**Introduction to Scientific Computing, Homework #5**

**Problem 1 Solution**

% Symbolic Math Toolbox must be installed first

clear;clc;

%% Derivative

syms x y z n;

fxy=cos(x)\*sin(y)+y^3\*log(x);

disp(diff(fxy,'x'));

%% Divergence

Fxyz=[2\*(cos(x))^2 (sin(y))^3 2\*x^2+y^2+4\*z^2];

disp(divergence(Fxyz,[x,y,z]));

%% Gradient

fxyz=x^2+2\*y^2+4\*z^2;

disp(gradient(fxyz));

%% Integral

fx=x^2-5\*x;

disp(int(fx));

fx=x^(2-n)-5\*x;

disp(int(fx,x));

%% Triple Integration

fxyz=sin(x)\*cos(y)\*tan(z)+y\*cos(x)+x\*sin(y);

disp(simplify(int(int(int(fxyz,x),y),z))); %simplify for pretty answer

**Problem 1 Output(Answers)**

**y^3/x - sin(x)\*sin(y)**

**8\*z - 4\*cos(x)\*sin(x) + 3\*cos(y)\*sin(y)^2**

**2\*x**

**4\*y**

**8\*z**

**(x^2\*(2\*x - 15))/6**

**- (5\*x^2)/2 - x^3/(x^n\*(n - 3))**

**- log(tan(z) - 1i)\*((cos(x)\*sin(y))/2 + (y^2\*sin(x)\*1i)/4 - (x^2\*cos(y)\*1i)/4) - log(tan(z) + 1i)\*((x^2\*cos(y)\*1i)/4 - (y^2\*sin(x)\*1i)/4 + (cos(x)\*sin(y))/2)**

(Next Page)

(Next Page)

**Problem 2 Solution(Conclusion included in CODE)**

clear;clc;

%% Step1: ezplot for funciton image

p=ezplot('sin(x)+(cos(x^2-pi/7))^2',[-5\*pi,5\*pi]);

set(p, 'LineStyle', '-', 'Color', 'black');

xlabel('x');

ylabel('f(x)');

title('Plot for sin(x)+cos^2(x^2-\pi/7)');

%% Step2: using trapz() to estimate the integral

figure;

X = 5:1:500;

Y = zeros(1,496);

for div=5:1:500

x=linspace(-5\*pi,5\*pi,div);

y=sin(x)+(cos(x.^2.-pi/7)).^2;

Y(div-4)=trapz(x,y);

end

plot(X,Y, 'b-', 'LineWidth',1);

axis auto;

%% Step3: compares to accurate answer generated by integral()

accurate\_Y = integral(@(x)sin(x)+(cos(x.^2.-pi/7)).^2, -5\*pi, 5\*pi);

hold on;

plot(X, repelem(accurate\_Y, length(X)));

title('trapz compares to integral');

legend('trapz', 'integral');

xlabel('divisions');

ylabel('result');

%% Conclusion

% From the figure we can derive that integral() is much accurate for

% computing integral in such cases; Also we can find that about ***350***

% divisions are needed for a converge.

图片包含 游戏机, 刀

描述已自动生成**Problem 2 Plots**

(Next Page)

(Next Page)

**Problem 3 Solution(Answer included in CODE)**

clear;clc;

load poly.mat X Y;

[R2,N85,p]=discoverDegree(X,Y,0.85);

plot(X,Y, 'bo');

xlim([-10 10]);

ylim([-40 40]);

hold on;

plot(X,polyval(p,X),'r-','LineWidth',1.5); % Pretty plot

title('a fit for R^2\geq0.85'); % a title that fits for a fit

legend('Data', 'Polynomial Fit', 'Location','southeast');

% Required: legend

xlabel('x');ylabel('y'); % Required:label

[~,N95,~]=discoverDegree(X,Y,0.95);

disp([N85, N95]); % Generate answer for questions

%% Answer:

% For R^2≥0.85, 3 polynomial degree is needed;

% For R^2≥0.95, 5 polynomial degree is needed.

%%

function [R2out,pOrder,pCoeff] = discoverDegree(x,y,R2crit)

mean\_y = mean(y);

pOrder = 1;

pCoeff=polyfit(x,y,pOrder);

R2out = 1-sum((y-polyval(pCoeff,x)).^2)/sum((y-mean\_y).^2);

while R2out<R2crit

pOrder = pOrder+1;

pCoeff=polyfit(x,y,pOrder);

R2out = 1-sum((y-polyval(pCoeff,x)).^2)/sum((y-mean\_y).^2);

end

end

图表, 直方图, 散点图

描述已自动生成**Problem 3 Plot**

(Next Page)

**Problem 4 Solution**

clear;clc;

%% Initial conditions

g=9.8;

c=0.2;

x0=0;

y0=0;

low\_x=0;

high\_x=100;

vy=2.0;

time\_step=0.01;

dis=10;

tol=0.01;

%% Calculation

disp(goldCornhole(g,c,x0,y0,low\_x,high\_x,vy,time\_step,dis,tol));

function vxOpt = goldCornhole(g,c,x0,y0,vxa,vxb,vy,tstep,dis,tol)

%% Golden section search

vxOpt = inf;

vx1=vxb-0.618\*(vxb-vxa); % Test point 1

vx2=0.618\*(vxb-vxa)+vxa; % Test point 2

dis1 = abs(projectileSim(g,c,x0,y0,vx1,vy,tstep));

dis2 = abs(projectileSim(g,c,x0,y0,vx2,vy,tstep));

while abs(dis1-dis)>tol || abs(dis2-dis)>tol % Totally in the range

if abs(dis1-dis) < abs(dis2-dis)

% Choose the better, eliminate the worse

vxOpt=vx1;

vxb=vx2;

vx2=vx1;dis2=dis1;

vx1=vxb-0.618\*(vxb-vxa);

dis1 = abs(projectileSim(g,c,x0,y0,vx1,vy,tstep));

else

vxOpt=vx2;

vxa=vx1;

vx1=vx2;dis1=dis2;

vx2=0.618\*(vxb-vxa)+vxa;

dis2 = abs(projectileSim(g,c,x0,y0,vx2,vy,tstep));

end

end

end

function disX = projectileSim(g,c,x0,y0,vx0,vy0,tstep)

% Calculate landing point of the beanbag, where air drag considered.

EPS=1e-6;

disX = x0;

while y0>-EPS || vy0>0

x0 = vx0\*tstep + x0;

y0 = vy0\*tstep + y0;

vy0 = vy0 - g\*tstep - tstep\*sign(vy0)\*c\*vy0^2;

% be careful about the force direction

vx0 = vx0 - tstep\*sign(vx0)\*c\*vx0^2;

end

disX = x0 - disX;

end

**Problem 5 Solution (Code is too long, hence no box placed)**

clear;clc;

%% initial conditions

t=struct(); % create struct

x=struct();

dx=struct();

d2x=struct();

t.raw = [0, 1, 2.5, 5.0, 10.5, 12.5, 16, 20.5, 26.5, 30.5, 32];

x.raw = [0, 0.3, 1.2, 1.3, 1.6, 2.2, 2.4, 3.0, 3.6, 4.5, 4.6];

domain=linspace(0,32,100);

%% data generation

x.interp.linear = interp1(t.raw,x.raw,domain,'linear');

x.interp.spline = interp1(t.raw,x.raw,domain,'spline');

dx.interp.linear = diff(x.interp.linear)\*100/32;

dx.interp.linear(end+1)=dx.interp.linear(end);

% 1 element is dropped after diff, must add an extra value

dx.interp.spline = diff(x.interp.spline)\*100/32;

dx.interp.spline(end+1)=dx.interp.spline(end);

d2x.interp.linear = diff(dx.interp.linear)\*100/32;

d2x.interp.linear(end+1) = d2x.interp.linear(end);

d2x.interp.spline = diff(dx.interp.spline)\*100/32;

d2x.interp.spline(end+1) = d2x.interp.spline(end);

%% Linear

figure('units','normalized','outerposition',[0 0 1 1]);

% Full screen

set(gcf, 'Color', 'white');

ax1 = axes('Position', [.1 .69 .8 .23]);% Manually specific position

plot(domain,x.interp.linear, 'k-', 'LineWidth',1.5);

set(ax1, 'Xticklabel', [], 'LineWidth', 1, 'FontSize', 11); % Remove label

xlim(ax1, [0 32]);ylim(ax1, [0 6]);

ylabel(ax1, 'x (m)');

hold on;

plot(t.raw,x.raw, 'k.', 'MarkerSize',20); % Plot raw data as markers

hold off;

ax2 = axes('Position', [.1 .42 .8 .23]);

plot(domain,dx.interp.linear, 'k-', 'LineWidth',1.5);

set(ax2, 'Xticklabel', [], 'LineWidth', 1, 'FontSize', 11);

xlim(ax2, [0 32]);ylim(ax2, [0 1]);

ylabel(ax2, 'dx/dt (m/s)');

ax3 = axes('Position', [.1 .15 .8 .23]);

plot(domain,d2x.interp.linear, 'k-', 'LineWidth',1.5);

set(ax3, 'LineWidth', 1, 'FontSize', 11);

xlim(ax3, [0 32]);ylim(ax3, [-2 1]);

ylabel(ax3, 'd^2x/dt^2 (m/s^2)');

xlabel('Times (s)');

%% Spline

figure('units','normalized','outerposition',[0 0 1 1]);

set(gcf, 'Color', 'white');

ax1 = axes('Position', [.1 .69 .8 .23]);

plot(domain,x.interp.spline, 'k-', 'LineWidth',1.5);

set(ax1, 'Xticklabel', [], 'LineWidth', 1, 'FontSize', 11);

xlim(ax1, [0 32]);ylim(ax1, [0 6]);

ylabel(ax1, 'x (m)');

hold on;

plot(t.raw,x.raw, 'k.', 'MarkerSize',20);

hold off;

ax2 = axes('Position', [.1 .42 .8 .23]);

plot(domain,dx.interp.spline, 'k-', 'LineWidth',1.5);

set(ax2, 'Xticklabel', [], 'LineWidth', 1, 'FontSize', 11);

xlim(ax2, [0 32]);ylim(ax2, [-1 1]);

ylabel(ax2, 'dx/dt (m/s)');

ax3 = axes('Position', [.1 .15 .8 .23]);

plot(domain,d2x.interp.spline, 'k-', 'LineWidth',1.5);

set(ax3, 'LineWidth', 1, 'FontSize', 11);

xlim(ax3, [0 32]);ylim(ax3, [-2 1]);

ylabel(ax3, 'd^2x/dt^2 (m/s^2)');

xlabel('Times (s)');

(Next Page)

**Problem 5 Plots(First:Linear Second:Spline)**

图表, 折线图

描述已自动生成图表

描述已自动生成

(The End)