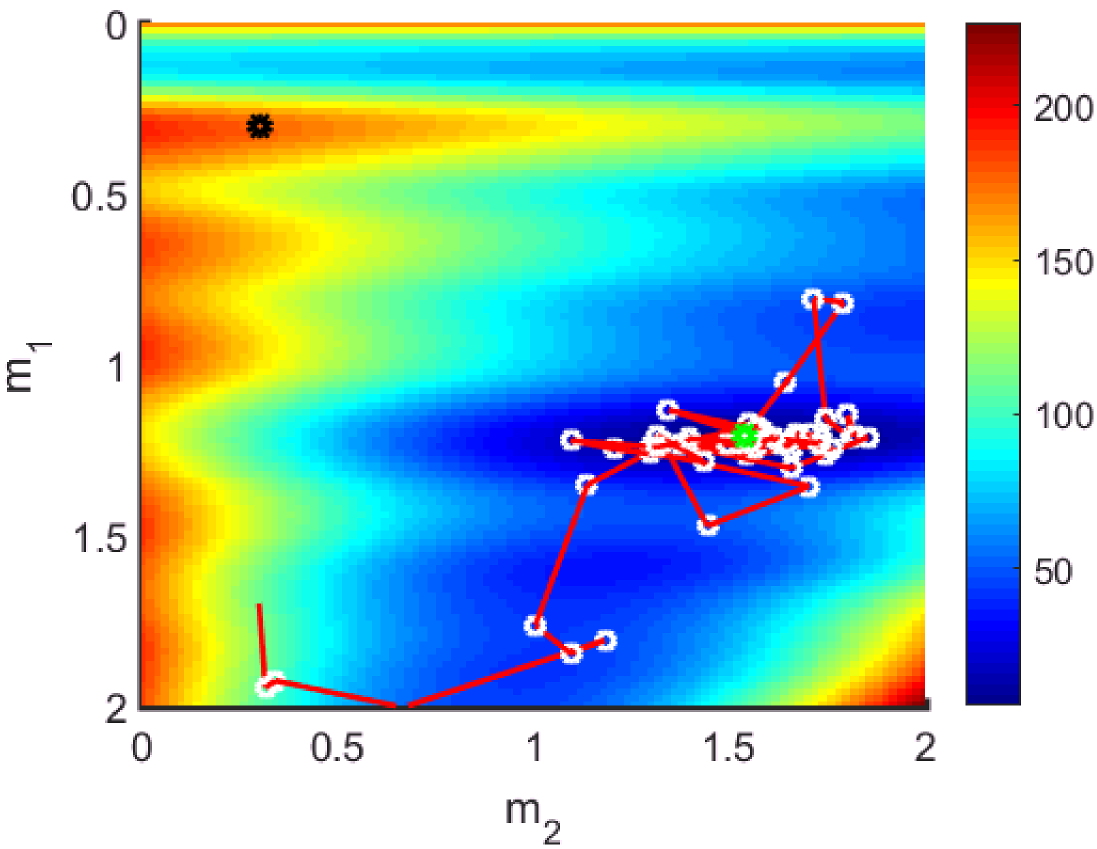
```
% gda09_16
% Simulated Annealing method applied to inverse problem
% d(x)=g(x, m1, m2) with g = sin(w0*m(1)*x) + m(1)*m(2)

% Supports Figire 9.14

% data are in a sinple auxillary variable, x
N=40;
xmin=0;
xmax=1.0;
Dx=(xmax-xmin)/(N-1);
x = Dx*[0:N-1]';

% true model parameters
M=2;
mt = [1.21, 1.54]';

% y=f(x, m1, m2);
w0=20;
dtrue = sin(w0*mt(1)*x) + mt(1)*mt(2);
sd=0.4;
dobs = dtrue + random('Normal',0,sd,N,1);

% plot data
figure(1);
clf;
set(gca,'LineWidth',3);
set(gca,'FontSize',14);
hold on;
axis( [0, xmax, 0, 4 ] );
plot(x,dtrue,'k-','LineWidth',3);
plot(x,dobs,'ko','LineWidth',3);
xlabel('x');
ylabel('d');

% 2D grid, for plotting purposes only
L = 101;
Dm = 0.02;
m1min=0;
m2min=0;
m1a = m1min+Dm*[0:L-1]';
m2a = m2min+Dm*[0:L-1]';
m1max = m1a(L);
m2max = m2a(L);

% compute error, E, on grid for plotting purposed only
E = zeros(L,L);
for j = [1:L]
for k = [1:L]
    dpre = sin(w0*m1a(j)*x) + m1a(j)*m2a(k);
    E(j,k) = (dobs-dpre)'*(dobs-dpre);
end
end

figure(2);
clf;
set(gca,'LineWidth',3);
set(gca,'FontSize',14);
colormap('jet');
```

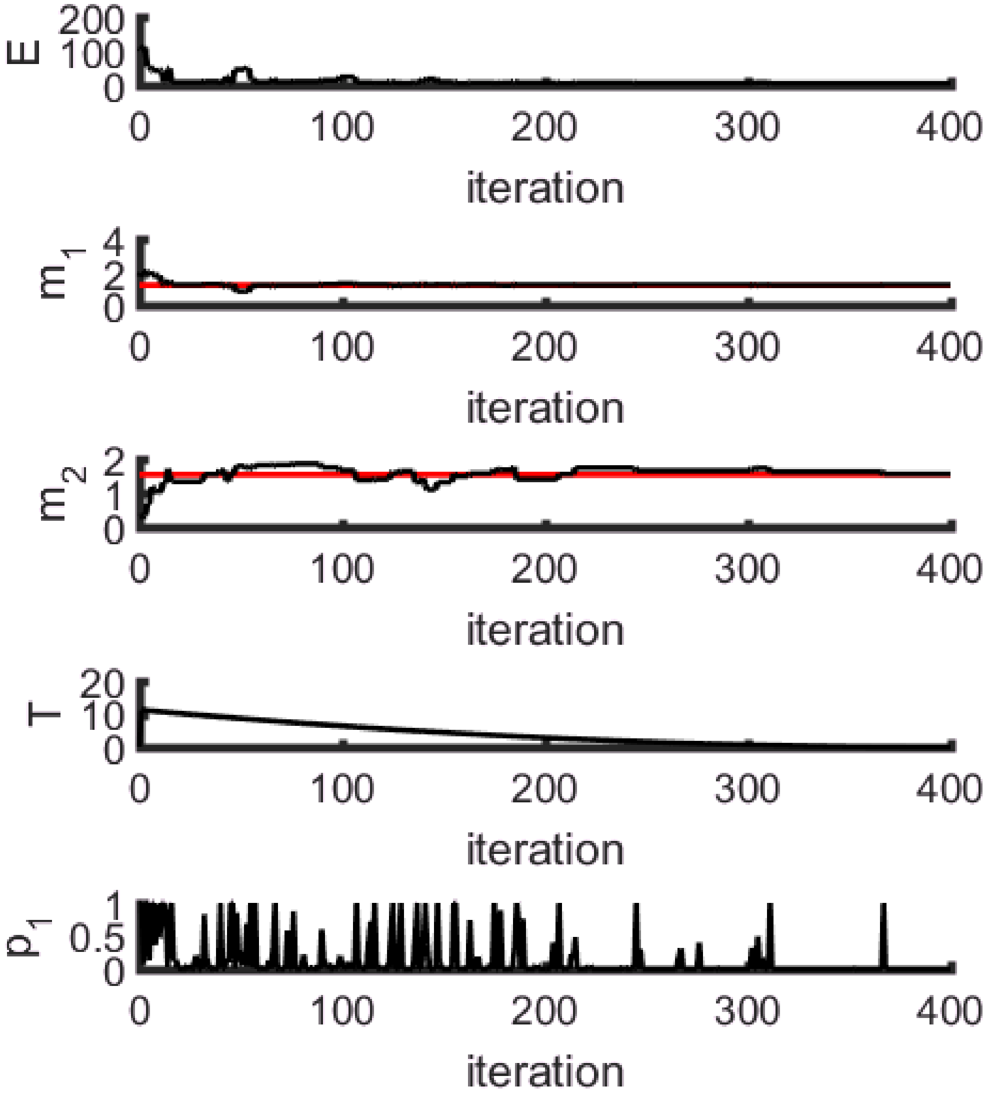
```
hold on;
axis( [m2min, m2max, m1min, m1max] );
axis ij;
imagesc( [m2min, m2max], [m1min, m1max], E);
colorbar;
xlabel('m_2');
ylabel('m_1');
plot( mt(2), mt(1), 'go', 'LineWidth', 3 );

% initial guess and corresponding error
mg=[1.7,0.3]';
dg = sin(w0*mg(1)*x) + mg(1)*mg(2);
Eg = (dobs-dg)'*(dobs-dg);
Eg0=Eg;
plot( mg(2), mg(2), 'ko', 'LineWidth', 3 );

Dm = 0.2;

Niter=400;
Ehis=zeros(Niter+1,1);
m1his=zeros(Niter+1,1);
m2his=zeros(Niter+1,1);
This=zeros(Niter+1,1);
p1his=zeros(Niter+1,1);
Ehis(1)=Eg;
m1his(1)=mg(1);
m2his(1)=mg(2);
This(1)=0;
p1his(1)=1;

for k = [1:Niter]

    % temperature falls off quadraticly with iteration
    T =  0.1 * Eg0 * ((Niter-k+1)/Niter)^2;

    % randomply pick model parameters and evaluate error
    ma(1) = random('norm',mg(1),Dm);
    ma(2) = random('norm',mg(2),Dm);
    da = sin(w0*ma(1)*x) + ma(1)*ma(2);
    Ea = (dobs-da)'*(dobs-da);

    % accept according to Metropolis rules
    if( Ea < Eg )
        mg=ma;
        Eg=Ea;
        p1his(k+1)=1;
    else
        p1 = exp( -(Ea-Eg)/T );
        p2 = random('unif',0,1);
        p1his(1+k)=p1;
        if( p1 > p2 )
            mg=ma;
            Eg=Ea;
        end
    end

    Ehis(1+k)=Eg;
    m1his(1+k)=mg(1);
    m2his(1+k)=mg(2);
    This(1+k)=T;
```

```
    plot( mg(2), mg(1), 'wo', 'LineWidth', 2 );
    plot( [m2his(1+k-1), m2his(1+k) ], [m1his(1+k-1), m1his(1+k) ], 'r', 'LineWidth', 2 );

end

m1est = m1his(Niter+1);
m2est = m2his(Niter+1);

plot( mt(2), mt(1), 'go', 'LineWidth', 3 );
```

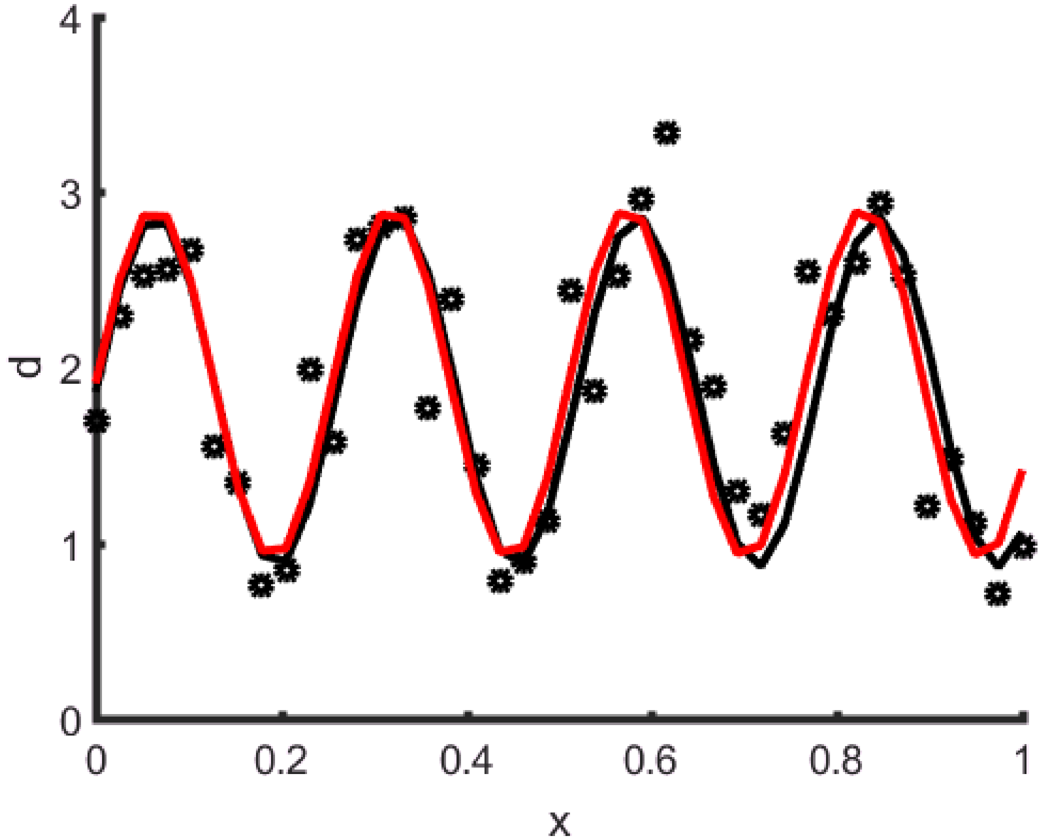



```
% Figure 9.14 Simulated annealing is used to solve the same nonlinear curve-fitting problem as
% in Figure 9.5. (B) Error surface (colors), showing true solution (green circle), and a serie
% solutions (white circles connected by red lines) determined by the method. (C) Plot of error
% E and model parameters m1 and m2 as a function of iteration number. MatLab script gda09_16.

figure(3);
set(gcf,'pos',[10, 10, 500, 600] );
clf;

subplot(5,1,1);
set(gca,'LineWidth',3);
set(gca,'FontSize',14);
hold on;
plot( [0:Niter], Ehis, 'k-', 'LineWidth', 2 );
xlabel('iteration');
ylabel('E');

subplot(5,1,2);
set(gca,'LineWidth',3);
set(gca,'FontSize',14);
hold on;
```

```
plot( [0, Niter], [mt(1), mt(1)], 'r', 'LineWidth', 2 );
plot( [0:Niter], m1his, 'k-', 'LineWidth', 2 );
xlabel('iteration');
ylabel('m_1');

subplot(5,1,3);
set(gca,'LineWidth',3);
set(gca,'FontSize',14);
hold on;
plot( [0, Niter], [mt(2), mt(2)], 'r', 'LineWidth', 2 );
plot( [0:Niter], m2his, 'k-', 'LineWidth', 2 );
xlabel('iteration');
ylabel('m_2');

subplot(5,1,4);
set(gca,'LineWidth',3);
set(gca,'FontSize',14);
hold on;
plot( [0:Niter], This, 'k-', 'LineWidth', 2 );
xlabel('iteration');
ylabel('T');

subplot(5,1,5);
set(gca,'LineWidth',3);
set(gca,'FontSize',14);
hold on;
plot( [0:Niter], p1his, 'k-', 'LineWidth', 2 );
xlabel('iteration');
ylabel('p_1');
```




```
% Figure 9.14 Simulated annealing is used to solve the same nonlinear curve-fitting problem as
% in Figure 9.5. (C) Plot of error E and model parameters m1 and m2 as a function of iteration
% number. MatLab script gda09_16.

% evaluate probability of acceptance and plot it
figure(1);
dpre = sin(w0*m1est*x) + m1est*m2est;
plot(x,dpre,'r-','LineWidth',3);
```




```
% Figure 9.14 Simulated annealing is used to solve the same nonlinear curve-fitting problem as
% in Figure 9.5. (A) The observed data (black circles) are computed from the true data (black
% curve) by adding random noise. The predicted data (red curve) are based on the results of th
% method. MatLab script gda09_16.

fprintf('m true %f %f\n', mt(1), mt(2) );
```

```
m true 1.210000 1.540000
```

```
fprintf('m est  %f %f\n', m1est, m2est );
```

```
m est  1.230720 1.553661
```