计算物理 第一部分

第3讲　线性方程组的直接解法

$$A\mathbf{x} = \mathbf{b}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

李强　北京大学物理学院西楼227

qliphy0@pku.edu.cn，15210033542

引言

本章的目的是讨论如下n阶线性方程组的直接求解方法：

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases} \qquad \sum_{j=1}^{n} a_{ij}x_j = b_i, \quad i = 1, 2, \cdots, n。$$

写出矩阵形式为：

$$\mathbf{Ax} = \mathbf{b},$$

$$\mathbf{A} = [a_{ij}]_{n\times n},\ x = (x_1, \cdots, x_n)^T,\ \mathbf{b} = (b_1, \cdots, b_n)^T,$$

分别称为系数矩阵、解向量和右端向量。我们这里仅讨论A和b均为实数向量的情况，此时x也为实向量。

引言

- **线性方程组的求解**是一个基本而十分重要的的问题。它与**矩阵特征值问题**构成矩阵计算的两大类问题。实际上，很多理论和实际问题都直接或间接的与线性方程组的求解问题相关。在自然科学和社会科学中，很多物理量之间的关系就是线性的，如：牛顿定理$\mathbf{F} = m\mathbf{a}$；欧姆定理$V = IR$；胡克定律$\mathbf{F} = -k\mathbf{x}$。欧姆定理和基尔霍夫定理导出的大型电路方程属于线性方程组，IBM 的大规模集成电路方程可达几十万。
- 在数值计算中，线性方程组的求解显得异常重要，很多计算问题最后归结为线性方程组的求解。譬如某些问题在局部做线性关系近似；微分和积分方程这类非代数问题，离散化后可以用线性代数方程组来近似。可以说，**线性方程组的求解构成了求解各种计算问题数值方法的基础。**
- 很多实际问题的需求也为线性方程组的数值求解提出了新的挑战，例如方程规模庞大，变量众多($> 10^6$)、存储问题、算法的稳定性和效率问题、舍入误差传递问题、并行算法的设计等等。

引言

- **线性相关：** 设$\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_m} \in \mathcal{R}^n$，若存在不全为零的数$\alpha_1, \alpha_2, \cdots, \alpha_m \in \mathcal{R}$，使得$\alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2 + \cdots + \alpha_m \mathbf{x}_m = \mathbf{0}$，则称向量$\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_m}$是线性相关的。若只有$\alpha_i$全为0时才成立，则称它们是线性无关的。
- **非奇异矩阵：** 是指满足以下等价条件之一的矩阵$\mathbf{A}_{n\times n}$：(1) 存在逆矩阵$\mathbf{A}^{-1}$，满足$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$，$\mathbf{I}$为单位矩阵；(2) 矩阵的行列式$\det(\mathbf{A}) \neq 0$；(3) $\mathbf{A}$ 的秩$\mathrm{rank}(\mathbf{A}) = \mathrm{n}$ (矩阵的秩是其包含的线性无关的行或列的最多个数)；(4) 对任意的非零向量$\mathbf{z} \neq \mathbf{0}$，有$\mathbf{A}\mathbf{z} \neq \mathbf{0}$。
- **方程组$\mathbf{A}\mathbf{x} = \mathbf{b}$解的存在性和唯一性：** (1) 若$\mathbf{A}$为非奇异矩阵，则有唯一的解：$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ (对$\mathbf{b}$没有要求)；(2)若$\mathbf{A}$为 奇异矩阵，且$\mathbf{b} \in \mathrm{span}(\mathbf{A})$，其中集合$\mathrm{span}(\mathbf{A})$表示$\mathbf{A}$ 的各个列向量张成的线性空间，则方程组有无穷多个解；(3) 若$\mathbf{A}$为奇异矩阵，且$\mathbf{b} \notin \mathrm{span}(\mathbf{A})$，则方程组无解。

引言

对方程组$\mathbf{Ax} = \mathbf{b}$:

- 若$A = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix}$，则无论$b$取任何值，方程组总有唯一解。
- 但是若$A = \begin{pmatrix} 2 & 3 \\ 4 & 6 \end{pmatrix}$，$A$是奇异矩阵。

  其解的情况依赖于b的取值，但无论如何都不可能有唯一解。

  $b = [4\ \ 7]^T$时，方程组无解；

  而当$b = [4\ \ 8]^T$时，$x = \begin{bmatrix} \gamma & \frac{4-2\gamma}{3} \end{bmatrix}$($\gamma$为任意实数)都是方程组的解。

克莱姆法则

- 如果线性方程组的系数行列式不为零，即$\det(\mathbf{A}) \neq 0$，则该方程组有唯一的解。由克莱姆(Cramer) 法则，其解为：

$$x_i = \frac{\det(A_i)}{\det(A),} \quad (i = 1, 2, \cdots, n).$$

上式中，A_i是被列向量$\mathbf{b}$取代了$\mathbf{A}$的第i列的列向量后得到的矩阵。

- **计算复杂度：** 如果按照这种方法直接计算方程的解，则需要计算$n+1$个n阶行列式，并作n次除法。每个n阶行列式计算需要$(n-1) \times n!$次乘法，计算量将十分可怕。例如，取$n = 30$，将有$\sim 10^{35}$次乘法。因此使用克莱姆法则求线性方程组的解的算法时间复杂度太高，一般没有实际计算价值。
- $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ 需要求矩阵的逆，再做矩阵与矢量乘积，计算复杂度低很多；但是我们要寻求更好的算法。

引言

- **直接解法：** 假定计算过程没有舍入误差的情况下，经过有限步算术运算后能求得线性方程组精确解的方法。这类精确解的方法，在实际计算中由于舍入误差的影响，也只能求得近似解(但准确度高！)。直接法的核心是高斯消元。
- **迭代解法：** 构造适当的向量序列，用某种极限过程去逐步逼近精确解。例如：雅可比迭代法、高斯-赛德尔迭代法、逐次超松弛迭代法、最速下降法、共轭梯度法等。迭代法的优点是所需计算机储存单元少，便于编写计算机程序。另外，迭代的极限过程一般不可能进行到底，因此只能得到满足一定精度要求的近似解。迭代法的收敛性依赖于系数矩阵的某些特性，为提高效率，往往要与预条件技术相结合。迭代法是高效求解大型稀疏系数矩阵线性方程组，尤其是微分方程离散后得到的大型方程组的一种重要方法。

在后面介绍微分方程求解时会简介迭代解法。

高斯消元法

$$Ax = b$$

数值上求解这类方程其实与我们在初中时学习的步骤基本一致。这实际上是一个古老的方法，称为高斯消元法(Gauss Elimination Method，GEM)。首先写出这些方程，然后对它们进行一系列的所谓的**初等变换–即将某个方程乘以一定的数然后将其与另一个方程相加减–其目的是消去某个变量前面的系数**。通过这样的变换，将原先的方程化为如下形式的方程：

$$U \cdot x = c \ , U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & & u_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & u_{nn} \end{bmatrix},$$

其中U是一个上三角矩阵。**如果矩阵A是非奇异的，那么上三角矩阵U一定也是如此，这意味着所有的对角元u_{ii}!=0**。因此这个上三角系统可以运用所谓**后代(back-substitution)**的方法来求解

$$x_i = \frac{c_i - \sum_{k=i+1}^{n} u_{ik} x_k}{u_{ii}}, \quad i = n, n-1, \cdots, 1 .$$

前代、后代

$$\begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \cdots \\ b_n \end{pmatrix}$$

下三角方程的前代算法

输入：$\mathbf{L}, n, \mathbf{b}$；　　输出：$\mathbf{x}$：

For $i = 1, 2, \cdots, n$

　If $l_{ii} = 0$ **then STOP;**

　$x_i := b_i;$

　For $j = 1, 2, \cdots, i-1$

　　$x_i := x_i - l_{ji} x_j;$

　End

　$x_i := x_i / l_{ii};$

End

上三角方程的回代算法

输入：$\mathbf{U}, n, \mathbf{b}$；　　输出：$\mathbf{x}$：

For $i = n, n-1, \cdots, 1$

　If $u_{ii} = 0$ **then STOP;**

　$x_i := b_i;$

　For $j = n, n-1, \cdots, i+1$

　　$x_i := x_i - u_{ij} x_j;$

　End

　$x_i := x_i / u_{ii};$

End

这个算法的乘除法次数为 $n + \cdots + 2 + 1 = \frac{1}{2}n^2 + \frac{1}{2}n$。加减法次数为 $(n-1) + \cdots + 2 + 1 = \frac{1}{2}n^2 - \frac{1}{2}n$。
如果忽略低次项，反代过程中的浮点数乘除法 + 加减法的计算总次数为 n^2。

增广矩阵、置换矩阵

将原先的矩阵A和$\vec{b}$合并为一个n×(n+1)的矩阵

$$(A,\vec{b}) = \begin{pmatrix} a_{11} & \cdots & a_{1n} & b_1 \\ \cdots & & \cdots & \cdots \\ a_{n1} & \cdots & a_{nn} & b_n \end{pmatrix}$$

我们随后的线性变换都是直接对增广矩阵(A,b)来进行操作。最终目的是将其变换为上三角的形式。我们**首先去寻找一个a_{r1}!=0**。对所有的$1 \le r \le n$来说，这样的矩阵元总是存在，否则矩阵A将是奇异的，与初始假设矛盾。如果我们找到了这样的一个矩阵元，我们**首先可以做的是将第1行与第r行进行互换**。这可以通过一个置换矩阵来实现。

$$(\bar{A},\bar{b}) = P^{1r} \cdot (A,b)$$

置换矩阵

n×n的置换矩阵P^(1r)的矩阵元为：

$$[P^{(1r)}]_{1r} = [P^{(1r)}]_{r1} = 1, \quad [P^{(1r)}]_{11} = [P^{(1r)}]_{rr} = 0$$

$$[P^{(1r)}]_{ij} = \delta_{ij}, \quad (ij) \neq (1r) \neq (r1)$$

除了第1和第r行、第1和第r列这四个矩阵元之外，其他的矩阵元与单位矩阵的相应矩阵元完全一样。很容易验证，**这个矩阵是一个非奇异的矩阵并且它的逆矩阵就是它本身。**

当用它左乘矩阵(A,b)时，其作用是交换该矩阵的第1行和第r行。

$$(\bar{A}, \bar{b}) = P^{(1r)} \cdot (A, b)$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 7 & 8 & 9 \\ 4 & 5 & 6 \end{pmatrix}$$

Frobinius 矩阵

$$G^{(1)} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ -l_{21} & 1 & \cdots & 0 \\ \cdots & \cdots & \ddots & \cdots \\ -l_{n1} & 0 & \cdots & 1 \end{pmatrix}$$

它是一个下三角矩阵，并且仅仅在一列（具体到这个例子是第一列）与单位矩阵不同，其他地方与单位矩阵相同。它的作用是，**只要我们适当地选取l_{r1}的数值，就可以将增广矩阵中第一列中除了第一个元素之外的其他元素统统变换为零：**

$$l_{r1} = \bar{a}_{r1}/\bar{a}_{11},\ r = 2, \cdots, n,$$

容易证明，**这个矩阵也是非奇异的，事实上它的逆矩阵也是一个Frobinius矩阵**，只不过其中的$-l_{r1}$都要换成$+l_{r1}$。

置换+Frobinius变换

$$(A', b') = G^{(1)}(\bar{A}, \bar{b}) = G^{(1)} P^{(1r)}(A, b)$$

$$(A', b') = \begin{pmatrix} a'_{11} & a'_{12} & \cdots & a'_{1n} & b'_1 \\ 0 & a'_{22} & \cdots & a'_{2n} & b'_2 \\ \cdots & \cdots & \ddots & \cdots & \cdots \\ 0 & a'_{n2} & \cdots & a'_{nn} & b'_n \end{pmatrix}$$

其中的第一个矩阵P^(1r)的作用是，首先选择一个不为零的元素a_{r1}并且将它调到第一行，**当然如果a_{11}本身就不等于零，这一步原则上也可以省略**；第二个矩阵则将新矩阵的第一列从第二个元素开始往下的矩阵元全部清零。

第一个待求的变量x_1仅仅出现在第一个方程之中，后面的(n-1)个方程仅仅包含其余的n-1个变量：x2,···,xn。

这个过程可一直迭代下去。最终结果就是我们将增广矩阵变换为了我们希望的上三角的形式并进一步利用反代的方法获得线性系统的解。

支点遴选

前面提及的寻找的不为零的矩阵元 $\bar{a}_{11}=a_{r1}$ 有个专门的名称，叫做**支点元**，或简称支点。而这个过程称为支点遴选。虽然任意的不为零的元素都可以作为支点，但是直觉告诉我们它应当尽可能地远离零，因此一个自然的选择是

$$\bar{a}_{11} = \max_r |a_{r1}|$$

数学上可证明这样的选择造成的误差比起随便胡乱选择的支点来说是比较小的。这个选择一般称为**部分支点遴选**。与之相比，我们还可以进行所谓完全支点遴选。在**完全支点遴选**中，我们不仅仅局限于第一列，而是在所有矩阵元中挑选模最大的作为支点：

$$\bar{a}_{11} = \max_{r,s} |a_{rs}|$$

随后我们将矩阵的第1行与第r行，第1列与第s列对换（这相当于将原先的解x的第1个分量与第s个分量进行了一次对换）。

高斯消元法

在第一步的支点遴选的过程中的置换矩阵P^(1r)的上标r其实并不是必须的，它只是临时从各个行中计算出的一个具有最大模的矩阵元的行指标而已。因此为了下面描述的方便，我们将这一步的置换矩阵记为P^(1)，即隐去上标r。并且我们将约化后的矩阵记为(A^(1)，b^(1))。原始的增广矩阵则给它一个上标0，即(A^(0)，b^(0))≡(A，b)。

$$(A^{(1)}, b^{(1)}) = G^{(1)} P^{(1)} (A^{(0)}, b^{(0)})$$

$$(A^{(j)}, b^{(j)}) = G^{(j)} P^{(j)} (A^{(j-1)}, b^{(j-1)}), \quad j = 1, 2, \cdots, (n-1)$$

$$(A^{(n-1)}, b^{(n-1)}) \equiv (U, c)$$

这个过程就是高斯消元法的完整过程。经过这个约化，原先的线性系统被成功约化为一个上三角线性系统。

$$(U, c) = G^{(n-1)} P^{(n-1)} G^{(n-2)} P^{(n-2)} \cdots G^{(1)} P^{(1)} (A, b)$$

高斯消元法复杂度

忽略置换的计算复杂度。
最外层循环的第k步计算n-k次除法，(n-k)(n-k+1)次乘法，因此除法、乘法的总数是

```
For k = 1, ··· , n − 1
    If a_kk = 0, then stop
    For i = k + 1, k + 2, ··· , n
        l := −a_ik/a_kk    计算倍数因子
        For j = k + 1, k + 2, ··· , n
            a_ij := a_ij + l · a_kj    更新矩阵元素
        End
        b_i := b_i + l · b_k    更新右端项
    End
End
```

$$\text{除法：}(n-1)+(n-2)+\cdots+1=\frac{n(n-1)}{2}$$

$$\text{乘法：}n(n-1)+(n-1)(n-2)+\cdots+2\times 1=\frac{(n+1)n(n-1)}{3}$$

忽略掉低次项，消元过程中的浮点数乘除法的计算次数约为n^3/3。加减法的次数也是n^3/3。**这就把行列式计算的复杂度O(n!)降为高斯消元的2n^3/3。**这对于量级为n~O(100) 的矩阵是没有任何问题的。当然，如果矩阵是10^6或者更大，我们还需要其他更好的方法。

高斯消元法复杂度

这里，需要提醒大家注意的是，由于计算复杂度较高，**在实际应用中应尽量避免对矩阵求逆**。

事实上，很多数学表达式中虽然包含了矩阵的逆，如A^{-1}b，但实际计算时并不需要真正计算A^{-1}，

比如求解方程Ax=b 就会得到A^{-1}b的结果，这比求出A^{-1}再做矩阵乘法更有效(事实上，光是矩阵乘法就需要O(n^3)次乘法运算)，也更准确。

实际问题中真正需要 A 的逆的情况很少，因此一旦见到公式中的A^{-1}，首先应该想到的是解方程组而不是求矩阵的逆。

矩阵的 LU分解

一个方阵A∈C^{n×n}的LU分解是指将其**分解为一个下三角和一个上三角矩阵的乘积**：

$$A = LU$$

其中L(U)分别是下(上)三角矩阵，L和U分别指代lower和upper。**如果一个矩阵A的LU分解可以获得，那么求解它的线性方程可以转化为先后求解两个三角形矩阵的线性问题**，而这个可以利用反代的方法解出。例如下面的两步走的方程的解x恰好就是Ax=b的解，只要A=LU ：

$$y = Ux, \quad Ly = b$$

我们下面论证，前一节的高斯消元法恰给出了矩阵A的一个LU分解。

矩阵的 LU分解

根据高斯消元过程，利用一系列置换矩阵与Frobinius矩阵的乘积，我们有**上三角矩阵**

$$U = G^{(n-1)}P^{(n-1)}G^{(n-2)}P^{(n-2)}\cdots G^{(1)}P^{(1)}A$$

其中Frobinius矩阵的一般形式为，

$$G^{(j)} = \begin{pmatrix} 1 & \cdots & 0 & 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \cdots & \cdots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & \cdots & 0 \\ 0 & \cdots & 0 & 1 & \cdots & \cdots & 0 \\ \vdots & \cdots & \vdots & -l_{j+1,j} & 1 & \cdots & 0 \\ \vdots & \cdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & -l_{n,j} & 0 & \cdots & 1 \end{pmatrix}$$

注意各个G^(j)矩阵，如果我们遍历所有的j=1,···,n −1，这些矩阵中非零的元素恰好填满一个n×n矩阵的左下三角区域，也就是主对角线的左下方的所有矩阵元。

定义矩阵

$$M = G^{(n-1)}P^{(n-1)}G^{(n-2)}P^{(n-2)}\cdots G^{(1)}P^{(1)}$$

可以证明，矩阵

$$L = PM^{-1}, \quad P = P^{(n-1)}\cdots P^{(1)}$$

是下三角矩阵，且对角元都是1。于是，我们得到：

$$PA = PM^{-1}U = LU$$

这个表达式实际上给出了**矩阵PA的一个所谓的LU分解**。由于下三角矩阵的对角元都已设成1，所以LU分解具有唯一性。

矩阵的LU可分解性

一个矩阵A是否存在LU分解，要取决于矩阵A以及它的各个子矩阵的秩。虽然对于任意矩阵的LU分解的定理有些复杂，但是对于一个实矩阵来说，它的判别标准还是比较简单的，这就是下面的定理

对于任意的实矩阵A∈R^{n×n}来说，它具有唯一的LU分解：A=LU，其中L是下三角矩阵且$l_{ii}=1$，i=1，···，n，U为上三角矩阵，其充要条件为A的所有主子矩阵A_i，i=1，···，(n −1)都是非奇异的

特别值得注意的是，即使是一个奇异的n×n矩阵也可以有唯一的LU分解，只要它的各个主子矩阵一直到(n−1)阶都是非奇异的即可。例如，对于奇异的矩阵

$$A=\begin{pmatrix}1 & 2\\ 1 & 2\end{pmatrix}$$

我们可以获得它的标准的、唯一的LU分解：

$$A=\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}\begin{pmatrix}1 & 2\\ 0 & 0\end{pmatrix}$$

这个定理的证明可以利用对i的数学归纳法展开。其步骤非常类似于我们下面要讲述的Cholesky分解中的证明，我们这里不再赘述。

Cholesky分解

如前面提到的，对于**正定的厄米矩阵**A∈C^{n×n}来说，我们可以找到一个矩阵H使得

$$A = H^{\dagger}H$$

事实上，我们**可以要求矩阵H是上三角矩阵**。这个分解一般称为Cholesky分解。因此，对于求解正定厄米矩阵的线性方程问题就化为寻找矩阵的Cholesky分解问题。

其实Cholesky分解可以从LU分解中得到。我们先对正定的厄米矩阵进行LU分解，**并把分解中的U矩阵写成对角矩阵乘以单位上三角矩阵的形式**

$$U = \begin{pmatrix} u_{11} & & & \\ & u_{22} & & \\ & & \ddots & \\ & & & u_{nn} \end{pmatrix} \begin{pmatrix} 1 & u_{12}/u_{11} & \cdots & u_{1n}/u_{11} \\ & 1 & \cdots & u_{2n}/u_{22} \\ & & \ddots & \cdots \\ & & & 1 \end{pmatrix} = DU_0$$

Cholesky分解

$$U = \begin{pmatrix} u_{11} & & & \\ & u_{22} & & \\ & & \ddots & \\ & & & u_{nn} \end{pmatrix} \begin{pmatrix} 1 & u_{12}/u_{11} & \cdots & u_{1n}/u_{11} \\ & 1 & \cdots & u_{2n}/u_{22} \\ & & \ddots & \cdots \\ & & & 1 \end{pmatrix} = DU_0$$

$$A = LDU_0$$

又因为A是厄米矩阵，则 $A = A^\dagger = U_0^\dagger D L^\dagger$

根据LU分解的唯一性，那么，我们有 $U_0^\dagger = L,$

$$A = LDL^\dagger = LD^{\frac{1}{2}} D^{\frac{1}{2}} L^\dagger = H^\dagger H$$

其中 $H^\dagger = LD^{\frac{1}{2}}$ 是下三角矩阵。

当然，实际我们在做Cholesky分解的时候，并不需要按照LU分解的步骤。由于对称性的存在，Cholesky分解要比普通的LU分解更省时。

Cholesky分解

按照我们前面提及的线性代数的结果，正定的厄米矩阵的所有主子矩阵也都是正定的。因此，寻找Cholesky分解可以按照**数学归纳法**的思路进行。对于n=1的一阶矩阵，问题的解是平庸的。

令A_i，$i=1,\cdots,n$是原矩阵的第i阶的主子矩阵。它们显然也都是正定的厄米矩阵。**假定我们已经找到了$A_{i-1}\in C^{(i-1)\times(i-1)}$的分解矩阵$H_{i-1}$，即$A_{i-1}=H^\dagger_{i-1}H_{i-1}$，我们试图来寻找$A_i$的分解矩阵 H_i。**

$$A_i=\begin{pmatrix} A_{i-1} & \vec{v} \\ \vec{v}^\dagger & \alpha \end{pmatrix}$$

其中α是一个正的实数，$\vec{v}\in C^{i-1}$为一矢量。事实上我们有

$$\vec{v}=(a_{1i},a_{2i},\cdots,a_{i-1,i})^T。$$

$$A_i=H_i^\dagger H_i=\begin{pmatrix} H_{i-1}^\dagger & 0 \\ \vec{h}^\dagger & \beta \end{pmatrix}\begin{pmatrix} H_{i-1} & \vec{h} \\ 0^\dagger & \beta \end{pmatrix}$$

其中的 $\vec{h}\in C^{i-1}$为一待定矢量而β为一个待定实数。将这个式子的右边乘出来我们就发现：

$$H_{i-1}^\dagger\cdot\vec{h}=\vec{v}$$

$$\vec{h}^\dagger\vec{h}+\beta^2=\alpha$$

Cholesky分解

$$A_i = H_i^\dagger H_i = \begin{pmatrix} H_{i-1}^\dagger & 0 \\ \vec{h}^\dagger & \beta \end{pmatrix} \begin{pmatrix} H_{i-1} & \vec{h} \\ 0^\dagger & \beta \end{pmatrix} \qquad \vec{v} = (a_{1i}, a_{2i}, \cdots, a_{i-1,i})^T 。$$

$$H_{i-1}^\dagger \cdot \vec{h} = \vec{v} \qquad \vec{h}^\dagger \vec{h} + \beta^2 = \alpha$$

由于$H^\dagger$，$\vec{v}$已知，并且$H^\dagger$是一个下三角矩阵，我们当然可以轻易求解出矢量$\vec{h}$。进而，

$$\beta = \sqrt{\alpha - \vec{h}^\dagger \vec{h}}$$

正定厄米矩阵的Cholesky**分解可以通过下面的算法获得。为了方便起见，我们考虑一个实空间而非复空间的矩阵。这样厄米矩阵就变成了实对称矩阵。**

记下三角矩阵$H^\dagger_{i-1}$的矩阵元为$h_{\alpha\beta}$，$\beta \le \alpha \le i-1$

Cholesky分解

记下三角矩阵$H^{\dagger}_{i-1}$的矩阵元为$h_{\alpha\beta}$，$\beta \le \alpha \le i-1$

$\vec{h}^{\dagger}$ 的向量元素为 $\vec{h}^{\dagger} = (h_{i1}, h_{i2}, \cdots, h_{i(i-1)})$，而 $\vec{h}$ 的向量元素为 $\vec{h} = \begin{pmatrix} h_{i1} \\ h_{i2} \\ \cdots \\ h_{i(i-1)} \end{pmatrix}$

$$\vec{v} = \begin{pmatrix} a_{i1} \\ a_{i2} \\ \cdots \\ a_{i(i-1)} \end{pmatrix}$$

这个反代的前几步可以写成

$$h_{1,1}h_{i,1} = a_{i,1} \quad \Rightarrow \quad h_{i,1} = \frac{a_{i,1}}{h_{1,1}}$$

$$h_{2,1}h_{i,1} + h_{2,2}h_{i,2} = a_{i,2} \quad \Rightarrow \quad h_{i,2} = \frac{1}{h_{2,2}}\left(a_{i,2} - h_{2,1}h_{i,1}\right)$$

$$h_{ij} = \frac{1}{h_{jj}}\left(a_{ij} - \sum_{k=1}^{j-1} h_{ik}h_{jk}\right), \quad j = 1, \cdots, i-1$$

Cholesky分解算法

For $j = 1, 2, \ldots, n$

For $k = 1, \cdots, (j-1)$

$a_{jj} := a_{jj} - a_{jk}^2$

End

$a_{jj} := \sqrt{a_{jj}}$ （这一步是为了实现 $\beta = h_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} h_{jk}^2}$）

For $i = j+1, j+2, \cdots, n$

For $k = 1, 2, \cdots, j-1$

$a_{ij} := a_{ij} - a_{ik}a_{jk}$

End

$a_{ij} := a_{ij}/a_{jj}$ （这一步是为了实现 $h_{ij} = \frac{1}{h_{jj}}\left(a_{ij} - \sum_{k=1}^{j-1} h_{ik}h_{jk}\right)$）

End

End

这里有个求平方根的运算，大家可以想想怎么快速求平方根。如果我们忽略n次求平方根运算，那么，**这个算法的需要n^3/6次乘除法和n^3/6次加减法，比LU分解要节省大约一半**。除此之外，因为对称性的缘故，**所需的内存也减半**。从稳定性上来说，**Cholesky分解的稳定性极佳**，只要矩阵确实是正定的厄米矩阵。

三对角矩阵线性方程组的求解

本节我们讨论一个三对角矩阵所给出的线性方程组的求解问题。这类问题会出现在不少的数值应用之中，例如后面会讨论到的利用三次样条函数进行拟合的问题，以及微分方程的边值问题。

$$A=\begin{pmatrix} a_1 & c_1 & \cdots & 0 \\ b_2 & a_2 & \ddots & \\ & \ddots & & c_{n-1} \\ 0 & & b_n & a_n \end{pmatrix} = LU$$

其中矩阵L和U分别是下和上双对角矩阵：

$$L=\begin{pmatrix} 1 & & \cdots & 0 \\ \beta_2 & 1 & \ddots & \\ & \ddots & \ddots & \\ 0 & & \beta_n & 1 \end{pmatrix}, \quad U=\begin{pmatrix} \alpha_1 & c_1 & \cdots & 0 \\ & \alpha_2 & \ddots & \\ & & \ddots & c_{n-1} \\ 0 & & & \alpha_n \end{pmatrix}$$

很容易证明矩阵U的副对角线元素与A的相同。另外参数α_i以及β_i与原先的系数之间的关系有下式给出：

$$\alpha_1 = a_1, \quad \beta_i = \frac{b_i}{\alpha_{i-1}}, \quad \alpha_i = a_i - \beta_i c_{i-1}, \quad i = 2, \cdots, n$$

三对角矩阵线性方程组的求解

这个算法又称为Thomas算法：

$$\alpha_1 = a_1$$

For $i = 2, \cdots, n$

$$\beta_i = \frac{b_i}{\alpha_{i-1}}$$

$$\alpha_i = a_i - \beta_i c_{i-1}$$

End

$$L = \begin{pmatrix} 1 & & \cdots & 0 \\ \beta_2 & 1 & \ddots & \\ & \ddots & \ddots & \\ 0 & & \beta_n & 1 \end{pmatrix}, \quad U = \begin{pmatrix} \alpha_1 & c_1 & \cdots & 0 \\ & \alpha_2 & \ddots & \\ & & \ddots & c_{n-1} \\ 0 & & & \alpha_n \end{pmatrix}$$

经过这个分解之后，我们可以利用该分解求解方程Ax=b。具体可以分成两步：

Ly=b，Ux=y

$$y_1 = b_1, \quad y_i = b_i - \beta_i y_{i-1}, \quad i = 2, \cdots, n$$

$$x_n = \frac{y_n}{\alpha_n}, \quad x_i = (y_i - c_i x_{i+1})/\alpha_i, \quad i = n-1, \cdots, 1$$

这个过程的计算量大约是**8n -7**。具体来说，
上述分解本身需要 3(n -1)，
而求解两个三角系统的计算量为 5n -4。

三对角矩阵线性方程组的变种

$$A=\begin{pmatrix} a_1 & c_1 & & b_1 \\ b_2 & a_2 & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ c_n & & b_n & a_n \end{pmatrix}, \quad Ax=f$$

$$A_1\begin{pmatrix} u_2 \\ u_3 \\ \cdots \\ u_n \end{pmatrix}=\begin{pmatrix} f_2 \\ f_3 \\ \cdots \\ f_n \end{pmatrix}, \quad A_1=\begin{pmatrix} a_2 & \ddots & \\ \ddots & \ddots & c_{n-1} \\ & b_n & a_n \end{pmatrix}$$

$$A_1\begin{pmatrix} v_2 \\ v_3 \\ \cdots \\ v_n \end{pmatrix}=\begin{pmatrix} -b_2 \\ 0 \\ \cdots \\ -c_n \end{pmatrix}$$

定义$x_i=u_i+x_1v_i, i=2,\cdots,n$很明显，这样定义列向量x后，**与$f_2,\cdots,f_n$对应的n-1个方程自动满足**。然后我们再来看与 f1 对应的方程

$$a_1x_1+c_1x_2+b_1x_n=f_1$$

$x_1=(f_1-c_1u_2-b_1u_n)/$
$(a_1+c_1v_2+b_1v_n)$

问题的敏感性和矩阵条件数

有了向量和矩阵范数的概念，下面分析线性方程组求解问题的敏感性和条件数，研究初始数据误差对解的影响。我们考虑方程组

$$A\vec{x} = \vec{b}$$

我们考虑方程右端发生扰动 $\Delta\vec{b}$ 的情况，相应的解变为 $\vec{x} + \Delta\vec{x}$,

$$A(\vec{x} + \Delta\vec{x}) = \vec{b} + \Delta\vec{b}$$

$$A\Delta\vec{x} = \Delta\vec{b} \quad \Rightarrow \quad \Delta\vec{x} = A^{-1}\Delta\vec{b} \quad \Rightarrow \quad ||\Delta\vec{x}|| \le ||A^{-1}|| \cdot ||\Delta\vec{b}||$$

$$A\vec{x} = \vec{b} \quad \Rightarrow \quad ||\vec{b}|| \le ||A|| \cdot ||\vec{x}||$$

根据问题条件数的定义，得到

$$\text{cond} = \frac{||\Delta\vec{x}||/||\vec{x}||}{||\Delta\vec{b}||/||\vec{b}||} = \frac{||\Delta\vec{x}|| \cdot ||\vec{b}||}{||\Delta\vec{b}|| \cdot ||\vec{x}||} \le ||A|| \cdot ||A^{-1}||$$

如果A是个厄米矩阵，而我们用的范数是p=2的欧氏模，那么

$$\text{cond} = \rho(A) \cdot \rho(A^{-1}) = |\lambda|_{\max}/|\lambda|_{\min}$$

病态问题的改进

本征值的数目与矩阵维数一致。一般来讲，一个超大矩阵(n~10^6)，这10^6个本征值的分布会出现下疏上密的情形。**如果我们能计算出最小的若干个本征值和本征矢量**，比方说前100个，然后把它们从原有系统中刨去，那么，**就能使$|\lambda|_{min}$大大增加，从而使得条件数大大减小**。

把矩阵A用正交归一本征矢量表示出来

$$A = \sum_{\lambda} \lambda |\lambda\rangle\langle\lambda|, \quad 1 = \sum_{\lambda} |\lambda\rangle\langle\lambda|$$

假设我们已知$|\lambda| \leq |\lambda_0|$时的所有本征值和本征矢量，并且$|\lambda_0| \gg |\lambda|_{min}$。我们可以构造投影算符

$$P_0 = \sum_{|\lambda|<|\lambda_0|} |\lambda\rangle\langle\lambda|, \quad 1 - P_0 = \sum_{|\lambda|>|\lambda_0|} |\lambda\rangle\langle\lambda|$$

然后我们把解方程组的事分成两步来完成

$$A\vec{x}_H = (1 - P_0)\vec{b}, \quad A\vec{x}_L = P_0\vec{b}$$

把 $\vec{x}_H$ 和 $\vec{x}_L$ 加起来就得到了方程组的解。

病态问题的改进

$$A\vec{x}_H = (1-P_0)\vec{b}, \quad A\vec{x}_L = P_0\vec{b}$$

把 $\vec{x}_H$ 和 $\vec{x}_L$ 加起来就得到了方程组的解。

对于 $\vec{x}_L$ 来讲，因为我们已经知道了最小的前n个本征值，所以我们可以直接得到

$$\vec{x}_L = A^{-1}P_0\vec{b} = \sum_{|\lambda|<|\lambda_0|} \lambda^{-1}|\lambda\rangle\langle\lambda|b\rangle$$

对于 $\vec{x}_H$ 来讲，我们需要解

$$A\vec{x}_H = (1-P_0)\vec{b}, \quad \Rightarrow \quad A\Delta\vec{x}_H = (1-P_0)\Delta\vec{b},$$

$$||\Delta\vec{x}_H|| \leq ||A^{-1}(1-P_0)|| \cdot ||\Delta\vec{b}|| \leq |\lambda_0|^{-1}||\Delta\vec{b}||$$

这样条件数就变成了 $\mathrm{cond} = |\lambda|_{\max}/|\lambda_0|$

所以线性方程组求解，或者矩阵求逆问题经常会与本征值和本征矢量的求解联系起来。我们在第6章会详细讲解本征值问题。

高斯消元过程中的舍入误差估计

我们来看LU分解中的舍入误差。假设A∈K^{n×n}。我们计算得到了下上三角矩阵^L和^U。^L和^U肯定是带有舍入误差的。于是

$$\hat{L}\hat{U} = A + H$$

这里H是用来衡量LU分解中舍入误差的大小。我们有一个结论

$$|H| \leq 2(n-1)u\left(|A| + |\hat{L}||\hat{U}|\right) + O(u^2), \quad u = \epsilon_M/2$$

这个误差分析告诉我们：

第一，对于LU分解，如果矩阵越大，也就是n越大，那么误差越大，这个误差正比于n。

第二，这个误差的大小与L和U这两个矩阵有关。假如我们不采用支点遴选，当我们的支点很接近0的时候，就意味着Frobinius矩阵的矩阵元很大，这个矩阵元会传递到下三角矩阵L中，从而使得整个LU的误差变大。

迭代法证明：

这件事情在 $n=1$ 的时候是显然成立的。对于 $n \geq 2$ 时，我们可以把 A 写成

$$A = \begin{pmatrix} \alpha & \omega^T \\ v & B \end{pmatrix}$$

这里的 $B \in K^{(n-1)\times(n-1)}$。根据高斯消元的算法，我们会有

$$\hat{z} = fl(v/\alpha), \quad \hat{C} = fl(\hat{z}\omega^T), \quad \hat{A}_1 = fl(B - \hat{C})$$

然后得到 $\begin{pmatrix} \alpha & \omega^T \\ 0 & \hat{A}_1 \end{pmatrix}$。这里的舍入误差有

$$\hat{z} = v/\alpha + f, \quad |f| \leq u|v/\alpha|$$
$$\hat{C} = \hat{z}\omega^T + F_1, \quad |F_1| \leq u|\hat{z}||\omega^T|$$
$$\hat{A}_1 = B - (\hat{z}\omega^T + F_1) + F_2, \quad |F_2| \leq u(|B| + |\hat{z}||\omega^T|) + O(u^2)$$

而另外一方面，对于 $\hat{A}_1$ 做 LU 分解，会得到

$$\hat{L}_1\hat{U}_1 = \hat{A}_1 + H_1$$

根据数学归纳法，我们假定已经得到了

$$H_1 \leq 2(n-2)u\left(|\hat{A}_1| + |\hat{L}_1||\hat{U}_1|\right)$$

这时，我们把 $\hat{L}$ 和 $\hat{U}$ 写出来就是

$$\hat{L} = \begin{pmatrix} 1 & 0 \\ \hat{z} & \hat{L}_1 \end{pmatrix}, \quad \hat{U} = \begin{pmatrix} \alpha & \omega^T \\ 0 & \hat{U}_1 \end{pmatrix}$$

然后根据 $\hat{L}\hat{U} = A + H$，我们可以得到

$$H = \begin{pmatrix} 0 & 0 \\ \alpha f & H_1 - F_1 + F_2 \end{pmatrix}$$

我们需要证明的是

$$|H| \leq 2(n-1)u \begin{pmatrix} 2|\alpha| & 2|\omega^T| \\ |v| + |\alpha||f| & |B| + |\hat{L}_1||\hat{U}_1| + |\hat{z}||\omega^T| + O(u^2) \end{pmatrix}$$

我们需要证明的是

$$|H| \leq 2(n-1)u \begin{pmatrix} 2|\alpha| & 2|\omega^T| \\ |v| + |\alpha||f| & |B| + |\hat{L}_1||\hat{U}_1| + |\hat{z}||\omega^T| + O(u^2) \end{pmatrix}$$

事实上我们只需要证明

$$|H_1| + |F_1| + |F_2| \leq 2(n-1)u \left(|B| + |\hat{L}_1||\hat{U}_1| + |\hat{z}||\omega^T| \right) + O(u^2)$$

我们只需要把关于 $|H_1|$, $|F_1|$ 和 $|F_2|$ 的不等式

$$|H_1| \leq 2(n-2)u \left(|B| + |\hat{z}||\omega^T| + |\hat{L}_1||\hat{U}_2| \right) + O(u^2)$$
$$|F_1| + |F_2| \leq u \left(|B| + 2|\hat{z}||\omega^T| \right) + O(u^2)$$

代入即可得到。

高斯消元过程中的舍入误差估计

对于三对角矩阵分解，Thomas算法，刘川老师的讲义里给出，它的舍入误差为

$$\hat{L}\hat{U} = A + H \quad \Rightarrow \quad |H| \le (4u + 3u^2 + u^3)|\hat{L}| \cdot |\hat{U}|$$

这里，我们并不要求证明。有一点需要说明的是，对普通的矩阵元，它的非零矩阵元个数为O(n^2)。因此，舍入误差在LU分解中通过多次运算，误差累加到O(n)的量级。**而对于三对角矩阵，它的非零矩阵元个数为O(n)，因此它是一个稀疏矩阵，无论在内存的要求还是运算的步骤，都大为缩减，而且它的舍入误差是O(1)的量级。因此Thomas算法的稳定性是非常不错的。**这里H的上限基本上由|^L|和|^U|矩阵中的最大矩阵元决定。只要各个系数β_i以及α_i不会太大。这些系数变大的一种可能是其中某个α_i非常接近于零。因此，如果矩阵的确是非奇异的，那么各个α_i必定不能等于零。但是如果矩阵接近于奇异，那么有可能造成算法出现不稳定的情况。如果矩阵A∈R^{n×n}是一个正定实对称矩阵，那么我们可以获得更好的估计，

$$|H| \le \frac{4u + 3u^2 + u^3}{1 - u}|A|.$$

这个分解更为稳定。

Krylov Method简介

Ax=b x=A^{-1}b

矩阵A的特征多项式(characteristic polynomial)

$$p(\lambda) = \det(A - \lambda I) = (-1)^n \left[\lambda^n + a_1\lambda^{n-1} + \cdots + a_n\right]$$

Cayley-Hamilton theorem

$$p(A) = (-1)^n \left[A^n + a_1 A^{n-1} + \cdots + a_n I\right] = 0.$$

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 2 \\ -1 & 2 & -1 \\ 1 & -1 & 2 \end{bmatrix}$$

特征多项式 $\lambda^3 - 6\lambda^2 + 8\lambda - 3 = 0$

$$\Rightarrow \quad A^3 - 6A^2 + 8A - 3I = 0$$

$$\Rightarrow \quad A^2 - 6A + 8I - 3A^{-1} = 0$$

A^{-1}可以用I，A，A^2 线性组合

Krylov Method简介

Ax=b x=A^{-1}b

Krylov matrix $\quad \mathbf{K}_j = [\, b \quad Ab \quad A^2b \quad \dots \quad A^{j-1}b \,].$

Krylov subspace $\quad \mathcal{K}_j = \textit{all combinations of } b, Ab, \dots, A^{j-1}b.$

这称为矢量b和矩阵A所生成的k+1维Krylov子空间

Aleksey Krylov

Official portrait (1910)

Native name	Russian: Алексей Николаевич Крылов
Born	August 3, 1863 O.S. (August 15, 1863 N.S.) Alatyrsky uezd of Simbirsk Gubernia, Russian Empire
Died	October 26, 1945 (aged 82) Leningrad, USSR

Thus $\mathcal{K}_j$ is the column space of $\mathbf{K}_j$. We want to choose the ***best combination*** as our improved x_j. Various definitions of "best" will give various x_j. Here are four different approaches to choosing a good x_j in $\mathcal{K}_j$—this is the important decision:

1. The residual $r_j = b - Ax_j$ is orthogonal to $\mathcal{K}_j$ (**Conjugate Gradients**).
2. The residual r_j has minimum norm for x_j in $\mathcal{K}_j$ (**GMRES** and **MINRES**).
3. r_j is orthogonal to a different space $\mathcal{K}_j(A^{\mathrm{T}})$ (**BiConjugate Gradients**).
4. The error e_j has minimum norm (**SYMMLQ**).

我们将在"方程求根与函数求极值"详细介绍CG算法

https://archive.siam.org/meetings/la09/talks/oleary.pdf

What was Lanczos' Role?

- 1950 paper: on an "iteration method" for the eigenproblem.
- Section 7: "method of minimized iterations". Develops an iteration based on choosing coefficients α_{kj} so that the norm of

$$v_{k+1} = Av_k - \sum_{j<k+1} \alpha_{kj} v_j$$

  is minimized.
- Shows that this yields an orthogonal basis (for what we now call the Krylov subspace) and a 3-term recurrence.
- develops a biorthogonalization algorithm when A is nonsymmetric.
- uses the recurrences to solve eigenproblems.

$$A = \begin{bmatrix} 1 & & & \\ & 2 & & \\ & & 3 & \\ & & & 4 \end{bmatrix} \qquad b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \qquad Ab = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \qquad A^{-1}b = \begin{bmatrix} 1/1 \\ 1/2 \\ 1/3 \\ 1/4 \end{bmatrix}$$

$$\mathbf{K_4} = V = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 1 & 4 & 16 & 64 \end{bmatrix}$$

K4 is a Vandermonde matrix

cond(V) ~1000

$$V = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} \\ 1 & x_3 & x_3^2 & \dots & x_3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & x_m^2 & \dots & x_m^{n-1} \end{bmatrix}$$