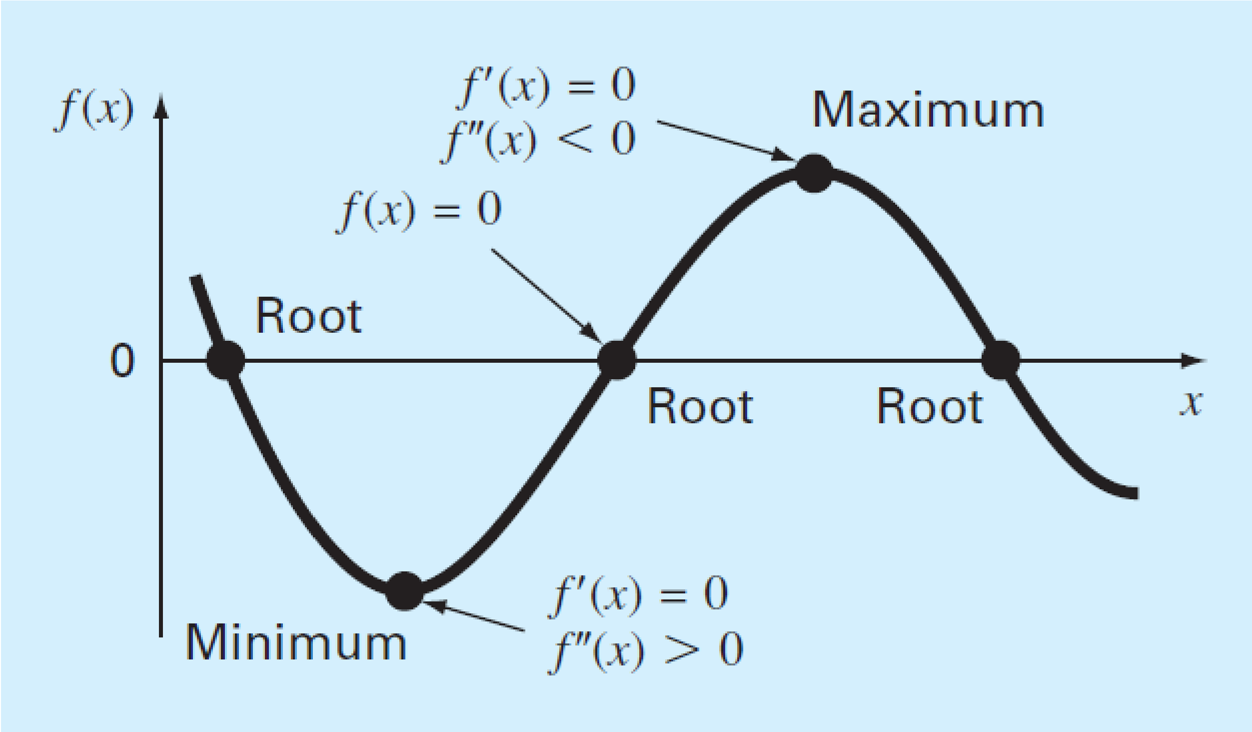
Optimization

Hsiao-Lung Chan
Dept Electrical Engineering
Chang Gung University, Taiwan
chanhl@mail.cgu.edu.tw

Optimization problem

- Optimization deals with finding the maxima and minima of a function that depends on one or more variables



A bungee jumper projects upward at a specified velocity (V_0) with a linear drag

$$\frac{dv}{dt} = g - \frac{c}{m}v$$

Laplace transforms

$$SV - v(0) = \frac{g}{s} - \frac{c}{m}V$$

$$V = \frac{g}{s(s+c/m)} + \frac{v(0)}{s+c/m}$$

$$\frac{g}{s(s+c/m)} = \frac{A}{s} + \frac{B}{s+c/m} = \frac{A(s+c/m)+Bs}{s(s+c/m)}$$

$$\Rightarrow A + B = 0 \quad g = c/m \times A$$

$$\Rightarrow A = mg/c \quad B = -mg/c$$

$$V = \frac{mg/c}{s} + \frac{-mg/c}{s+c/m} + \frac{v(0)}{s+c/m}$$

Inverse Laplace transforms

$$v(t) = \frac{mg}{c} - \frac{mg}{c}e^{-(c/m)t} + v(0)e^{-(c/m)t}$$

$$= -v_0 e^{-(c/m)t} + \frac{mg}{c}(1 - e^{-(c/m)t})$$

Inverse z-direction

$$v_U(t) = v_0 e^{-(c/m)t} - \frac{mg}{c}(1 - e^{-(c/m)t})$$

$$z = \frac{-m}{c}v_0 e^{-(c/m)t} - \frac{mg}{c}(t + \frac{m}{c}e^{-(c/m)t}) + k$$

$$z_0 = \frac{-m}{c}v_0 - \frac{mg}{c}\frac{m}{c} + k$$

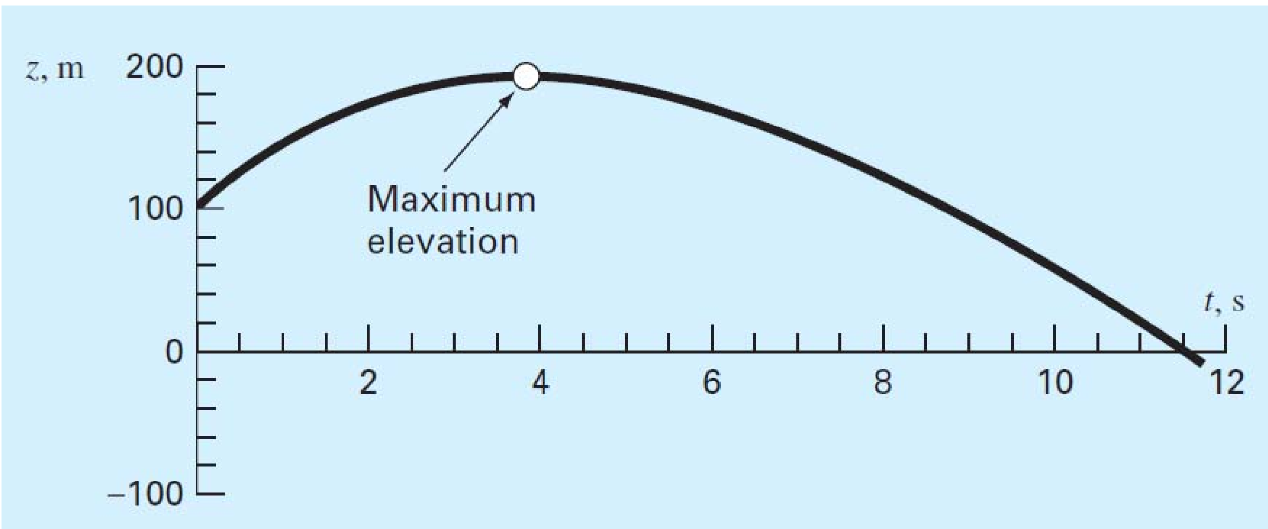
Altitude as a function of time

$$z = z_0 + \frac{m}{c}(v_0 + \frac{mg}{c})(1 - e^{-(c/m)t}) - \frac{mg}{c}t$$

M-function for function plot

```
function plotFunc(func,xmin,xmax,ns)
  if nargin < 4, ns = 50; end
  x = linspace(xmin,xmax,ns);
  f = func(x);
  plot(x,f);
end
```

$g = 9.81$ m/s^2, $z_0 = 100$ m, $v_0 = 55$ m/s, $m = 80$ kg, $c = 15$ kg/s



Determining the optimum analytically by root location

$$\frac{dz}{dt} = v_0 e^{-(c/m)t} - \frac{mg}{c}\left(1 - e^{-(c/m)t}\right)$$

Find the root that makes $\frac{dz}{dt} = 0$

$$t = \frac{m}{c} \ln\left(1 + \frac{c v_0}{mg}\right)$$

Assume that $g = 9.81$ m/s^2, $z_0 = 100$ m, $v_0 = 55$ m/s, $m = 80$ kg, $c = 15$ kg/s

$$t = \frac{80}{15} \ln\left(1 + \frac{15(55)}{80(9.81)}\right) = 3.83166 \text{ s}$$

$$z = 100 + \frac{80}{15}\left(50 + \frac{80(9.81)}{15}\right)\left(1 - e^{-(15/80)3.83166}\right) - \frac{80(9.81)}{15}(3.83166) = 192.8609 \text{ m}$$

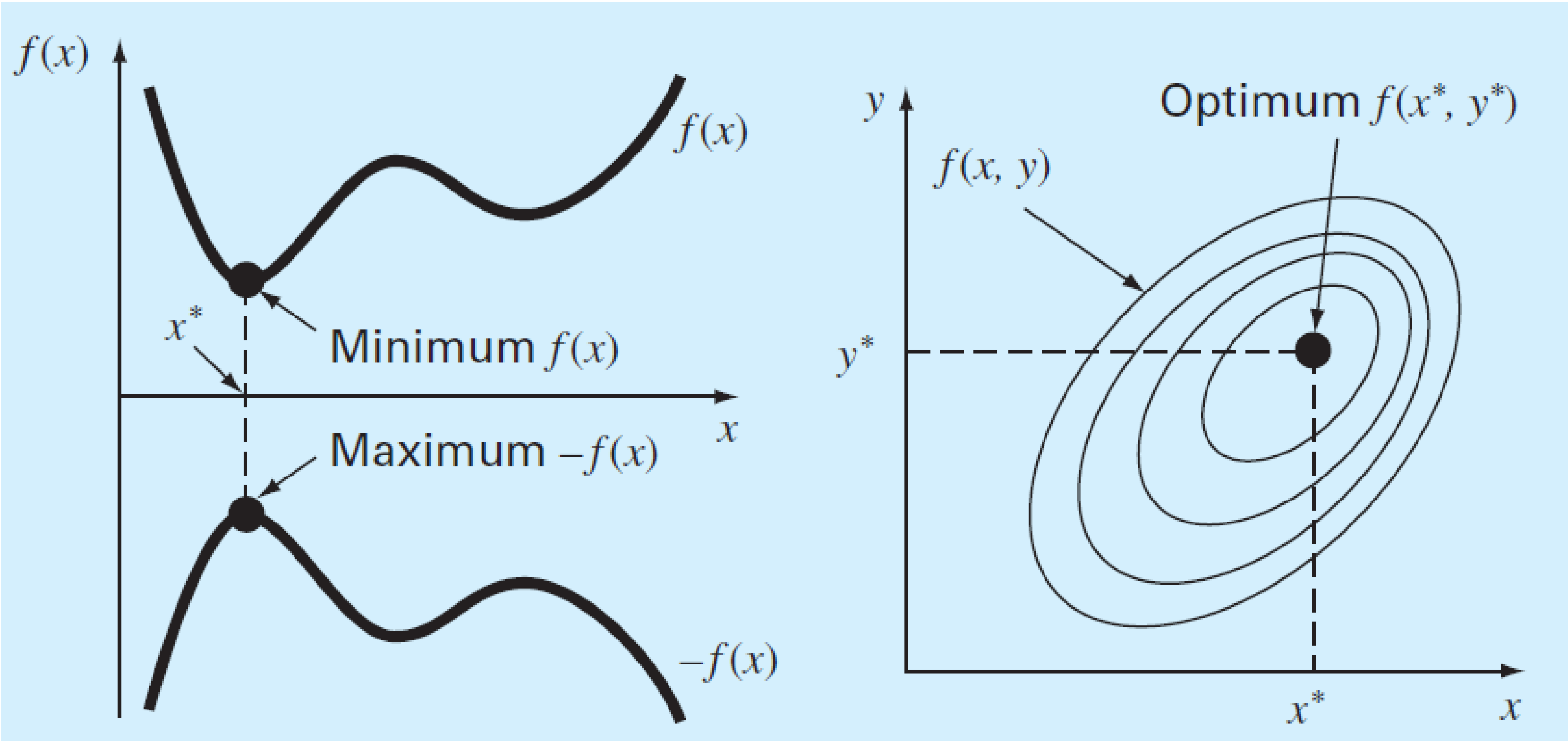
Determine whether is maximum or minimum by second derivative

$$\frac{d^2z}{dt^2} = -\frac{c}{m} v_0 e^{-(c/m)t} - g e^{-(c/m)t} = -9.81 \frac{\text{m}}{\text{s}^2}$$

Optimization

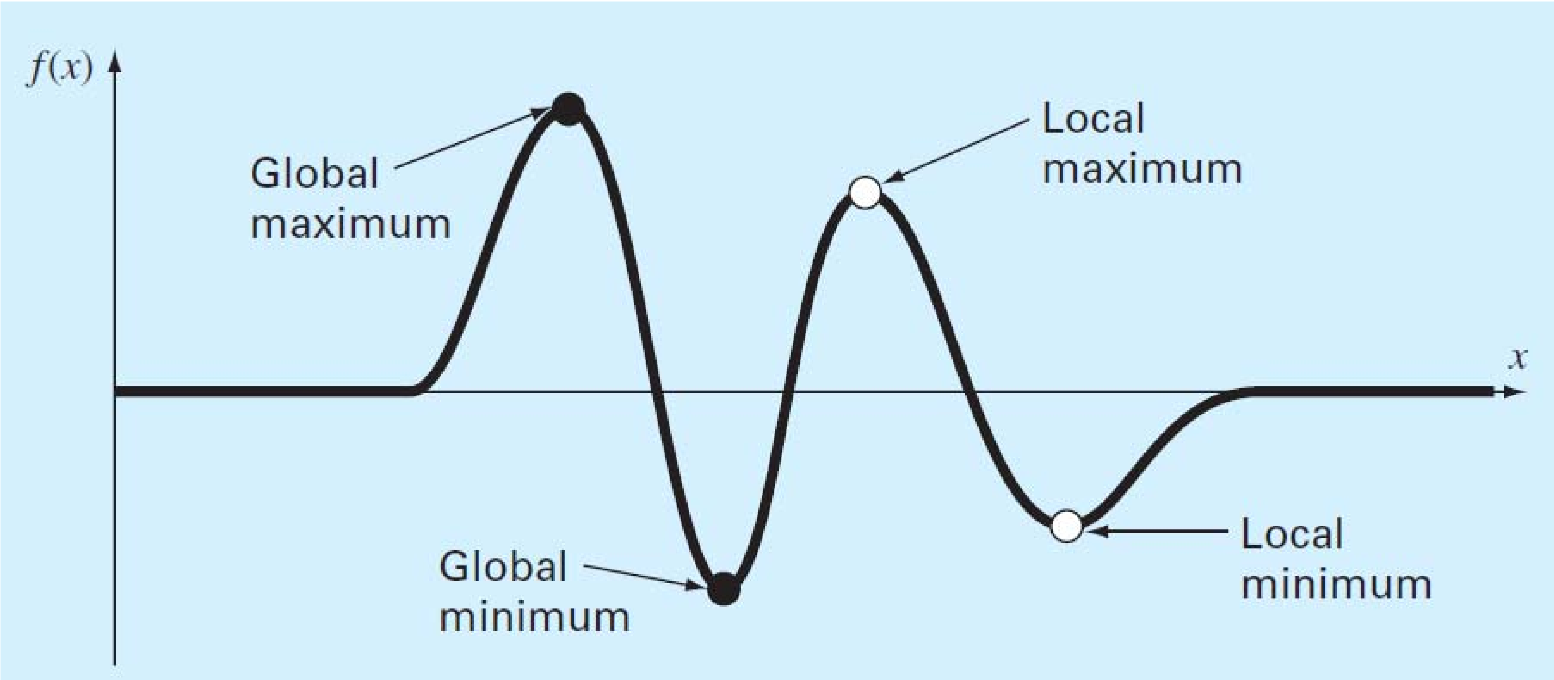
One-dimensional optimization

Two-dimensional optimization

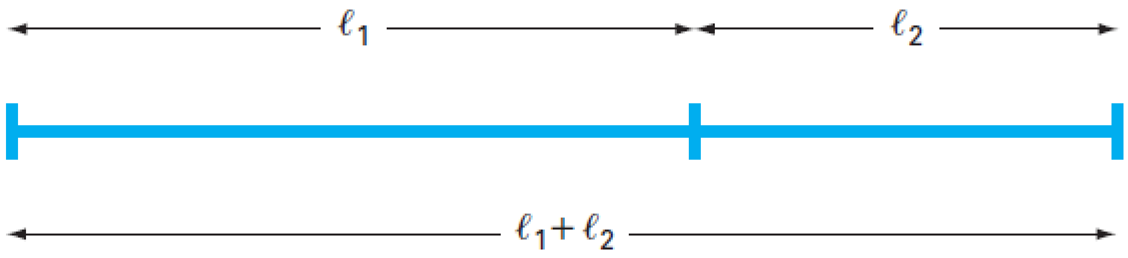


Global vs. local

- A *global optimum* represents the best solution.
- A *local optimum* is better than its immediate neighbors.
- Cases that include local optima are called *multimodal*



Golden ratio



$$\frac{\ell_1 + \ell_2}{\ell_1} = \frac{\ell_1}{\ell_2}$$

$$\phi^2 - \phi - 1 = 0 \qquad \text{where } \phi = \ell_1/\ell_2$$

$$\phi = \frac{1+\sqrt{5}}{2} = 1.61803398874989\ldots$$

Golden-section search

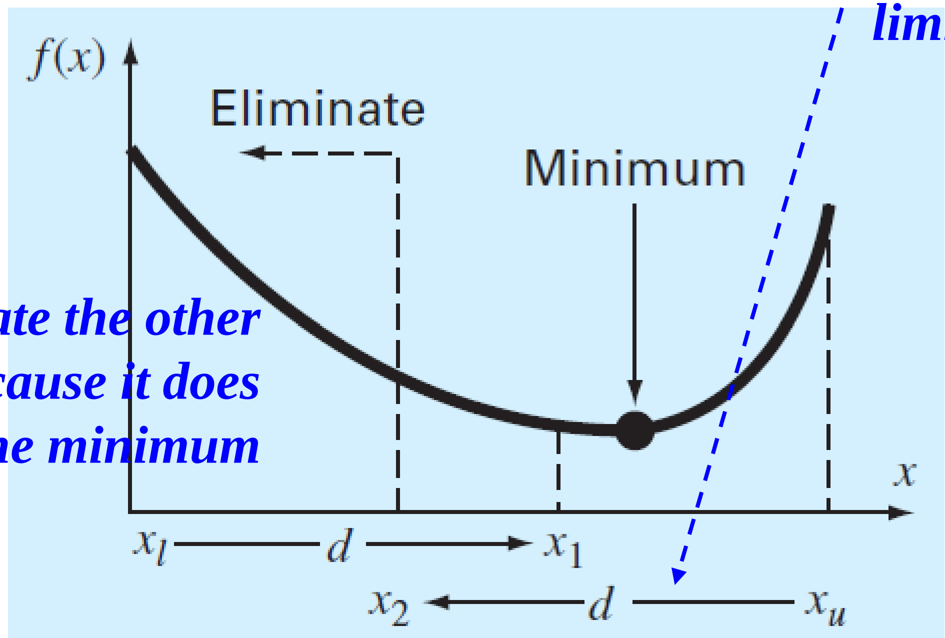
- Specify a lower guess x_l and an upper guess x_u that bracketed a optimum
- Choosing two interior points according to the golden ratio

$$d = (\phi - 1)(x_u - x_l)$$

$$x_1 = x_l + d$$

$$x_2 = x_u - d$$

Condition 1: If $f(x_1)<f(x_2)$, select this segment and x_2 becomes the new lower limit



Eliminate the other segment because it does not contain the minimum

Condition 2: If $f(x_2)<f(x_1)$, x_1 becomes the new upper limit

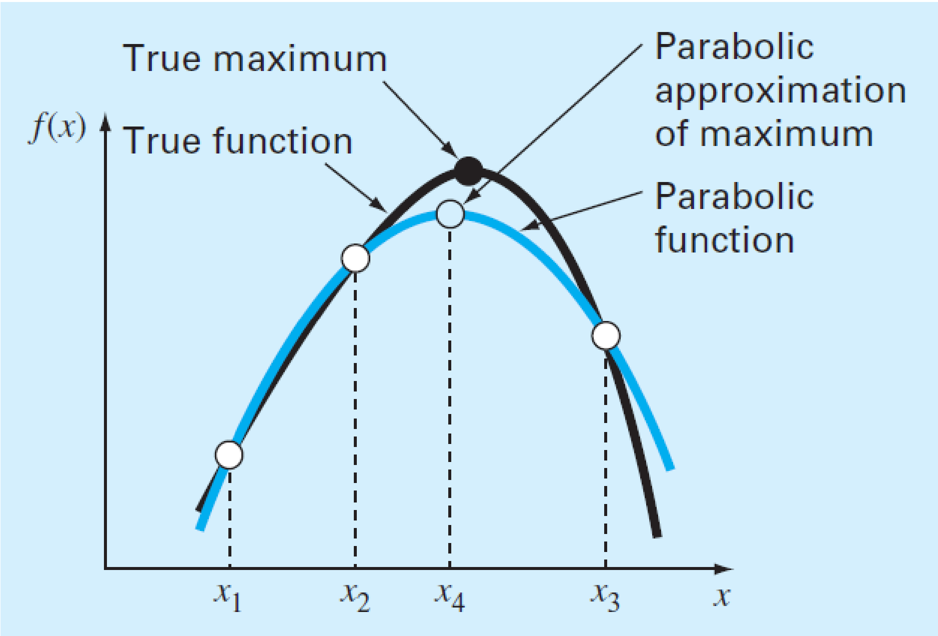
M-file to determine the minimum by the golden-section search

```
function [x,fx,ea,iter]=goldmin(f,xl,xu,es,maxit)
 if nargin<4|isempty(es), es=0.0001;end
 if nargin<5|isempty(maxit), maxit=50;end
 phi=(1+sqrt(5))/2;
 iter=0;
 while(1)
   d = (phi-1)*(xu - xl);
   x1 = xl + d;   x2 = xu - d;
   if f(x1) < f(x2)
     xopt = x1;   xl = x2;
   else
     xopt = x2;   xu = x1;
   end
   iter=iter+1;
   if xopt~=0, ea = (2 - phi) * abs((xu - xl) / xopt) * 100; end
   if ea <= es | iter >= maxit, break, end
 end
 x=xopt; fx=f(xopt);
```

Parabolic interpolation

- Construct a parabola based on x_1, x_2 and x_3
- The location of the maximum/minimum of a parabola defined as

$$x_4 = x_2 - \frac{1}{2}\frac{(x_2 - x_1)^2[f(x_2) - f(x_3)] - (x_2 - x_3)^2[f(x_2) - f(x_1)]}{(x_2 - x_1)[f(x_2) - f(x_3)] - (x_2 - x_3)[f(x_2) - f(x_1)]}$$



MATLAB’s fminbnd function

- Combines the golden-section search and the parabolic interpolation
- Example: Determine the highest altitude of a bungee jumper with a upward initial velocity

```
g=9.81; v0=55; m=80; c=15; z0=100;
z=@(t) -(z0+m/c*(v0+m*g/c)*(1-exp(-c/m*t))-m*g/c*t);
[x,f]=fminbnd(z,0,8);
```

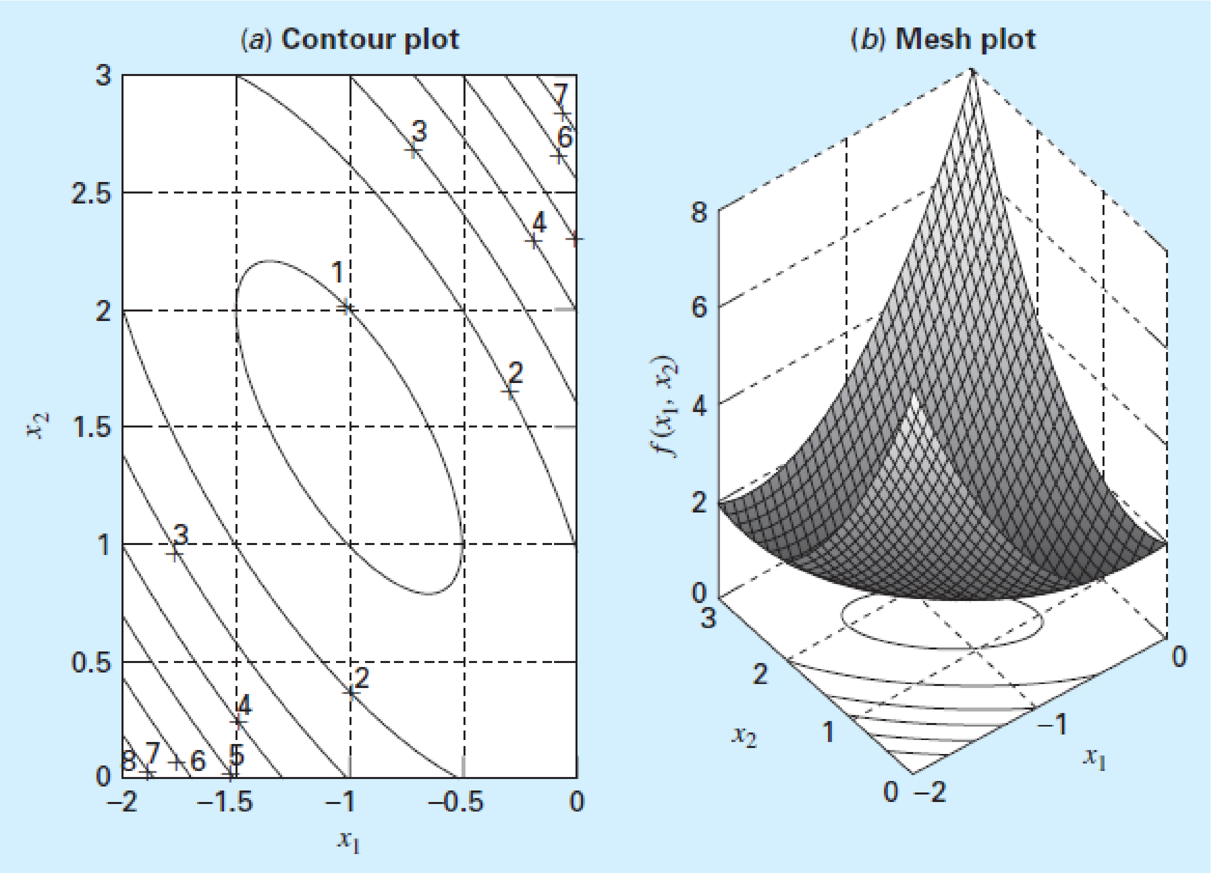
fminbnd function (cont.)

```
options = optimset('display','iter');
fminbnd(z,0,8,options)
```

Func-count	x	f(x)	Procedure
1	3.05573	-189.759	initial
2	4.94427	-187.19	golden
3	1.88854	-171.871	golden
4	3.87544	-192.851	parabolic
5	3.85836	-192.857	parabolic
6	3.83332	-192.861	parabolic
7	3.83162	-192.861	parabolic
8	3.83166	-192.861	parabolic
9	3.83169	-192.861	parabolic

Multidimensional optimization

- Graphic approach provides a handy means to visualize the function



Visualization of $f(x_1, x_2) = 2 + x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$

```
x=linspace(-2,0,40);
y=linspace(0,3,40);
[X,Y] = meshgrid(x,y);
Z=2+X-Y+2*X.^2+2*X.*Y+Y.^2;
subplot(1,2,1);
cs=contour(X,Y,Z); clabel(cs);
xlabel('x_1'); ylabel('x_2');
title('(a) Contour plot'); grid;
subplot(1,2,2);
surfc(X,Y,Z);
xlabel('x_1'); ylabel('x_2'); zlabel('f(x_1,x_2)');
title('(b) Mesh plot');
```

```
function plotFunc2(func,range1,range2,ns)
  if nargin<4,   ns=40;   end
  x=linspace(range1(1),range1(2),ns);
  y=linspace(range2(1),range2(2),ns);
  [X,Y] = meshgrid(x,y);
  Z=func(X,Y);
  subplot(1,2,1);
  cs=contour(X,Y,Z); clabel(cs);
  xlabel('x_1'); ylabel('x_2');
  title('(a) Contour plot'); grid;
  subplot(1,2,2);
  surfc(X,Y,Z);
  xlabel('x_1'); ylabel('x_2'); zlabel('f(x_1,x_2)');
  title('(b) Mesh plot');
```

```
fm=@(X,Y) 2+X-Y+2*X.^2+2*X.*Y+Y.^2;
plotFunc2(fm,[-2 0],[0 3],40)
```

MATLAB's fminsearch function

- Determine the minimum of a multidimensional function based on the Nelder-Mead method.
- Example

```
f=@(x) 2+x(1)-x(2)+2*x(1)^2+2*x(1)*x(2)+x(2)^2;
[x,fval]=fminsearch(f,[-0.5,0.5]);
```

Reference

- Steven C. Chapra "Applied Numerical Methods with MATLAB", 3rd ed., McGraw Hill, 2012.