

计算物理学 (A) 第一次作业

第三题与编程相关部分

王子毓

2020.10.30

wangziyu15@pku.edu.cn

Contents

Contents

1. Hilbert 矩阵

2. 编程相关

Hilbert 矩阵

1. Hilbert 矩阵

1.1 矩阵 H_n 矩阵元的表达式和矢量 b 的表达式

1.2 证明 H_n 对称正定且非奇异

1.3 估计 H_n 的行列式

1.4 求解线性方程组

2. 编程相关

矩阵 H_n 矩阵元的表达式和矢量 b 的表达式

为求 D 的极值, 在表达式中对 c_i 求偏导:

$$\frac{\partial D}{\partial c_i} = \int_0^1 \left(\sum_{j=1}^n c_j x^{j-1} - f(x) \right) x^{i-1} dx, \quad (1)$$

$$= 2(i-1) \int_0^1 \sum_{j=1}^n c_j x^{j+i-2} - x^{i-1} f(x) dx, \quad (2)$$

$$= 2(i-1) \left(\sum_{j=1}^n \frac{c_j}{i+j-1} - \int_0^1 x^{i-1} f(x) dx \right). \quad (3)$$

矩阵 H_n 矩阵元的表达式和矢量 b 的表达式

令上式等于零，可得一组关于 $c_i, i = 1, 2, 3, \dots, n$ 的线性方程组：

$$\sum_{j=1}^n \frac{c_j}{i+j-1} = \int_0^1 x^{i-1} f(x) dx. \quad (4)$$

与题中所给的形式比较可知：

$$(H_n)_{ij} = \frac{1}{i+j-1}, \quad (5)$$

$$b_i = \int_0^1 x^{i-1} f(x) dx. \quad (6)$$

评分标准：给出最终结果满分， H_{ij} 表达式未积分扣 2 分。

1. Hilbert 矩阵

1.1 矩阵 H_n 矩阵元的表达式和矢量 b 的表达式

1.2 证明 H_n 对称正定且非奇异

1.3 估计 H_n 的行列式

1.4 求解线性方程组

2. 编程相关

证明 H_n 对称正定且非奇异

$$\mathbf{c}^T \mathbf{H}_n \mathbf{c} = c_i (\mathbf{H}_n)_{ij} c_j = \sum_{i,j=1}^n \frac{c_i c_j}{i+j-1}, \quad (7)$$

$$= \int_0^1 \sum_{i,j=1}^n c_i c_j x^{i+j-2} dx, \quad (8)$$

$$= \int_0^1 \left(\sum_{i=1}^n c_i x^{i-1} \right)^2 dx \geq 0. \quad (9)$$

- 等号当且仅当 $c_i = 0$ 时成立，因此 Hilbert 矩阵正定。
- 存在非奇异相似变换使之变为对角元均大于零的对角矩阵，其行列式大于零，矩阵非奇异 (随便说就行)。

评分标准：基本全对，个别同学扣 1 分是因为过于简略。

1. Hilbert 矩阵

1.1 矩阵 H_n 矩阵元的表达式和矢量 b 的表达式

1.2 证明 H_n 对称正定且非奇异

1.3 估计 H_n 的行列式

1.4 求解线性方程组

2. 编程相关

估计 H_n 的行列式

对题目中的公式取对数，可得：

$$\ln \det(\mathbf{H}_n) = 4 \ln c_n - \ln c_{2n}. \quad (10)$$

根据题目中所给公式，可以求得 $\ln c_n$ ：

$$\ln c_n = \sum_{i=1}^{n-1} \ln i!. \quad (11)$$

为了估计大 n 时的情形，可使用 Stirling 公式：

$$\ln n! \approx n(\ln n - 1) + \frac{1}{2} \ln 2\pi n. \quad (12)$$

估计 H_n 的行列式

近似方法一

用积分代替求和，计算得：

$$\ln c_n \approx \int_1^{n-1} dx \, x(\ln x - 1) + \frac{1}{2} \ln 2\pi x, \quad (13)$$

$$= \frac{1}{2} n^2 \ln(n-1) + \mathcal{O}(n^2). \quad (14)$$

于是 Hilbert 矩阵行列式为：

$$\det(\mathbf{H}_n) \sim \exp \left(2n^2 \log \frac{n-1}{2n-1} \right), \quad (15)$$

$$\sim 4^{-n^2}. \quad (16)$$

估计 H_n 的行列式

近似方法二

考虑对 $\ln \det(\mathbf{H}_{n+1}) - \ln \det(\mathbf{H}_n)$ 做近似:

$$\begin{aligned}\ln \det(\mathbf{H}_{n+1}) - \ln \det(\mathbf{H}_n) &= 4 \ln n! - \ln(2n+1)! - \ln(2n)!, \\ &\approx -4n \ln 2 + \mathcal{O}(\ln n).\end{aligned}\quad (17)$$

又因为 $\ln \det(\mathbf{H}_1) = 0$, 求和得:

$$\ln \det H_n \sim -2n^2 \ln 2 + \mathcal{O}(n), \quad (18)$$

$$\det H_n \sim 4^{-n^2}. \quad (19)$$

也可用其他方法合理近似。

估计 H_n 的行列式

评分标准

- 大 n 近似下的结果中包含领头阶贡献 4^{-n^2} 就给满分。
- 用其他方法进行了估计，但没有得到以上结果的，根据推导情况酌情扣分，最多扣 3 分。
- 给出 $n \geq 10$ 以上部分 n 数值的结果且数量级与严格计算结果相差不太大（差几个数量级以内）的酌情扣分，最多扣 3 分。
- 只给出 $1 \leq n \leq 10$ 结果或者给出的表达式仍保留求和或累乘的，根据做近似的情况扣分，至少扣 3 分。（因为题目写了 $n \geq 10$ ）

1. Hilbert 矩阵

1.1 矩阵 H_n 矩阵元的表达式和矢量 b 的表达式

1.2 证明 H_n 对称正定且非奇异

1.3 估计 H_n 的行列式

1.4 求解线性方程组

2. 编程相关

求解线性方程组

算法实现

老师讲义伪代码都给了，抄就完事。

误差分析

言之有理即可，方法例如：

- 与用 Mathematica 或 Matlab 给出的严格解比较。
- 把解代回方程检查左边与右边的差 $|\mathbf{H}x - b|$ 。

求解线性方程组

评分标准

- 正确实现两种算法，给出求解误差的估计并大致分析了原因（误差依赖具体实现和估计方法，实事求是言之有理即可），满分。
- 正确实现两种算法，但误差估计缺乏依据或者没有分析原因，扣 1-2 分。
- 程序不能给出正确结果，每个不正确的算法扣 3 分。
- 没有提交源代码、源代码抄袭或与题目无关，0 分。
- * 程序编写、编译方面存在问题，但助教 Debug 后能看到正确结果，这次不扣分。

编程相关

一些常识

编译器 (Compiler)

- 作用：将编译型语言 (如 C / C++) 的源代码编译、连接成可执行文件。
- 常见有 GCC, Clang, MSVC 等。

解释器 (Interpreter)

- 作用：逐行执行解释型语言 (如 Python) 的源文件。

集成开发环境 (IDE)

- 辅助编写和调试代码的软件。
- 常见有 CLion, Visual Studio, Xcode, CodeBlocks 等。

助教是如何检查作业的

一些要求

- 只需提交**源代码**，不要 IDE 工程文件。
- 编译说明需指明 C / C++ 版本以及需要编译哪些文件，可以用 *nix makefile 替代。
- 使用 jupyter notebook 的同学请同时提交.ipynb 文件，不要只提交 pdf 或 html 文件。
- 程序直接输出**所有**解答中用到的结果，不需要设计交互式输入。¹
- 作业解答请提交 PDF 文件，建议使用 \LaTeX 编写。²

¹C / C++ 在 main 里按照题目要求的参数调用求解函数，Python 可以写在最后的 `if __name__ == '__main__':` 里。

²如果觉得输入公式很累可以试试 Mathpix Snipping tool。

如何提高编程、Debug 效率

- **命名风格：**保证写完过段时间还能看懂你想干嘛。
Python 可参考 PEP 8，C / C++ 选个符合你品味的。
- **定义函数：**一个函数只做一件事，不要全塞 main 里。
便于复用和减小代码之间的耦合。
- **面向对象：**合理用类抽象可减小代码耦合提升可读性，
但为了面向对象而面向对象可能会害死自己。³

³非课程要求，量力而行。对面向对象的批评可参考*Bad Engineering Properties of Object-Oriented Languages, Why OO Sucks* by Joe Armstrong等。

如何提高编程、Debug 效率

- **用新版语言：**语言新加入的特性往往能提升开发效率。
例如 C++11 之后的 “Modern C++”。
- **合适的工具：**顺手的 IDE 或代码编辑器 + 插件⁴。
如果不知道用什么，可以试试 JetBrains 全家桶，
educational license 免费。
- **告别谭浩强：**如果之前只学过谭氏 C 与谭 ++，
有空建议看看更好、更现代的教程。
例如 *C++ Primer*, *C Primer Plus*, 《C 和指针》等。

⁴删掉 VC6.0 并打死叫你用 VC6.0 的人。

如何提高程序运行效率

- **合理的算法实现：**以 GEM 中对一列进行消元为例，Forbinus 矩阵和 naive 矩阵乘法： $\mathcal{O}(n^3)$ ，直接两行相减： $\mathcal{O}(n^2)$ 。
- **开启编译器优化：**这是效果拔群的黑魔法。
- **并行计算：**使用 OpenMP, MPI 等实现并行，或使用 GPU 并行。⁵

⁵非课程要求。

一些可能有用的网站

- <https://cppreference.com>
- <https://www.python.org>
- <https://stackoverflow.com>