ECE 133A — Homework #8 — Disc 1B — Angel Jimenez — 604 667 905

18.8 *Fitting an ellipse to points in a plane.* An ellipse in a plane can be described as the set of points

$$\hat{f}(t;\theta) = \begin{bmatrix} c_1 + r\cos(\alpha+t) + \delta\cos(\alpha-t) \\ c_2 + r\sin(\alpha+t) + \delta\sin(\alpha-t) \end{bmatrix}$$

where t ranges from 0 to 2π. The vector $\theta = (c_1, c_2, r, \delta, \alpha)$ contains five parameters, with geometrical meanings illustrated in figure 18.27. We consider the problem of fitting an ellipse to N points $x^{(1)}, \ldots, x^{(N)}$ in a plane, as shown in figure 18.28. The circles show the N points. The short lines connect each point to the nearest point on the ellipse. We will fit the ellipse by minimizing the sum of the squared distances of the N points to the ellipse.

(a) The squared distance of a data point $x^{(i)}$ to the ellipse is the minimum of $\|\hat{f}(t^{(i)};\theta) - x^{(i)}\|^2$ over the scalar $t^{(i)}$. Minimizing the sum of the squared distances of the data points $x^{(1)}, \ldots, x^{(N)}$ to the ellipse is therefore equivalent to minimizing

$$\sum_{i=1}^{N} \|\hat{f}(t^{(i)};\theta) - x^{(i)}\|^2$$

over $t^{(1)}, \ldots, t^{(N)}$ and θ. Formulate this as a nonlinear least squares problem. Give expressions for the derivatives of the residuals.

(b) Use the Levenberg–Marquardt algorithm to fit an ellipse to the 10 points:

$(0.5, 1.5)$, $(-0.3, 0.6)$, $(1.0, 1.8)$, $(-0.4, 0.2)$, $(0.2, 1.3)$

$(0.7, 0.1)$, $(2.3, 0.8)$, $(1.4, 0.5)$, $(0.0, 0.2)$, $(2.4, 1.7)$.

To select a starting point, you can choose parameters θ that describe a circle with radius one and center at the average of the data points, and initial values of $t^{(i)}$ that minimize the objective function for those initial values of θ.

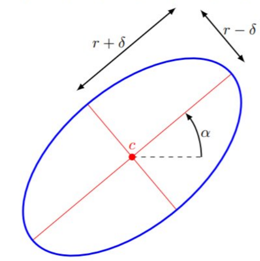


Figure 18.27 Ellipse with center (c_1, c_2), and radii $r+\delta$ and $r-\delta$. The largest semi-axis makes an angle α with respect to horizontal.

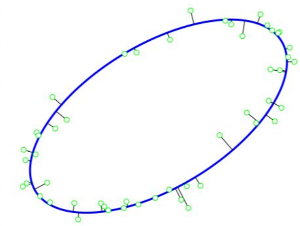


Figure 18.28 Ellipse fit to 50 points in a plane.

$-c \quad s \quad c \quad -s$

$$\hat{f}(t;\theta) = \begin{bmatrix} c_1 + r\cos(\alpha+t) + \delta\cos(\alpha-t) \\ c_2 + r\sin(\alpha+t) + \delta\sin(\alpha-t) \end{bmatrix}$$

$$\frac{\partial \hat{f}}{\partial c_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \frac{\partial \hat{f}}{\partial c_2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \frac{\partial \hat{f}}{\partial r} = \begin{bmatrix} \cos(\alpha+t) \\ \sin(\alpha+t) \end{bmatrix}, \quad \frac{\partial \hat{f}}{\partial \delta} = \begin{bmatrix} \cos(\alpha-t) \\ \sin(\alpha-t) \end{bmatrix}, \quad \frac{\partial \hat{f}}{\partial \alpha} = \begin{bmatrix} -r\sin(\alpha+t) - \delta\sin(\alpha-t) \\ r\cos(\alpha+t) + \delta\cos(\alpha-t) \end{bmatrix} \quad \frac{\partial f}{\partial t} = \begin{bmatrix} -r\sin(\alpha+t) + \delta\sin(\alpha-t) \\ r\cos(\alpha+t) - \delta\cos(\alpha-t) \end{bmatrix}$$

$$\therefore Df(x^{(k)}) = \begin{bmatrix} \frac{\partial f}{\partial c_1} & \frac{\partial f}{\partial c_2} & \frac{\partial f}{\partial r} & \frac{\partial f}{\partial \delta} & \frac{\partial f}{\partial \alpha} & \frac{\delta f}{\partial t} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & \cos(\alpha+t) & \cos(\alpha-t) & -r\sin(\alpha+t) - \delta\sin(\alpha-t) & -r\sin(\alpha+t) + \delta\sin(\alpha-t) \\ 0 & 1 & \sin(\alpha+t) & \sin(\alpha-t) & r\cos(\alpha+t) + \delta\cos(\alpha-t) & r\cos(\alpha+t) - \delta\cos(\alpha-t) \end{bmatrix}$$

$$\hat{x} = x^{(k)} - (Df^T Df + \lambda^{(k)} I)^{-1} Df^T \cdot f$$

$Df(x^{(k)})$ is 20×15 matrix

diagonals ?

minimize $\|\hat{f}(x; x^{(k)})\|^2 + Df(x^{(k)})(x - x^{(k)})$

$\|\hat{f}(t; \theta^{(k)}) - \theta^{(k)}\|$

$$\left\| \begin{bmatrix} Df(x^{(k)}) \\ \sqrt{\lambda^{(k)}} I \end{bmatrix} x - \begin{bmatrix} Df(x^{(k)}) x^{(k)} - f(x^{(k)}) \\ \sqrt{\lambda^{(k)}} x^{(k)} \end{bmatrix} \right\|^2$$

I couldn't finish this problem.

the algorithm is on page 392 of the book

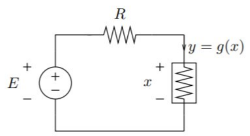
but I had a final and a project due monday/tuesday

So I couldn't fix this

12.1 The nonlinear resistive circuit shown below is described by the nonlinear equation

$$f(x) = g(x) - (E - x)/R = 0.$$

The function $g(x)$ gives the current through the nonlinear resistor as a function of the voltage x across its terminals.



Slide 14-3

Use Newton's method to find all the solutions of this nonlinear equation, assuming that $g(x) = x^3 - 6x^2 + 10x$. Consider three cases:

(a) $E = 5$, $R = 1$.

(b) $E = 15$, $R = 3$.

(c) $E = 4$, $R = 0.5$.

Select suitable starting points by plotting f over the interval $[0, 4]$, and visually selecting a good starting point. You can terminate the Newton iteration when $|f(x^{(k)})| < 10^{-8}$. Compare the speed of convergence for the three problems, by plotting $|f(x^{(k)})|$ versus k, or by printing out $|f(x^{(k)})|$ at each iteration.

$f(x) = g(x) - \frac{(E-x)}{R} = 0$ $\quad y = g(x) \rightarrow g(x) = x^3 - 6x^2 + 10x$

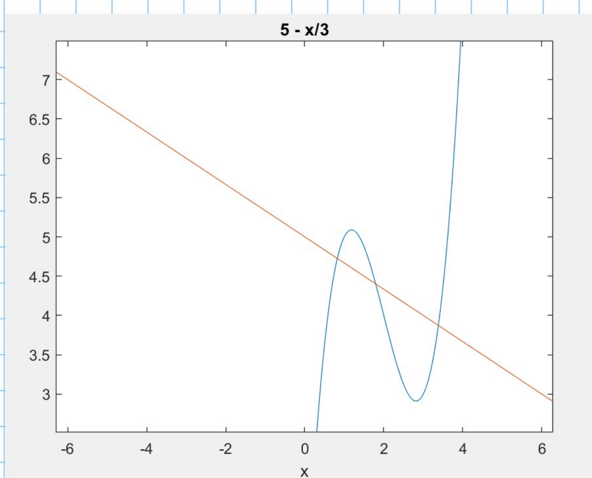
$\hookrightarrow f(x) = x^3 - 6x^2 + 10x - \frac{(E-x)}{R} = x^3 - 6x^2 + 10x + \frac{x}{R} - \frac{E}{R}$

From additional lecture notes,

Compute $f(x)$ and $f'(x)$

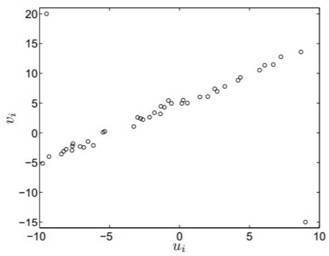
$\therefore f'(x) = 3x^2 - 12x + 10 + \frac{1}{R}$, $\epsilon = 10^{-8}$

$x^+ = x - \frac{f(x)}{f'(x)} = x - \frac{x^3 - 6x^2 + 10x + x/R - E/R}{3x^2 - 12x + 10 + 1/R}$




```
% Homework 8, Problem 2
clear all;
syms x;
E = 15;
R = 3;
%f = x - (x^3 - 6*x^2 + 10*x - (E-x)/R)/(3*x^2 - 12*x + 10 + 1/R);
g = x^3 - 6*x^2 +10*x;
%fo = g -(E-x)/R;
%xhat = x - (x^3-6*x^2+10*x + x/R - E/R)/(3*x^2 - 12*x +10 +1/R);
go = (E-x)/R;
ezplot(g);
hold on;
ezplot(go);
```

13.2 The figure shows 42 data points (u_i, v_i). The points are approximately on a straight line, except for the first and the last point.



The data are available in the file **robappr.m** as `[u,v] = robappr;`.

(a) Fit a straight line $v = \alpha + \beta u$ to the points by solving a least squares problem

$$\text{minimize} \quad g(\alpha, \beta) = \sum_{i=1}^{42} (\alpha + \beta u_i - v_i)^2,$$

with variables α, β.

(b) Fit a straight line $v = \alpha + \beta u$ to the points by solving the unconstrained minimization problem

$$\text{minimize} \quad g(\alpha, \beta) = \sum_{i=1}^{42} \sqrt{(\alpha + \beta u_i - v_i)^2 + 25}$$

using Newton's method. Use as initial points the values of α and β computed in part (a). (With this starting point, no line search should be necessary; for other starting points, a line search may be needed.) Terminate the iteration when $\|\nabla g(\alpha, \beta)\| \le 10^{-6}$.

a) $v = \alpha + \beta u$

$b = Ax$

minimize

$$g(\alpha,\beta) = \sum_{i=1}^{42} (\alpha + \beta u_i - v_i)^2 = \|Ax - b\|^2$$

$$\therefore b = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{42} \end{bmatrix} \quad A = \begin{bmatrix} 1 & u_1 \\ 1 & u_2 \\ \vdots & \vdots \\ 1 & u_{42} \end{bmatrix} \quad x = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

$x = A \backslash b.$

```
% HW8Pro3
clear all;
[u, v] = robappr;
m = length(u); %rows for A
A = [ones(m,1), u]; %Block matrix A [1_i u]
x = A\v;
disp(x);
```

```
x =

    4.1841
    0.5755
```

b) minimize $\quad g(\alpha,\beta) = \sum_{i=1}^{42} \sqrt{(\alpha + \beta u_i - v_i)^2 + 25}$ from discussion (14-23)

$g(x) = h(Cx + d)$

$\nabla g(x) = C^T \nabla h(Cx + d)$ $\qquad h(z) = \sum_{i=1}^{42} \sqrt{z_i^2 + 25}$ $\qquad \dfrac{\partial h}{\partial z_i} = \dfrac{2 z_i}{2(z_i^2 + 25)^{1/2}} = \dfrac{z_i}{(z_i^2 + 25)^{1/2}}$

$\dfrac{\partial h}{\partial z_i \partial z_j} \leftarrow$ 2 cases, $i = j$ and $i \neq j$

$$\therefore \frac{\partial h}{\partial z_i \partial z_j} = \begin{cases} \dfrac{25}{(z_i^2 + 25)^{3/2}} & i = j \\ 0 & i \neq j \quad \text{(all constant)} \end{cases}$$

$\therefore \nabla^2 h$ is a diagonal matrix

```
>> HW8Pro3
    4.1841
    0.5755

    4.8164
    0.8975
```

```
% HW8Pro3
%part a
clear all;
[u, v] = robappr;
m = length(u); %rows for A
A = [ones(m,1), u]; %Block matrix A [1_i u]
x = A\v; %had to change variables
disp(x);
%part b
xls = x;
for i = 1:50
    z = A*xls - v;
    tot = sum(sqrt(z.^2+25));
    grad = A'*(z./sqrt(z.^2+25));
    if(norm(grad) < 1e-6)
        break;
    end;
    hess= A'*diag((25)./(z.^2+25).^(3/2))*A;
    xls = xls - hess \ grad;
end;
disp(xls);
```