Matlab implementation: run fminunc

1. Write a Matlab function containing the variables x (here, called z) as the first input and at least one ouput (objective value f):

```
function [f,g] = myobj1(z,beta)

x=z(1);
y=z(2);

f = (y-sin(beta*x)-0.1*x*x)^2;
g(1) = -2*(beta*cos(beta*x)+0.2*x)*(y-sin(beta*x)-0.1*x*x);
g(2) = 2*(y-sin(beta*x) -0.1*x*x);
```

- Name of the file: same as the name of the function (here, `myobj1.m`).
- At least one output (`f`), gradient output `g` is optional.
- Recommendation: to speed up the optimization process, provide `g` as output when computation of `g` is possible.

Matlab implementation: run fminunc

2. Run the Matlab solver . This is another Matlab file, *e.g.*, `runfminunc.m`:

```
clear all; % clear memory
close all; % close figures

beta = 2;

options = optimoptions(’fminunc’);
options = optimoptions(options,’Display’,’iter’,...
   ’MaxFunctionEvaluations’,1000,’StepTolerance’,1e-10,...
   ’SpecifyObjectiveGradient’,true,’CheckGradients’,true);

% Initial solution
x0=[10;10];

% Compute initial objective value
fprintf(’x0=(%.2f,%.2f), f=%f\n’,x0(1),x0(2),...
         myobj1(x0,beta));
```

```
% Run optimization solver
 xsol = fminunc(@(x)myobj1(x,beta),x0,options);

fprintf('Final solution=(%.2f,%.2f), f=%f\n',...
        xsol(1),xsol(2), myobj1(xsol,beta));

% 2D and 3D display

x=0:.1:6;
y=-1:.1:5;

% Pixel coordinates:
[xx,yy]=meshgrid(x,y);

zz=zeros(size(xx));
for i=1:length(xx(:)),
  zz(i)=myobj1([xx(i),yy(i)],beta);
end;
```

```
% 2D image
figure(1);
imagesc(x,y,zz);
colorbar;
axis xy
xlabel(’x’); ylabel(’y’);

% 3D surface
figure(2);
mesh(x,y,zz);
colormap pink
shading flat
xlabel(’x’); ylabel(’y’);
```