MTE 203 - Advanced Calculus

Project 1

Extrema of Functions

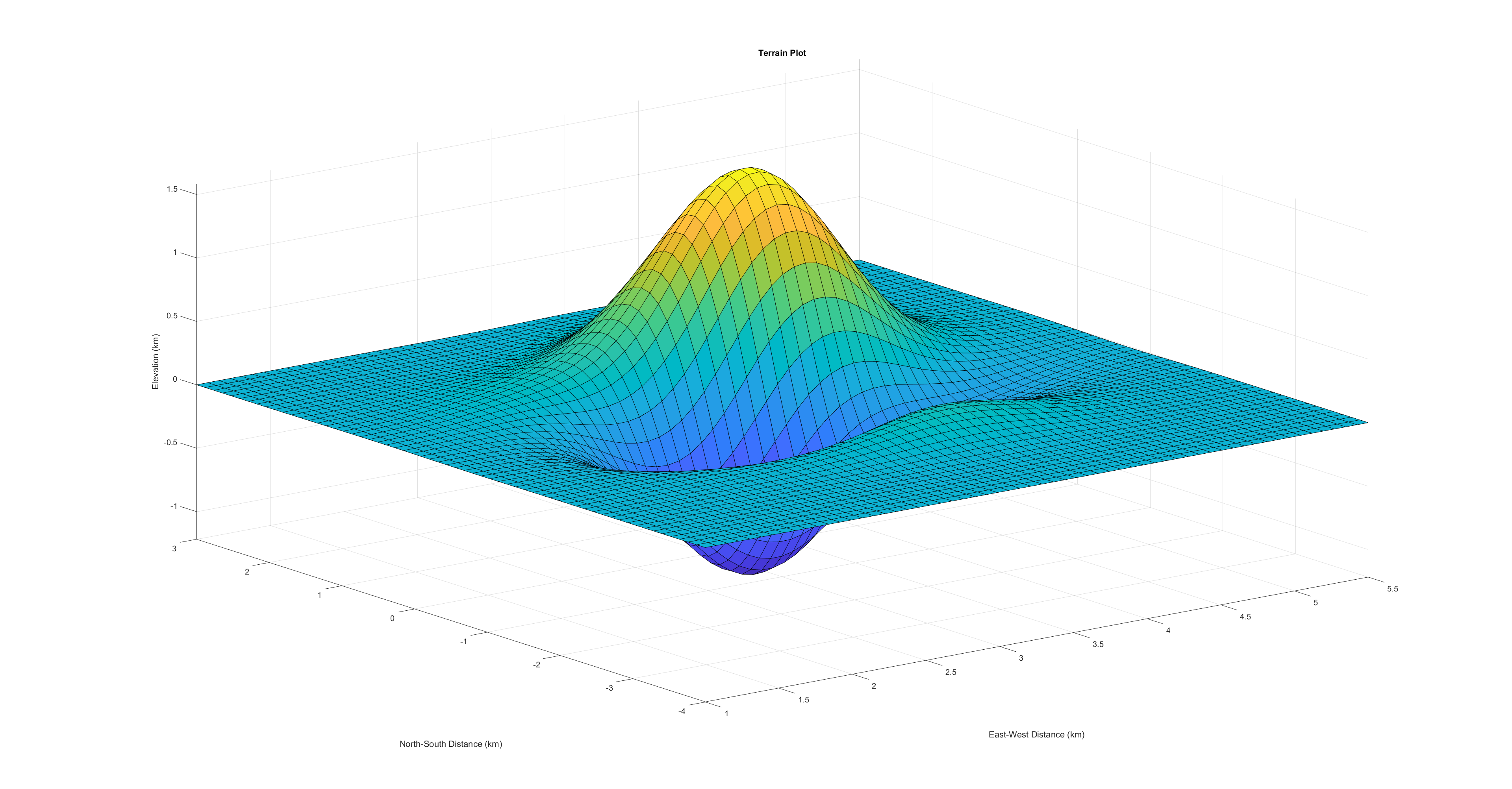
Name: DEREK YU XING LIN

Student ID: 20724532

November 11, 2019

**Part 1:**

1. **a)** The MATLAB surface and contour plotting functions to generate terrain plot and contours. An x-y meshgrid was initialized with (100 x 100) = 10,000 points.





1. **b)** The steepest slope can be determined by seeing which region has the greatest contour line density. If there are more counter curves within a certain region, then it has a steeper slope.   
   Mathematically, the point with the steepest slope can be found by taking the gradient of the magnitude of the gradient of the function, and setting it equal to zero, then solving for x, and y. The point with the largest slope is at (3.2436, -0.0961).

The solution was found with the MATLAB fsolve function, and an initial guess of (3,0).

1. **c)** Highest and lowest elevations can be found by finding the critical points of the function. The critical points are the x and y values when the partial derivatives of the function are equal zero. The second derivative test was used to determine whether critical points were saddle points, local maxima, or local minima. To determine absolute highest and lowest elevation, the elevation was calculated for each critical point and then directly compared.

The highest elevation is 1.5883 km at (3.5551, 0.6003). Lowest elevation is -1.2236 km at (2.9191, -0.7505).

1. **d)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Point** | **(x, y)** | **f(x,y)** | **A** | **B** | **C** | **D=B2-AC** | **Point Type** |
| 1 | (3.5551, 0.6003) | 1.5883 | -5.8770 | -0.5400 | -4.2212 | -24.5164 | Local Max |
| 2 | (2.9191, -0.7505) | -1.2236 | 4.4892 | 0.7479 | 4.8362 | -21.1517 | Local Min |
| 3 | (3.5963, -2.0459) | 0.1796 | -0.7658 | 0.2414 | -1.4006 | -1.0143 | Local Max |

A function was created that accepted critical point x-y coordinates as inputs, and then determined the type of point using the second derivative test, and the following rules:

The points are acceptable as they correspond to local maxima and local minima that are clearly visible in the surface plot. The points are also all within the ranges specified.

**PART 2:**

1. **a)** The temperature at the highest and lowest elevations correspond to:
2. **b)**

The temperature that the hiker feels at the point (4, -0.3) is 13.6987 °C.

Isotherms at the elevation of the hiker were found by plotting the contour plot with inputs x, y, and 0.1675, which is the elevation of the hiker.

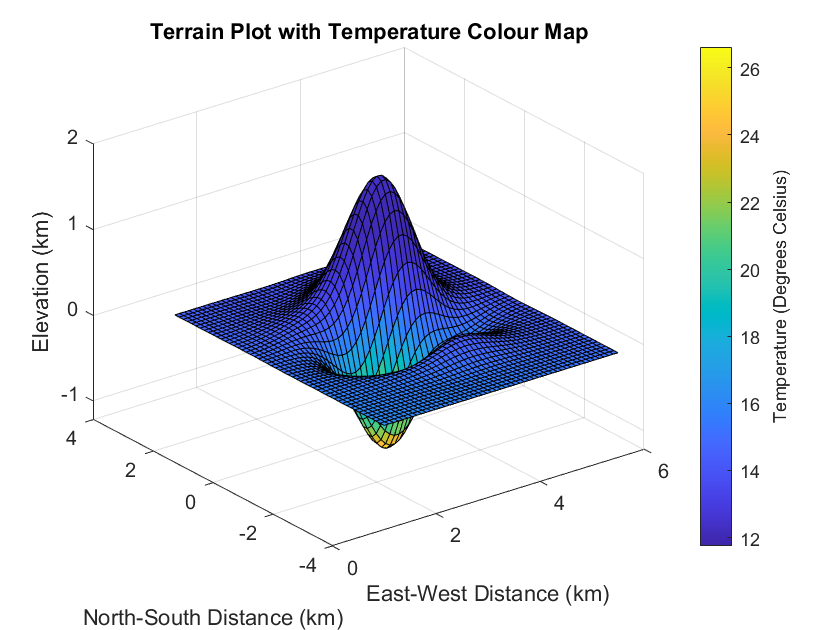


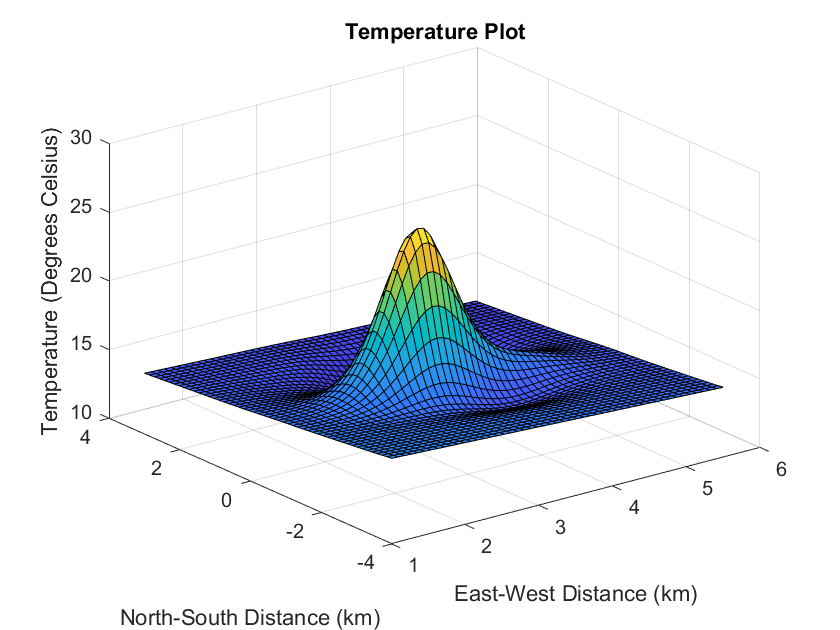
1. **c)** The directional gradient of the elevation can be determined by dotting the 2-D unit vector of the direction with the gradient function of the surface at the point (4, -0.3).

The hiker is ascending when he hikes in the northwest direction because the gradient is positive, indicating an increase in elevation. When the hiker hikes in the southwest direction, they are descending because the gradient is negative, indicating a decrease in elevation.

The directional gradient of the temperature can be determined by dotting the 3-D unit vector of the direction along the tangent surface of the mountain with the gradient function of the temperature at the point (4, -0.3, 0.1675). Z-values for the unit vector directions were calculated by determining the elevation change when the hiker moves in the direction specified.

When the hiker walks in the northwest direction, they will experience a temperature rate of change of . In the southwest direction, they will experience a temperature rate of change of .

1. **d)** This plot is useful to a hiker if they want to avoid hotter or colder regions of the mountain. With this plot, they can see how the temperature ranges with mountain surface. 
2. **e)** This plot is useful to a hiker if they only care about temperature and do not care about elevation.

. ****

1. **f)** Using Lagrange multipliers, the point with the highest temperature is at (2.9149, -0.7547, -1.2235), with a lambda value of 0.

Finally, use MATLAB fsolve with initial guess (3, -0.8, -1.2, 1).

**SUMMARY OF RESULTS**  
  
**Equations and Values used for Q1:**

|  |  |
| --- | --- |
| z(x, y) | 0.00125 \* exp(-((0.5\*y^2)+(x-3)^2))\*(sin(2\*x)+2\*sin(0.75\*(0.5\*y-2)^2))\*(16\*x+64\*x^2 + y^2) |
|  | (cos(2\*x)\*exp(- (x - 3)^2 - y^2/2)\*(64\*x^2 + 16\*x + y^2))/400 + (exp(- (x - 3)^2 - y^2/2)\*(128\*x + 16)\*(sin(2\*x) + 2\*sin((3\*(y/2 - 2)^2)/4)))/800 - (exp(- (x - 3)^2 - y^2/2)\*(2\*x - 6)\*(sin(2\*x) + 2\*sin((3\*(y/2 - 2)^2)/4))\*(64\*x^2 + 16\*x + y^2))/800 |
|  | (y\*exp(- (x - 3)^2 - y^2/2)\*(sin(2\*x) + 2\*sin((3\*(y/2 - 2)^2)/4)))/400 - (y\*exp(- (x - 3)^2 - y^2/2)\*(sin(2\*x) + 2\*sin((3\*(y/2 - 2)^2)/4))\*(64\*x^2 + 16\*x + y^2))/800 + (cos((3\*(y/2 - 2)^2)/4)\*exp(- (x - 3)^2 - y^2/2)\*((3\*y)/8 - 3/2)\*(64\*x^2 + 16\*x + y^2))/400 |
| Gradient Magnitude |  |
| Gradient of Magnitude of Gradient |  |
| A |  |
| B |  |
| C |  |
| D |  |
| Highest Elevation | 1.5883 at (3.5551, 0.6003) |
| Lowest Elevation | -1.2236 km at (2.9191, -0.7505) |
| Critical Points | (3.5551, 0.6003) (2.9191, -0.7505) (3.5963, -2.0459) |
| Temperature at Highest Point |  |
| Temperature at Lowest Point |  |
| Height at point (4,-0.3) |  |
| Temperature at (4, -0.3) |  |
|  | -(17\*exp((z - 1)^2/10 - x/100 + (y/20 - 1)^2/10 + 1/5))/100 |
|  | 17\*exp((z - 1)^2/10 - x/100 + (y/20 - 1)^2/10 + 1/5)\*(y/2000 - 1/100) |
|  | 17\*exp((zz - 1)^2/10 - x/100 + (y/20 - 1)^2/10 + 1/5)\*(zz/5 - 1/5) - zz/5 |
| T(x,y) |  |
| Lagrange Equation |  |
| Location of highest temperature | (2.9149, -0.7547, -1.2235) |

**APPENDIX**

clear all;

clc;

%1A

[x, y] = meshgrid(1:0.045:5.5, -4:0.07:3);

z = 0.00125 \* exp(-((0.5\*y.^2)+(x-3).^2)).\*(sin(2\*x)+2\*sin(0.75\*(0.5\*y-2).^2)).\*(16\*x+64\*x.^2 + y.^2);

figure(1)

surf(x,y,z)

xlabel('East-West Distance (km)')

ylabel('North-South Distance (km)')

zlabel('Elevation (km)')

title('Terrain Plot')

figure(2)

[c, h]=contour(x, y, z, 25);

legend = colorbar;

clabel(c,h, 'FontSize', 6);

xlabel('East-West Distance (km)')

ylabel('North-South Distance (km)')

ylabel(legend, 'Elevation (km)')

title('Contour Plot of Terrain')

syms x y z T zz l zzz

%1B

z(x,y)= 0.00125 \* exp(-((0.5\*y^2)+(x-3)^2))\*(sin(2\*x)+2\*sin(0.75\*(0.5\*y-2)^2))\*(16\*x+64\*x^2 + y^2);

dzdx(x,y) = diff(z, x)

dzdy(x,y) = diff(z, y)

slope = sqrt(dzdx.^2 + dzdy.^2);

gradientSlope = gradient(slope);

gradientSlopeMatlab = matlabFunction(gradientSlope, 'Vars', {[x y]});

guess = [3,0];

pointMaxSlope = fsolve(gradientSlopeMatlab, guess);

%1C

gradientZ = gradient(z);

gradientZMatlab = matlabFunction(gradientZ, 'Vars', {[x y]});

guess1 = [3.5,1];

guess2 = [3,-1];

guess3 = [3.5,-2];

cp1=fsolve(gradientZMatlab, guess1);

cp2=fsolve(gradientZMatlab, guess2);

cp3=fsolve(gradientZMatlab, guess3);

A = diff(dzdx, x);

B = diff(dzdx, y);

C = diff(dzdy, y);

Astatus = deriv2Test(A, B, C, cp1, x, y);

Bstatus = deriv2Test(A, B, C, cp2, x, y);

Cstatus = deriv2Test(A, B, C, cp3, x, y);

maxHeight = double(z(cp1(1), cp1(2)));

minHeight = double(z(cp2(1), cp2(2)));

localMax = double(z(cp3(1), cp3(2)));

%2A

T(x,y,zz) = (-0.1\*(zz^2))+17\*exp(-0.1\*((0.1\*x-2)-(0.05\*y-1)^2-(zz-1)^2))-10;

tempAtPeak = double(T(cp1(1), cp1(2), maxHeight));

tempAtValley = double(T(cp2(1), cp2(2), minHeight));

%2B

heightAtPoint = z(4,-0.3);

tempAtPoint = T(4,-0.3,heightAtPoint);

[xx, yy] = meshgrid(1:0.09:5.5,-4:0.14:3);

isotherm = (-0.1\*(heightAtPoint^2))+17\*exp(-0.1\*((0.1\*xx-2)-(0.05\*yy-1).^2-(heightAtPoint-1)^2))-10;

figure(3)

[c, h] = contour(xx,yy,isotherm,30);

axis([1 5.5 -4 3]);

legend = colorbar;

clabel(c,h);

xlabel('East-West Distance (km)');

ylabel('North-South Distance (km)');

ylabel(legend, 'Temperature (Degrees Celsius)');

title('Isotherms at Same Elevation as Point(4,-0.3)');

%2C

NWUnit2D = [-1/sqrt(2), 1/sqrt(2)];

SWUnit2D = [-1/sqrt(2), -1/sqrt(2)];

gradientAtPoint = [dzdx(4,-0.3), dzdy(4,-0.3)];

heightChangeNW = double(dot(NWUnit2D, gradientAtPoint));

heightChangeSW = double(dot(SWUnit2D, gradientAtPoint));

NWZ = z(3,0.7)-0.1675;

SWZ = z(3,-1.3)-0.1675;

NWUnit3D = [-1, 1,NWZ]/sqrt(1+1+NWZ^2);

SWUnit3D = [-1, -1,SWZ]/sqrt(1+1+SWZ^2);

gradientT = gradient(T)

gradientTAtPoint = gradientT(4,-0.3,heightAtPoint);

tempChangeNW = double(dot(gradientTAtPoint, NWUnit3D));

tempChangeSW = double(dot(gradientTAtPoint, SWUnit3D));

%2D

zz = 0.00125 \* exp(-((0.5\*yy.^2)+(xx-3).^2)).\*(sin(2\*xx)+2\*sin(0.75\*(0.5\*yy-2).^2)).\*(16\*xx+64\*xx.^2 + yy.^2);

ColorT = (-0.1\*(zz.^2))+17\*exp(-0.1\*((0.1\*xx-2)-(0.05\*yy-1).^2-(zz-1).^2))-10;

figure(4)

surf(xx,yy,zz,ColorT);

legend=colorbar;

xlabel('East-West Distance (km)');

ylabel('North-South Distance (km)');

ylabel(legend, 'Temperature (Degrees Celsius)');

zlabel('Elevation (km)');

title('Terrain Plot with Temperature Colour Map');

%2E

figure(5)

surf(xx,yy,ColorT)

xlabel('East-West Distance (km)');

ylabel('North-South Distance (km)');

zlabel('Temperature (Degrees Celsius)');

title('Temperature Plot');

%2F

C=0.00125.\*exp(-((x-3).^2+0.5.\*y.^2)).\*(sin(2.\*x)+2.\*sin(0.75\*(0.5\*y-2).^2)).\*(16.\*x+64.\*x.^2+y.^2)-zzz;

T(x,y) = 0.1.\*(0.00125.\*exp(-((x-3).^2+0.5.\*y.^2)).\*(sin(2.\*x)+2.\*sin(0.75\*(0.5\*y-2).^2)).\*(16.\*x+64.\*x.^2+y.^2)).^2+17\*exp(-0.1.\*((0.1.\*x-2)-(0.05.\*y-1).^2-((0.00125.\*exp(-((x-3).^2+0.5.\*y.^2)).\*(sin(2.\*x)+2.\*sin(0.75\*(0.5\*y-2).^2)).\*(16.\*x+64.\*x.^2+y.^2))-1).^2))-10;

Lagrange = T - l\*C;

dLdx = diff(Lagrange, x);

dLdy = diff(Lagrange, y);

dLdz = diff(Lagrange, zzz);

dLdl = diff(Lagrange, l);

lagVF = [dLdx, dLdy, dLdz, dLdl];

vars= [x y zzz l];

LagGuess = [3, -0.8, -1.2, 1];

lagVFmatlab = matlabFunction(lagVF,'Vars',{vars});

pointMaxTemp = fsolve(lagVFmatlab, LagGuess);

function [determination]= deriv2Test(A, B, C, cp, x, y)

A1 = double(subs(A, [x y], cp));

B1 = double(subs(B, [x y], cp));

C1 = double(subs(C, [x y], cp));

D = (B1).^2 - (A1).\*(C1);

if D<0

if A1<0

determination = 'Local Max';

elseif A1>0

determination = 'Local Min';

end

elseif D>0

determination = 'Saddlepoint';

else

determination = 'No conclusion';

end

end