x = linspace(-2, 3.5);

y = linspace(-2, 3.5);

[X, Y] = meshgrid(x, y);

%plot\_f1\_2d(1, 0, X, Y, x);

%plot\_f2\_2d(1, 0, X, Y, x);

%plot\_f3\_2d(1, 0, X, Y, x);

%plot\_f1\_3d(2.5, -1, X, Y, x);

%plot\_f2\_3d(1, 0, X, Y, x);

plot\_f3\_3d(2.5, -1, X, Y, x);

function plot\_f1\_2d(x01, x02, X, Y, x)

f1 = 2\*X + 3\*Y + 1;

norm = sqrt(2^2 + 3^2);

% plot gradient vec

gradient\_vec = [2, 3]/norm

quiver(x01, x02, gradient\_vec(1), gradient\_vec(2))

hold on

% plot tangent line

y\_t = -2/3\*(x-x01) + x02;

plot(x, y\_t);

hold on

% plot contour

contour(X, Y, f1)

hold on

axis equal

axis([-2, 3.5, -2, 3.5])

title(['f1-2d x=' num2str(x01) ', y=' num2str(x02)])

end

function plot\_f2\_2d(x01, x02, X, Y, x)

f2 = X.\*X + Y.\*Y - X.\*Y - 5;

norm = sqrt((2\*x01-x02)^2 + (2\*x02-x01)^2);

% plot gradient vec

gradient\_vec = [2\*x01-x02, 2\*x02-x01]/norm

quiver(x01, x02, gradient\_vec(1), gradient\_vec(2))

hold on

% plot tangent line

y\_t = -1 \* (2\*x01-x02) / (2\*x02-x01) \* (x-x01) + x02

plot(x, y\_t);

hold on

% plot contour

contour(X, Y, f2)

hold on

axis equal

axis([-2, 3.5, -2, 3.5])

title(['f2-2d x=' num2str(x01) ', y=' num2str(x02)])

end

function plot\_f3\_2d(x01, x02, X, Y, x)

f3 = (X-5).\*cos(Y-5) - (Y-5).\*sin(X-5);

norm = sqrt((cos(x02-5)-(x02-5)\*cos(x01-5))^2 + (-1\*(x01-5)\*sin(x02-5)-sin(x01-5))^2);

% plot gradient vec

gradient\_vec = [cos(x02-5)-(x02-5)\*cos(x01-5), -1\*(x01-5)\*sin(x02-5)-sin(x01-5)]/norm

quiver(x01, x02, gradient\_vec(1), gradient\_vec(2))

hold on

% plot tangent line

y\_t = -1 \* (cos(x02-5)-(x02-5)\*cos(x01-5)) / (-1\*(x01-5)\*sin(x02-5)-sin(x01-5)) \* (x-x01) + x02

plot(x, y\_t);

hold on

% plot contour

contour(X, Y, f3)

hold on

axis equal

axis([-2, 3.5, -2, 3.5])

title(['f2-2d x=' num2str(x01) ', y=' num2str(x02)])

end

function plot\_f1\_3d(x01, x02, X, Y, x)

f1 = 2\*X + 3\*Y + 1;

mesh(X, Y, f1)

hold on

norm = sqrt(2^2 + 3^2);

% plot gradient vec

gradient\_vec = [2, 3]/norm

quiver(x01, x02, gradient\_vec(1), gradient\_vec(2))

hold on

% plot tangent line

y\_t = -2/3\*(x-x01) + x02;

plot(x, y\_t);

hold on

% quadratic approx

approx = 2\*x01 + 3\*x02 + 1 + 2\*(X-x01) + 3\*(Y-x02)

mesh(X, Y, approx)

title(['f1-3d x=' num2str(x01) ', y=' num2str(x02)])

end

function plot\_f2\_3d(x01, x02, X, Y, x)

f2 = X.\*X + Y.\*Y - X.\*Y - 5;

mesh(X, Y, f2)

hold on

norm = sqrt((2\*x01-x02)^2 + (2\*x02-x01)^2);

% plot gradient vec

gradient\_vec = [2\*x01-x02, 2\*x02-x01]/norm

quiver(x01, x02, gradient\_vec(1), gradient\_vec(2))

hold on

% plot tangent line

y\_t = -1 \* ((2\*x01-x02) / (2\*x02-x01)) \* (x-x01) + x02

plot(x, y\_t);

hold on

% quadratic approx

hessian = [2, -1; -1, 2]

approx = x01^2+x02^2-x01\*x02-5 + (2\*x01-x02)\*(X-x01) + (2\*x02-x01)\*(Y-x02);

for i = 1:100

for j = 1:100

quadractic\_term = 1/2 \* [X(i, j)-x01, Y(i,j)-x02]\*hessian\*[(X(i, j)-x01); (Y(i,j)-x02)];

approx(i, j) = approx(i, j) + quadractic\_term;

end

end

mesh(X, Y, approx)

title(['f2-2d x=' num2str(x01) ', y=' num2str(x02)])

end

function plot\_f3\_3d(x01, x02, X, Y, x)

f3 = (X-5).\*cos(Y-5) - (Y-5).\*sin(X-5);

mesh(X, Y, f3)

hold on

norm = sqrt((cos(x02-5)-(x02-5)\*cos(x01-5))^2 + (-1\*(x01-5)\*sin(x02-5)-sin(x01-5))^2);

% plot gradient vec

gradient\_vec = [cos(x02-5)-(x02-5)\*cos(x01-5), -1\*(x01-5)\*sin(x02-5)-sin(x01-5)]/norm

quiver(x01, x02, gradient\_vec(1), gradient\_vec(2))

hold on

% plot tangent line

y\_t = -1 \* (cos(x02-5)-(x02-5)\*cos(x01-5)) / (-1\*(x01-5)\*sin(x02-5)-sin(x01-5)) \* (x-x01) + x02

plot(x, y\_t);

hold on

% quadratic approx

hessian = [(x02-5)\*sin(x01-5), -sin(x02-5)-cos(x01-5); -1\*sin(x02-5)-cos(x01-5), -1\*(x01-5)\*cos(x02-5)]

approx = (x01-5)\*cos(x02-5) - (x02-5)\*sin(x01-5) + (cos(x02-5)-(x02-5)\*cos(x01-5))\*(X-x01) + -1\*(x01-5)\*sin(x02-5)-sin(x01-5)\*(Y-x02);

for i = 1:100

for j = 1:100

quadractic\_term = 1/2 \* [X(i, j)-x01, Y(i,j)-x02]\*hessian\*[(X(i, j)-x01); (Y(i,j)-x02)];

approx(i, j) = approx(i, j) + quadractic\_term;

end

end

mesh(X, Y, approx)

title(['f2-2d x=' num2str(x01) ', y=' num2str(x02)])

end