数值分析内部课程习题解答

第二章第一部分

SCAU DataHub

2025 年 3 月 29 日

目录

1 Polynomial interpolation

Solution 1.1. 参考本节一开始给出的例子，关注多项式的形式和 Vandermonde 矩阵.

(a) 4 次. 对于 $n-1$ 次多项式，其中就有 n 个系数. 我们已有的 5 组数据可以构成 5 个方程，所以至多可以解 5 个未知系数，因此插值多项式的最大次数是 $5-1=4$.

(b) 插值多项式的线性系统为

$$\begin{bmatrix} 1 & -1 & 1 & -1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 9 & 27 & 81 \\ 1 & 4 & 16 & 64 & 256 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 2 \end{bmatrix}$$

(c) 代码参考 Gitee 仓库.

Solution 1.2. 这里的插值虽然不同于前面，多了一些导数，但是原理还是一样的，即: 4 组数据 -> 构成 4 个方程 -> 最多解 4 个未知量 -> 多项式系数有 4 个 -> 构造 3 次多项式.

(a) 构造 3 次多项式

$$p(x) = c_1 + c_2 x + c_3 x^2 + c_4 x^3,$$

对于一阶导，有

$$p'(x) = c_2 + 2c_3 x + 3c_4 x^2.$$

因此 cubic 多项式的线性系统就是

$$p(x_1) = c_1 + c_2 x_1 + c_3 x_1^2 + c_4 x_1^3 \tag{1}$$
$$p'(x_1) = 0 + c_2 + 2c_3 x_1 + 3c_4 x_1^2 \tag{2}$$
$$p(x_2) = c_1 + c_2 x_2 + c_3 x_2^2 + c_4 x_2^3 \tag{3}$$
$$p'(x_2) = 0 + c_2 + 2c_3 x_2 + 3c_4 x_2^2 \tag{4}$$

, 注意到 x_1, x_2 和 $p(x_1), p'(x_1), p(x_2), p'(x_2)$ 都是已知的数据，待求解的未知量是 $c_1, \cdots, c_4$. 我们进一步写成矩阵的形式就是

$$\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 0 & 1 & 2x_1 & 3x_1^2 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 0 & 1 & 2x_2 & 3x_2^2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} p(x_1) \\ p'(x_1) \\ p(x_2) \\ p'(x_2) \end{bmatrix},$$

至于具体的数值，可以参考代码部分.

(b) 代码参考 Gitee 仓库.

Solution 1.3. 代码参考 Gitee 仓库.

Solution 1.4. 代码参考 Gitee 仓库.

2 Computing with matrices

Solution 2.1. 根据矩阵乘法的定义来算.

(a)

$$\mathbf{C}^2 = \mathbf{C}\mathbf{C} = \begin{bmatrix} \mathbf{I}-\mathbf{A} & \mathbf{A}+\mathbf{A}\mathbf{B} \\ -\mathbf{I}-\mathbf{B} & -\mathbf{A}+\mathbf{B}^2 \end{bmatrix}$$

(b)

$$\mathbf{C}^3 = \mathbf{C}^2 \times \mathbf{C} = \begin{bmatrix} \mathbf{I}-2\mathbf{A}-\mathbf{A}\mathbf{B} & \mathbf{A}-\mathbf{A}^2+\mathbf{A}\mathbf{B}+\mathbf{A}\mathbf{B}^2 \\ -\mathbf{I}-\mathbf{B}+\mathbf{A}-\mathbf{B}^2 & -\mathbf{A}-\mathbf{B}\mathbf{A}-\mathbf{A}\mathbf{B}+\mathbf{B}^3 \end{bmatrix}$$

Solution 2.2. 代码参考 Gitee 仓库.

Solution 2.3. 代码参考 Gitee 仓库.

Solution 2.4. 代码参考 Gitee 仓库.

Solution 2.5. 根据矩阵乘法的性质以及矩阵逆的定义证明.

所谓 $(\mathbf{AB})^{-1}$ 其实就是满足 $(\mathbf{AB})(\mathbf{AB})^{-1} = \mathbf{I}$ 式子的一个东西，那么 $(\mathbf{AB})\boxed{?} = \mathbf{I}$ 呢，我们知道

$$(\mathbf{AB})(\mathbf{B}^{-1}\mathbf{A}^{-1}) = \mathbf{A}(\mathbf{B}\mathbf{B}^{-1})\mathbf{A}^{-1} = \mathbf{A}\mathbf{I}\mathbf{A}^{-1} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I},$$

所以把 $(\mathbf{B}^{-1}\mathbf{A}^{-1})$ 代入到上面的 $\boxed{?}$ 是合适的，于是我们证明了 $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$.

Solution 2.6. 参考 Example 2.2.3，需要熟悉矩阵乘法的性质. 如果有困难，可以写一个 2×2 的矩阵出来，然后验算，这里的关键就在于，理解左/右乘一个列向量对矩阵的行/列的作用.

(a) 当我们想对 $\mathbf{B}$ 的列向量操作，就得考虑右乘

$$\mathbf{A} = \mathbf{B}\begin{bmatrix} 1 \\ 0 \\ -2 \end{bmatrix}.$$

(b) 当我们想对 $\mathbf{B}$ 的行向量操作，就得考虑左乘，并且是单位列向量的转置

$$\mathbf{A} = \begin{bmatrix} \mathbf{e_4}^T & \mathbf{e_3}^T & \mathbf{e_2}^T & \mathbf{e_1}^T \end{bmatrix}\mathbf{B}$$

(c) 综合了前面两问

$$\mathbf{A} = \mathbf{B}\begin{bmatrix}\mathbf{e_1} & 2\mathbf{e_2} & 3\mathbf{e_3}\end{bmatrix}$$

(d) 定义 $\mathbf{1}$ 是全为 1 的列向量，于是

$$\mathbf{A} = \mathbf{1}^T\mathbf{B}\mathbf{1}$$

Solution 2.7. 按照矩阵乘法的定义来证.

(a) 对任意给定的 n 维列向量 $\mathbf{v}$ 和 $\mathbf{w}$，我们可以展开具体形式

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix},$$

根据矩阵乘法的定义

$$\mathbf{v}^T\mathbf{w} = \sum_{i=1}^{n} v_i w_i, \quad \mathbf{w}^T\mathbf{v} = \sum_{i=1}^{n} w_i v_i,$$

可见二者是相等的.

(b) 命题不成立. 因为 $\mathbf{v}\mathbf{w}^T = (\mathbf{w}\mathbf{v}^T)^T$，与 $\mathbf{w}\mathbf{v}^T$ 是一个转置的关系，所以不一定相等. 反例容易举.

3 Linear systems

Solution 3.1. 只需使系数矩阵与增广矩阵的秩不一样就可以了.

例如令 $\mathbf{b} = [1, 1]^T$，那么系数矩阵和增广矩阵分别为

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix},$$

前者秩为 1，后者秩为 2，容易验证方程无解.

Solution 3.2. 计算过程参考本节 Triangular systems，验证结果用下一题的程序.

Solution 3.3. 代码参考 Gitee 仓库.

Solution 3.4. 这题给了一个现实场景，很有意思，里面的三对角矩阵是我们 PDE 数值解的常客.

(a) 线性系统 $\mathbf{Aq}=\mathbf{f}$ 具体形式为

$$\begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ 0 & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_{n-2} \\ q_{n-1} \end{bmatrix} = \begin{bmatrix} \frac{m_1 g}{n\tau} \\ \frac{m_2 g}{n\tau} \\ \frac{m_3 g}{n\tau} \\ \vdots \\ \frac{m_{n-2} g}{n\tau} \\ \frac{m_{n-1} g}{n\tau} \end{bmatrix}$$

(b) 代码参考 Gitee 仓库.

(c) 代码参考 Gitee 仓库.

Solution 3.5. 代码参考 Gitee 仓库.

Solution 3.6. 代码参考 Gitee 仓库.

4 LU factorization

Solution 4.1. 按照正文中给出的算法即可计算得到，下面直接给出结果，以供参考:

(a)

$$\mathbf{L}=\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 2 & -2 & 1 \end{bmatrix}, \quad \mathbf{U}=\begin{bmatrix} 2 & 3 & 4 \\ 0 & -1 & 2 \\ 0 & 0 & -2 \end{bmatrix}$$

(b)

$$\mathbf{L}=\begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ -2 & -2 & 1 & 0 \\ -1 & 1 & -2 & 1 \end{bmatrix}, \quad \mathbf{U}=\begin{bmatrix} 6 & -2 & -4 & 4 \\ 0 & -2 & -4 & -1 \\ 0 & 0 & 5 & -2 \\ 0 & 0 & 0 & 8 \end{bmatrix}$$

Solution 4.2. 代码参考 Gitee 仓库.

Solution 4.3. 代码参考 Gitee 仓库.

Solution 4.4. 代码参考 Gitee 仓库.

Solution 4.5. 代码参考 Gitee 仓库.

Solution 4.6. (a) 根据行列式的性质

$$\det(\mathbf{AB})=\det(\mathbf{A})\cdot\det(\mathbf{B})$$

因此给定 $\mathbf{A}=\mathbf{LU}$，我们有

$$\det(\mathbf{A})=\det(\mathbf{LU})=\det(\mathbf{L})\cdot\det(\mathbf{U})$$

由于 $\mathbf{L}$ 的对角元素为 1, 而 $\mathbf{U}$ 的对角元素为 $U_{11}, \cdots, U_{nn}$, 因此

$$\det(\mathbf{A}) = 1 \cdot \prod_{i=1}^{n} U_{ii} = \prod_{i=1}^{n} U_{ii}.$$

得证.

(b) 代码参考 Gitee 仓库.

5 Efficiency of matrix computations

Solution 5.1. 除了 (a) 以外，(b, c, d) 都非常容易验证，因此本解答只给出问 (a) 的解答.

(a) 不失一般性的，我们取对数函数 $\log x$ 为 $\ln x$，令 $h(x) = \frac{\ln x}{x}$，求导可得

$$h'(x) = \frac{1}{x^2}(1 - \ln x)$$

显然，函数 $h(x)$ 在 $x = e$ 处取得极大值，且当 $0 < x \leq e$ 时，函数 $h(x)$ 单调增加，$x > e$ 时，函数 $h(x)$ 单调下降，注意到 $h(e) = 1/e$，因此，$\frac{n}{\ln n} \geq e$，进一步的

$$\frac{n^2}{\ln n} \geq ne \to \infty, \text{as}\, n \to \infty,$$

因此 $n^2 \neq O(\ln n)$.

Solution 5.2. 除了 (a) 以外，(b, c, d) 都容易验证，因此本解答只给出 (a) 的解答.

(a) 当 $h \to 0$ 时，$\ln h \to -\infty$，因此 $\ln h < -100$, 那么

$$\frac{\ln h}{h} \leq -100\frac{1}{h} \to -\infty, \text{as}\, h \to 0,$$

从而 $h^2 \ln h \neq O(h^3)$.

Solution 5.3. 两个 n 维向量相乘的乘法次数是 n，加法次数是 $n-1$，所以所有浮点数运算数量为 $2n-1$.

Solution 5.4. 两个 $n \times n$ 维矩阵相乘的元素数量为 $n \times n$，一个元素的浮点数运算数量是 $2n-1$，那么全部的运算数为 $n^2(2n-1) = 2n^3 - n^2 \sim 2n^3$.

Solution 5.5. (a) 每一次循环有 2 次乘法和 1 次加法，因此总共的浮点数运算为

$$\sum_{i=2}^{n}(2+1) = 3n - 3.$$

(b) 针对 horner 算法，每一次循环有 1 次乘法和 1 次加法，因此总共的浮点数运算为

$$\sum_{i=1}^{n-1}(1+1) = 2n - 2.$$

比起 (a) 中的算法有 $n-1$ 的浮点数运算节省.

Solution 5.6. (a) 总共的浮点数运算为

$$\begin{aligned}\sum_{k=1}^{n-1} &= \big(n-k+1+2(n-k+1)^2\big)\\ &= \sum_{j=1}^{n-1}(3+5j+2j^2)\\ &= 3(n-1)+\frac{5n(n-1)}{2}+\frac{n(n-1)(2n-1)}{3}.\end{aligned}$$

(b) 代码参考 Gitee 仓库.

Solution 5.7. 我们对 p 用数学归纳法证明命题成立，根据已知结论，$p=0,1,2$ 时，命题成立，假设 $p\le s-1$ 时，命题成立，即

$$\sum_{k=0}^{n-m} k^p \sim \frac{n^{p+1}}{p+1}, \quad (p\le s-1).$$

不难看出

$$(n-m)^{s+1} = \sum_{k=1}^{n-m}(k^{s+1}-k^s).$$

进一步化简等式右端，可得上面的等式的变形

$$\begin{aligned}(n-m)^{s+1} &= \sum_{k=1}^{n-m}\sum_{i=0}^{s}\binom{s+1}{i}k^i(-1)^{s-i}\\ &= (s+1)\sum_{k=1}^{n-m}k^s+\sum_{i=0}^{s-1}(-1)^{s-i}\binom{s+1}{i}\sum_{k=1}^{n-m}k^i\end{aligned}$$

等式两边同时除以 n^{s+1}，得到

$$\frac{(n-m)^{s+1}}{n^{s+1}} = (s+1)\frac{\sum_{k=1}^{n-m}k^s}{n^{s+1}}+\frac{\sum_{i=0}^{s-1}(-1)^{s-i}\binom{s+1}{i}\sum_{k=1}^{n-m}k^i}{n^{s+1}}$$

令 $n\to\infty$ 并结合归纳假设可得

$$\sum_{k=1}^{n-m}k^s \sim \frac{n^{s+1}}{s+1}.$$

因此 $p=s$ 时，命题正确，根据数学归纳法可知，原命题对于任意的非负整数 m,p 成立.

Solution 5.8. 在 Julia 中，LowerTriangular 和 tril 是不一样的，前者专为加速求下三角形解线性方程组 $\mathbf{Lx}=\mathbf{b}$ 服务，而后者则是取矩阵的下三角，功能与 Numpy 中的 tril 一致，但是 LowerTriangular 在 Python 以及 Numpy 里面没有, UpperTriangular 和 triu 同理, 所以本题不作要求.