

## MATLAB 软件入门综合例题

例1 验证行列式的展开定理：矩阵的行列式可以按任一行展开，等于该行元素和对于代数余子式的乘积之和。

解题思路：要求由计算结果验证使用代数余子式展开之后的值与用`det()`直接求取的值是一致的。编写一个for循环中嵌套的条件语句，实现代数余子式的求取。可以任意取矩阵的大小进行验证，具有随机性，实验结果更加真实可靠。

MATLAB 程序:

```
A=rand(100,100); %其行列数可以任取
a=0;
for i=1:length(A)
    if i==1
        B=A(2:length(A),2:length(A));
        a(i)=((-1)^(i+1))*A(1,i)*det(B);
    else
        A1=A(2:length(A),1:i-1);
        A2=A(2:length(A),i+1:length(A));
        B=[A1 A2];
        a(i)=((-1)^(i+1))*A(1,i)*det(B);
    end
end
result1=sum(a)
result2=det(A)
```

运行该程序得到

```
result1=
-3.3363e+25
result2 =
-3.3363e+25
```

例2. 现在的人们能够知道圆周率的精确值，但在古代这是一个难题。中国古代数学家，比如祖冲之，得到了圆周率的高度近似结果。用分数近似的圆周率，其中  $22/7$  称为约率， $355/113$  称为密率。编程验证这两个数是两位数分数和三位数分数中最接近的数。同时找出四位数分数中最接近于圆周率的数。简单说明你的程序的优点，如果要使运算更加高效，如何改进。

MATLAB 程序（一）

```
% 求在两位数以内的分数中与圆周率最接近的分数
m1=1;
format long
for i=1:99
    for j=1:99
```

```

        if abs(i/j-pi)<m1
            m1=abs(i/j-pi);
            a1=i;b1=j;
        end
    end
end
m1,a1,b1
pi1=a1/b1
% 求在三位数以内的分数中与圆周率最接近的分数
m2=1;
for i=1:999
    for j=1:999
        if abs(i/j-pi)<m2
            m2=abs(i/j-pi);
            a2=i;b2=j;
        end
    end
end
m2, a2,b2
pi2=a2/b2
% 求与圆周率最接近的四位数分数
m3=1;
for i=1000:9999
    for j=1000:9999
        if abs(i/j-pi)<m3
            m3=abs(i/j-pi);
            a3=i;b3=j;
        end
    end
end
m3,a3,b3
pi3=a3/b3

```

运行该程序的结果：

```

m1 =
    0.001264489267350
a1 =
    22
b1 =
     7
pi1 =
    3.142857142857143
m2 =
    2.667641894049666e-07

```

```

a2 =
    355
b2 =
    113
pi2 =
    3.141592920353983
m3 =
    2.667641894049666e-07
a3 =
    3195
b3 =
    1017
pi3 =
    3.141592920353983

```

总结：使用i和j做两次历遍，就保证每个四位数都包含在内，使用format long来显示更多位小数。

也可用数组运算来实现

MATLAB 程序（二）

% 求在两位数以内的分数中与圆周率最接近的分数

```

format long
x=1:99;
y=1:99;
[X,Y]=meshgrid(x,y);
M1=abs(X./Y-pi);
M1_1=reshape(M1,1,99^2);
[m1,I]=min(M1_1);
m1
a1=X(I)
b1=Y(I)

```

% 求在三位数以内的分数中与圆周率最接近的分数

```

x=1:999;
y=1:999;
[X,Y]=meshgrid(x,y);
M2=abs(X./Y-pi);
M2_1=reshape(M2,1,999^2);
[m2,I]=min(M2_1);
m2
a2=X(I)
b2=Y(I)

```

计算结果与程序（一）相同。

例 3. 数列和级数的收敛性验证是高等数学中的两个非常重要的问题，使用图形的方法是发现结论的最直观方法，实际上只要将数列（或者部分和数列）的前面一些项在图形上画出来并看看是否有收敛于某值的趋势。针对下面的数列，在图形上画出这些点并猜测极限是多少。

(1)  $a_n = \frac{1}{1 \cdot 2 \cdot \dots \cdot n}$ ，考察级数  $\sum_{k=1}^n a_k$  的部分和数列。

(2) p-级数的收敛性，特别的计算 p=1, 2 和 3 的情况。

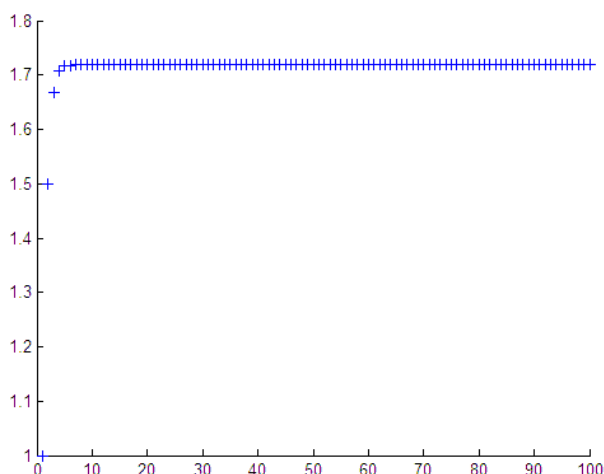
(3)  $a_n = 0.5(a_{n-1} + \frac{5}{a_{n-1}}), a_1 = 28$

(1) MATLAB程序:

```
s=0;
a=1;
hold on
for i=1:100
    a=a/i;
    s=s+1;
    plot(i,s,'+')
end
s
hold off
```

运行该程序的结果:

```
s =
    1.71828182845905
```



数列收敛，极限约等于1.718

(2) MATLAB程序

```
p=[1 2 3];
s=[0 0 0];
for i=1:100
```

```

s(1)=s(1)+1/i^p(1);
s(2)=s(2)+1/i^p(2);
s(3)=s(3)+1/i^p(3);
subplot(2,2,1),plot(i,s(1),'+'),title('p=1'),hold on
subplot(2,2,2),plot(i,s(2),'y+'),title('p=2'),hold on
subplot(2,2,3),plot(i,s(3),'g+'),title('p=3'),hold on
end

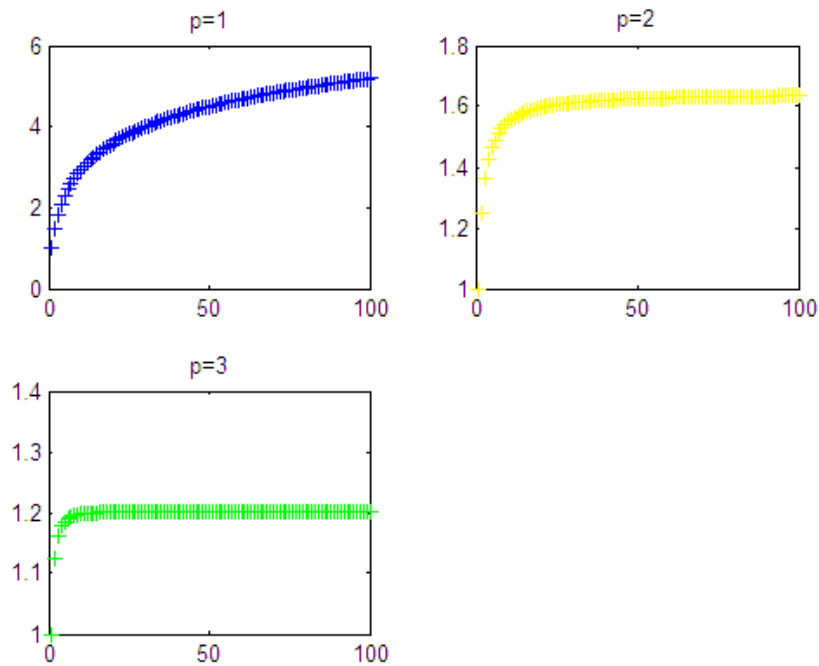
```

运行结果:

```

s =
    5.18737751763962    1.63498390018489    1.20200740065968

```



当 $p=1$ 时 $p$ 级数发散,  $p=2$ 时收敛, 大约收敛于1.640,  $p=3$ 时 $p$ 级数收敛, 大约收敛于1.202.

### (3) MATLAB程序

```

x=28;
hold on
for i=1:20
    plot(i,x,'g+')
    x=0.5*(x+5/x);
end
hold off
x

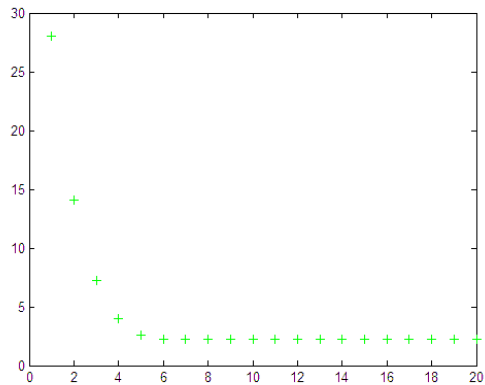
```

运行结果:

```

x =
    2.23606797749979

```



答：数列收敛，大约收敛于2.236

例 4. 计算一个弹性小球，在高度  $h$  自由下落，每次和地面接触时能量损失 20%，小于 1 毫米时认为不再弹起。计算物体运动的轨迹，并作出模拟的动画。

解：MATLAB 程序如下所示：

1、小球主程序：

```
clear;
h=10;
flag=0;
while h>0.001
    if rem(flag,2)==0
        down(h)
        flag=flag+1;
        h=0.8*h;
    else
        flag=flag+1;
        up(h)
    end
end
```

2、上升函数：

```
function f=up(h)
g=9.8;
x=0;
tmax=sqrt(2*h/g);
for t=tmax:0.1:0
    y=h-0.5*g*t^2;
    pause(0.15);
    plot(x,y,'o','markersize',12,'markerfacecolor','r')
    axis([-1,1,0,11]);
end
```

3、下降函数：

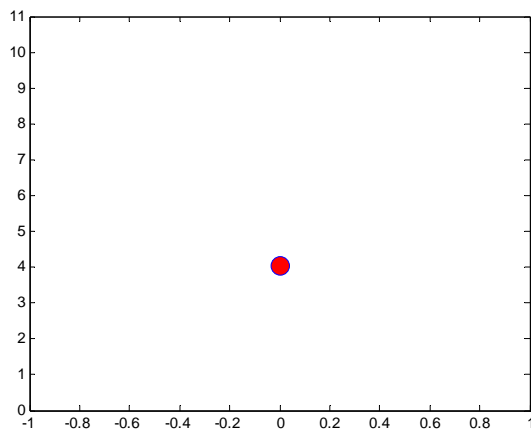
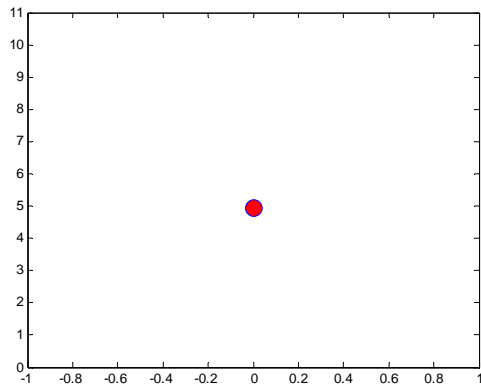
```
function f=down(h)
g=9.8;
x=0;
```

```

tmax=sqrt(2*h/g);
for t=0:0.1:tmax
    y=h-0.5*g*t^2;
    pause(0.15);
    plot(x,y,'o','markersize',12,'markerfacecolor','r')
    axis([-1,1,0,11]);
end

```

得到的结果为动画，一下截出几幅图作为说明：



例5. 函数在某点的切线可以用割线进行逼近，选择一个函数，画出某点附近的几条割线和切线，说明这个结论。

函数M文件：

```

function y=f(x)
y=x^2+x*sin(x)+2;

```

主程序（脚本M文件）：

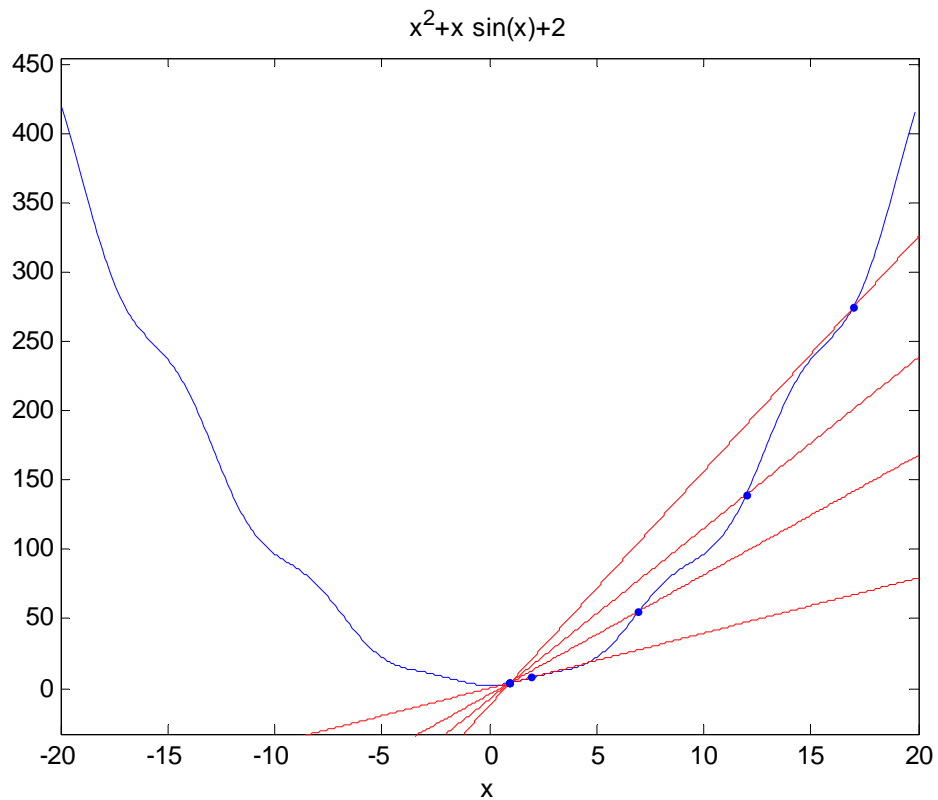
```

x0=1;delta=1;
ezplot('x^2+x*sin(x)+2',[-20,20])
hold on
x=-20:0.01:20;

```

```
for i=16:-5:1
    y0=f(x0);
    y1=f(x0+i*delta);
    k=(y1-y0)/(i*delta);
    plot(x,k*(x-x0)+y0,'r',[x0,x0+i*delta],[y0,y1],'.')
end
hold off
```

运行结果:



总结: 函数可以先用另一个 m 文件储存, 这样调用起来更加方便。