超大规模稀疏矩阵计算方法

刘长学编著

CHAODA GUIMU XISHUJUZHEN JISUAN FANGFA

上海科学技术出版社

超大规模稀疏矩阵计算方法

刘 长 学

上海科学技术出版社

超大规模稀疏矩阵计算方法

刘 长 学

上海科学技术出版社出版

（上海瑞金二路 450 号）

新华书店上海发行所发行　　常熟文化印刷厂印刷

开本 787×1092　1/16　印张 22.25　字数 523,000

1991 年 9 月第 1 版　　1991 年 9 月第 1 次印刷

印数：1—5,000

ISBN 7-5323-2397-8/O·148

定价：10.30 元

序　言

随着计算机功能和速度的增长，大规模矩阵计算得到了很大的发展．60年代国际上产生了5000阶、10000阶或更高阶的矩阵计算问题，70年代在测绘领域中提出了400000×200000的超定方程和2500000×400000的超定方程，到了80年代又提出了具有6000000×400000的超定方程，其系数矩阵具有2.4×10^{12}个元素．在特征值计算中，矩阵规模同样有类似的增长趋势，其中大的矩阵有2000、4900和12000阶，可以断言随着科学技术的发展，矩阵的阶将还会不断增加，然而这仅是问题的一个方面．另一方面，矩阵规模的增大，存贮量和工作量都有所增加，矩阵问题的计算代价将迅速增长．虽然计算技术有较快的发展，但还是跟不上实际的需要．因此，对于在小型计算机上或微型计算机上出现的大型或超大型存贮量、运算工作量和计算解精度等问题，已成为计算数学工作者和工程技术人员的重要研究课题，也是本书的主要内容．

稀疏矩阵技术这门科学，是理论发展、数值经验和实际考虑的有机结合．它不仅在广大应用领域中是一种重要的计算工具，而且它的本身对计算机软件发展也是一个有价值的贡献．在近二十年的发展时间内，对各类矩阵问题设计出许多近似优化算法，在许多领域中应用得很活跃，且应用范围将继续不断地扩大，稀疏矩阵计算技术也将会不断地完善和发展．

本书材料主要取自作者70年代以来的稀疏矩阵方面的若干篇论文，以及"稀疏矩阵和直接解法"讲稿资料，并吸收了国际上的一些新近成果．

全书共分十章．第一章介绍稀疏矩阵概念、基本类型、应用范围和稀疏矩阵各种存贮技术；第二章介绍稀疏矩阵方程各种有效解法、求逆方法和提高精度方法；第三章介绍稀疏矩阵特征问题的计算方法；第四章介绍稀疏矩阵计算过程中的误差分析；第五章介绍大型变带状矩阵方程和大型阶梯矩阵方程的有效算法；第六章介绍稀疏单重子结构矩阵方程算法；第七章介绍超大型稀疏二重子结构矩阵方程算法；第八章讨论稀疏矩阵的求逆算法；第九章以图论为基础，详细讨论对称与非对称稀疏矩阵排序算法和介绍产生稀疏矩阵有利结构的各种方法；第十章结合大地测量平差计算实例，介绍超大型稀疏最小二乘平差问题的几种解法．

本书对研究稀疏矩阵结构，高效率的算法和高质量应用软件，均有良好的帮助和参考价值．书中算法是经过精心描绘，详细讨论和上机测试，实用性较强．

本书的初稿是1980年1月到1982年3月在上海计算所完成的，1989年1月到1990年3月在江苏省盐城市微机应用研究所对原稿的有关章节作了较大的增删和文字修改，并最后定稿．在本书的写作过程中，曾得到盐城市科学技术委员会、盐城市微机应用研究所领导和同志们的关心和支持，在此表示衷心的感谢．

书中错误之处在所难免，敬请读者批评指正．

刘长学

1990年3月于盐城

内 容 提 要

本书是超大规模稀疏矩阵求解和求逆的一部专著，第一章到第八章介绍稀疏矩阵类型、存贮方案、基本解法、特征值计算和误差分析，着重介绍大型带状矩阵，超大规模单重与多重分块结构矩阵的多种解法和求逆等中心问题，详细讨论矩阵的结构分析、公式推导和计算方案，并精心描绘算法的基本技巧和高难度技术问题。在存贮量、运算量和数值解的精度方面，均能获得近似最佳结果。第九章介绍产生各种稀疏矩阵排序技术。最后，第十章介绍超大型最小二乘大地平差问题，本章不仅能为测绘领域提供全套大地平差技术，而且作为一个例子，介绍稀疏矩阵技术应用的全过程，可供其他行业有关技术人员应用参考。

本书可供计算数学、应用数学、计算机科学、数值分析工作者和高等院校有关专业师生，以及广大科研和工程技术人员参考。

目　　录

第一章

稀疏矩阵、基本类型和存贮方式

§1.1 稀疏矩阵及其应用领域

矩阵是数字有规律的排列，稀疏矩阵通常的概念是，当某一矩阵有许多元素等于零时，则称其为稀疏的．也有用极限概念来描述的，n 阶矩阵是稀疏的，当它有 $o(n)$ 非零元素．即对充分大的 n，非零元素与 n 成比例．这一定义在理论上是有效的，例如试图估价算法的渐近行为，对于 n 阶稀疏矩阵，非零元素的数字必须是 $n^{1+\gamma}$，其中 $\gamma<1$．对于电学问题，γ 的标准值为 0.2，对于和网络相联的带状矩阵，γ 的标准值为 0.5．1000 阶矩阵 $\gamma=0.9$ 时，有 501187 个非零元素．

一个稀疏矩阵，其数字可以不规则排列，但和它们相联系的编号位置是规则的，因此，依次规则排列可以作为研制算法的参考依据．考虑下列 8 阶稀疏矩阵：

$$\begin{array}{cc}
 & \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} &
\begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 1 & -2 & 0 & 1 & 0 \\
0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\
0 & 0 & 0 & -2 & 0 & 0 & 0 & 1
\end{bmatrix}
\end{array}$$

这个矩阵仅有 16 个元素，但占了 64 个单元．考虑到这些零元素充满那些和非零元素不相干的位置，是完全不必要的，我们应当充分利用那么许多零元素的存在．严格地讲，稀疏矩阵不应当完全看作矩阵，更确切地讲稀疏矩阵应看为一个图：每个方程与未知数是和网络顶点相联系的，并且每个系数和某一条边相联系，这就是为什么图论在稀疏矩阵技术中扮演一个重要角色的原因，它也是稀疏矩阵技术自身的起源．稀疏矩阵是无规则的数字集合，不能象稠密矩阵那样用简单途径，把它们表示在计算机的存贮器中．如果要存贮方程组的系数数值，还必须要存贮和方程的系数相应的行与列的数值．用稀疏矩阵的行话来说，存放非零元素值，还要存贮索引信息，告诉每一个非零元值在排列规则中的位址，这些附加信息构成总的开销，是避免存贮零元素所付出的代价．

稀疏矩阵算法设计途径是处理被存贮的非零元素，并且可以利用前面知道的非零元素的位址，用来避免那些和零有关的加法或乘法运算．在计算机算法执行期间，所完成的运算次数与非零元素总数成比例，而不是和整个矩阵元素总数成比例．注意到它没有精确存贮包含零元素的全部元素，于是用一个 IF 语句方式，能够跳过包含零元素的运算，在 n 阶矩阵中构造二次方的算法，IF 语句不需要执行 n^2 次或更多次．当稀疏矩阵每行非零元素的

个数为常数时，一个好的算法的阶数，在一般情况下和n的阶一样低。

描述稀疏矩阵所给出的信息，即可靠地给出矩阵的稀疏性。我们问是否存在一种算法，使用该算法作用在此稀疏矩阵上，能够产生较少的非零元素，用来改善矩阵的稀疏性。在一般情况下回答是肯定的，如高斯消去法和侨勒斯基分解及其改进方案等，均对稀疏矩阵是有效的方法。但是高斯消去法引进了新的非零元素，削弱了原始矩阵的稀疏性，所产生的矩阵较原始矩阵有较少的稀疏性，但填充幅度取决于主元选取的次序。一个好的稀疏矩阵算法，应该保持稀疏性，保持填充量极小化。

众所周知，稀疏矩阵技术应用极为广泛，需要列出它的全部应用领域是一件困难的事，我们把它分为几大类，每类再分成一些具体的应用领域，其中某些应用领域对稀疏矩阵研究的影响和稀疏矩阵的研究对这些领域的本身的影响一样深远：① 数值分析——特征值计算、积分方程、线性和非线性方程、线性规划、常微分方程、偏微分方程；② 数学——组合学、计算理论、动态规划、模型建立、运筹学与统计；③ 工程技术——网络的补偿法、计算机辅助网络设计、电子和电气网络、电力网、频域分析、水力网、结构分析；④ 应用科学——化学工程、数据处理、遗传学、摄影测绘学、大地测量学、水库模拟、勘测、水的运输现象；⑤ 理论科学——化学、反应动力学、结晶学、物理；⑥ 人文科学——汽车的发展、商业管理、经济学、学校的排课、社会科学、城市规划。

§1.2 稀疏矩阵基本类型

在本节中就其经常出现的稀疏矩阵基本类型给出如下：

1. 上三角阵、下三角阵、对角阵和单位阵

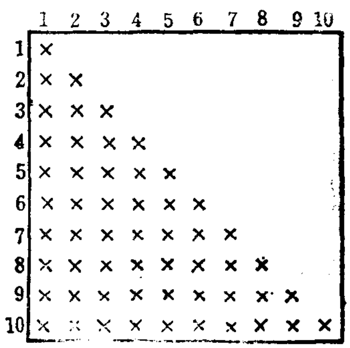

图 1.1 下三角阵

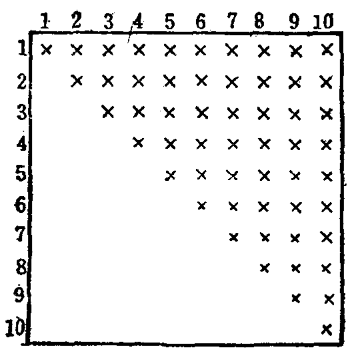

图 1.2 上三角阵

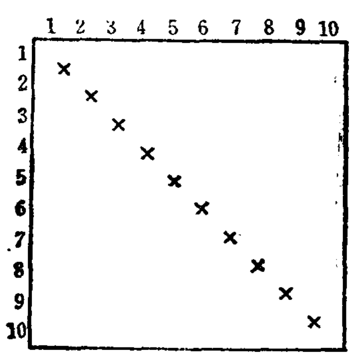

图 1.3 对角阵

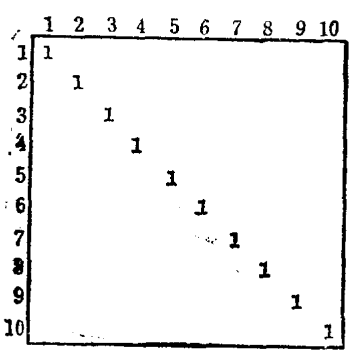

图 1.4 单位阵

2. 对称等带状矩阵和非对称等带状矩阵

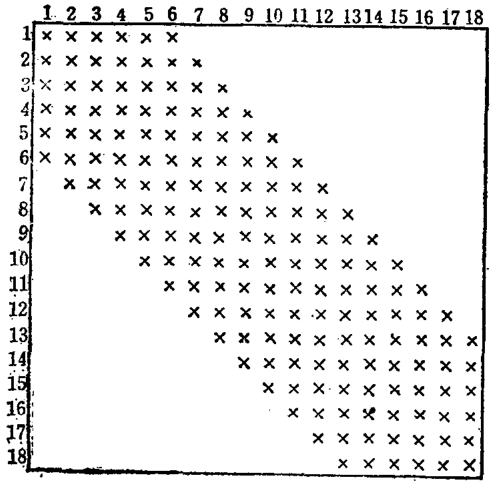

图 1.5 对称等带状矩阵

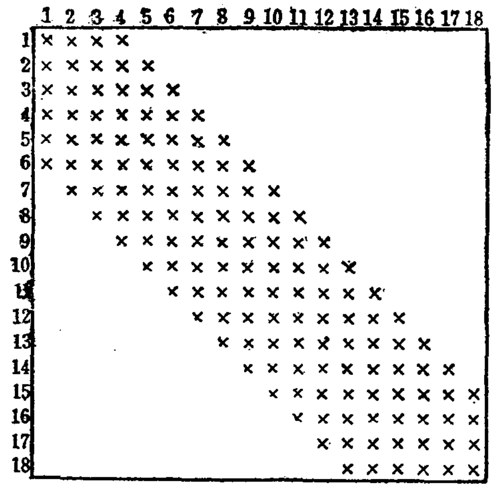

图 1.6 非对称等带状矩阵

3. 对称等带状循环矩阵和非对称等带状循环矩阵

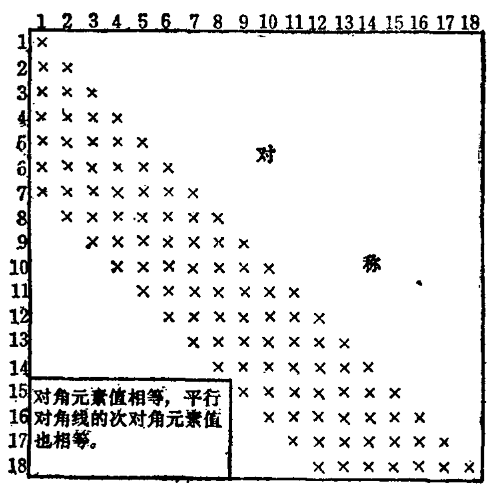

图 1.7 对称等带状循环矩阵

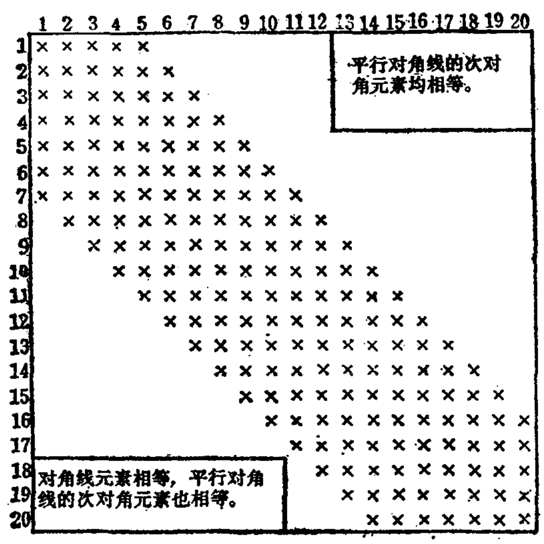

图 1.8 非对称等带状循环矩阵

4．对称变带状矩阵和非对称变带状矩阵

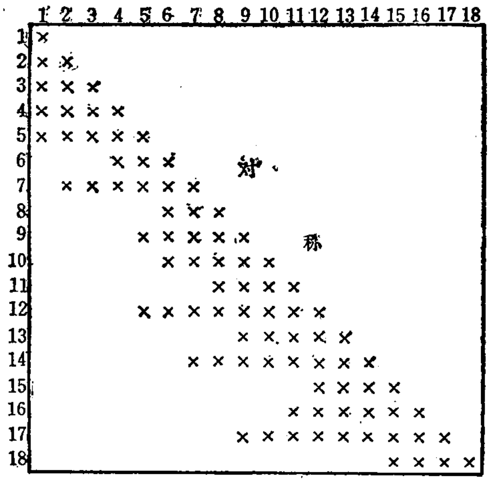

图 1.9 对称变带状矩阵

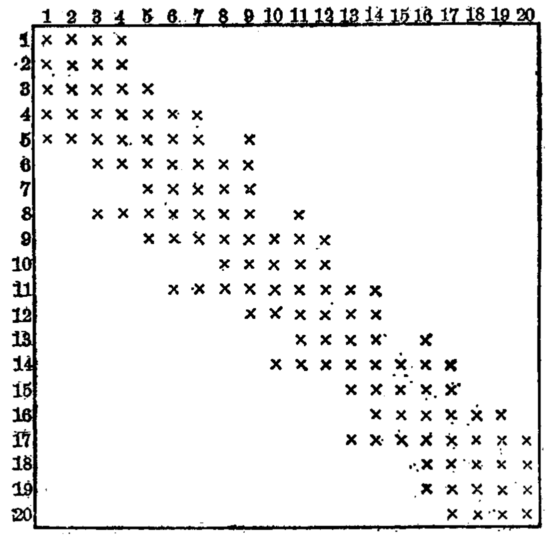

图 1.10 非对称变带状矩阵

5．对称阶梯矩阵和非对称阶梯矩阵

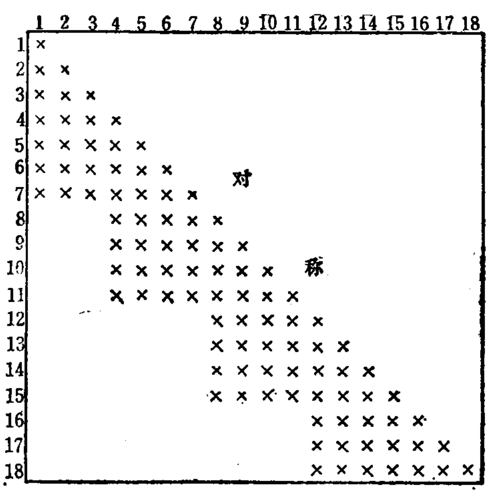

图 1.11 对称阶梯矩阵

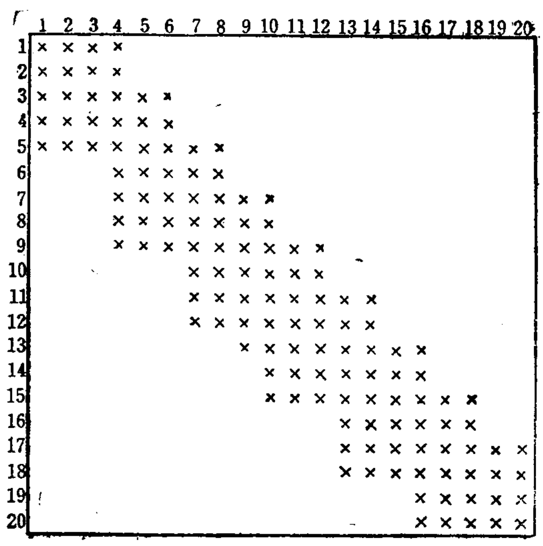

图 1.12 非对称阶梯矩阵

6. 对称稀疏矩阵和非对称稀疏矩阵

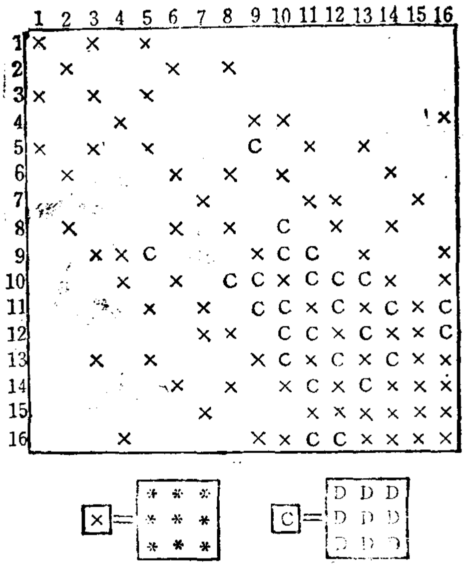

图 1.13 对称稀疏矩阵

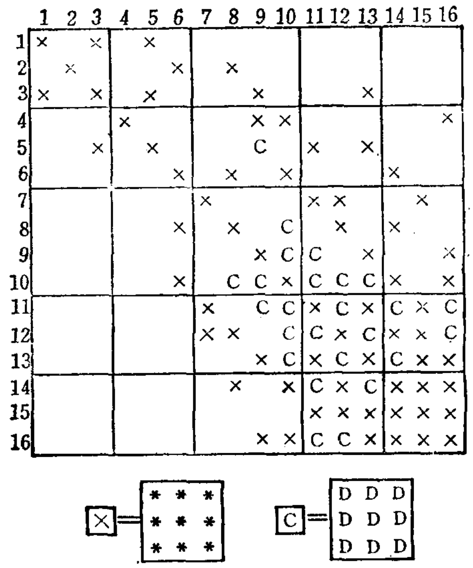

图 1.14 非对称稀疏矩阵

7. 稀疏对称单重子结构矩阵

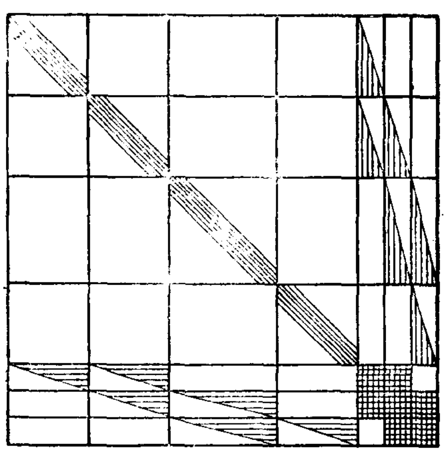

图 1.15 稀疏对称单重子结构矩阵(一)

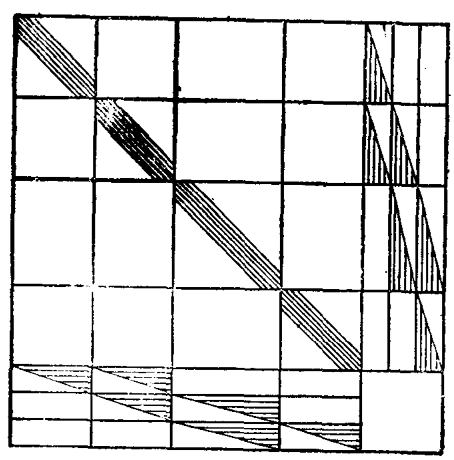

图 1.16 稀疏对称单重子结构矩阵(二)

8．稀疏对称二重子结构矩阵和稀疏对称增广矩阵

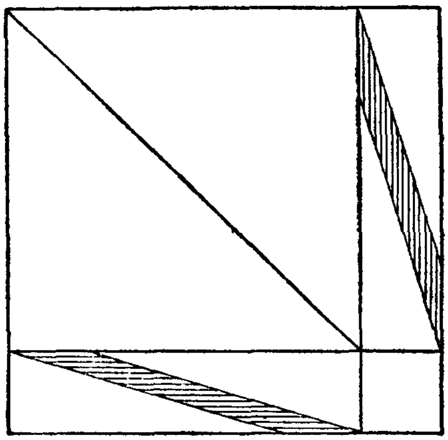

图 1.17　稀疏对称增广矩阵

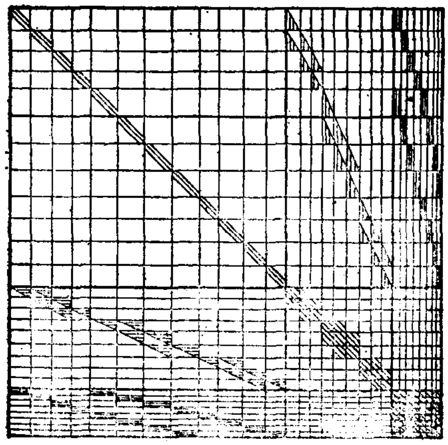

图 1.18　稀疏对称二重子结构矩阵

9．稀疏对称反向单重和二重子结构矩阵

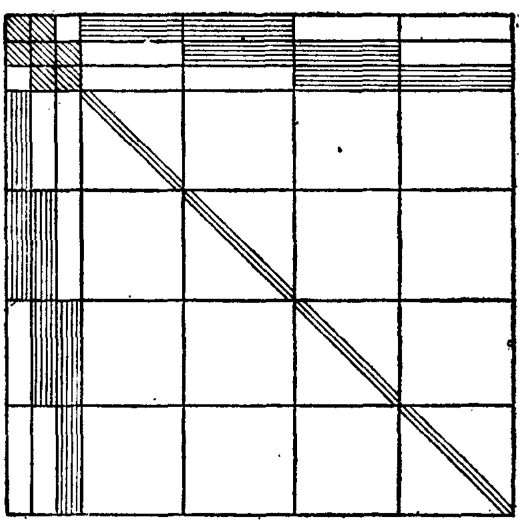

图 1.19　稀疏对称反向单重子结构矩阵

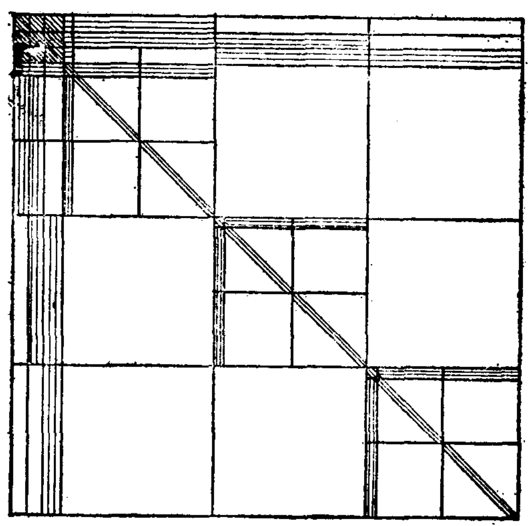

图 1.20　稀疏对称反向二重子结构矩阵

10．稀疏长方带状矩阵

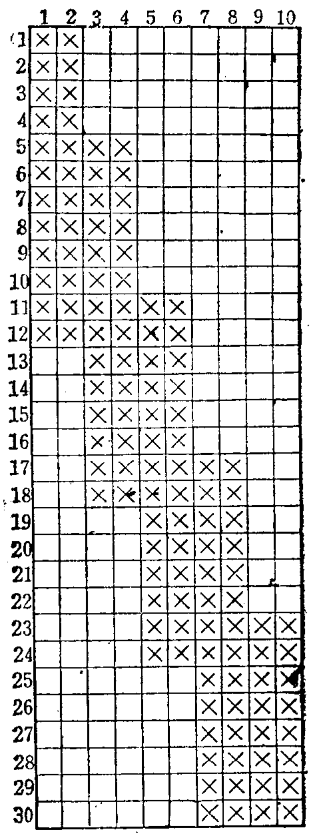

图 1.21 长方带状矩阵

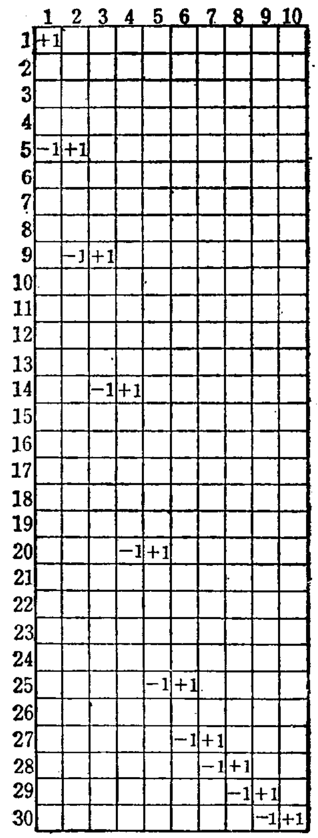

图 1.22 极稀疏长方带状矩阵

11．三对角线矩阵和上海森堡矩阵

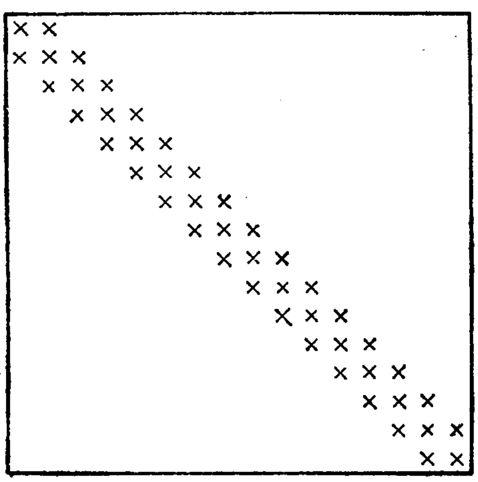

图 1.23 三对角线矩阵

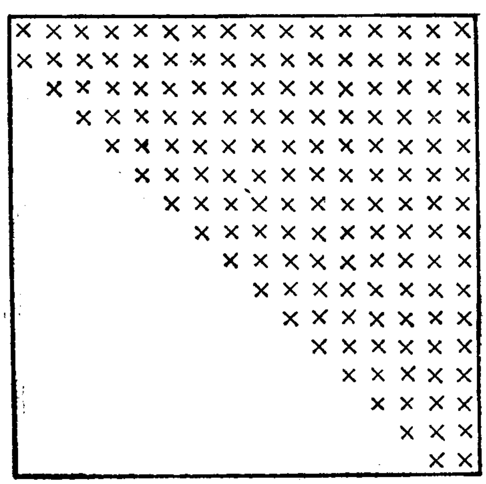

图 1.24 上海森保矩阵

§1.3　覆盖存贮法

当用高斯消去法或三角因子分解法，对给定矩阵按行分解时，对于计算的当前元素而言，只用到它的左边及其上面对应行中有关元素，而不涉及它的右边和下边的矩阵元素。也就是说在矩阵元素进行分解时，具有严格的规律性，一旦矩阵元素被分解之后，它的数值和位址一直保持不变。此外，利用计算机工作单元能向其自身叠加并赋值的特性，可实现将分解好了的矩阵元素，仍存放在它本身所存放单元之中，而不需重新开设新的存贮单元，称之为*覆盖存贮法*。例如用公式

$$l_{ij}=a_{ij}-\sum_{p=2}^{j-1} l_{ip}l_{jp}/l_{pp} \qquad (i,\ j=1,\ 2,\ \cdots\cdots n)$$

对给定的矩阵分解时，其元素 l_{ij} 可以存贮在元素 a_{ij} 的存贮单元中。由此可见使用覆盖存贮法，可以节省一倍存贮单元。如果和外部设备配合使用时，还可产生一个多重覆盖存贮技术，能更充分地发挥内存贮器单元的作用。

§1.4　拼凑存贮法

在有限元素法和测量平差等实际应用中，常常需要存贮大量的正整常数数据，而且该数据一般为1～5位整数。为减少存贮量，根据机器的尾数位数，按照一定的规律性，将二个或多个正整数拼凑起来作为一个实数，待需要时再拆成二个或多个正整数，这就是称为*拼凑存贮法*。下面我们以二个正整数拼拆方法为例，说明拼凑存贮法的过程。

设计算机尾数有 t 位(十进制)，二个整数的位数分别用 t_1 和 t_2 表示，二个正整数的一般形式为：

$$\begin{cases} m=m_1m_2\cdots m_{t_1} \\ n=n_1n_2\cdots n_{t_2} \end{cases} \tag{1.1}$$

式中 $m_1,\ m_2,\ \cdots,\ m_{t_1}$ 和 $n_1,\ n_2,\ \cdots,\ n_{t_2}$ 为0～9的数字，则可以人为的将整数 m 和 n 拼凑为一个实数，分别放在实数 D 的整数和小数部分，即

$$D=m\cdot n=m_1m_2\cdots m_{t_1}\cdot n_1n_2\cdots n_{t_2} \tag{1.2}$$

式中 $t_1+t_2\leqslant t$。

显而易见，用(1.1)式存贮整数 m 和 n 需要用二个单元，而用(1.2)式存贮方式仅需用一个存贮单元，若有一大批整数需要存贮，并且二个以上整数拼凑为一个实数存放时，其效果尤为显著，经过这样处理可以节省 $\frac{1}{2}$ 以上的存贮单元，但需要花费少量的拆开数据D的计算工作量。

下面讨论将实数D拆开成(1.1)式所示的二个整数 m，n 的方法：

$$\begin{cases} m=\mathrm{INTEGER}(D)=m_1m_2\cdots m_{t_1} \\ n=\mathrm{INTEGER}(10^{t_2}*(D-m))=n_1n_2\cdots n_{t_2} \end{cases} \tag{1.3}$$

由(1.3)式可见，仅需经过简单的运算，便可获得整数 m 和 n。

例如，给出十个整数于表1.1中，拼凑而得到实数如表1.2所示：

位址＝1	2	3	4	5	6	7	8	9	10
LIST＝123	234	345	456	567	678	789	891	912	101