一、稀疏矩阵

1.1定义

稀疏矩阵:只有少部分为非零项（大部分只占1%）

应用：微分方程数值解、线性规划、网格分析、结构和非结构问题的有限元分析等领域提出了求解高阶稀疏线性方程组的问题。方程组的阶数可以上千阶到几十万阶，甚至几百万阶或更高。

1.2求解方法

对于一般的大规模稀疏矩阵，一般方法有直接法和迭代法。

其中直接法指使用矩阵的分解方法，直接计算出方程的解（或者判定方程是否有解），

在不考虑计算机计算精度丢失问题，该方法可以经过有限步算术运算，可求得方程组的精确解的方法

迭代法

用某种极限过程去逐步逼近线性方程组精确解的方法。迭代法具有占存储单元少，程序设计简单，原始系数矩阵在迭代过程中不变等优点，但存在收敛性及收敛速度等问题。千万阶矩阵求解以迭代法（含多重网格法）为主。迭代法算法简单，易于实现。先天不足

1.2直接法求解

优势

直接法的优势

稳定，通用，精度高基于恰当的主元技术及预处理技术，目前直接法的表现相当稳定，精度也比迭代法局很多。

CPU效率可高一个数量级目前单颗的峰值浮点性能己有超过即每秒亿次运算。但只有为数不多的代码能实现这个峰值，其中就有。通过算法优化，稀疏分解的绝大部分运算可直接调用，而迭代法一般只能调用。两者的效率相差一个数量级。

更适合于反复求解的场合

稀疏矩阵通常需要反复求解，即矩阵不变，而右端项不断变化。例如非线性方程组求解，拟牛顿迭代，有限元中的多种荷载等。直接法只需数值分解一次，然后针对不同的右端项回代。而回代的速度很快，一般远小于迭代法的时间。因此，当问题中需要反复求解的次数较多，直接法所需的时间会更少。

但是也有一定的局限：

劣势

需要大量内存

这个是限制直接法的主要瓶颈所在。就百万阶矩阵实测来看，需要左右内存，千万阶所需内存估计在以上。估计工作站的内存要数年后才能达到这个数量级。因此需要外存模式，辅助完成求解的过程，这将增加额外的时间，并使算法更加复杂。

如pardiso

最优内存为4G

算法相对比较复杂

主流求解器的核心代码都在万行以上。这些求解器都采用很多参数，针对具体问题如何取值也有一些难度。目前很多直接求解器已经商业化，但总体上还有很多提局的空。

如pardiso ,19参数

不乏iparm(64)

二、求解过程

求解的前提是完成矩阵方程AX=B的A矩阵的存储和基本运算

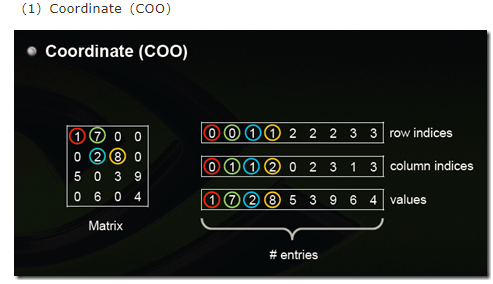
2.1储存矩阵

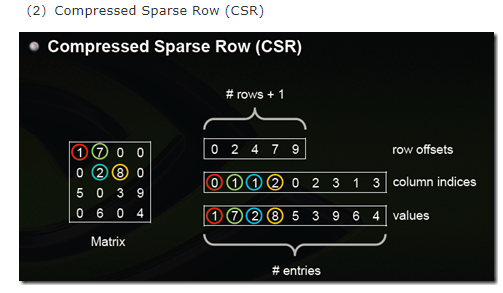
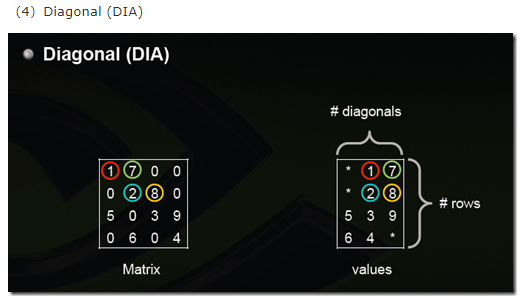
储存方式为压缩的方法，主要有以下一些常见的储存

Skyline

CRS

COO



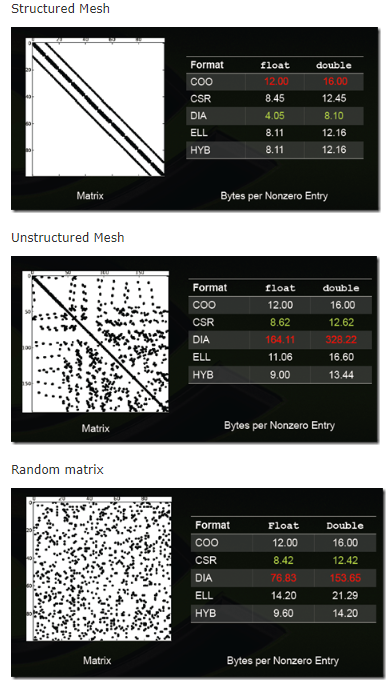


基本实现：

使用特定的存储规律（压缩储存）

如skyline(配合bandwidth (优化)

那么储存方式是否影响求解呢？答案显然是，因为不同储存方式其实就是为了求解特定形式的矩阵而提出的，如COO型专门储存对角线型矩阵，而CRS储存随机性矩阵较好，如下图所示：



因此，在计算某些特定矩阵时应该使用适合的压缩储存方式。能够显著提高运算速度。

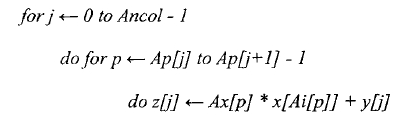
2.2基本运算

矩阵的运算：

乘法运算：

使用类储存为例，AX+Y=Z的计算如下：

利用只对列元素所在位置的储存性质，只需要把对应列的非零元素位置提取（储存为行号）提出进行计算即可。



转置：

可以根据列储存还原矩阵A，并且按照行储存的方式进行存储，输入列储存进行计算即为AT

2.3直接求解

2.3.1基础性方法

Gauss消去法

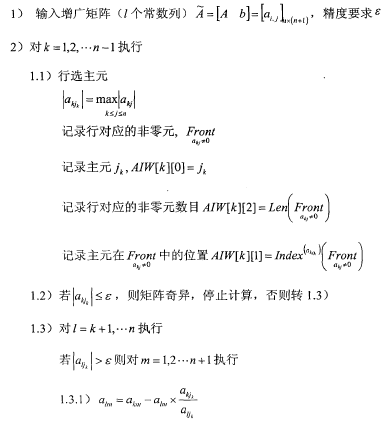
高斯法结合Cramer法则可以判定和求解矩阵。但是Cramer有着O(n！)复杂度，判定不能在运算前进行。高斯法是一个古老的求解线性方程组的直接方法，但基于消去法的基本思想改进或变形而得到的选主元消去法和三角分解法则是目前计算机上常用的求解低阶稠密矩阵线性方程组的有效方法。消去法的基本思想是将系数矩阵或增广矩阵），消去为上三角矩阵来求解，消去法有三种形式：按自然顺序选主元法，即按主对角元的顺序，按列选主元法，以及整体选主元法。衍生方法有Gauss-Jordon消去，LU,QR分解法。

2.3.2波前法

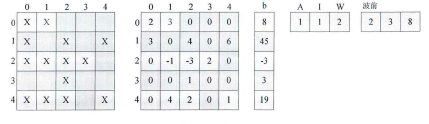
下面集中介绍以下一种基于Gauss法的求解法——波前法：

波前法基本原理是基于对高斯消去法的再分析，在高斯消去法中，主要采用将主元变换到对角线位置，然后进行消元操作。而波前法则是提取出需要消元的非零元素组成波前，而不考虑零元，然后对集成的波前进行统一处理完成计算。

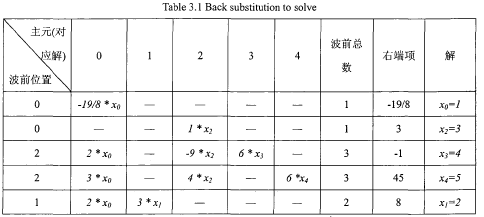
波前法的算法如下所显示，基本原理是算一步储存一部分：



同时一个其做消去的算例是：



回代求解过程如下表格，整个求救过程是按照生成波前相反的顺序进行，每回代一次生成一个新的解，回代过程比较容易理解，计算过程中需要注意的是已经求取的解在后续的计算中使用时的顺序。因为回代过程中右端项是被单次使用，求出的新解可以保持在原来右端项存储的位置。



而可以见到波前法求解的特点是：刚度矩阵和载荷列矩阵不按照自然编号进入内存而按计算时参加运算的顺序排列；在内存中保留尽可能少的一部分和中的元素。波前法适用于中小型计算机求解大型有限元方程组的常用解法，它是对高斯消去法的改进，主要特点是集成和消元操作分离交替进行，不对零元存储和计算，从而有效地节省了内存。

因此，可以使用波前法在内存很小的计算机上进行大规模矩阵直接法判定与求解，由于使用的是单步执行，内外存使用的情况，运算速度较慢。

2.3求解器结构

为了方便求解器用户接口设计和求解器幵发中的调试工作，求解器提供了多种类型的函数接口，图是对的接口函数功能说明，常用的内部调用的函数主要分为以下几类

1.求解核心函数，通过该部分函数可以完成对矩阵核心求解工作。该部分最为重要，是完成2.求解最主要的部分，占据了求解器的大部分实现代码，调用关系可参见图；

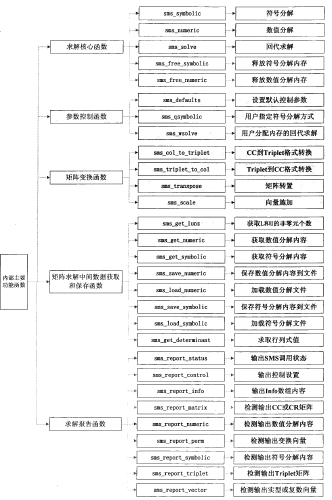
参数控制函数，如果用户对默认的求解过程不满意，可以用使用该部分函数对求解行为进行改变；

3.矩阵变换函数，完成对矩阵各种形式之间的转换；

4.矩阵求解中间数据获取和保存函数，可以对矩阵的求解过程进行保存和读取，主要用于调试程序的使用；

5.求解报告函数，通过该部分函数，可以输出矩阵元素值，已经控制参数所代表的含义等诸多内容；

参考以下的结构组成：



下面以Intel@MKL数学库的pardiso 矩阵求解器作为案例：

1.Pardiso 使用介绍：

例程pardiso使用并行LU，LDL或LL T分解来 计算具有多个右侧的一组稀疏线性方程的解，其中A是n\*n矩阵，并且X和B是n\*m矩阵。

A\*X = B

支持的矩阵类型。

由pardiso执行的分析步骤取决于输入矩阵A的结构。

对称矩阵：解算器首先根据Metis包[Karypis98]中的最小度数算法[Liu85]或嵌套分解算法[Karypis98]计算对称填充减少排列p（两者都包含在Intel MKL中），然后是并行的左-右查找数字Cholesky因式分解[Schenk00-2]of PAPT=LLT，对称正定矩阵，对称不定矩阵的使用PAPT=LDLT。解算器对对称不定矩阵采用对角支点法，即1X1和2X2簇和考夫曼支点法，通过正、反替换和迭代精化得到X的近似值。

结构对称矩阵：解算器首先计算对称填充减少排列p，然后计算PAPT=qlut的并行数值因子分解。解算器在超节点中使用了部分旋转，通过前向和后向替换和迭代优化找到了x的近似值。

非对称矩阵：解算器首先计算非对称置换PMP和缩放矩阵Dr和Dc，目的是在对角线上放置大的条目，以增强数值因子分解过程的可靠性[Duff99]。在下一步中，解算器根据矩阵pmpsa+（pmpsa）t计算填充减少排列p，然后进行并行数值分解。

非对称线性系统的直接迭代预处理。

解算器可以使用直接和迭代方法的组合[SOnn89]来加速瞬态模拟的线性解过程。稀疏解算器的大多数应用都需要非零系数矩阵值逐渐变化但具有相同稀疏模式的系统的解。在这些应中，求解器的分析阶段只需执行一次，数值分解是模拟过程中的重要耗时步骤。Pardiso对第一个系统使用数值因子分解a=l u，并在预处理的Krylow子空间迭代中对下一步应用因子l和u。如果迭代不收敛，解算器将自动切换回数值因子分解。该方法可应用于Pardiso中的非对称矩阵。只能使用一个输入参数选择方法。有关更多详细信息，请参见参数说明（IParm（4）、IParm（20））。

单精度和双精度计算。

Pardiso使用单精度或双精度解决任务。每种精度都有其优点和缺点。双精度变量有更多的数字来存储值，因此解算器使用更多的内存来保存数据。但该模式能较好地求解矩阵，输入矩阵可以有较大的条件数。

单精度变量存储值的位数较少，因此求解器使用的内存比双精度模式的内存少。此外，此模式通常花费较少的时间。但由于计算比较粗略，只有数值稳定的过程才能使用单精度。

分离正向和反向替换。

解算器执行步骤（见下面的参数phase=33）可分为两个或三个单独的替换：正向、反向和可能的对角线。这种分离可以用求解不同矩阵类型的系统的例子来解释。

实对称正定矩阵a（mtype=2）被pardiso分解为a=l\*lt。在这种情况下，系统a\*x=b的解可以作为替换序列找到：l\*y=b（正向替换，phase=331）和lt\*x=y（反向替换，phase=333）。

真正的非对称矩阵a（mtype=11）被pardiso分解为a=l\*u。在这种情况下，系统a\*x=b的解可以按以下顺序找到：l\*y=b（正向替换，phase=331）和u\*x=y（反向替换，phase=333）。

用实对称不定矩阵A（mtype=-2）求解系统与上述情况略有不同。pardiso因子这个矩阵为a=l d lt，系统a\*x=b的解可以按如下替换顺序计算：l\*y=b（正向替换，phase=331）s:d\*v=y（对角替换，phase=332），最后lt\*x=v（反向替换，phase=333）。对角替换只对不定矩阵（mtype=-2，-4，6）有意义。对于其他类型的矩阵，可以找到解决方案，如前两个示例中所述。

PARDISO中的稀疏数据存储遵循稀疏矩阵存储格式中描述的方案，其中ja代表列，ia代表rowIndex，a代表值。PARDISO中的算法要求列索引ja每行递增顺序，并且每行中的对角元素对于任何结构对称矩阵都存在。对于对称或非对称矩阵，对角元素不是必需的：它们可以存在或不存在。

建议为对称矩阵明确设置零对角元素，因为在相反的情况下，PARDISO创建数组的内部副本ia，ja和一个需要额外内存和计算时间的完整对角元素。然而，通常，与存储器和分解和求解矩阵所需的时间相比，存储器和时间开销并不重要

四、总结

对于大型矩阵的求解是很重要的工作，目前，包括从算法、矩阵储存乃至硬件方面都有许多科研工作者在进行研究，本次有限元课程中能使用到pardiso 库作为求解使用的辅助库能加快矩阵运算。