would have to coincide with the data perfectly. Thus the quantity, χ^2 serves as a measure of how far the theory is from the experiment. To get a good "fit" to the data we want the smallest χ^2 possible. However, it is unreasonable to expect a value of χ^2 smaller than the number of data points because of the (assumed) random errors on the data. If it is much less than the number of data points some problem is to be suspected, the errors in the data are not uncorrelated, the errors are overestimated, etc.

We will suppose that the theoretical values, T_i, to be compared with data can be written as:

$$T_i = \sum_{j=1}^{n} b_j f_{ji} \tag{5.3}$$

where the f_{ji} are a series of functions, labeled by j, evaluated for the experimental conditions i, and the b_j are the parameters to be fit. For example, if the data were taken at series of points t_i then the functions, f_{ij} might be monomials in t_i with j denoting the power of t_i. The theoretical values would then be polynomials in the independent variable t. For the example of constant acceleration given above we would have

$$f_{1i} = t_i^0 \equiv 1; \quad f_{2i} = t_i \text{ and } f_{3i} = t_i^2 \tag{5.4}$$

and

$$b_1 = x_0; \quad b_2 = v_0 \text{ and } b_3 = \frac{1}{2}a. \tag{5.5}$$

In the following discussion we assume that the theoretical function is linear in the parameters b_j as expressed by Eq. 5.3. If it is not, then one may still apply the following analysis if we suppose that the b_j represent the coefficients in a Taylor's series expansion about some point. We wish to formulate the problem of finding the "best" fit of the theory to the data by minimizing the function

$$\chi^2 = \sum_{i=1}^{N} \frac{(\sum_{j=1}^{n} b_j f_{ji} - d_i)^2}{e_i^2}. \tag{5.6}$$

Differentiating with respect to each parameter in turn,

$$\frac{d\chi^2}{db_k} = 2\sum_{i=1}^{N} \frac{(\sum_{j=1}^{n} b_j f_{ji} - d_i) f_{ki}}{e_i^2} = 0; \quad k = 1,\ 2,\ 3,\ \ldots\ n \tag{5.7}$$

leading to the system of equations:

$$\sum_{j=1}^{n} a_{kj} b_j = c_k; \quad k = 1,\ 2,\ 3,\ \ldots\ n \tag{5.8}$$

where

$$c_k = \sum_{i=1}^{N} \frac{d_i f_{ki}}{e_i^2}; \text{ and } a_{kj} = \sum_{i=1}^{N} \frac{f_{ki} f_{ji}}{e_i^2}.$$

The solution of these equations will give the desired parameters. Note that the number of equations to be solved is given by the number of parameters and has nothing to do with number of data points, except that if there are not (at least) as many independent data points as parameters the system will not have a unique solution.

5.2 Solution of Linear Equations

Let us consider a set of equations of the form:

$$\begin{array}{lllll} a_{11}x_1 & +a_{12}x_2 & +a_{13}x_3 & \ldots & +a_{1n}x_n = y_1 \\ a_{21}x_1 & +a_{22}x_2 & +a_{23}x_3 & \ldots & +a_{2n}x_n = y_2 \\ a_{31}x_1 & +a_{32}x_2 & +a_{33}x_3 & \ldots & +a_{3n}x_n = y_3 \\ \vdots & \vdots & \vdots & & \vdots \\ a_{n1}x_1 & +a_{n2}x_2 & +a_{n3}x_3 & \ldots & +a_{nn}x_n = y_n \end{array} \tag{5.9}$$

which we can write in abbreviated form as an augmented matrix

$$\left(\begin{array}{ccccc|c} a_{11} & a_{12} & a_{13} & \ldots & a_{1n} & y_1 \\ a_{21} & a_{22} & a_{23} & \ldots & a_{2n} & y_2 \\ a_{31} & a_{32} & a_{33} & \ldots & a_{3n} & y_3 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \ldots & a_{nn} & y_n \end{array}\right). \tag{5.10}$$

In matrix notation Eq. 5.9 is written as:

$$\mathbf{A}\mathbf{x} = \mathbf{y} \tag{5.11}$$

where we ask for the solution of $\mathbf{x}$ for a given $\mathbf{y}$.

We may multiply any of the equations in 5.9 by a constant and subtract it from another without changing the value of the system. Thus we can multiply the first equation by $\frac{a_{21}}{a_{11}} \equiv m_{21}$ and subtract it from the second equation. This leads to:

$$\left(\begin{array}{ccccc|c} a_{11} & a_{12} & a_{13} & \ldots & a_{1n} & y_1 \\ 0 & a_{22} - m_{21}a_{12} & a_{23} - m_{21}a_{13} & \ldots & a_{2n} - m_{21}a_{1n} & y_2 - m_{21}y_1 \\ a_{31} & a_{32} & a_{33} & \ldots & a_{3n} & y_3 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \ldots & a_{nn} & y_n \end{array}\right). \tag{5.12}$$

Next we can multiply the first equation by $m_{31} \equiv \frac{a_{31}}{a_{11}}$ and subtract it from the third equation to reduce the entry in the third row, first column to zero. This process

can be continued until the entire first column is zero except for the first row. One can then use the new diagonal element of the second row-column as the denominator (the pivot) to calculate the m_{32}, m_{42}, ..., m_{n2}, multiply them by the second equation and subtract them from each corresponding row in turn to reduce the elements beyond the second to zero in the second column. Since the element in the first column-second row has already been reduced to zero the other elements in this column will be unchanged.

It is possible in the course of this process that the denominator in m_{ij} may become very small, even zero. In that case it is useful (perhaps necessary) to rearrange the equations so that this is not the case. Thus it is often advisable to search among the remaining equations for the largest value in the column being reduced and reorder the equations so that it becomes the pivot value. This procedure is called partial pivoting. Although the time required for this operation is not great, the coding is somewhat more complex and we will not consider this procedure further. It is often not necessary in physics problems since the diagonal elements tend to be the largest. Note that the denominator in the multiplier m_{ij} is not the original diagonal element of the matrix (except for the first column) but the one at the time the operation is performed.

This process can be continued until all of the elements below the diagonal are zero. At this point the system will have the form:

$$\left(\begin{array}{ccccc|c} b_{11} & b_{12} & b_{13} & \dots & b_{1n} & z_1 \\ 0 & b_{22} & b_{23} & \dots & b_{2n} & z_2 \\ 0 & 0 & b_{33} & \dots & b_{3n} & z_3 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & b_{nn} & z_n \end{array}\right), \tag{5.13}$$

where the b_{ij} and z_i are the result of this process of elimination. Writing again the full equations:

$$\begin{array}{lllll} b_{11}x_1 & +b_{12}x_2 & +b_{13}x_3 & \dots & +b_{1,n-1}x_{n-1} & +b_{1,n}x_n & = & z_1 \\ 0 & +b_{22}x_2 & +b_{23}x_3 & \dots & +b_{2,n-1}x_{n-1} & +b_{2,n}x_n & = & z_2 \\ 0 & +0 & +b_{33}x_3 & \dots & +b_{3,n-1}x_{n-1} & +b_{3,n}x_n & = & z_3 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & +0 & +0 & \dots & +b_{n-1,n-1}x_{n-1} & +b_{n-1,n}x_n & = & z_{n-1} \\ 0 & +0 & +0 & \dots & +0 & +b_{n,n}x_n & = & z_n \end{array} \tag{5.14}$$

we see that the last equation alone determines $x_n = \frac{z_n}{b_{n,n}}$. With this value known, the next-to-last equation gives x_{n-1} and we can continue this process of "back substitution" to generate all of the values of the x_i. This process is called Gaussian elimination and is (with its variants) the basis of many of the techniques for the solution of linear equations.

Note that if two solutions are desired (for two different vectors y) one can simply add the additional vector to the part of the matrix to the right of the solid line and carry out the remainder of the elimination operations on this part of the row as well. Any number of input vectors can be included in this manner. While the operations on the entire row are performed at the same time in the elimination part of the procedure, in the back-substitution phase each y vector is treated independently.

Since the value of the z_i is no longer needed after its use in the back substitution, the value of x_i can replace z_i once it is known. In this way the answer vector x is returned in the same space as was used for the initial y vector (or vectors).

We could also ask to find the inverse of A, A^{-1}, such that

$$\mathrm{A}\mathrm{A}^{-1} = \mathrm{I} \tag{5.15}$$

where I is the identity matrix (all zeros except for the diagonal elements which are unity). We see that this equation has the same form (column by column) as the equations above. Written in terms of an augmented matrix it becomes:

$$\left(\begin{array}{ccccc|ccccc} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & 1 & 0 & 0 & \dots & 0 \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} & 0 & 1 & 0 & \dots & 0 \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} & 0 & 0 & 0 & \dots & 1 \end{array}\right) \tag{5.16}$$

Thus a unit matrix entered on the right of the solid line will be replaced by the inverse of A in this process. The elimination procedure requires the reduction to zero of $\frac{n(n-1)}{2}$ elements and each reduction requires (on average) $\frac{n}{2}$ operations so elimination is an "n^3" algorithm. By this last statement is meant that it takes of the order of n^3 floating point operations to carry out the calculation. The time required for the back substitution, on the other hand, is proportional in leading order to n^2. Thus, for large matrices, the elimination part of the procedure will dominate the time required. Since the elimination can operate over the entire row (the operation is what is referred to in Chapter 4 as "SAXPY") it will vectorize on vector machines. An increase in the length of the row will alter the time required very little.

Thus we see that the time consuming part of the algorithm is the reduction of the matrix to a triangular form.

5.2.1 LU Reductions

The Gaussian elimination procedure just described is but one member in a family of reductions to triangular matrices. Under fairly general assumptions we can write the initial matrix, A, in the form:

$$\mathrm{A} = \mathrm{L}\mathrm{U} \tag{5.17}$$

where L is a matrix with zeros above the diagonal and U is a matrix with zeros below the diagonal. The matrix U for the Gaussian elimination presented in the previous section is the one which results after the elimination, i.e. the b_{ij} in Eq. 5.13. It can be shown (see e.g. Ref. 1, p. 56) that the lower matrix for this case is:

$$\begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ m_{21} & 1 & 0 & \dots & 0 & 0 \\ m_{31} & m_{32} & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{n1} & m_{n2} & m_{n3} & \dots & m_{n,n-1} & 1 \end{pmatrix} \tag{5.18}$$

where the m_{ij} are just the ratios needed to reduce the ij element to zero.

Thus the system,

$$\mathbf{Ax} = \mathbf{y}, \quad \text{or} \quad \mathbf{LUx} = \mathbf{y} \tag{5.19}$$

may be considered as two systems;

$$\mathbf{Ux} = \mathbf{z}, \quad \text{and} \quad \mathbf{Lz} = \mathbf{y}. \tag{5.20}$$

Since both systems are triangular they only require back substitution to solve them and one can first solve for z followed by x with two "n^2" procedures. The back substitution for L in the Gaussian elimination method is done at the same time as the elimination by operating on the y vector simultaneously. Since we know that there are only zeros below the diagonal when we finish the elimination, and that the diagonal values of L are unity, there is no reason to store either the "zeros" of U or the "ones" of L and we have just enough room to store the non-trivial elements of L in the locations that would be occupied by the zeros of U. Thus, with (essentially) no cost we obtain an LU decomposition of the matrix at the same time that a solution is obtained. Many equation solvers operate in two steps. They first perform an LU decomposition with one subroutine and then perform the back substitution with another.

Following is a subroutine which eliminates the first column in a matrix from the second element on. As input it has the preceeding row, normalized to have the first element equal to -1. Note that it operates on m columns, independent of the size of the real matrix so it can perform the elimination for several input vectors or the unit matrix in the case of a matrix inversion. The first step in the "do 3" loop is to zero the first element in the row. After the loop is finished the next statement fills in that element with the value of "m_{ij}" to create the L matrix

```
      subroutine elim(a,t,nn,m,ldim,con1) ! eliminates 1 column
      dimension a(ldim,1),t(1)
      do 4 i=2,nn+1    ! count down the column
```



```
        con=a(i,1)
        do 3 j=1,m
    3   a(i,j)=a(i,j)+con*t(j)    ! SAXPY
        a(i,1)=con1*con   ! fill in L matrix
    4   continue
        return
        end
```

The following routine calls "elim", column by column, to perform a complete elimination (or LU decomposition). The two routines were designed to work together and could have been constructed as a single unit. The reason for separating them is the the inner code might well be replaced by one written in machine language for greater speed.

```
      subroutine gauss(a,n,m,ldim)
      dimension t(1000),a(ldim,1)
      do 10 i=1,n-1
      con=1./a(i,i)
      do 5 k=1,m+1-i
    5 t(k)=-con*a(i,k+i-1)
      call elim(a(i,i),t,n-i,m+1-i,ldim,con)
   10 continue
      return
      end
```

The LU decomposition just studied is by no means unique nor is the way of obtaining it. Doolittle's method finds the same LU decomposition as Gauss' method by a different technique and Crout's method chooses the diagonal elements of U to be unity instead of those of L. The principal advantage of these methods over Gauss' method is that the accumulation of errors is better controlled. On a machine with a large word length there is little to choose among them and we will primarily consider the Gauss method.

For studies of the matrices it is useful to have a practice system. Below is such a matrix to be used in the problems.

$$\begin{pmatrix} 0.546 & 0.447 & 0.242 & 0.194 & 0.795 \\ 0.380 & 0.276 & 0.581 & 0.108 & 0.416 \\ 0.721 & 0.022 & 0.853 & 0.068 & 0.312 \\ 0.151 & 0.759 & 0.186 & 0.597 & 0.757 \\ 0.192 & 0.509 & 0.041 & 0.411 & 0.632 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 0.468 \\ 0.695 \\ 0.398 \\ 0.913 \\ 0.483 \end{pmatrix} \tag{5.21}$$

5.2.2 The Gauss-Seidel Method

For those systems in which the diagonal elements of the matrix A are dominant iterative methods for the solution of linear systems are often useful. Let us consider the solution to Eq. 5.9:

$$\begin{array}{lllll} a_{11}x_1 & +a_{12}x_2 & +a_{13}x_3 \quad \ldots & +a_{1n}x_n & = y_1 \\ a_{21}x_1 & +a_{22}x_2 & +a_{23}x_3 \quad \ldots & +a_{2n}x_n & = y_2 \\ a_{31}x_1 & +a_{32}x_2 & +a_{33}x_3 \quad \ldots & +a_{3n}x_n & = y_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1}x_1 & +a_{n2}x_2 & +a_{n3}x_3 \quad \ldots & +a_{nn}x_n & = y_n \end{array} \tag{5.22}$$

in a situation in which the matrix A is dominated by the diagonal elements. In this case we might believe that a first crude approximation would be given by the solution that one would have if only the diagonal elements were non-zero i.e,

$$\begin{aligned} x_1 &= y_1/a_{11} \\ x_2 &= y_2/a_{22} \\ x_3 &= y_3/a_{33} \\ &\vdots \\ x_n &= y_n/a_{nn}. \end{aligned} \tag{5.23}$$

We can imagine an approximation scheme in which we refine this first approximation by computing successive approximations to the x_i as,

$$\begin{aligned} x_1' &= \frac{y_1 - \sum_{j=2}^{n} a_{1j}x_j}{a_{11}} \\ x_2' &= \frac{y_2 - a_{21}x_1' - \sum_{j=2}^{n} a_{2j}x_j}{a_{22}} \\ x_3' &= \frac{y_3 - \sum_{j=1}^{2} a_{3j}x_j' - \sum_{j=4}^{n} a_{3j}x_j}{a_{33}} \\ &\vdots \\ x_i' &= \frac{y_i - \sum_{j=1}^{i-1} a_{ij}x_j' - \sum_{j=i+1}^{n} a_{ij}x_j}{a_{ii}} \\ &\vdots \\ x_n' &= \frac{y_n - \sum_{j=1}^{n-1} a_{nj}x_j'}{a_{nn}} \end{aligned} \tag{5.24}$$

where we have used the new values of x_i (x_i') if they are defined and the old values if they are not. We may attempt to improve on this algorithm by defining the "new" values as a linear combination of the result of Eqs. 5.24 and the old values.

$$\begin{aligned} x_1'' &= \frac{y_1 - \sum_{j=2}^{n} a_{1j}x_j}{a_{11}} \\ x_2'' &= \frac{y_2 - a_{21}x_1' - \sum_{j=2}^{n} a_{2j}x_j}{a_{22}} \\ x_3'' &= \frac{y_3 - \sum_{j=1}^{2} a_{3j}x_j' - \sum_{j=4}^{n} a_{3j}x_j}{a_{33}} \\ &\vdots \\ x_i'' &= \frac{y_i - \sum_{j=1}^{i-1} a_{ij}x_j' - \sum_{j=i+1}^{n} a_{ij}x_j}{a_{ii}} \\ &\vdots \\ x_n'' &= \frac{y_n - \sum_{j=1}^{n-1} a_{nj}x_j'}{a_{nn}} \end{aligned} \tag{5.25}$$

where now

$$x_i' = (1-w)x_i + wx_i''. \tag{5.26}$$

The parameter w is allowed to vary between 1 and 2. This last method is referred to as the Gauss-Seidel algorithm with successive overrelaxation (SOR). Of course, if $w = 1$, it is the same as the original Gauss-Seidel method given in Eqs. 5.24. By varying the parameter, w, the speed of convergence of the method may be increased by as much as a factor of 10. Of course, there is no guarantee that this process converges. One condition that assures convergence is

$$|a_{ii}| > \sum_{i \neq j} |a_{ij}| \quad \text{for all i.} \tag{5.27}$$

For matrices which are nearly diagonal this technique may be much faster than the elimination method. Of course, if most of the off-diagonal elements are zero it is particularly efficient since those elements do not need to be included in the sums.

5.2.3 The Householder Transformation

There is one other method, introduced by Householder in 1958, which is useful not only for the solution of equations but for the solution of eigenvalue problems considered in the next section. Let us treat a slightly more general decomposition than the LU form, called the QR decomposition. The R denotes the same type of matrix as

U before, an upper (or right) triangular matrix, but Q can represent any matrix. We now have (as before) the two systems,

$$\mathbf{Qz} = \mathbf{y}, \quad \text{and} \quad \mathbf{Rx} = \mathbf{z}. \tag{5.28}$$

Naturally this decomposition is only useful if the system $\mathbf{Qz} = \mathbf{y}$ can be easily solved, as was the case for the lower triangular matrix. Another type of matrix which is easily inverted is the orthogonal matrix which has the property that

$$\mathbf{P}\tilde{\mathbf{P}} = \mathbf{I} \tag{5.29}$$

or

$$\sum_j P_{ij}P_{kj} = \delta_{ik} \tag{5.30}$$

i.e. its transpose *is* its inverse. The problem of the solution of linear equations can now be restated:

Find a matrix (an orthogonal matrix), $\mathbf{Q}^{-1}$, such that

$$\mathbf{Q}^{-1}\mathbf{A} = \mathbf{R}. \tag{5.31}$$

That is, find an orthogonal matrix such that, when multiplied by A the result is an upper triangular matrix. The (orthogonal) matrix , $\mathbf{Q}^{-1}$ can be broken down into a product of several (orthogonal) matrices, each one producing some zeros in the lower triangle (without destroying the ones that were produced by the previously acting matrices).

$$\mathbf{Q}^{-1} \equiv \mathbf{P}^{(n)}\mathbf{P}^{(n-1)}\ldots\mathbf{P}^{(1)}\mathbf{P}^{(0)}. \tag{5.32}$$

Householder solved this problem in the following way. Consider a matrix of the form

$$\mathbf{P}^{(r)} = \mathbf{I} - G\mathbf{h}^{(r)}\tilde{\mathbf{h}}^{(r)} \tag{5.33}$$

i.e., in terms of components,

$$P_{ij}^{(r)} = \delta_{ij} - Gh_i^{(r)}h_j^{(r)} \tag{5.34}$$

where

$$h_i^{(r)} = 0 \quad \text{for} \quad i = 1, 2, \ldots r \tag{5.35}$$

but the values of h_i for $i = r+1,\ r+2,\ \ldots,\ n$ are yet to be determined and will be chosen as a function of the elements of the matrix to be put in upper triangular

form. Thus $\mathbf{P}^{(r)}$ has the form:

$$\begin{pmatrix}
1 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots \\
0 & 1 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots \\
\vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \\
0 & 0 & 0 & \ldots & 1 & 0 & 0 & 0 & \ldots \\
0 & 0 & 0 & \ldots & 0 & 1-Gh_{r+1}^2 & -Gh_{r+1}h_{r+2} & -Gh_{r+1}h_{r+3} & \ldots \\
0 & 0 & 0 & \ldots & 0 & -Gh_{r+1}h_{r+2} & 1-Gh_{r+2}^2 & -Gh_{r+2}h_{r+3} & \ldots \\
0 & 0 & 0 & \ldots & 0 & -Gh_{r+1}h_{r+3} & -Gh_{r+2}h_{r+3} & 1-Gh_{r+3}^2 & \ldots \\
\vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots &
\end{pmatrix}. \tag{5.36}$$

Since we are considering the case in which $\mathbf{P}^{(r)}$ is orthogonal

$$\sum_{j=1}^{n} P_{ij}P_{jk} = \delta_{ik} \tag{5.37}$$

$$= \sum_j \delta_{ij}\delta_{jk} - G\sum_j \delta_{ij}h_jh_k - G\sum_j h_ih_j\delta_{jk} + G^2\sum_j h_ih_j^2h_k \tag{5.38}$$

so

$$G^2\sum h_j^2 = 2G$$

or

$$G = \frac{2}{\sum h_j^2}. \tag{5.39}$$

Thus we are free to chose the h_i to be anything we want and the matrix $\mathbf{P}^{(r)}$ will be orthogonal provided we chose G according to this equation. Since the goal is to u these matrices to bring another matrix, A, into triangular form column by colun consider the action of the Householder matrix on a single column vector x, where is to be thought of as one of the columns of the matrix A.

$$\mathbf{b} = \mathbf{P}^{(r)}\mathbf{x} = \mathbf{x} - G\mathbf{h}^{(r)}\tilde{\mathbf{h}}^{(r)}\mathbf{x} \tag{5.40}$$

or

$$b_i = x_i - Gh_i^{(r)}\sum_j h_j^{(r)}x_j. \tag{5.41}$$

Since the first r values of h are zero, the first r components of b are the same as thc of x. We see that there is a good chance to reduce the remainder of the componer to zero by choosing $h_i^{(r)}$ to be proportional to x_i. In fact we cannot make all of the zero because we have Eq. 5.39 to satisfy, but by an appropriate choice of $h_i^{(r)}$ we c annihilate all but one entry. Let us pick:

$$h_{r+1}^{(r)} = x_{r+1} + z \quad \text{and} \quad h_m^{(r)} = x_m; \quad m = r+2,\ r+3,\ \cdots\ n \tag{5.42}$$

where z is a constant to be determined. In this case the sum in Eq. 5.41 is

$$\sum_j h_j^{(r)} x_j = (x_{r+1} + z)x_{r+1} + \sum_{j=r+2}^{n} x_j^2 \tag{5.43}$$

$$= zx_{r+1} + S^2; \qquad S^2 \equiv \sum_{j=r+1}^{n} x_j^2. \tag{5.44}$$

In order to make the higher components of **b** zero from Eq. 5.41 we must have

$$G(zx_{r+1} + S^2) = 1 \tag{5.45}$$

giving

$$b_{r+1} = -z. \tag{5.46}$$

and

$$b_i = 0 \qquad i = r+2, r+3, \cdots n \tag{5.47}$$

To determine z we combine Eqs. 5.39 and 5.45:

$$2(zx_{r+1} + S^2) = \sum h_j^2 = z^2 + 2zx_{r+1} + S^2$$

or

$$z = \pm\sqrt{S^2}. \tag{5.48}$$

By operating successively with the matrices $P^{(r)}$, a general matrix can be reduced to triangular form.

$$\mathbf{R} = \mathbf{P}^{(n)}\mathbf{P}^{(n-1)} \ldots \mathbf{P}^{(1)}\mathbf{P}^{(0)}\mathbf{A}. \tag{5.49}$$

Here $P^{(0)}$ is chosen to wipe out all elements of the first column except the first one, $P^{(1)}$ zeros all elements of the second column except the first two, etc. Note that the application of $\mathbf{P}^{(1)}$ does not change the first column of $\mathbf{P}^{(0)}\mathbf{A}$ so that the zeros created there remain. Recalling Eq. 5.32

$$\mathbf{Q}^{-1} \equiv \mathbf{P}^{(n)} \ldots \mathbf{P}^{(1)}\mathbf{P}^{(0)} \tag{5.50}$$

then $\mathbf{A} = \mathbf{QR}$ and the equation $\mathbf{Ax} = \mathbf{y}$ can be solved by back substitution on

$$\mathbf{Q}^{-1}\mathbf{Ax} = (\mathbf{Q}^{-1}\mathbf{A})\mathbf{x} = \mathbf{Rx} = \mathbf{Q}^{-1}\mathbf{y} = \mathbf{P}^{(n)} \ldots \mathbf{P}^{(1)}\mathbf{P}^{(0)}\mathbf{y}. \tag{5.51}$$

It is important to realize that the number of steps for multiplication of an $n \times n$ matrix by a Householder matrix is of order n^2 and not n^3 since

$$(PA)_{ij} = A_{ij} - Gh_i \sum_k h_k A_{kj}$$

so that multiplication can be done one row at a time, first calculating

$$C_j = \sum_k h_k A_{kj} \qquad \text{n operations}$$

then for each j.

$$(PA)_{ij} = A_{ij} - Gh_i C_j.$$

5.3 The Eigenvalue Problem

Eigenvalues of matrix (or differential) systems form the basis of much of modern physics. They represent the frequency of the principal modes of motion of systems, energy levels of atoms, nuclei (or any quantum system) and indeed, the masses of the particles making up the universe. Hence the study of numerical techniques for finding the eigenvalues of a mathematical system are of fundamental importance.

5.3.1 Coupled Oscillators

Consider, as a model physics problem, a system of equal masses constrained to move in one dimension. Mass 1 is connected to a fixed wall by a spring with constant k_{01}, mass 2 is connected to mass 1 with a spring with constant k_{12} and mass 3 with k_{23}. In addition there is a spring connecting mass 1 and mass 3 with constant k_{13}. If u_1, u_2, and u_3 denote the displacement of the masses from the unstretched configuration then Newton's equations become:

$$\begin{array}{lllll} m\ddot{u}_1 = & -(k_{01}+k_{12}+k_{13})u_1 & +k_{12}u_2 & +k_{13}u_3 & \\ m\ddot{u}_2 = & k_{12}u_1 & -(k_{12}+k_{23})u_2 & +k_{23}u_3 & \\ m\ddot{u}_3 = & k_{13}u_1 & +k_{23}u_2 & -(k_{23}+k_{13})u_3 & \end{array} . \tag{5.52}$$

In matrix form:

$$m\begin{pmatrix} \ddot{u}_1 \\ \ddot{u}_2 \\ \ddot{u}_3 \end{pmatrix} = \begin{pmatrix} -k_{01}-k_{12}-k_{13} & k_{12} & k_{13} \\ k_{12} & -k_{12}-k_{13} & k_{23} \\ k_{13} & k_{23} & -k_{23}-k_{13} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \tag{5.53}$$

or

$$m\ddot{\mathbf{u}} = \mathbf{Au} \tag{5.54}$$

If we suppose a solution of the form:

$$\mathbf{u} = \sum_j \alpha_j \mathbf{x}_j e^{i\omega_j t}, \tag{5.55}$$

then

$$-m\sum_j \alpha_j \mathbf{x}_j \omega_j^2 e^{i\omega_j t} = \sum_j \alpha_j \mathbf{A}\mathbf{x}_j e^{i\omega_j t}. \tag{5.56}$$

Since each function $e^{i\omega_j t}$ is linearly independent of the others (assuming non-degenerate ω_j) each term will individually satisfy a matrix equation of the form: