稀疏矩阵存储技术

张永杰，孙秦

（西北工业大学 航空学院，西安 710072）

摘 要：在科学与工程计算领域，有许多问题都最终归结为求解稀疏线性方程组；其稀疏矩阵中只有少量元素不为零，为了节省计算机的存储空间，加快存取运算速度，开展稀疏矩阵存储技术的研究是十分必要的。本文从基本的矩阵存储技术出发，介绍了一些常用的稀疏矩阵存储方法，比较了它们的优缺点，并给出了它们的适用条件。期望能够对稀疏线性方程组的高效求解提供一些有益帮助。

关键词：稀疏线性方程组；稀疏矩阵；稀疏存储技术

中图分类号：O241 **文献标识码**：A **文章编号**：1672-9870（2006）03-0038-04

Sparse Storage Technique for Sparse Matrix

ZHANG Yongjie, SUN Qin

(*School of Aeronautics, Northwestern Polytechnical University, Xi'an* 710072)

Abstract: In the computational field of science and engineering, many problems finally go to solve sparse linear equations and the sparse matrix has a few nonzero elements. For saving storage of computer and quickening elements' operation, it is very necessary to study storage technique for sparse matrix. Based on basal storage technique for matrix, this paper introduces some common storage methods for sparse matrix. After analyzing advantages and disadvantages of the methods, we show applied conditions for them. It is anticipant to provide some availability ways about effective solution for sparse linear equations.

Key words: sparse linear equation group; sparse matrix; sparse storage technique

1 稀疏矩阵

如果一个矩阵中只有少量的元素不为零，则称这样的矩阵为稀疏矩阵。在科学工程计算领域，有许多问题都归结为求解稀疏线性方程组。如果用稠密矩阵方法求解稀疏线性方程组，存贮量和计算量都将十分大。而在实际应用中，经常需要求解几十万直至几亿阶的稀疏线性方程组，所以对稀疏线性方程组进行稀疏化存储是十分必要的。

传统的稀疏矩阵存储技术通常针对直接法，因为传统迭代法中只需要保存稀疏矩阵和少量向量，同时计算时间主要表现为稀疏矩阵与稠密向量的乘法操作。近年来，迭代法逐渐与直接法相互融合，在一些投影迭代法中，利用对直接法进行近似，在迭代前作对稀疏矩阵预处理，可以改善迭代矩阵的条件数，从而减少迭代次数，所以稀疏矩阵存储技术已成为直接法与迭代法中都要用到的关键技术。

对一个矩阵 A，称 $P(A)=\{(i,j);a_{ij}\neq 0\}$ 为 A 的非零元结构，如果对任意 $(i,j)\in P$ 都有 $(j,i)\in P$，则称 P 对称非零元结构，且称 A 是具有对称结构的稀疏矩阵。稀疏矩阵有两种基本类型：有规则结构的稀疏矩阵与无规则结构的稀疏矩阵。例如带状矩阵就是一种十分典型的具有规则结构的稀疏矩阵，通常对应于一维问题离散得到的线性方程组，同时也经常在采用交替方向进行求解的方法中用到。块三对角矩阵是另一类具有规则结构的稀疏矩

收稿日期：2006-03-22
基金项目：国家自然科学基金资助（10477018）
作者简介：张永杰（1979-），男，博士研究生，研究方向，飞行器设计，E-mail：zyj19191@tom.com
通信作者：孙秦（1955-），男，博士生导师。

阵。然而我们遇到更多则是无规则结构的稀疏矩阵，有限元方法得到的线性方程组多具有这样的性质，所以这类矩阵的存储更具有挑战性。

本文的主要研究内容就是，从稀疏矩阵的一些特性出发，介绍了一些常用的稀疏矩阵存储方法，比较了它们的优缺点，给出了适用范围。

2　稀疏矩阵分解中的非零元填入

在对矩阵 $A=(a_{ij})_{n\times n}$ 进行 LU 分解时，单位下三角矩阵 $L=(l_{ij})_{n\times n}$ 与上三角矩阵 $U=(u_{ij})_{n\times n}$ 第 i 行中的元素通过公式

$$l_{ij}=(a_{ij}-\sum_{k=1}^{j-1}l_{ik}u_{kj})/u_{jj},j=1,2,\cdots,i-1 \quad (1)$$

$$u_{ij}=a_{ij}-\sum_{k=1}^{i=1}l_{ik}u_{kj},j=i,i+1,\cdots,n \quad (2)$$

计算得到。由（1）可以看出，即使 $a_{ij}=0$，只要存在 $1\leqslant k\leqslant j-1$，使 l_{ik} 和 u_{kj} 同时不为 0，则 l_{ij} 一般不为 0。这就是说，在对稀疏矩阵进行分解时，即使原矩阵中某些位置处的元素为 0，在分解因子中相应的位置上的元素也可能不为 0，这种新增的非零元素称为分解中引入的填入元。

由上述分析可知，如要用矩阵分解求解稀疏线性方程组，必须采用适当的存贮方式，以便于存贮这些填入元。对于迭代法，矩阵向量相乘的操作，以及预条件子的构造与应用，也都与稀疏矩阵的存贮技术密切相关。

3　稀疏矩阵存储方法

3.1　对角线存贮法

在许多实际问题中，经常遇到这样的矩阵，其每行中只有几个元素不为零，而且这些非零元素分布在与对角线平行的几条直线上，例如：

$$A=\begin{bmatrix}T_1 & -I & \\ -I & T_2 & -2I \\ & -2I & T_3\end{bmatrix} \quad (3)$$

其中

$$T_1=\begin{bmatrix}4 & -1 & \\ -1 & 5 & -2 \\ & -2 & 6\end{bmatrix},$$

$$T_2=\begin{bmatrix}7 & -1 & \\ -1 & 8 & -2 \\ & -2 & 9\end{bmatrix},$$

$$T_3=\begin{bmatrix}6 & -1 & \\ -1 & 7 & -1 \\ & -1 & 8\end{bmatrix}$$

对这种矩阵，只需要存贮矩阵中处于这几条直线上的元素即可。为了寻址方便引入一个整型数组，其长度为各行中非零元个数的最大值，其中每个整数指明非零元所在位置偏离对角线的偏移量。

设矩阵 $A\in C^{n\times n}$ 中仅有 $\{a_{i,j+l_j};\ 1\leqslant j\leqslant m,\ 1\leqslant i\leqslant n\}$ 中的元素不为零，则可用矩阵 $B=(b_{ij})\in C^{n\times m}$ 与整型向量 $\gamma=(l_1,l_2,\cdots,l_m)^T$ 存贮 A，其中

$$b_{ij}=\begin{cases}a_{i,i+l_j},1\leqslant i+l_j\leqslant n\\ 0,otherwise\end{cases}$$

以矩阵（3）为例，则

$$B=\begin{bmatrix}0 & 0 & 4 & -1 & -1\\ 0 & -1 & 5 & -2 & -1\\ 0 & -2 & 6 & 0 & -1\\ -1 & 0 & 7 & -1 & -2\\ -1 & -1 & 8 & -2 & -2\\ -1 & -2 & 9 & 0 & -2\\ -2 & 0 & 6 & -1 & 0\\ -2 & -1 & 7 & -1 & 0\\ -2 & -1 & 8 & 0 & 0\end{bmatrix},\gamma=\begin{bmatrix}-3\\ -1\\ 0\\ 1\\ 3\end{bmatrix}$$

当然对对称矩阵，可以只存贮稀疏矩阵的上三角或是下三角部分。

在以上存贮方式下，当需要访问 A 的元素 a_{ij} 时，先在向量 γ 中查找，看是否存在 k，使得 $l_k=j-i$，如果存在这样的 k，则将 b_{ik} 赋给 a_{ij}，否则 $a_{ij}=0$。

当 $l_1,l_2,\cdots,l_m$ 为连续整数时，这种方法称为等带宽存贮法，此时，不需要存贮整型向量 γ，只需要记下 γ 的最大分量、最小分量即可。假设利用等带宽存贮法将矩阵 A 存贮到 B 中，且当 $m_1\leqslant j-i\leqslant m_2$ 时，a_{ij} 才不为 0。如果需要访问 a_{ij}，则计算 $l=j-i$，如果 $m_1<l\leqslant m_2$，则 $a_{ij}=b_{i,l-m_1}$，否则 $a_{ij}=0$。

如果对于带状稀疏矩阵 A 进行 LU 分解，不会在原带状结构之外引入填入元，所以利用等带宽存贮法将十分合适。

3.2　对称矩阵的变带宽存贮法

设 A 是一个 n 阶对称矩阵，只存贮其下三角部分，设 m_i 为第 i 行的局部带宽，则可以对 A 的下三角部分元素进行逐行存贮，且各行中的元素从最左边的非零元开始直到对角元，这样可以将这些元素存贮到一个向量 x 中。为了访问 A 的元素，需要再引入一个 n 维的整形数组 y，其中第 i 个元素指明 A 的第 i 个对角元存贮在 x 中的位置。以矩阵

(3) 为例，按这种存贮方法，有

$x=(4,-1,5,-2,6,-1,0,0,7,-1,0,-1,8,-1,0,-2,9,-2,0,0,6,-2,0,-1,7,-2,0,-1,8)^T$

$y=(1,3,5,7,9,13,17,21,25,29)^T$

显然，当矩阵中远离对角线处存在非零元时，变带宽存贮法比等带宽存贮法更节省存贮空间。

当需要访问 A 的元素 a_{ij} 时，计算 $\theta=y_i-(j-i)$，如果 $\theta>y_{i-1}$，则 $a_{ij}=x_l$，否则 $a_{ij}=0$。

对 n 阶对称矩阵 A，设下三角部分第 $1\leqslant i\leqslant n$ 行中，只有当 $j>i-m_i$ 时，才有 $a_{ij}\neq 0$。如果对 A 作 LDL^T 分解，其中 L 与 D 分别为单位下三角矩阵与对角矩阵，则

$$d_i=a_{ii}-\sum_{k=1}^{i-1}l_{ik}^2d_k,$$

$$l_{ij}=(a_{ij}-\sum_{k=1}^{j-1}l_{ik}l_{jk}d_k)/d_j,j=1,2,\cdots,i-1$$

不妨设 $i>m_i$，则在第 i 行中，$l_{i1}=a_{i1}=0$，从而依次可知对 $j=2,3,i-m_i$，都有 $l_{ij}=0$。这说明，对对称矩阵进行 LDL^T 分解时，将不会在其包络之处引入填入元，所以采用变带宽存贮法相当合适。

3.3 坐标存贮法

对非零元结构不规则的稀疏矩阵，有多种可行的存贮方法，坐标法就是其中之一。当利用坐标法存贮一个稀疏矩阵 A 时，需要利用一个数据类型与 A 相同的向量 x，以及两个整型向量 $x^{(I)}$ 与 $x^{(J)}$，例如，对如下矩阵

$$A=\begin{bmatrix}1 & & & 2 & \\ 3 & 4 & & 1 & \\ & & 5 & 2 & \\ & 5 & & 8 & \\ 3 & & & & 9\end{bmatrix} \tag{4}$$

采用坐标法存贮，则

$$x=(1,2,3,4,1,5,2,5,8,3,9)^T,$$
$$x^{(I)}=(1,1,2,2,2,3,3,4,4,5,5)^T,$$
$$x^{(J)}=(1,4,1,2,4,3,4,2,4,1,5)^T$$

显然上述存贮方法并不能实现对矩阵元素的有效访问，例如需要访问矩阵中的某个元素时，在最坏情况下，需要遍历整个向量 x。为此，也提出了一些改进的方法，这里就不在详细介绍了。

3.4 Ellpack – Itpack 存贮法

采用 Ellpack – Itpack 方法存贮 n 阶矩阵 A 时，如果 A 第 i 行中有 m_i 个元素不为 0，且 $m=\max\{m_i;1\leqslant i\leqslant n;\}$，则需要用一个与 A 同类型的 $n\times m$ 矩阵 $B=(b_{ij})_{n\times m}$ 存贮 A 中的元素，A 第 i 行中的非零元素依次存放到 $\{b_{ij};j=1,2,\cdots,m_i\}$，而 $b_{i,m_i+1},\cdots,b_{i,m}$ 可赋为任意元素，通常赋为 0。为了正确访问 A 中的元素，还需要引入一个 $n\times m$ 的整型矩阵 $F=(f_{ij})_{n\times m}$，f_{ij} 为 b_{ij} 对应的 A 中的某非零元素 a_{ik} 的列号 k，$f_{i,m_i+1},\cdots,f_{i,m}$ 可赋为任意 1 与 n 间的整数，但通常可以赋 $f_{ij}=i,j=m_i+1,\cdots,m$。

例如对矩阵 (4) 采用这种方法存贮，则

$$B=\begin{bmatrix}1 & 2 & 0\\ 3 & 4 & 1\\ 5 & 2 & 0\\ 5 & 8 & 0\\ 3 & 9 & 0\end{bmatrix},F=\begin{bmatrix}1 & 4 & 1\\ 1 & 2 & 4\\ 3 & 4 & 3\\ 2 & 4 & 4\\ 1 & 5 & 5\end{bmatrix}$$

3.5 CSR 存贮法

CSR 为 Compressed Sparse Row 的缩写，在许多文献中翻译为行格式存贮法，顾名思义，这种方法是对矩阵进行逐行压缩存贮的。采用这种方法存贮 n 阶矩阵 A 时，假设 A 中共有 l 个非零元，则需要用一个 l 维向量 x 按先行后列的顺序依次存放 A 中的非零元素，用一个 l 维向量 $x^{(J)}$ 按同样的顺序依次存放 A 中这些非零元素的列号，还需要引入一个 $n+1$ 维整型向量 $x^{(R)}$，用 $x_i^{(R)}$ $(1\leqslant i\leqslant n)$ 指明 A 中第 i 行中第一个非零元素被存贮在 x 中的位置，而 $x_{n+1}^{(R)}=l+1$，例如对矩阵 (4)，则有

$$x=(1,2,3,4,1,5,2,5,8,3,9)^T,$$
$$x^{(J)}=(1,4,1,2,4,3,4,2,4,1,5)^T,$$
$$x^{(R)}=(1,3,6,8,10,12)^T$$

用 CSR 格式存贮矩阵时，A 的第 i 行中非零元素为 $x_j,j=x_i^{(R)},x_i^{(R)}+1,\cdots,x_{i+1}^{(R)}-1$，而这些非零元素的列号分别由 $x_j^{(J)},j=x_i^{(R)},x_i^{(R)}+1,\cdots,x_{i+1}^{(R)}-1$ 给出。

当然也可以采用先列后行的顺序逐列进行压缩存贮，这种方法被称为 CSC (Compressed Sparse Column) 格式。对于对称矩阵，自然可以只存贮上三角或下三角部分非零元素。

3.6 Sherman's 存贮法

对稀疏矩阵进行 LU 分解时，通常分解因子 L 和 U 中许多行中非零元素的列号互相覆盖，特别常见的一种情况是对某满足 $i<j$ 的两行 i，j，它们在第 j 列以后的元素列号相同，这时，可以对列号再进行压缩存贮，Sherman's 存贮法就是这样一种技术。

对 n 阶上三角矩阵 U，采用 Sherman's 存贮法时，需将其对角元素存贮到一个 n 维向量 $x^{(D)}$，非对角线上的非零元素按行存贮到向量 $x^{(N)}$，同时引入一个 $n+1$ 维整型向量 $x^{(R)}$，用 $x_i^{(R)}$ $(1\leqslant i\leqslant n)$ 指明矩阵第 i 行非对角线上的非零元在 $x^{(N)}$ 中的起始位置。用另一个整型数组 $x^{(J)}$ 以压缩形式记录 $x^{(N)}$ 中元素的列号，如果第 j 行中非零元素的列号与某 $i<j$ 行中最后部分的非零元列号完全相同，则第 j 行中的列号不需要记录。此外，还需要引入一个 $n-1$ 维整型向量 $x^{(P)}$，用 $x_i^{(P)}$ $(1\leqslant i\leqslant n)$ 指明第 i 行非对角线上的非零元列号在 $x^{(J)}$ 中的起始位置。

例如对如下矩阵采用这种存贮法，则有

$$U=\begin{bmatrix}1 & 11 & & & & 12 & & & 13\\ & 2 & & & 14 & & & 15 & \\ & & 3 & & & 16 & & & 17\\ & & & 4 & & 18 & & & 19\\ & & & & 5 & 20 & & 21 & \\ & & & & & 6 & 22 & & 23\\ & & & & & & 7 & 24 & 25\\ & & & & & & & 8 & 26\\ & & & & & & & & 9 & 27\\ & & & & & & & & & 10\end{bmatrix} \tag{5}$$

$$x^{(D)}=(1,2,3,4,5,6,7,8,9,10)^T,$$
$$x^{(N)}=(11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27)^T,$$
$$x^{(R)}=(1,4,6,8,10,12,14,16,17,18,18)^T,$$
$$x^{(J)}=(3,7,10,6,9,8,10,9,10)^T,$$
$$x^{(P)}=(1,4,2,2,4,6,6,8,9)^T$$

3.7　超矩阵存贮法

在一些科学计算当中，常会出现这样一些分块矩阵，当各个分块都较小时，只有少量块内存在非零元素，而其他块中的元素都是 0。将每个块看成一个元素，再将前面所述的存贮方法进行适当的修改，就可以实现对这种分块稀疏矩阵的存贮了，通常这种技术被称为超矩阵存贮法。

3.8　动态存贮方案

在有些情况下，对稀疏矩阵进行分解时，难以事先确定分解因子的非零元结构，这时，可以在分解中的每步上，逐个确定填充元的位置和数值，这就需要为这个填充元动态分配存贮空间。这种随分解进行，存贮结构不断变化的存贮方法称为动态存贮方案。

4　结论

本文从基本的稀疏矩阵存储技术出发，介绍了一些常用的稀疏矩阵存储方法，比较了它们的优缺点，并给出了它们的适用条件。由于不同的计算领域产生的稀疏矩阵具有不同的稀疏结构和特性，所以并没有一种通用的存储方法；而要根据实际情况，选取合适的稀疏存储方法，以期达到较好的计算效果。

参考文献

[1] 胡家赣. 线性代数方程组的迭代解法 [M]. 北京：科学出版社，1991：173 - 203.

[2] 周树荃. 有限元结构分析并行计算 [M]. 北京：科学出版社，1994：175 - 236.

[3] 徐树方. 矩阵计算的理论与方法 [M]. 北京：北京大学出版社，1995：161 - 200.

[4] 李庆扬. 现代数值分析 [M]. 北京：高等教育出版社，1995：214 - 231.

[5] Georges P, Warzee G. High - performance PCG Solves for FEM Structural Analysis [J]. International Journal for Numerical Methods in Engineeing, 1996 (39)：1313 - 1340.

[6] Arany I. Numerical Experiences of Solving Elasticity Systems by PCG Methods [J]. Computers and Mathematics with Applications, 2001 (42)：1025 - 1033.

[7] Beaumens Robert. Iterative Solution Methods [J]. Applied Numerical Mathemetics, 2004 (51)：437 - 450.