可编程计算器 测试报告

姓名: **崔冠宇**　　学号: **2018202147**

(使用LaTeX编辑)

注: 测试运行环境为macOS Mojave 10.14.6, 终端是iTerm2, Shell是Oh My Zsh. 以下截图均基于此环境.

1　模块1——向量计算

1.1　测试数据

1. $(1.5, 2) - (-3, -2.5)$
2. $(1.5, 2.5, 3.5) - (1.5, 2.5, 3.5)$
3. $(0, 0) + (0, 0)$
4. $(1, 1, 1) + (-1, -1, -1)$

1.2　测试结果

结果一　(展示了输入过程, 之后的测试图片由于篇幅所限仅给出结果)

```
以下是向量运算(顺序表)演示:
请输入向量维度:2
请输入向量A:1.5 2
请输入向量B:-3 -2.5
请输入运算符(+/-):-
结果:4.5 4.5
```

结果二

```
结果:0 0 0
```

结果三
```
结果:0 0
```

结果四
```
结果:0 0 0
```

2 模块2——向量夹角余弦值

2.1 测试数据

1. $(1, 1)$, $(1, 1)$
2. $(1.5, 1.5)$, $(-2, -2)$
3. $(1, 0, 0)$, $(0, 1, 0)$
4. $(2.5, 0)$, $(1, 1)$

2.2 测试结果

结果一 (展示了输入过程, 之后的测试图片由于篇幅所限仅给出结果)
```
以下是向量夹角余弦值(顺序表)演示:
请输入向量维度:2
请输入向量A:1 1
请输入向量B:1 1
结果:1
```

结果二
```
结果:-1
```

结果三
```
结果:0
```

结果四
```
结果:0.707107
```

3 模块3、模块6——多项式运算(加减)

这两个模块功能相同, 只是由不同的数据结构实现, 所以把它们一起测试. 以下几个模块情况相同.

3.1 测试数据

1. $(-2x^2+x-1)+(-x^3-2x)$

2. $(x^3+x-1)-(x^3+5.5x)$

3. $(1.5x+1)-(1.5x+1)$

4. $(x^{50}+x-1)-(x+1)$

3.2 测试结果

结果一 (展示了输入过程, 之后的测试由于篇幅所限仅给出结果)

```
以下是一元多项式运算(顺序表)演示:
请输入第一个多项式项数:3
请按次数降序输入多项式，两个数字为一项，前者为系数，后者为次数(0也要写):
例: 3x^2+2x+1 => 3 2 2 1 1 0
-2 2 1 1 -1 0
你输入的多项式为:-2x^2+x-1
请输入第二个多项式项数:2
请按次数降序输入多项式，两个数字为一项，前者为系数，后者为次数(0也要写):
例: 3x^2+2x+1 => 3 2 2 1 1 0
-1 3 -2 1
你输入的多项式为:-x^3-2x
请输入运算符(+/-):+
结果:-x^3-2x^2-x-1
```

结果二
```
结果:-4.5x-1
```

结果三
```
结果:0
```

结果四
```
结果:x^50-2
```

4 模块4、模块7——多项式乘法

4.1 测试数据

1. $(1.5x^2 + 2x - 1) \times (-x + 1)$
2. $(-x^5 + x^3 - x - 1) \times (x^2 + 3x + 1)$
3. $0 \times (x^5 - 2x + 1)$
4. $2 \times x^2$

4.2 测试结果

结果一 (输入方式与模块3、6相同)

```
结果:-1.5x^3-0.5x^2+3x-1
```

结果二

```
结果:-x^7-3x^6+3x^4-4x^2-4x-1
```

结果三

```
结果:0
```

结果四

```
结果:2x^2
```

5 模块5、模块8——多项式导数(任意阶数)

5.1 测试数据

1. $(x^3 + 2x + 2)$, 一阶导.
2. $(1.5x + 1)$, 二阶导.
3. $(-x^2 + 2x)$, 二阶导.
4. $(x^5 + x + 1)$, 四阶导.

5.2 测试结果

结果一 (输入方式与模块3、6相同)

```
你输入的多项式为:x^3+2x+2
请输入求导次数(>=1): 1
结果:3x^2+2
```

结果二

```
结果:0
```

结果三

```
结果:-2
```

结果四

```
结果:120x
```

6 模块9——表达式求值/可编程计算器

6.1 测试数据

- DEF f(x)=2+3*(x+1)^2
- DEF g(x)=f(f(x+2)+1)
- RUN f(1.5)+g(1)
- RUN f(f(1))
- RUN 1.5/x+2*(y-1)^2 //x=1.5, y=3
- RUN 5^(-1)+3^(1/2)
- RUN 1.0/0
- QUIT

6.2 测试结果

结果 (由于是交互式界面, 测试数据合为一张图)

```
以下是表达式求值/可编程计算器演示:
"Function C0mp1ler" Written by G.Cui.   Ver.0.1
请输入命令(DEF定义函数, RUN求值, QUIT退出):
>>>DEF f(x)=2+3*(x+1)^2
函数f定义成功.
>>>DEF g(x)=f(f(x+2)+1)
函数g定义成功.
>>>RUN f(1.5)+g(1)
结果: 8134.75
>>>RUN f(f(1))
结果: 677
>>>RUN 1.5/x+2*(y-1)^2
请输入标识符x的值:1.5
请输入标识符y的值:3
结果: 9
>>>RUN 5^(-1)+3^(1/2)
结果: 1.93205
>>>RUN 1.0/0
结果: inf
>>>QUIT
```

(经过计算验证, 上述结果是正确的)

7 模块10——矩阵计算

7.1 测试数据

(加减乘实现容易, 不在此演示)

1. $\begin{bmatrix} 0 & 2 \\ 5 & 4 \end{bmatrix}$, INV

2. $\begin{bmatrix} 2 & 1 & 3 \\ 3 & 5 & 1 \\ 6 & 5 & 1 \end{bmatrix}$, DET

3. RDINT, 3 3, -5 5, 1000 //指令(整数随机矩阵测试), 行数与列数, 随机矩阵元素最小值与最大值, 特征值(QR算法)迭代次数.

7.2 测试结果

结果一 (展示了输入过程, 之后的测试由于篇幅所限仅给出结果)

```
以下是矩阵运算演示:
请输入所需的运算
(+/-/*/DET(行列式)/INV(逆矩阵)/
RANK(秩)/TRACE(迹)/
QR(QR分解)/EIGEN(特征值与特征向量))/
RDINT(随机整数矩阵测试)/RDDBL(随机浮点数矩阵测试):INV
请输入矩阵的行数与列数:2 2
请按行顺序输入矩阵的数据:
0 2 5 4
你输入的矩阵为:
[   0.00000,    2.00000]
[   5.00000,    4.00000]
结果:
[  -0.40000,    0.20000]
[   0.50000,    0.00000]
```

结果验证 我们用Wolfram Alpha验证所得结果的正确性. (下同)

Basic Step-by-Step Solution

Decimal form | Use the inverse formula

Inverse

Find the inverse:

$$\begin{pmatrix} 0 & 2 \\ 5 & 4 \end{pmatrix}^{-1}$$

Using a formula for the inverse of a 2×2 matrix, $\begin{pmatrix} 0 & 2 \\ 5 & 4 \end{pmatrix}^{-1} = \frac{1}{0\times 4 - 5\times 2}\begin{pmatrix} 4 & -2 \\ -5 & 0 \end{pmatrix}$:

$$\frac{1}{0\times 4 - 5\times 2}\begin{pmatrix} 4 & -2 \\ -5 & 0 \end{pmatrix}$$

Simplify: $\frac{1}{0\times 4 - 5\times 2} = -\frac{1}{10}$:

Answer:

$$-\frac{1}{10}\begin{pmatrix} 4 & -2 \\ -5 & 0 \end{pmatrix}$$

结果二

```
你输入的矩阵为：
[    2.00000,     1.00000,     3.00000]
[    3.00000,     5.00000,     1.00000]
[    6.00000,     5.00000,     1.00000]
结果：
-42.00000
```

结果验证

The determinant of an upper triangular matrix is the product of its diagonal elements:

Answer:

$$= -\begin{vmatrix} 6 & 5 & 1 \\ 0 & \frac{5}{2} & \frac{1}{2} \\ 0 & 0 & \frac{14}{5} \end{vmatrix} = -\frac{6\times 5\times 14}{2\times 5} = -42$$

结果三　(由于是随机矩阵演示, 每次运行结果不同)

```
A=
[   4.00000,    1.00000,    1.00000]
[   0.00000,    2.00000,    1.00000]
[  -3.00000,    0.00000,    3.00000]

|A|=27.00000

A^-1=
[   0.22222,   -0.11111,   -0.03704]
[  -0.11111,    0.55556,   -0.14815]
[   0.22222,   -0.11111,    0.29630]

rank(A)=3

trace(A)=9.00000

QR Decomposition:
Q=
[  -0.80000,   -0.17241,   -0.57470]
[   0.00000,   -0.95783,    0.28735]
[   0.60000,   -0.22988,   -0.76626]

R=
[  -5.00000,   -0.80000,    1.00000]
[   0.00000,   -2.08806,   -1.81987]
[   0.00000,    0.00000,   -2.58613]

Eigenvalues and eigenvectors of A:
注意：当为Hessenberg/Symmetric Matrix时，结果是准确的，否则只有部分结果是准确的！
(实)特征值:1.54384
(单位)特征向量:
[   0.39296,   -0.86144,    0.32172]
```

结果验证

WolframAlpha

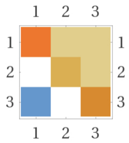
Input

$$\begin{pmatrix} 4 & 1 & 1 \\ 0 & 2 & 1 \\ -3 & 0 & 3 \end{pmatrix}$$

Dimensions

3 (rows) × 3 (columns)

Matrix plot



Trace

9

Step-by-step solution

Determinant

27

Step-by-step solution

Inverse

$$\begin{pmatrix} 0.222222 & -0.111111 & -0.037037 \\ -0.111111 & 0.555556 & -0.148148 \\ 0.222222 & -0.111111 & 0.296296 \end{pmatrix}$$

Exact form Step-by-step solution

Characteristic polynomial

$-\lambda^3 + 9\lambda^2 - 29\lambda + 27$

Step-by-step solution

Eigenvalues

$\lambda_1 \approx 3.72808 + 1.89481\,i$

$\lambda_2 \approx 3.72808 - 1.89481\,i$

$\lambda_3 \approx 1.54384$

Exact forms | Step-by-step solution

Eigenvectors

$v_1 \approx (-0.242694 - 0.631604\,i,\ 0.262763 - 0.288115\,i,\ 1)$

$v_2 \approx (-0.242694 + 0.631604\,i,\ 0.262763 + 0.288115\,i,\ 1)$

$v_3 \approx (0.485388,\ -2.19219,\ 1)$

Exact forms | Step-by-step solution

(注: 我们的特征值算法只给出实特征值和单位化的特征向量. 有少许误差可能是迭代次数不够多的缘故, 也可能是矩阵条件数太大(微小扰动对解的干扰太强)的问题, 详见《数值分析》.)

WolframAlpha

QR decomposition {{4,1,1},{0,2,1},{-3,0,3}}

Input

QR decomposition $\begin{pmatrix} 4 & 1 & 1 \\ 0 & 2 & 1 \\ -3 & 0 & 3 \end{pmatrix}$

Result

$A = Q^{\dagger}.R$

where

$$A = \begin{pmatrix} 4 & 1 & 1 \\ 0 & 2 & 1 \\ -3 & 0 & 3 \end{pmatrix}$$

$$Q = \begin{pmatrix} 0.8 & 0 & -0.6 \\ 0.172409 & 0.957826 & 0.229878 \\ 0.574696 & -0.287348 & 0.766261 \end{pmatrix}$$

$$R = \begin{pmatrix} 5 & 0.8 & -1 \\ 0 & 2.08806 & 1.81987 \\ 0 & 0 & 2.58613 \end{pmatrix}$$

Exact forms

(注: 注意Wolfram Alpha给出的形式是$A = Q^T R$, 而我们的是$A = QR$, 所以结果相差

一个转置, 同时我们的QR矩阵与Wolfram Alpha的结果各自符号相反.)

8 测试结论

每个模块进行了一定的测试(有很多测试因为篇幅原因没有加入报告中, 结果均为正确)后, 我们有理由相信我们设计的计算器具有一定程度的健壮性与可靠性, 可以在一定范围内使用.