

一篇文章入门大规模线性方程组求解

原创 邓子平 [多物理场仿真技术](#)



前面介绍过主要的线性方程组求解库，参考附录。求解大规模线性方程组是仿真软件求解器的底层技术，求解器时间基本都消耗在方程组求解上。线性方程组的解法比较成熟，方法也有很多，而且不同的方法对应不同类型方程组，所以在方法选择上实际很讲究。

商业软件通常将方法封装起来，用户包括开发人员都接触不到线性方程组求解方法。商业软件内部一般会根据求解规模，求解类型等选择合适的线性方程组求解方法。

有些商业软件开放了部分接口供用户选择；开源软件比如OpenFOAM，以及使用开源软件的平台Simscale等提供了许多选项供用户选择。

本文简单介绍下线性方程组的常用解法。通常将线性方程组表示为：

$$A \cdot x = b$$

A为已知 $N \times N$ 的矩阵，通常称为刚度矩阵（刚度是力学中的概念，电磁，热等也习惯性这么称呼），b为已知向量，x为待求向量。解线性方程组的操作基本围绕矩阵A展开。

首先介绍一些相关术语：

1. 矩阵条件数

条件数是一个表征矩阵稳定特性的标志，条件数越大，说明矩阵越不稳定，即当矩阵中数据出现微小变化时，x结果变化非常大。Matlab中可直接使用命令`cond(A)`查看矩阵条件数。

2. 满秩矩阵

用初等行变换将矩阵A化为阶梯形矩阵，矩阵中非零行的个数就定义为这个矩阵的秩，记做 $r(A)$

即如果矩阵 A 为 $N \times N$ ， $r(A)=N$ ，则矩阵 A 为满秩矩阵

在边界元，矩量法等计算方法中，最终形成的 A 为满秩矩阵。

3.稀疏矩阵

矩阵中绝大部分元素都是0

4.对称矩阵

矩阵的上三角和下三角关于对角线对称

5.求解线性方程组直接法

先求出矩阵 A 的逆矩阵，再乘以向量 b

6.求解线性方程组迭代法

简单讲，就是给出一个初始解 x' ，带入原方程中，每次评价差距逐步修正，直到最终 $A \cdot x' - b$ 接近0为止。

7. OOC (out of core)

对于大型方程组，内存通常无法装下，在求解过程中需要将部分数据写到硬盘上，再次使用的时候再读回来。

常用直接法：

直接法的本质是要计算出 A 的逆矩阵，通常在求解小规模和特征值问题是可以考虑使用直接法。

1. 高斯消去法/Doolittle /三角分解法/追赶法

这是最基本的方法，时间复杂度和空间复杂度都是 N 的三次方，软件一般都不会使用。

2. 矩阵分解相关

2.1. LU分解法

LU分解就是将矩阵分解成单位下三角矩阵L和上三角矩阵U，本质上仍然属于高斯消去法

2.2. Cholesky分解

Cholesky 分解是把一个对称正定的矩阵表示成一个下三角矩阵L和其转置的乘积的分解。如果矩阵是正定的，使用 Cholesky分解会比LU分解更加高效。

2.3. LDLT分解法

Cholesky 分解法的改进

2.4. QR分解

QR分解是把矩阵分解成一个正交矩阵与一个上三角矩阵的积。

2.5. Schur分解

2.6. SVD/GSVD

奇异值分解/一般奇异值分解

在求解大规模线性方程组中，一般不会使用直接法求解，但在使用迭代法过程中需要使用直接法中的方法加工数据。

间接法的本质是迭代，不同方法的区别在于如何选取初始值以及迭代方法。

1. 牛顿迭代, Jacobi(雅克比)迭代, Gauss-Seidel迭代

线性迭代方法，一般情况下收敛太慢，大规模方程组不推荐使用。

2. JOR -- Jacobi Over Relaxation

Jacobi 方法加入松弛因子

3. Lanczos方法

适用于稀疏矩阵特征值问题

4. 共轭梯度方法以及改进方法

共轭梯度法

Conjugate gradient method--CG

双共轭梯度法

Bi-conjugate gradient method--BCG

稳定双共轭梯度法

Bi-conjugate gradient stabilized method--GCGS

预条件共轭梯度法

Preconditioned gradient stabilized method--PCG

相比牛顿和Jacobi，通过优化使其共轭的求解向量和方向，加速了求解性能。

共轭梯度法的思想就是找到N个两两共轭的方向，每次沿着一个方向优化得到该方向上的极小值，后面再沿其它方向求极小值的时候，不会影响前面已经得到的沿哪些方向上的极小值，所以理论上对n个方向都求出极小值就得到了N维问题的极小值。

5. GMRAS广义最小残量

(Generalized Minimal Residual Algorithm)

非对称系统的线性方程组的数值解迭代法，该方法与最小残量的 Krylov 子空间中向量来逼近解，是求解大规模线性方程组的常用算法之一,具有收敛速度快、稳定性好等优点。

改进的GMRAS：

SGMRAS Simpler GMRAS

PGMRAS Preconditioned GMRAS

6. 快速多级（Fast Multi pole Method）

当矩阵为满秩矩阵时，传统计算方法资源需求和求解规模呈指数级上升，大规模的系统求解需要使用快速多级方法。本技术博客和公众号中有详细介绍。

改进方法：多层快速多级

7. Successive Over Relaxation(SOR)连续松弛

连续过度松弛（SOR）方法是高斯-赛德尔（Gauss-Seidel）方法的一种变体，用于求解线性方程组，从而可以更快地收敛。任何缓慢收敛的迭代过程都可以使用类似的方法。

改进的SOR：

Accelerated Over relaxation (AOR)过松弛

Preconditioned AOR -- PAOR 预条件过松弛

Quasi AOR -- QAOR准过松弛

8. Krylov子空间迭代法

将矩阵A分解成多个子矩阵，并用一系列线性表达式组合表示。将整个系统降维，利于并行计算，是大规模线性方程组有效的一种解法。其中涉及到了Lanczos方法和Arnoldi方法。

按照矩阵A的特点，我们可以做如下分类：

1. 是否是满秩矩阵；
2. 是否是稀疏矩阵；
3. 是否是对称矩阵；
4. 是否是病态矩阵；
5. 是否是正定矩阵
6. 矩阵规模（矩阵中N的大小）

在选择求解方法的时候，需要考虑到方程组矩阵以上特点。根据作者的经验，方程组的求解性能高度依赖矩阵特征，矩阵规模和硬件。

问题：

求解线性方程组 $Ax=b$.

Matlab 中提供了方程的求解方法：

$x=A\backslash b$ (\backslash 反除法符号)

最简单的问题，一群兔子和鸭子，一共有40条腿，15张嘴，问兔子鸭子各多少只，两个变量，两个方程组成一个简单的线性方程组：

$$4x+2y=40$$

$$x+y=15$$

在Matlab中输入：

```
>> A =
```

```
4 2
```

```
1 1
```

```
>> b=[40, 15]'
```

```
b =
```

```
40
```

```
15
```

```
>> x=A\b
```

```
x =
```

```
5
```

```
10
```

当方程求解规模达到1000*1000的时候，Matlab还能应付，速度也不错。当规模达到10w*10w的时候，Matlab会运行相当长时间，然后弹出一个错误。这是因为对于10w*10w的矩阵，用原始的数据结构，性能会下降很快。由有限元计算的矩阵特征可知，矩阵是数据大部分为0的稀疏矩阵，Matlab提供了稀疏矩阵功能，在使用 $x=A\backslash b$ 之前 使用如下命令：

```
x=sparse(x);
```

10万*10万的方程组1秒之内求出结果。

对于大型线性方程组的求解，使用Matlab做验证是个不错的选择

回到主题:

1. BLAS(Basic Linear Algebra Subprograms)

blas 是许多数值计算软件库的核心，用Fortran语言写成，库一共有三个level，第一level包含了vector计算的一些function；第二个level则是矩阵与vector计算；第三个level是矩阵与矩阵的计算。BLAS接口稍微复杂，一般很少直接使用。

开源

2. Intel MKL

商用是不错的选择。

Intel数学核心函数库（MKL）是一套高度优化、线程安全的数学例程、函数，面向高性能的工程、科学与财务应用。英特尔 MKL 的集群版本包括 ScaLAPACK 与分布式内存快速傅立叶转换，并提供了线性代数 (BLAS、LAPACK 和Sparse Solver)、快速傅立叶转换、矢量数学 (Vector Math) 与随机号码生成器支持。

主要包括：

LAPACK (线形代数工具linear algebra package)

DFTs (离散傅立叶变换 Discrete Fourier transforms)

VML (矢量数学库Vector Math Library)

VSL (矢量统计库Vector Statistical Library)

MKL的主要功能

2.1) BLAS 和 LAPACK

在英特尔处理器中部署经过高度优化的基本线性代数例程BLAS（Basic Linear Algebra Subroutines）和 线性代数包LAPACK（Linear Algebra Package）例程，它们提供的性能改善十分显著。

2.2) ScaLAPACK

ScaLAPACK是一个并行计算软件包，适用于分布存储的MIMD并行机。ScaLAPACK提供若干线性代数求解功能，具有高效、可移植、可伸缩、高可靠性的特点，利用它的求解库可以开发出基于线性代数运算的并行应用程序。ScaLAPACK 的英特尔MKL 实施可提供显著的性能改进，远远超出标准 NETLIB 实施所能达到的程度。

2.3) PARDISO稀疏矩阵解算器

利用 PARDISO 直接稀疏矩阵解算器解算大型的稀疏线性方程组，该解算器获得了巴塞尔大学的授权，是一款易于使用、具备线程安全性、高性能的内存高效型软件库。英特尔? MKL 还包含共轭梯

度解算器和 FGMRES 迭代稀疏矩阵解算器。

2.4) 快速傅立叶变换 (FFT)

充分利用带有易于使用的新型 C/Fortran 接口的多维 FFT 子程序（从 1 维至 7 维）。英特尔 MKL 支持采用相同 API 的分布式内存集群，支持将工作负载轻松地分布到大量处理器上，从而实现大幅的性能提升。此外，英特尔 MKL 还提供了一系列 C 语言例程（“wrapper”），这些例程可模拟 FFTW 2.x 和 3.0 接口，从而支持当前的 FFTW 用户将英特尔 MKL 集成到现有应用中。

2.5) 矢量数学库 (VML)

矢量数学库 (Vector Math Library) 借助计算密集型核心数学函数（幂函数、三角函数、指数函数、双曲函数、对数函数等）的矢量实施显著提升应用速度。

2.6) 矢量统计库—随机数生成器 (VSL)

利用矢量统计库 (Vector Statistical Library) 随机数生成器加速模拟，从而实现远远高于标量随机数生成器的系统性能提升

商用

3. Eigen

主页：

http://eigen.tuxfamily.org/index.php?title=Main_Page

Eigen 是一个轻量级的库，使用只需头文件，接口比较齐全，文档也很全面，一般的矩阵操作运算都能满足。提供了 DenseMatrix 和 SparseMatrix，对于稀疏矩阵的效率也不错。

开源

4. Spoolse

早期很多人使用，性能不错，使用简单。维护的不是很好，文档使用 ps 格式。

<http://www.netlib.org/linalg/spooles/spooles.2.2.html>

开源

5. Lapack

使用最广泛的线性代数包，底层调用 BLAS，能求解 $AX=b$ ，矩阵分解、求逆，求矩阵特征值、奇异值等。文档齐全，使用灵活，稳定性好，用作研究和商业开发都是不错的选择。提供 SVN 源码下载

<http://www.netlib.org/lapack/>

开源

6. Pardiso

PARDISO是为解决大 稀疏对称和非对称线性系统的方程的软件包，如下特点：

线程安全，,高性能的, 和内存使用率低，支持共享内存和内存多处理器。

<http://www.pardiso-project.org/>

商用，学术用免费

7. Mumps

使用多波前法的稀疏矩阵求解库，支持并行计算。

<http://mumps.enseeiht.fr/>

开源

8. PETSc

PETSc是一个高大上的科学计算库，求解线性方程是其中一个功能，支持内存共享并行计算机，支持多线程，GPU加速等，支持求解稀疏矩阵

<http://www.mcs.anl.gov/petsc/documentation/linearsolverstable.html>

开源

9. SuperLU

SuperLU是一个求解大规模，稀疏，非对称系统对线性方程组的通用库,c语言编写，可供Fortran和C调用。最新版本为4.3（2014/10/1）支持并行计算和分布式计算。

<http://www.cs.berkeley.edu/~demmel/SuperLU.html>

开源

10. Umfpack

UMFPACK是 用来求解不对称稀疏线性系统软件包, $Ax = b$,使用非对称多波方法, SuiteSparse有此算法

使用比较简单, Windows下需要自己编译。

<http://www.cise.ufl.edu/research/sparse/umfpack/>

开源

11. TAUCS

求解稀疏矩阵线性方程软件包, 最新版本支持多线程

<http://www.tau.ac.il/~stoledo/taucs/>

开源

12. SuiteSparse

SuiteSparse是一组C、Fortran和MATLAB函数集, 用来生成空间稀疏矩阵数据。在SuiteSparse中几何多种稀疏矩阵的处理方法, 包括矩阵的LU分解, QR分解, Cholesky分解, 提供了解非线性方程组、实现最小二乘法等多种函数代码

<http://faculty.cse.tamu.edu/davis/suitesparse.html>

开源

13. Sparselib++

用C++写的 稀疏矩阵库, 可以和 IML++一起使用做线性方程组的迭代求解。

<http://math.nist.gov/sparselib++/>

开源

14. Trilinos

Trilinos也是个高大上的东西, 求解线性方程只是其中一部分功能

<http://trilinos.org/capability-areas/#ScalableLinearAlgebra>

开源

15. IML++

使用C++模板库 和迭代方法 求解对称, 非对称矩阵库

<http://math.nist.gov/iml++/>

开源

使用推荐:

1. 简单快速上手 Eigen
2. 商业使用 Intel MKL
3. 研究使用 Lapack, PETSc, SuitSparse