## *% plot_2D_Fcost.m*

```
%
% This MATLAB m-file makes a contour plot of
% a cost function of 2 variables.
%
% K. Beers
% MIT ChE
% 12/6/2001

% Define the parameters that specify the form of the cost function.
x_min = [0; 0];
kappa_1 = 5; kappa_2 = 1;

% Now, we calculate the cost function on a fine grid of points.
num_fine = 50;
x1_fine = linspace(-5,5,num_fine);
x2_fine = linspace(-5,5,num_fine);

[X1_fine,X2_fine] = meshgrid(x1_fine,x2_fine);
F_fine = zeros(num_fine,num_fine);
for ix1=1:num_fine
   for ix2=1:num_fine
      x1 = X1_fine(ix1,ix2);
      x2 = X2_fine(ix1,ix2);
      x = [x1; x2];
      distsq = dot(x-x_min,x-x_min);
      F_fine(ix1,ix2) = distsq + ...
         kappa_1*((x(1)-x_min(1))^2) + ...
         kappa_2*((x(2)-x_min(2))^4);
   end
end

% We now calculate the gradient on a coarse grid.
num_coarse = 10;
x1_coarse = linspace(-5,5,num_coarse);
delta_x1_coarse = x1_coarse(2) - x1_coarse(1);
x2_coarse = linspace(-5,5,num_coarse);
delta_x2_coarse = x2_coarse(2) - x2_coarse(1);

[X1_coarse,X2_coarse] = meshgrid(x1_coarse,x2_coarse);
F_coarse = zeros(num_coarse,num_coarse);
for ix1=1:num_coarse
   for ix2=1:num_coarse
      x1 = X1_coarse(ix1,ix2);
      x2 = X2_coarse(ix1,ix2);
      x = [x1; x2];
      distsq = dot(x-x_min,x-x_min);
      F_coarse(ix1,ix2) = distsq + ...
         kappa_1*((x(1)-x_min(1))^2) + ...
```

```
        kappa_2*((x(2)-x_min(2))^4);
   end
end

[DF_DX1,DF_DX2] = gradient(F_coarse, ...
   delta_x1_coarse,delta_x2_coarse);


% Then, we make a plot of the cost function with the
% gradients marked.
figure;
hold on;
contour(X1_fine,X2_fine,F_fine,20);
quiver(X1_coarse,X2_coarse,DF_DX1,DF_DX2);
plot(x_min(1),x_min(2),'o');

% We now call the minimization routine to find the
% minimum starting at an initial guess of the solution.
fun_name = 'cost_function_1';
Opt.iprint_traj = 1;
x_guess = input('Enter initial guess [x1; x2] : ');
[x,iflag,Traj] = simple_minimizer(fun_name,x_guess,Opt);
disp(' ');
disp('Final estimate of minimum : ');
x,
iflag,
num_traj = length(Traj.iter);
x_last = x_guess;
plot(x_guess(1),x_guess(2),'o');
for itraj=1:num_traj;
   x_traj = Traj.x(itraj,:)';
   plot(x_traj(1),x_traj(2),'o');
   plot([x_last(1) x_traj(1)],[x_last(2) x_traj(2)]);
   x_last = x_traj;
end

title('Cost function');
xlabel('x_1'); ylabel('x_2');
```