

实验 1 MATLAB 基本操作

【实验目的】

- 1、熟悉 MATLAB 操作环境，认识命令窗口、内存工作区窗口、历史命令窗口；
- 2、掌握 MATLAB 中数组与矩阵的创建；掌握矩阵运算与数组运算；掌握基本元素群运算
- 3、掌握 MATLAB 程序的编写、调试、调用、程序设计方法、MATLAB 程序控制结构中顺序结构、选择结构、循环结构的使用、
- 4、掌握 MATLAB 函数的编写规则，及函数的调用
- 5、掌握二维图形的绘制；掌握图形的标注；

【实验内容】

一 矩阵操作

1、已知矩阵 $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$, $B = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9]$, $C = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$, 要求完成

- 1)、分别在 MATLAB 命令窗内完成以上 3 个矩阵的赋值，总结逗号，分号在矩阵赋值的使用方法。
- 2)、分别用两种方法，写出引用 A 矩阵中' 4 ' 元素表达式。
- 3)、写出表达式，验证：赋值元素下标超维时，矩阵行列自动扩展，引用元素下标超维时(超维调用)，则会报错。
- 4)、写出表达式，给 B 矩阵的第 2 行赋值为[9 8 7 6 5 4 3 2 1],C 矩阵的第 2 列赋值为[4 3 2 1]。

答：

- 1)逗号：一行中每个元素之间的分隔，也就是矩阵的列的分隔；
分号：矩阵的行的分隔；

```
命令窗口
>> A = [1, 2, 3; 4, 5, 6; 7, 8, 9]

A =

     1     2     3
     4     5     6
     7     8     9

>> B = [1, 2, 3, 4, 5, 6, 7, 8, 9]

B =

     1     2     3     4     5     6     7     8     9

>> C = [1; 2; 3; 4]

C =

     1
     2
     3
     4

fx >>
```

图 1.1.1

2)索引矩阵 A 中的元素'4';

第一种方法根据位置索引，'4'在矩阵中的第 2 行，第 1 列， 所以用 A(2,1)进行索引。

第二种方法矩阵中从列从上到下开始计数，'4'在矩阵 A 中位于第 2 个，所以用 A(2)索引。

```
A =

     1     2     3
     4     5     6
     7     8     9

>> A(2,1)

ans =

     4

>> A(2)

ans =

     4

fx >> |
```

图 1.1.2

3)

矩阵 A 一开始是 3*3 的矩阵，对矩阵 A 的第 4 行进行超维赋值，结果赋值成功，矩阵进行自动行扩展，然后在对 A 的第 4 列进行赋值，矩阵自动进行列扩展。

```
>> A

A =

     1     2     3
     4     5     6
     7     8     9

>> A(4,:)=[10, 11, 12;]

A =

     1     2     3
     4     5     6
     7     8     9
    10    11    12
```

图 1.1.3

```
>> A

A =

     1     2     3
     4     5     6
     7     8     9
    10    11    12

>> A(:, 4)=[1; 1; 1; 1]

A =

     1     2     3     1
     4     5     6     1
     7     8     9     1
    10    11    12     1
```

图 1.1.4

引用元素下标超维，会出现报错，所以不能超维引用元素。

```
A =
```

1	2	3	1
4	5	6	1
7	8	9	1
10	11	12	1

```
>> A(5,5)
```

位置 1 处的索引超出数组边界(不能超出 4)。

图 1.1.5

4)

B 矩阵的第 2 行进行赋值[9 8 7 6 5 4 3 2 1]

```
>> B
```

```
B =
```

1	2	3	4	5	6	7	8	9

```
>> B(2,:)=[9 8 7 6 5 4 3 2 1]
```

```
B =
```

1	2	3	4	5	6	7	8	9
9	8	7	6	5	4	3	2	1

图 1.1.6

C 矩阵的第 2 列进行赋值[4 3 2 1]

```
C =  
  
     1  
     2  
     3  
     4  
  
>> C(:,2)=[4; 3; 2; 1;]  
  
C =  
  
     1     4  
     2     3  
     3     2  
     4     1
```

图 1.1.7

2、在命令窗内写下语句 $i=2;j=2$;而后, 写出表达式, 试对复数 $c=[1+2*i,3+4*i,5+6*i]$, 用两种方式对其进行赋值, 并表达式写出 C 的转置。总结复数赋值时需要注意什么, i 之前的* 符号什么时候才能省略。

答:

第一种方式:直接进行赋值

当 i 已经被赋值了后,在进行复数赋值时使用乘号会把 i 的值代入计算,需要省略乘号进行复数赋值.

```
>> i = 2;  
>> j = 2;  
>> c=[1+2*i, 3+4*i, 5+6*i]  
  
c =  
  
     5     11     17  
  
>> c=[1+2i, 3+4i, 5+6i]  
  
c =  
  
 1.0000 + 2.0000i   3.0000 + 4.0000i   5.0000 + 6.0000i
```

图 1.2.1

第二种方式:先设置好实部和虚部数组后,用 `complex` 创建对象后加入到数组中

```
>> re = [1, 3, 5];  
>> im = [2, 4, 6];  
>> c2 = zeros(1, 3);  
>> for i = 1:3  
    c2(i)=complex(re(i), im(i));  
end  
>> c2  
  
c2 =  
  
    1.0000 + 2.0000i    3.0000 + 4.0000i    5.0000 + 6.0000i  
fx >>
```

图 1.2.2

用 `transpose(c)` 写出 `c` 的转置

```
>> c3=transpose(c)  
  
c3 =  
  
    1.0000 + 2.0000i  
    3.0000 + 4.0000i  
    5.0000 + 6.0000i  
  
>> c  
  
c =  
  
    1.0000 + 2.0000i    3.0000 + 4.0000i    5.0000 + 6.0000i  
fx >>
```

图 1.2.3

```
>> c=[1+2j, 3+4j, 5+6j]

c =

    1.0000 + 2.0000i    3.0000 + 4.0000i    5.0000 + 6.0000i

>> c=[1+2k, 3+4k, 5+6k]
c=[1+2k, 3+4k, 5+6k]
      ↑
错误：表达式无效。请检查缺失的乘法运算符、缺失或不对称的分隔符或者其他语法错误。要构造矩阵，请使用方括号而不是圆括号。

是不是想输入：
>> c=[1+2*k, 3+4*k, 5+6*k]
```

图 1.2.4

总结:

- 1.matlab 中的虚数单位是 i , 在进行复数赋值时可以使用 i 或 j , 此外的字母不能用于复数表示
- 2.如果 i 或 j 已经被进行了赋值, 在进行复数赋值时要省略乘号, 否则 $5*i$ 就会计算出 5 和 i 被赋值的值的乘积的结果
- 3.matlab 可以通过 `complex` 将实部和虚部组合成复数

3、练习基本矩阵 `zeros`, `ones` 命令的使用, 分别写出表达式, 要求构成 N 阶方阵, $M*N$ 阶方阵, 与某任意已知矩阵 A 同阶的矩阵。

答:

```
>> A

A =

     1     2     3     1
     4     5     6     1
     7     8     9     1
    10    11    12     1

>> A1 = zeros(size(A))

A1 =

     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
```

图 1.3.1

```
>> A2 = ones(size(A))

A2 =

     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1
```

图 1.3.2

4、数组赋值：

1)、对时间变量 t 进行赋值，要求：在 0-20 秒间每隔 0.2 秒取一个值，用两种办法实现。
(冒号,linspace)，指明当函数 `linspace(a,b,n)`, n 缺省时，维数是多少？

2)、在命令窗内逐条输入以下命令

`x=logspace(1,4,4)`

`y=linspace(1,4)`

观察结果，说明语句的功能。

答：

1)用冒号实现,0~20, 每 0.2 取一个,一共有 101 个值

```
>> t = 0 : 0.2 : 20

t =

    列 1 至 13

    0    0.2000    0.4000    0.6000    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000    2.2000    2.4000
```

名称	值
t	1x101 double

图 1.4.1

用 linspace 实现

```
>> t1 = linspace(0, 20, 101)

t1 =

    列 1 至 12

    0    0.2000    0.4000    0.6000    0.8000    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000    2.2000

    列 13 至 24
```

名称	值
t	1x101 double
t1	1x101 double

图 1.4.2

linspace(a, b, n)当 n 缺省时,维数是 100, 默认 100 等分

2)

logspace(1, 4, 4)是以 10 为底的对数进行等分, 生成从 10^1 到 10^4 之间的 4 个对数间距点组成的行向量

```
>> x = logspace(1, 4, 4)

x =

    10    100   1000  10000
```

图 1.4.3

linspace(1, 4)生成从 1 到 4 的 100 个等分点

```
>> y = linspace(1, 4)

y =

列 1 至 12

    1.0000    1.0303    1.0606    1.0909    1.1212    1.1515    1.1818    1.2121    1.2424    1.2727    1.3030    1.3333

列 13 至 24
```

名称	值
t	1x101 double
t1	1x101 double
tmp	1x100 double
x	[10,100,1000,1...
y	1x100 double

图 1.4.4

5、矩阵初等运算：

1)、矩阵相加减乘分别需满足什么条件？表达式验证：矩阵和一个标量相加减时，标量会扩展成同阶等元素矩阵。

2)、MATLAB 中矩阵除法分为左除和右除,已知 $A=[1\ 2\ 3\ 4\ 5\ 6]$, $B=[\ 2\ 4\ 0;1\ 3\ 5]$, $C=[1\ 4\ 7;8\ 5\ 2;3\ 6\ 0]$, 分别求解 $A*B$, $A' * B'$, $A*B/C$, $C\backslash A$, $C\backslash A'$, A/C 。说明左除和右除的满足的条件。

答:

1)矩阵加减要满足相加减的两个矩阵是同阶矩阵

```
>> A = [1 2 3; 4 5 6];
>> B = [-6 -5 -4; -3 -2 -1];
>> A + B

ans =

    -5    -3    -1
     1     3     5

>> A - B

ans =

     7     7     7
     7     7     7
```

图 1.5.1

矩阵乘法要满足 乘号左边矩阵的列数 等于 乘号右边矩阵的行数

```
>> A * B
错误使用 *
用于矩阵乘法的维度不正确。请检查并确保第一个矩阵中的列数与第二个矩阵中的行数匹配。要执行按元素相乘，请使用 '.*'。

相关文档

>> C = [2 2 2; 2 2 2; 2 2 2];
>> A * C

ans =

    12    12    12
    30    30    30
```

图 1.5.2

矩阵和一个标量相加减时,标量会自动扩展为同阶等元素矩阵进行运算

```
>> A

A =

     1     2     3
     4     5     6

>> A - 1

ans =

     0     1     2
     3     4     5
```

图 1.5.3

2)

A 是一个 1x6 矩阵, B 是一个 2x3 矩阵, C 是一个 3x3 矩阵.

A' 是一个 6x1 矩阵, B' 是一个 3x2 矩阵.

A*B, A 的列数和 B 的行数不相等, 相乘会报错.

A'*B', A'的列数是 1, B'的行数是 3, 不相等相乘会报错.

```
>> A = [1 2 3 4 5 6]

A =

     1     2     3     4     5     6

>> B = [2 4 0; 1 3 5];
>> C = [1 4 7; 8 5 2; 3 6 0];
>> A * B
错误使用 *
用于矩阵乘法的维度不正确。请检查并确保第一个矩阵中的列数与第二个矩阵中的行数匹配。要执行按元素相乘，请使用 '.*'。
```

[相关文档](#)

```
>> A' * B'
错误使用 *
用于矩阵乘法的维度不正确。请检查并确保第一个矩阵中的列数与第二个矩阵中的行数匹配。要执行按元素相乘，请使用 '.*'。
```

[相关文档](#)

图 1.5.4

$C \backslash A$, 左除式, 相当于 $\text{inv}(C) * A$, $\text{inv}(C)$ 为 3×3 矩阵, 和矩阵 A 1×6 矩阵相乘报错
 $C \backslash A'$, 左除式, 相当于 $\text{inv}(C) * A'$, $\text{inv}(C)$ 为 3×3 矩阵, 和矩阵 A' 6×1 矩阵相乘报错
 A / C , 右除式, 相当于 $A * \text{inv}(C)$, A 1×6 矩阵和 $\text{inv}(C)$ 3×3 矩阵相乘报错。

```
>> C \ A
错误使用 \
矩阵维度必须一致。

>> C \ A'
错误使用 \
矩阵维度必须一致。

>> A / C
错误使用 /
矩阵维度必须一致。
```

二 常用函数的应用

部分题目需要参阅 help，及附件参考文件完成

1、总结函数 max,min,mean,median,sum, trapz, diff 的功能。

max: 返回数组的最大值，如果是矩阵则返回包含每一列的最大值的一个行向量

min: 返回数组的最小值，如果是矩阵则返回包含每一列的最小值的一个行向量

mean: 返回数组的平均值，mean(A,1)返回矩阵每列的平均值，mean(A,2)返回矩阵每行的平均值

median: 返回数组的中位数，median(A)将矩阵各列视为向量 返回中位数值行向量

min max	mean	media
<pre>>> A = [3 8 1; 8 1 2] A = 3 8 1 8 1 2 >> max(A) ans = 8 8 2 >> min(A) ans = 3 1 1</pre>	<pre>>> mean(A, 1) ans = 5.5000 4.5000 1.5000 >> mean(A, 2) ans = 4.0000 3.6667</pre>	<pre>B = 1 2 8 9 3 4 8 9 >> median(B) ans = 6 >> A A = 3 8 1 8 1 2 >> median(A) ans = 5.5000 4.5000 1.5000</pre>

图 2.1.1

sum:返回数组的和, **sum(A)**返回数值为每个列向量的和的行向量, **sum(A, 2)**数值为每个行向量的和的列向量

trapz:通过梯形算法求 **y** 的积分, **trapz(x,y)**返回 **x,y** 所描述的图形的面积

diff:计算向量中后一个元素与前一个元素的差值, **diff(A)**把每一列看做向量计算差值

```
B =  
    1    2    8    9    3    4    8    9  
  
>> sum(B)  
ans =  
  
    44  
  
>> A  
  
A =  
     3     8     1  
     8     1     2  
  
>> sum(A)  
ans =  
  
    11     9     3  
  
>> sum(A, 2)  
ans =  
  
    12  
    11
```

```
>> x = 0:0.01:10;  
y = x;  
plot(x, y)  
>> x = 0:0.01:10;  
>> y = x;  
>> plot(x, y)  
>> trapz(x, y)  
  
ans =  
  
    50
```

```
A =  
     1     2     3  
     5     1     3  
     5     2     6  
  
>> diff(A)  
ans =  
  
     4    -1     0  
     0     1     3  
  
>> B  
  
B =  
     3     5     1     3     6     7     2  
  
>> diff(B)  
ans =  
  
     2    -4     2     3     1    -5
```

图 2.1.2

2、产生 8×6 阶的随机数矩阵 $R1$ ，服从 $N(4,8)$ 的正态分布，并求其各列的平均值和均方差。

产生 4×6 阶的均匀分布随机数矩阵 $R2$ ，并验证其均值是否为 0.5。

答：

```
>> R1 = 4 + 2*sqrt(2)*randn(8, 6)

R1 =

    7.0922    1.8231    8.3677    3.4558    8.0144    4.6100
    7.1375    5.0504    4.2430    6.5134    4.8247    0.7025
    1.5572    3.3620   -0.2189    1.8367    4.5595    0.7531
    4.2188    7.1604    1.9005    0.0338    8.4907    4.2966
    0.5660    0.9197    0.9974   -0.0231    1.7246    6.0428
    0.8505    4.0921   10.6481    5.3808    5.9704   11.3129
    3.9806    5.5628    2.2588    3.4983    6.3620    2.1137
    8.3349    7.1130    6.1159    3.4455    3.3107    4.5299

>> mean(R1)

ans =

    4.2172    4.3854    4.2891    3.0176    5.4071    4.2952

>> std(R1)

ans =

    3.0600    2.2891    3.8059    2.3285    2.2850    3.4365
```

图 2.2.1

```
R2 =

    0.9619    0.8687    0.8001    0.2638    0.5797    0.6221
    0.0046    0.0844    0.4314    0.1455    0.5499    0.3510
    0.7749    0.3998    0.9106    0.1361    0.1450    0.5132
    0.8173    0.2599    0.1818    0.8693    0.8530    0.4018

>> mean(R2(:))

ans =

    0.4969
```

图 2.2.2

3、查询函数 `conv` 的功能，并总结其用法。

答:

`conv` 用于计算两个向量的卷积;

计算一维离散信号的卷积用 `conv`, 如 `res = conv(a, b);`

计算二维离散信号的卷积用 `conv2`, `A B` 是两个进行卷积运算的矩阵, `res = conv(A,B);`

在数字信号处理中可以用与信号滤波, `after = conv(before, x)`, `x` 表示要进行滤波的参数.

4、设方程的根 `x=[-3,-5,-8,-9]`,求它们对应的 `x` 多项式的系数。

答:

`Poly` 可以根据多项式的根, 返回其对应的多项式的系数.

```
>> root = [-3, -5, -8, -9];
>>
>> poly(root)

ans =

           1           25          223          831         1080

>>
```

图 2.4.1

5、设 $f(x) = x^5 - 4x^4 + 3x^2 - 2x + 6$

1)、`x=[-2,8]`之间函数的值(取 100 个点) (提示: 用 `polyval` 函数)

2)、用 `roots` 函数求此多项式的根。

答:1)

```
>> x = linspace(-2, 8, 100);
>> b = [1, -4, 3, -2, 6];
>> f = polyval(b, x);
>> f

f =

1.0e+03 *

列 1 至 13

    0.0700    0.0610    0.0530    0.0459    0.0396    0.0341    0.0292    0.0250    0.0214    0.0184    0.0158    0.0136    0.0118

列 14 至 26

    0.0103    0.0091    0.0082    0.0075    0.0069    0.0065    0.0062    0.0060    0.0058    0.0057    0.0055    0.0054    0.0053

列 27 至 39

    0.0051    0.0049    0.0046    0.0043    0.0039    0.0034    0.0029    0.0023    0.0017    0.0011    0.0004    -0.0003    -0.0010

列 40 至 52

   -0.0016   -0.0022   -0.0028   -0.0032   -0.0035   -0.0037   -0.0036   -0.0034   -0.0028   -0.0020   -0.0008   0.0009   0.0025
```

名称	值
ans	99x1 complex...
b	[1,-4,3,-2,6]
f	1x100 double
R1	0x5 double
R2	4x5 double
root	[-3,-5,-8,-9]
x	1x100 double

图 2.5.1

2)用 roots 函数求多项式的根

```
b =  
  
    1    -4     3    -2     6  
  
>> root = roots(b)  
  
root =  
  
    3.0000 + 0.0000i  
    1.6956 + 0.0000i  
   -0.3478 + 1.0289i  
   -0.3478 - 1.0289i
```

图 2.5.2

【思考题】

1、总结在完成实验的过程中，遇到了哪些问题，如何解决的。

象 `trapz` `diff` 这样陌生的函数不知道他的具体作用和用法，通过在命令行中用 `help + 函数名` 查看解释和文档解决问题。

符合正态分布的随机矩阵通过搜索引擎和 `matlab` 的 `help` 功能解决了问题。

2、说明什么是元素群运算，元素群运算的优势在哪里。

元素群运算是指对一组元素进行特定处理后得到的结果仍然是该元素群中的元素，比如 `matlab` 中的矩阵通过一次运算如加减 转置 矩阵的逆之后得到的结果仍然是元素群中的元素。

抽象性,可以将一些复杂的处理抽象成一个元素群的运算,如将一个多元方程的求根问题可以抽象转化成一个矩阵的变化问题,简化了处理方法,提高了解决问题的可行性。

重复性,一个元素群经过同样大的处理后得到的结果是相同的,可以通过重复处理来判断结果的准确定。

可扩展性,元素群可以根据情况进行扩张,从而可以适应不同的问题情况。

应用领域多,可以在多个领域应用,在物理学 数学 几何学 化学等多个领域,适用性广,可以利用元素群运算更好的处理相关问题。

3、总结幂次运算需要注意什么。

在进行较大数值的幂次运算时要先对数值进行一个预估避免出现数值过大或过小而发生的溢出问题，从而导致运算结果错误。

矩阵的幂次运算需要注意矩阵的维数是否满足乘法需求。

在进行矩阵的幂次运算时要注意区分 `^` 和 `.^`，前者是矩阵的乘方要保证是方阵，后者是对矩阵的每个元素进行乘方。

三 MATLAB 程序设计

1、在文本编辑窗内输入以下程序段，并存储调用，分析该程序的功能。

答:

```
x=0:pi/20:pi/3;  
y=sin(x); % x 对应的正弦值 y  
p=polyfit(x,y,5) % 用最小二乘法对 x 和 y 进行多项式拟合,得到一个有 5 个多项式系数  
% 的向量 p  
x1=0:pi/30:pi*2;  
y1=sin(x1); % x1 对应的正弦值 y1  
y2=polyval(p,x1); % 通过 p 的多项式系数利用 polyval 函数计算该多项式在 x1 各点的取  
% 值  
plot(x1,y1,'b-',x1,y2,'r*') % 画出原始曲线 x1-y1 和 拟合曲线 x1-y2  
legend('原曲线','拟合曲线')  
axis([0 7 -1.2 4]) % 设置坐标轴显示范围 x0~7 y-1.2~4
```

程序的功能是通过多项式拟合来逼近正弦函数并会出图像进行比较

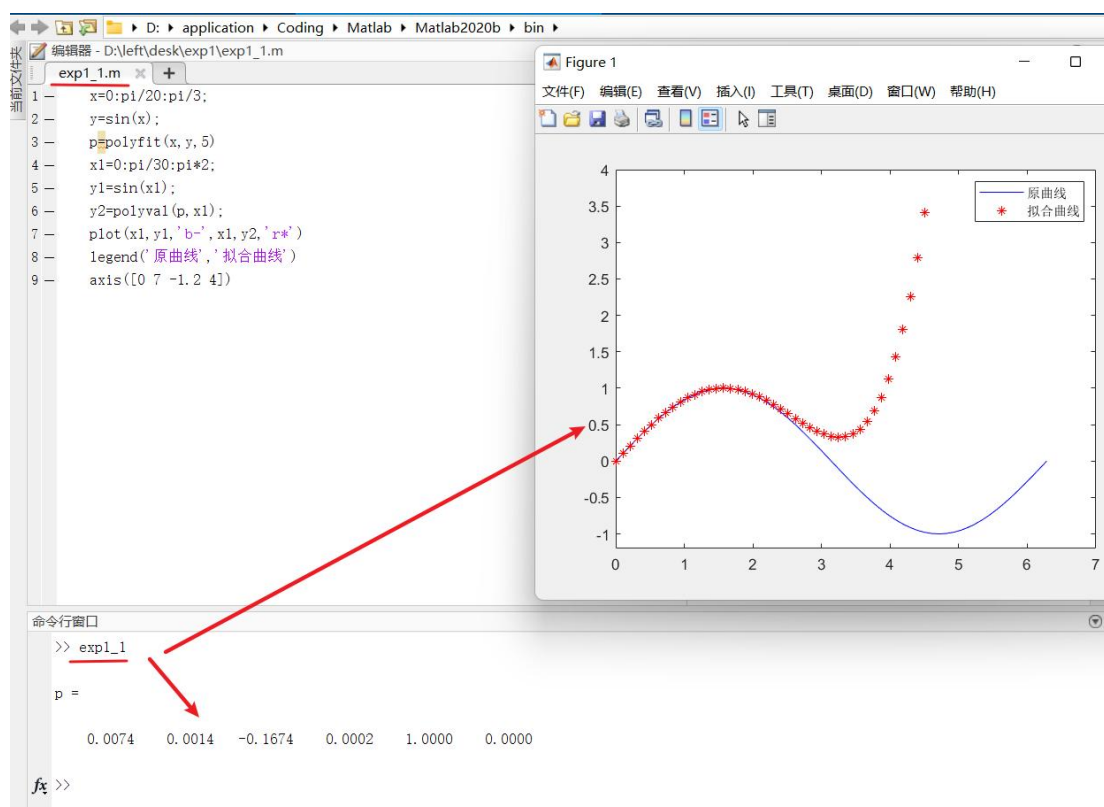
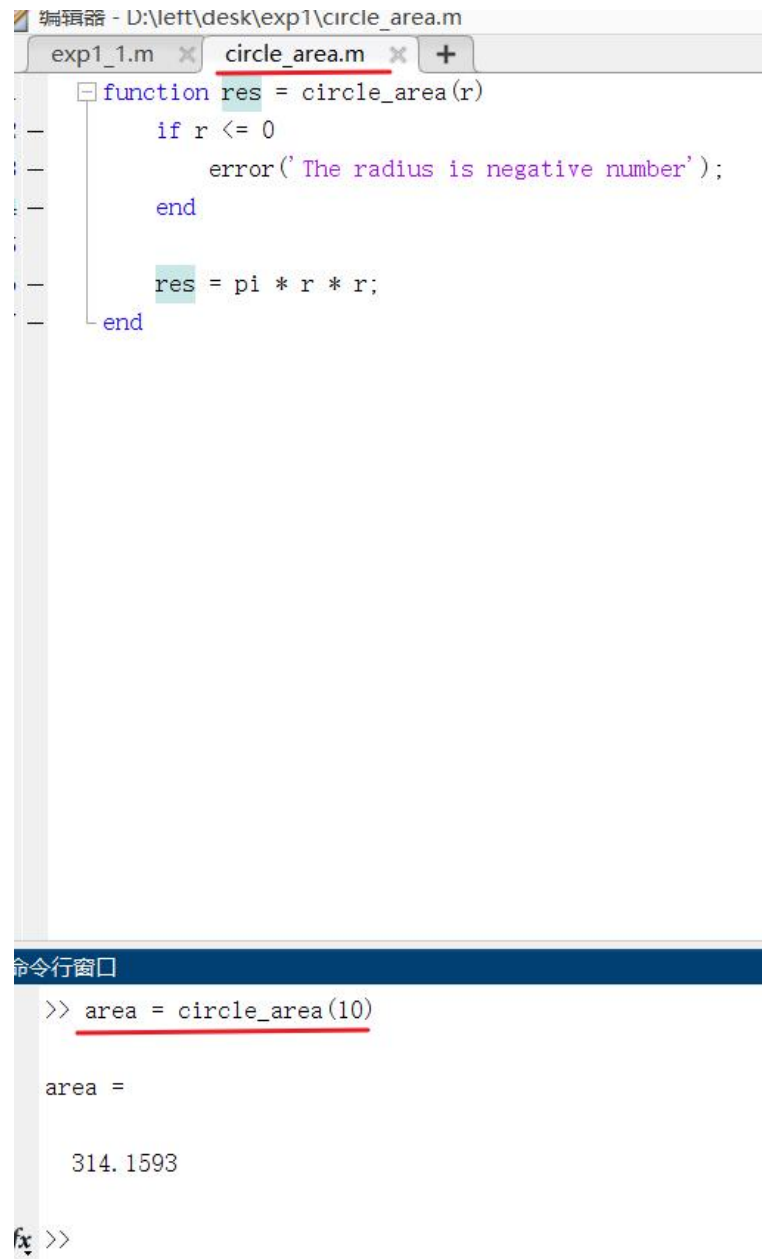


图 3.1.1

2、编写一个通用函数文件，求圆的面积，在命令窗给定半径，调用该函数求解。

答:



The image shows a MATLAB environment with two windows. The top window is the editor, titled '编辑器 - D:\left\desk\exp1\circle_area.m'. It contains the following code:

```
function res = circle_area(r)
    if r <= 0
        error(' The radius is negative number');
    end
    res = pi * r * r;
end
```

The bottom window is the command window, titled '命令行窗口'. It shows the execution of the function:

```
>> area = circle_area(10)

area =

    314.1593

fx >>
```

图 3.2.1

3、选择结构

1)、输入一个百分制成绩，要求输出成绩等级 A、B、C、D、E。其中 90 分~100 分为 A，80 分~89 分为 B，70 分~79 分为 C，60 分~69 分为 D，60 分一下为 E。

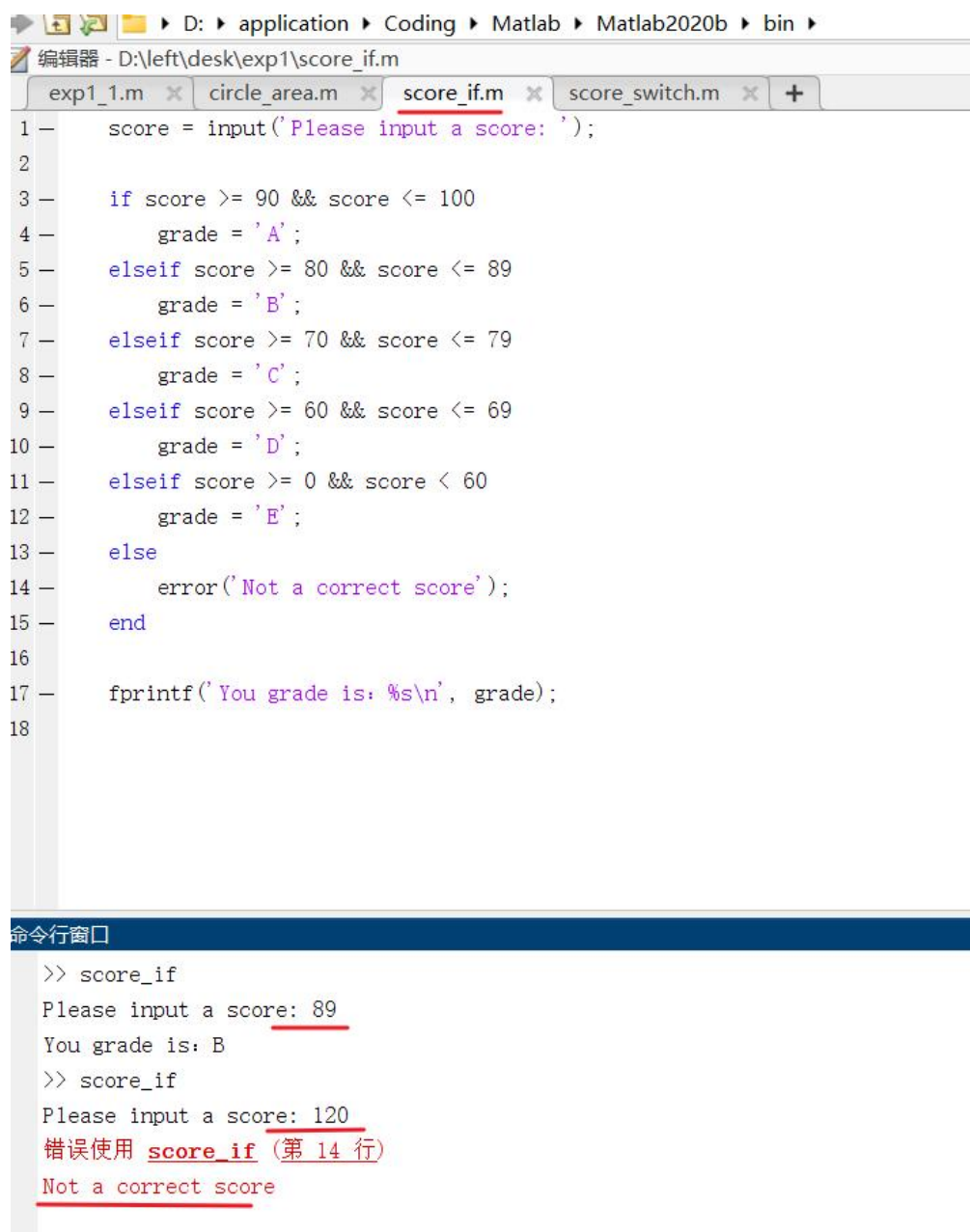
要求：

(a)分别用 if 语句和 switch 语句实现。

(b)输入百分制成绩后要判断该成绩的合理性，对不合理的成绩应输出出错信息。

答：

If 实现



```

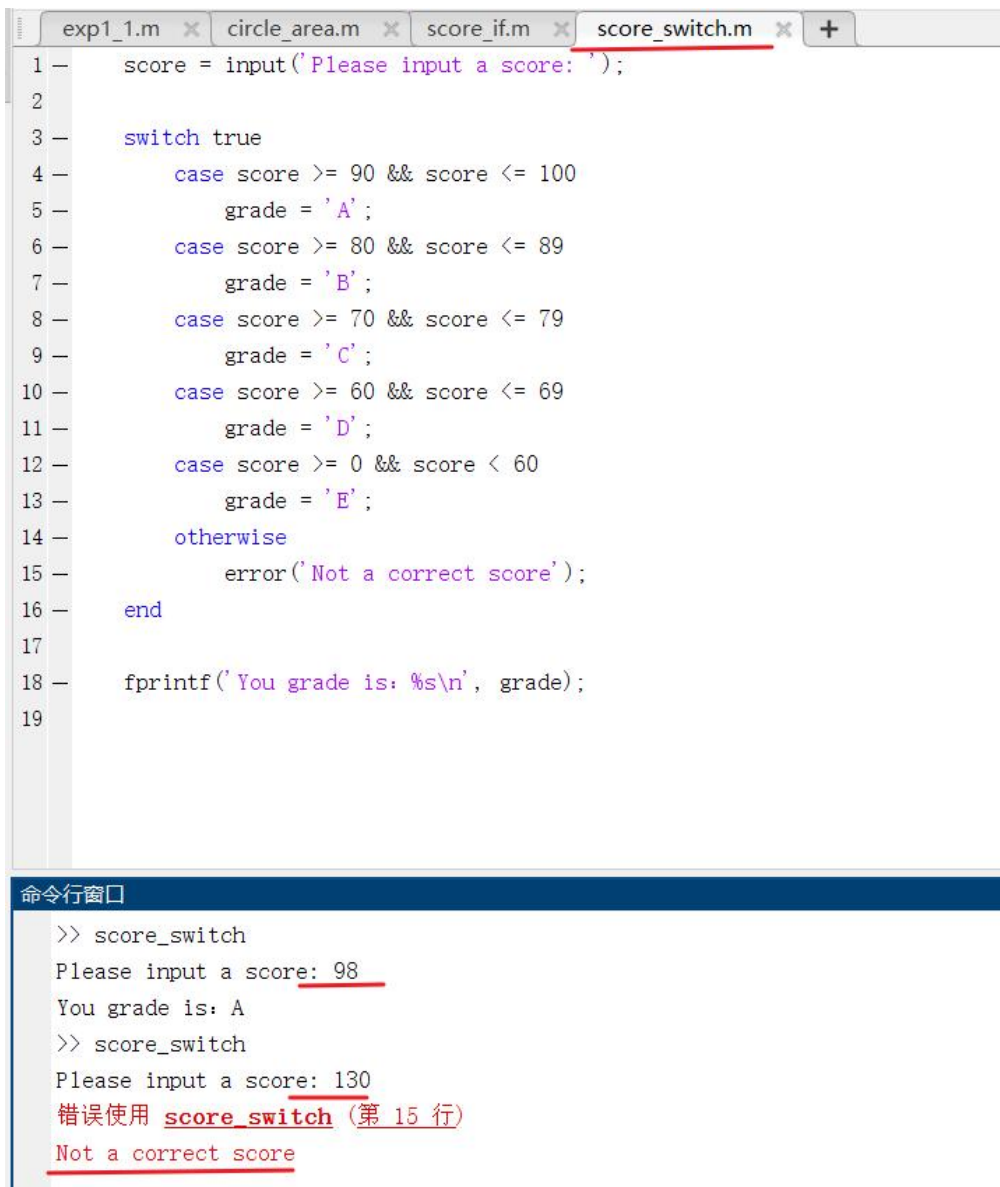
D:\application\Coding\Matlab\Matlab2020b\bin
编辑器 - D:\left\desk\exp1\score_if.m
exp1_1.m x circle_area.m x score_if.m x score_switch.m x +
1 - score = input('Please input a score: ');
2
3 - if score >= 90 && score <= 100
4 -     grade = 'A';
5 - elseif score >= 80 && score <= 89
6 -     grade = 'B';
7 - elseif score >= 70 && score <= 79
8 -     grade = 'C';
9 - elseif score >= 60 && score <= 69
10 -     grade = 'D';
11 - elseif score >= 0 && score < 60
12 -     grade = 'E';
13 - else
14 -     error('Not a correct score');
15 - end
16
17 - fprintf('You grade is: %s\n', grade);
18

命令窗口
>> score_if
Please input a score: 89
You grade is: B
>> score_if
Please input a score: 120
错误使用 score_if (第 14 行)
Not a correct score

```

图 3.3.1

Switch 实现



```
exp1_1.m x circle_area.m x score_if.m x score_switch.m x +
1 — score = input('Please input a score: ');
2
3 — switch true
4 —     case score >= 90 && score <= 100
5 —         grade = 'A';
6 —     case score >= 80 && score <= 89
7 —         grade = 'B';
8 —     case score >= 70 && score <= 79
9 —         grade = 'C';
10 —    case score >= 60 && score <= 69
11 —        grade = 'D';
12 —    case score >= 0 && score < 60
13 —        grade = 'E';
14 —    otherwise
15 —        error('Not a correct score');
16 —    end
17
18 — fprintf('You grade is: %s\n', grade);
19
```

命令行窗口

```
>> score_switch
Please input a score: 98
You grade is: A
>> score_switch
Please input a score: 130
错误使用 score_switch (第 15 行)
Not a correct score
```

图 3.3.2

4、循环结构

1)、 根据 $\pi^2/6 = 1/1^2 + 1/2^2 + 1/3^2 + \dots + 1/n^2$, 求 π 的近似值。当 n 分别取 100, 1000, 10000 时, 结果是多少?

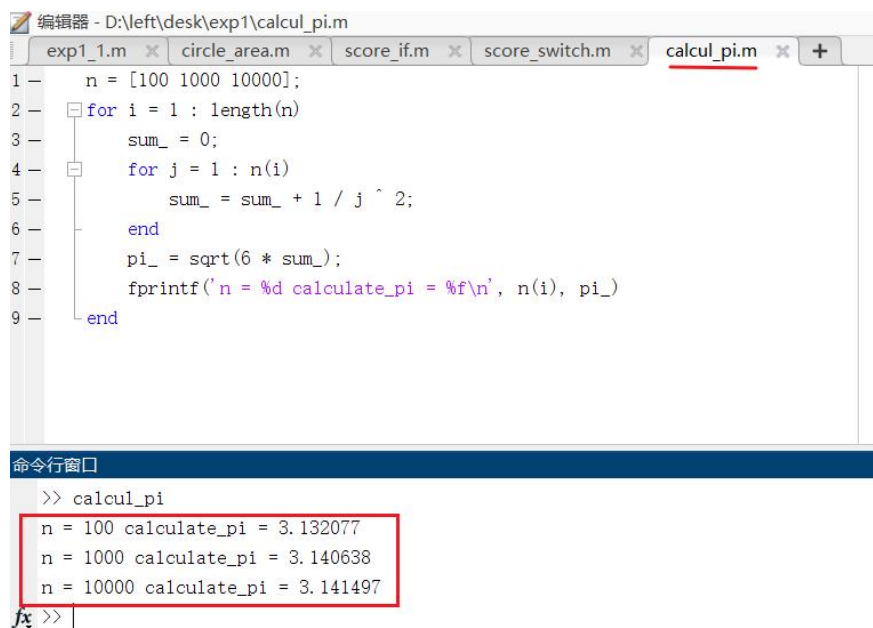
2)、 根据 $y = 1 + 1/3 + 1/5 + \dots + 1/(2n-1)$, 求

(1) $y < 3$ 时的最大 n 值

(2) 与 (1) 的 n 值对应的 y 值

答:

1)

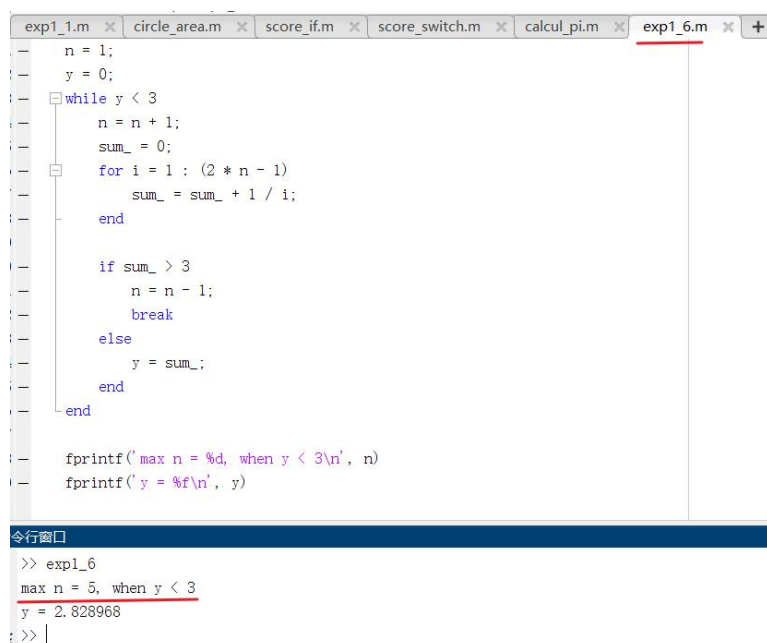


```
编辑器 - D:\left\desk\exp1\calcul_pi.m
exp1_1.m x circle_area.m x score_if.m x score_switch.m x calcul_pi.m x +
1 n = [100 1000 10000];
2 for i = 1 : length(n)
3     sum_ = 0;
4     for j = 1 : n(i)
5         sum_ = sum_ + 1 / j ^ 2;
6     end
7     pi_ = sqrt(6 * sum_);
8     fprintf('n = %d calculate_pi = %f\n', n(i), pi_)
9 end

命令行窗口
>> calcul_pi
n = 100 calculate_pi = 3.132077
n = 1000 calculate_pi = 3.140638
n = 10000 calculate_pi = 3.141497
>>
```

图 3.4.1

2)



```
exp1_1.m x circle_area.m x score_if.m x score_switch.m x calcul_pi.m x exp1_6.m x +
1 n = 1;
2 y = 0;
3 while y < 3
4     n = n + 1;
5     sum_ = 0;
6     for i = 1 : (2 * n - 1)
7         sum_ = sum_ + 1 / i;
8     end
9
10    if sum_ > 3
11        n = n - 1;
12        break
13    else
14        y = sum_;
15    end
16 end
17
18 fprintf('max n = %d, when y < 3\n', n)
19 fprintf('y = %f\n', y)

命令行窗口
>> exp1_6
max n = 5, when y < 3
y = 2.828968
>>
```

图 3.4.2

四 MATLAB 绘图

1、绘制出相同周期，不同相位的两个正弦曲线在 $t=[0 \ 4\pi]$ 区间上的图形，要求使用不同的线型，颜色，并对图形添加横纵坐标，注释，标题，图注信息，要求横坐标显示范围为 $(0 \ 4\pi)$ ，纵坐标显示范围为正弦曲线幅值的最小最大值。

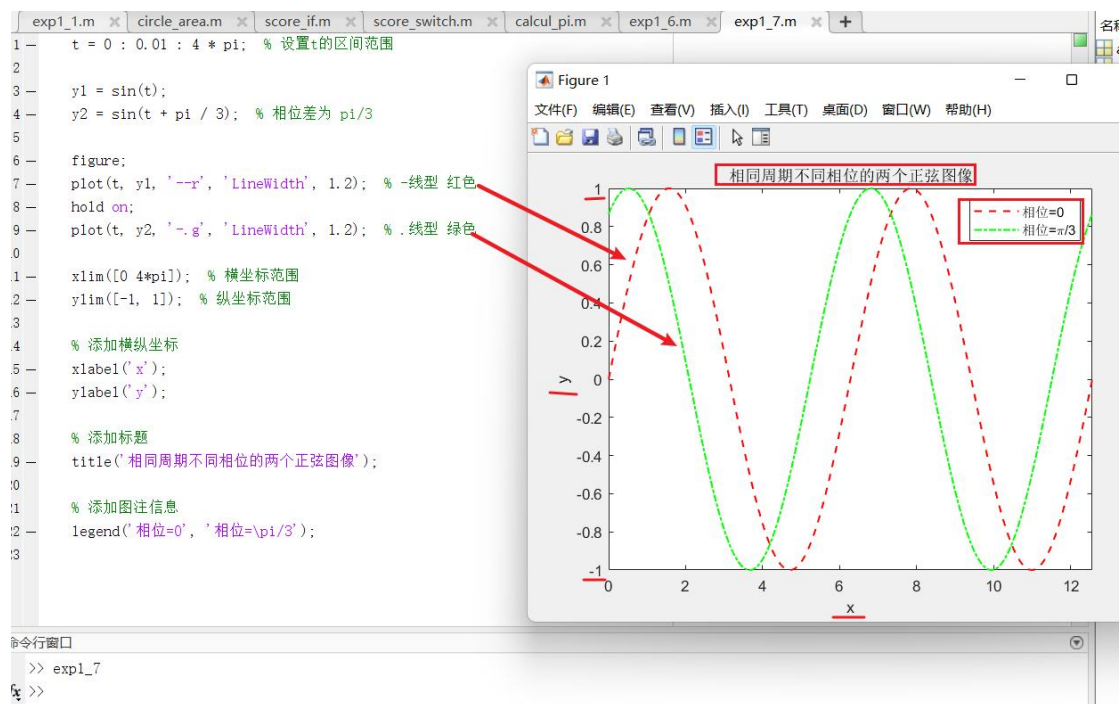


图 4.1.1

2、已知复数 $Z=\exp((-2-j)*t)$,要求将图形窗分成 6 个子图, 分别绘制出 z 的实部, 虚部, 模, 相角随 t 变化的曲线, 极坐标图, 及实虚部的关系图。

```
exp1_1.m x circle_area.m x score_if.m x score_switch.m x
1 t = linspace(0, 2 * pi, 1000);
2 z = @(t)exp((-2 - 1j) * t); % 匿名函数用z(t)调用函数
3 subplot(2, 3, 1);
4 % 实部
5 plot(t, real(z(t)));
6 title('实部');
7 xlabel('t');
8 ylabel('real');
9
10 subplot(2, 3, 2);
11 % 虚部
12 plot(t, imag(z(t)));
13 title('虚部');
14 xlabel('t');
15 ylabel('imag');
16
17 subplot(2, 3, 3);
18 % 模
19 plot(t, abs(z(t)));
20 title('模');
21 xlabel('t');
22 ylabel('|z|');
23
24 subplot(2, 3, 4);
```

图 4.2.1

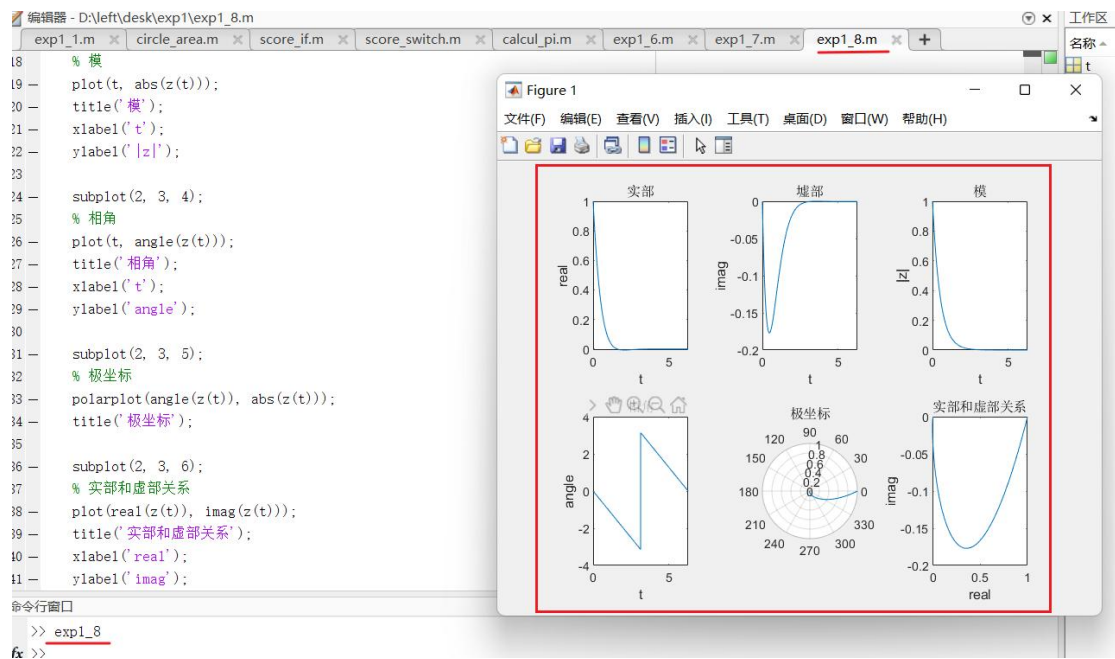


图 4.2.2

3、

命令窗内输入以下内容：

```
t=0:0.5:4*pi; y=exp(-0.1*t).*sin(t);  
subplot(2,2,1),stem(t,y), title('stem(t,y)')  
subplot(2,2,2),stairs(t,y),title('stairs(t,y)')  
subplot(2,2,3),bar(t,y),title('bar(t,y)'),axis([0,20,-1,1])  
subplot(2,2,4),fill(t,y,'r'),title('fill(t,y,"r")')
```

观察并分析 stem,stairs,bar,fill 命令的显示效果。

Stem 是每一个样本点一个向坐标轴的, 适合绘制离散信号

Stairs 绘制的是阶梯形状的图形,每个样本点有一个水平线和垂直线连接

Bar 是绘制的柱状图

Fill 会填充满波形图围城的面积

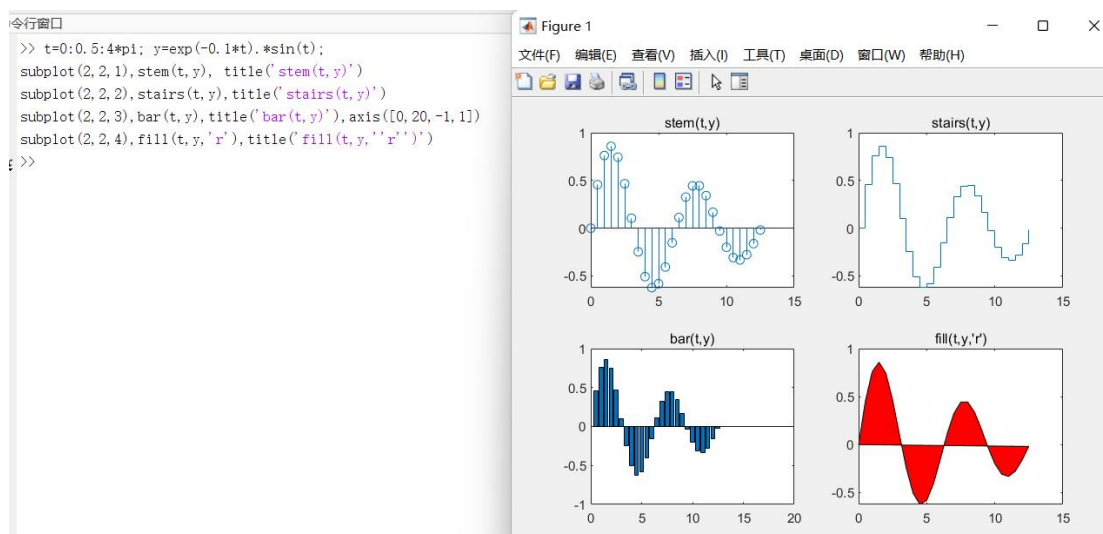


图 4.3.1

【思考题】

1、写出 hold 命令的用法

Hold on 会在原来的图形中继续画图,不会进行覆盖掉

Hold off 如果有两个图形后边的画图会覆盖掉前边的图形

进行两个图形拟合比较时,开启 hold on 可以在一个图上对比两个图形的拟合程度.

2、总结交互式命令, input,display,pause,keyboard,fprintf, sprintf 命令的功能及使用方法

input:用于用户输入数据, a = input('input a number'); 用户输入一个数据后会存入到 a 中

display:用于在命令窗口显示文本, display('liuyunuo');

pause:可以暂停程序执行,用户按下任意键后继续执行

keyboard:程序执行到 keyboard 后用户可以用键盘输入命令操作 matlab,输入 dbquit 后会退出键盘模式继续执行

fprintf:将文本输出到文件或屏幕上 fprintf('a = %d, b = %d', a, b);

`sprintf`:用于将文本格式化存储在一个字符串变量中 `string_=(‘a = %d, y = %d’, a, b);`

【实验总结】

通过对实验原理的学习对 **Matlab** 的基本操作有了初步的了解。经过在 **Matlab** 实验的上机实际操作，这 **Matlab** 的操作方法有了更深入的理解，在实验过程中也遇到了很多苦难，比如要求使用的函数不知道函数的作用 对函数的使用方法不够了解，通过阅读函数的文档说明和搜索查询解决了问题，在这个过程中也锻炼了自己查阅资料的能力。

matlab 为 **matrix lab** 的简写，为矩阵实验室，在实验过程中基本都是以矩阵的形式完成各个任务的求解，一切以矩阵为基础进行操作。对于要经常调用的但是没有现成函数的方法，可以自己写一个脚本保存，写入路径后可以在命令行中方便调用使用。画图模块的练习，体会到了 **Matlab** 的绘图能力，一个比较抽象的方程或式子可以通过图像的方式更加方便的理解。**conv** 函数的学习，**conv** 可以用于滤波，和学习的信号系统课程也相联系。

通过此次实验，受益匪浅，也为后期的信号系统的学习和数据处理打下了基础。