计算方法

Numerical Methods

数 学 系　　陈美蓉

Email: cmrzl@126.com

作业情况(Homework)

- 每周周二上课前（教4B205)

（由课代表代收，按班级序号排好）

2303—2317，2347-2363　(2，4，6周);

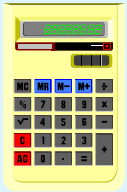
2319—2345 (3，5，7周)

上机时间

另行通知

参考书目 (Reference)

- 数值分析 李庆杨等编 （清华大学出版社）
- 计算方法典型例题分析 孙志忠 编 （科学出版社）
- **Numerical Analysis (Seventh Edition)**
  数值分析 （第七版 影印版）
  Richard L. Burden & J. Douglas Faires （高等教育出版社）



课程评分方法 (Grading Policies)

Lecture Grade (100) = Laboratory Projects (10-20)

+ Homework(10-20)

+ Final Exam (60-80)

内　　容

第一章　绪论

§1 计算方法的研究对象和特点

§2 误差及其基本概念

§3 数值计算的原则

§1 计算方法的研究对象和特点

计算方法的研究内容：构造算法（数学问题数值解的计算方法）

- 基本的数学问题？

1. 大型线性代数方程组$Ax=b$求解；
2. 矩阵A的特征值和特征向量计算；
3. 非线性方程 $f(x)=0$ 的求解(求根)；
4. 积分 $\int_a^b f(x)\,dx$ 计算；
5. 常微分方程初值问题求解；
6. 其它。

- 为什么要求数值解？

例如　常微分方程的初值问题 $\begin{cases} y' = 1 - 2xy \\ y(0) = 0 \end{cases}$

其解析解（精确解）为 $y(x) = e^{-x^2} \int_0^x e^{t^2} dt$

而实际只需知道 $y(1), y(1.5)$ 等近似值。这些近似值就是数值解。

- 如何构造方法（主要思想）

1. 迭代法
2. 以直线代替曲线（非线性问题线性化）
3. 化整为零（离散化）
4. 外推法（加速）

- 构造什么样的方法

实用的好的算法有三个标准：

快 ——— 计算步骤少，收敛速度快
准 ——— 数值稳定性好，计算结果可靠性高
省 ——— 节省计算机内存(大型稀疏矩阵问题)

1. 快

例1　多项式求值的Horner算法(秦九韶算法P7)

$$P_n(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$$

给定x的值，计算 $P_n(x)$ 的值。

算法一　按自然顺序计算　加法次数＝ n

$$\text{乘法次数}=n+(n-1)+\cdots+1=\frac{n(n+1)}{2}$$

算法二　嵌套算法（Hornor,秦九韶）

$$P_n(x)=(((a_nx+a_{n-1})x+a_{n-2})x+\cdots+a_1)x+a_0$$

乘法次数＝加法次数＝ n

例2 解线性方程组 $\begin{cases} a_{11}x_1+a_{12}x_2+\cdots+a_{1n}x_n=b_1 \\ a_{21}x_1+a_{22}x_2+\cdots+a_{2n}x_n=b_2 \\ \cdots\cdots \\ a_{n1}x_1+a_{n2}x_2+\cdots+a_{nn}x_n=b_n \end{cases}$

算法1： Cramer法则

$$x_i=\frac{D_i}{D},(i=1,2,\cdots,n),\ D=\sum(-1)^{\tau}a_{1j_1}a_{2j_2}\cdots a_{nj_n}$$

乘除法次数$A_n=\ n!(n-1)(n+1)+n$

如$n=20, A_{20}\approx 9.7\times 10^{20}$，假设计算机 1 秒钟进行

1 亿=10^8 次乘除法，共需时：

$$t_1 = \frac{9.7\times10^{20}}{10^8\times60\times60\times24\times365} \approx 30 \text{ (万年)}$$

算法2： Gauss消去法

乘除法次数：$A_n = \frac{1}{3}n^3 + n^2 - \frac{1}{3}n$

$$A_{20} = 3060$$

耗时：$t_2 = 3\times10^{-5}$ (秒)

例3 计算积分的梯形公式与Simpson公式；

非线性方程求根，Newton法比二分法快。

2. 准

例4 求根$x^2-56x+1=0$，假设计算机有尾数为5位，

$$\sqrt{783}=27.982$$

算法一 $$x_{1,2}=\frac{56\pm\sqrt{783\times 4}}{2}=28\pm\sqrt{783}$$

$$x_1=28+\sqrt{783}=55.982$$

$$x_2=28-\sqrt{783}=0.018$$

第二个解的精确值为0.0178628…

算法二 $$x_1=28+\sqrt{783}=55.982$$

$$x_2=\frac{1}{28+\sqrt{783}}=\frac{1}{55.982}=0.017863$$

例5 （P1例1，P5例3）计算积分

$$I(n)=e^{-1}\int_0^1 x^n e^x dx \quad (n=0,1,2,\cdots,9)$$

解法一　直接积分 $I(0)=1-e^{-1}$

由分部积分法可得 $I(n)=1-nI(n-1)$

取初值　　$I(0)=1-e^{-1}=0.6321\triangleq \tilde{I}_0$

则递推公式　　$\tilde{I}_n=1-n\tilde{I}_{n-1}$

计算得 $\tilde{I}_1=0.3679,\cdots,\tilde{I}_8=-0.7280,\tilde{I}_9=7.552$

	算法1结果	算法2结果	精确值
I(1)	0.3679000	0.3678795	0.3678794
I(2)	0.2642000	0.2642411	0.2642411
I(3)	0.2074000	0.2072768	0.2072766
I(4)	0.1704000	0.1708929	0.1708934
I(5)	0.1480000	0.1455357	0.1455329
I(6)	0.1120000	0.1267857	0.1268024
I(7)	0.2160000	0.1125000	0.1123836
I(8)	-0.7280000	0.1000000	0.1009323
I(9)	7.5520000	0.1000000	0.0916123
I(10)	-74.5200000	0.0000000	0.0838771

$$I_n = e^{-1}\int_0^1 x^n e^x dx$$

$$\frac{1}{(n+1)e} = e^{-1}\int_0^1 x^n e dx = \frac{1}{n+1}$$

$\tilde{I}_8 = -0.7280,$

$\tilde{I}_9 = 7.552$

$$\frac{1}{9e} \le I_8 \le \frac{1}{9}$$

$$\frac{1}{10e} \le I_9 \le \frac{1}{10}$$

$\Rightarrow$ I_8, I_9严重失真.

不稳定的算法，结果不可靠。

解法二　易知 $0 \le I(n) \le \dfrac{1}{n+1} \to 0$

取初值　$\tilde{I}_{10} = 0$

将递推公式 $\tilde{I}_n = 1 - n\tilde{I}_{n-1}$ 变形成如下格式

$$\tilde{I}_{n-1} = \frac{1-\tilde{I}_n}{n} \quad (n = 10, 9, \cdots, 2, 1)$$

计算结果相当好，见P5表1-2

问题：两个递推公式都对，为何会出现上面这两种截然不同的现象？

误差分析

例5中对于算法一中的递推公式进行稳定性分析

$$\tilde{I}_n = 1 - n\tilde{I}_{n-1} \quad (n = 1, 2, \cdots, 9) \qquad (*)$$

记 $I(n)$ 的误差为 $\varepsilon_n = I(n) - \tilde{I}_n$

则 $\varepsilon_n = I(n) - \tilde{I}_n = (1 - nI(n-1)) - (1 - n\tilde{I}_{n-1})$

$$= -n(I(n-1) - \tilde{I}_{n-1}) = -n\varepsilon_{n-1}$$

$$= -n[-(n-1)]\varepsilon_{n-2} = \cdots = (-1)^n n!\varepsilon_0$$

误差逐渐增大，(*)式不稳定

同样对于算法二中的递推公式进行稳定性分析

$$\tilde{I}_{n-1}=\frac{1-\tilde{I}_n}{n}\quad (n=10,9,\cdots,2,1)$$

记 $I(n)$ 的误差为 $\varepsilon_n=I(n)-\tilde{I}_n$

则 $\varepsilon_{n-1}=I(n-1)-\tilde{I}_{n-1}=\frac{1}{n}(1-I(n))-\frac{1}{n}(1-\tilde{I}_n)$

在我们今后的讨论中，误差将不可回避，

算法的稳定性会是一个非常重要的话题。

稳定性的定义

- 算法一是数值不稳定的
- 算法二是数值稳定的

求解

数值稳定性指的是方法，与问题无关；

数值不稳定的算法是不能用的；

不能说方法正确，程序正确，结果就正确。

最后，为了使计算稳定性好要注意以下几个问题：**P6-7**

(1) 防止大数吃小数

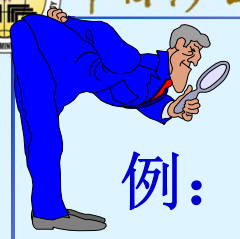
例　在五位十进制计算机上计算下式的值

$$A = 51234 + \sum_{i=1}^{1000} \delta_i \qquad 0.1 \le \delta_i \le 0.9$$

注：求和时从小到大相加，可使和的误差减小。

例：按从小到大、以及从大到小的顺序分别计算

$$1 + 2 + 3 + \cdots + 40 + 10^9$$



例：用单精度计算$x^2-(10^9+1)x+10^9=0$ 的根。

精确解 $x_1=10^9$，$x_2=1$

✍ 算法1：利用求根公式 $x=\dfrac{-b\pm\sqrt{b^2-4ac}}{2a}$

在计算机内，10^9存为0.1×10^{10}，1存为0.1×10^{1}。做加法时两加数的指数先向大指数对齐，再将浮点部分相加。即1 的指数部分须变为10^{10}，则：1 = 0.0000000001 × 10^{10}，取单精度时就成为：

$10^9+1=0.10000000\times10^{10}+0.0000000\times10^{10}=0.10000000\times10^{10}$

大数吃小数

$$\Rightarrow x_1=\frac{-b+\sqrt{b^2-4ac}}{2a}=10^9,\quad x_2=\frac{-b-\sqrt{b^2-4ac}}{2a}=0$$

✍算法2：先解出 $x_1 = \dfrac{-b - sign(b)\cdot\sqrt{b^2 - 4ac}}{2a} = 10^9$

再利用 $x_1 \cdot x_2 = \dfrac{c}{a} \Rightarrow x_2 = \dfrac{c}{a \cdot x_1} = \dfrac{10^9}{10^9} = 1$

(2) 防止相近的数相减

例 利用 $f'(x)\approx\dfrac{f(x+h)-f(x-h)}{2h}$，求 $f(x)=\sqrt{x}$在$x=2$

处的导数值。

解 $f'(x)\approx\dfrac{\sqrt{x+h}-\sqrt{x-h}}{2h}$ ⇒ $f'(2)\approx\dfrac{\sqrt{2+h}-\sqrt{2-h}}{2h}$

在4位机上取 h=0.0001

$$f'(2)\approx\frac{1.4142-1.4142}{0.0002}=0$$

精确值 $f'(2)=\dfrac{1}{2\sqrt{2}}=0.353553$

解决办法：$f'(2)\approx\dfrac{\sqrt{2+h}-\sqrt{2-h}}{2h}=\dfrac{2h}{(\sqrt{2+h}+\sqrt{2-h})2h}$

在4位机上仍取$h=0.0001$, 算得有$f'(2)\approx 0.35356.$

几种经验性避免方法：

$$\sqrt{x+\varepsilon}-\sqrt{x}=\frac{\varepsilon}{\sqrt{x+\varepsilon}+\sqrt{x}};\quad \ln(x+\varepsilon)-\ln x=\ln\left(1+\frac{\varepsilon}{x}\right);$$

当$|x|<<1$时：$1-\cos x=2\sin^2\dfrac{x}{2}$;

$$e^x-1=x\left(1+\frac{1}{2}x+\frac{1}{6}x^2+\ldots\right)$$

(3) 防止绝对值很小的数做分母

例 $t=1.01/x$, 取 $x=10^{-5}$, 则 $t=1.01/x=1.01\times10^{5}$

$\tilde{x}=1.01\times10^{-5}$是近似值，则 $\tilde{t}=1.01/\tilde{x}=10^{5}$

$\varepsilon=t-\tilde{t}=1000$

3、省

$$
\begin{cases}
b_1x_1 + c_1x_2 = d_1 \\
a_2x_1 + b_2x_2 + c_2x_3 = d_2 \\
a_3x_2 + b_3x_3 + c_3x_4 = d_3 \\
\quad\ddots \quad\ddots \quad\ddots \\
a_{n-1}x_{n-2} + b_{n-1}x_{n-1} + c_{n-1}x_n = d_{n-1} \\
a_nx_{n-1} + b_nx_n = d_n
\end{cases}
$$

Matlab 简介

Matlab是由美国的Clever Moler博士于1980年开发的。设计者的初衷是为解决“线性代数”课程的矩阵运算问题。

取名MATLAB即Matrix Laboratory 矩阵实验室的意思。

它是一个适用于不同规格计算机和各种操作系统的数学软件平台。 Matlab集科学计算、图象处理、声音处理于一身,它以超群的风格与性能风靡全世界，成功地应用于各工程学科的研究领域。

例 求解

$$\begin{cases} 3x_1 + x_2 - x_3 = 3.6 \\ x_1 + 2x_2 + 4x_3 = 2.1 \\ -x_1 + 4x_2 + 5x_3 = -1.4 \end{cases}$$

```
A=[3 1 -1;1 2 4;-1 4 5];
b=[3.6;2.1;-1.4];
x=A\b
```

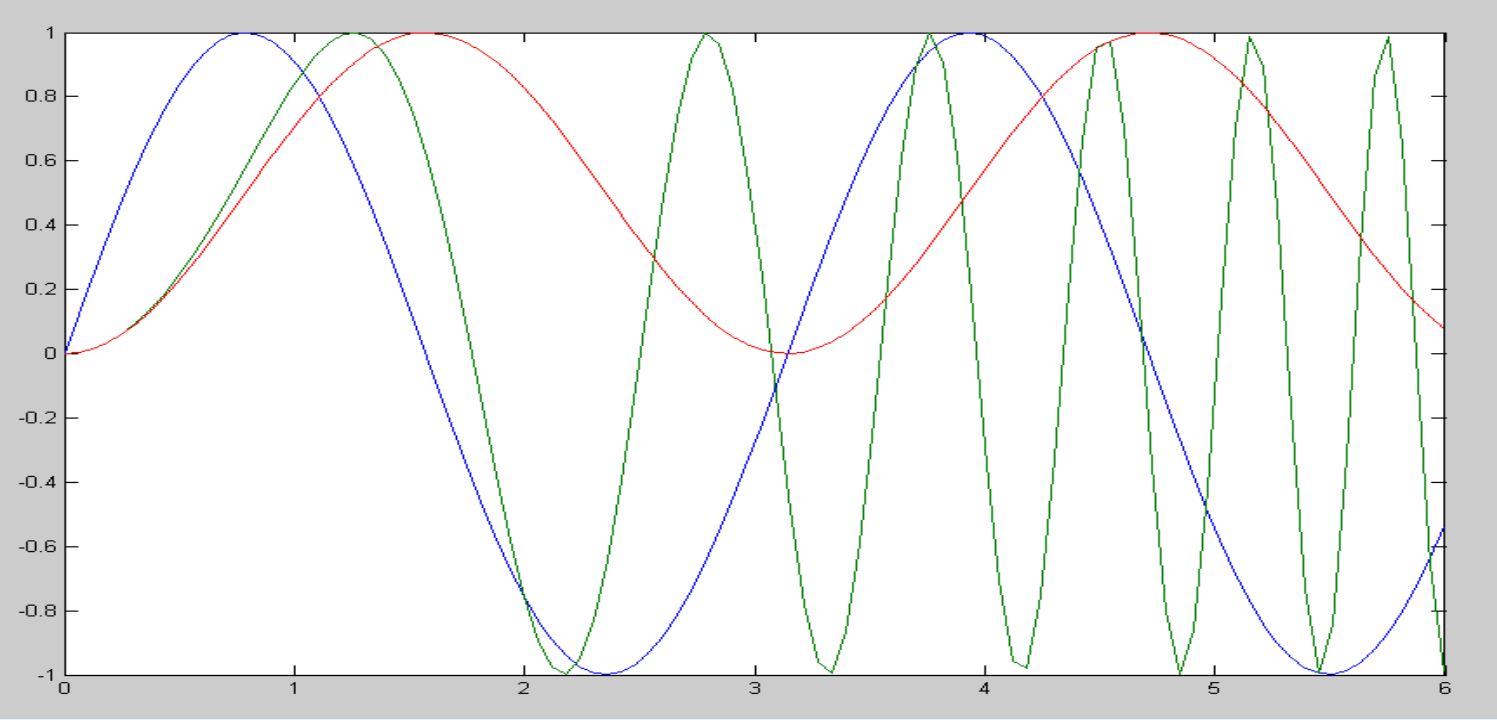
```
x =
    1.4818
   -0.4606
    0.3848
```

例 绘制在 $0 \le x \le 6$ 范围内的 $\sin(2x), \sin x^2, \sin^2 x$ 图象

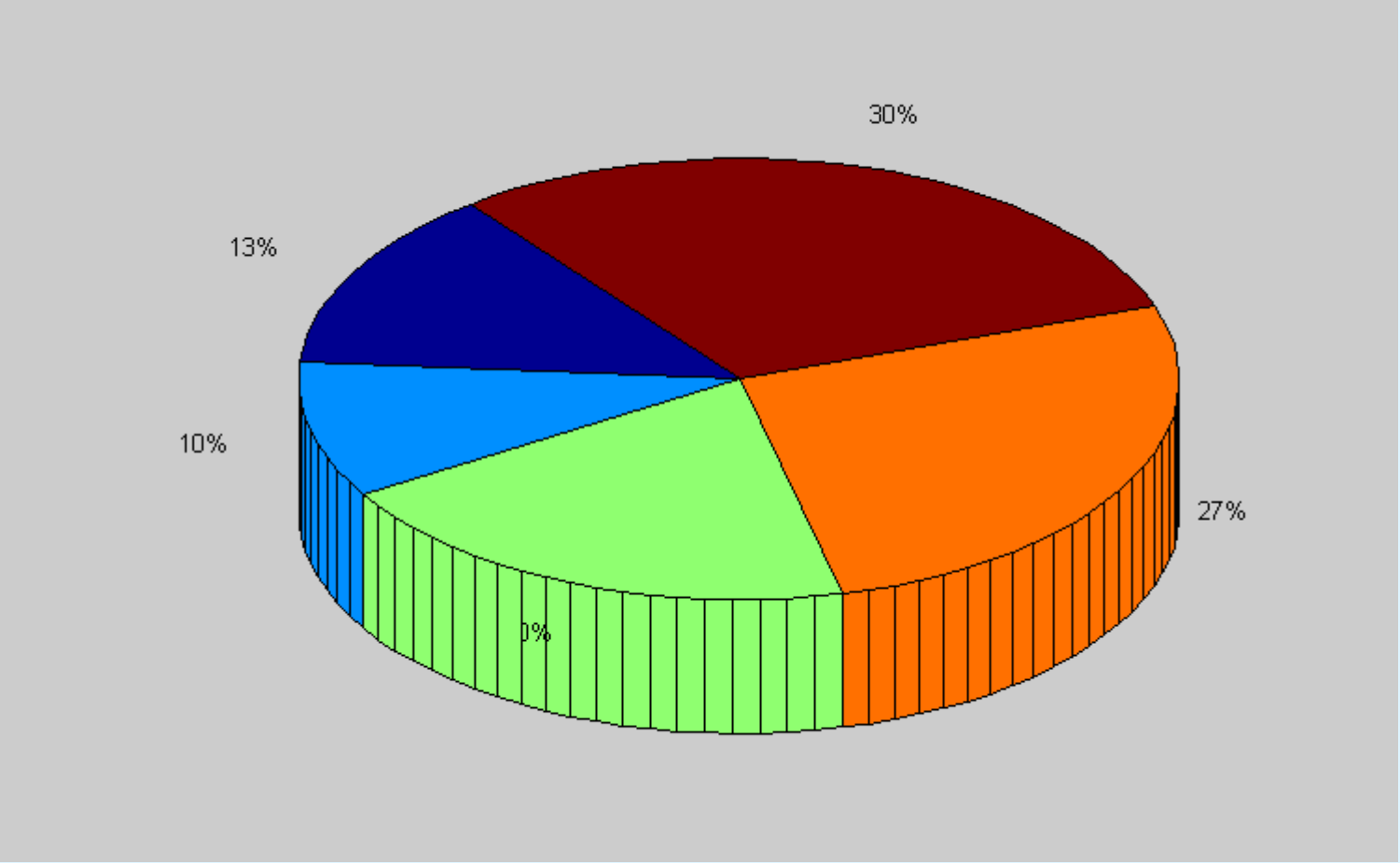
$$x = linspace(0,6)$$

$$y1 = \sin(2 * x),\ y2 = \sin(x.\wedge 2),\ y3 = (\sin(x)).\wedge 2$$

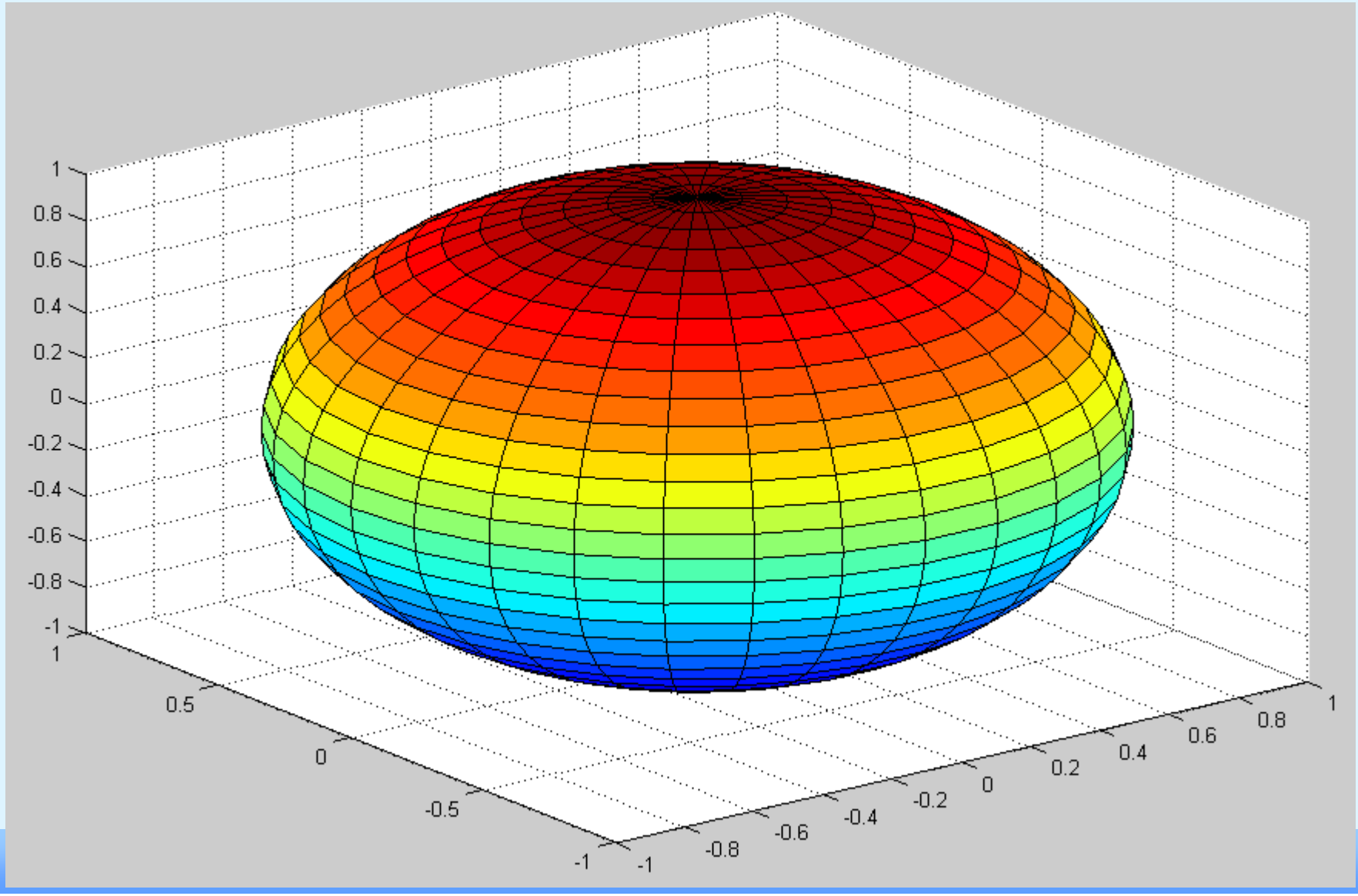
$$plot(x, y1, x, y2, x, y3)$$



pie3([4 3 6 8 9])



[x,y,z]=sphere(30); surf(x,y,z);



§2 误差的来源和基本概念

一、误差的来源

建立模型时产生
本课不讨论

1.模型误差
2.观测误差

求解模型时产生
本课主要讨论

3.方法误差或截断误差 —涉及→ 收敛性
4.舍入误差 —涉及→ 稳定性

如 $I(x)=e^x=1+x+\frac{x^2}{2!}+\cdots+\frac{x^n}{n!}+\cdots$

实际计算中：$I(x)\approx\tilde{I}(x)=1+x+\frac{x^2}{2!}+\cdots+\frac{x^n}{n!}$

截断误差 $R_n(x)=I(x)-\tilde{I}(x)=\frac{e^{\xi}}{(n+1)!}x^{n+1},(0<\xi<x)$

再如：用3.14159近似代替π,则产生的误差

$R=\pi$-3.14059=0.0000026…为舍入误差.

二、基本概念

➢ 误 差

假设x为准确值，x^*为近似值

绝对误差 $e = x - x^*$

绝对误差限 $\varepsilon : |e|$ 的一个上界，$|e| \le \varepsilon$

相对误差 $e_r = \dfrac{e}{x} \approx \dfrac{e}{x^*} \overset{\Delta}{=} e_r^*$

相对误差限 $\varepsilon_r : |e_r^*|$ 的一个上界，$|e_r^*| \le \varepsilon_r$

➢ 有效数字

用科学计数法，记 $x=\pm 0.a_1a_2\cdots\times 10^m$（其中 $a_1\neq 0$）。
则按四舍五入得

$$x^*=\begin{cases}\pm 0.a_1a_2\cdots a_n\times 10^m, & 0\le a_{n+1}\le 4\\ \pm\left(0.a_1a_2\cdots a_n+10^{-n}\right)\times 10^m, & 5\le a_{n+1}\le 9\end{cases}$$

易知 $\left|x-x^*\right|\le\frac{1}{2}\times 10^{m-n}.$

定义 设x^*为x的近似值，$x^*=\pm 0.a_1a_2\cdots\times 10^m\left(a_1\neq 0\right)$，若 $\left|x-x^*\right|\le\frac{1}{2}\times 10^{m-n}$且 n是满足此不等式的最大正整数，则称 x^*具有 n 位有效数字。

结论：1、用四舍五入得到的近似数从最后一位到前面第一位非零数字为止的所有数字都是有效数字。

2、若x^*具有n位有效数字，则 $|x-x^*|\le\frac{1}{2}\times10^{m-n}$. (绝对误差限)

例 $\pi=3.1415926535\,897932\cdots\cdots$； $\pi^*=3.1415$

问： π^* 有几位有效数字？请证明你的结论。

证明 $\because \pi^*=0.31415\times10^{1}$,

and $|\pi^*-\pi|<0.5\times10^{-3}=0.5\times10^{1-4}$

$\therefore \pi^*$ 有 4 位有效数字，精确到小数点后第 3位

注：0.2300有4位有效数字，而0.0023只有2位有效。12300如果写成0.123×10^5，则表示只有3位有效数字。

数字末尾的0不可随意省去！

举例分析:

例　作为**0.0509966……**的近似值，下列各数有几位有效数字？

0.051　　**0.0509**　　**0.05100**　　**0.05099**

2位　　2位　　4位　　3位

三、有效数字与误差限的关系

1、有效数字与绝对误差限的关系

设 x 的近似值 x^* 为：$x^* = \pm 0.a_1 a_2 \cdots a_n \times 10^m$

其中 a_i 为0到9中某个数字，$a_1 \neq 0$

若 x^* 具有 n 位有效数字，则

$$\left| x - x^* \right| \leq \frac{1}{2} \times 10^{m-n}$$

有效数字位数越多 $\Longleftrightarrow$ 绝对误差限越小

2、有效数字与相对误差的关系

☞ 有效数字 ⇒ 相对误差限

已知 x^* 有 n 位有效数字，则其相对误差限为

$$\varepsilon_r^* = \left|\frac{\varepsilon^*}{x^*}\right| = \frac{0.5\times 10^{m-n}}{0.a_1a_2\cdots a_n\times 10^m} = \frac{10^{-n}}{2\times 0.a_1\cdots} \le \frac{1}{2a_1}\times 10^{-n+1}$$

☞ 相对误差限 ⇒ 有效数字

已知 x^* 的相对误差限可写为 $\varepsilon_r^* = \dfrac{1}{2(a_1+1)}\times 10^{-n+1}$

则 $|x-x^*| \le \varepsilon_r^* \cdot |x^*| = \dfrac{10^{-n+1}}{2(a_1+1)}\times 0.a_1a_2\cdots\times 10^m$

$$< \frac{10^{-n+1}}{2(a_1+1)}\cdot(a_1+1)\times 10^{m-1} = 0.5\times 10^{m-n}$$

可见 x^* 至少有 n 位有效数字。

举例分析:

例　为使 π^*的相对误差小于0.001%,至少应取几位有效数字?

解　假设 π^* 取到 n 位有效数字，则其相对误差上限为

$$\varepsilon_r^* \le \frac{1}{2a_1} \times 10^{-n+1}$$

要保证其相对误差小于0.001%，只要保证其上限满足

$$\varepsilon_r^* \le \frac{1}{2a_1} \times 10^{-n+1} < 0.001\%$$

已知 $a_1 = 3$，则从以上不等式可解得 $n > 6 - \log 6$，即 $n \ge 6$，应取 $\pi^* = 3.14159$。

举例分析:

例 要使$\sqrt{70}$的近似值相对误差小于0.1%，应取几位有效数字？

解 $\sqrt{70}=0.8\cdots\times10\Rightarrow a_1=8$

欲使$\varepsilon_r\le\dfrac{1}{2a_1}\times10^{-(n-1)}=\dfrac{1}{16}\times10^{-(n-1)}\le0.1\%$

只要取$n=3$即可，即3位有效数字。

课后作业:

1、复习思考题:

1、3、4、5

2、习题一:

1、(1)(3)(6)、2、(3)

5、7

MATLAB
入门简介

1. Matlab 的特点与主要功能

❑ Matlab是一个交互式软件系统

给出一条命令，立即就可以得出该命令的结果

❑ 数值计算功能

✓ **Matlab**以矩阵作为基本单位，但无需预先指定维数（动态定维）

✓提供十分丰富的数值计算函数，方便计算，提高效率

✓ **Matlab**命令与数学中的符号、公式非常接近，可读性强，容易掌握

❑ 符号运算功能

和著名的 **Maple** 相结合，使得 **Matlab** 具有强大的符号计算功能

❑ 绘图功能

Matlab 提供了丰富的绘图命令，能实现一系列的可视化操作

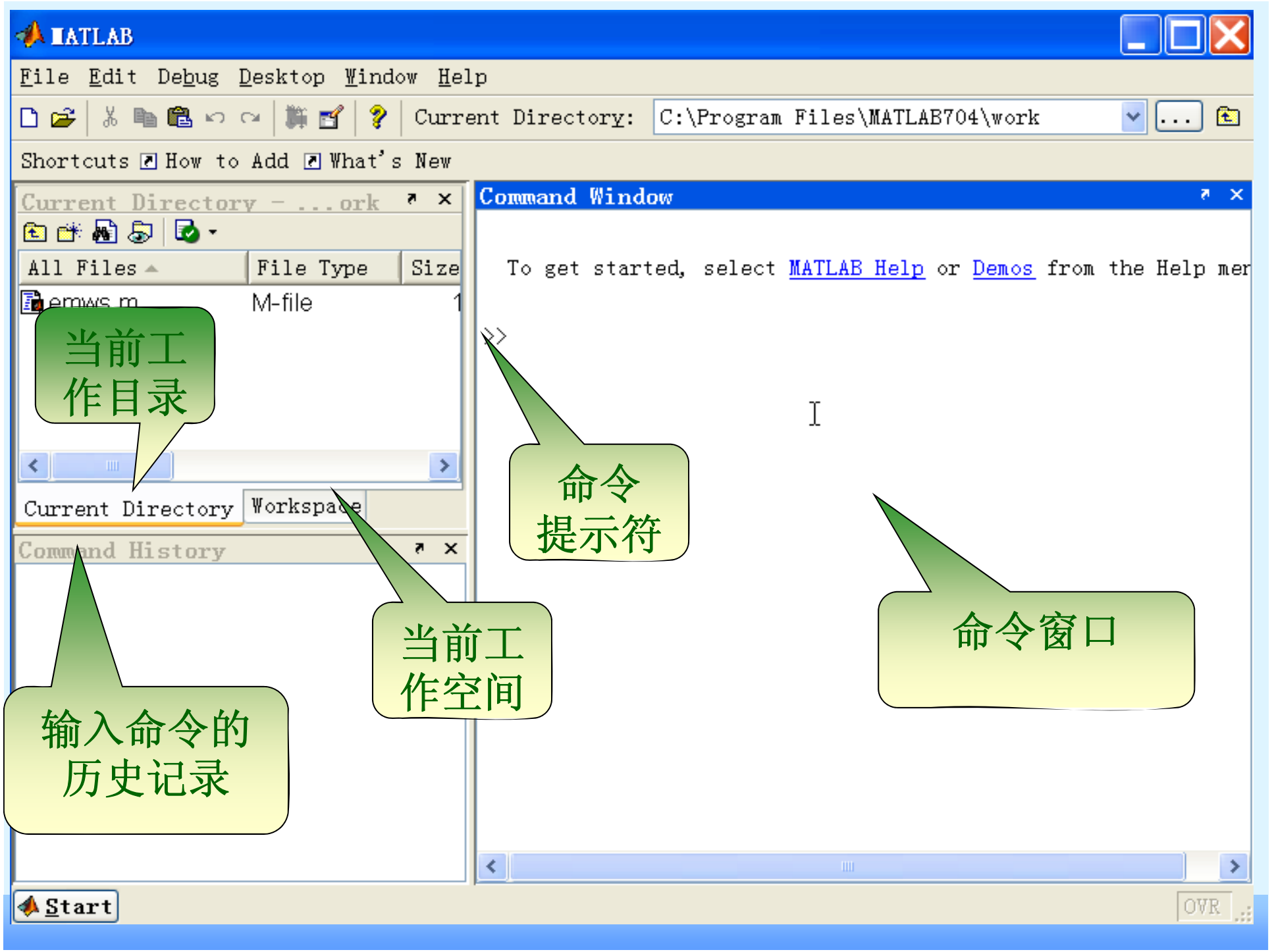
❑ 编程功能

Matlab 具有程序结构控制、函数调用、数据结构、输入输出、面向对象等程序语言特征，而且简单易学、编程效率高。

❑ 丰富的工具箱（**toolbox**）

Matlab 包含两部分内容：基本部分和根据专门领域中的特殊需要而设计的各种可选工具箱。

Optimization	Signal process	Control System
Symbolic Math	Image Process	System Identification
		……


MATLAB
File Edit Debug Desktop Window Help
Current Directory: C:\Program Files\MATLAB704\work
Shortcuts How to Add What's New
Current Directory - ...ork
All Files File Type Size
emws.m M-file
Current Directory Workspace
Command History
Command Window
To get started, select MATLAB Help or Demos from the Help mer
Start
OVR
当前工作目录
命令提示符
当前工作空间
命令窗口
输入命令的历史记录

2.MATLAB的联机帮助

- **MATLAB** 具有完善的帮助系统

(1) help / doc 命令

查看指定命令的使用方法

(2) lookfor 命令

根据用户提供的关键词，去搜索出一组与之有关的命令

3.Matlab 语言规则

- **Matlab** 区分大小写，它的命令和函数全是小写的
- 一行可以输入几个命令，用分号“;”或逗号“,”隔开
- 续行符：“…”（三个点），如果语句很长，可用续行符将一个语句写成多行。续行符的前最好留一个空格。
- 注释符：“%”，其后面的内容为注释，对 **Matlab** 的计算不产生任何影响

4.Matlab 变量

- Matlab中变量名是以字母开头，可以由字母、数字或下划线组成，最多 63 个字符（6.5 版本前为 19 个）
- 变量赋值： 变量 = 表达式

  赋值号左边必须是变量!
- 系统预定义变量：`eps, pi, Inf, NaN, i, j`
- who、whos、clear
- save、load

  `save` – 将所有变量存入文件 matlab.mat

  `save mydat / save mydat.mat` – 将所有变量存入 mydata.mat

  `save` 文件名 变量名列表

5. 数与算术表达式

- **Matlab** 中的数值表示一般采用十进制，可以带小数点和正负号。

如：`6，+6，-6.6，0.006，6.6e-6，100e60，-0.06e-012`

(1) Matlab 中的数默认是双精度实数

(2) 浮点运算的相对精度为 eps，Matlab 中 eps≈2.22e-016，即大约保持 16 位有效数字。

(3) Matlab 中数的表示范围为：10^{-308} ~ 10^{308}

(4) Matlab 中的虚部单位：*i*，*j*

`z=3+4i`（4 与 *i* 之间无空格），`z=3+4*i`

6. 数据的输入

❑ 数据输入

(1) 直接输入： `a=[1 2 ;3, 4]`

同一行中各元素之间用“空格”或“,”(英文状态下)分开;
行与行之间用“;”或“回车”分开

(2) 冒号“:”运算符：初值 : 步长 : 终值

`a=[1:5]， b=[0:pi/4:pi]`

(3) 由向量或小矩阵生成：`X=[a ;b]`

(4) 由 数据文件生成

(5) 交互式输入： `input`

```
n=input('Please input n: ')
```

7. 一些生成特殊矩阵的函数

zeros(m,n)	生成一个m行n列的零矩阵，m=n 时可简写为 zeros(n)
ones(m,n)	生成一个m行n列的元素全为1的矩阵, m=n 时可写为 ones(n)
eye(m,n)	生成一个主对角线全为1的m行n列矩阵, m=n 时可简写为 eye(n)，即为 n 维单位矩阵
diag(X)	若X是矩阵，则diag(X)为X的主对角线向量 若X是向量，diag(X)产生以X为主对角线的对角矩阵
tril(A)	提取一个矩阵的下三角部分
triu(A)	提取一个矩阵的上三角部分
rand(m,n)	产生 0~1 之间均匀分布的随机矩阵 m=n 时简写为 rand(n)
randn(m,n)	产生均值为0，方差为1的标准正态分布随机矩阵 m=n 时简写为 rand(n)
magic, vander, pascal, hilb	

8. 矩阵元素的操作

❑ 矩阵元素的提取

(1) 单个元素： $A(2,3)$

(2) 整行或整列： $A(2,:)$， $A(:,3)$

(3) 子矩阵： $A(2:5,4:8)$， $A([1,3],[2,4])$， $A([3,2],[2,4])$

(4) 删除矩阵的行列： $A=[]$， $A(3,:)=[]$， $A(:,[2,4])=[]$

9. 数据输出格式

- **Matlab** 以双精度执行所有的运算，结果可以在屏幕上输出，同时赋给指定变量，若无指定变量，则系统会自动将结果赋给变量 “**ans**”

- **Matlab** 中数据的输出格式可以通过 **format** 命令指定

```
>> format long
```

```
>> format compact
```

format 命令只改变变量的输出格式，但不会影响变量的值

特殊矩阵生成函数

格式	解释	例
format	短格式（缺省显示格式），同**short**	**3.1416**
format short	短格式（缺省显示格式），只显示5位	**3.1416**
format long	长格式，双精度数15位，单精度数7位	**3.14159265358979**
format short e	短格式 **e** 方式（科学计数格式）	**3.1416e+000**
format long e	长格式 **e** 方式	**3.141592653589793e+000**
format compact	压缩格式/紧凑格式	
format loose	自由格式/宽松格式	
format + / format bank / format rat / format hex (详情查看联机帮助)		

10. MATLAB 矩阵运算

- 矩阵的转置：共轭“'”，非共轭“.'”
- 矩阵的翻转和旋转： `fliplr`、`flipud`、`rot90`
- 矩阵元素重组： `reshape`(*A*, *m*, *n*)
- 查看矩阵的大小：`size`(*A*)、`size`(*A*, 1)、`size`(*A*, 2)
- 矩阵算术： `+`，`-`，`*`，`/`，`\`，`^`
- 数组运算（点运算）： `.*`，`./`，`.\`，`.^`

参与 “+，−，.*，./ ，.\” 运算的对象必须具有相同的形状

11. MATLAB 函数取值

❑ Matlab 普通函数取值

设 x 是变量，f 是一个函数，则

(1) 当 $x = a$ 是标量时，$f(x) = f(a)$

(2) 当 x 是向量或矩阵时，f 作用在 x 的每个分量上，结果为一个与 x 具有相同形状的向量或矩阵

$$\exp(A) = \begin{pmatrix} \exp(a_{11}) & \cdots & \exp(a_{1n}) \\ \vdots & \ddots & \vdots \\ \exp(a_{m1}) & \cdots & \exp(a_{mn}) \end{pmatrix}$$

❑ Matlab 矩阵函数：expm、sqrtm、logm，funm

```
funm(A,@cos)
```

❑ 三角函数

sin(x)	正弦函数	asin(x)	反正弦函数
cos(x)	余弦函数	acos(x)	反余弦函数
tan(x)	正切函数	atan(x)	反正切函数
cot(x)	余切函数	acot(x)	反余切函数
sec(x)	正割函数	asec(x)	反正割函数
csc(x)	余割函数	acsc(x)	反余割函数
...	...		

❑ 基本数学函数

abs(x)	绝对值	sum(x)	求和
max(x)	最大值	min(x)	最小值
sqrt(x)	开平方	exp(x)	以e为底的指数
log(x)	自然对数	log10(x)	以10为底的对数
sign(x)	符号函数	mod(x,y)	两整数相除的余数
conj(x)	求复数的共轭		
real(x)	取复数的实部	imag(x)	取复数的虚部

❑ 取整函数

round(x)	四舍五入到最近的整数
fix(x)	朝零方向取整
ceil(x)	朝正无穷方向取整
floor(x)	朝负无穷方向取整

❑ 矩阵相关函数

norm(A)	向量或矩阵的范数	rank(A)	矩阵的秩
det(A)	矩阵的行列式	trace(A)	矩阵的迹
inv(A)	方阵的逆矩阵	eig(A)	特征值及特征向量
size(A)	矩阵的阶数	cond(A)	矩阵的条件数
lu(A)	矩阵的LU分解	qr(A)	矩阵的QR分解