稀疏矩阵模版类SparseMatrix设计说明*

2007 年 5 月 7 日

1 稀疏矩阵

设计一个稀疏矩阵模版类, 用于处理m行和n列的稀疏矩阵A, $m > 0$, $n > 0$. A具有如下特点:

1. A中非零元的个数为N;
2. 如果A是方阵, 则A的对角元非零;

2 存储策略

使用两个整型数组和一个实型数组来存放一个稀疏矩阵. 分别是:

- 整型数组rowIdx[$m+1$];
- 整型数组colIdx[N];
- 实型数组entries[N];

其中rowIdx[i]($0 \le i < m$)存放的是第i行的第一个非零元在colIdx中的指标. 因此, rowIdx[0]总是0. 而rowIdx[m]的值指定为N. 非零元总是按行依次连续存放在entries中, 而每个非零元对应的列指标存放在colIdx的相应位置.

例

$$A = \begin{bmatrix} 3.14 & 0 & 0 & 0 & 1.31 \\ 0 & 0 & 0 & 6.1 & 0 \\ 0 & 4.75 & 0 & 0 & 1.9 \\ 1.1 & 0 & 0 & 0 & 0 \\ 5.9 & 0 & 0 & 0 & 4.6 \end{bmatrix}$$

*该文档为李若教授《数值分析高等算法》的课堂笔记, 由王何宇整理.

则

- rowIdx=$\{0, 2, 3, 5, 6, 8\}$,
- colIdx=$\{0, 4, 3, 1, 4, 0, 0, 4\}$,
- entries=$\{3.14, 1.31, 6.1, 4.75, 1.9, 1.1, 5.9, 4.6\}$.

事实上, 同一行中的元素在entries中存放的时候, 次序是没有关系的, 因此为了实现迭代算法时方便, 我们附带要求如果稀疏矩阵是方阵, 则每一行的第一个元素总是对角元. 而对非方阵, 则不需做这种安排. 为了编程方便, 我们将认定方阵的对角元总是非零元, 即便它是零, 我们也把它当作一个非零元处理. 因此, A的存储形式现在变为

- rowIdx=$\{0, 2, 4, 7, 9, 11\}$,
- colIdx=$\{0, 4, 1, 3, 2, 1, 4, 3, 0, 4, 0\}$,
- entries=$\{3.14, 1.31, 0, 6.1, 0, 4.75, 1.9, 0, 1.1, 4.6, 5.9\}$.

3 模版类的实现

要求使用GNU C++编写一个模版类, 实现上述稀疏矩阵的存储, 并实现一定的基本运算功能. 要求对矩阵元素的类型模版化处理. 在编写的时候, 可以考虑单独建立一个Pattern类来专门存放和处理标识了矩阵结构的数组rowIdx和colIdx. (仅仅是建议)

在考虑矩阵的基本操作前, 我们首先因该考虑矩阵的产生和使用的问题. 首先, 矩阵从哪里来? 可能的来源有两个, 一个是从某个格式化的数据文件直接读取, 一个是直接在程序中生成, 比如一个差分离散的程序, 直接形成一个稀疏矩阵. 任何时候都不应该考虑将一个非稀疏的矩阵转化为稀疏矩阵或是相反的问题. 这个问题从根本上违反了稀疏矩阵存在的意义. 但是直接从程序中形成时会产生新的问题: 我们如何初始化一个稀疏矩阵? 在还不知道具体的矩阵数据, 甚至不知道具体的非零元的坐标的时候, 我们如何开辟一个恰当的内存空间来为一个即将形成的稀疏矩阵做准备? 在很多情况下, 我们可以期望即将产生的矩阵是某种形式的带状阵, 或接近带状的矩阵, 这种情况在有限差分和有限体积离散的时候非常常见. 因此我们可以先将矩阵Pattern初始化成一个能存放一个带宽为w(w为用户给的参数)的带状稀疏阵的样子, 但是colIdx中的值全部为-1. 如果将Pattern单独写成类, 此时可以不必理会entries的情况. 然后根据实际情况, 逐个将非零元的坐标标识出来, 将对应的列指标存入colIdx. 在标识完所有的非零元后, 将所有列指标仍然为-1的元素删除掉, 并释放空间. 这个过程一般称为Pattern的压缩. 然后可以根据这个Pattern, 生成一个稀疏矩阵, 并且存入实际的数据. 而有些时候, 我们甚至连带宽都无法指定, 比如有限元离散后的矩阵. 但我们至少可以预估一下每一行可能出现的非零元的个数. 因此还应该提供一种能根据每一行非零元可能的个数初始化Pattern的操作. 所以, 这里我们需要如下的操作:

- 矩阵Pattern根据指定的带宽初始化;
- 矩阵Pattern根据用户提供的一个数组初始化, 这个数组存放的是每一行的非零元的个数;
- 对初始化后的Pattern指定第(i,j)元素为非零元的操作. 这里(i,j)表示实际矩阵的i行j列;
- 压缩矩阵Pattern;
- 根据矩阵Pattern初始化一个矩阵;
- 对初始化的矩阵实际赋值;
- 从一个格式化数据文件读取并形成一个稀疏矩阵, 文件格式自己设计;
- 矩阵加法; (出于实际需要, 这里总是假定相加的矩阵具有相同的Pattern)
- 矩阵乘向量;
- 向量乘矩阵;
- 矩阵乘矩阵;
- 其他, 比如打印一个矩阵, 打印一个Pattern, 等等.

4 测试

用5点中心差分离散Poisson方程

$$-\Delta u = f, u|_{\partial\Omega} = g.$$

其中

$$f = 2\sin x \sin y, g = 0, \Omega = [0,\pi] \times [0,\pi].$$

可以将网格密度定为1000×1000, 这样将得到一个100万阶的矩阵, 并用Gauss-Seidel法求解该矩阵. 并将解与真解

$$u = \sin x \sin y$$

做比较.