光电子器件计算机辅助分析教程

陈维友 著

吉林大学出版社
JILIN UNIVERSITY PRESS

光电子器件计算机辅助分析教程

陈维友 著

吉林大学出版社

内容简介

本书从光波导模式分析、光波导器件 BPM 模拟、MOCVD 反应室模拟、半导体激光器模拟、OEIC 模拟等几个方面，较系统地阐述了光电子器件计算机辅助分析(CAA)的基本思想和一些基础知识。通过本书，可以使读者了解光电子器件 CAA 的概貌和较深入的理论知识以及一些实用的数值技术，完全可以胜任本领域的 CAA 研究与应用工作。

本书主要作为光电子专业高年级本科生和研究生教材或教学参考书，亦可供光电子领域科研人员参考。

图书在版编目(CIP)数据

光电子器件计算机辅助分析教程/陈维友著.—长春：吉林大学出版社，2008.10

ISBN 978-7-5601-3971-5

Ⅰ.光… Ⅱ.陈… Ⅲ.光电子技术-电子器件-计算机辅助分析-教材 Ⅳ.TN2

中国版本图书馆 CIP 数据核字(2008)第 156516 号

书　名：光电子器件计算机辅助分析教程
作　者：陈维友　著

责任编辑、责任校对：赵洪波
吉林大学出版社出版、发行
开本：787×1092　毫米 1/16
印张：13.5　　字数：326 千字
ISBN 978-7-5601-3971-5

封面设计：孙　群
吉林科华印刷厂　印刷
2008 年 10 月　第 1 版
2008 年 10 月　第 1 次印刷
定价：32.00 元

社址：长春市明德路 421 号　邮编：130021
发行部电话：0431-88499826
网址：http://www.jlup.com.cn
E-mail:jlup@mail.jlu.edu.cn

前　言

计算机辅助设计(CAD)技术几乎在各个领域都得到了重要的应用，并越来越为人们所重视，特别是在信息领域，如果没有CAD几乎就没有信息技术蓬勃发展的今天。光电子器件CAD正在光电子器件研发过程中发挥着越来越重要的作用。光电子器件CAA是光电子器件CAD的重要组成部分，属于与物理机制密切相关的技术CAD(TCAD)领域。

作者多年从事光电子器件计算机辅助分析的研究和教学工作，并且出版了《光电子器件模型与OEIC模拟》(国防科技图书出版基金资助)、《半导体激光器计算机辅助分析与设计》(国家自然科学基金专著出版基金资助)和《光波导器件CAD》三部学术著作。多年的研究生教学工作和研究生指导工作使我们深感这方面教学书籍的缺乏，并且有很多从事光电子器件研究的科技人员迫切需要这方面的基础理论知识和数值技术，这些正是作者编写本书的起因。

本书以上述三本书为基础，以光电子专业高年级本科生和研究生教学为目的，以学术思想、基本知识介绍为主要内容，较系统全面地给出了基本的数值计算技术、光电子CAA概貌和较深入的模型技术。

实际问题的计算机辅助分析过程主要包括四个环节：第一环节是对实际问题做深入细致的分析，明晰所要解决问题的要求和提供的条件。第二环节是确定问题的物理模型，这个模型要能充分反映实际问题的内在物理机制，并能体现分析要求。通常，一个实际问题往往都有其特殊性，针对其特殊性，往往可以从最基本的理论和定律(经典力学、电磁场理论、量子力学、它的对论、物质不灭、能量守恒、动量守恒等)出发得到简化而实用的物理模型。确定模型是实现计算机辅助分析的关键环节，直接影响问题求解的准确性和求解效率。第三环节是采用一定的数值计算方法对物理模型进行离散。比较常用的解决边值问题的数值计算方法是有限差分法和有限元法，比较常用的解决初值问题的数值计算方法是欧拉方法、梯形法等。第四环节是求解线性方程组或本征值方程。物理模型经过数值离散后，最后形成的是线性方程组或本征值方程，因此，求解线性方程组和本征值方程是计算机辅助分析中最基本的计算技术。这四个环节可以归结为16个字：剖析问题、建立模型、数值离散、求解方程。前两个环节具有特殊性，不同问题需要建立不同的物理模型；后两个环节具有通用性。按这四个环节概括本书如下：

剖析问题：剖析光波导模式求解问题；剖析光在光波导中传输问题；剖析气流在MOCVD反应室流动问题；剖析半导体激光器微观物理机制及其模拟问题；剖析光电集成回路分析问题。

建立模型：以麦克斯韦电磁场理论为基础建立光波导模式求解物理模型；以麦克斯韦电磁场理论为基础建立光在光波导中传输过程物理模型；以流体力学和热力学为基础建立MOCVD反应室气流流动物理模型；以电磁场理论、量子力学等理论为基础建立能够充分反映半导体激光器微观物理机制的物理模型；借鉴微电子电路分析理论及微电子元器件电路模型的构造方法，构造光电子器件等效电路模型，在此基础上，借用微电子电路模拟软件，实现对光

电集成回路的模拟。

数值离散：二阶微分方程边值问题常用的离散方法是有限差分法、有限元方法；一阶微分方程初值问题常用的离散方法是欧拉方法、梯形法、诺曼(Von Neumann)方法。

求解方程：线性方程组求解方法一般用高斯消去法，直接迭代法；本征值方程求解方法一般用直接迭代法，雷利(Rayleigh)商迭代法。

本书共分七章：第一章介绍基本的数值技术，重点介绍上述第三和第四环节涉及的通用数值技术，特别注重与稀疏矩阵有关的数值计算方法。第二章为光波导模式求解技术。第三章为光传播过程模拟技术。第四章给出光波导编辑技术。这三章重点给出了与光波导有关的模式求解、BPM模拟以及波导编辑技术。第二章和第三章叙述的比较详尽，基本体现了上述四个环节的实现过程。第五章为MOCVD反应室流体力学模拟技术，属于工艺模拟范畴。第六章为半导体激光器模拟技术，属于器件模拟范畴，重点阐述用于半导体激光器模拟的物理模型。第五章和第六章主要介绍物理模型的构造方法，重点放在第一和第二个环节，后续两个环节可借鉴第二和第三章的做法来处理。第七章为光电集成回路模拟技术，属于电路模拟范畴，重点阐述OEIC电路级模拟的基本思想和光电子器件电路模型的构造方法。

由于作者水平有限，书中难免存在错误和不妥之处，恳请广大读者批评指正。

作者于吉林大学

2004年3月

目 录

第一章　数值计算基础

本章简单介绍计算机辅助分析四个环节中的后两个环节(数值离散、求解方程)涉及的一些必要的实用的数值计算方法,特别是与大规模稀疏矩阵有关的数值计算方法。本章概貌如下:

1. 数值离散环节

(1)二阶微分方程边值问题的离散方法:有限差分方法,其核心思想是用差分代替微分;有限元方法,其核心思想是用积分代替微分。

(2)一阶微分方程初值问题的离散方法:向前欧拉方法、向后欧拉方法、梯形法、诺曼(Von Neumann)方法。第一种算法属于显式计算方法,优点是算法相对简单,缺点是数值稳定性欠佳。其它三种方法属于隐式计算方法,优点是数值稳定性较好,缺点是算法相对复杂。

2. 求解方程环节

(1)线性方程组的求解方法:高斯消去法,是经典的直接求解方法,适合于各种线性方程组;迭代求解法,适合于主对角线元素数值占优的线性方程组,优点是求解过程中不改变系数矩阵和常数向量的结构和元素数值,适合稀疏矩阵线性方程组的求解。

(2)本征值方程求解方法:直接迭代法,可以求出最大本征值和相应的特征向量;雷利(Rayleigh)商迭代法,用来求某些本征值及本征向量。两种方法都具有不改变系数矩阵和常数向量的结构和元素数值的优点,适合稀疏矩阵本征值问题的求解。

3. 其它技术

(1)非线性方程的求解方法:牛顿迭代法。

(2)非线性方程组的求解方法。

(3)稀疏矩阵的概念及存储技术:固定对角带状矩阵的存储法,按行存储法,链表存储法。

这里特别指出,对于从事计算机辅助分析与设计的研究人员来说,数值计算方法是绝对不能忽视的。数值计算方法是把物理方程转化为可以用计算机进行求解的方法。对于同一个物理方程,可以采用不同的数值计算方法进行数值计算,譬如,在求解空间分布问题时,可以采用有限差分法和有限元法,这两种方法是完全不同的,物理方程转化为离散方程的形式也完全不同,计算效果也有所区别。可以说没有数值计算方法,就没有科学计算。

在实际物理问题的求解过程中,往往最终都化解为对一个大规模或超大规模稀疏矩阵的处理,如线性方程组求解,本征值求解等。稀疏矩阵的含义是矩阵中不为零的元素(非零元)的数量很少,大部分元素的数值都为零。对于大规模或超大规模稀疏矩阵来说,非零元只占很少一部分,非零元的数量一般与矩阵的行数成正比,譬如,二维分析时,比例系数一般小于 10,也就是说,不管矩阵的维数多大,每行的非零元一般不多于 10 个。对于一个 $n\times n$ 的稀疏矩阵来说,非零元所占的比例一般小于 $10n/(n\times n)=10/n$,如果 $n=1000$,非零元所占比例小于 1%。对于这样的稀疏矩阵,如果采用常规的矩阵处理方法有两个缺点,一是要存贮很多零元素,浪

费存储空间，二是会出现很多 $0+0, 0-0, 0\times x, 0/x$ 等无效运算，降低计算效率。因此，必须针对稀疏矩阵的特点采用相应的存储技术和计算方法。目前，这方面的研究工作很多，也有很多有效的计算方法，这里不作更多介绍，只给出一些实用的方法。

1.1 非线性方程(组)的求解

实际的物理问题往往都不具有解析解，也就是说不能采用解析方法进行求解。对于这样的问题，只能采用一些近似方法来求解，得到的结果也是近似的。实际物理问题在进行近似求解时，常遇到的问题就是非线性方程和非线性方程组的求解。

1.1.1 非线性方程的求解—牛顿迭代法

对于一个不具有解析解的方程

$$f(x) = 0 \tag{1.1.1}$$

如果函数 $f(x)$ 具有一阶导数，依据任意一点 x_0 的函数值及其一阶导数值(不为零)，该函数可以写成近似的线性函数形式，即

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) \tag{1.1.2}$$

这种近似的意义在于用一个可以很容易求解的线性方程代替一个不能进行解析求解的非线性方程。近似的线性方程为

$$f(x_0) + f'(x_0)(x - x_0) = 0 \tag{1.1.3}$$

方程(1.1.3)的解可以很容易得到

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \tag{1.1.4}$$

这个解可以作为方程(1.1.1)的一个近似解，即向真实解靠近了一步，如果我们重复上述过程，就会逐渐向真实解逼近，当相邻两个近似解之间的误差小于预设精度，就可以认为得到问题的解，如

$$| x_{i+1} - x_i | < \varepsilon_a \tag{1.1.5a}$$

或

$$| x_{i+1} - x_i | < \varepsilon_a + \varepsilon_r * | x_{i+1} + x_i | /2 \tag{1.1.5b}$$

式中 ε_a 为绝对误差，ε_r 为相对误差。这就是牛顿迭代法的基本思想，如图 1.1.1(a)所示。一般情况下，利用牛顿迭代法都能很好地得到非线性方程的近似解，并且收敛速度很快。但也有出现不收敛的很特殊的情况，如图 1.1.1(b)。当然，这种情况是很少见的，并且可以通过改变初始值 x_0 的选择得到克服。

有关非线性方程的求解方法还有很多，如二分法，增值寻根法等，但用的最多的还是牛顿迭代法。

例题：用牛顿迭代法求解非线性方程 $x^2-1=0$。允许的绝对误差为 0.01。

解题：该题的精确解是 $x=\pm 1$。下面我们用牛顿迭代法来求解该方程的近似解。这个方程左边函数表示为 $f(x)=x^2-1$，相应的一阶导数为 $f'(x)=2x$。

首先任意选取一个初始值。显然选择 $x_0=0$ 是不可以的，因为对应该初值，函数的导数为

零。这里选择 $x_0=0.1$。对应该初值有 $f(x_0)=-0.99$，$f'(x_0)=0.2$。按照计算下一个近似值公式 $x=x_0-f(x_0)/f'(x_0)$，可以得到第一个近似值 $x_1=5.05$。误差 $|x_1-x_0|=4.95>0.01$，不满足要求，需要继续迭代。

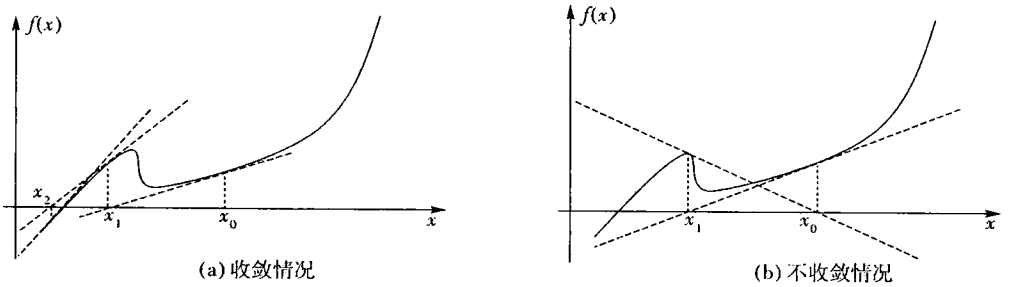

图 1.1.1　牛顿迭代法示意图

对应 x_1，有 $f(x_1)=24.5025$，$f'(x_1)=10.1$，下一个近似值 $x_2\cong 2.624$。误差 $|x_2-x_1|=2.426>0.01$，不满足要求，继续迭代过程。此后相继得到 $x_3\cong 1.312$，$x_4\cong 1.0371$，$x_5\cong 1.00066$，$x_6\cong 1.0$。误差 $|x_6-x_5|\cong 0.0007<0.01$，满足要求，该问题的一个近似解就是 $x\cong 1.0$。

如果初值选择 $x_0<0$，就可以得到另一个近似值。

1.1.2　非线性方程组的求解方法

很多物理问题都可以归结为求解一个非线性方程组。非线性方程组的求解方法和非线性方程类似。下面给出非线性方程组的牛顿迭代方法。对于非线性方程组

$$\begin{cases} f_1(x_1,x_2,\cdots,x_n)=0 \\ f_2(x_1,x_2,\cdots,x_n)=0 \\ \cdots\cdots\cdots\cdots \\ f_N(x_1,x_2,\cdots,x_n)=0 \end{cases} \tag{1.1.6}$$

选择一组初始值

$$X^0=(x_1^0,x_2^0,\cdots,x_n^0) \tag{1.1.7}$$

针对这组初始值，对方程组(1.1.6)左边的非线性函数进行线性化，并用这组线性函数作为原非线性函数的近似，有

$$\begin{aligned} &f_1(X^0)+\frac{\partial f_1}{\partial x_1}(X^0)(x_1-x_1^0)+\cdots+\frac{\partial f_1}{\partial x_n}(X^0)(x_n-x_n^0)=0 \\ &f_2(X^0)+\frac{\partial f_2}{\partial x_1}(X^0)(x_1-x_1^0)+\cdots+\frac{\partial f_2}{\partial x_n}(X^0)(x_n-x_n^0)=0 \\ &\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ &f_n(X^0)+\frac{\partial f_n}{\partial x_1}(X^0)(x_1-x_1^0)+\cdots+\frac{\partial f_n}{\partial x_n}(X^0)(x_n-x_n^0)=0 \end{aligned} \tag{1.1.8}$$

方程(1.1.8)为一个线性方程组，可以写成矩阵的形式

$$\boldsymbol{AX}=\boldsymbol{C} \tag{1.1.9}$$

式中

$$\boldsymbol{X}=(x_1,x_2,\cdots,x_N)^{\mathrm{T}}$$

$$\boldsymbol{A}=\begin{bmatrix}\dfrac{\partial f_1}{\partial x_1}(X^0) & \dfrac{\partial f_1}{\partial x_2}(X^0) & \cdots & \dfrac{\partial f_1}{\partial x_N}(X^0)\\ \dfrac{\partial f_2}{\partial x_1}(X^0) & \dfrac{\partial f_2}{\partial x_2}(X^0) & \cdots & \dfrac{\partial f_2}{\partial x_N}(X^0)\\ \dfrac{\partial f_N}{\partial x_1}(X^0) & \dfrac{\partial f_N}{\partial x_2}(X^0) & \cdots & \dfrac{\partial f_N}{\partial x_N}(X^0)\end{bmatrix}$$

$$\boldsymbol{C}=[f_1(X^0)f_2(X^0)\cdots f_N(x^0)]^{\mathrm{T}}+\boldsymbol{A}\cdot X^0$$

式中上角标符号“T”表示矩阵转置。方程(1.1.9)可以很容易用线性方程组的求解方法求解，得到一组近似解。重复上述过程，直到相邻两组近似解的误差满足误差要求，就认为得到了方程组的解。

例题：用迭代法求解非线性方程组

$$\begin{cases}x^2+y^2=2\\ x+y=0\end{cases}$$

允许的绝对误差为 0.01。

解题：该题有两组精确解，它们是 $x=1,y=-1$ 和 $x=-1,y=1$。

下面我们用迭代法来求解该方程组的近似解。这个方程组对应的两个函数是

$$\begin{cases}f_1(x,y)=x^2+y^2-2\\ f_2(x,y)=x+y\end{cases}$$

相应的一阶偏导数为

$$\begin{cases}\dfrac{\partial}{\partial x}f_1(x,y)=2x;\dfrac{\partial}{\partial y}f_1(x,y)=2y\\ \dfrac{\partial}{\partial x}f_2(x,y)=1;\dfrac{\partial}{\partial y}f_2(x,y)=1\end{cases}$$

首先任意选取一组初始值。显然选择 $x_0=y_0=0$ 是不可以的，因为对应该组初值，函数 $f_1(x,y)$的两个偏导数都为零。另外选择 $x_0=y_0$ 也不可以，因为此时 $f_1(x,y)$的两个偏导数相等，会得到无效的近似求解方程，以至于无法求解。这里选择 $x_0=0.1;y_0=0.2$。对应该初值有

$$\begin{cases}f_1(x_0,y_0)=-1.95\\ f_2(x_0,y_0)=0.3\end{cases}$$

$$\begin{cases}\dfrac{\partial}{\partial x}f_1(x_0,y_0)=0.2; & \dfrac{\partial}{\partial y}f_1(x_0,y_0)=0.4\\ \dfrac{\partial}{\partial x}f_2(x_0,y_0)=1; & \dfrac{\partial}{\partial y}f_2(x_0,y_0)=1\end{cases}$$

按照计算下一组近似值计算公式

$$\begin{cases}f_1(x_0,y_0)+\dfrac{\partial f_1(x_0,y_0)}{\partial x}(x-x_0)+\dfrac{\partial f_1(x_0,y_0)}{\partial y}(y-y_0)=0\\ f_2(x_0,y_0)+\dfrac{\partial f_2(x_0,y_0)}{\partial x}(x-x_0)+\dfrac{\partial f_2(x_0,y_0)}{\partial y}(y-y_0)=0\end{cases}$$

有

$$\begin{cases} x+2y=10.25 \\ x+y=0 \end{cases}$$

求解方程可以得到第一组近似值 $x_1=-10.25, y_1=10.25$。误差

$$\begin{cases} |x_1-x_0|=10.15>0.01 \\ |y_1-y_0|=10.05>0.01 \end{cases}$$

不满足要求，继续迭代过程。以 x_1, y_1 为初值，重复上述过程可得到后续的几个解为

$$x_2\cong 5.17378,\ y_2\cong -5.17378;\ x_3\cong 2.68755,\ y_3\cong -2.68755$$

$$x_4\cong 1.5298,\ y_4\cong -1.5298;\ x_5\cong 1.092,\ y_5\cong -1.092$$

$$x_6\cong 1.0039,\ y_6\cong -1.0039;\ x_7\cong 1.000008,\ y_7\cong -1.000008$$

误差

$$\begin{cases} |x_5-x_4|\cong 0.0039<0.01 \\ |y_5-y_4|\cong 0.0039<0.01 \end{cases}$$

满足要求，该问题的近似解就是 $x=-y\cong 1.000008$。

可以通过选择其它初值来得到另一组近似值。

1.2　稀疏矩阵的存储方法

稀疏矩阵的存储，就是只存储非零元，每个非零元需要存储三个基本信息：非零元的值，所在行的行号和所在列的列号。然而，这三个信息往往还不够，为了便于矩阵元素的快速查找和处理，还需要设置存储指针，使非零元之间相互关联。如何选择存储形式，不仅要考虑具体计算方法，还要综合考虑计算效率和存储量，一般地说，提高计算效率就需要更多的存储信息。

1.2.1　固定对角带状矩阵的存储法

固定对角带状矩阵是指非零元分布在主对角线及其两侧相对固定的位置上，如矩阵

$$\begin{Bmatrix} x & x & 0 & x & 0 & \cdots & 0 & 0 & 0 & 0 \\ x & x & x & 0 & x & \cdots & 0 & 0 & 0 & 0 \\ 0 & x & x & x & 0 & \cdots & 0 & 0 & 0 & 0 \\ x & 0 & x & x & x & \cdots & x & 0 & 0 & 0 \\ 0 & x & 0 & x & x & \cdots & 0 & x & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & x & 0 & \cdots & x & x & 0 & x \\ 0 & 0 & 0 & 0 & x & \cdots & x & x & x & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & x & x & x \\ 0 & 0 & 0 & 0 & 0 & \cdots & x & 0 & x & x \end{Bmatrix} \tag{1.2.1}$$

这里所说的“固定”的意思是指：非零元相对于主对角线，其相对位置是确定的，并且是有规律可循的，如主队角线右边第一个、右边第十个、左边第一个和左边第十个元素不为零。

对于这样的稀疏矩阵可以用一个二维数组来存储，由于每行的非零元的相对位置固定，可以不设置存储指针，具体存储方式如表 1.2.1 所示。这是一个按行存储的方案，二维数组的行

数就是稀疏矩阵的行数，二维数组的列数是所有行中每行非零元个数的最大值。这种存储方式的好处是最大限度地节省存储单元，适合于按行操作的计算方法，如线性方程组的迭代求解方法，本征值问题的直接迭代法。

行号	…	左边第 $L(>K)$ 个元素	左边第 K 个元素	主对角线元素	右边第 I 个元素	右边第 $J(>I)$ 个元素	…
1		—	—	x	x	x	
…	…	…	…	…	…	…	…
K		—	x	x	x	x	
…	…	…	…	…	…	…	…
L		x	x	x	x	x	
…	…	…	…	…	…	…	
$N-J$		x	x	x	x	x	
…	…	…	…	…	…	…	…
$N-I$		x	x	x	x	—	
…	…	…	…	…	…	…	…
N		x	x	x	—	—	

表 1.2.1 固定对角带状矩阵的存储方式

1.2.2 按行存储法

前面介绍了固定对角带状矩阵的存储方法，主要是针对每行的非零元的个数和位置相对固定，采用一个二维数组进行存储。这里再介绍一种简单的按行存储方法，该方法适用于每行的非零元个数和位置不固定的情况。由于每行的非零元个数和位置不固定，每个非零元需要两个存储单元，一个用于存储非零元的数值，一个用于存储非零元的列号，此外还需要设置一个行指针，用于指示每行第一个非零元的存储位置。这样我们需要三个一维数组，一个用于行指针，数组大小等于稀疏矩阵的行数加一，另外两个用于存储非零元的数值和列号，数组大小等于非零元个数。具体存储方式见图 1.2.1。这种存储方法适合于按行操作的计算方法，优点是存储单元少，缺点是不能很方便地进行非零元增加和删除等操作。

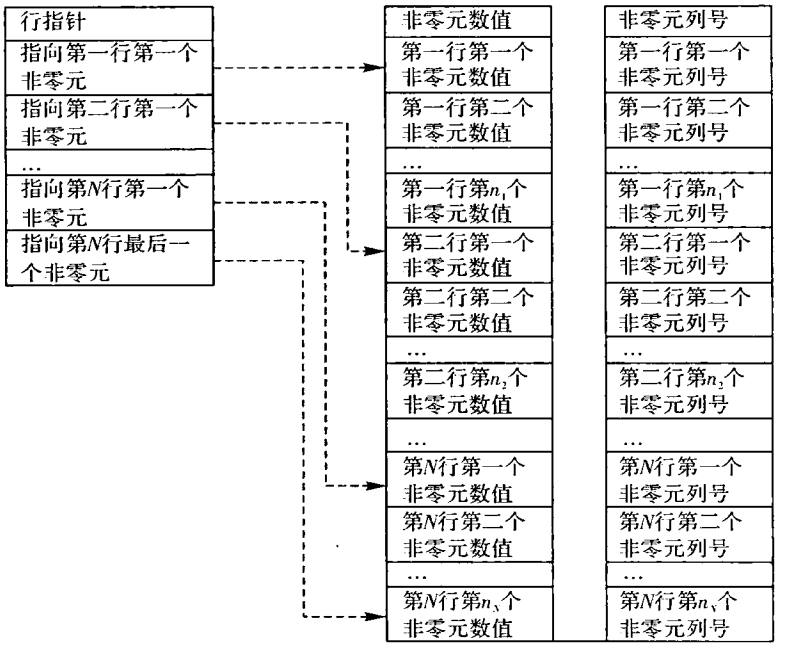

图 1.2.1　按行链表存储示意图

1.2.3　链表存储法

前面给出的两种存储方法的优点是存储量少，存储方法简单，对于很多实际问题都很实用，缺点是很难处理非零元的增减问题。譬如，在采用高斯法进行方程组求解时，就涉及非零元的删除和产生，因此，必须设计一种存储结构适应这样的操作。这里介绍一种稍微复杂一点的链表存储法。链表存储法是以每个非零元作为一个基本的存储单元。存储一个非零元需要七个信息：非零元的值，所在行的行号，所在列的列号，所在行中，其左边一个相邻非零元的存储位置，其右边一个相邻非零元的存储位置，所在列中，其上一个相邻非零元的存储位置，其下一个相邻非零元的存储位置，如图 1.2.2 所示。图 1.2.3 给出一个非零元和与其相关联的非零元之间的关联图。

非零元数值
所在行号
所在列号
所在行左一个相邻非零元的存储位置
所在行右一个相邻非零元的存储位置
所在列上一个相邻非零元的存储位置
所在列下一个相邻非零元的存储位置

图 1.2.2　一个非零元的存储结构

这种存储方法的优点是，只要找到一个非零元就可以找到所有非零元，并且可以非常容易地处理非零元的删除和增加问题。删除一个非零元，只需要在整个存储链网结构中删去这个环节，并把由此而断开的链路连接上即可。增加一个非零元，也只需要增加一个环节，并按所在行列的相邻元素的连接关系接入这个链网即可。为了更快速地进行非零元的查找，还可以设置两个指针数组，一个作为行指针，指向每一行的第一个非零元的存储位置，一个作为列指针，指向每一列的第一个非零元的存储位置。

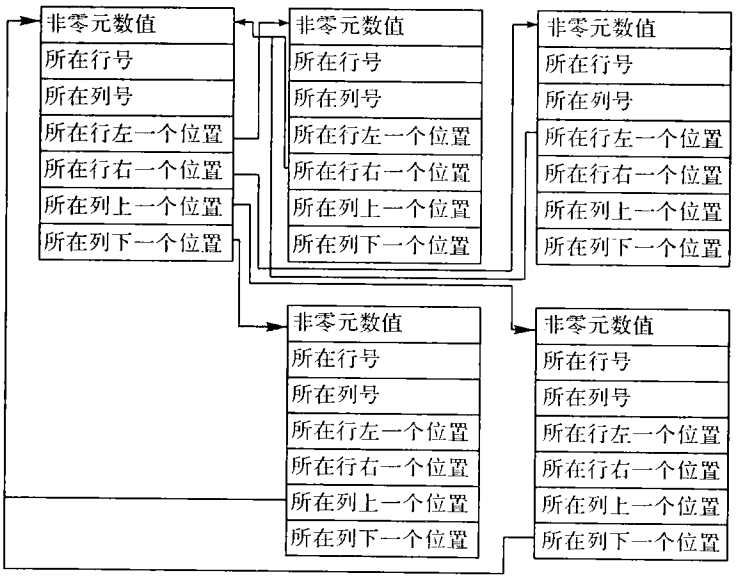

图 1.2.3 相邻非零元之间的关联图

1.3 线性方程组求解方法

在各种数值分析中，往往最终归解为求解一个线性代数方程组。线性方程组的求解方法很多，最经典的就是高斯消去法，以及派生出来的主元高斯消去法，此外还有 *LU* 分解法等。对于大规模稀疏矩阵来说，这些求解方法在求解过程中都会产生一定量的非零元，如果不采取一些必要的措施，有时会产生相当多的非零元，使针对稀疏矩阵而采用的存储方法的优越性丧失，相反会降低运算效率。

迭代法是比较有效的大规模稀疏矩阵求解方法，该方法的优点是即不增加非零元，也不删除非零元，还不改变非零元的位置和数值，在整个求解过程中对系数矩阵和常数向量不产生任何影响。但该方法的缺点是需要主对角线上的元素数值占优，即主对角线上的元素和同行中的其它元素相比，数值不能太小。对于双精度计算机运算不能差 16 个量级，单精度运算不能差 8 个量级，否则，会出现不收敛，或结果完全不正确。

1.3.1 高斯消去法

高斯消去法是线性方程组经典的直接求解方法，实际应用中经常被采用。

考虑一个 n 阶线性代数方程组

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= c_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= c_2 \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{mn}x_n &= c_N \end{aligned} \tag{1.3.1}$$

写成矩阵形式

$$\boldsymbol{A} \cdot \boldsymbol{X} = \boldsymbol{C} \tag{1.3.2}$$

式中

$$\boldsymbol{A}=\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ A_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{m} \end{pmatrix},\ \boldsymbol{X}=\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix},\ \boldsymbol{C}=\begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}$$

高斯消去过程一般分为两个操作过程。消元过程：经过若干次初等变换，把矩阵 $\boldsymbol{A}$ 化成上三角矩阵；回代过程：通过上三角矩阵把 x_n 到 x_1 依次求出来，最后得到方程的解。下面对高斯消去法的操作过程作简单介绍。

消元过程：从系数矩阵的第一行到第 n 行依次进行。首先，如有 $a_{11}\neq 0$，则按先后次序实施下面的计算过程可把第一行第一例的元素化为“1”，而第一列上的其它元素化为零。常数向量作为系数矩阵的第 $n+1$ 列实施同样的操作：

$$a_{1j}^{1}=a_{1j}^{0}/a_{11}^{0},\ c_1^1=c_1^0/a_{11}^0$$
$$a_{ij}^{1}=a_{ij}^{0}-a_{i1}^{0}a_{1j}^{1},\ c_i^1=c_i^0-a_{i1}^0c_1^1$$
$$i=2,\cdots,n,\ j=1,\cdots,n$$

上述操作过程先后次序不能颠倒，上角标“0”表示原始矩阵及常数向量的元素，“1”表示经第一步消元得到的新矩阵及常数向量的元素。以下用上角标“k”表示第 k 步消元得到的新矩阵和常数向量的元素。如果还有 $a_{22}^1\neq 0, a_{33}^2\neq 0, a_{k-1,k-1}^{k-2}\neq 0$，则可按上述过程进行第 $2,3,\cdots,k-1$ 步消元。一般地，若有 $a_{kk}^{k-1}\neq 0$，则可进行第 k 步消元，所施行的操作过程如下：

$$a_{kj}^{k}=a_{kj}^{k-1}/a_{kk}^{k-1},\ c_k^k=c_k^{k-1}/a_{kk}^{k-1}$$
$$a_{ij}^{k}=a_{ij}^{k-1}-a_{ik}^{k-1}a_{kj}^{k},\ c_i^k=c_i^{k-1}-a_{ik}^{k-1}c_k^k$$
$$i=k+1,\cdots,n,\ j=k,\cdots,n$$

如果 $a_{kk}^{k-1}\neq 0(k=1,\cdots,n)$ 总成立，则高斯消去法的每步消元过程都是可实现的，这样，经过最多 n 步消元就可把矩阵 $\boldsymbol{A}$ 化为单位上三角矩阵。消元过程结束后，方程化为

$$\begin{pmatrix} 1 & a_{12}^n & \cdots & a_{1n}^n \\ 0 & 1 & \cdots & a_{2n}^n \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}=\begin{pmatrix} c_1^n \\ c_2^n \\ \vdots \\ c_n^n \end{pmatrix} \tag{1.3.3}$$

回代过程：由(1.3.3)式可直接得到 $x_n=c_n^n$，把 x_n 的值代入第 $n-1$ 个方程，即可得到 $x_{n-1}=c_{n-1}^n-a_{n-1,n}^n x_n$。然后再把 x_n, x_{n-1} 的值代入第 $n-2$ 个方程，即可得到 x_{n-2} 的值。按此过程，可很容易地把 x_n 到 x_1 依次求出来。

应特别注意，高斯消去法能够得以实现的条件是 $a_{kk}^{k-1}\neq 0(k=1,\cdots,n)$ 总成立，否则消元过程就会失败。还要看到，即使 $a_{kk}^{k-1}\neq 0$ 成立，如果 a_{kk}^{k-1} 的绝对值与同行中其它元素比相对很小，则得不到正确解。如，对于方程组

$$\begin{cases} x_2=1 \\ x_1+x_2=2 \end{cases}$$

消元过程因有“0”作除数而不能进行下去。而这个方程组是有唯一解的，即 $x_1=x_2=1$。再譬如方程组