, third, fourth and so on, then repeating the process using multiples of the newly-formed second equation, and again with the newly-formed third equation, and so on until the penultimate equation has been used in this way. Having reduced the system to upper triangular form backward substitution is used to find the solution. This systematic approach, known as **Gaussian elimination**, is illustrated further in the following example.

Problem

Solve the following system of four equations in four unknowns

$$2x_1 + 3x_2 + 4x_3 - 2x_4 = 1 \quad (2.17)$$

$$x_1 - 2x_2 + 4x_3 - 3x_4 = 2 \quad (2.18)$$

$$4x_1 + 3x_2 - x_3 + x_4 = 2 \quad (2.19)$$

$$3x_1 - 4x_2 + 2x_3 - 2x_4 = 5. \quad (2.20)$$

Solution

As a first step towards an upper triangular system we eliminate x_1 from (2.18), (2.19) and (2.20). To do this we subtract $1/2$ times (2.17) from (2.18), 2 times (2.17) from (2.19), and $3/2$ times (2.17) from (2.20). In each case the fraction involved in the multiplication (the **multiplier**) is the ratio of two coefficients of the variable being eliminated from (2.18)–(2.20). The numerator is the coefficient of the variable to be eliminated and the denominator is the coefficient of that variable in the equation which is to remain unchanged. As a result of these computations the original system is transformed into

$$2x_1 + 3x_2 + 4x_3 - 2x_4 = 1 \quad (2.21)$$

$$- \frac{7}{2}x_2 + 2x_3 - 2x_4 = \frac{3}{2} \quad (2.22)$$

$$- 3x_2 - 9x_3 + 5x_4 = 0 \quad (2.23)$$

$$- \frac{17}{2}x_2 - 4x_3 + x_4 = \frac{7}{2}. \quad (2.24)$$

To proceed we ignore (2.21) and repeat the process on (2.22)–(2.24). We eliminate x_2 from (2.23) and (2.24) by subtracting suitable multiples of (2.22). We take $(-3)/(-\frac{7}{2})$ times (2.22) from (2.23) and $(-17)/(-\frac{7}{2})$ times (2.22) from (2.24) to give (after the removal of common denominators) the system

$$2x_1 + 3x_2 + 4x_3 - 2x_4 = 1 \quad (2.25)$$

$$- \frac{7}{2}x_2 + 2x_3 - 2x_4 = \frac{3}{2} \quad (2.26)$$

$$- \frac{75}{7}x_3 + \frac{47}{7}x_4 = -\frac{9}{7} \quad (2.27)$$

$$- \frac{62}{7}x_3 + \frac{41}{7}x_4 = -\frac{1}{7}. \quad (2.28)$$

Finally we subtract $62/75$ times (2.27) from (2.28) to give the system

$$2x_1 + 3x_2 + 4x_3 - 2x_4 = 1 \quad (2.29)$$

$$- \frac{7}{2}x_2 + 4x_3 - 2x_4 = \frac{3}{2} \quad (2.30)$$

$$- \frac{75}{7}x_3 + \frac{47}{7}x_4 = -\frac{9}{7} \quad (2.31)$$

$$\frac{161}{525}x_4 = -\frac{483}{525}. \quad (2.32)$$

We now have an upper triangular system which may be solved by backward substitution. From (2.32) we have $x_4 = 3$. Substituting this value in (2.31) gives $x_3 = 2$. Substituting the known values for x_4 and x_3 in (2.30) gives $x_2 = -1$. Finally using (2.29) we find $x_1 = 1$. The extension of Gaussian elimination to larger systems

is straightforward. The aim, as before, is to transform the original system to upper triangular form which is solved by backward substitution.

Discussion

The process we have described may be summarised in matrix form as pre-multiplying the coefficient matrix \mathbf{A} by a series of lower-triangular matrices to form an upper-triangular matrix.

We have

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{62}{75} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -3/(\frac{7}{2}) & 1 & 0 \\ 0 & -\frac{17}{7} & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 \\ -2 & 0 & 1 & 0 \\ -\frac{3}{2} & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 3 & 4 & -2 \\ 1 & -2 & 4 & -3 \\ 4 & 3 & -1 & 1 \\ 3 & -4 & 2 & -2 \end{pmatrix} \\ = \begin{pmatrix} 2 & 3 & 4 & -2 \\ & -\frac{7}{2} & 4 & -2 \\ & & -\frac{75}{7} & \frac{47}{7} \\ & & & \frac{161}{525} \end{pmatrix}$$

where it can be seen that sub-diagonal entries correspond to the multipliers in the elimination process. Since the product of lower triangular matrices is also lower triangular we may write the above as $\mathbf{L}\mathbf{A} = \mathbf{U}$ where \mathbf{L} is a lower triangular matrix and \mathbf{U} is upper triangular. Since the inverse of a lower triangular matrix is also lower triangular multiplying both sides by the inverse of \mathbf{L} produces

$$\mathbf{A} = \mathbf{L}\mathbf{U}$$

where \mathbf{L} is another lower triangular matrix and \mathbf{U} is upper triangular. This factorisation is generally known as the **LU decomposition**. If the factorisation is known it may be used repeatedly to solve the same set of equations for different right hand sides, as will be shown in Sect. 2.3.3.

2.3.1 Row Interchanges

In certain circumstances, Gaussian elimination fails as we see in the following example.

Problem

Solve the system of 4 equations in 4 unknowns

$$2x_1 - 6x_2 + 4x_3 - 2x_4 = 8 \quad (2.33)$$

$$x_1 - 3x_2 + 4x_3 + 3x_4 = 6 \quad (2.34)$$

$$4x_1 + 3x_2 - 2x_3 + 3x_4 = 3 \quad (2.35)$$

$$x_1 - 4x_2 + 3x_3 + 3x_4 = 9. \quad (2.36)$$

Solution

To eliminate x_1 from (2.34), (2.35) and (2.36) we take multiples $\frac{1}{2}$, 2 and $\frac{1}{2}$ of (2.33) and subtract from (2.34), (2.35) and (2.36) respectively. We now have the linear system

$$\begin{aligned} 2x_1 - 6x_2 + 4x_3 - 2x_4 &= 8 \\ 2x_3 + 4x_4 &= 2 \end{aligned} \quad (2.37)$$

$$15x_2 - 10x_3 + 7x_4 = -13 \quad (2.38)$$

$$-x_2 + x_3 + 4x_4 = 5. \quad (2.39)$$

Since x_2 does not appear in (2.37) we are unable to proceed as before. However a simple re-ordering of the equations enables x_2 to appear in the second equation of the system and this permits further progress. Exchanging (2.37) and (2.38) we have the system

$$\begin{aligned} 2x_1 - 6x_2 + 4x_3 - 2x_4 &= 8 \\ 15x_2 - 10x_3 + 7x_4 &= -13 \end{aligned} \quad (2.40)$$

$$\begin{aligned} 2x_3 + 4x_4 &= 2 \\ -x_2 + x_3 + 4x_4 &= 5. \end{aligned} \quad (2.41)$$

Resuming the elimination procedure we eliminate x_2 from (2.41) by taking a multiple $(-1)/15$ of (2.40) from (2.41) to give

$$\begin{aligned} 2x_1 - 6x_2 + 4x_3 - 2x_4 &= 8 \\ 15x_2 - 10x_3 + 7x_4 &= -13 \\ 2x_3 + 4x_4 &= 2 \end{aligned} \quad (2.42)$$

$$\frac{1}{3}x_3 + \frac{67}{15}x_4 = \frac{62}{15}. \quad (2.43)$$

Finally, to eliminate x_3 from (2.43) we take a multiple $1/6$ of (2.42) from (2.43). As a result we now have the linear system

$$\begin{aligned} 2x_1 - 6x_2 + 4x_3 - 2x_4 &= 8 \\ 15x_2 - 10x_3 + 7x_4 &= -13 \\ 2x_3 + 4x_4 &= 2 \\ \frac{57}{15}x_4 &= \frac{57}{15}, \end{aligned}$$

which is an upper triangular system that can be solved by backward substitution to give the solution $x_4 = 1$, $x_3 = -1$, $x_2 = -2$, and $x_1 = 1$.

Table 2.1 Gaussian elimination with row interchanges

Solve the $n \times n$ linear system $\mathbf{Ax} = \mathbf{b}$ by Gaussian elimination	
1	Reduce to upper triangular form: For $i = 1, 2, \dots, n - 1$
1.1	Ensure $a_{ii} \neq 0$. Exchange rows (including b_i) if necessary
1.2	Subtract multiples of row i (including b_i) from rows $i + 1, i + 2, \dots, n$ so that the coefficient of x_i is eliminated from those rows
2	Backward substitution: a_{ij} and b_i are current values of the elements of \mathbf{A} and \mathbf{b} as the elimination proceeds
2.1	$x_n = b_n/a_{nn}$
2.2	For $i = n - 1, n - 2, \dots, 1, x_i = (b_i - \sum_{j=i+1}^n a_{ij}x_j)/a_{ii}$

Discussion

The system (2.33)–(2.34) introduced the complication of a zero appearing as a leading coefficient (the **pivot**) in the equation (the **pivotal equation**) which would normally be used to eliminate the corresponding unknown from all subsequent equations. This problem was easily overcome by a re-arrangement which ensured a non-zero pivot. The method of Gaussian elimination with row interchanges is summarised in Table 2.1.

2.3.2 Partial Pivoting

In the previous section we saw that a re-arrangement of the equations is sometimes necessary if Gaussian elimination is to yield an upper triangular system. Here we show that re-arrangements of this form can be useful in other ways.

Up to this point we have used exact arithmetic to solve the equations, even though at times the fractions involved have been a little cumbersome. In real life of course we would use a computer to perform the arithmetic. However, a computer does not normally carry out arithmetic with complete precision. The difference between accurate and computer arithmetic is known as **rounding error**.

Unless specified otherwise Matlab uses arithmetic conforming to IEEE¹ standards by which numbers are represented by the formula $(-1)^s 2^{(e-1023)}(1 + f)$ using a 64-bit word. This is traditionally known as double precision arithmetic. The sign, s of the number is held in a 1-bit word, an eleven-bit binary number holds the exponent, e and a fifty-two bit binary number called the mantissa holds the precision, f of the number which can range from 1 to 2046.

We consider two problems, each of which is a system of two equations in two unknowns. They are, in fact, the same problem mathematically; all that we have

¹Institute of Electrical and Electronics Engineers

done is to reverse the order of the two equations. The solution can be seen to be $x_1 = 1$, $x_2 = 1$. To illustrate the effect of rounding error we consider the following problem.

Problem

Solve the following systems rounding the results of all arithmetic operations to three significant figures.

System (1):

$$0.124x_1 + 0.537x_2 = 0.661 \quad (2.44)$$

$$0.234x_1 + 0.996x_2 = 1.23 \quad (2.45)$$

System (2):

$$1.234x_1 + 0.996x_2 = 1.23 \quad (2.46)$$

$$0.124x_1 + 0.537x_2 = 0.661. \quad (2.47)$$

Solution

- System (1): In the usual manner we eliminate x_2 from (2.45) by subtracting the appropriate multiple of (2.44). In this case the multiple is $0.234/0.124$ which, rounded to three significant figures, is 1.89. Multiplying 0.537 by 1.89 gives 1.01 and subtracting this from 0.996 gives -0.0140 . Similarly for the right-hand side of the system, 0.661 multiplied by 1.89 is 1.25 and when subtracted from 1.23 gives -0.0200 . The system in its upper triangular form is

$$0.124x_1 + 0.537x_2 = 0.661 \quad (2.48)$$

$$-0.014x_2 = -0.020. \quad (2.49)$$

In backward substitution, x_2 is obtained from $\frac{0.020}{0.014}$ to give 1.43 which is used in (2.48) to give $x_1 = -0.863$. In this case the solution obtained by using arithmetic which retains three significant figures is nothing like the true solution.

- System (2): We eliminate x_2 from (2.47) by subtracting the appropriate multiple of (2.46). In this case the multiple is $0.124/0.234$, which is 0.530 to three significant figures. Multiplying 0.537 by 0.530 and subtracting from 0.996 gives 0.00900. Multiplying 0.661 by 0.530 and subtracting from 1.23 also gives 0.00900 and so the system in upper triangular form is

$$0.234x_1 + 0.996x_2 = 1.23$$

$$0.009x_2 = 0.009.$$

By backward substitution we have $x_2 = 1.00$, followed by $x_1 = 1.00$ which is the exact solution.

Discussion

We can infer from these two simple examples that the order in which equations are presented can have a significant bearing on the accuracy of the computed solution. The example is a little artificial, given that a modern computer carries out its arithmetic to many more significant figures than the three we have used here, often as many as sixteen or more. Nevertheless, for larger systems it may be verified that the order in which the equations are presented has a bearing on the accuracy of a computer solution obtained using Gaussian elimination. The next example describes a strategy for rearranging the equations within Gaussian elimination with a view to reducing the effect of rounding errors on a computed solution. We note in passing that an LU decomposition which takes account of partial pivoting of a non-singular matrix \mathbf{A} is available in the form $\mathbf{PA} = \mathbf{LU}$ where \mathbf{L} and \mathbf{U} are lower and upper triangular matrices and \mathbf{P} is a permutation matrix, a matrix of zeros and ones that in pre-multiplying \mathbf{A} performs the necessary row exchanges. For example exchanging rows 2 and 3 of a 3×3 matrix \mathbf{A} may be achieved by pre-multiplying \mathbf{A} by the permutation matrix $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$.

Problem

Solve the system (2.33)–(2.36) using row exchanges to minimise pivotal values at each step of the elimination.

Solution

We recall that in obtaining the previous solution example it was necessary to rearrange the equations during the reduction to upper triangular form to avoid a zero pivot. In this example we perform similar rearrangements, not just out of necessity, but also with accuracy in mind. Looking down the first column we see that the largest (in magnitude) coefficient of x_1 occurs in (2.35). We thus re-order the equations to make this the pivot. This involves exchanging (2.35) and (2.33). We now have the linear system

$$4x_1 + 3x_2 - 2x_3 + 3x_4 = 3 \quad (2.50)$$

$$x_1 - 3x_2 + 4x_3 + 3x_4 = 6 \quad (2.51)$$

$$2x_1 - 6x_2 + 4x_3 - 2x_4 = 8 \quad (2.52)$$

$$x_1 - 4x_2 + 3x_3 + 3x_4 = 9 \quad (2.53)$$

and we eliminate x_1 from (2.51), (2.52) and (2.53) by subtracting multiples $\frac{1}{4}$, $\frac{2}{4}$ and $\frac{1}{4}$ of (2.50) from (2.51), (2.52) and (2.53) respectively. Looking back to the system (2.33)–(2.36) we see that the multipliers without re-arrangement were $\frac{1}{2}$, 2 and $\frac{1}{2}$. Now they are uniformly smaller. This will prove to be significant. At the end of this, the first stage, of elimination, the original system is transformed to

$$\begin{aligned} 4x_1 - 3x_2 - 2x_3 + 3x_4 &= 3 \\ -3\frac{3}{4}x_2 + 4\frac{1}{2}x_3 + 2\frac{1}{4}x_4 &= 5\frac{1}{4} \end{aligned} \quad (2.54)$$

$$-7\frac{1}{2}x_2 + 5x_3 - 3\frac{1}{2}x_4 = 6\frac{1}{2} \quad (2.55)$$

$$-4\frac{3}{4}x_2 + 3\frac{1}{2}x_3 + 2\frac{1}{4}x_4 = 8\frac{1}{4}. \quad (2.56)$$

Considering (2.54)–(2.56) we see that the largest coefficient of x_2 occurs in (2.55). We exchange (2.55) and (2.54) to obtain

$$\begin{aligned} 4x_1 - 3x_2 - 2x_3 + 3x_4 &= 3 \\ -7\frac{1}{2}x_2 + 5x_3 - 3\frac{1}{2}x_4 &= 6\frac{1}{2} \end{aligned} \quad (2.57)$$

$$-3\frac{3}{4}x_2 + 4\frac{1}{2}x_3 + 2\frac{1}{4}x_4 = 5\frac{1}{4} \quad (2.58)$$

$$-4\frac{3}{4}x_2 + 3\frac{1}{2}x_3 + 2\frac{1}{4}x_4 = 8\frac{1}{4}. \quad (2.59)$$

To eliminate x_2 from (2.58) and (2.59) we take multiples $\frac{1}{2}$ and $\frac{19}{30}$ of (2.57) and subtract them from (2.58) and (2.59) respectively. As a result we have the linear system

$$\begin{aligned} 4x_1 - 3x_2 - 2x_3 + 3x_4 &= 3 \\ -7\frac{1}{2}x_2 + 5x_3 - 3\frac{1}{2}x_4 &= 6\frac{1}{2} \\ 2x_3 + 4x_4 &= 2 \\ \frac{1}{3}x_3 + 4\frac{7}{15}x_4 &= 4\frac{2}{15}. \end{aligned} \quad (2.60)$$

No further row interchanges are necessary since the coefficient of x_3 having the largest absolute value is already in its correct place. We eliminate x_3 from (2.60) using the multiplier $\frac{1}{6}$ and as a result we have the linear system

$$\begin{aligned} 4x_1 - 3x_2 - 2x_3 + 3x_4 &= 3 \\ -7\frac{1}{2}x_2 + 5x_3 - 3\frac{1}{2}x_4 &= 6\frac{1}{2} \\ 2x_3 + 4x_4 &= 2 \\ 3\frac{4}{5}x_4 &= 3\frac{4}{5}. \end{aligned}$$

Solving by backward substitution we obtain $x_4 = 1.0$, $x_3 = -1.0$, $x_2 = -2.0$, and $x_1 = 1.0$.

Discussion

The point of this example is to illustrate the technique known as **partial pivoting**. At each stage of Gaussian elimination the pivotal equation is chosen to maximise the absolute value of the pivot. Thus the multipliers in the subsequent subtraction process are reduced (a division by the pivot is involved) so that they are all at most one in magnitude. Any rounding errors present are less likely to be magnified as they permeate the rest of the calculation.

The 4×4 example of this section was solved on a computer using an implementation of Gaussian elimination and computer arithmetic based on a 32-bit word

Table 2.2 Gaussian elimination with partial pivoting

Solve the $n \times n$ linear system $\mathbf{Ax} = \mathbf{b}$ by Gaussian elimination with partial pivoting	
1	Reduce to upper triangular form: For $i = 1, 2, \dots, n - 1$
1.1	Find the j from $j = i, i + 1, \dots, n$ for which $ a_{ji} $ is a maximum
1.2	If $i \neq j$ exchange rows i and j (including b_i and b_j)
1.3	Subtract multiples of row i (including b_i) from rows $i + 1, i + 2, \dots, n$ so that the coefficient of x_i is eliminated from those rows
2	Backward substitution: a_{ij} and b_i are current values of the elements of \mathbf{A} and \mathbf{b} as the elimination proceeds
2.1	$x_n = b_n / a_{nn}$
2.2	For $i = n - 1, n - 2, \dots, 1$, $x_i = (b_i - \sum_{j=i+1}^n a_{ij}x_j) / a_{ii}$

(equivalent to retaining at least six significant figures). Without partial pivoting the solution obtained was $x_1 = 1.00000$, $x_2 = -2.00000$, $x_3 = -0.999999$, and $x_4 = 1.00000$, whereas elimination with pivoting gave the same solution except for $x_3 = -1.00000$. This may seem to be an insignificant improvement but the example illustrates how the method works. Even with 64-bit arithmetic it is confirmed by experience that partial pivoting particularly when used in solving larger systems of perhaps 10 or more equations is likely to be beneficial. As we will see, the form of the coefficient matrix can be crucial. For these reasons commercial implementations of Gaussian elimination always employ partial pivoting. The method is summarised in Table 2.2.

2.3.3 Multiple Right-Hand Sides

In a modelling process the right-hand side values often correspond to boundary values which may, for example, be the external loads on a structure or the resources necessary to fund various activities. As part of the process it is usual to experiment with different boundary conditions in order to achieve an optimal design. Gaussian elimination may be used to solve systems having several right-hand sides with little more effort than that involved in solving for just one set of right-hand side values.

Problem

Find three separate solutions of the system of (2.33)–(2.36) corresponding to three different right-hand sides.

Writing the system in the form $\mathbf{Ax} = \mathbf{b}$ we have three different vectors \mathbf{b} , namely

$$\begin{pmatrix} 8 \\ 6 \\ 3 \\ 9 \end{pmatrix}, \quad \begin{pmatrix} 2 \\ 3 \\ -1 \\ 4 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 \\ -6 \\ -3 \\ 9 \end{pmatrix}.$$

The three problems may be expressed in the following compact form

$$\begin{aligned} 2x_1 - 6x_2 + 4x_3 - 2x_4 &= 8, & 2, & 1 \\ x_1 - 3x_2 + 4x_3 + 3x_4 &= 6, & 3, & -6 \\ 4x_1 + 3x_2 - 2x_3 + 3x_4 &= 3, & -1, & -3 \\ x_1 - 4x_2 + 3x_3 + 3x_4 &= 9, & 4, & 9. \end{aligned}$$

Solution

We can solve all three equations at once by maintaining not one, but three columns on the right-hand side. The interchanges required by partial pivoting and the operations on the coefficient matrix do not depend on the right-hand side values and so need not be repeated unnecessarily. After the first exchange we have

$$\begin{aligned} 4x_1 + 3x_2 - 2x_3 + 3x_4 &= 3, & -1, & -3 \\ x_1 - 3x_2 + 4x_3 + 3x_4 &= 6, & 3, & -6 \\ 2x_1 - 6x_2 + 4x_3 - 2x_4 &= 8, & 2, & 1 \\ x_1 - 4x_2 + 3x_3 + 3x_4 &= 9, & 4, & 9 \end{aligned}$$

followed by

$$\begin{aligned} 4x_1 - 3x_2 - 2x_3 + 3x_4 &= 3, & -1, & -3 \\ -3\frac{3}{4}x_2 + 4\frac{1}{2}x_3 + 2\frac{1}{4}x_4 &= 5\frac{1}{4}, & 3\frac{1}{4}, & -5\frac{1}{4} \\ -7\frac{1}{2}x_2 + 5x_3 - 3\frac{1}{2}x_4 &= 6\frac{1}{2}, & 2\frac{1}{2}, & 2\frac{1}{2} \\ -4\frac{3}{4}x_2 + 3\frac{1}{2}x_3 + 2\frac{1}{4}x_4 &= 8\frac{1}{4}, & 4\frac{1}{4}, & 9\frac{3}{4}. \end{aligned}$$

Continuing the elimination we obtain

$$\begin{aligned} 4x_1 - 3x_2 - 2x_3 + 3x_4 &= 3, & -1, & -3 \\ -7\frac{1}{2}x_2 + 5x_3 - 3\frac{1}{2}x_4 &= 6\frac{1}{2}, & 2\frac{1}{2}, & 2\frac{1}{2} \\ 2x_3 + 4x_4 &= 2, & 2, & -6\frac{1}{2} \\ 3\frac{4}{5}x_4 &= 3\frac{4}{5}, & 2\frac{1}{3}, & 9\frac{1}{4}. \end{aligned}$$

Backward substitution is applied to each of the right-hand sides in turn to yield the required solutions, which are $(1, -2, -1, 1)^T$, $(-0.2456, -0.7719, -0.2281, 0.6190)^T$ and $(-1.4737, -6.8816, -8.1184, 2.4342)^T$ (to 4 decimal places).

Discussion

In solving a system of equations with several right-hand sides the reduction to upper triangular form has been performed once only. This is significant since the computation involved in the elimination process for larger systems is significantly greater than the computation involved in backward substitution and hence dominates the overall effort of obtaining a solution. If the system $\mathbf{Ax} = \mathbf{b}$ is to be repeatedly solved for the same \mathbf{A} but for different \mathbf{b} it makes sense to save and re-use the LU decomposition of \mathbf{A} .

2.4 Singular Systems

Not all systems of linear equations have a unique solution. In the case of two equations in two unknowns we may have the situation whereby the equations may be represented as parallel lines in which case there is no solution. On the other hand the representative lines may be coincident in which case every point on the lines is a solution and so we have an infinity of solutions. These ideas extend to larger systems. If there is no unique solution then there is either no solution at all or else there is an infinity of solutions and we would expect Gaussian elimination to break down at some stage.

Problem

Solve the system

$$2x_1 - 6x_2 + 4x_3 - 2x_4 = 8 \quad (2.61)$$

$$x_1 - 3x_2 + 4x_3 + 3x_4 = 6 \quad (2.62)$$

$$x_1 - x_2 + 2x_3 - 5x_4 = 3 \quad (2.63)$$

$$x_1 - 4x_2 + 3x_3 + 3x_4 = 9. \quad (2.64)$$

Solution

Applying Gaussian elimination with partial pivoting to the system we first eliminate x_1 to obtain

$$2x_1 - 6x_2 + 4x_3 - 2x_4 = 8$$

$$2x_3 + 4x_4 = 2$$

$$2x_2 - 4x_4 = -1$$

$$-x_2 + x_3 + 4x_4 = 5.$$

Partial pivoting and elimination of x_2 gives

$$2x_1 - 6x_2 + 4x_3 - 2x_4 = 8$$

$$2x_2 - 4x_4 = -1$$

$$2x_3 + 4x_4 = 2$$

$$x_3 + 2x_4 = 4\frac{1}{2}.$$

Finally, elimination of x_3 gives

$$2x_1 - 6x_2 + 4x_3 - 2x_4 = 8$$

$$2x_2 - 4x_4 = -1$$

$$2x_3 + 4x_4 = 2$$

$$0x_4 = 3\frac{1}{2}.$$

Clearly we have reached a nonsensical situation which suggests that 0 is equal to $3\frac{1}{2}$. This contradiction suggests that there is an error or a wrong assumption in the modelling process which has provided the equations.

Changing the right-hand side of (2.61)–(2.64) to $(8, 6, 3, 5.5)^T$ and applying elimination gives

$$\begin{aligned} 2x_1 - 6x_2 + 4x_3 - 2x_4 &= 8 \\ 2x_2 - 4x_4 &= -1 \\ 2x_3 + 4x_4 &= 2 \\ 0x_4 &= 0 \end{aligned}$$

which is more sensible than before, if a little odd. There is no longer any contradiction. In effect we are free to choose *any* value we like for x_4 and to employ this value into the backward substitution process. In the first system we had no solution; now we have an infinity of solutions.

Discussion

It is possible to check that the rows (as vectors) of the coefficient matrix of the system (2.61)–(2.64) are linearly dependent. It follows that the columns are also linearly dependent and vice-versa. Such matrices are said to be **singular**. A system for which the coefficient matrix is singular (a **singular system**) has no unique solution. Further, if for a singular system the right-hand side as a column vector is a linear combination of the column vectors of the matrix of coefficients then there is an infinity of solutions, otherwise there are no solutions. In practice however, because of the limitations of computer arithmetic it may be difficult to determine if a system is genuinely singular or if it is just close to being singular. In any event if partial pivoting produces a zero or nearly zero pivot the validity of the model should be questioned.

2.5 Symmetric Positive Definite Systems

It is not uncommon for matrices which arise in the modelling of physical processes to be symmetric and positive definite. A matrix \mathbf{A} is symmetric and positive definite if $\mathbf{A} = \mathbf{A}^T$ and $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0$ for all non-zero vectors \mathbf{x} . As an example the symmetric stiffness matrix which arises in finite element analysis and which may be regarded as a generalisation of the modulus of elasticity from Hooke's Law (2.1) is positive definite. We refer to a linear system in which the coefficient matrix is symmetric and positive definite as a symmetric positive definite system.

Problem

Solve the 4×4 system:

$$\begin{aligned}
16x_1 + 3x_2 + 4x_3 + 2x_4 &= 25 \\
3x_1 + 12x_2 + 2x_3 - x_4 &= 16 \\
4x_1 + 2x_2 + 8x_3 - x_4 &= 13 \\
2x_1 - x_2 - x_3 + 2x_4 &= 2.
\end{aligned}$$

Solution

By Gaussian elimination we have

$$\begin{aligned}
16x_1 + 3x_2 + 4x_3 + 2x_4 &= 25 & (2.65) \\
11.4375x_2 + 1.25x_3 - 1.375x_4 &= 11.3125 \\
1.25x_2 + 7x_3 - 1.5x_4 &= 6.75 \\
-1.375x_2 - 1.5x_3 + 1.75x_4 &= -1.125
\end{aligned}$$

then

$$\begin{aligned}
16x_1 + x_2 + 4x_3 + 2x_4 &= 25 & (2.66) \\
11.4375x_2 + 1.25x_3 - 1.375x_4 &= 11.3125 \\
6.8634x_3 - 1.3497x_4 &= 5.5137 \\
-1.3497x_3 + 1.5847x_4 &= 0.2350
\end{aligned}$$

and finally

$$\begin{aligned}
16x_1 + 3x_2 + 4x_3 + 2x_4 &= 25 \\
11.4375x_2 + 1.25x_3 - 1.375x_4 &= 11.3125 \\
6.8634x_3 - 1.3497x_4 &= 5.5137 \\
1.3913x_4 &= 1.3913.
\end{aligned}$$

Backward substitution produces $x_4 = 1$, $x_3 = 1$, $x_2 = 1$, $x_1 = 1$.

Discussion

It should be noted that at each stage the size of the pivot cannot be increased by row interchanges and so partial pivoting is not required. This is a feature of symmetric positive definite systems. Furthermore it is possible to get by with fewer calculations. For example in the first reduction the numbers 1.25, -1.375 and -1.5 need only be calculated once, whilst in the second reduction the number -1.3497 need only be calculated once. In effect it is sufficient to compute only the entries in the upper triangular portion of the coefficient matrix as it is transformed. Elements in the lower triangle are deduced by symmetry arguments. Roughly speaking, we have halved the workload and halved the storage requirements. In a large system such economies could be significant.

Although it is easy to test the symmetry of a given matrix, it is not so easy to establish if a symmetric matrix is positive definite. A symmetric matrix is positive definite if it is diagonally dominant (that is if each diagonal element is larger in absolute value than the sum of the absolute values of the other elements on the same row). However being diagonally dominant is not a necessary condition for a symmetric matrix to be positive definite. Often the only real way to check is to apply Gaussian elimination. If, at any stage, partial pivoting is necessary then the matrix is not positive definite. In practice therefore when the modelling process requires the repeated solution of a symmetric linear system it is advisable to try the system on a program which assumes that the coefficient matrix is positive definite, but which flags an error if a non-positive definite condition is detected.

2.6 Iterative Refinement

The effect of rounding error in Gaussian elimination is to yield a computed solution which is a perturbation of the mathematically correct solution. Partial pivoting may help to reduce the accumulative effect of these errors, but there is no guarantee that it will eliminate them entirely. Iterative refinement is a process by which a first computed solution can sometimes be improved to yield a more accurate solution. In the following example we illustrate the method by working to limited accuracy and drawing conclusions that may be applied when using computer arithmetic.

Problem

Solve the 4×4 system

$$0.366x_1 + 0.668x_2 - 0.731x_3 - 0.878x_4 = -0.575 \quad (2.67)$$

$$0.520x_1 + 0.134x_2 - 0.330x_3 + 0.484x_4 = 0.808 \quad (2.68)$$

$$0.738x_1 - 0.826x_2 + 0.194x_3 - 0.204x_4 = -0.098 \quad (2.69)$$

$$0.382x_1 - 0.667x_2 + 0.270x_3 - 0.255x_4 = -0.270 \quad (2.70)$$

using Gaussian elimination.

Solution

The system has the exact solution $x_1 = 1$, $x_2 = 1$, $x_3 = 1$ and $x_4 = 1$. However, working to just three significant figures, Gaussian elimination with partial pivoting produces a solution $x_1 = 1.14$, $x_2 = 1.18$, $x_3 = 1.23$ and $x_4 = 0.993$. Substituting these values back into the left-hand side of the equations and working to three-figure accuracy produces -0.575 , 0.808 , -0.0908 and -0.270 , which suggests that this computed solution is accurate. However repeating the substitution and working to six significant figures produces a slightly different set of values. Subtracting the three-figure result from the corresponding six-figure result for each equation, gives

differences of -0.00950 for (2.67), -0.0176 for (2.68), -0.000688 for (2.69) and 0.00269 for (2.70). These differences are known as **residuals**. Since these residual values are small in relation to the coefficients in the equation we can suppose we are fairly close to the true solution.

More formally, if we were to assume that the true solution differs from our calculated 3-figure solution by amounts δ_1 , δ_2 , δ_3 and δ_4 , that is the true solution is $1.14 + \delta x_1$, $1.18 + \delta_2$, $1.23 + \delta_3$, $0.993 + \delta x_4$, we would have

$$\begin{aligned} 0.366(1.14 + \delta_1) + 0.668(1.18 + \delta_2) \\ - 0.731(1.23 + \delta_3) - 0.878(0.993 + \delta_4) &= -0.575 \\ 0.520(1.14 + \delta_1) + 0.134(1.18 + \delta_2) \\ - 0.330(1.23 + \delta_3) + 0.484(0.993 + \delta_4) &= 0.808 \\ 0.738(1.14 + \delta_1) - 0.826(1.18 + \delta_2) \\ + 0.194(1.23 + \delta_3) - 0.204(0.993 + \delta_4) &= -0.098 \\ 0.382(1.14 + \delta_1) - 0.667(1.18 + \delta_2) \\ + 0.270(1.23 + \delta_3) - 0.255(0.993 + \delta_4) &= -0.270. \end{aligned}$$

We are now in a position to find the corrections δ_1 , δ_2 , δ_3 and δ_4 . Using the calculations we have already performed to find the residuals we have

$$\begin{aligned} 0.366\delta_1 + 0.668\delta_2 - 0.731\delta_3 - 0.878\delta_4 &= -0.00950 \\ 0.520\delta_1 + 0.134\delta_2 - 0.330\delta_3 + 0.484\delta_4 &= -0.0176 \\ 0.738\delta_1 - 0.826\delta_2 + 0.194\delta_3 - 0.204\delta_4 &= -0.000688 \\ 0.382\delta_1 - 0.667\delta_2 + 0.270\delta_3 - 0.255\delta_4 &= 0.00269. \end{aligned}$$

Using 3 significant figure arithmetic again we find that Gaussian elimination with partial pivoting produces a solution $\delta_1 = -0.130$, $\delta_2 = -0.166$, $\delta_3 = -0.210$ and $\delta_4 = 0.00525$, and adding these values to our initial solution gives $x_1 = 1.01$, $x_2 = 1.01$, $x_3 = 1.02$ and $x_4 = 0.998$. Although this is still not completely accurate, it is slightly better than the previous solution.

Discussion

The example is an illustration of the technique known as **iterative refinement**. Computer calculations inevitably involve rounding errors which result from performing arithmetic to a limited accuracy. Iterative refinement is used to try to restore a little more accuracy to an existing solution. However it does depend upon being able to calculate the residuals with some accuracy and this will involve using arithmetic which is more accurate than that of the rest of the calculations. Note that the original coefficient matrix is used in finding the improved solution and this can lead to economies when applied to large systems. The method of iterative refinement as used with Gaussian elimination is summarised in Table 2.3.

Table 2.3 Gaussian elimination with iterative refinement

Solve the $n \times n$ linear system $\mathbf{Ax} = \mathbf{b}$ by Gaussian elimination with iterative refinement	
1	Set \mathbf{s} to $\mathbf{0}$
2	Solve $\mathbf{Ax} = \mathbf{b}$ using Gaussian elimination with partial pivoting and add the solution to \mathbf{s}
3	Compute the residual $\mathbf{r} = \mathbf{b} - \mathbf{As}$ using a higher accuracy than the rest of the calculation
4	If \mathbf{r} is acceptably small then stop, \mathbf{s} is the solution otherwise set $\mathbf{b} = \mathbf{r}$
5	Repeat from step 2 unless \mathbf{r} shows no signs of becoming acceptably small

Iterative refinement could be continued until the residuals stabilise at or very near to zero. In practice one step of iterative refinement usually suffices. If iterative refinement fails to stabilise it is likely that meaningful solutions cannot be obtained using conventional computing methods. Such systems will be discussed in the following section.

2.7 Ill-Conditioned Systems

Ill-conditioned systems are characterised by solutions which vary dramatically with small changes to the coefficients of the equations. Ill-conditioned systems pose severe problems and should be avoided at all costs. The presence of an ill-conditioned system may be detected by making slight changes to the coefficients and noticing if these produce disproportionate changes in the computed solution. An alternative method would be to apply iterative refinement as signs of non-convergence would indicate ill-conditioning. There can be no confidence in the computed solution of an ill-conditioned system, since the transfer to a computer may well introduce small distortions in the coefficients which, in addition to the limitations of computer arithmetic, is likely to produce yet further divergence from the original solution. The presence of an ill-conditioned system should be taken as a warning that the situation being modelled is either so inherently unpredictable or unstable as to defy analysis, or that the modelling process itself is at fault, perhaps through experimental or observational error not supplying sufficiently accurate information. Examples of ill-conditioned system are given in Exercises 7 and 8.

2.8 Gauss–Seidel Iteration

Systems of equations which display a regular pattern often occur in modelling physical processes for example in solving steady state temperature distributions. Linear approximations to partial derivatives over relatively small intervals are combined so that they model the differential equations over a whole surface. The resulting equations not only display a regular pattern they are also sparse in the sense that

although the number of equations may be large, possibly hundreds or thousands, the number of variables in each equation is very much smaller, reflecting the way in which they have been assembled over small areas. It is perfectly possible to solve such systems using the elimination techniques considered earlier in this chapter, but that may cause storage and indexing problems. It would be possible to store a sparse coefficient matrix in a condensed form by just recording and indexing non-zero elements but the sparse quality would probably be quickly destroyed by Gaussian elimination. We consider an alternative, iterative approach to solving large sparse systems.

Problem

Solve the following system of equations using Gauss–Seidel iteration.

$$\begin{aligned}
 6x_1 + x_2 &= 1 \\
 x_1 + 6x_2 + x_3 &= 1 \\
 x_2 + 6x_3 + x_4 &= 1 \\
 &\vdots \\
 x_8 + 6x_9 + x_{10} &= 1 \\
 x_9 + 6x_{10} &= 1.
 \end{aligned}$$

Solution

We adopt an iterative approach to obtaining a solution. We start with an initial guess at the solution and then form a series of further approximations which we hope eventually leads to a solution. We start by writing the equations as

$$x_1 = \frac{1 - x_2}{6} \quad (2.71)$$

$$x_2 = \frac{1 - x_1 - x_3}{6} \quad (2.72)$$

$$\vdots \quad (2.73)$$

$$x_9 = \frac{1 - x_7 - x_8}{6} \quad (2.74)$$

$$x_{10} = \frac{1 - x_9}{6}. \quad (2.75)$$

As an initial guess at the solution we set each of x_1, x_2, \dots, x_{10} to $\frac{1}{6}$, not an unreasonable choice since this is the solution if we ignore the off-diagonal terms in the original coefficient matrix.

We have from (2.71)

$$x_1 = \frac{1 - \frac{1}{6}}{6} = 0.1389$$

Table 2.4 Gauss–Seidel iteration, results

Step	x_1	x_2	x_3	x_4	x_5	x_6	...	x_{10}
1	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667	...	0.1667
2	0.1389	0.1157	0.1196	0.1190	0.1191	0.1190	...	0.1468
3	0.1474	0.1222	0.1265	0.1257	0.1259	0.1258	...	0.1465
4	0.1463	0.1212	0.1255	0.1248	0.1249	0.1249	...	0.1464
5	0.1465	0.1213	0.1256	0.1249	0.1250	0.1250	...	0.1464
6	0.1464	0.1213	0.1256	0.1249	0.1250	0.1250	...	0.1464
7	0.1464	0.1213	0.1256	0.1249	0.1250	0.1250	...	0.1464

Table 2.5 Gauss–Seidel iteration

Solve the $n \times n$ sparse linear system $\mathbf{Ax} = \mathbf{b}$ by Gauss–Seidel iteration	
1	Set \mathbf{x} to an estimate of the solution
2	For $i = 1, 2, \dots, n$ evaluate $x_i = (b_i - \sum_{j=1}^n a_{ij}x_j)/a_{ii}$ using the most recently found values of x_j
3	Repeat from step 2 until the values x_i settle down. If this does not happen abandon the scheme

and so from (2.72)

$$x_2 = \frac{1 - 0.1389 - \frac{1}{6}}{6} = 0.1157$$

and we find values for x_3, x_4, \dots, x_{10} in a similar manner to give the values shown in the first line of Table 2.4. Using these as new values for the right-hand sides, in the second iteration we obtain

$$x_1 = \frac{1 - 0.1157}{6} = 0.1474 \quad (2.76)$$

and so from (2.72)

$$x_2 = \frac{1 - 0.1474 - 0.1196}{6} = 0.1222 \quad (2.77)$$

and we find values for x_3, x_4, \dots, x_{10} in a similar manner to give the values shown in the second line of Table 2.4. After further iterations the values x_1, x_2, \dots, x_{10} converge to 0.1464, 0.1213, 0.1256, 0.1249, 0.1250, 0.1250, 0.1249, 0.1256, 0.1213 and 0.1464 respectively. A calculation of residuals would confirm that we have a solution to the equations. The process is known as **Gauss–Seidel iteration**. The method is summarised in Table 2.5 in which it is assumed coefficients are stored in a conventional matrix, although in practice coefficients and variables would not necessarily be stored in full matrix or vector form.

Discussion

Unfortunately it can be readily demonstrated that Gauss–Seidel iteration does not always converge (see Exercise 9). For systems in which the coefficient matrix is diagonally dominant (page 35), then Gauss–Seidel iteration is guaranteed to converge. This is not to say that non-diagonal systems cannot be solved in this way, it is just that convergence cannot be guaranteed. For this reason the method would in general only be applied to large systems, which might otherwise cause the storage problems already mentioned.

Summary In this chapter we examined what is meant by a linear relationship and how such a relationship is represented by a linear equation. We looked at Gaussian elimination for solving systems of such equations and noted that

- Gaussian elimination is essentially a reduction to upper triangular form followed by backward substitution.
- re-arranging the order in which the equations are presented can affect the accuracy of the computed solution, led us to the strategy known as partial pivoting.

We also considered special types of linear systems, namely

- singular systems, for which there is no unique solution and which may be identified by a failure in Gaussian elimination even when partial pivoting is included.
- symmetric positive definite systems, for which we mentioned that more efficient methods may be used.
- ill-conditioned systems, that are likely to cause problems no matter what computational methods are used.
- sparse systems for which the Gauss–Seidel iterative method may be used if computer time and storage is at a premium.

Mindful of the errors inherent in computer arithmetic we looked at ways of improving an existing solution using

- iterative refinement.

although we pointed out that in the case of ill-conditioned systems it might make matters worse.

Exercises

1. A baker has 2.5 kilos of flour, 3.5 kilos of sugar and 5.3 kilos of fruit with which to make fruit pies. The baker has utensils to make pies in three sizes—small, medium and large. Small pies require 30 grams of flour, 20 grams of sugar and 40 grams of fruit. Similarly the requirement for medium size pies is 40, 30 and 60

grams respectively, and for large pies 60, 50 and 90 grams. The baker wishes to use up all the ingredients. Formulate a linear system to decide if this is possible.

Let the number of small, medium and large pies to be made be x , y and z . Taking into account that 1 kilo = 1000 grams we have

$$3x + 2y + 4z = 250 \quad (\text{flour to be used})$$

$$4x + 3y + 6z = 350 \quad (\text{sugar to be used})$$

$$6x + 5y + 9z = 530 \quad (\text{fruit to be used}).$$

If the solution to these equations turns out to consist of whole numbers then the baker can use up all the ingredients. If some or all of x , y and z were not whole numbers then the baker is not able to use up all the ingredients, deciding how many pies of each size to make in order to minimise the quantity of ingredients left over is the type of problem we look at in Chap. 7.

Assuming the equations are written in the form $\mathbf{Ax} = \mathbf{b}$ use the following code to find a solution. The code uses the left-division operator, `\` as described in Sect. 1.4.

```
A = [ 3  2  4; 4  3  6; 6  5  9 ]
b = [ 250 350 530 ]'
x = A\b
```

2. (i) Reduce the following linear system to upper triangular form.

$$2x_1 - x_2 - 4x_3 = 5$$

$$x_1 + x_2 + 2x_3 = 0$$

$$6x_1 + 3x_2 - x_3 = -2.5.$$

Form the matrix of coefficients of the upper triangular system in the matrix \mathbf{U} and the corresponding right-side in the vector \mathbf{r} . Use the following Matlab code to make copies of \mathbf{A} and \mathbf{b} in \mathbf{U} and \mathbf{r} respectively. A multiple (the pivot) of row 1 is subtracted from row 2 and row 3, then a multiple (another pivot) of row 2 is subtracted from row 3.

```
U = A;
r = b;
for row = 1 : 2;
    for i = row + 1 : 3;
        pivot=U(i, row)/U(row, row)
        U(i, :) = U (i, :) - pivot * U(row, :)
        r(i) = r(i) - pivot* r(row)
    end
end
```

As something of a programming challenge try modifying the code to handle larger matrices and to allow partial pivoting and deal with zero pivots.

(ii) Take the solution from part (i) and apply backward substitution to find a solution to the original linear system. Use the following Matlab code, which

assumes that the coefficients of the upper triangular system are stored in the matrix **U** and that the modified right hands are stored in **r**.

```
x(3) = r(3)/U(3, 3);
for i = 2 : -1 : 1
    sum = 0;
    for j = i+1 : 3 ;
        sum = sum + U(i, j)*x(j);
    end;
    x(i) = (r(i) - sum)/U (i, i);
end;
x
```

Check the validity of the solution by evaluating $\mathbf{U} * \mathbf{x}$, where \mathbf{x} is the solution as a column vector. All being well this should produce the original **b**.

3. Matlab has a function *lu* which provides the **LU** factorisation of a matrix. Form the matrix, **A** of coefficients of the following system in a Matlab program.

$$\begin{aligned}x_1 + 2x_2 + 3x_3 + x_4 &= 8 \\2x_1 + 4x_2 - x_3 + x_4 &= 1 \\x_1 + 3x_2 - x_3 + 2x_4 &= 1 \\-3x_1 - x_2 - 3x_3 + 2x_4 &= -7.\end{aligned}$$

Use the *lu* command in the form $[\mathbf{L} \ \mathbf{U}] = \text{lu}(\mathbf{A})$, which returns an upper triangular **U** and a (possibly) permuted lower triangular **L** so that $\mathbf{Ax} = \mathbf{LUb}$, where **b** is the column vector of right-sides.

Having established matrices **L** and **U** we have $\mathbf{LUx} = \mathbf{b}$. Writing $\mathbf{Ly} = \mathbf{b}$, find **y** by using the left-division operator. We now have $\mathbf{Ux} = \mathbf{y}$. Find **x** by using the left-division operator. Check the result by comparing **Ax** and **b**.

Alternatively use the *lu* command in the form $[\mathbf{L} \ \mathbf{U} \ \mathbf{P}] = \text{lu}(\mathbf{A})$ so that **U** is not permuted and a permutation matrix **P** is returned. In this case **b** would be replaced by **Pb** in the scheme shown above, and so **x** could be found by left-division.

4. As an example of how rounding error can affect the solutions to even relatively small systems solve the following equations using the Matlab left-division \ operator.

$$\begin{aligned}-0.6210x_1 + 0.0956x_2 - 0.0445x_3 + 0.8747x_4 &= 5.3814 \\0.4328x_1 - 0.0624x_2 + 8.8101x_3 - 1.0393x_4 &= -1.0393 \\-0.0004x_1 - 0.0621x_2 + 5.3852x_3 - 0.3897x_4 &= -0.3897 \\3.6066x_1 + 0.6536x_2 + 0.8460x_3 - 0.2000x_4 &= -0.0005\end{aligned}$$

Use both 64-bit precision (the default option) and then use 32-bit precision by converting the variables using the *single* operator, for example $B = \text{single}(A)$. For both cases set *format long* to show the maximum number of decimal places for the chosen word length.

5. Form an **LU** factorisation of the following symmetric matrix to show that it is not positive definite.

$$\begin{pmatrix} 4 & 1 & -1 & 2 \\ 1 & 3 & -2 & -1 \\ -1 & -2 & 1 & 6 \\ 2 & -1 & 6 & 1 \end{pmatrix}$$

Using a little ingenuity we can find a non-zero vector such as $\mathbf{x}^T = (0 \ 1 \ 1 \ 0)$ that does not satisfy the requirement $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0$.

6. Show that the following system has no solution.

$$2x_1 + 3x_2 + 4x_3 = 2$$

$$x_1 - 2x_2 + 3x_3 = 5$$

$$3x_1 + x_2 + 7x_3 = 8$$

Use the LU factorisation as provided by the function *lu* to show that the matrix of coefficients is singular. Show by algebraic manipulation that the system does not have a unique solution. Decide if there is no solution or an infinity of solutions.

7. Solve the following systems of linear equations, one of which is a slight variation of the other to show that we have an example of an ill-conditioning

$$\begin{array}{ll} 0.9740x_1 \ 0.7900x_2 \ 0.3110x_3 = 1.0 & 0.9736x_1 \ 0.7900x_2 \ 0.3110x_3 = 1.0 \\ -0.6310x_1 \ 0.4700x_2 \ 0.2510x_3 = 0.5 & -0.6310x_1 \ 0.4700x_2 \ 0.2506x_3 = 0.5 \\ 0.4550x_1 \ 0.9750x_2 \ 0.4250x_3 = 0.75 & 0.4550x_1 \ 0.9746x_2 \ 0.4250x_3 = 0.75. \end{array}$$

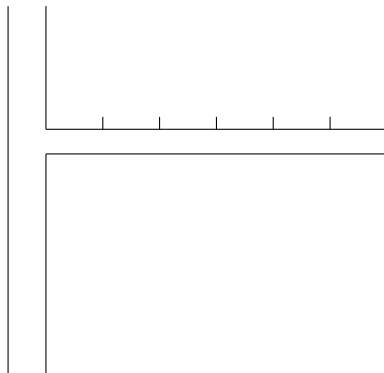
8. As an example of severe ill-conditioning consider the Hilbert² matrix \mathbf{H}_n , which has the form

$$\mathbf{H}_n = \begin{pmatrix} 1 & 1/2 & 1/3 & 1/4 & \dots & 1/n \\ 1/2 & 1/3 & 1/4 & 1/5 & \dots & 1/(n+1) \\ 1/3 & 1/4 & 1/5 & 1/6 & \dots & \\ 1/4 & 1/5 & 1/6 & 1/7 & \dots & \\ \vdots & & & & & \vdots \\ 1/n & 1/(n+1) & \dots & & & 1/(2n-1) \end{pmatrix}.$$

Use the Matlab command $x = \text{ones}(n, 1)$ to generate a column vector \mathbf{x} of n 1's and the command $H = \text{hilb}(n)$ to generate the Hilbert matrix H_n of order n . Form the vector $\mathbf{b}_n = \mathbf{H}_n \mathbf{x}$.

Solve the system $\mathbf{H}_n \mathbf{x} = \mathbf{b}_n$ for various n using the left division operator. In theory \mathbf{x} should be $(1, \dots, 1)$, but see what happens. Begin with $n = 3$ increase the value with suitable increments up to around $n = 18$ to show the effects of ill-conditioning and rounding error. Iterative refinement will not be of any help in this case.

²David Hilbert, 1862–1943. In 1900 he put forward a list of 23 important unsolved problems some of which remain unsolved.

Fig. 2.2 Branch of a tree

9. Write a Matlab program to verify the results quoted for the Gauss–Seidel problem of Sect. 2.8. Use the program to show that Gauss–Seidel iteration applied to the following system with initial estimates of $x_1, x_2, \dots, x_{10} = 0.5$ (or other estimates which appear close to the solution) does not converge.

$$\begin{aligned}\frac{1}{6}x_1 + x_2 &= 1 \\ x_1 + \frac{1}{6}x_2 + x_3 &= 1 \\ x_2 + \frac{1}{6}x_3 + x_4 &= 1 \\ &\vdots \\ x_8 + \frac{1}{6}x_9 + x_{10} &= 1 \\ x_9 + \frac{1}{6}x_{10} &= 1.\end{aligned}$$

10. A bird lands on the branch of a tree and takes steps to the left or to the right in a random manner. If the bird reaches the end of the branch or the tree trunk it flies away. Assume that the branch has the form shown in Fig. 2.2, namely that it has a length equal to six steps. What are the probabilities of the bird reaching the end of the branch from any of the five positions shown in Fig. 2.2. Let the probability of the bird reaching the end of the branch from a position i steps from the trunk be P_i , $i = 0, 1, \dots, 6$. $P_0 = 0$ since if the bird is at the tree trunk it flies away, equally $P_6 = 1$ since the bird is at the end of the branch. For any other position we have $P_i = \frac{1}{2}(P_{i+1} + P_{i-1})$ for $i = 1, 2, \dots, 5$.

This gives rise to the equations

$$\begin{aligned}P_1 &= \frac{1}{2}P_2 \\ P_2 &= \frac{1}{2}(P_3 + P_1) \\ P_3 &= \frac{1}{2}(P_4 + P_2) \\ P_4 &= \frac{1}{2}(P_5 + P_3) \\ P_5 &= \frac{1}{2}(1 + P_4).\end{aligned}$$

Set up a Gauss–Seidel iterative scheme in Matlab to solve the problem. Choose initial estimates such as $P_i = 0$, $i = 1, 2, \dots, 5$. Confirm that the results are in line with expectations.

<http://www.springer.com/978-94-007-1365-9>

Numerical Methods with Worked Examples: Matlab
Edition

Woodford, C.; Phillips, C.

2012, X, 256 p. With online files/update., Hardcover

ISBN: 978-94-007-1365-9