数据科学与工程数学基础

作业提交规范及第 12 次作业

教师：黄定江
助教：陈诺、刘文辉

2022 年 4 月 9 日

作业提交规范

1. 作业提交形式：**练习本或笔记本**（建议统一使用一般的**练习本**即可，不接收以纸张的方式书写的作业）。
2. 作业书写说明：
   (a) 可以讨论，禁止抄袭！
   (b) 练习本封面至少包含两方面信息：**姓名**和**学号**
   (c) 每一次的作业**请另起一页**，并在**第一行标明第几次作业**。例如“第 12 次作业”；
   (d) 每一题请**标注题号**，无需抄题，直接解答；
   (e) 题与题之间**请空一行**；
   (f) 不要求字好，但要求书写整体清晰易读。
3. 作业提交途径：纸质作业交给**学习委员**，由学习委员**按学号顺序**收齐后统一在截止日期前交到**助教实验室**。**单数周**布置的作业交到助教刘文辉处**数学馆西 109**；**双数周**布置的作业交到助教陈诺处**地理馆 353**。
4. 作业评分说明：正常提交作业的按照实际评分记录；逾期补交作业的根据逾期情况在实际评分基础上酌情扣分；未交作业的当次作业记为 0 分。

第 12 次作业

提交截至时间：**暂定 2022/04/08 下周五 20:00（晚上）**

理论部分 (线性方程组 1)

习题 1. 在作业 *10* 中介绍到，如下二元一次方程组

$$\begin{cases} 2x_1 + 4x_2 = 5 \\ 4x_1 + 5x_2 = 3 \end{cases}$$

可以通过 LU 分解求解。如果我们对 $A = \begin{pmatrix} 2 & 4 \\ 4 & 5 \end{pmatrix}$ 使用 LU 分解，则可得到

$$\begin{pmatrix} 2 & 4 \\ 4 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 2 & 4 \\ 0 & -3 \end{pmatrix}$$

利用该结果表示线性方程组的解。

解.

$$A = LU$$

$$Ax = b$$

$$LUx = b$$

$$x = U^{-1}L^{-1}b = \begin{pmatrix} 2 & 4 \\ 0 & -3 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 5 \\ 3 \end{pmatrix}$$

习题 2. 写出一种 LU 分解不能分解的矩阵 A，并分析该矩阵在线性方程组 $Ax = b$ 中时方程解集不可能出现的情况。

解.

定理 0.0.1.

矩阵$A \in \mathbb{R}^{n\times n}$ 能够进行LU 分解的充分必要条件是 A 的前 n 阶顺序主子式不为 *0*。

A 能进行 LU 分解需满足 A 的前 n 阶顺序主子式不为 *0*，即 A 可逆。

一个不能进行 LU 分解的矩阵如

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

如果我们对其使用 LU 分解，能得到

$$\begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \ell_{21} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix} = \begin{bmatrix} u_{11} & u_{12} \\ \ell_{21}u_{11} & \ell_{21}u_{12} + u_{22} \end{bmatrix}$$

求解该方程我们需要同时满足

$$u_{11} = 0$$

$$\ell_{21}u_{11} = 2$$

该方程无解，所以 A 无法进行 LU 分解。实际上，A 的一阶顺序主子式为 *0*。(当然，对于存在顺序主子式为 *0* 的矩阵 A，也可以进行 LUP 分解，其不在该题讨论范围内)。

习题 3. 以习题 *1* 中的线性方程组

$$\begin{cases} 2x_1 + 4x_2 = 5 \\ 4x_1 + 5x_2 = 3 \end{cases}$$

为例，其中 $A = \begin{pmatrix} 2 & 4 \\ 4 & 5 \end{pmatrix}, n = 2$，请问利用 *LU* 分解解该方程组的时间复杂度是否小于 $o(n^3)$？通过查阅资料，提出任一解该方程复杂度小于 $o(n^3)$ 的方法。*(*写出算法名称及大致流程，有能力的同学可以展开分析算法流程及复杂度*)*

解. 题目本意是优化矩阵分解解方程组中矩阵分解 *(*如 *LU* 分解*)* 步的复杂度。*LU* 分解求解线性方程组步骤分为对 A 进行 *LU* 分解，并求解 $LUx = b$，*python* 代码为

```
    import numpy as np
def linear_solve_without_pivoting(A, b):
    """x = linear_solve_without_pivoting(A, b) is the solution to A x = b (
      computed without pivoting)
       A is any matrix
       b is a vector of the same leading dimension as A
       x will be a vector of the same leading dimension as A
    """
    (L, U) = lu_decomp(A)
    x = lu_solve(L, U, b)
    return x
```

其中 *LU* 步复杂度为 $o(2/3 * n^3)$, 方程求解步复杂度为 $o(n^2)$，故 *LU* 分解解方程组时间复杂度为 $o(2/3*n^3)$，与 $o(n^3)$ 同阶。将复杂度降至小于 $o(n^3)$ 算法有如 *Solvay Strassen* 算法、*Coppersmith–Winograd* 算法等 *(*实际上是在降低矩阵乘法的复杂度*)*，参考链接如下：

https://courses.engr.illinois.edu/cs357/sp2020/notes/ref-9-linsys.html

https://stackoverflow.com/questions/8546756/matrix-multiplication-algorithm-time-complexity

https://math.stackexchange.com/questions/1330759/time-complexity-of-lu-decomposition

https://en.wikipedia.org/wiki/LU_decomposition