

# Optimization

Berlin Chen

Department of Computer Science & Information Engineering  
National Taiwan Normal University

Reference:

1. *Applied Numerical Methods with MATLAB for Engineers*, Chapter 7 & Teaching material

# Chapter Objectives (1/2)

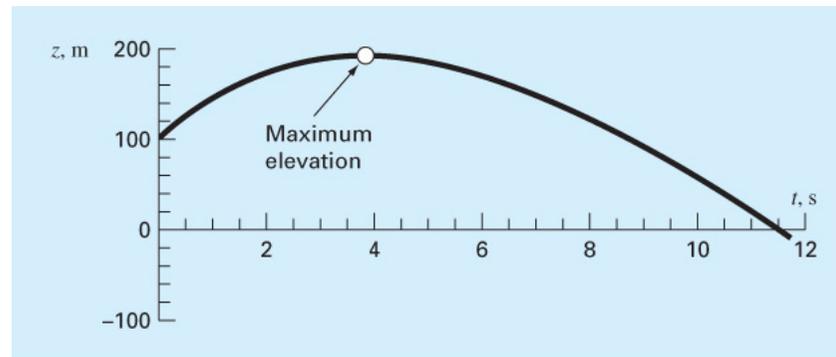
- Understanding why and where optimization occurs in engineering and scientific problem solving
- Recognizing the difference between one-dimensional and multi-dimensional optimization
- Distinguishing between global and local optima
- Knowing how to recast a maximization problem so that it can be solved with a minimizing algorithm
- Being able to define the golden ratio and understand why it makes one-dimensional optimization efficient

# Chapter Objectives (2/2)

- Locating the optimum of a single-variable function with the golden-section search
- Locating the optimum of a single-variable function with parabolic interpolation
- Knowing how to apply the *fminbnd* function to determine the minimum of a one-dimensional function
- Being able to develop MATLAB contours and surface plots to visualize two-dimensional functions
- Knowing how to apply the *fminsearch* function to determine the minimum of a multidimensional function

# Optimization (1/2)

- Optimization is the process of creating something that is as effective as possible
- From a mathematical perspective, optimization deals with finding the maxima and minima of a function that depends on one or more variables



a bungee jumper is being projected upward at a specific velocity  $v_0$

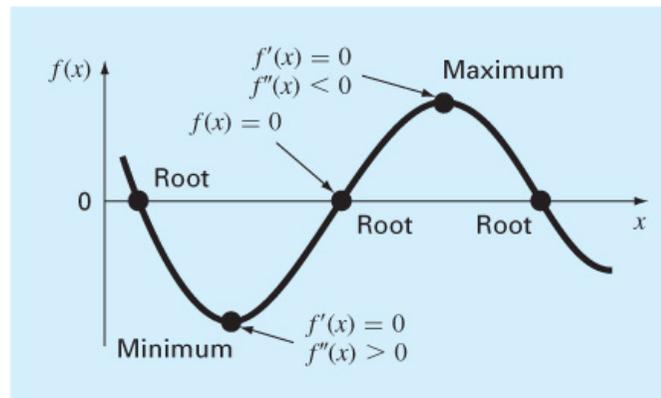
**FIGURE 7.1**

Elevation as a function of time for an object initially projected upward with an initial velocity.

$$z = z_0 + \frac{m}{c} \left( v_0 + \frac{mg}{c} \right) (1 - e^{-(c/m)t}) - \frac{mg}{c} t \quad (7.1)$$

# Optimization (2/2)

- The optima are the points where the curve is flat
  - Maximum:  $f'(x)=0$  and  $f''(x)<0$
  - Minimum:  $f'(x)=0$  and  $f''(x)>0$



**FIGURE 7.2**

A function of a single variable illustrating the difference between roots and optima.

- Intuitively, we can differentiate the function and locate the root (i.e., the zero) of the new function  $f'(x)$ 
  - Viz. solving the root problem  $f'(x)=0$

# Optimization: An Example (1/2)

- Example 7.1

Determining the Optimum Analytically by Root Location

**Problem Statement.** Determine the time and magnitude of the peak elevation based on Eq. (7.1). Use the following parameter values for your calculation:  $g = 9.81 \text{ m/s}^2$ ,  $z_0 = 100 \text{ m}$ ,  $v_0 = 55 \text{ m/s}$ ,  $m = 80 \text{ kg}$ , and  $c = 15 \text{ kg/s}$ .

**Solution.** Equation (7.1) can be differentiated to give

$$\frac{dz}{dt} = v_0 e^{-(c/m)t} - \frac{mg}{c} (1 - e^{-(c/m)t}) \quad \text{(taking the first derivative)} \quad (\text{E7.1.1})$$

Note that because  $v = dz/dt$ , this is actually the equation for the velocity. The maximum elevation occurs at the value of  $t$  that drives this equation to zero. Thus, the problem amounts to determining the root. For this case, this can be accomplished by setting the derivative to zero and solving Eq. (E7.1.1) analytically for

$$t = \frac{m}{c} \ln \left( 1 + \frac{cv_0}{mg} \right)$$

Substituting the parameters gives

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

# Optimization: An Example (2/2)

- Example 7.1

This value along with the parameters can then be substituted into Eq. (7.1) to compute the maximum elevation as

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

We can verify that the result is a maximum by differentiating Eq. (E7.1.1) to obtain the second derivative

(taking the second derivative)

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

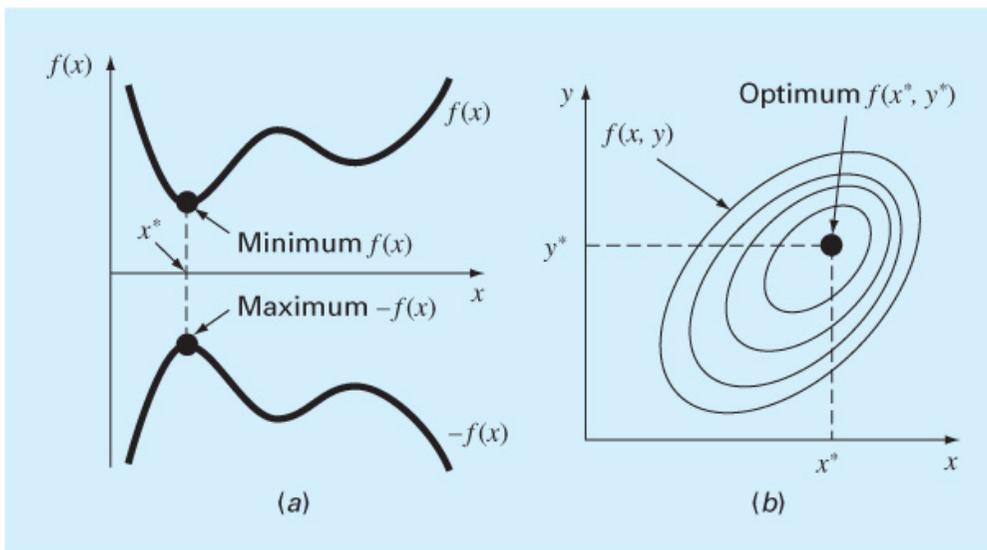
The fact that the second derivative is negative tells us that we have a maximum.

# Multidimensional Optimization

- One-dimensional problems involve functions that depend on a single dependent variable; for example,  $f(x)$
- Multidimensional problems involve functions that depend on two or more dependent variables; for example,  $f(x,y)$

**FIGURE 7.3**

(a) One-dimensional optimization. This figure also illustrates how minimization of  $f(x)$  is equivalent to the maximization of  $-f(x)$ . (b) Two-dimensional optimization. Note that this figure can be taken to represent either a maximization (contours increase in elevation up to the maximum like a mountain) or a minimization (contours decrease in elevation down to the minimum like a valley).



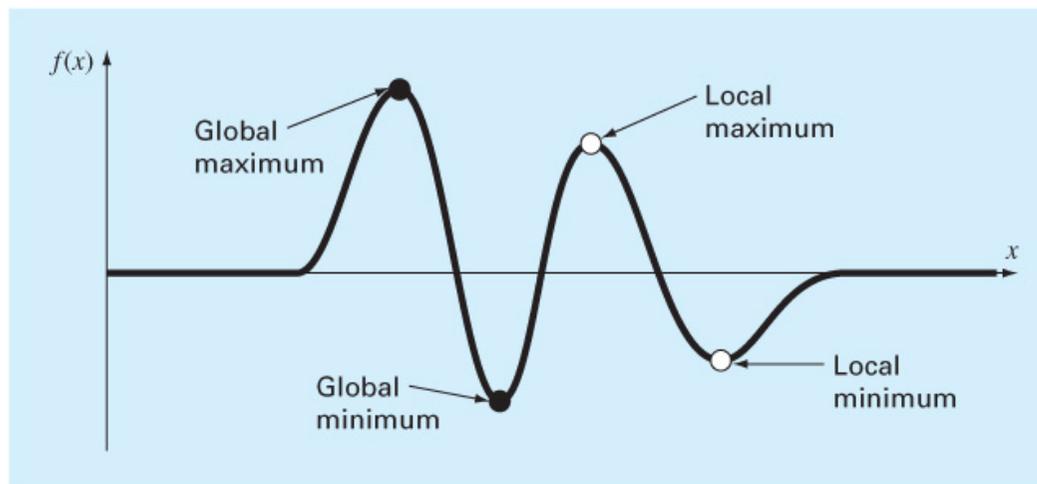
The process of finding a maximum versus finding minimum is essentially identical because the same value  $x^*$  both minimizes  $f(x)$  and maximizes  $-f(x)$

# Global Optimum vs. Local Optimum

- A **global optimum** represents the very best solution while a **local optimum** is better than its immediate neighbors
  - Cases that include local optima are called **multimodal**
- Generally desire to find the global optimum

**FIGURE 7.4**

A function that asymptotically approaches zero at plus and minus  $\infty$  and has two maximum and two minimum points in the vicinity of the origin. The two points to the right are local optima, whereas the two to the left are global.

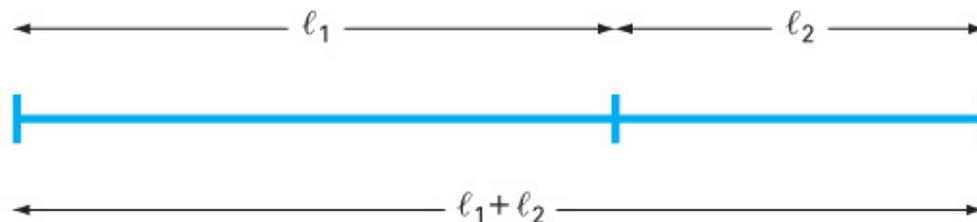


# Golden-Section Search (1/4)

- Search algorithm for finding a minimum on an interval  $[x_l, x_u]$  with a *single* minimum (*unimodal* interval)
- Uses the **golden ratio**  $\phi = 1.6180\dots$  (pronounced: *fee*) to determine location of two interior points  $x_1$  and  $x_2$ ; by using the golden ratio, one of the interior points can be re-used in the next iteration

**FIGURE 7.5**

Euclid's definition of the golden ratio is based on dividing a line into two segments so that the ratio of the whole line to the larger segment is equal to the ratio of the larger segment to the smaller segment. This ratio is called the golden ratio.



$$\frac{l_1 + l_2}{l_1} = \frac{l_1}{l_2}$$

If we set  $\frac{l_1}{l_2} = \phi$

then we will have  $\phi^2 - \phi - 1 = 0$ .

The positive of the above equation will be (i.e., the golden ratio)

$$\phi = \frac{1 + \sqrt{5}}{2} = 1.6180\dots$$

# Golden-Section Search (2/4)

- ***golden ratio*** was originally defined by Euclid
  - A straight line is said to have been cut in extreme and mean ratio, as the whole line is to the greater segment, so is the greater to the lesser
  - It has long been considered aesthetically pleasing in Western cultures

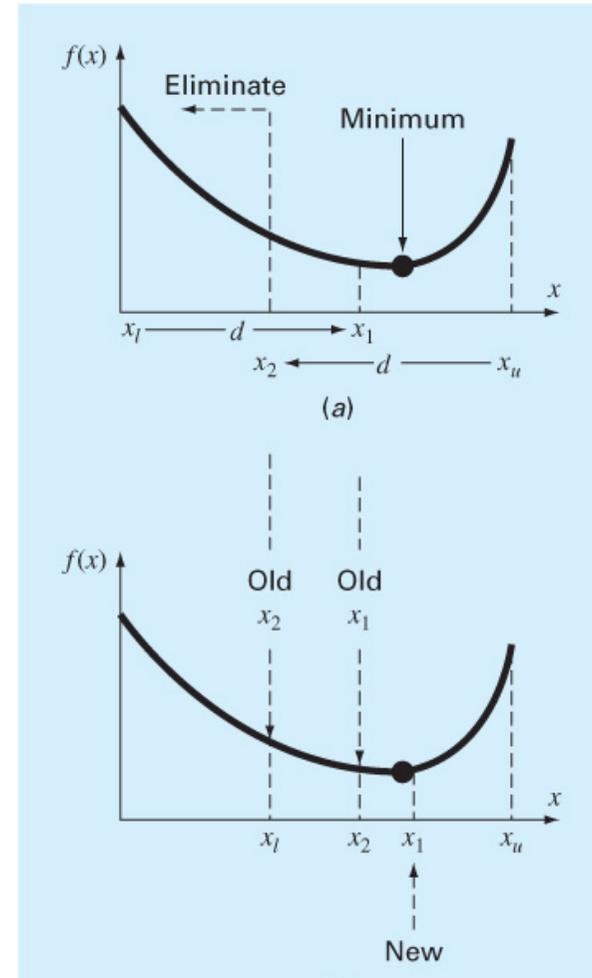
# Golden-Section Search (3/4)

$$x_1 = x_l + d \quad (7.6)$$

$$x_2 = x_u - d \quad (7.7)$$

$$d = (\phi - 1)(x_u - x_l) \quad (7.8)$$

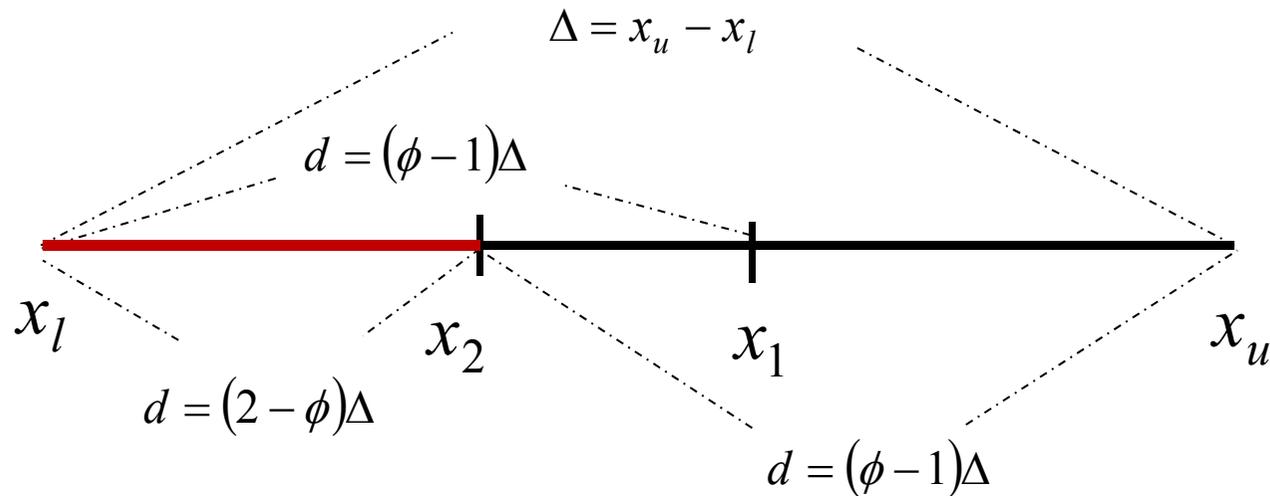
- If  $f(x_1) < f(x_2)$ ,  $x_2$  becomes the new lower limit and  $x_1$  becomes the new  $x_2$  (as in the right figure)
- If  $f(x_2) < f(x_1)$ ,  $x_1$  becomes the new upper limit and  $x_2$  becomes the new  $x_1$
- In either case, only one new interior point is needed and the function is only evaluated one more time



**FIGURE 7.6** (a) The initial step of the golden-section search algorithm involves choosing two interior points according to the golden ratio. (b) The second step involves defining a new interval that encompasses the optimum.

# Golden-Section Search (4/4)

If  $f(x_2) < f(x_1)$ ,  $x_1$  becomes the new upper limit and  $x_2$  becomes the new  $x_1$



To test in what condition,  
 $x_2$  would be the new  $x_1$

if  $\frac{(2 - \phi)\Delta}{(\phi - 1)\Delta} = \phi - 1$

then

$$(\phi - 1)^2 = 2 - \phi$$

$$\Rightarrow \phi^2 - \phi - 1 = 0$$

$$\phi = 1.6183$$

# Golden-Section Search: An Example (1/2)

## Golden-Section Search

**Problem Statement.** Use the golden-section search to find the minimum of

$$f(x) = \frac{x^2}{10} - 2 \sin x$$

Example 7.2

within the interval from  $x_l = 0$  to  $x_u = 4$ .

**Solution.** First, the golden ratio is used to create the two interior points:

$$d = 0.61803(4 - 0) = 2.4721$$

$$x_1 = 0 + 2.4721 = 2.4721$$

$$x_2 = 4 - 2.4721 = 1.5279$$

The function can be evaluated at the interior points:

$$f(x_2) = \frac{1.5279^2}{10} - 2 \sin(1.5279) = -1.7647$$

$$f(x_1) = \frac{2.4721^2}{10} - 2 \sin(2.4721) = -0.6300$$

Because  $f(x_2) < f(x_1)$ , our best estimate of the minimum at this point is that it is located at  $x = 1.5279$  with a value of  $f(x) = -1.7647$ . In addition, we also know that the minimum is in the interval defined by  $x_l$ ,  $x_2$ , and  $x_1$ . Thus, for the next iteration, the lower bound remains  $x_l = 0$ , and  $x_1$  becomes the upper bound, that is,  $x_u = 2.4721$ . In addition, the former  $x_2$  value becomes the new  $x_1$ , that is,  $x_1 = 1.5279$ . In addition, we do not have to recalculate  $f(x_1)$ , it was determined on the previous iteration as  $f(1.5279) = -1.7647$ .

All that remains is to use Eqs. (7.8) and (7.7) to compute the new value of  $d$  and  $x_2$ :

# Golden-Section Search: An Example (2/2)

All that remains is to use Eqs. (7.8) and (7.7) to compute the new value of  $d$  and  $x_2$ :

$$d = 0.61803(2.4721 - 0) = 1.5279$$

$$x_2 = 2.4721 - 1.5279 = 0.9443$$

The function evaluation at  $x_2$  is  $f(0.9443) = -1.5310$ . Since this value is less than the function value at  $x_1$ , the minimum is  $f(1.5279) = -1.7647$ , and it is in the interval prescribed by  $x_2$ ,  $x_1$ , and  $x_u$ . The process can be repeated, with the results tabulated here:

$i$	$x_l$	$f(x_l)$	$x_2$	$f(x_2)$	$x_1$	$f(x_1)$	$x_u$	$f(x_u)$	$d$
1	0	0	1.5279	-1.7647	2.4721	-0.6300	4.0000	3.1136	2.4721
2	0	0	0.9443	-1.5310	1.5279	-1.7647	2.4721	-0.6300	1.5279
3	0.9443	-1.5310	1.5279	-1.7647	1.8885	-1.5432	2.4721	-0.6300	0.9443
4	0.9443	-1.5310	1.3050	-1.7595	1.5279	-1.7647	1.8885	-1.5432	0.5836
5	1.3050	-1.7595	1.5279	-1.7647	1.6656	-1.7136	1.8885	-1.5432	0.3607
6	1.3050	-1.7595	1.4427	-1.7755	1.5279	-1.7647	1.6656	-1.7136	0.2229
7	1.3050	-1.7595	1.3901	-1.7742	1.4427	-1.7755	1.5279	-1.7647	0.1378
8	1.3901	-1.7742	1.4427	-1.7755	1.4752	-1.7732	1.5279	-1.7647	0.0851

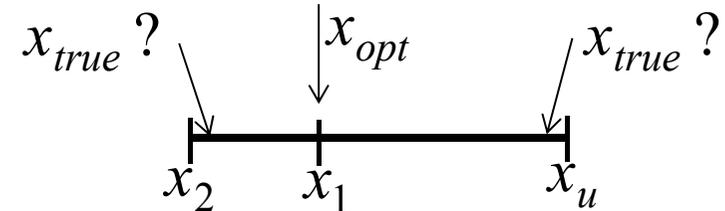
Note that the current minimum is highlighted for every iteration. After the eighth iteration, the minimum occurs at  $x = 1.4427$  with a function value of  $-1.7755$ . Thus, the result is converging on the true value of  $-1.7757$  at  $x = 1.4276$ .

**Example 7.2**

# Golden-Section Search: Error Estimate

Looking at the upper interval  $(x_2, x_1, x_u)$ , if the true value were at the far left, the maximum distance from the estimate would be

$$\begin{aligned} \Delta x_a &= x_1 - x_2 \\ &= \underline{x_l + (\phi - 1)(x_u - x_l) - x_u + (\phi - 1)(x_u - x_l)} \\ &= (x_l - x_u) + 2(\phi - 1)(x_u - x_l) \\ &= (2\phi - 3)(x_u - x_l) \end{aligned}$$



or  $0.2361(x_u - x_l)$ . If the true value were at the far right, the maximum distance from the estimate would be

$$\begin{aligned} \Delta x_b &= x_u - x_1 \\ &= \underline{x_u - x_l - (\phi - 1)(x_u - x_l)} \\ &= (x_u - x_l) - (\phi - 1)(x_u - x_l) \\ &= (2 - \phi)(x_u - x_l) \end{aligned}$$

or  $0.3820(x_u - x_l)$ . Therefore, this case would represent the maximum error. This result can then be normalized to the optimal value for that iteration  $x_{opt}$  to yield

$$\varepsilon_a = (2 - \phi) \left| \frac{x_u - x_l}{x_{opt}} \right| \times 100\% \quad (7.9)$$

This estimate provides a basis for terminating the iterations.

# Code for Golden-Section Search

```
function [x,fx,ea,iter]=goldmin(f,xl,xu,es,maxit,varargin)
% goldmin: minimization golden section search
% [xopt,fopt,ea,iter]=goldmin(f,xl,xu,es,maxit,p1,p2,...):
%     uses golden section search to find the minimum of f
% input:
% f = function handle
% xl, xu = lower and upper guesses
% es = desired relative error (default = 0.0001%)
% maxit = maximum allowable iterations (default = 50)
% p1,p2,... = additional parameters used by f
% output:
% x = location of minimum
% fx = minimum function value
% ea = approximate relative error (%)
% iter = number of iterations

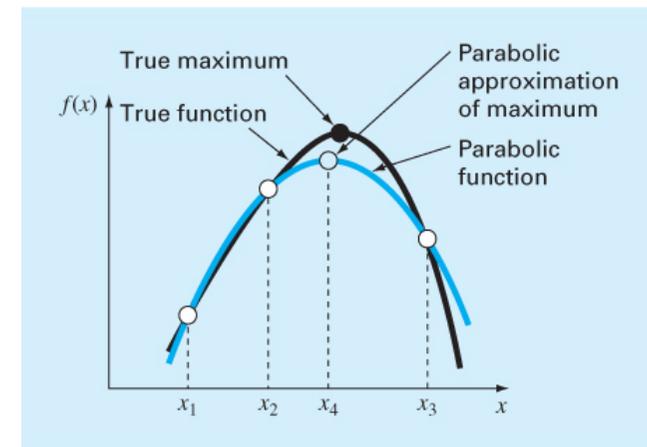
if nargin<3,error('at least 3 input arguments required'),end
if nargin<4||isempty(es), es=0.0001;end
if nargin<5||isempty(maxit), maxit=50;end
phi=(1+sqrt(5))/2;
iter=0;
while(1)
    d = (phi-1)*(xu - xl);
    x1 = xl + d;
    x2 = xu - d;
    if f(x1,varargin{:}) < f(x2,varargin{:})
        xopt = x1;
        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,varargin{:});
```

# Parabolic Interpolation

- Another algorithm uses **parabolic interpolation** of three points to estimate optimum location
  - The location of the **maximum/minimum** of a parabola defined as the interpolation of three points ( $x_1$ ,  $x_2$ , and  $x_3$ ) is:

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

- **By differentiating the parabola and set the result to zero to obtain  $x_4$**
- The new point  $x_4$  and the two surrounding it (either  $x_1$  and  $x_2$  or  $x_2$  and  $x_3$ ) are used for the next iteration of the algorithm



**FIGURE 7.8**

Graphical depiction of parabolic interpolation.

# Parabolic Interpolation: An Example (1/3)

## Example 7.3

### Parabolic Interpolation

**Problem Statement.** Use parabolic interpolation to approximate the minimum of

$$f(x) = \frac{x^2}{10} - 2 \sin x$$

with initial guesses of  $x_1 = 0$ ,  $x_2 = 1$ , and  $x_3 = 4$ .

**Solution.** The function values at the three guesses can be evaluated:

$x_1 = 0$	$f(x_1) = 0$
$x_2 = 1$	$f(x_2) = -1.5829$
$x_3 = 4$	$f(x_3) = 3.1136$

# Parabolic Interpolation: An Example (2/3)

and substituted into Eq. (7.10) to give

Example 7.3

$$x_4 = 1 - \frac{1}{2} \frac{(1-0)^2 [-1.5829 - 3.1136] - (1-4)^2 [-1.5829 - 0]}{(1-0)[-1.5829 - 3.1136] - (1-4)[-1.5829 - 0]} = 1.5055$$

which has a function value of  $f(1.5055) = -1.7691$ .

Next, a strategy similar to the golden-section search can be employed to determine which point should be discarded. Because the function value for the new point is lower than for the intermediate point ( $x_2$ ) and the new  $x$  value is to the right of the intermediate point, the lower guess ( $x_1$ ) is discarded. Therefore, for the next iteration:

$$\begin{array}{ll} x_1 = 1 & f(x_1) = -1.5829 \\ x_2 = 1.5055 & f(x_2) = -1.7691 \\ x_3 = 4 & f(x_3) = 3.1136 \end{array}$$

which can be substituted into Eq. (7.10) to give

$$\begin{aligned} x_4 &= 1.5055 - \frac{1}{2} \frac{(1.5055 - 1)^2 [-1.7691 - 3.1136] - (1.5055 - 4)^2 [-1.7691 - (-1.5829)]}{(1.5055 - 1)[-1.7691 - 3.1136] - (1.5055 - 4)[-1.7691 - (-1.5829)]} \\ &= 1.4903 \end{aligned}$$

which has a function value of  $f(1.4903) = -1.7714$ .

# Parabolic Interpolation: An Example (3/3)

- The process can be repeated with the results tabulated here:

$i$	$x_1$	$f(x_1)$	$x_2$	$f(x_2)$	$x_3$	$f(x_3)$	$x_4$	$f(x_4)$
1	0.0000	0.0000	1.0000	-1.5829	4.0000	3.1136	1.5055	-1.7691
2	1.0000	-1.5829	1.5055	-1.7691	4.0000	3.1136	1.4903	-1.7714
3	1.0000	-1.5829	1.4903	-1.7714	1.5055	-1.7691	1.4256	-1.7757
4	1.0000	-1.5829	1.4256	-1.7757	1.4903	-1.7714	1.4266	-1.7757
5	1.4256	-1.7757	1.4266	-1.7757	1.4903	-1.7714	1.4275	-1.7757

Thus, within five iterations, the result is converging rapidly on the true value of  $-1.7757$  at  $x = 1.4276$ .

Example 7.3

# fminbnd Function

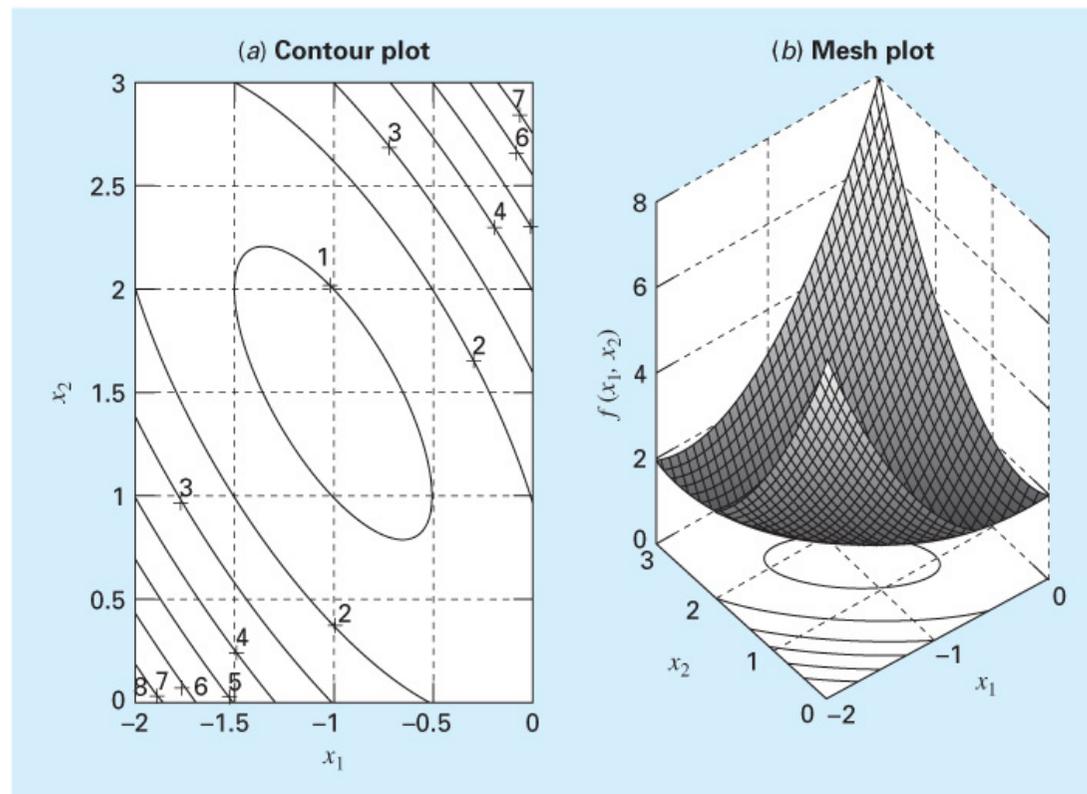
- MATLAB has a built-in function, `fminbnd`, which combines the golden-section search and the parabolic interpolation
  - `[xmin, fval] = fminbnd(function, x1, x2)`
- Options may be passed through a fourth argument using `optimset`, similar to `fzero`

# Multidimensional Visualization

- Functions of two-dimensions may be visualized using contour or surface/mesh plots

**FIGURE 7.9**

(a) Contour and (b) mesh plots of a two-dimensional function.



## `fminsearch` Function (2/2)

- MATLAB has a built-in function, `fminsearch`, that can be used to determine the minimum of a multidimensional function
  - `[xmin, fval] = fminsearch(function, x0)`
  - `xmin` in this case will be a row vector containing the location of the minimum, while `x0` is an initial guess. Note that `x0` must contain as many entries as the function expects of it
- The `function` must be written in terms of a single variable, where different dimensions are represented by different indices of that variable

## fminsearch Function (2/2)

- To minimize  
 $f(x,y)=2+x-y+2x^2+2xy+y^2$   
rewrite as  
 $f(x_1, x_2)=2+x_1-x_2+2(x_1)^2+2x_1x_2+(x_2)^2$ 
  - `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])`
- Note that  $x_0$  has two entries -  $f$  is expecting it to contain two values
- MATLAB reports the minimum value is 0.7500 at a location of [-1.000 1.5000]