

```

% gda09_10
%
% Newton's Method solution to exemplary 1D non-linear
% inverse problem,  $d(z) = \sin(k*m*z)*z$ 
% This version uses a starting guess for the model
% parameter m1 that leads to convergence to a local
% minimum of the error surface (and hence to an incorrect
% value of the model parameter).
% Supports Figure 9.8

clear all;

% auxiliary parameter z
N = 21;
zmin = 0;
zmax = 5.0;
Dz = (zmax-zmin)/(N-1);
z = zmin + Dz*[0:N-1]';
zp = z.^0.5;

% define 1D grid for model parameter m1
Mg = 501;
mmin = 0;
mmax = 5;
Dm = (mmax-mmin)/(Mg-1);
m = mmin + Dm*[0:Mg-1];

% only one model parameter, m1
M=1;

% true model parameter
mtrue=3;

% true data
k=1;
dtrue = sin(k*mtrue.*zp).*z;

% observed data is true data plus random noise
sd=2;
dobs=dtrue+random('Normal',0,sd,N,1);

% tabulate error E(m) on the grid
E1=zeros(Mg,1);
for i=1:Mg
    dpre = sin(k*m(i).*zp).*z;
    e = dobs - dpre;
    E1(i) = (e'*e);
end

% find global minimum
[Elmin, iElmin] = min(E1);
mlest=m(iElmin);

% Newton's method
% derivative
%  $d = \sin(k*m(i).*zp).*z$ ;
%  $dd/dm = k.*zp.*z.*\cos(k*m(i).*zp)$ ;
%  $d = d(m_0) + (dd/dm)|_{m_0} (m-m_0)$ 

```

```

% tabulate a quadratic version of the error
% corresponding to a linearized version of
% the inverse problem (linearized around a point m0)E2=zeros(Mg,1);
m0 = 1.0;
im0 = floor((m0-mmin)/Dm)+1;
d0 = sin(k*m0.*zp).*z;
dddm = k.*zp.*z.*cos(k*m0.*zp);
for i=1:Mg
    dpre = d0 + dddm * (m(i)-m0);
    e = dobs - dpre;
    E2(i) = (e'*e);
end

% find minimum of error surface
[E2min, iE2min] = min(E2);
m2est=m(iE2min);

figure(1);
clf;
set(gca,'LineWidth',3);
set(gca,'FontSize',14);
hold on;
mminp=1; mmaxp=4;
axis( [mmin, mmax, 0, max(E1)] );
axis xy;

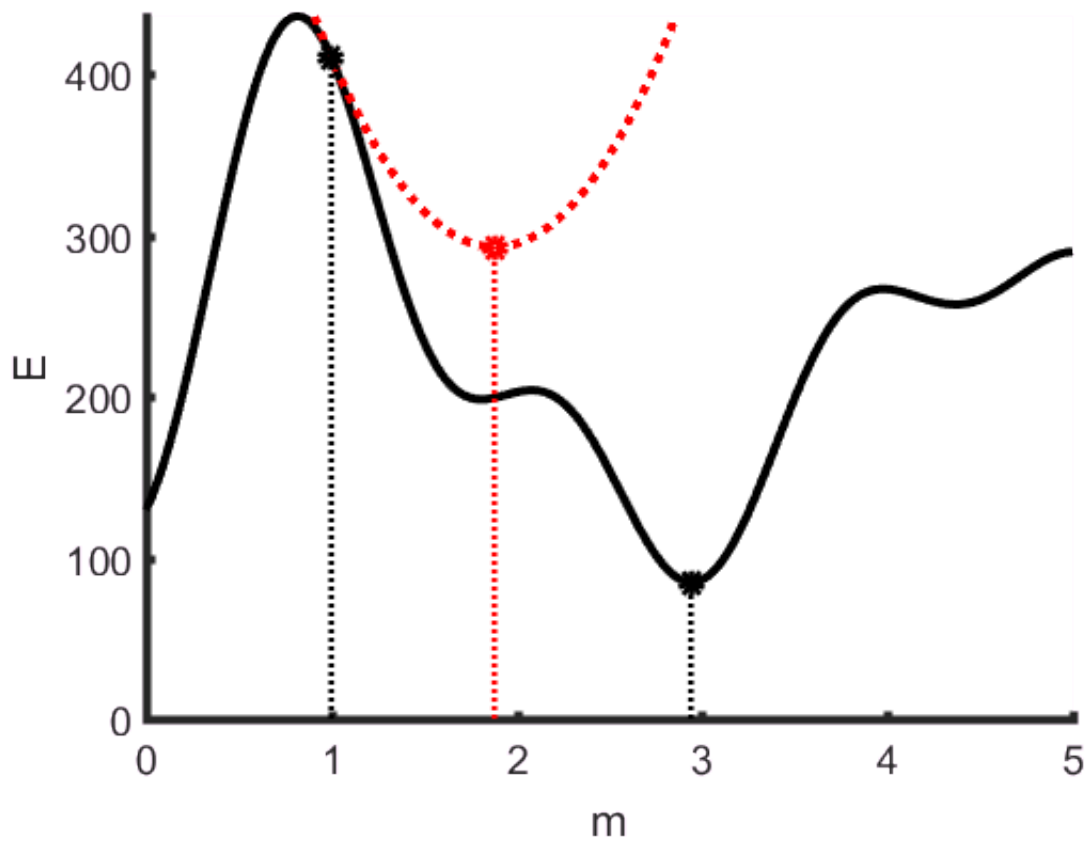
% true error surface, with circle at minimum, and line dropped to axis
plot( m, E1, 'k-', 'LineWidth', 3 );
plot( mlest, Elmin, 'ko', 'LineWidth', 3 );
plot( [mlest, mlest], [0, Elmin], 'k:', 'LineWidth', 2 );

% parabolic approximation, with circle at minimum, and line dropped to
% axis
plot( m, E2, 'r:', 'LineWidth', 3 );
plot( m2est, E2min, 'ro', 'LineWidth', 3 );
plot( [m2est, m2est], [0, E2min], 'r:', 'LineWidth', 2 );

plot( m0, E2(im0), 'ko', 'LineWidth', 3 );
plot( [m0, m0], [0, E2(im0)], 'k:', 'LineWidth', 2 );

xlabel('m');
ylabel('E');

```



% Figure 9.8 If the trial solution, , is too far from the global minimum mGM,  
 % the method may converge to a local minimum. MatLab script gda09\_10.