# 计算物理 Matlab

老王

2022 年 1 月 14 日

[0]发现问题或者有啥建议–>QQ2228031371

# 目录

# 第一部分　例题

# 1　解方程

## 1.1　二分法

### 1.1.1　例题及代码

用二分法求函数 $f(x) = x^2 - 2$ 在区间 (1,2) 内的根

```
clc;clear all;
time=20;
l=1;r=2;
for i=1:time
    m=(l+r)/2;
    if m*m-2>0
        r=m;
    else
        l=m;
    end
end
disp(m);
```

### 1.1.2　部分代码解释

**if**　选择结构，见第23页

**for**　循环结构，见第25页

**i=1:time**　`time=20`，这句代码在上面代码中与`for`一起的作用是做 20 次循环，每一次循环`i`都有对应的值，分别为 1，2，3 ••• 19，20

**disp(m)**　在命令行窗口输出`m`的值

**clc**　清空命令行窗口

**clear all**　工作区所有变量，如上面代码运行一次过后，会有`time l r m i`这几个变量

## 1.2 牛顿法

$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$

### 1.2.1 例题及代码

例题

已知 $\sqrt{2}$ 的一个近似值是 $x_0 = 1.414$，用 Newton 法 $x_1 = \frac{1}{2}(x_0 + \frac{2}{x_0})$ 计算 $x_1$，保留 9 位有效数字，并与准确值 $x = 1.414213562373\cdots$ 比较.

```
clc;clear all;
x0=1.414;
x=1.414213562373;
x1=0.5*x0+1/x0;
x1=round(x1*10^8)/10^8;
error=x1-x;
disp(x1);disp(error);
%Newton法得到的结果保留9位有效数字为1.41421358,
%与准确值相差约1.76269999e-8, 相差很小
```

### 1.2.2 部分代码解释

**10^8** 10 的 8 次方，也可以写成1e8

**round()** 对括号内的数的小数部分四舍五入

% 注释，类似 C 语言中的//

## 1.3 迭代法

$x_{n+1} = f(x_n)$

### 1.3.1 例题及代码

例题

找 $x = \frac{1}{2} + sin(x)$ 在 1.5 附近的根.

```
% 237页第8题
clear all;
x=1.5;
N=100;
for i=1:N
    x=0.5+sin(x);
end
anser=x;
disp(anser);
%方程在1.5附近的根约为1.497300389
```

### 1.3.2 部分代码解释

**sin()** 三角函数，见第22页

## 1.4 收敛的快慢 P211

设 $x_n \to \alpha$，$n \to \infty$，若存在常数 $p \geqslant 1$，$c > 0$ 使

$$\frac{|x_{n+1} - \alpha|}{|x_n - \alpha|^p} \to c, n \to \infty$$

则称该序列是 p 阶收敛的。

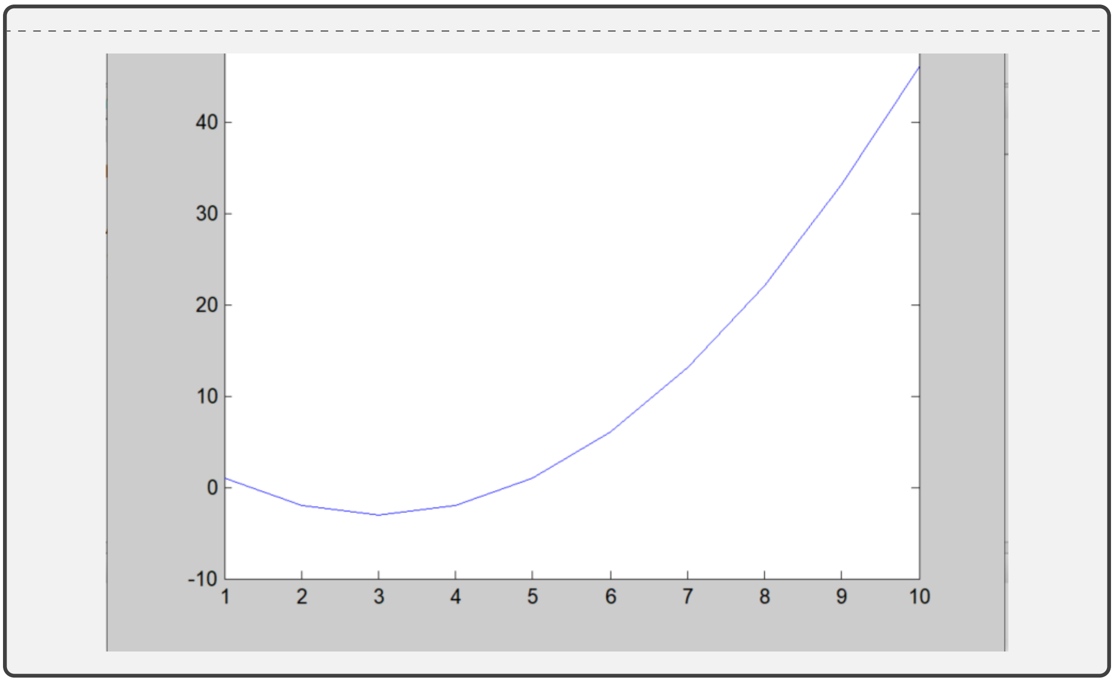
## 1.5 验证答案的方法

### 1.5.1 画图

画出 $y = f(x)$ 的图像，观察该函数图像与 $x$ 轴的交点，可以观察得到方程解的个数以及大概的值。

例：$(x-3)^2 - 3 = 0$

```
clc;clear all;
x=1:10;
y=(x-3).^2-3;
plot(x,y);
```




### 1.5.2 代入

带入方程，计算与 0 的差值（即误差）。

### 1.5.3 比较

比较不同方法得到的答案是否差别不大。

# 2 插值

## 2.1 拉格朗日（Lagrange）插值

### 2.1.1 一次插值 P15

$$\begin{cases} \varphi(x) = y_0 l_0(x) + y_1 l_1(x) \\ l_0(x) = \frac{x-x_1}{x_0-x_1} \\ l_1(x) = \frac{x-x_0}{x_1-x_0} \end{cases}$$

### 2.1.2 例题及代码

例题

设 $y=\sqrt{x}$，用 $x=100,121$ 两点构造线性插值函数，估计 $\sqrt{115}$ 的近似值，并给出误差分析。

```
clc;clear all;
x=100:0.1:121;
x0=100;y0=10;
x1=121;y1=11;
for i=1:length(x)
    l0=(x(i)-x1)/(x0-x1);
    l1=(x(i)-x0)/(x1-x0);
    y(i)=y0*l0+y1*l1;
end
plot(x,y);
disp(y(151));
error=sqrt(115)-y(151)
```

### 2.1.3 部分代码解释

**plot(x,y);** x 和 y 都是矩阵，且维度相同，此时 plot 的作用是以 x 的每个值为横坐标，y 的每个值为对应的中坐标，画出所有的点并将其连线；当 x 和 y 都只是一个数时，则只画出那个点

**sqrt(115)** $\sqrt{115}$

### 2.1.4 二次插值 P18

$$\begin{cases}\varphi(x)=y_0l_0(x)+y_1l_1(x)+y_2l_2(x)\\ l_0(x)=\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}\\ l_1(x)=\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}\\ l_2(x)=\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}\end{cases}$$

### 2.1.5 例题及代码

例题

设 $y=\sqrt{x}$，在 $x=100,121,144$ 三处的值是容易求得的，试以这三点建立 $y=\sqrt{x}$ 的二次插值多项式，并用此多项式计算 $\sqrt{115}$ 的近似值且给出误差估计。

```
%(100,10)(121,11)(144,12)
clc;clear all;
x=100:0.1:144;
x0=100;y0=10;
x1=121;y1=11;
x2=144;y2=12;
y=zeros(1,length(x));
for i=1:length(x)
    l0=(x(i)-x1)*(x(i)-x2)/(x0-x1)/(x0-x2);
    l1=(x(i)-x0)*(x(i)-x2)/(x1-x0)/(x1-x2);
    l2=(x(i)-x0)*(x(i)-x1)/(x2-x0)/(x2-x1);
    y(i)=y0*l0+y1*l1+y2*l2;
end
plot(x,y,'g',x,sqrt(x),'r');
disp(y(151);
error=sqrt(115)-y(151)
```

### 2.1.6 部分代码解释

**zeros()** 见第23页

**length(x)** 等于一维矩阵 x 的元素个数

**'r','g'** 所画图线的颜色，r 为红色，g 为绿色，可以不用管，至少 19 级的考试不会涉及

## 2.2 四个点的牛顿（Newton）插值

$$
\begin{cases}
f_{12} = \frac{y_1-y_2}{x_1-x_2} \\
f_{23} = \frac{y_2-y_3}{x_2-x_3} \\
f_{34} = \frac{y_3-y_4}{x_3-x_4} \\
f_{123} = \frac{f_{12}-f_{23}}{x_1-x_3} \\
f_{234} = \frac{f_{23}-f_{34}}{x_2-x_4} \\
f_{1234} = \frac{f_{123}-f_{234}}{x_1-x_4} \\
\varphi(x) = y_1 + (x-x_1)f_{12} + (x-x_1)(x-x_2)f_{123} + (x-x_1)(x-x_2)(x-x_3)f_{1234}
\end{cases}
$$

## 2.3 验证答案的方法

画出插值得到的图线，观察是否经过插值所用到的几个点。

# 3 数值积分

## 3.1 矩形法

左矩形 $\int_a^b f(x)\mathrm{d}x \approx (b-a)f(a)$
中矩形 $\int_a^b f(x)\mathrm{d}x \approx (b-a)f(\frac{a+b}{2})$
右矩形 $\int_a^b f(x)\mathrm{d}x \approx (b-a)f(b)$

## 3.2 梯形法

书 P78 $\int_a^b f(x)\mathrm{d}x \approx \frac{b-a}{2}(f(a)+f(b))$

## 3.3 抛物线法

书 P79 $\int_a^b f(x)\mathrm{d}x \approx \int_a^b P_2(x)\mathrm{d}x = \frac{b-a}{6}(f(a)+4f(\frac{a+b}{2})+f(b))$

### 3.3.1 例题及代码

例题

分别用矩形法、梯形公式和抛物线公式计算积分 $\int_0^1 \frac{x}{4+x^2}\mathrm{d}x$(八等分)

```
%P107-2(1)
clear all;
```

```
step=1/8;

x0=0:step:1;
y0=x0./(4+x0.^2);
sl=sum(step.*y0(1:end-1));%左矩形
sr=sum(step.*y0(2:end));%右矩形

x1=0:step:1;
y1=x1./(4+x1.*x1);
s1=step*((sum(y1))-0.5*y1(1)-0.5*y1(end));
%梯形法

x2=0:step/2:1;
y2=x2./(4+x2.*x2);
s2=0;
for i=1:8
    s2=s2+1/6*step*(y2(2*i-1)+4*y2(2*i)+y2(2*i+1));
end
%抛物线法

sm=sum(step.*y2(2:2:end-1));%中矩形法

format long;
Real=quad('x./(4+x.^2)',0,1);
disp('标准值：');disp(Real);

errorl=abs(sl-Real);
errorm=abs(sm-Real);
errorr=abs(sr-Real);
error1=abs(s1-Real);
error2=abs(s2-Real);
disp('左矩形法结果为：');disp(sl);
disp('误差大小为');disp(errorl);
disp('中矩形法结果为：');disp(sm);
disp('误差大小为');disp(errorm);
disp('右矩形法结果为：');disp(sr);
```

```
disp('误差大小为');disp(errorr);
disp('梯形法结果为: ');disp(s1);
disp('误差大小为');disp(error1);
disp('抛物线法结果为: ');disp(s2);
disp('误差大小为');disp(error2);
```

#### 3.3.2 部分代码解释

**sum()** 对括号内的内容求和，例如 $x=1:100$，则 $sum(x)=5050$，即从 1 加到 100 结果为 5050

**./ .* .^** 见第23页

**abs()** 括号内的数的绝对值

### 3.4 验证答案的方法

使用 Matlab 中的 *quad* 函数。

例

计算积分 $\int_0^1 \frac{x}{4+x^2}\mathrm{d}x$

```
clc;clear all;
s=quad('x./(4+x.^2)',0,1);
disp(s);
```

#### 3.4.1 部分代码说明

**quad()** 第一个逗号前为被积分函数 $f(x)$，第一第二个逗号之间为积分下限，第二第三个逗号之间为积分上限

# 4 幂法求矩阵特征值和特征向量

## 4.1 例题及代码

例题

用幂法计算矩阵

$$\mathbf{A} = \begin{pmatrix} 4 & 2 & 2 \\ 2 & 5 & 1 \\ 2 & 1 & 6 \end{pmatrix}$$

的最大特征值和相应的特征向量.

```
%P209-1
clear all;
A=[4 2 2
   2 5 1
   2 1 6];
x=rand(3,1);
for s=1:10
    x=A*x;
    x=x/norm(x);
end

disp('由幂法得到A的最大特征值');disp(x'*A*x);
disp('其对应的特征向量');disp(x);
```

## 4.2 部分代码说明

**rand(3,1)** 生成三行一列的随机矩阵

**norm(x)** 求范数

## 4.3 验证答案的方法

使用 Matlab 中的 *eig* 函数。

例

求出矩阵

$$\mathbf{A}=\begin{pmatrix}4&2&2\\2&5&1\\2&1&6\end{pmatrix}$$

的特征值和特征向量.

```
clc;clear all;
A=[4 2 2
   2 5 1
   2 1 6];
[V,D]=eig(A)
```

D 为所有的特征值，V 为所有特征值对应的特征向量。

# 5 微分方程

根据书上的内容，主要讨论一阶常微分方程。

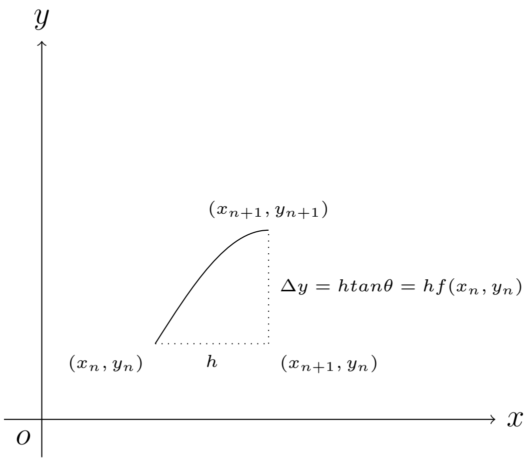
$$\begin{cases}\frac{\mathrm{d}y}{\mathrm{d}x}=y'=f(x,y) \quad x\in[a,b]\\ y(x_0)=y_0\end{cases}$$

## 5.1 Euler 方法 P239

$$\begin{cases}y(x_0)=y_0\\ y_{n+1}=y_n+hf(x_n,y_n)\end{cases}$$

[0]Euler 方法是一阶方法

### 5.1.1 理解




### 5.1.2 例题

在区间 $[0,2]$ 上，取步长 $h = 0.1$。$y' = y$，$y(0) = 1$

```
%y'=y&&y0=1      Euler法
clc;clear all;
h=0.1;
x=0:h:2;
y=zeros(1,length(x));
y(1)=1;
for i=2:length(x)
    y(i)=y(i-1)+h*y(i-1);
end
y2=exp(x);
plot(x,y2,x,y,'k');
```

### 5.1.3 部分代码解释

**exp(x)** 即 $e^x$

## 5.2 改进 Euler 方法 P244

$$y_{n+1} = y_n + \frac{h}{2}(f(x_n, y_n) + f(x_{n+1}, y_n + hf(x_n, y_n)))$$

[0]改进 Euler 方法是二阶方法

在区间 $[0,1]$ 上，取 $h=0.1$。$y'=y-\frac{2x}{y}, y(0)=1$

```
%y'=y-2*x/y&&y(0)=1 (y=sqrt(1+2*x))  改进Euler法
clear all;clc;
h=0.1;
x=0:h:1;
y=zeros(1,length(x));
y(1)=1;
for s=2:length(x)
    k1=y(s-1)-2*x(s-1)/y(s-1);
    %f(xn,yn)
    k2=(y(s-1)+h*k1)-2*x(s)/(y(s-1)+h*k1);
    %f(xn+1,yn+1)=f(xn+1,yn+h*f(xn,yn))
    y(s)=y(s-1)+h/2*(k1+k2);
end
plot(x,y,x,sqrt(1+2*x));
error=abs(y-sqrt(1+2*x));
```

## 5.3 经典 Runge-Kutta(R-k) 方法 P244

$$\begin{cases} K_1 = hf(x_n, y_n) \\ K_2 = hf(x_n + \frac{h}{2}, y_n + \frac{K_1}{2}) \\ K_3 = hf(x_n + \frac{h}{2}, y_n + \frac{K_2}{2}) \\ K_4 = hf(x_n + h, y_n + K_3) \\ y_{n+1} = y_n + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) \end{cases}$$

在区间 $[0,1]$ 上，取 $h=0.2$。$y'=y-\frac{2x}{y}, y(0)=1$

```
%y'=y-2*x/y&&y(0)=1 (y=sqrt(1+2*x))    R-K法
clc;clear all;
h=0.2;
x=0:h:1;
y=zeros(1,length(x));
```

[0] R-K 方法是四阶方法

```
y(1)=1;
for i=2:length(x)
    k1=h*(y(i-1)-2*x(i-1)/y(i-1));
    k2=h*((y(i-1)+k1/2)-2*(x(i-1)+h/2)/(y(i-1)+k1/2));
    k3=h*((y(i-1)+k2/2)-2*(x(i-1)+h/2)/(y(i-1)+k2/2));
    k4=h*((y(i-1)+k3)-2*(x(i-1)+h)/(y(i-1)+k3));
    y(i)=y(i-1)+1/6*(k1+2*k2+2*k3+k4);
end
y2=sqrt(1+2*x);
plot(x,y2,x,y);
```

## 5.4 两个未知数的 Euler、改进 Euler、R-k P254

$$\begin{cases} \frac{dy}{dx} = f(x,y,z), \quad y(x_0) = y_0 \\ \frac{dz}{dx} = g(x,y,z), \quad z(x_0) = z_0 \end{cases}$$

### 5.4.1 Euler

$$\begin{cases} y_{n+1} = y_n + hf(x_n, y_n, z_n) \\ z_{n+1} = z_n + hg(x_n, y_n, z_n) \end{cases}$$

### 5.4.2 改进 Euler

$$\begin{cases} y_{n+1}^{(k)} = y_n + hf(x_n, y_n, z_n) \\ z_{n+1}^{(k)} = z_n + hg(x_n, y_n, z_n) \\ y_{n+1}^{(k+1)} = y_n + \frac{h}{2}(f(x_n, y_n, z_n) + f(x_{n+1}, y_{n+1}^{(k)}, z_{n+1}^{(k)})) \\ z_{n+1}^{(k+1)} = z_n + \frac{h}{2}(g(x_n, y_n, z_n) + g(x_{n+1}, y_{n+1}^{(k)}, z_{n+1}^{(k)})) \end{cases}$$

[0]两个未知数似乎往年有考过一次，不过个人觉得今年（2021）考这个的概率不会很大

### 5.4.3 R-K

$$
\begin{cases}
k_1 = hf(x_n, y_n, z_n) \\
m_1 = hg(x_n, y_n, z_n) \\
k_2 = hf(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}, z_n + \frac{m_1}{2}) \\
m_2 = hg(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}, z_n + \frac{m_1}{2}) \\
k_3 = hf(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}, z_n + \frac{m_2}{2}) \\
m_3 = hg(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}, z_n + \frac{m_2}{2}) \\
k_4 = hf(x_n + h, y_n + k_3, z_n + m_3) \\
m_4 = hg(x_n + h, y_n + k_3, z_n + m_3) \\
y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\
z_{n+1} = z_n + \frac{1}{6}(m_1 + 2m_2 + 2m_3 + m_4)
\end{cases}
$$

# 第二部分　误差

书中主要讨论截断误差（方法误差）与舍入误差。

## 6　误差、误差限、有效数字

### 6.1　误差

用 $x^*$ 表示准确值 $x$ 的一个近似值，则此近似值 $x^*$ 与准确值 $x$ 的差称为误差，用 $e^*$ 表示，即

$$e^* = x^* - x$$

并且把 $-e^*$ 叫做近似值 $x^*$ 的“修正值”，误差为正时叫做“强近似”，误差为负时称为“弱近似”。

### 6.2　误差限

如果

$$|e^*| = |x^* - x| \leqslant \varepsilon^*$$

$\varepsilon^*$ 就叫做近似值 $x^*$ 的“误差限”。可以用

$$x = x^* \pm \varepsilon^*$$

来表示近似值 $x^*$ 的精确度或准确值所在的范围。

用四舍五入法得到的近似值的误差限为

$$\frac{1}{2} \times 10^{-n}$$

n 为 $x^*$ 的有效数字位数。

## 7　相对误差和相对误差限

$$e_r^* = \frac{e^*}{x} = \frac{x^* - x}{x}$$

为近似数 $x^*$ 的相对误差（也叫绝对误差），$e_r^*$ 比较小时可以用

$$e_r^* = \frac{e^*}{x^*} = \frac{x^* - x}{x^*}$$

表示。

$$\varepsilon_r^* = \frac{\varepsilon^*}{|x|}$$

或

$$\varepsilon_r^* = \frac{\varepsilon^*}{|x^*|}$$

称为相对误差限。

# 8 基本算数运算的误差传播

令 $dx \approx x^* - x, dy \approx y^* - y$，则

$$d(x \pm y) = dx \pm dy$$

$$d(xy) = ydx + xdy$$

$$d(x/y) = (-xdy + ydx)/y^2, y \neq 0$$

**例**

设 $a = 1.21 \times 3.65 + 9.81$，其中每个数据的绝对误差限为 0.005，求 $a$ 的绝对误差限。

$$da = d(1.21 \times 3.65) + d9.81$$
$$|da| \leqslant 1.21 \times 0.005 + 3.65 \times 0.005 + 0.005 \approx 0.0293 \leqslant 0.03$$

# 第三部分 语法

## 9 常用命令

表 1: 常用命令及其说明

| 命令 | 说明 |
|---|---|
| hold on | 作图控制，保留上一次已经画出来的点 |
| 2e3 | $2 * 10^3$ |
| pi | $3.14159265358979 \cdots$ |
| ans | 上一次的运算结果 |
| ; | 在赋值语句的结尾表示不在命令行窗口输出 |
| & \| ~ | 与或非 |
| sin(),cos(),tan(),asin(),acos(),atan() | 三角函数 |
| exp | e 为底的指数 |
| log | 以 e 为底的对数 |
| abs | 绝对值 |
| round() | 对括号内的数的小数部分四舍五入 |
| rand | 生成随机数 |
| min,max | 最小值, 最大值 |
| length | 矩阵的元素个数 |
| sum | 总和 |
| disp | 显示内容 |
| eig() | 求矩阵的所有特征值及对应的特征向量 |
| quad() | 求积分 |
| plot() | 画图 |
| axis([xmin xmax ymin ymax]) | 设定图像中 x 轴范围和 y 轴范围 |
| vpa(x,n) | 对 x 保留 n 位小数（四舍五入） |
| ′阿巴阿巴′ | 字符串 |

# 10 矩阵与数组

## 10.1 矩阵建立

最简单的方法：用方括号[ ]包围，同行元素用空格或逗号隔开，不同行用分号;或回车键分隔

## 10.2 指定元素

用`A(i,j)`来表示

## 10.3 特殊矩阵

| 特殊矩阵 | 说明 |
|---|---|
| ones() | 全为 1 的矩阵 |
| zeros() | 全为 0 的矩阵 |

## 10.4 要不要加点

要不要加 . 肯定是要看对计算过程有什么要求，目前涉及的可能常用的几种形式 [1]

| 类型 | 说明 |
|---|---|
| 数字除矩阵 | 数字./矩阵 |
| 矩阵除数字 | 矩阵/数字 |
| 数字乘矩阵 | 数字 * 矩阵或数字.* 矩阵都允许 |
| 矩阵乘数字 | 矩阵 * 数字或矩阵.* 数字都允许 |
| 矩阵的 n 次方 | 矩阵.^n |
| 矩阵除矩阵 | 矩阵/矩阵或矩阵./矩阵都允许 |

# 11 选择结构

## 11.1 if

单分支

[1]矩阵/矩阵的用法在这门课应该暂时用不到。

```
if (条件)
    语句
end
```

双分支

```
if (条件)
    语句
else
    语句
end
```

多分支

```
if (条件1)
    语句1
elseif 条件2
    语句2
...
else 条件n
    语句n
end
```

## 11.2 switch

```
switch 表达式
    case 数值1
        语句1; %不需要break
    case 数值2
        语句2;
    case 数值3
        语句3;
    otherwise
        语句4;
```

```
end
```

# 12 循环结构

## 12.1 for

```
for 控制变量=初值:步长:终值
    语句;
end
```

## 12.2 while

```
while 表达式
    循环体及控制语句;
end
```

# 13 改变变量显示格式

**format short** 十进制，保留 4 位小数。

**format long** 十进制，保留 15 位小数。

# 第四部分　拓展

## 14　R-K 方法与 kapler

```
clear all; close all; clc;

dt = 0.2;
tlist = 0: dt : 200;
xplist = zeros(4, length(tlist));
xplist(:,1) = [2; 0; 0; 0.5];
elist = 0* tlist;
h1 = figure;

for s = 1 : length(tlist) -1
    y = xplist(:, s);

    yy = y;
    r = sqrt(yy(1)^2 + yy(2)^2);
    K1 = dt * [yy(3); yy(4); -yy(1)/r^3; -yy(2)/r^3 ];

    yy = y + 0.5* K1;
    r = sqrt(yy(1)^2 + yy(2)^2);
    K2 = dt * [yy(3); yy(4); -yy(1)/r^3; -yy(2)/r^3 ];

    yy = y + 0.5* K2;
    r = sqrt(yy(1)^2 + yy(2)^2);
    K3 = dt * [yy(3); yy(4); -yy(1)/r^3; -yy(2)/r^3 ];

    yy = y +  K3;
    r = sqrt(yy(1)^2 + yy(2)^2);
    K4 = dt * [yy(3); yy(4); -yy(1)/r^3; -yy(2)/r^3 ];

    xplist(:,s+1) = y + (K1 + 2 *K2 + 2*K3 + K4)/6;

    xlim([-2 2])
    ylim([-2 2])
    plot(xplist(1,1 :s), xplist(2,1:s),...
```

```
        xplist(1,s), xplist(2, s),'o')
    pause(0.1)
    % plot(xplist(1,s), xplist(2, s),'o')
end
elist = 0.5*( xplist(3,:).^2 + xplist(4,:).^2) ...
    -1./(sqrt(xplist(1,:).^2 + xplist(2,:).^2));

% h1 = figure;
% plot(xplist(1, :), xplist(2,:))
%
% h2 = figure;
% plot(tlist, elist)
```

## 14.1 beta 版

```
clear all; close all; clc;

beta = 1.1;
dt = 0.1;
tlist = 0: dt : 200;
xplist = zeros(4, length(tlist));
xplist(:,1) = [2; 0; 0; 0.45 ];
elist = 0* tlist;
h1 = figure;

for s = 1 : length(tlist) -1
    y = xplist(:, s);

    yy = y;
    r = sqrt(yy(1)^2 + yy(2)^2);
    K1 = dt * [yy(3); yy(4); -beta*yy(1)/r^...
        (beta+2); -beta*yy(2)/r^(beta+2) ];

    yy = y + 0.5* K1;
    r = sqrt(yy(1)^2 + yy(2)^2);
    K2 =  dt * [yy(3); yy(4); -beta*yy(1)/r^...
        (beta+2); -beta*yy(2)/r^(beta+2) ];
```

```
    yy = y + 0.5* K2;
    r = sqrt(yy(1)^2 + yy(2)^2);
    K3 =  dt * [yy(3); yy(4); -beta*yy(1)/r^...
        (beta+2); -beta*yy(2)/r^(beta+2) ];

    yy = y +  K3;
    r = sqrt(yy(1)^2 + yy(2)^2);
    K4 =  dt * [yy(3); yy(4); -beta*yy(1)/r^...
        (beta+2); -beta*yy(2)/r^(beta+2) ];

    xplist(:,s+1) = y + (K1 + 2 *K2 + 2*K3 + K4)/6;

    xlim([-2 2])
    ylim([-2 2])
    axis square
    plot(xplist(1,1 :s), xplist(2,1:s),...
        xplist(1,s), xplist(2, s),'*')
    pause(0.01)
    % plot(xplist(1,s), xplist(2, s),'o')
end
elist = 0.5*( xplist(3,:).^2 + xplist(4,:).^2) ...
    -1./(sqrt(xplist(1,:).^2 + xplist(2,:).^2));

% h1 = figure;
% plot(xplist(1, :), xplist(2,:))
%
% h2 = figure;
% plot(tlist, elist)
```

# 15 R-K 方法与谐振

```
clear all; close all; clc;

A = [0, 1; -1, 0];
dt = 0.2;
```

```
tlist = 0 : dt : 2*pi * 5 ;
xplist = zeros(2, length(tlist));
xplist(:,1) = [1; 0 ];
for s = 1 : length(tlist) - 1
    y = xplist(:,s);
    K1 = dt * A * y ;
    K2 = dt * A *( y + 0.5*K1);
    K3 = dt * A * (y + 0.5 *K2);
    K4 = dt * A * (y + K3);
    xplist(:,s+1) = y + (K1 + 2*K2 + 2*K3 + K4)/6;
end
elist = xplist(1,:).^2 + xplist(2,:).^2;

h1 = figure;
plot(tlist, xplist(1,:), tlist, xplist(2,:)...
,':', 'linewidth',1.5)

h2 = figure;
plot(xplist(1,:), xplist(2,:))

h3 = figure;
plot(tlist, elist)
```

# 第五部分　说明

本文件中目录的标题设有超链接，可以通过点击标题直接到达对应位置，如果是用电脑查看，可以打开书签视图，一样可以通过点击到达对应位置，正文中出现页数的地方也有超链接，可以点击到达。

本文件的内容是根据 19 级的上课内容、考试内容以及作业来完成的。

理论力学书的后面有一些 Matlab 的基础知识。

本文件旨在通过课本例题帮助同学学习、理解计算物理课程中所涉及到的 Matlab 程序以及语法，而非提供习题参考答案，请同学自觉独立完成。

本文件会对书上的习题做一定修改，避免有的同学直接抄，同时本文件例题部分只是提供题目解法的其中一两种 Matlab 写法，并不是唯一的写法。

————老王

# 第六部分 鸣谢及更新（不分先后）

1. 王炜尧 • 我就是要有名字嘿嘿，老王本王高调一点
2. 张艺凡 • 第22页 • 完善 vpa 函数的用法
3. 江旭峰 • 第17页 • 提供一种改进 Euler 法代码的写法
4. 罗文斌 • 第23页 • 根据问题补充了什么时候加点
5. 刘世林 • 第23页 • 订正了原本 ones 和 zeros 的错误 • 2022.1.1
6. 老王 • 添加了目录；添加了插值 • 2022.1.1
7. 赵纯 • 第17页 • 订正改进 Euler 法代码公式错误 • 2022.1.1
8. 何海霞 • 第8页 • 提供插值笔记 • 2022.1.1
9. 林雅慧 • 第8页 • 提供插值例题 • 2022.1.1
10. 罗文斌 • 第16页 • 订正梯形公式与改进 Euler 公式有区别 • 2022.1.3
11. 老王 • 在一些不必要输出的地方加上分号，影响不大；第??页加了一些不会用到的函数 • 2022.1.4
12. 陈东 • 第5页 • 小改动 • 2022.1.5
13. 陈靖 • 第9页 • 订正了代码一个小错误 • 20221.1.6
14. 郑楠宇 • 第18页 • 加上了两个未知数的微分方程解法 • 2022.1.6
15. 罗文斌 • 第7页 • 加上了收敛速度 • 2022.1.6
16. 陈东 • 第11页 • 订正了中矩形法 • 2022.1.6
17. 老王 • 添加了说明部分；删除了 19 级考试文件命名规范；加入了验证计算结果的方法；完善了一些小地方 • 2022.1.11
18. 吴劭炜（Matlab 大佬、学霸）• 第26页 • 提供拓展部分的代码 • 2022.1.11