大学数学实验

Experiments in Mathematics

第4讲

数值计算III 线性代数方程组的数值解法

实验4的主要内容

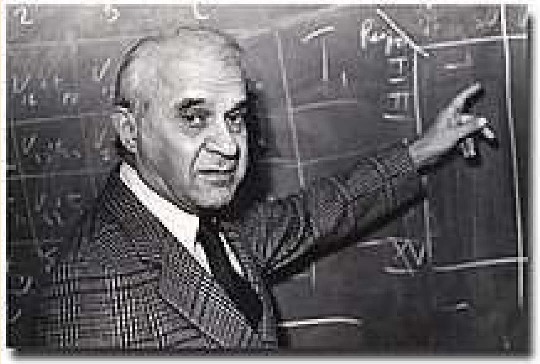
1. 线性方程组举例

2. 数值解法与误差分析:

高斯消去法（直接方法)
误差与扰动，病态方程组
迭代方法

3. 线性方程组数值解法的MATLAB实现

1. 经济学中的线性方程组计算问题



- Wassily Leontief　哈佛大学教授
- 1973年的诺贝尔经济学奖

- 他把美国经济分解为500个部门，
- 例如煤炭工业、汽车工业、交通系统等。
- 对每个部门考察该部门的产出如何分配给其他
- 经济部门，利用这些关系列出线性方程组。

- 1949年夏， Leontief用Mark II计算机进行计算，
- 当时的计算机还无法处理500个未知量的线性方程组，
  Leontief只好将问题简化为42个未知量的线性方程组，

求解这个方程组，MarkII用了56小时

诺贝尔经济学奖（1969年起）

1973	华西里·列昂惕夫 Wassily Leontief （美国）	发展了投入产出方法，该方法在许多重要的经济问题中得到运用	美国哈佛大学	投入产出分析
1975	列奥尼德·康托罗维奇 Leonid Vitaliyevich Kantorovich（苏联）	前者在1939年创立了享誉全球的线性规划单纯形算法，后者将数理统计学成功运用于经济计量学他们对资源最优分配理论做出了贡献	俄罗斯科学院	资源优化配置理论
	佳林·库普曼斯 Tjalling C. Koopmans （美国）		美国耶鲁大学	

我的个人电脑上，用Matlab计算的测试结果

n	时间（秒）
42	2.1×10^{-5}
100	7.3×10^{-5}
200	2.8×10^{-4}
400	1.4×10^{-3}
800	5.7×10^{-3}
1600	0.038
3200	0.25
6400	1.44
12800	9.71

摘自

Nick Higham

曼彻斯特大学

Turing， Wilkinson
and
Gaussian Elimination

1996年

Machine	Year	n	Time
Logarithm tables	c. 1884	29	7 weeks
Desk computing equipment	c. 1946	18	2 weeks
Harvard Mark 1	1947	10	45 minutes
IBM 602 Calculating Punch	1949	10	4 hours
Pilot ACE	1951	17	over 3 hours
Pilot ACE (mag. drum)	1954	30	$1\frac{1}{2}$ mins
ACE	1958	30	5 seconds
EDSAC 2	1960	31	4 seconds
EDSAC 2 (mag. tape)	1960	100	7 minutes
MATLAB on DX2-66	1996	100	0.4 secs

$\mathbf{7.3 \times 10^{-5}}$

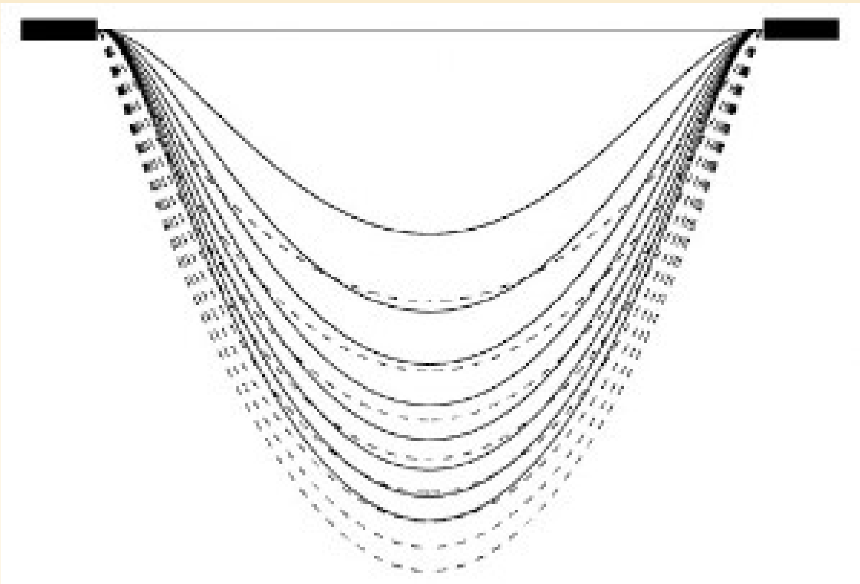
Largest Linear Systems Solved

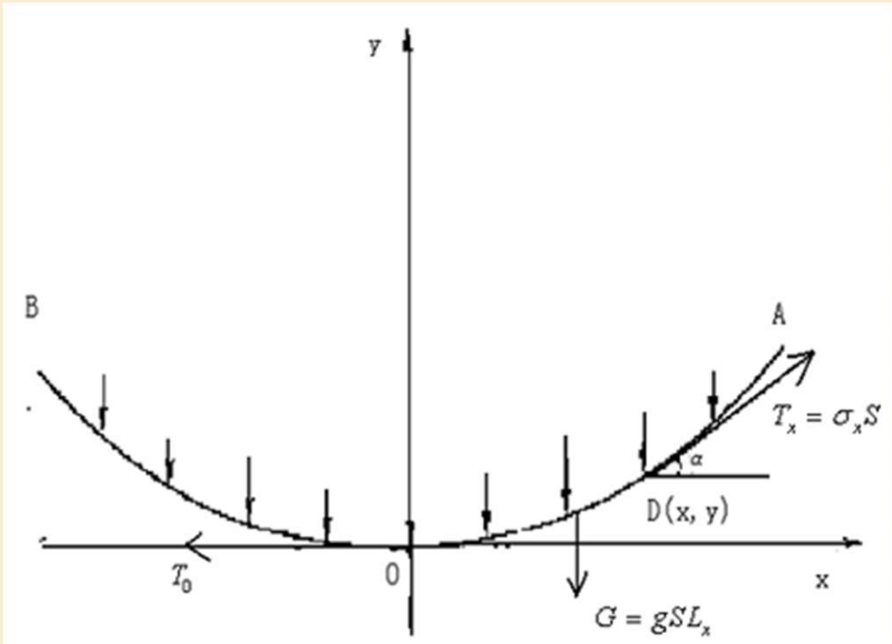
Year	n	Computer	Time
1991	55, 296	Connection Machine CM-2	4.4 days
1992/3	75, 264	Intel iPSC/860	$2\frac{2}{3}$ days
1994	76, 800	Connection Machine CM-5	4.1 days
1995	128, 600	Intel Paragon	$\approx$ 1 hour

2 常微分方程边值问题

$$y'' + p(x)y' + q(x)y = r(x), x \in [a,b]$$

$$y(a) = \alpha, y(b) = \beta$$





悬链线

数值微分

用离散方法近似计算函数 $y=f(x)$ 在某点 $x=a$ 的导数值

$$f'(a)\cong\frac{f(a+h)-f(a)}{h}$$ 前差公式 误差为 $O(h)$

$$f'(a)\cong\frac{f(a)-f(a-h)}{h}$$ 后差公式 误差为 $O(h)$

$$f'(a)\cong\frac{f(a+h)-f(a-h)}{2h}$$ 中点公式 误差为 $O(h^2)$

泰勒展开: $$f(a\pm h)=f(a)\pm hf'(a)+\frac{h^2}{2}f''(a)\pm O(h^3)$$

$$f''(a)\approx\frac{\frac{f(a+h)-f(a)}{h}-\frac{f(a)-f(a-h)}{h}}{h}=\frac{f(a+h)-2f(a)+f(a-h)}{h^2}$$

差分方程的建立

对区间$[a,b]$作等距分划：$x_j = a + jh(j = 0,1,2,...n)$

$h = \dfrac{b-a}{n}$。由数值微分公式

$$y'(x_j) = \frac{y(x_{j+1}) - y(x_{j-1})}{2h} - \frac{h^2}{6} y^{(3)}(\xi_j)$$

$$y''(x_j) = \frac{y(x_{j+1}) - 2y(x_j) + y(x_{j-1})}{h^2} - \frac{h^2}{12} y^{(4)}(\eta_j)$$

其中,$\xi_j, \eta_j \in (x_{j-1}, x_{j+1})$。

差分方程的建立

代入$y''+p(x)y'+q(x)y=r(x), x\in[a,b]$得差分方程:

$$\frac{y_{j+1}-2y_j+y_{j-1}}{h^2}+p_j\frac{y_{j+1}-y_{j-1}}{2h}+q_jy_j=r_j$$

$$(j=1,2,\cdots,n-1)$$

这是求$y_j(j=0,1,2,...,n)$的$n-1$个方程,还缺的两个方程由边界条件给出。

差分方程的建立

对于第一类边界条件：$y_0=\alpha, y_n=\beta$,即已给出两个未知量的解,这时整理后有

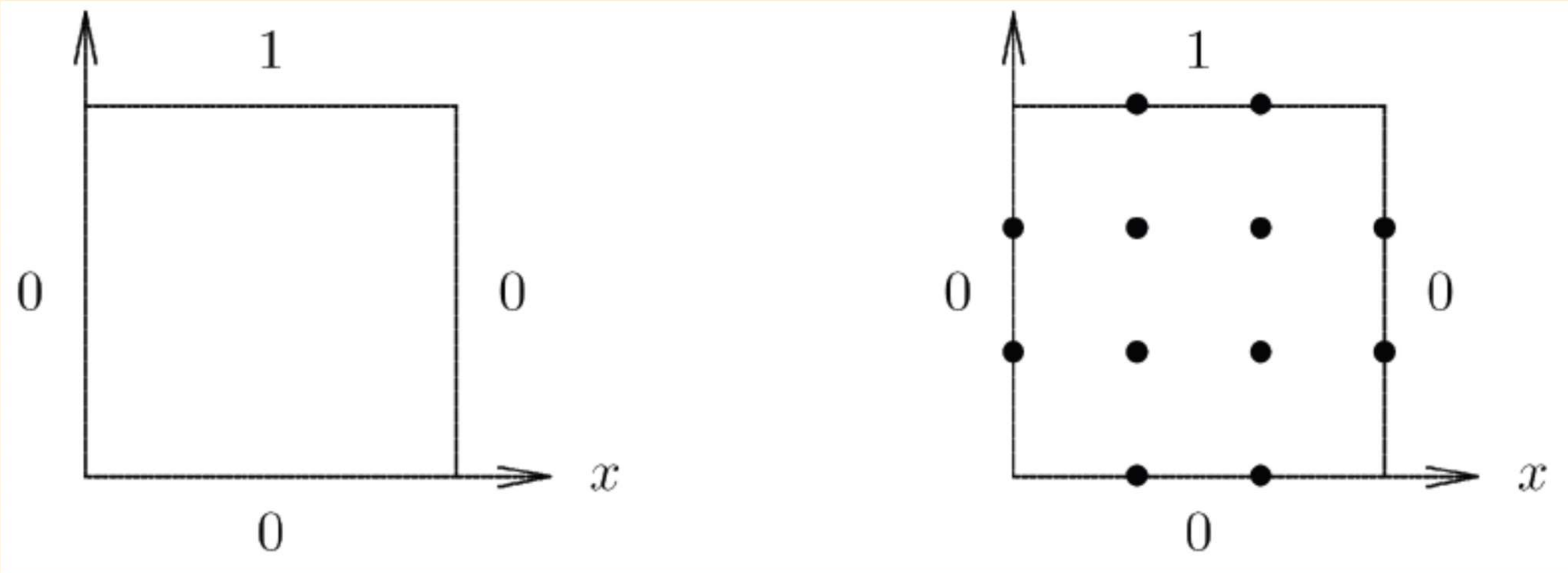
$$\begin{bmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-2} & b_{n-2} & c_{n-2} \\ & & & a_{n-1} & b_{n-1} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-2} \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} d_1 - a_1\alpha \\ d_2 \\ \vdots \\ d_{n-2} \\ d_{n-1} - c_{n-1}\beta \end{bmatrix}$$

其中$\begin{cases} a_j = 1 - \dfrac{h}{2} p_j; b_j = -2 + h^2 q_j \\ c_j = 1 + \dfrac{h}{2} p_j; d_j = h^2 r_j \end{cases}$。

3 Laplace 方程和泊松方程

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=f(x,y)$$

泊松方程（Poisson ‘s equation），
常见于静电学、机械工程等领域

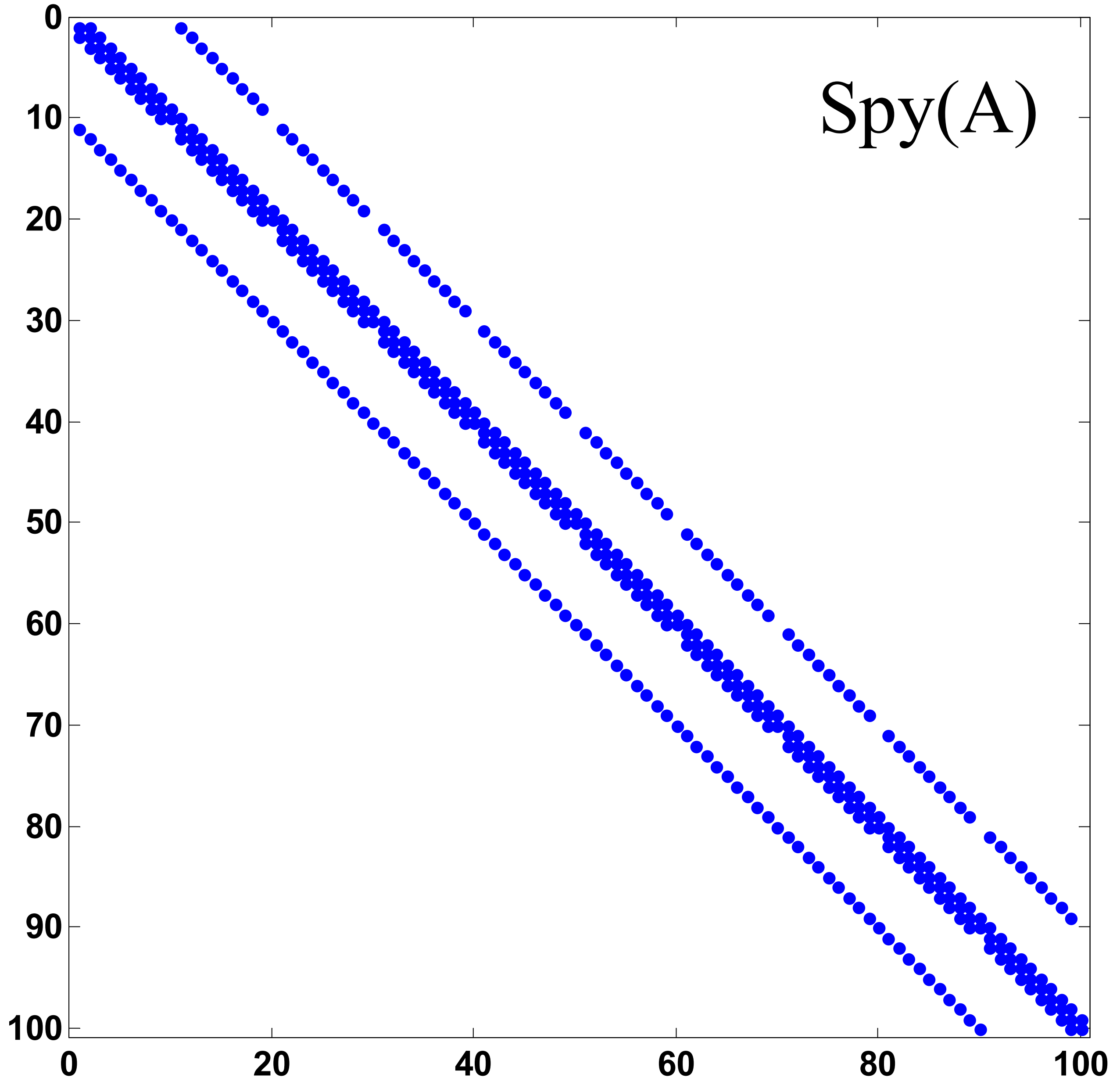


五点差分格式，以前图为例

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0$$

$$\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{h^2}+\frac{u_{i,j+1}-2u_{i,j}+u_{i,j-1}}{h^2}=0$$

$$\begin{aligned}
4u_{1,1}-u_{0,1}-u_{2,1}-u_{1,0}-u_{1,2} &= 0,\\
4u_{2,1}-u_{1,1}-u_{3,1}-u_{2,0}-u_{2,2} &= 0,\\
4u_{1,2}-u_{0,2}-u_{2,2}-u_{1,1}-u_{1,3} &= 0,\\
4u_{2,2}-u_{1,2}-u_{3,2}-u_{2,1}-u_{2,3} &= 0.
\end{aligned}$$


Spy(A)
0
10
20
30
40
50
60
70
80
90
100
0
20
40
60
80
100

线性方程组的一般形式、两类解法

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\
\cdots\cdots & \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n
\end{aligned}
$$

或　AX=b

直接法　经过有限次算术运算求出精确解（实际上由于有舍入误差只能得到近似解）---- 高斯（Gauss）消元法及与它密切相关的矩阵LU分解

迭代法　从初始解出发，根据设计好的步骤用逐次求出的近似解逼近精确解 ---- 雅可比（Jacobi）迭代法和高斯—塞德尔（Gauss—Seidel）迭代法

直接法---高斯消元法

消元过程

$$\begin{aligned} a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 + \cdots + a_{1n}^{(1)}x_n &= b_1^{(1)} \\ a_{22}^{(2)}x_2 + \cdots + a_{2n}^{(2)}x_n &= b_2^{(2)} \\ \cdots\cdots \\ a_{n2}^{(2)}x_2 + \cdots + a_{nn}^{(2)}x_n &= b_n^{(2)} \end{aligned}$$

$$\begin{aligned} a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 + \cdots\cdots + a_{1n}^{(1)}x_n &= b_1^{(1)} \\ a_{22}^{(2)}x_2 + \cdots\cdots + a_{2n}^{(2)}x_n &= b_2^{(2)} \\ \cdots\cdots \\ a_{nn}^{(n)}x_n &= b_n^{(n)} \end{aligned}$$

回代过程

条件

$$a_{kk}^{(k)} \neq 0$$

$$(k = 1,2,\cdots,n)$$

高斯消去法的矩阵表示

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{bmatrix}$$

$$A = LU$$

$Ax = b$ 等价为 $LUx = b$

先解 $Ly = b$

再解 $Ux = y$

若A为对称正定矩阵 $A = LL^T$ （*Cholesky*分解）

线性方程组直接法的MATLAB实现

1. 求解 $Ax=b$ 用左除： $x=A\backslash b$ 。

若A为可逆方阵，输出原方程的解$\boldsymbol{x}$

若A为$n\times m$矩阵（$n>m$），且A^TA可逆，输出原方程的最小二乘解$\boldsymbol{x}$

2. 矩阵LU分解

`[x,y]=lu(A)` 若 A 可逆且顺序主子式不为零， 输出 x 为单位下三角阵 L，y 为上三角阵 U，使 $A=LU$ ； 若 A 可逆，x 为一交换阵与单位下三角阵之积.

线性方程组数值解法的MATLAB实现

`[x,y,p]=lu(A)`　若 A 可逆，输出 x 为单位下三角阵 L，y 为上三角阵 U，p 为一交换阵 P，使 $PA=LU$.

`u =chol(A)`　对正定对称矩阵 A 的 Cholesky 分解，输出 u 为上三角阵 U，使 $A=U^TU$

LU 分解和GE的Matlab实现

- A=hilb(5);
- [L,U,P]=lu(A);
- [norm(L*U-A), norm(L*U-P*A)]

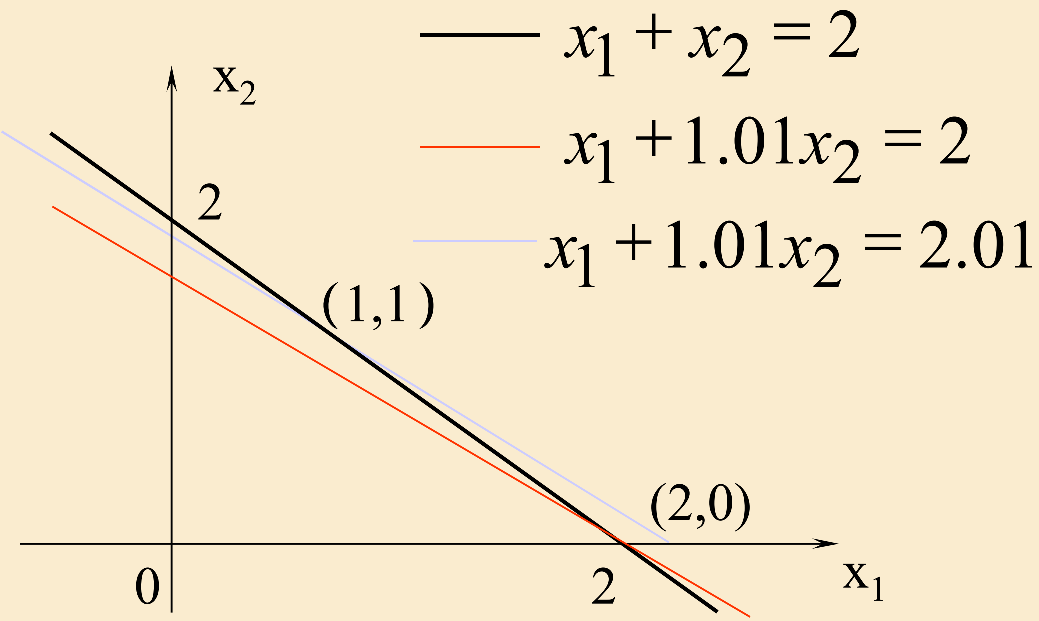
ans =[0.2947   0.0000]

b=A*ones(5,1); A\b

直 接 法 – 误 差 分 析

$$A \quad x = b$$

$$\begin{bmatrix} 1 & 1 \\ 1 & 1.01 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \Rightarrow x = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$



$$b = \begin{bmatrix} 2 \\ 2.01 \end{bmatrix} \Rightarrow x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

x对b的扰动敏感

$Ax = b$，如果解 $\boldsymbol{x}$ 对 $\boldsymbol{b}$ 或 $\boldsymbol{A}$ 的扰动敏感，就称方程组是病态的，也称系数矩阵 $\boldsymbol{A}$ 是病态的。

向量和矩阵的范数 **度量向量、矩阵大小的数量指标**

向量范数 设 $x=(x_1,\cdots x_n)^T$，范数记作 $\|x\|$

最常用的向量范数是 2-范数 $\|x\|_2=(x_1^2+\cdots+x_n^2)^{1/2}$

$1-$范数 $\|x\|_1=|x_1|+\cdots+|x_n|$ $\infty-$范数 $\|x\|_\infty=\max(|x_1|,\cdots|x_n|)$

矩阵范数 设 $A=(a_{ij})_{n\times n}$，范数记作 $\|A\|$

$\|A\|_2=\sqrt{\lambda_{\max}(A^TA)}$ $2-$范数 $\lambda_{\max}$ 表示最大特征根

$\|A\|_1=\max\limits_j\sum\limits_{i=1}^n|a_{ij}|$ （$1-$范数） $\|A\|_\infty=\max\limits_i\sum\limits_{j=1}^n|a_{ij}|$ （$\infty-$范数）

矩阵的从属（导出）范数 $\|A\|_p=\max\limits_{\|x\|_p=1}\|Ax\|_p$

条件数与误差分析

$\boxed{Ax = b}$ $\quad \|Ax\| \le \|A\| \cdot \|x\|$

1）设b有扰动 δb，分析x的误差 δx

$$A(x+\delta x) = b + \delta b \Rightarrow A\,\delta x = \delta b$$

$$\Rightarrow \delta x = A^{-1}\delta b \Rightarrow \|\delta x\| \le \|A^{-1}\| \cdot \|\delta b\|$$

$$\|b\| \le \|A\| \cdot \|x\| \Rightarrow 1/\|x\| \le \|A\| / \|b\|$$

$$\Rightarrow \boxed{\frac{\|\delta x\|}{\|x\|} \le \|A^{-1}\| \cdot \|A\| \cdot \frac{\|\delta b\|}{\|b\|}} \boxed{= cond\,(A)\frac{\|\delta b\|}{\|b\|}}$$

定义A的条件数为$Cond(A) = \|A^{-1}\| \cdot \|A\|$

A的条件数越大,(由b的扰动引起的)x的误差可能越大

条件数与误差分析

$$Ax = b$$

2）设A有扰动 δA，分析x的误差 δx

$$(A + \delta A)(x + \delta x) = b$$

$$\frac{\|\delta x\|}{\|x\|} \le \|A^{-1}\| \cdot \|A\| \cdot \frac{\|\delta A\|}{\|A\|} = cond(A)\frac{\|\delta A\|}{\|A\|}$$

A的条件数越大,(由A的扰动引起的)x的误差可能越大

x的(相对)误差不超过b的(相对)误差的$Cond(A)$倍

条件数大的矩阵可能是病态矩阵

范数，条件数和特征值

- n=norm(x) 输入x为向量或矩阵，输出为x的2-范数
- c=cond(x) 输入x为矩阵, 输出为x的2-条件数
- r=rcond(x) 输入x为方阵, 输出为x条件数倒数
- e=eig(x) 输入x为矩阵，输出x的全部特征值

- A=hilb(5);
- [norm(A), norm(A,1), norm(A,inf)]
- [cond(A), cond(A,1), cond(A,inf)]
- eig(A)

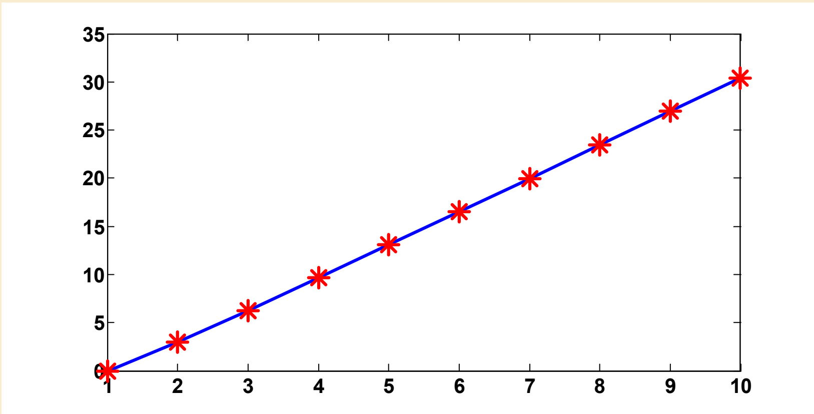
著名的病态矩阵Hilbert矩阵

$$H=\begin{bmatrix} 1 & 1/2 & \cdots\cdots & 1/n \\ 1/2 & 1/3 & \cdots\cdots & 1/(n+1) \\ \cdots\cdots & & & \\ 1/n & 1/(n+1) & \cdots 1/(2n-1) & \end{bmatrix}$$

$$H_{ij}=\frac{1}{i+j-1}. \qquad H_{ij}=\int_0^1 x^{i+j-2}\,dx,$$

% Hilbert矩阵条件数随阶数的增长

- for k=1:10
- c(k)=cond(hilb(k));
- end
- close; plot(1:10, log(c))



高斯消去法求解病态矩阵

- n=20; A=hilb(n); b=A*ones(n,1);
- norm(A\b-ones(n,1))
- Warning: Matrix is close to singular or badly scaled.
- Results may be inaccurate. RCOND = 1.155429e-019.

- ans = 201.1761

- norm(pcg(A,b)-ones(n,1)), ans =0.0186
- n=100; A=hilb(n); b=A*ones(n,1);
- norm(pcg(A,b)-ones(n,1)), ans =0.0515

迭代法

- **病态矩阵**
- **稀疏矩阵(直接法在消去过程中会破坏稀疏性)**

线性方程组 $\boldsymbol{Ax}=\boldsymbol{b}$，解是$\boldsymbol{x}^*$,即 $\boldsymbol{Ax}^*=\boldsymbol{b}$

如果矩阵 $\boldsymbol{B}$ 和向量 $\boldsymbol{f}$ 满足 $\boldsymbol{x}^*=\boldsymbol{Bx}^*+\boldsymbol{f}$

即可构造迭代法 $\boldsymbol{x}^{(k+1)}=\boldsymbol{Bx}^{(k)}+\boldsymbol{f}$

$\boldsymbol{x}^{(0)}$为选定的初值

迭代法 --- 一个例子

$$\begin{cases} 10x_1 + 3x_2 + x_3 = 14 \\ 2x_1 - 10x_2 + 3x_3 = -5 \\ x_1 + 3x_2 + 10x_3 = 14 \end{cases} \Rightarrow \begin{cases} x_1 = -0.3x_2 - 0.1x_3 + 1.4 \\ x_2 = 0.2x_1 + 0.3x_3 + 0.5 \\ x_3 = -0.1x_1 - 0.3x_2 + 1.4 \end{cases}$$

$$\boxed{\begin{cases} x_1^{(k+1)} = -0.3x_2^{(k)} - 0.1x_3^{(k)} + 1.4 \\ x_2^{(k+1)} = 0.2x_1^{(k)} + 0.3x_3^{(k)} + 0.5 \\ x_3^{(k+1)} = -0.1x_1^{(k)} - 0.3x_2^{(k)} + 1.4 \end{cases}} \quad k = 0,1,2,\cdots$$

$$x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0 \qquad x_1^{(1)} = 1.4, x_2^{(1)} = 0.5, x_3^{(1)} = 1.4$$

$$\Rightarrow \quad x_1^{(4)} = 0.9906,\ x_2^{(4)} = 0.9645,\ x_3^{(4)} = 0.9906$$

精确解 $x_1 = x_2 = x_3 = 1$

迭代法 – 雅可比（Jacobi）迭代

将 A 分解为 $A = D - L - U$，其中 $D = diag(a_{11}, a_{22}, \cdots a_{nn})$，

$$L = -\begin{bmatrix} 0 & & & \\ a_{21} & 0 & & \\ \vdots & \ddots & \ddots & \\ a_{n1} & a_{n2} & \cdots a_{n,n-1} & 0 \end{bmatrix}, \quad U = -\begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ & 0 & \ddots & \vdots \\ & & \ddots & a_{n-1,1} \\ & & & 0 \end{bmatrix}$$

设对角阵 D 非奇异（即 $a_{ii} \neq 0, i = 1, \cdots n$）　$Ax = b$

$$\Rightarrow Dx - (L+U)x = b \Rightarrow \boxed{x = D^{-1}(L+U)x + D^{-1}b}$$

记 $B_1 = D^{-1}(L+U)$

$f_1 = D^{-1}b$

$\Downarrow$ 迭代格式

$$x^{(k+1)} = B_1 x^{(k)} + f_1 \quad (k = 0,1,2\cdots)$$

迭代法 – 高斯-塞德尔(Gauss-Sedeil)迭代

Jacobi迭代公式 $Dx^{(k+1)} = Lx^{(k)} + Ux^{(k)} + b$

$$x_1^{(k+1)} = -0.3x_2^{(k)} - 0.1x_3^{(k)} + 1.4$$
$$x_2^{(k+1)} = 0.2x_1^{(k)} + 0.3x_3^{(k)} + 0.5$$
$$x_3^{(k+1)} = -0.1x_1^{(k)} - 0.3x_2^{(k)} + 1.4$$

改进

$$\boldsymbol{x}_1^{(k+1)} = -0.3\boldsymbol{x}_2^{(k)} - 0.1\boldsymbol{x}_3^{(k)} + 1.4$$
$$\boldsymbol{x}_2^{(k+1)} = 0.2\boldsymbol{x}_1^{(k+1)} + 0.3\boldsymbol{x}_3^{(k)} + 0.5$$
$$\boldsymbol{x}_3^{(k+1)} = -0.1\boldsymbol{x}_1^{(k+1)} - 0.3\boldsymbol{x}_2^{(k+1)} + 1.4$$

Gauss-Sedeil迭代公式

$$\boldsymbol{D}\boldsymbol{x}^{(k+1)} = \boldsymbol{L}\boldsymbol{x}^{(k+1)} + \boldsymbol{U}\boldsymbol{x}^{(k)} + \boldsymbol{b} \Rightarrow (\boldsymbol{D}-\boldsymbol{L})\boldsymbol{x}^{(k+1)} = \boldsymbol{U}\boldsymbol{x}^{(k)} + \boldsymbol{b} \Rightarrow \boldsymbol{x}^{(k+1)} = (\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{U}\boldsymbol{x}^{(k)} + (\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{b}$$

假设$(D-L)$可逆，于是得到

$$B_2 = (D-L)^{-1}U, \quad f_2 = (D-L)^{-1}b$$

$$x^{(k+1)} = B_2 x^{(k)} + f_2 \quad (k = 0,1,2\cdots)$$

迭代法的收敛性

Jacobi迭代 $x^{(k+1)} = B_1 x^{(k)} + f_1$

$B_1 = D^{-1}(L+U)$

$f_1 = D^{-1}b$

Gauss-Seideil迭代 $x^{(k+1)} = B_2 x^{(k)} + f_2$

$B_2 = (D-L)^{-1}U$

$f_2 = (D-L)^{-1}b$

一般迭代形式 $x^{(k+1)} = Bx^{(k)} + f$

原方程组的解x^*满足： $x^* = Bx^* + f$

$\boldsymbol{x^{(k+1)} - x^* = B(x^{(k)} - x^*)}$

迭代k次得到 $\boldsymbol{x^{(k)} - x^* = B^k(x^{(0)} - x^*)}$

序列收敛 $x^{(k)} \to x^* (k \to \infty)$的充要条件

$B^k \to 0 (k \to \infty)$ $\boxed{\Leftrightarrow B\text{的所有特征根（取模）小于}1}$

$\Updownarrow$

$\boxed{\rho(B) < 1}$

B的谱半径 $\rho(B) = \max\limits_{1 \le i \le n} |\lambda_i|$

$\lambda_i (i = 1, \cdots n)$是$B$的特征根

迭代法思想的另一种描述

假设 $x^{(k)}$是 $Ax=b$ 的一个近似解，$Ax^*=b$ 则

$$x^* = x^{(k)} - x^{(k)} + A^{-1}b = x^{(k)} + A^{-1}\left(b - Ax^{(k)}\right)$$

如果 C 是 A^{-1} 的一个足够好的近似，构造迭代法

$$x^{(k+1)} = x^{(k)} + C\left(b - Ax^{(k)}\right)$$

通过逐步校正，即可得到更接近 x^* 的 $x^{(k+1)}$

迭代法思想的另一种描述

$$x^{(k+1)} = x^{(k)} + C\left(b - Ax^{(k)}\right)$$

$$A = D - L - U$$

取 C 为 D^{-1} 即得到 *Jacobi* 迭代

取 C 为 $(D-L)^{-1}$ 即得到 *Gauss – Seidel* 迭代

迭代法的收敛性

序列收敛 $x^{(k)} \to x^{*}(k \to \infty)$ 的充分条件

1) 若A是严格对角占优的，即 $\left|b_{ii}\right| > \sum_{j \neq i}\left|b_{ij}\right| \quad (i = 1, \cdots n)$,

则雅可比和高斯－赛德尔迭代均收敛；

2）若 A 对称正定，则高斯—塞德尔迭代收敛；

3) 若$\|B\| = q < 1$，则迭代公式 $x^{(k+1)} = Bx^{(k)} + f$ 收敛

且 $\left\|x^{(k+1)} - x^{*}\right\| \leq \dfrac{q}{1-q}\left\|x^{(k+1)} - x^{(k)}\right\|$, q越小收敛越快

谱半径性质：$\rho(B) \leq \| B \|$ 其中||B||是任何一种矩阵范数

迭代法 - 超松弛(SOR)迭代

Gauss-Seideil迭代公式

$$Dx^{(k+1)} = b + Lx^{(k+1)} + Ux^{(k)} \quad \Rightarrow \quad x^{(k+1)} = D^{-1}\left(b + Lx^{(k+1)} + Ux^{(k)}\right)$$

$$x^{(k+1)} = (1-\omega)x^{(k)} + \omega D^{-1}\left(b + Lx^{(k+1)} + Ux^{(k)}\right)$$

$\omega > 1$	$\omega < 1$	$\omega = 1$
超松弛迭代	低松弛迭代	**Gauss-Seideil迭代**

迭代法 -超松弛 SOR 迭代

$$x^{(k+1)} = B_\omega x^{(k)} + f_\omega,$$

$$B_\omega = (D-\omega L)^{-1}[\omega U + (1-\omega)D],$$

$$f_\omega = \omega(D-\omega L)^{-1} b$$

若A对称正定 收敛充要条件

$$\boxed{0 < \omega < 2}$$

SOR 迭代------解大型稀疏矩阵方程组

线性方程组数值解法的MATLAB实现

1. 提取（产生）对角阵

v=diag(x) 输入向量x，输出v是以x为对角元素的对角阵；输入矩阵x，输出v是x的对角元素构成的向量；

v=diag(diag(x)) 输入矩阵x，输出v是x的对角元素构成的对角阵，可用于迭代法中从A中提取D。

2. 提取（产生）上（下）三角阵

y=triu(x) 输入矩阵 x，输出 v 是 x 的上三角阵；

v=tril(x) 输入矩阵 x，输出 v 是 x 的下三角阵；

例. 解 $\begin{cases} 10x_1 + 3x_2 + x_3 = 14 \\ 2x_1 - 10x_2 + 3x_3 = -5 \\ x_1 + 3x_2 + 10x_3 = 14 \end{cases}$

```
A=[10 3 1;2 -10 3;1 3 10],
b=[14 -5 14]',

[x,k]=JacDD(A,b,100,1e-7);
xx=zeros(3,1);
for k=1:5  xx=[xx, JacDD(A,b,k,1e-7) ]; end

[x,k]=GSDD(A,b,100,1e-7);

[x,k]=SORDD(A,b,100,1e-7,1.2);
```

Hilbert矩阵的求解

- n=20; A=hilb(n); b=A*ones(n,1);
  [xJ,k]=JacDD(A,b,1000,10^(-3));
- ans=NaN

- [xD,k]=GSDD(A,b,1000,10^(-5));
- k=472

- [xS,k]=SORDD(A,b,1000,10^(-5),1.75);
- k=405

Hilbert矩阵的求解

- n=200; A=hilb(n); b=A*ones(n,1);
  [xD,k]=GSDD(A,b,1000,10^(-5));
- k=919
- [xS,k]=SORDD(A,b,1000,10^(-5),1.95);
- k=917
- [xcg,flag,relres,k] = pcg(A,b,10^(-6),1000);
- k=10

动态迭代法Krylov 子空间

$$span\{r^{(0)},r^{(1)},r^{(2)},\cdots,r^{(k)}\}=span\{r^{(0)},Ar^{(0)},A^2r^{(0)},\cdots,A^kr^{(0)}\}$$

$$x^{(k+1)}=\min_{x\in span\{r^{(0)},r^{(1)},\cdots,r^{(k)}\}}\|Ax-b\|_2 \qquad r^{(0)}=Ax^{(0)}-b$$

help pcg

pcg - 求解线性方程组 - 预条件共轭梯度法（A对称正定）

此 MATLAB 函数，尝试使用预条件共轭梯度法求解关于 x 的线性方程组 A*x = b。如果尝试成功，pcg 会显示一条消息来确认收敛。

如果 pcg无法在达到最大迭代次数后收敛或出于任何原因暂停，则会显示一条包含相对残差 norm(b-A*x)/norm(b) 以及该方法停止时的迭代次数的诊断消息。

```
x = pcg(A,b)
x = pcg(A,b,tol)
x = pcg(A,b,tol,maxit)
```

稀疏矩阵的处理 ~ MATLAB进行大规模计算的优点

a=sparse(r,c,v,m,n)　在第r行、第c列输入数值v，矩阵共m行n列，输出a为稀疏矩阵，只给出(r,c)及v

aa=full(a) 输入稀疏矩阵a，输出aa为满矩阵（包含零元素)

```
a=sparse(2,2:3,8,2,4),  aa=full(a),
```

输出

```
a =(2,2)     8          aa= 0   0   0   0
   (2,3)     8              0   8   8   0
```

例. 分别用稀疏矩阵和满矩阵求解$Ax=b$, 比较计算时间

设 $A=\begin{bmatrix} 4 & 1 & & & \mathbf{0} \\ 1 & 4 & 1 & & \\ & 1 & 4 & \ddots & \\ & & \ddots & \ddots & 1 \\ \mathbf{0} & & & 1 & 4 \end{bmatrix}_{n\times n}$

$$b=[1,2,\cdots n]^T$$

```
n=4000;b=[1:n]';
a1=sparse(1:n,1:n,4,n,n);
a2=sparse(2:n,1:n-1,1,n,n);
a=a1+a2+a2';
tic;x=a\b;t(1)=toc;
aa=full(a);
tic;xx=aa\b;t(2)=toc
```

t =[0.00006, 0.26]　　400倍

```
norm(x-xx)
```

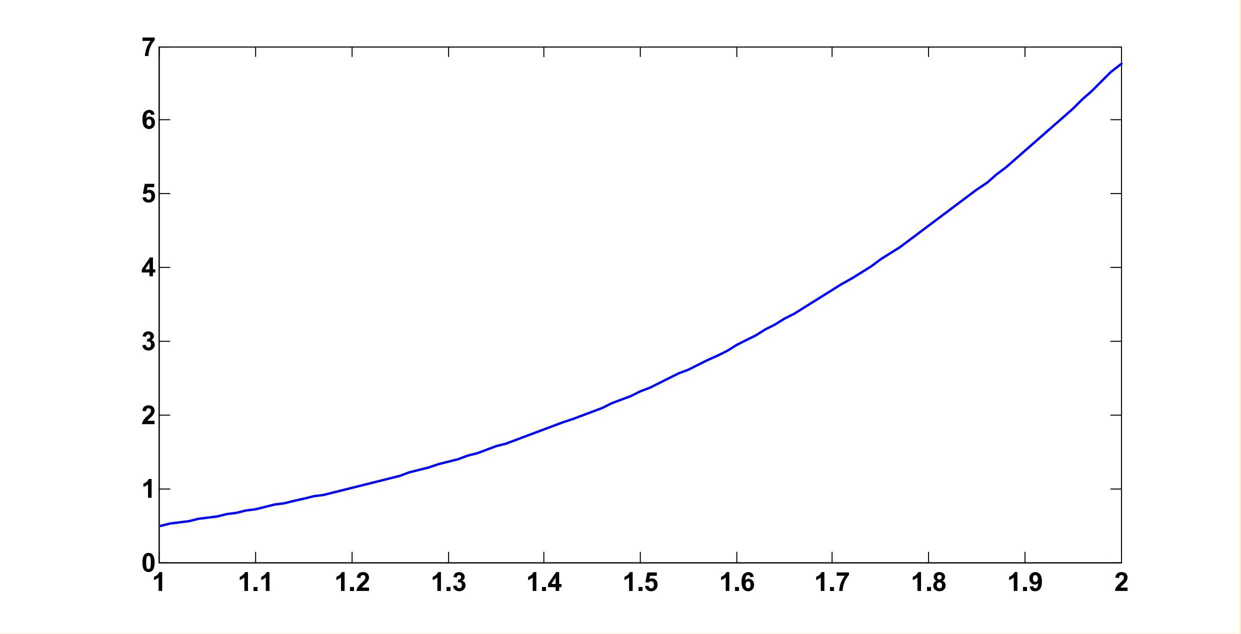
t(1), t(2)相差巨大，说明用稀疏矩阵计算的优点

（ norm(x-xx)用于简单地验证两种方法结果的一致）

常微分方程边值问题实例

$$\begin{cases} y'' + \dfrac{2}{x} y' - \dfrac{6}{x^2} y = 5 - 6x + 7x^2 \\ y(1) = \dfrac{1}{2}, y(2) = 4 + 4\ln 2 \end{cases}$$

解：此方程的解析解为 $y = x^2 - x^3 + \dfrac{1}{2} x^4 + x^2 \ln x$。



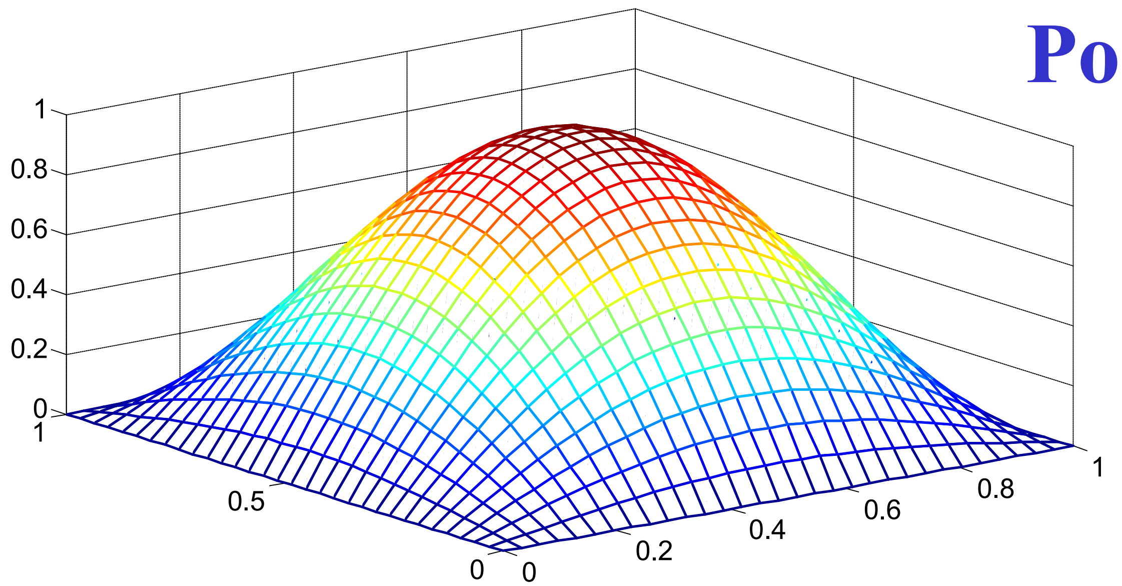
BVP_example.m

Poisson问题实例

考虑 Poisson 方程边值问题

$$\begin{cases} -\left(\dfrac{\partial^2 u}{\partial x^2}+\dfrac{\partial^2 u}{\partial y^2}\right)=2\pi^2\sin\pi x\sin\pi y,(x,y)\in\Omega, \\ u(x,y)=0, \qquad\qquad\qquad\qquad (x,y)\in\partial\Omega, \end{cases}$$

其中 $\Omega=\{(x,y)\,|\,0<x,y<1\}$， $\partial\Omega$ 是 Ω 的边界. 边值问题的解是 $u(x,y)=\sin\pi x\sin\pi y$.



Poisson_example.m

布置实验

目的

用MATLAB软件数值求解线性代数方程组，直接发与迭代法
理解病态方程的概念
对迭代法的收敛性误差做初步的分析与判断

作业　　见网络学堂