第五章 线性方程组与矩阵运算：电路计算与本征值问题

线性代数是物理学研究和工程技术中常见的计算工具。线性代数的计算问题主要包括求解线性方程组求解和矩阵本征值问题。相关的计算工具比较成熟，通常都有现成的库函数供使用。常用的线性代数求解程序包有 LINPACK 和 LAPACK。本章将学习基于计算机求解线性方程组和矩阵的本征值的相关数值方法。

5.1 简单电路的电流计算及高斯消去法

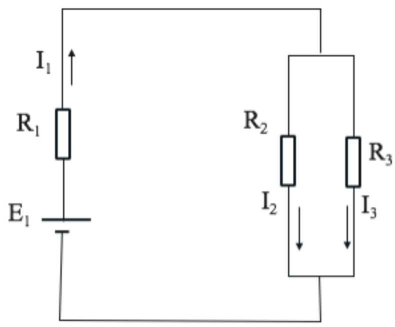

图 5.1

图 5.1 所示简单电路中，电源电动势 E_1 以及电阻 R_1，R_2，R_3 已知，求通过三个电阻的电流，I_1，I_2，I_3。根据基尔霍夫定律，独立回路的电压和为零，独立节点的电流和为零，可列出如下方程组：

$$\begin{cases} R_1 I_1 \qquad\quad + R_3 I_3 = E_1 \\ R_1 I_1 + R_2 I_2 \qquad\quad = E_1 \\ \quad I_1 - \quad I_2 - \quad I_3 = 0 \end{cases} \qquad (5.1)$$

更一般地，上式三元一次方程组可以写为：

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{13}x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{13}x_3 = b_3 \end{cases} \qquad (5.2)$$

其矩阵形式为：

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \qquad (5.3)$$

其中

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \qquad (5.3)$$

称为系数矩阵，

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad (5.4)$$

称为未知数向量，

$$b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \qquad (5.5)$$

称为常数项向量。原矩阵方程可简写为$Ax = b$。更一般的 N 元线性方程组可以相应地写为如下矩阵方程：

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ & & \cdots & & \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \cdots \\ b_n \end{bmatrix} \qquad (5.6)$$

常用的求解线性方程组的方法有高斯消去法和 LU 分解法。高斯消去法主要思路是将原方程组通过初等变换，转化为上三角方程组，然后通过回代得到方程的解。具体地，首先写出方程(5.3)对应的增广矩阵，即将常数项向量和系数矩阵合并：

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{bmatrix} (5.6)$$

然后通过矩阵初等行变换将增广矩阵转换为如下上三角矩阵：

$$\begin{bmatrix} 1 & a'_{12} & a'_{13} & b'_1 \\ 0 & 1 & a'_{23} & b'_2 \\ 0 & 0 & 1 & b'_3 \end{bmatrix} \qquad (5.7)$$

则方程的解可以写为：

$$\begin{aligned} x1 &= b_1' - a_{12}'x2 - a_{13}'x3 \\ x2 &= b_3' - a_{23}'x3 \qquad (5.8) \\ x3 &= b_3' \end{aligned}$$

矩阵初等行变换包括以下操作：1. 互换矩阵两行位置；2. 用非零数乘/除矩阵某行；3. 将矩阵某行的倍数加到矩阵的另一行上。数学上可以证明，如果系数矩阵是非奇异矩阵，则总可以通过初等行变换将系数矩阵 A 转化为上三角矩阵，进而可以通过回代得到方程的解。

设图 5.1 中的 R_1=2.0, R_2=1.0, R_3=4.0 E_1=1.0,则有

$$\begin{bmatrix} 2 & 0 & 4 \\ 2 & 1 & 0 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \qquad (5.9)$$

以下初等行变换将其增广矩阵转化为上三角矩阵，即：

$$\begin{bmatrix} 2 & 0 & 4 & 1 \\ 2 & 1 & 0 & 1 \\ 1 & -1 & -1 & 0 \end{bmatrix} \rightarrow \frac{\text{第 1 行}}{2} \rightarrow \begin{bmatrix} 1 & 0 & 2 & 0.5 \\ 2 & 1 & 0 & 1 \\ 1 & -1 & -1 & 0 \end{bmatrix} \rightarrow \begin{matrix} \text{第 1 行} \times (-2) + \text{第 2 行} \\ \text{第 1 行} \times (-1) + \text{第 3 行} \end{matrix} \rightarrow$$

$$\begin{bmatrix} 1 & 0 & 2 & 0.5 \\ 0 & 1 & -4 & 0 \\ 0 & -1 & -3 & -0.5 \end{bmatrix} \rightarrow \text{第 2 行} + \text{第 3 行} \rightarrow \begin{bmatrix} 1 & 0 & 2 & 0.5 \\ 0 & 1 & -4 & 0 \\ 0 & 0 & -7 & -0.5 \end{bmatrix} \rightarrow \text{第 3 行}$$

$$/(-7) \rightarrow \begin{bmatrix} 1 & 0 & 2 & 0.5 \\ 0 & 1 & -4 & 0 \\ 0 & 0 & 1 & 1/14 \end{bmatrix} \qquad (5.10)$$

将以上得到的上三角矩阵进行回代，得：

$$\begin{cases} I1 = 0.5 - 2I3 = 5/14 \\ I2 = 0 - (-4)I3 = 2/7 \\ I3 = 1/14 \end{cases} \qquad (5.11)$$

```python
# -*- coding: utf-8 -*-
from numpy import array, empty
A = array([[ 2.0,  0.0,  4.0],  #系数矩阵
           [ 2.0,  1.0,  0.0],
           [ 1.0, -1.0, -1.0]])
b = array([1.0, 1.0, 0.0]) #常数项向量
N = len(b)
# Gaussian elimination
for m in range(N):
    #divid by the diagonal element
    div = A[m,m]
    A[m,:] = A[m,:]/div  #除以每行主元
    b[m] = b[m]/div

    # substract from the lower rows
    for i in range(m+1,N):
        mult = A[i,m]
        A[i,:] = A[i,:] - A[m,:]*mult
        b[i] = b[i] - b[m]*mult

# 回代
x = empty(N)
for m in range(N-1,-1,-1):
    x[m] = b[m]
    for i in range(m+1,N):
        x[m] = x[m] - A[m,i]*x[i]
# output results
print("x=",x)
```

Code-5-1

以上高斯消去法可由代码 code-5-1 实现，运行结果为 I_1=0.35714286, I_2=0.28571429, I_3=0.07142857。以上高斯消去算法可直接推广到如下更一般的 N 元线性方程组的求解。

高斯消去法是最直接的求解线性方程组的方法，但是在一些情况下，直接应用高斯消去法会遇到困难。例如，图 5.1 所示电路也可以列出如下方程：

$$\begin{cases} \qquad\quad R_2 I_2 + R_3 I_3 = 0 \\ R_1 I_1 \qquad\qquad + R_3 I_3 = E_1 \\ \;\, I_1 - \quad I_2 - \quad I_3 = 0 \end{cases} \qquad (5.12)$$

则对应的矩阵方程为：

$$\begin{bmatrix} 0 & 2 & 2 \\ 2 & 0 & 2 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad (5.13)$$

方程（5.13）中，由于系数矩阵的对角元为零，无法进行消元，因此高斯消去法遇到困难。显然，方程组（5.12）中，将第一个方程和第二个方程互换位置，便可进行消元，进而将增广矩阵变换为上三角矩阵，求得方程组的解。因此，在采用消去法求解方程组时，首先需要调换方程次序，使得系数矩阵的对角元非零且尽可能大，对应的方法称为主元消去法。

5.2 更一般的线性方程组求解方法：LU 分解法

以上高斯消去法采用逐步消元，将系数矩阵每一列的对角元以下元素消为零需要多步操作，计算效率低。事实上，这种将线性方程组系数矩阵每一列的对角元以下元素消为零的操作可通过矩阵运算来实现，从而大大提高计算效率。例如，可用如下变换矩阵

$$M_1=\frac{1}{a_{11}}\begin{bmatrix}1 & 0 & 0\\ -a_{21} & a_{11} & 0\\ -a_{31} & 0 & a_{11}\end{bmatrix} \quad (5.14)$$

左乘三元一次方程组的系数矩阵，来消去第一列对角元以下元素，即：

$$M_1A=\frac{1}{a_{11}}\begin{bmatrix}1 & 0 & 0\\ -a_{21} & a_{11} & 0\\ -a_{31} & 0 & a_{11}\end{bmatrix}\begin{bmatrix}a_{11} & a_{12} & a_{13}\\ a_{21} & a_{22} & a_{23}\\ a_{31} & a_{32} & a_{33}\end{bmatrix}=\begin{bmatrix}1 & a_{12}^1 & a_{13}^1\\ 0 & a_{22}^1 & a_{23}^1\\ 0 & a_{32}^1 & a_{33}^1\end{bmatrix}=A_1$$

（5.15）

然后，再用如下变换矩阵

$$M_2=\frac{1}{a_{22}^1}\begin{bmatrix}a_{22}^1 & 0 & 0\\ 0 & 1 & 0\\ 0 & -a_{32}^1 & a_{22}^1\end{bmatrix} \quad (5.16)$$

左乘矩阵A_2，消去A^1第二列对角元以下元素，即：

$$M_2A_1=M_2M_1A=\frac{1}{a_{22}^1}\begin{bmatrix}a_{22}^1 & 0 & 0\\ 0 & 1 & 0\\ 0 & -a_{32}^1 & a_{22}^1\end{bmatrix}\begin{bmatrix}1 & a_{12}^1 & a_{13}^1\\ 0 & a_{22}^1 & a_{23}^1\\ 0 & a_{32}^1 & a_{33}^1\end{bmatrix}=\begin{bmatrix}1 & a_{12}^2 & a_{13}^2\\ 0 & 1 & a_{23}^2\\ 0 & 0 & a_{33}^2\end{bmatrix}=A_2$$

（5.17）

最后再用变换矩阵

$$M_3=\frac{1}{a_{33}^2}\begin{bmatrix}a_{33}^2 & 0 & 0\\ 0 & a_{33}^2 & 0\\ 0 & 0 & 1\end{bmatrix} \quad (5.18)$$

左乘矩阵A_2，消去第三列对角元以下元素，即：

$$M_3A_2 = M_3M_2M_1A = \frac{1}{a_{33}^2}\begin{bmatrix} a_{33}^2 & 0 & 0 \\ 0 & a_{33}^2 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & a_{12}^2 & a_{13}^2 \\ 0 & 1 & a_{23}^2 \\ 0 & 0 & a_{33}^2 \end{bmatrix} = \begin{bmatrix} 1 & a_{12}^3 & a_{13}^3 \\ 0 & 1 & a_{23}^3 \\ 0 & 0 & 1 \end{bmatrix} = A_3$$

（5.19）

以上步骤通过对线性方程组的系数矩阵连续左乘变换矩阵，将系数矩阵变换为上三角矩阵。同样地，可以对常数项向量左乘相同的变换矩阵，即：

$$b_3 = M_3M_2M_1b \quad (5.20)$$

因此，原矩阵方程$Ax= b$可以变换为：

$$M_3M_2M_1Ax= M_3M_2M_1b \quad (5.21)$$

即 $A_3x = b_3$，进一步可写为：

$$\begin{bmatrix} 1 & a_{12}^3 & a_{13}^3 \\ 0 & 1 & a_{23}^3 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1^3 \\ b_2^3 \\ b_3^3 \end{bmatrix} \quad (5.22)$$

上式为上三角方程，通过回代可得方程的解。

从以上过程可以看出，求解线性方程组的关键是构造变换矩阵M_k。事实上，对于包含 n 个未知数的线性方程组，变换矩阵M_k可以写为如下一般形式：

$$M_k = \frac{1}{a_{kk}^{k-1}}\begin{bmatrix} a_{kk}^{k-1} & \cdots & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & -a_{k+1k}^{k-1} & a_{kk}^{k-1} & 0 & \cdots \\ 0 & \cdots & -a_{k+2k}^{k-1} & 0 & a_{kk}^{k-1} & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & \cdots \end{bmatrix} \quad (5.23)$$

其逆矩阵也可以很容易写出：

$$M_k^{-1} = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & a_{kk}^{k-1} & 0 & \cdots & 0 \\ \cdots & \cdots & a_{k+1k}^{k-1} & 1 & 0 & \cdots \\ 0 & \cdots & a_{k+2k}^{k-1} & 0 & 1 & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & \cdots \end{bmatrix} \quad (5.24)$$

以上变换矩阵M_k称为高斯变换矩阵，通过对方程系数矩阵 A 连续左乘高斯矩阵M_k，可得上三角矩阵，从而得到方程的解。

例如，对图 5.1 的简单电路，可根据上式构造变换矩阵，对系数矩阵和常数项矩阵进行连续左乘，进而将原方程变换为上三角方程，具体过程如下：

首先利用变换矩阵M_1将第一列变换为 1，并消去第一列对角元以下元素，同时对常数项向量进行变换：

$$M_1A=\frac{1}{2}\begin{bmatrix}1&0&0\\-2&2&0\\-1&0&2\end{bmatrix}\begin{bmatrix}2&0&4\\2&1&0\\1&-1&-1\end{bmatrix}=\begin{bmatrix}1&0&2\\0&1&-4\\0&-1&-3\end{bmatrix}\quad(5.24)$$

$$M_1b=\frac{1}{2}\begin{bmatrix}1&0&0\\-2&2&0\\-1&0&2\end{bmatrix}\begin{bmatrix}1\\1\\0\end{bmatrix}=\begin{bmatrix}1/2\\0\\-1/2\end{bmatrix}\quad(5.25)$$

然后利用变换矩阵M_2将第一列变换为 1，并消去第二列对角元以下元素，同时对常数项向量进行变换

$$M_2M_1A=\begin{bmatrix}1&0&0\\0&1&0\\0&1&1\end{bmatrix}\begin{bmatrix}1&0&2\\0&1&-4\\0&-1&-3\end{bmatrix}=\begin{bmatrix}1&0&2\\0&1&-4\\0&0&-7\end{bmatrix}\quad(5.26)$$

$$M_2M_1b=\begin{bmatrix}1&0&0\\0&1&0\\0&1&1\end{bmatrix}\begin{bmatrix}1/2\\0\\-1/2\end{bmatrix}=\begin{bmatrix}1/2\\0\\-1/2\end{bmatrix}\quad(5.27)$$

最后利用变换矩阵M_3将第三列对角元变换为 1，同时对常数项向量进行变换

$$M_3M_2M_1A=-\frac{1}{7}\begin{bmatrix}-7&0&0\\0&-7&0\\0&0&1\end{bmatrix}\begin{bmatrix}1&0&2\\0&1&-4\\0&0&-7\end{bmatrix}=\begin{bmatrix}1&0&2\\0&1&-4\\0&0&1\end{bmatrix}\quad(5.28)$$

$$M_3M_2M_1b=-\frac{1}{7}\begin{bmatrix}-7&0&0\\0&-7&0\\0&0&1\end{bmatrix}\begin{bmatrix}1/2\\0\\-1/2\end{bmatrix}=\begin{bmatrix}1/2\\0\\5/70\end{bmatrix}\quad(5.29)$$

因此有：

$$\begin{bmatrix}1&0&2\\0&1&-4\\0&0&1\end{bmatrix}\begin{bmatrix}I_1\\I_2\\I_2\end{bmatrix}=\begin{bmatrix}1/2\\0\\5/70\end{bmatrix}\quad(5.30)$$

回代可得电流值。

以上过程可以用更为简洁的方式实现。根据以上讨论，线性方程组的系数矩阵可分解为如下形式：

$$A = LU \quad （5.31）$$

其中，$U = M_3M_2M_1A$ 为上三角矩阵，$L = M_1^{-1}M_2^{-1}M_3^{-1}$为下三角矩阵。则原矩阵方程$Ax = LUx = b$ 可分解为两个线性方程组：

$$Ly = b \quad （5.31）$$

和

$$Ux = y \quad （5.32）$$

方程组（5.31）为下三角方程组，因此可以通过回代直接求得未知数向量 y。然后将向量 y 代入上三角方程组（5.32），以通过回代直接求得方程的解 x。这种将系数矩阵分解为上三角矩阵 U 和矩阵三角矩阵 L，从而将原方程组转换为可以直接回代求解的上三角方程组和下三角方程组的方法称为 LU 分解法。LU 分解分有两个显著的优点。首先，用矩阵运算的方法将系数矩阵变换为三角矩阵可以提高计算效率；其次 LU 分解过程不依赖常数项矩阵，因此对具有相同系数矩阵，不同常数项矩阵的线性方程组的求解，可以避免反复 LU 分解过程，提高计算效率。LU 分解法是最为常用的线性方程组求解算法，代码 code-5-2 给出了调用函数求解线性方程组（5.9）的实现过程。

```python
# -*- coding: utf-8 -*-
from numpy import array, linalg
A = array([[ 2.0,  0.0,  4.0],
           [ 2.0,  1.0,  0.0],
           [ 1.0, -1.0, -1.0]])
b = array([1.0, 1.0, 0.0])
x = linalg.solve(A,b) #线性方程求解函数 (LU)
print("x=",x)
```

Code-5-2

5.3 矩阵对角化与本征值问题

以上描述线性方程组的矩阵方程$\boldsymbol{Ax} = \boldsymbol{b}$中，将常数项向量用一常数$\boldsymbol{\lambda}$与未知数向量 x 的乘积代替，得如下方程：

$$\boldsymbol{Ax} = \boldsymbol{\lambda x} \qquad （5.33）$$

方程（5.33）表示用矩阵 $\boldsymbol{A}$ 对一向量$\boldsymbol{x}$进行变换操作后，只是改变向量x的大小而不改变其方向。除零向量外，只有特殊的非零向量$\boldsymbol{x}$和常数λ值才能使方程(5.33）成立。我们称方程(5.33）为本征方程，满足本征方程(5.33）的非零向量x称为本征向量，对应的λ值称为本征值。给定矩阵求解其本征值和本征向量是基本的线性代数运算。很多物理问题的求解可归结为计算矩阵的本征值和本征向量。

图 5.3 给出的是用弹簧链接的小球，小球质量为 m，弹簧的倔强系数为 K，弹簧两端固定，且小球只能沿 x 方向做一维运动。求该一维弹性系统的可能振动模式和对应的振动频率[1]。

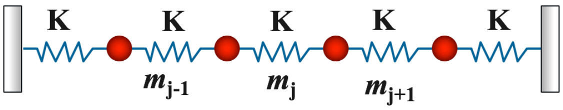

图 5.3 一维弹性系统振动问题。

设x_i表示第j个小球相对于平衡位置的位移，则根据小球受力情况可以写出如下微分方程：

$$\begin{cases} m\dfrac{d^2x_1}{dt^2} = -Kx_1 + K(x_2 - x_1) \\ m\dfrac{d^2x_2}{dt^2} = -K(x_1 - x_2) + K(x_3 - x_2) \\ \qquad\cdots \\ m\dfrac{d^2x_j}{dt^2} = -K(x_{j-1} - x_j) + K(x_{j+1} - x_j) \\ \qquad\cdots \\ m\dfrac{d^2x_N}{dt^2} = -K(x_{N-1} - x_N) - Kx_N \end{cases} \tag{5.34}$$

设小球质量 m=1，弹簧倔强系数 K=1，并将第 j个小球运动方程的试探解$x_j =$ $A_j e^{-i\omega t}$代入上式得：

$$\omega^2 \begin{pmatrix} A_1 \\ A_2 \\ \cdots \\ A_j \\ \cdots \\ A_6 \end{pmatrix} = \begin{pmatrix} 2 & -1 & 0 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 \\ 0 & 0 & -1 & 2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & & & \cdots & 2 \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \\ \cdots \\ A_j \\ \cdots \\ A_6 \end{pmatrix} \tag{5.35}$$

上式与本征方程$Ax = \lambda x$具有相同的形式，其中每一个满足方程的本征向量$[A_1 \quad A_2 \quad \cdots \quad A_j \quad \cdots \quad A_N]^T$表示系统的一种特定的集体振动模式，由每个小球的相对振幅决定。而本征值ω^2表示这种振动模式对应的圆频率的平方。矩阵

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 \\ 0 & 0 & -1 & 2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & & & \cdots & 2 \end{pmatrix} \quad (5.36)$$

表示连接矩阵（也称为 Kirchhoff 矩阵），其中对角元表示小球的连接数，非对角元-1 表示有链接，非对角元 0 表示无连接。对自由度为 N（N 个小球）的系统，共有 N 个不同的振动模式（N 个本征向量），对应 N 个不同的振动频率（N 个本征值）。因此，一维振动系统的求解过程转变为求$Ax = \lambda x$的本征值和本征向量的问题，即：

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1j} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2j} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{i1} & \cdots & \cdots & a_{ij} & \cdots & a_{in} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nj} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_i \\ \cdots \\ x_n \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_i \\ \cdots \\ x_n \end{bmatrix} \quad (5.37)$$

若将 N 个本征向量做为矩阵 X 的每一列，N 个本征值做为对角矩阵Λ的对角元，则有如下等式：

$$AX = X\Lambda \quad (5.38)$$

式（5.37）可进一步写为：

$$\begin{bmatrix} a_{11-\lambda} & a_{12} & \cdots & a_{1j} & \cdots & a_{1n} \\ a_{21} & a_{22}-\lambda & \cdots & a_{2j} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{i1} & \cdots & \cdots & a_{ij}-\lambda & \cdots & a_{in} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nj} & \cdots & a_{nn}-\lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_i \\ \cdots \\ x_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \cdots \\ 0 \\ \cdots \\ 0 \end{bmatrix} \quad (5.39)$$

以上齐次线性方程组具有非零解的充分必要条件是系数矩阵的行列式为零，即：

$$\det(A - \lambda I) = 0 \quad (5.39)$$

显然，如果矩阵 A 为对角矩阵，则满方程（5.39）解的 N 个λ值对应对角矩阵 A 的 N 个对角元。根据线性代数的知识，相似变换不改变矩阵的本征值。设存在正交矩阵 Q，通过相似变换$Q^T AQ$将矩阵 A 变换为对角矩阵，则对角矩阵的对角元为矩阵 A 的本征值，正交矩阵 Q 的每一列为对应本征值的本征向量。因此，求矩阵本征值问题的关键是构建正交矩阵 Q，通过相似变换将矩阵 A 变换为对角矩阵的，即矩阵对角化。以下将介绍一种矩阵对角化的常用算法，即 QR 分解法。

QR 分解法假设矩阵 A 可以分解为正交矩阵 $\boldsymbol{Q}$ 与上三角矩阵 $\boldsymbol{R}$ 的乘积，即

$$A = QR \quad (5.40)$$

具体操作如下：

首先将系数矩阵分解为正交矩阵Q_1与上三角矩阵R_1的乘积 $A = Q_1 R_1$，并定义一个新的矩阵 $A_1 = R_1 Q_1$，则A_1可写为： $A_1 = Q_1^T AQ_1$；

同样地，将矩阵A_1分解为正交矩阵Q_2与上三角矩阵R_2的乘积$A_1 = Q_2 R_2$，并定义新的矩阵$A_2 = R_2 Q_2$，则A_2可写为： $A_2 = Q_2^T AQ_2$。

重复以上操作，将A_2分解为$A_2 = Q_3 R_3$，定义$A_3 = R_3 Q_3$，则 $A_3 = Q_3^T Q_2^T Q_1^T AQ_1 \ Q_2 Q_3$

数学上可以证明，以上操作一直重复下去， A_i 将收敛到对角矩阵Λ，Λ的对角元为 A 的本征值。另外，如果设$X = Q_1 Q_2 Q_3 \cdots Q_k$， 则 X 的每一列为对应矩阵 A 的本征向量。这样便可得到矩阵 A 的所有本征值和本征向量。该方法称为 QR 分解法，其核心是构造出正交矩阵 Q 与上三角矩阵 R，使得$\mathrm{A} = \mathrm{QR}$成立。满足方程（5.40）的正交矩阵 Q 与上三角矩阵 R 并不唯一，通常选择最简洁方便的构造方法，其中 Gram-Schmidt 变换是广泛使用的一种构造正交矩阵 Q 与上三角矩阵 R 算法。具体操作步骤如下：

首先将矩阵 A 写为列向量形式，即$A = [\boldsymbol{a_1}, \boldsymbol{a_2}, \cdots, \boldsymbol{a_i}, \cdots, \boldsymbol{a_n}]$。设$\boldsymbol{u_i} = \boldsymbol{a_i} - \sum_{j=0}^{i-1} (\boldsymbol{q_i} \cdot \boldsymbol{a_i}) \boldsymbol{q_j}$ ，其中 $\boldsymbol{q_i} = \boldsymbol{u_i} / |\boldsymbol{u_i}|$，则 Q，R 可分别写为：

$$\boldsymbol{Q} = [\boldsymbol{q_1}, \boldsymbol{q_2}, \cdots, \boldsymbol{q_i}, \cdots, \boldsymbol{q_n}] \quad (5.41)$$

和

$$\boldsymbol{R} = \begin{bmatrix} |\boldsymbol{u_1}| & \boldsymbol{q_1} \cdot \boldsymbol{a_2} & \boldsymbol{q_1} \cdot \boldsymbol{a_3} & \cdots & \boldsymbol{q_1} \cdot \boldsymbol{a_n} \\ \boldsymbol{0} & |\boldsymbol{u_2}| & \boldsymbol{q_2} \cdot \boldsymbol{a_3} & \cdots & \boldsymbol{q_2} \cdot \boldsymbol{a_n} \\ \boldsymbol{0} & \boldsymbol{0} & |\boldsymbol{u_3}| & \cdots & \boldsymbol{q_3} \cdot \boldsymbol{a_n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & |\boldsymbol{u_n}| \end{bmatrix} \quad (5.42)$$

基于（5.41）和（5.42），可以由系数矩阵 A 构造出正交矩阵 Q 与上三角矩阵 R，并进而对系数矩阵进行一系列相似变换，得到本征值与本征向量。Numpy 的线性代数包 linalg 中的 eigh()是常用的求解本征值问题的函数。

回到图 5.1 的一维弹性系统的振动问题，设该系统包含 1000 个小球，则其求解过程由代码 code-5-3 实现。

```
# -*- coding: utf-8 -*-
from numpy import arange, zeros, linalg
import pylab as pl

n=1000
a = zeros([n,n])
for i in range(n): #构建连接矩阵
    for j in range(n):
        if(i == j + 1):
            a[i,j] = -1.
        if(j == i + 1):
            a[i,j] = -1.
        if(i == j):
            a[i,j] = 2
eigenValues,eigenVectors = linalg.eigh(a) #计算本征值, 本征向量
idx = eigenValues.argsort()  #按本征值生序排序
eigenValues = eigenValues[idx] #按本征值生序排序
eigenVectors = eigenVectors[:,idx] #按本征值生序排序
# print out
print(eigenValues)  #输出
print(eigenVectors)

index=arange(n)
fig = pl.figure(figsize=(12,4))
ax1 =fig.add_subplot(1,2,1)
ax2 =fig.add_subplot(1,2,2)

ax1.plot(index,eigenValues,'ro',label='mode') #输出本征值
ax2.plot(index,eigenVectors.T[50,:],'r-',label='mode') #输出指定本征向量

ax1.set_xlabel(r'mode id', fontsize=20)
ax1.set_ylabel(r'frequence', fontsize=20)
ax2.set_xlabel(r'position', fontsize=20)
ax2.set_ylabel(r'amplitude', fontsize=20)

pl.subplots_adjust(left=0.15,bottom=0.1,top=0.9,right=0.95, \
                   hspace=0.25,wspace=0.3)
pl.show()
```

Code-5-3

图 5.4 是运行代码 code-5-3 得到的不同振动模式的振动频率平方以及最低频率的振动模式（即每个小球的相对振幅）。在最低频振动模式中，中间位置的小球具有更大的振幅，覆盖半个波长。图 5.5 给出了更高频率的振动模式，具有更短的波长。

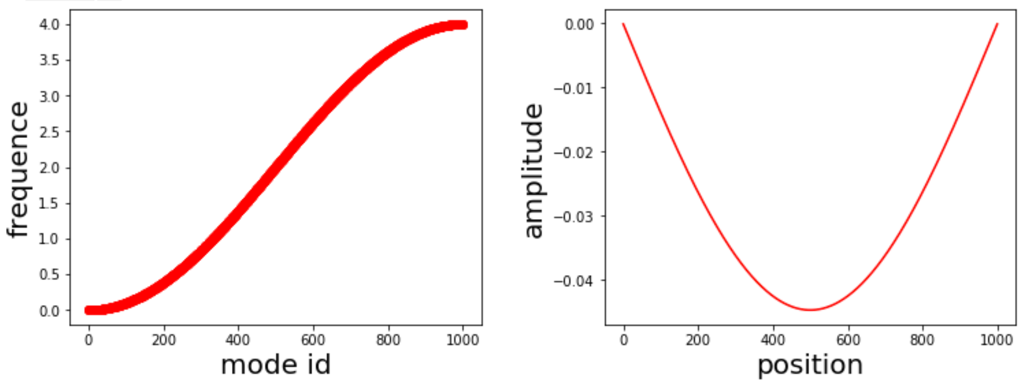

图 5.4 本征值最低频震动模式。

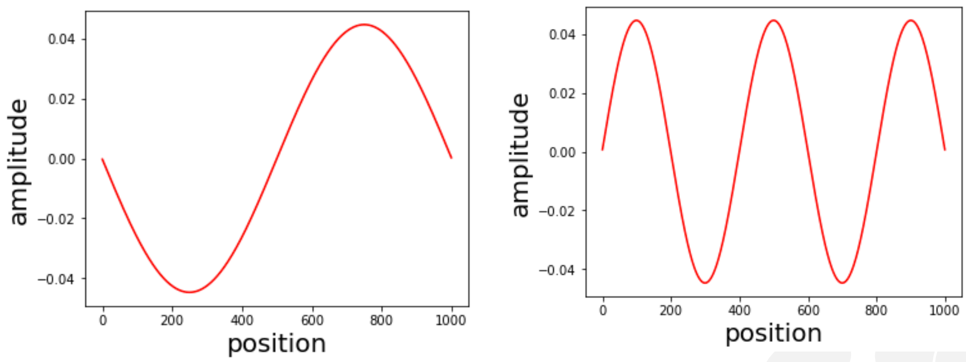

图 5.5 更高频率的两种振动模式。

类似的方法可以用于讨论二维薄膜的振动问题。图 5.6 的二维薄膜可以看作是由二维平面上用弹簧链接的小球组成[1]。

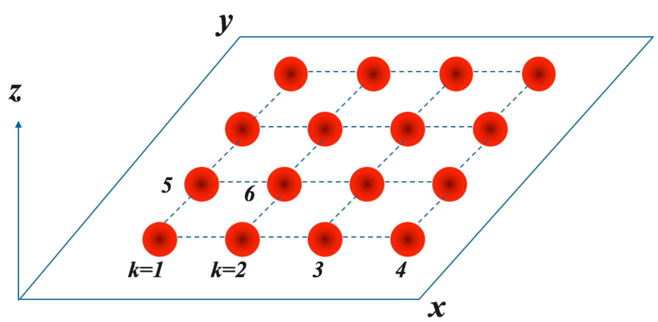

图 5.6 二维薄膜弹性系统

按照图 5.6 所示的小球编号方式，可以写出二维薄膜的连接矩阵为：

$$A=\begin{bmatrix} -1 & 4 & -1 & \cdots & 0 & 0 & -1 & \cdots \\ 0 & -1 & 4 & \cdots & 0 & 0 & 0 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 4 & -1 & 0 & \cdots \\ -1 & 0 & 0 & \cdots & -1 & 4 & -1 & \cdots \\ 0 & -1 & 0 & \cdots & 0 & -1 & 4 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots \end{bmatrix} \quad (5.43)$$

代码 Code-5-4 给出了连接矩阵的构建以及本征值问题的求解过程。

```python
# -*- coding: utf-8 -*-
from numpy import arange, zeros, linalg, histogram, meshgrid, reshape, sqrt
import pylab as pl
from mpl_toolkits.mplot3d import Axes3D

n=900
n0 = 30
a = zeros([n,n])
for i in range(n): #构建连接矩阵
    for j in range(n):

        if(i == j + 1):
            a[i,j] = -1.
        if(j == i + 1):
            a[i,j] = -1.
        if(i==j+n0 and j+n0<n):
            a[i,j] = -1
        if(j==i+n0 and i+n0<n):
            a[i,j] = -1
        if(i == j):
            a[i,j] = 4

eigenValues,eigenVectors = linalg.eigh(a) #计算本征值，本征向量
idx = eigenValues.argsort()#按本征值生序排序
eigenValues = eigenValues[idx]
eigenVectors = eigenVectors[:,idx]
# spectrum
eigenValues=sqrt(eigenValues)
prob, edge = histogram(eigenValues,bins = 15)#直方图计算频谱
xedge = 0.5*(edge[0:-1]+edge[1:])

index=arange(n)
#  ploting
fig = pl.figure(figsize=(12,4))
ax1 =fig.add_subplot(1,2,1)
ax2 =fig.add_subplot(1,2,2)
ax1.plot(index,eigenValues,'ro',label='mode')#输出本征值
ax2.plot(xedge,prob,'r-',label='spectrum') #输出频率分布
ax1.set_xlabel(r'mode id', fontsize=20)
ax1.set_ylabel(r'frequence', fontsize=20)
ax2.set_xlabel(r'frequence', fontsize=20)
ax2.set_ylabel(r'Prob', fontsize=20)
pl.legend(loc='upper left')

#------------------------------
Amplitude = eigenVectors.T[20,:]
Amplitude0=reshape(Amplitude,(n0,n0))
X=arange(0,n0)
Y=arange(0,n0)
X, Y = meshgrid(X, Y)

fig2 = pl.figure()
ax3 = Axes3D(fig2)
ax3.plot_wireframe(X,Y,Amplitude0,color='r')#输出指定本征向量
ax3.set_ylabel(r'Y', fontsize=20)
ax3.set_xlabel(r'X', fontsize=20)
ax3.set_zlabel(r'Amplitude', fontsize=20)
pl.show()
```

Code-5-4

运行代码 code-5-4 可以得到连接矩阵的本征值和本征向量，即薄膜的所有振动模式和相应的频率。图 5.7（左）给出的是不同振动模式的频率（本征值的开平方）。频率的分布由图 5.7（右）给出。通常，分子振动光谱对应红外波段，能够通过红外光谱仪测定样品的红外光谱，并可与图 5.7（右）给出频谱相比较。这种通过矩阵对角化计算分子振动谱的方法称为正则模分析。系统单元的运动情况可以通过所有振动模式的叠加得到，这种描述方式等等价于相空间轨迹的描述。图 5.8 给出了低频区的几种震动模式（本征向量），即薄膜各点处的振幅。对照前面学过的波动方程的数值求解结果（驻波情形）一致。

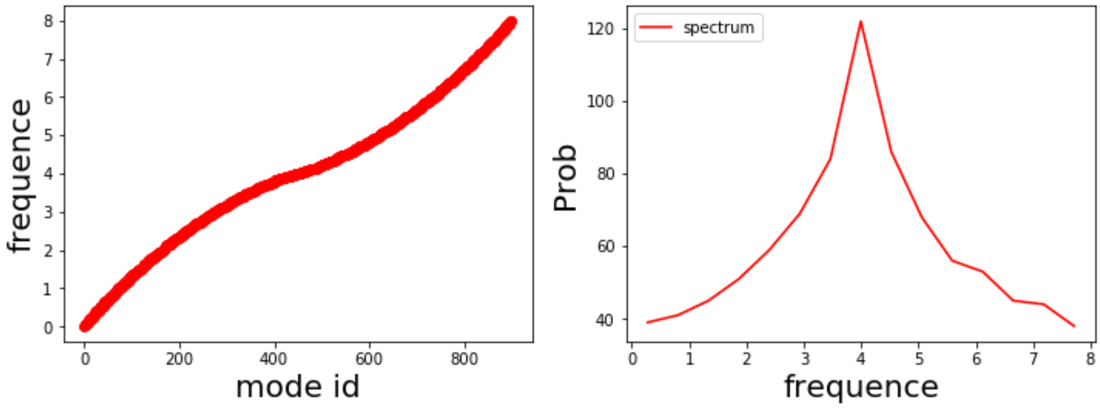

图 5.7 二维薄膜系统的本征值与频率分布。

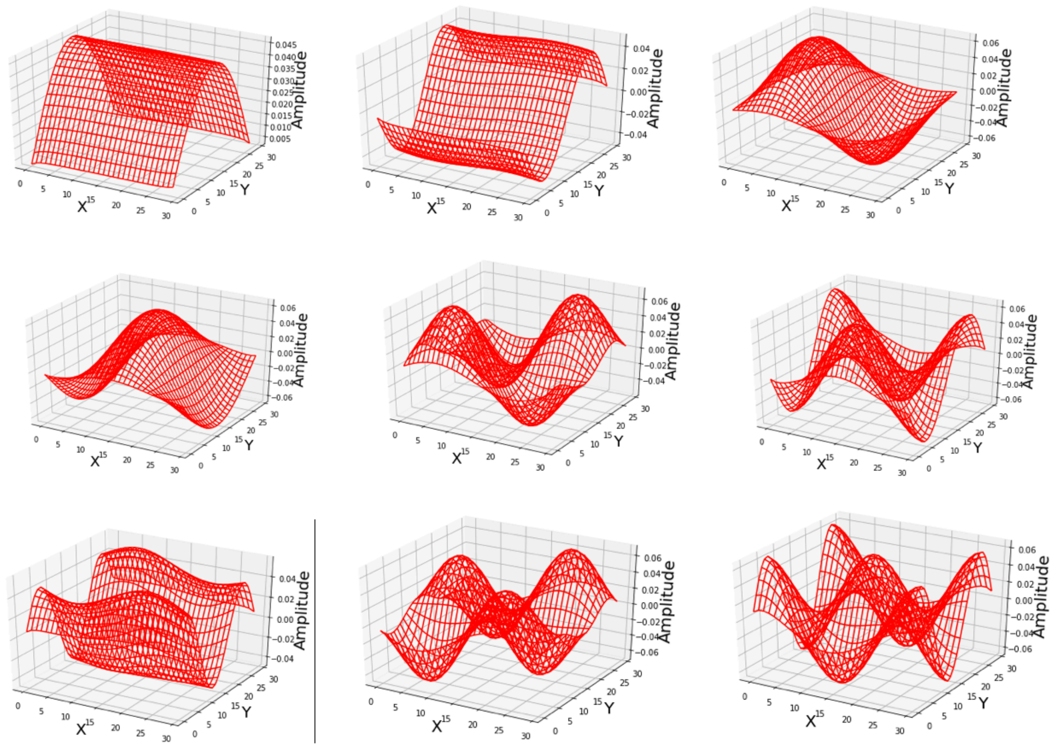

图 5.8 二维薄膜系统的几种低频振动模式图。