稀疏矩阵

就像在上一节描述的一样，标准的离散化的偏微分方程往往会伴随着一个庞大的且稀疏的矩阵。稀疏矩阵可以被模糊的描述为一个具有非常少的非零元的矩阵。但是，事实上，当特殊的技巧需要利用到大量的非零元以及它们的位置时，一个矩阵是可以被稀疏化的。这些稀疏化矩阵的技巧是从不储存零元的想法开始的。一个关键的问题是制定能够适合于高效地使用不论是直接还是迭代的标准计算方法的存储稀疏矩阵的数据结构。这一章节将简介稀疏矩阵，它们的属性、呈现，以及用以存储它们的数据结构。

3.1 介绍

利用一个矩阵中的零元以及它们的位置的自然的想法最初是由在不同学科的工程师们提出的。在涉及带状矩阵的简单地例子中，特殊的技巧直接的被发明了。在 20 世纪 60 年代研发电子网络的电子工程师们是最早的去利用稀疏性来对于具有特殊结构的矩阵解决一般稀疏线性系统。对于稀疏矩阵技巧而言，最主要也是最早需要解决的问题是去设计一个在线性系统中得直接求解算法。这些算法需要是可以接受的，在存储和计算效率上。直接的稀疏算法可以被用于计算那些庞大的难以被稠密算法来实现的问题。

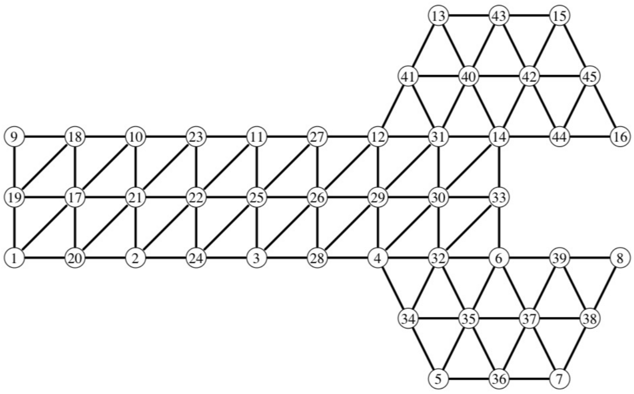

Figure 3.1 *A finite element grid model.*

基本上，有两个明显的类别的稀疏矩阵，结构化的和非结构化的。一个结构

化的矩阵是指一个非零元的位置形成某个规律的矩阵，通常这些非零元在对角线附近。要不然，这些非零元会在相同大小的块内（稠密子矩阵），而这也会形成一个规律，通常这些非零元在对角线（块）附近。一个具有着不规则位置的非零元的矩阵会被称作是非结构化的。最好的一个结构化的矩阵的例子是一个只有着少量对角元的矩阵。网格上的有限差分矩阵，就像上一节中提到的，是典型的具有着规律结构的例子。大部分的对于复杂几何的有限元和有限体积技巧会导致非结构化的矩阵。图 3.2 展示了一个与图 3.1 所呈现的有限元网格问题的一个小规模的非结构化的矩阵。

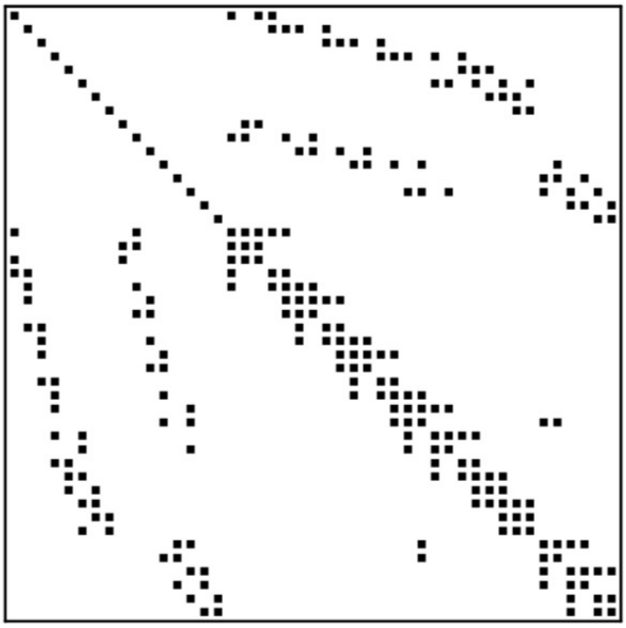

Figure 3.2 *Sparse matrix associated with the finite element grid of Figure 3.1.*

3.2 图论

图论是用来表示稀疏矩阵结构的一个理想的工具，因此，在稀疏矩阵技巧中，它扮演着一个主要的角色。例如，图论是用于解决并行稀疏高斯消除和预处理技术的关键。在下一节中，将讨论图的一般特性，以及它们在有限元和有限差分矩阵中得应用。

3.2.1 图与邻接图

记住一个图由两个集合定义，一个顶点集合 $V = \{v_1, v_2, \cdots, v_n\}$ 和一个边的集合 E, E 是由点对 (v_i, v_j) 组成的，v_i, v_j 都是 V 中的元素，换而言之，$E \subseteq V \times V$。这个图 $G = (V, E)$ 通常被平面内的一系列的被边联系的点的向量来表示。这个图被用来描述集合 V 中元素间的关系。例如，V 可以被用来描述世界上的主要城市。线就是两个城市间的直达航线。那么这个图就会描述这样一个关系"在城市 A 和城市 B 间存在一条直达航线"。在这个特殊的例子中，二元关系很可能是对称的，换而言之，如果有一条 A 到 B 的直达航线，那么也有一条 B 到 A 的直达航线。在这样的情形中，图被称作是无向的，用以与通常的有向图相对。

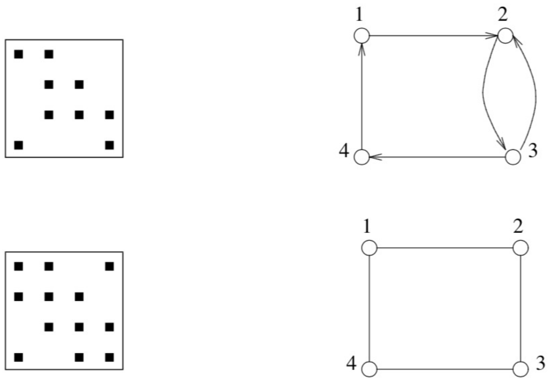

Figure 3.3 *Graphs of two* 4×4 *sparse matrices.*

回到稀疏矩阵，稀疏矩阵的邻接图是一个图 $G = (V, E)$，V 中得 n 个顶点代表 n 个未知数。它的边是按照以下规则建立的方程式建立的二元关系：当 $a_{ji} \neq 0$ 时，有一条从节点 i 指向节点 j 的边。而这条边将因此描述包含未知量 j 的二元关系方程式 i。注意，这个图是有向的，除非这个矩阵是有对称结构的（对任意的 $1 \leq i, j \leq n$，若 $a_{ji} \neq 0$，那么 $a_{ij} \neq 0$ ）。
当一个矩阵的非零元总有一个对称非零元，换而言之 a_{ij} 和 a_{ji} 总是同时为非零元，那么这图就是无向的。因此，对于无向图，每条边都有两个方向。因此，无向图可以用无向边来表现。
作为利用图模型的例子，并行高斯消去法可以通过寻找在指定消去阶段的未知数来获得。根据以上的二元关系，这些未知数两两独立。这些与未知数一致的行可以被用作基。因此，在一个极端情况下，当一个矩阵是对角阵，那么所有的未知数是独立的。与之相反的是，当一个矩阵是稠密的，那么每一个未知量都与其他未知量相关。稀疏矩阵则介于这两种极端情况之间。
邻接图有一些有趣的简单性质。A^2 的图可以被解释成一个 n 顶点图，对每条边的点对 (i,j)，表示在原图 A 中至少存在一条长度确切的说是 2 的从节点 i 到节点 j 的路径。与之相似的时，A^k 的图包含的时用以描述从节点 i 到节点 j 的至少存在一条长度为 k 的路径的二元关系的边。欲知详情，请看练习 4.

3.2.2PDE 矩阵的图

对于在每个网格点只涉及一个屋里未知量的偏微分方程，离散矩阵的邻接图通常就是用来描述网格的图。但是，在每个网格点上有着多个未知量是很常见的。例如，模拟流体流动的方程可能涉及流体的两个速度分量（二维）以及在每个网格点的能量和动量。在这样的情况下，有两种用来标记未知量的选择。在每个网格点，它们可以被连续的标记。因此，在刚才的例子中，我们可以在一个指定的网格点例如 u(k),$\cdots$,u(k+3) 上标记所有的四个未知量（两个速度的分量，动量以及压力）。另外，所有的与一类变量相关的未知量可以最先被标记（比如，第一个速度分量），接下来是第二类的变量（比如，第二个速度分量）等等。在任意情况下，很明显邻接矩阵是有冗余信息的。物理网格的商图可以被用来替代使用。这将节约大量的存储量和计算量。在上述的流体流动的例子中，用以描述图的整数数组的存储可以被缩小到接近 1/16。这是因为边的数量被所见到了大约这么多，但是通常很小的顶点数却保持着不变。

3.3 置换和重新排序

对于稀疏矩阵而言，重排序行或列，或者行和列是一个常见操作。事实上，重排序行和列是一个用于直接求解法和迭代法并行实现的一个最重要的部分。本节介绍这些重排技术和矩阵的邻接图的相关思想的关系。记得在第一章中，矩阵的第 j 列记作 a_{*j}，第 i 行则记作 a_{i*}。

3.3.1 基础概念

我们先开始一个定义与符号。
定义 3.1 有一个矩阵 A，以及 $\pi = \{i_1, i_2, \cdots, i_n\}$ 的一个交换集合 $\pi = \{i_1, i_2, \cdots, i_n\}$。那么矩阵

$$A_{\pi,*} = \{a_{\pi(i),j}\}_{i=1,\cdots,n;j=1,\cdots,m}$$

$$A_{*,\pi} = \{i, a_{\pi(j)}\}_{i=1,\cdots,n;j=1,\cdots,m}$$

就分别被称作 A 的行 π-交换和列 π-交换。
广为周知的时，最多 n 个交换（换而言之，就是只互换两项的基本排列）可以产生集合 $\{1, 2, \cdots, n\}$ 的任意置换。一个交换矩阵就是一个两行互换了得单位矩阵。用 X_{ij} 来表示第 i 和 j 行交换了的单位矩阵。注意到，为了交换矩阵 A 的第 i 和 j 行，我们可以用矩阵 X_{ij} 来左乘矩阵 A。让 $\pi = \{i_1, i_2, \cdots, i_n\}$ 为一个任意序列。这个置换就是一系列连续的交换矩阵 $\sigma(i_k, j_k), k = 1, \cdots, n$ 的乘积。那么，我们就可以通过交换矩阵的 i_1 和 j_1 行，然后再结果矩阵的基础上交换 i_2 和 j_2 行，以此类推，最后交换 i_n 和 j_n 行。每一步我们都可以通过左乘矩阵 X_{i_k,j_k} 来实现。同样的，对于矩阵的列也是一样的：为了交换矩阵的第 i 和 k 列，通过右乘矩阵 X_{i_k,j_k} 可以实现。从上我们可以得到下述命题。
命题 3.1 让 π 是交换 $\sigma(i_k, j_k), k = 1, \cdots, n$ 的乘积得到的置换。那么，$A_{\pi,*} = P_\pi A$，$A_{*,\pi} = AQ_\pi$，当

$$P_\pi = X_{i_n,j_n} X_{i_{n-1},j_{n-1}} \cdots X_{i_1,j_1} \tag{3.1}$$

$$Q_\pi = X_{i_1,j_1} X_{i_2,j_2} \cdots X_{i_n,j_n} \tag{3.2}$$

这些交换矩阵的乘积被称作置换矩阵。显然，一个置换矩阵只不过是进行了行列交换的单位矩阵。
注意到 $X_{i,j}^2 = I$，换而言之，置换矩阵的平方是一个单位矩阵，或者等价地，置换矩阵的逆等于它本身，这是很显然的一个属性。易见，矩阵（3.1）和（3.2）满足

$$P_\pi Q_\pi = X_{i_n,j_n} X_{i_{n-1},j_{n-1}} \cdots X_{i_1,j_1} \times X_{i_1,j_1} X_{i_2,j_2} \cdots X_{i_n,j_n}$$

表示了矩阵 Q_π 和 P_π 都是非退化的，且互为另一个的逆。换而言之，用同一个置换矩阵来交换一个矩阵的行和列事实上做了类似的变换。因为定义 (3.1) 和（3.2）中得 P_π 和 Q_π 的乘积是相反的顺序，另一个推论就显而易见了。由于每一个基矩阵 XX_{i_k,j_k} 是对称的，那么 Q_π 是 P_π 的转置。因此

$$Q_\pi = P_\pi^T = P_\pi^{-1}$$

因为矩阵 P_π 的逆矩阵是它的转置，置换矩阵就是唯一的。
另一个用来推出上述关系的方法是用置换矩阵 P_π 和 P_π^T 来代表行列交换了的单位矩阵。（在练习 3 中）显而易见

$$P_\pi = I_{\pi,*} \; P_\pi^T = I_{*,\pi}$$

那么，接下来就可以直接验证

$$A_{\pi,*} = I_{\pi,*}A = P_\pi A \; A_{*,\pi} = AI_{*,\pi} = AP_\pi^T$$

这对于在线性系统中解释置换操作很重要。当矩阵的行交换了，方程的顺序就改变了。换而言之，当列交换了，那么未知量就会对应的改变标记或者改变顺序。
例子 3.1 思考，比如，线性系统 $Ax = b$，当

$$A = \begin{pmatrix} a_{11} & 0 & a_{13} & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & 0 \\ 0 & a_{42} & 0 & a_{44} \end{pmatrix}$$

以及 $\pi = \{1, 3, 2, 4\}$，那么（列）交换线性系统是

$$\begin{pmatrix} a_{11} & a_{13} & 0 & 0 \\ 0 & a_{23} & a_{22} & a_{24} \\ a_{31} & a_{33} & a_{32} & 0 \\ 0 & 0 & a_{42} & a_{44} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}$$

注意到，不只是未知量交换了，方程也是，特别的，右边没有变。
在上述例子中，只有 A 的列交换了。在稀疏矩阵技术中，这样的单侧变换不如两侧变换寻常。事实上，这通常与线性系统中得对角元起着一个明显且重要的角色的事实有关。比如，在偏微分应用中，对角元通常很大，而且，在交换矩阵

中可能很需要去保留这一性质。为了达到这一目的，很典型的是去同时对 A 的行和列进行相同的交换。这样的操作被叫做对称置换，若果用 $A_{\pi,\pi}$ 来表示，那么，这样的对称置换的结果就满足这一关系

$$A_{\pi,\pi} = P_{\pi}^{T} A P_{\pi}$$

对称置换的解释很简单。由此产生的矩阵用相同的方式重命名，或重标记，或重排序未知量和重排序等式。

例子 3.2 对于前面的例子，如果行和列用相同的置换矩阵来置换，那么线性系统可以这样来获得

$$\begin{pmatrix} a_{11} & a_{13} & 0 & 0 \\ a_{31} & a_{33} & a_{32} & 0 \\ 0 & a_{23} & a_{22} & a_{24} \\ 0 & 0 & a_{42} & a_{44} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}$$

注意到，对角元是原来的矩阵的对角元在主对角线上德一个不一样的顺序。

3.3.2 与邻接图的关系

从图论的角度，另一个对于对称置换的重要解释是这相当于不不改变边来重新标记顶点。事实上，(i,j) 是原矩阵 A 的邻接图的边，A’ 是交换了得矩阵。当且仅当 $(\pi(i), \pi(j))$ 是原始矩阵 A 的图中的一条边时，那么 $a'_{ij} = a_{\pi(i),\pi(j)}$，结果 (i,j) 是交换了的矩阵 A’ 的邻接图中一条边。因此，交换了的矩阵的图没有改变；甚至，顶点的标记也是。与之相对的是，不对称置换就不会保存好图。事实上，它们可以将一个无向图转换为一个有向图。尽管邻接矩阵的一般图是相同的，对称置换可能会对矩阵的结构造成一些重大的影响。

例子 3.3 考虑图 3.4 描述的矩阵和它的邻接图。因为它们的形状，这样的矩阵又是被叫做“箭头”矩阵，但是，因为它们图的结构可能更适合把它们叫做“星”矩阵。

如果用置换 $9, 8, \cdots, 1$ 来重排序等式，图 3.5 所描述的矩阵和图就得到了。尽管两张土建的区别看起来很小，但是矩阵可能会有一个对于算法有着重要影响的完全不同的结构。以此为例，如果用高斯消去法来重排序矩阵，那么填充就不会发生。换而言之，LU 分解的 L 和 U 部分会与 A 的下和上两部分有着一样的结构。在另一方面，在原始矩阵上作高斯消去法会导致灾难性的填充。特别的，在高斯消去法第一部以后，LU 分解的 L 和 U 部分是稠密矩阵。用直接稀疏矩阵技术，找到在高斯消去过程中对于较少填充有作用的矩阵的置换很重要。最后这一段，总的来说，两侧非对称置换也可能在实践中出现。然而，在直接法中它们很常见。

3.3.3 常用重排序

在实践中，重排序和置换的种类七绝与直接或者迭代法是否被考虑。下面是一个这样的对赌迭代法更为有用的一个重排序的例子。

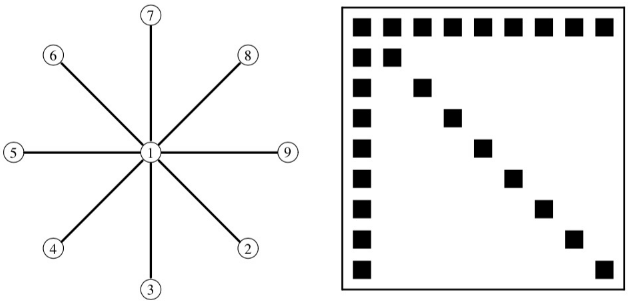

Figure 3.4 *Pattern of a 9 × 9 arrow matrix and its adjacency graph.*

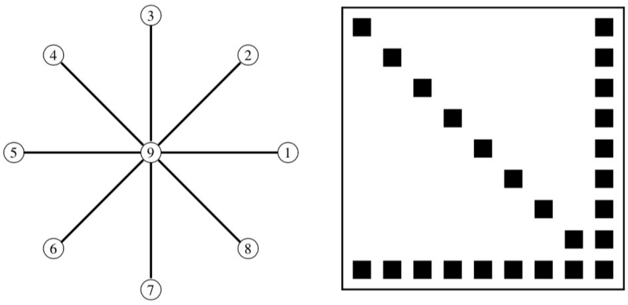

Figure 3.5 *Adjacency graph and matrix obtained from above figure after permuting the nodes in reverse order.*

水平集序这种顺序类型包含了许多基于水平集的图的便利的技巧。水平集是递归定义的上一级的所有节点的所有未标记的邻集。最初，一个水平集有一个节点，虽然有几个将来会被讨论的也重要的起始点。当一个水平集被遍历完成，它的节点就被标记了且排了编号。比如，它们可以按照遍历的顺序来编号。另外，遍历顺序的不同会产生不同的顺序。例如，某一个水平集中的节点可以按照它们列出的自然序访问。然后，可以检查它们每一个的邻节点。每一次，当遇到一个访问过的顶点有一个没有编号的邻节点，那么它就被添加到列表中并标记为下一水平集的下一元素。在图论中，这个简单地策略被称为广度优先搜索。在每个水平集中，顺序取决于节点遍历的方式。在广度优先搜索中，水平集中元素总是以它们列出的自然序来遍历。在 Cuthill-McKee 排序中，水平集的元素被以从最低到最高的顺序来遍历。