小测七

# 1、自定义三个三维向量a、b，c，求它们任意两个的点积、矢积，并求它们的混合积、二重矢积。

**代码：**

close all;clear;clc;

a=[1,3,7]; b=[3,4,9]; c=[1,2,5];

disp('任意两个的点积')

dot1 = dot(a,b) % 点积

dot2 = dot(a,c)

dot2 = dot(b,c)

disp('任意两个的矢积')

cross1 = cross(b,c) % 矢积

cross2 = cross(a,c)

disp('混合积')

A3 = dot(a,cross(b,c)) % 混合积

disp('二重矢积')

A4 = cross(a,cross(b,c)) % 二重矢积

结果：

任意两个的点积

dot1 =

78

dot2 =

42

dot2 =

56

任意两个的矢积

cross1 =

2 -6 2

cross2 =

1 2 -1

混合积

A3 =

-2

二重矢积

A4 =

48 12 -12

# 2、用系数的形式自定义一个7次多项式yl和一个2次多项式y2，将它们用符号出来，并完成：

1）求y1（-3），y1（0），y1（3）的值语句

2）求y1=0的根的语句

3）求yl\*y2，y1/y2的二阶导数，显示结果的多项式形式

y1 = 2\*x^7 - 3\*x^6 + 19\*x^5 - 63\*x^4 + 155\*x^3 + 22\*x^2 - 59\*x + 21

y2 = 100\*x^2 + 26\*x + 31

**代码：**

close all;clear;clc;

y1 = [2, -3, 19, -63, 155, 22, -59, 21]; % exp5\_31.m 多项式的描述

Y1 = poly2sym(y1) % 符号形式多项式输出

y2 = [0, 0, 0, 0, 0, 100, 26, 31];

Y2 = poly2sym(y2) % 符号形式多项式输出

% 1）求y1（-3），y1（0），y1（3）的值

y1\_f3 = polyval(y1,-3) % y1（-3）

y1\_0 = polyval(y1,0) % y1（0）

y1\_3 = polyval(y1,3) % y1（3）

% 2）求 y1=0 的根的语句

Gen = roots(y1) % y1=0 的根

% 3）求 yl\*y2，y1/y2 的二阶导数，显示结果的多项式形式

% y1y2 = conv(y1,y2); % 多项式的乘法 exp5\_34

% y1convy2 = poly2sym(y1y2) % 输出多项式

% [q,r] = deconv(y1,y2); % 多项式的除法，q：商多项式，r：余多项式

% y1deconvy2 = poly2sym(q) % 输出多项式

fprintf('乘法的导数：\n')

k = polyder(y1,y2); % 求多项式 y1 y2 乘积的导数

K = poly2sym(k);

k2 = polyder(k); % 再求一次导数

K = poly2sym(k2) % 输出

% Q = poly2sym(y1) % 分子

% R = poly2sym(y2) % 分母

[q1,r1] = polyder(y1,y2); % 求 y1 y2 相除的一阶导数

% Q = poly2sym(q1) % 分子

% R = poly2sym(r1) % 分母

[q2,r2] = polyder(q1,r1); % 求 y1 y2 相除的二阶导数

fprintf('除法的导数：\n分子：')

Q = poly2sym(q2) % 分子

fprintf('分母：')

R = poly2sym(r2) % 分母

% 化简

fprintf('分子/分母：')

F = Q/R % 仅仅是组合

fprintf('化简：')

f = simplify(F) % 化简

% 经过简单函数实测，本算法结果和手算和算法不一样的原因是，算法并没有分解因式并化简

% 导致分子分母可能存在可以化简的公因式所以最后进行化简操作会变的干净利落

结果：

Y1 =

2\*x^7 - 3\*x^6 + 19\*x^5 - 63\*x^4 + 155\*x^3 + 22\*x^2 - 59\*x + 21

Y2 =

100\*x^2 + 26\*x + 31

y1\_f3 =

-20070

y1\_0 =

21

y1\_3 =

5928

Gen =

-1.1769 + 3.3079i

-1.1769 - 3.3079i

1.9248 + 1.6842i

1.9248 - 1.6842i

-0.7042 + 0.0000i

0.3543 + 0.2437i

0.3543 - 0.2437i

乘法的导数：

K =

14400\*x^7 - 13888\*x^6 + 79128\*x^5 - 176970\*x^4 + 289020\*x^3 + 51324\*x^2 - 3138\*x + 2496

除法的导数：

分子：

Q =

40000000\*x^11 - 640000\*x^10 + 137665600\*x^9 - 41459040\*x^8 + 40789440\*x^7 - 122458032\*x^6 - 270643320\*x^5 - 10120746\*x^4 + 177307116\*x^3 + 90935164\*x^2 + 72805814\*x + 1103104

分母：

R =

100000000\*x^8 + 104000000\*x^7 + 164560000\*x^6 + 103750400\*x^5 + 83264176\*x^4 + 32162624\*x^3 + 15814216\*x^2 + 3098264\*x + 923521

分子/分母：

F =

(40000000\*x^11 - 640000\*x^10 + 137665600\*x^9 - 41459040\*x^8 + 40789440\*x^7 - 122458032\*x^6 - 270643320\*x^5 - 10120746\*x^4 + 177307116\*x^3 + 90935164\*x^2 + 72805814\*x + 1103104)/(100000000\*x^8 + 104000000\*x^7 + 164560000\*x^6 + 103750400\*x^5 + 83264176\*x^4 + 32162624\*x^3 + 15814216\*x^2 + 3098264\*x + 923521)

化简：

f =

(400000\*x^9 - 110400\*x^8 + 1281360\*x^7 - 713520\*x^6 + 196188\*x^5 - 1054398\*x^4 - 2493108\*x^3 + 873864\*x^2 + 2318730\*x + 35584)/(100\*x^2 + 26\*x + 31)^3

# 第二题用 TestFunction.m 文件测试了算法的正确性，具体数据：

乘法

乘法一阶导

乘法二阶导

除法

除法一阶导

除法二阶导