Optimization
الامثلية

ا.م اسيل مؤيد قاسم

م. هبة شكرالبنى

lec1(iteration method):

Iteration methods

We using iteration methods because we either cannot the procedure analytically or because the analytical methods is intractable.

Iteration Methods of One Dimension

We sometimes need to determine the maximum or minimum value of a one-variable nonlinear function.

The solution techniques for one-dimensional optimization problems be the derivative of the function is driven to zero. The algorithm is terminated when the derivative of the function is very close to zero.

We discussing the different algorithms (approaches) for finding the minimum of a function of one variable. These algorithms are follows :

lec1(iteration method)::

1. Bisection method

Idea give an interval $[a, b]$ use $f' = 0$ to find a local minimizer.

The out line of the technique is as follows :

1. Set two points a, b such that $f'(a)$, $f'(b)$ are of opposite sign.
2. Find $c = \frac{a+b}{2}$, and calculate $f'(c)$.
3. If $f'(a) \cdot f'(b) < 0$, then $b = c$ else $a = c$.
4. The iteration is terminated if $|a - b| < \varepsilon$ or $f'(c) < \varepsilon$, then c is minimum, otherwise go to 2.

lec1(iteration method):

2. Newton's method

Aim, minimize a function $f(x)$ of a single real variable x. Use Newton's method to solve $f^{/}(x)=0$.

The out line of the technique is as follows :

1. Set x_0 , ε; where ε is smallest number.
2. Find a point $x_1 = x_0 - \dfrac{f^{/}(x_0)}{f^{//}(x_0)}$.
3. If $|x_1 - x_0| < \varepsilon$, then x_1 is the minimum else set $x_0 = x_1$, go to 2.

Example:

The function :

$$f(x) = x^2 + 54/x \ .$$

Using Newton's method, with the starting guess $x_0 = 1$.

lec1(iteration method):

Solution:

The first and second derivatives of function are:

$$f'(x) = 2x - 54x^{-2} \quad \text{and} \quad f''(x) = 2 + 108x^{-3}$$

Evaluate $f'(x)$ and $f''(x)$ at the point $x_0 = 1$, such that :

$$f'(1) = -52 \quad \text{and} \quad f''(1) = 110$$

We compute the next guess,

$$x_1 = x_0 - \frac{f'(x_0)}{f''(x_0)} = 1 - \frac{-52}{110} = 1.4727$$

The derivative computed using (1.4727) at this point is found to be $f'(1.4727) = -21.9526$.

Since $|f'(1.4727)| = |-21.9526|$ not less than ε , we increment k to 2 and go to Step 2. This completes one iteration of the Newton Raphson method.

The Second Iteration needed to : $f''(1.4727) = 35.8128$

$$x_2 = x_1 - \frac{f'(x_1)}{f''(x_1)} = 1.4727 - \frac{-21.9526}{35.8128} = 1.4727 + 0.6130 = 2.0857$$

Since at every iteration first and second order derivatives are evaluated, a total of three function values are evaluated for every iteration.

lec2(unconstrained optimization):

2. Unconstrained optimization in multi dimensions

Given a function $f : R^n \to R$, we seek to find a minimum of $f(x)$, i.e., we solve the following unconstrained optimization problem :

find x to minimize $f(x)$.

First Derivative or Gradient.

Let $f : R^n \to R$ be a function with continuous derivatives. The gradient of $f(x)$ is defined as the column vector containing the first order partial derivatives of $f(x)$:

$$g(x) = \nabla f(x) = \left[\frac{\partial f(x)}{\partial x_1} \ \frac{\partial f(x)}{\partial x_2} \ \ldots\ldots\ldots\ldots\ldots\ldots \frac{\partial f(x)}{\partial x_n} \right]^T . \qquad \ldots\ldots\ldots\ldots(2)$$

lec2(unconstrained optimization):

Second Derivative or Hessian.

The Hessian $\nabla^2 f(x)$ of $f(x)$, is the matrix defined by the second order partial derivatives of $f(x)$, as :

$$G = \nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots\cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial^2 x_2} & \cdots\cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ . & . & \cdots\cdots & . \\ . & . & \cdots\cdots & . \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots\cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix} \quad \text{..........(2)}$$

lec2(unconstrained optimization):

Taylor's Series in n Dimensions.

The Taylor series expansion of $f(x)$ about some $x_k \in R^n$ is :

$$f(x) = f(x_k) + g_k^T(x - x_k) + \frac{1}{2}(x - x_k)^T G_k (x - x_k) + \ldots\ldots$$

where $g_k \in R^n$ and $G_k \in R^{n*n}$.

In multiple dimensions, the conditions are simply the multivariate extensions of the one dimension conditions.

The following theorem can be proved for functions of multivariables.

Theorem (A First-order Necessary conditions)

Let $f : R^n \to R$, have continuous first order partial derivatives. If x^* is a local minimum of $f(x)$, then $f'(x^*) = 0$.

We call a point x^* as a stationary point of $f(x)$, if $f'(x^*) = 0$.

lec2(unconstrained optimization):

3. Classification of Matrices.

Before we prove a second order sufficient condition for the local minimums, let us first review the positive definiteness of a matrix. We say that a matrix A is symmetric positive definite, if

$$x^T\left[\nabla^2 f(x)\right] x > 0.$$

For any vector $x \neq 0$.

For the next result we recall that a matrix $\nabla^2 f(x)$ is positive definite if $x^T\left[\nabla^2 f(x)\right] x > 0$ for all $x \neq 0$, and positive semi definite if $x^T\left[\nabla^2 f(x)\right] x \geq 0$ for all $x \neq 0$.

Example : Let

$$G(x) = \nabla^2 f(x) = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$$

Then for any $x = (x_1 , x_2)^T$.

$$x^T G(x) x = (x_1 , x_2)\begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 2x_1^2 - 2x_1x_2 + x_2^2 = x_1^2 + (x_1 - x_2)^2 \geq 0$$

Thus, G is positive semi definite.

lec2(unconstrained optimization):

Theorem (A second-order sufficient conditions)

Let $f : R^n \to R$, have continuous first and second order partial derivatives. If $f^{/}(x^*) = 0$ and $\nabla^2 f(x)$ is positive definite, then x^* is a local minimum of the $f(x)$.

Example :

Find the critical point of the function $f(x_1, x_2) = x_1^3 + 3x_2 - x_2^3 - 3x_1$.

Solution :

Critical points are **:**

$$\frac{\partial f}{\partial x_1} = 3x_1^2 - 3 = 0 \quad \Rightarrow \quad x = \pm 1$$

$$\frac{\partial f}{\partial x_2} = 3 - 3x_2^2 = 0 \quad \Rightarrow \quad x = \pm 1$$

For all four point :

lec2(unconstrained optimization):

$$(1,1),\quad (1,-1),\quad (-1,1),\quad (-1,-1)$$

The second-order partial derivatives of $f(x)$ are given by :

$$\frac{\partial^2 f}{\partial x_1^2} = 6x_1 \quad , \quad \frac{\partial^2 f}{\partial x_1 \partial x_2} = 0 \quad , \quad \frac{\partial^2 f}{\partial x_2^2} = -6x_2$$

The Hessian matrix of f is given by :

$$G = \begin{bmatrix} 6x_1 & 0 \\ 0 & -6x_2 \end{bmatrix}$$

Evaluating at :

Point X	**Value of G_1**	**Value of G_2**	**Nature of G**	**Nature of X**
$(1,1)$	$+6$	-36	Indefinite	Saddle point
$(1,-1)$	$+6$	$+36$	Positive definite	minimum
$(-1,1)$	-6	$+36$	Negative definite	maximum
$(-1,-1)$	-6	-36	Indefinite	Saddle point

lec3(secant method):

5.8 Secant Method

In the secant method, magnitude and sign of derivatives are used to generate a new point. If $f'(x_1)$ and $f'(x_2)$ are the derivatives of a function computed at points x_1 and x_2, respectively, then the minimizer of the function lies between x_1 and x_2 if

$$f'(x_1)f'(x_2) < 0.$$

It is assumed that $f'(x)$ varies linearly between points x_1 and x_2. A line drawn between these two points is called secant line which crosses the point α on the x-axes is the improved point in the next iteration, and one of the points either x_1 or x_2 is eliminated. Thus, either interval $[x_1, \alpha]$ or $[\alpha, x_2]$ is retained for generating new point in next iteration. The iteration continues until the derivative of function $f(x)$ is close to zero at point α which is given as

$$\alpha = x_2 - \frac{f'(x_2)(x_2 - x_1)}{f'(x_2) - f'(x_1)}. \tag{5.29}$$

The iterative process can be implemented in the form of following algorithm:

lec3(secant method):

Algorithm 5.7 (Secant Algorithm)

1: Choose $a, b \in \mathbb{R}, \varepsilon$ ▹ a and b are lower and upper bound, ε is a tolerance value
2: Set $flag \leftarrow 0$
3: Compute $\alpha \leftarrow \frac{a+b}{2}$ ▹ mid point of a and b
4: Compute $f'(a)$, $f'(\alpha)$ ▹ derivatives of f at α
5: **if** $f'(a)f'(\alpha) < 0$ **then**
6: set $b \leftarrow \alpha$
7: set $flag \leftarrow 1$ ▹ zero is bracketed
8: **else**
9: Set $a \leftarrow \alpha$
10: **if** $flag == 1$ **then**
11: goto Step 16
12: **else**
13: goto Step 3
14: **end if**
15: **end if**
16: Compute $\alpha \leftarrow b - \frac{f'(b)(b-a)}{f'(b)-f'(a)}$ ▹ a new point
17: **if** $f'(\alpha) > 0$ **then**
18: Set $b \leftarrow \alpha$
19: **else**
20: Set $a \leftarrow \alpha$
21: **end if**
22: **if** $|f'(\alpha)| < \varepsilon$ **then** ▹ stopping criterion
23: goto Step 26
24: **else**
25: got to Step 16
26: Print $x^* \leftarrow \alpha$, $f(x^*) \leftarrow f(\alpha)$. ▹ best solution
27: **end if**

lec3(secant method):

Example 5.13 Apply the Secant method to minimize the function $f(x) = 2x^2 + \frac{16}{x}$ over the interval [1,5].

We find $f'(x) = 4x - \frac{16}{x^2}$. Given that $a = 1, b = 5$. We find $f'(a) = -12$, $f'(b) = 19.36$. We compute $\alpha = 5 - \frac{19.36}{(19.36+12)/4} = 2.53$. We find $f'(\alpha) = 7.62 > 0$, thus set $b = 2.53$ and move to the next iteration. We again find $\alpha = 2.53 - \frac{7.62}{(7.62+12)/1.53} = 1.94$, thus $f'(\alpha) = 3.51 > 0$ and set $b = 1.94$. We continue this process until $|f'(\alpha)| < \varepsilon$ where ε is a tolerance value.

The R function `Secant(a, b, epsilon, maxiter)` is written to implement the Secant algorithm.

lec3(secant method):

Example 5.14 Apply the secant method to minimize the function $f(x) = \log(\cos x^{\cos x} + 1)$ over the interval $[0, \frac{\pi}{2}]$ using R.

The R function `func_var(x)` is used to write the above function as

```
func_var<-function(x)
{
 fx=log10(cos(x)^cos(x)+1)
 return(fx)
}
```

On executing the R function `Secant(a, b, epsilon, maxiter)` with `a = 0, b = pi/2, epsilon = 1e-7` and `maxiter = 100` the following output is displayed on the R Console as (Fig. 5.17):

```
> Secant(a = 0, b = pi/2, epsilon = 1e-7, maxiter = 100)

k        alpha        f(alpha)       Deriv
0        1.3744       0.2373          0.1138
1        1.1886       0.2285         -0.002223
2        1.1922       0.2285         -0.0007564
3        1.1934       0.2285         -0.0002563
4        1.1938       0.2285         -8.67e-05
5        1.1939       0.2285         -2.932e-05
6        1.1940       0.2285         -9.913e-06
7        1.1940       0.2285         -3.351e-06
8        1.1940       0.2285         -1.133e-06
```

lec3(secant method):

```
{
 fx=log10(cos(x)^cos(x)+1)
 return(fx)
}
```

On executing the R function `Secant(a, b, epsilon, maxiter)` with a = 0, b = pi/2, epsilon = 1e-7 and maxiter = 100 the following output is displayed on the R Console as (Fig. 5.17):

```
> Secant(a = 0, b = pi/2, epsilon = 1e-7, maxiter = 100)

k          alpha         f(alpha)       Deriv
0          1.3744        0.2373          0.1138
1          1.1886        0.2285        -0.002223
2          1.1922        0.2285        -0.0007564
3          1.1934        0.2285        -0.0002563
4          1.1938        0.2285        -8.67e-05
5          1.1939        0.2285        -2.932e-05
6          1.1940        0.2285        -9.913e-06
7          1.1940        0.2285        -3.351e-06
8          1.1940        0.2285        -1.133e-06
```

lec4 (Golden section):

Golden Section Search Method

Golden Section Search method is the improved version of the Fibonacci search method. The major difficulty in the Fibonacci search method is that the Fibonacci numbers have to be calculated and stored. The proportion of the eliminated region is different at every iteration. To overcome these problems, the Golden section search method is used. In this method, we choose the intermediate points in such a way that the reduction in the range is symmetric in the sense that

lec4 (Golden section):

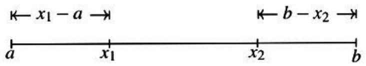

Fig. 5.11 Evaluating objective function at two intermediate points

$$x_1 - a = b - x_2 = r(b - a), \quad \text{where } r < \frac{1}{2}, \tag{5.11}$$

which can be depicted in Fig. 5.11. We then evaluate $f(x)$ at two intermediate points x_1 and x_2. If $f(x_1) < f(x_2)$, then the minimizer x^* must lie in the range $[a, x_2]$. If on the other hand, $f(x_1) \geq f(x_2)$, then the minimizer is located in the range $[x_1, b]$. Starting with the reduced range of uncertainty, we can repeat the process and similarly find two new points, say x_3, and x_4 using the same value of $r < \frac{1}{2}$. However, we want to minimize the number of function evaluations while reducing the width of the uncertainty interval. Suppose for example, if we have $f(x_1) < f(x_2)$, then we get the minimizer x^* lies in interval $[a, x_2]$. Since x_1 coincides with x_4. Thus, only new evaluations of f at x_3 would be necessary. Without loss of generality, imagine that the original range $[a, b]$ is of unit length. Then to find one new evaluation of f it is enough to choose r such that

$$r(x_2 - a) = x_2 - x_4.$$

Since $x_2 - a = 1 - r$ and $x_2 - x_4 = 1 - 2r$, then

$$r(1 - r) = 1 - 2r,$$

which provides a quadratic equation as

$$r^2 - 3r + 1 = 0. \tag{5.12}$$

The solutions are

$$r_1 = \frac{3 + \sqrt{5}}{2}, \quad r_2 = \frac{3 - \sqrt{5}}{2}.$$

Because we require that $r < \frac{1}{2}$, we take

$$r = \frac{3 - \sqrt{5}}{2} \approx 0.3819.$$

Observe that

$$1 - r = \frac{\sqrt{5} - 1}{2}, \tag{5.13}$$

and

$$\frac{r}{1 - r} = \frac{3 - \sqrt{5}}{\sqrt{5} - 1} = \frac{1 - r}{1},$$

that is,

$$\frac{r}{1 - r} = \frac{1 - r}{1}. \tag{5.14}$$

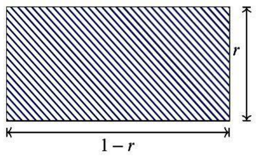

Fig. 5.12 Golden Section design

The dividing in the range of ratio of r and $1 - r$ has the effect that the ratio of the shorter segment to the longer equals the ratio of the longer to the sum of the two. This rule was referred as the golden section by ancient Greek geometers. There is an interesting historical note on the golden section. The ratio was known to the ancients as the golden section ratio. Some of the early Greek architecture was designed with the ratio of the shorter segment to the longer equals to the ratio of the longer to the sum of the two as shown in Fig. 5.12.

The uncertainty range is reduced by the ratio $1 - r \approx 0.61803$ at every stage. After the second iteration, it is

$$(1 - r) - r(1 - r) = (1 - r)^2.$$

Hence, N steps of reduction using the golden section method reduces the range by the factor

$$(1 - r)^N \approx (0.61803)^N.$$

The following golden section algorithm is presented on the basis of above principles.

Algorithm 5.3 (Golden Section Search Algorithm)

```
Choose a, b ∈ ℝ, 0 < ε << 1          ▹ a and b are lower and upper bound, ε is tolerance
Set r ← (3−√5)/2                      ▹ r is a golden ratio
Compute x1 ← a + r(b − a) and x2 ← b − r(b − a)
Compute f1 ← f(x1) and f2 ← f(x2)
k ← 0                                 ▹ k is the number of iterations of method
for k to maxiter do                   ▹ maxiter is number of iterations to perform
    if f1 > f2 then
        Set a ← x1, x1 ← x2, f1 = f2
        Set x2 ← ra + (1 − r)b, f2 = f(x2)
    else
        Set b ← x2 x2 ← x1 and f2 = f1
        Set x1 ← rb + (1 − r)a, and f1 = f(x1)
    end if
    if |f1 − f2| < ε then             ▹ stopping criterion
        Converged. Print x1 and f(x1) ▹ best solution
    end if
end for
```

lec4 (Golden section):

Example 5.5 Apply the golden Section Search method to minimize the function $f(x) = x^2 - 3x + 1$ in the interval $[0, 2]$. We wish to locate this value of x within a range of 0.3.

Given that $a = 0, b = 2$. We compute the values of x_1 and x_2 as below:

$$x_1 = (1-r)a + rb = (1-r) \times 0 + 0.3819 \times (1 - 0.3819) = 0.7639,$$
$$x_2 = ra + (1-r)b = 0.3819 \times 0 + (1 - 0.3819) \times 2 = 1.236066$$

with corresponding values $f(x_1) = -0.70820$ and $f(x_2) = -1.180339$.
Since $f(x_1) > f(x_2)$, thus the uncertainty interval is reduced to $[0.7639 , 2]$. We set

$$\begin{aligned} a &= x_1 = 0.7639, \\ b &= 2, \\ x_1 &= x_2 = 1.236066, \\ x_2 &= ra + (1-r)b = 0.3819 \times 0.7639 + (1 - .3819) \times 2 = 1.527864, \\ f(x_1) &= f(x_2) = -1.180339, \\ f(x_2) &= -1.249224. \end{aligned}$$

Since $f(x_1) > f(x_2)$, thus the uncertainty interval is reduced to $[1.236066 , 2]$. We set

$$\begin{aligned} a &= x_1 = 1.236066, \\ b &= 2, \\ x_1 &= x_2 = 1.527864, \\ x_2 &= ra + (1-r)b = 0.3819 \times 1.236066 + (1 - 0.3819) \times 2 = 1.708202, \\ f(x_1) &= f(x_2) = -1.249224, \\ f(x_2) &= -1.206652. \end{aligned}$$

Since $f(x_1) < f(x_2)$, thus the uncertainty interval is reduced to $[1.236066, 1.708202]$. We set

$$\begin{aligned} b &= x_2 = 1.708202, \\ x_2 &= x_1 = 1.527864, \\ f(x_2) &= f(x_1) = -1.249224, \\ x_1 &= ra + (1-r)b = 0.3819 \times 1.236066 + (1 - 0.3819) \times 1.708202 = 1.416407 \\ f(x_1) &= -1.243012. \end{aligned}$$

Since $f(x_1) > f(x_2)$, thus the uncertainty interval is reduced to $[1.416407, 1.708202]$. We set

$$\begin{aligned} a &= x_1 = 1.416407, \\ b &= 1.708202, \\ x_1 &= x_2 = 1.527864, \\ f(x_1) &= f(x_2) = -1.249224, \\ x_2 &= ra + (1-r)b = 0.3819 \times 1.236066 + (1 - 0.3819) \times 2 = 1.596746, \\ f(x_2) &= -1.24064. \end{aligned}$$

Since $f(x_1) < f(x_2)$, thus the uncertainty interval is reduced to $[1.416407 , 1.596746]$. We set

$$\begin{aligned} b &= x_2 = 1.596746, \\ x_2 &= x_1 = 1.527864, \\ f(x_2) &= f(x_1) = -1.249224, \\ x_1 &= ra + (1-r)b = 0.3819 \times 1.416407 + (1 - 0.3819) \times 1.596746 = 1.48529, \\ f(x_1) &= -1.249784. \end{aligned}$$

Since $f(x_1) < f(x_2)$, then value x that minimizes f is located in the interval $[1.416407 , 1.527864]$. We get minimum point 1.4992 and minimum function value -1.2500 in 15 iterations.
The R function **Golden_Section_Search(a, b, epsilon, maxiter)** for the Golden Section Search algorithm is given below in Function 5.3.

R Function 5.3 Golden_Section_Search.R

```
# Description            : Golden Section Search Method
# Theory                 : See Section 5.4 and Algorithm 5.3
# Function Called        : func_var(x)
# Input: a               : Lower bound
#        b               : Upper bound
#        epsilon         : Tolerance Value
#        maxiter         : Maximum number of itertions
# Output:xm              : minimum point
#        fxm             : minimum function value
# Execute on R Console   : Golden_Section_search(a,b,1e-7,100)
Golden_Section_Search<-function(a, b, epsilon, maxiter)
  {
  tau <- 0.381967
  x1 <- a*(1-tau) + b*tau
  x2 <- a*tau + b*(1-tau)
  fx1 <- func_var(x1)
  fx2 <- func_var(x2)
  k=0
  cat('k    a            b            f(x1)          f(x2)    \n')
  cat(k,'\t', format(a,digits=4, nsmall=4),'\t',format(b,digits
      =4, nsmall=4),'\t',format(fx1,digits=4, nsmall=4),'\t',
      format(fx2,digits=4, nsmall=4),'\n')
  for (k in 1:maxiter)
    {
    if (fx1 > fx2)
      {
```

lec4 (Golden section):

```
        a <- x1
        x1 <- x2
        fx1 <- fx2
        x2 <- tau*a + (1-tau)*b
        fx2 <- func_var(x2)
        }
     else
        {
          b <- x2
          x2 <- x1
          fx2 <- fx1
          x1 <- tau*b + (1-tau)*a
          fx1 <- func_var(x1)
        }
     cat(k,'\t',format(a,digits=4, nsmall=4),'\t',format(b,
          digits=4, nsmall=4),'\t',format(fx1,digits=4, nsmall=4)
          ,'\t',format(fx2,digits=4, nsmall=4),'\n')
     if (abs(func_var(x1)-func_var(x2))< epsilon)
        {
        break
     }
  }
  xm=a
  fxm=func_var(xm)
  cat('\nMinimum point: ',format(xm,digits=4, nsmall=4))
  cat('\nMinimum function value:',format(fxm,digits=4, nsmall
      =4))
}
```

Example 5.6 Apply the golden section search method to minimize the function f(x)=$x^3 + 5x^2 + 4x + 6$ on the interval [-2, 2] using R.

The R function **func_var(x)** is used to write the above function as

```
func_var<-function(x)
{
 fx=x^3+5*x^2+4*x+6
 return(fx)
}
```

On executing function **Golden_Section_Search(a, b, epsilon, maxiter)** with a = -2, b = 2, epsilon = 1e-7 and maxiter = 100 the following output is displayed on the R Console as

```
> Golden_Section_Search(a = -2, b = 2, epsilon = 1e-7,
maxiter = 100)

k    a          b          f(x1)      f(x2)
0    -2.0000    2.0000     5.1208     9.1083
1    -2.0000    0.4721     6.1732     5.1208
2    -1.0557    0.4721     5.1208     5.6149
3    -1.0557    -0.1115    5.2995     5.1208
4    -0.6950    -0.1115    5.1208     5.1842
5    -0.6950    -0.3344    5.1506     5.1208
```

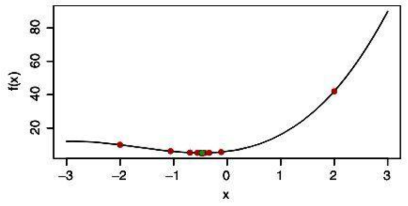

Fig. 5.13 Graph of $f(x) = x^3 + 5x^2 + 4x + 6$

```
6    -0.5573    -0.3344    5.1208    5.1281
7    -0.5573    -0.4195    5.1262    5.1208
8    -0.5047    -0.4195    5.1208    5.1212
9    -0.5047    -0.4520    5.1220    5.1208
10   -0.4846    -0.4520    5.1208    5.1206
11   -0.4721    -0.4520    5.1206    5.1207
12   -0.4721    -0.4597    5.1206    5.1206
13   -0.4674    -0.4597    5.1206    5.1206
14   -0.4674    -0.4626    5.1206    5.1206
15   -0.4656    -0.4626    5.1206    5.1206
16   -0.4656    -0.4638    5.1206    5.1206
17   -0.4656    -0.4645    5.1206    5.1206
18   -0.4652    -0.4645    5.1206    5.1206

Minimum point:  -0.4652
Minimum function value: 5.1206
```

See Fig. 5.13.

There is another class to find the minimum of a unimodal function, which locates a point x near x^* by interpolating using polynomial approximation as models of $f(x)$. This x is the value of the independent variable that corresponds to the minimum of $f(x)$. The method discussed here uses function and derivative values.

lec5(Fibonacci search method):

5.3 Fibonacci Search Method

This search technique determines the smallest possible interval of uncertainty, in which a minimum lies after completing n experiments. We have seen that the interval halving method reduces the interval size to one-half of its size in every iteration. Fibonacci search method, on the other hand, reduces the interval size by a different factor at each iteration. The reduction follows a sequence number satisfying the following Fibonacci formula:

$$F_n = F_{n-1} + F_{n-2}, \tag{5.2}$$

where $n = 2, 3, \ldots$ with

$$F_0 = F_1 = 1.$$

Some numbers in the Fibonacci sequence are given in Table 5.1. Note that the Fibonacci method can also be applied even if the function is not continuous. Consider a unimodal function f of one variable and define the interval $[a, \ b]$. We choose points a and b in such a way that the minimizer x^* of f may be achieved in the few function

lec5(Fibonacci search method):

Table 5.1 Fibonacci numbers

n	F_n
0	1
1	1
2	2
3	3
4	5
5	8
6	13
7	21
8	34
9	55
10	89

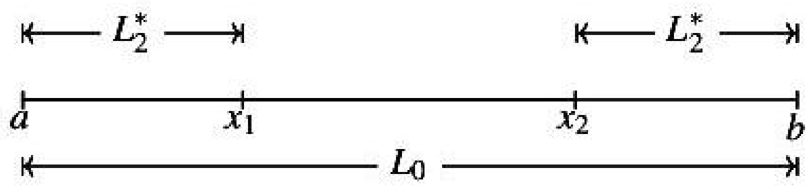

Fig. 5.4 Fibonacci search in uncertanity interval L_0

lec5(Fibonacci search method):

evaluations. Let L_0 be the initial interval of uncertainty defined by $a \le x \le b$ and n be the number of experiments to be performed. We present

$$L_2^* = \frac{F_{n-2}}{F_n} L_0$$

to determine x_1 and x_2 for the first two experiments. These points are placed at a distance of L_2^* from each end of L_0, which can be shown in Fig. 5.4 and expressed in the following form:

$$x_1 = a + L_2^* = a + \frac{F_{n-2}}{F_n} L_0, \tag{5.3}$$

and

$$x_2 = b - L_2^* = b - \frac{F_{n-2}}{F_n} L_0.$$

The length of initial interval of uncertainty is $L_0 = b - a$, therefore

$$x_2 = b - \frac{F_{n-2}}{F_n}(b - a) = \left(1 - \frac{F_{n-2}}{F_n}\right) b + \frac{F_{n-2}}{F_n} a = \left(\frac{F_n - F_{n-2}}{F_n}\right) b + \frac{F_{n-2}}{F_n} a.$$

Since $F_{n-1} = F_n - F_{n-2}$, therefore

lec5(Fibonacci search method):

$$x_2 = \frac{F_{n-1}}{F_n} b + \frac{F_{n-2}}{F_n} a. \tag{5.4}$$

Similarly, we can find the value of x_1. We find the value of $f(x_1)$ and $f(x_2)$. After the first iteration, either $[a, x_1]$ or $[x_2, b]$ will be eliminated by using the assumption of unimodality. In either case new interval of uncertainty of length L_2 is a smaller interval of uncertainty which is given as:

$$L_2 = L_0 - L_2^* = L_0 - \frac{F_{n-2}}{F_n} L_0 = \frac{F_n - F_{n-2}}{F_n} L_0.$$

Since $F_n - F_{n-2} = F_{n-1}$, then

$$L_2 = \frac{F_{n-1}}{F_n} L_0, \tag{5.5}$$

and one experiment left. This experiment will be at a distance of

$$L_2^* = \frac{F_{n-2}}{F_n} L_0.$$

lec5(Fibonacci search method):

F_n

from each end of the interval L_2. Assume the unimodality property and discard a portion of interval L_2, and obtain L_3 as

$$L_3 = L_2 - L_3^* = L_2 - \frac{F_{n-3}}{F_n} L_0.$$

Since $L_0 = \frac{F_n}{F_{n-1}} L_2$, then

$$L_3 = L_2 - \frac{F_{n-3}}{F_n} \times \frac{F_n}{F_{n-1}} L_2 = L_2 - \frac{F_{n-3}}{F_{n-1}} L_2 = \frac{F_{n-1} - F_{n-3}}{F_{n-1}} L_2.$$

Since $F_{n-1} - F_{n-3} = F_{n-2}$, then we get

$$L_3 = \frac{F_{n-2}}{F_{n-1}} L_2.$$

lec5(Fibonacci search method):

Algorithm 5.2 (Fibonacci Search Algorithm)

Choose $a, b \in \mathbb{R}, n \in \mathbb{Z}_+$ ▷ a and b are lower and upper bound, n is total number of experiments to conduct
Set $F_0 \leftarrow 1$ and $F_1 \leftarrow 1$ ▷ first two Fibonacci numbers
$i \leftarrow 2$
for i to n **do** ▷ generate sequence of Fibonacci numbers
  $F_i \leftarrow F_{i-1} + F_{i-2}$
end for
Compute $L_0 \leftarrow b - a$ ▷ length of initial interval of uncertainty $[a, b]$
Compute $L_2^* \leftarrow \frac{F_{n-2}}{F_n} L_0$.
$j \leftarrow 2$.
for j to n **do**
  if $L_j^* > \frac{L_0}{2}$ **then** ▷ comparison is made to ensure x_1 lies to the left of x_2
    Set $x_1 \leftarrow b - L_j^*,\ x_2 \leftarrow a + L_j^*$.
  else
    Set $x_1 \leftarrow a + L_j^*, x_2 \leftarrow b - L_j^*$
  end if
  Compute $f_1 \leftarrow f(x_1)$, and $f_2 \leftarrow f(x_2)$
  if $f_1 < f_2$ **then** ▷ compare f_1 with f_2
    Set $b \leftarrow x_2$
    Compute $L_j^* \leftarrow \frac{F_{n-j}}{F_{n-(j-2)}} \times L_0$ ▷ new interval
  else
    if $f_2 < f_1$ **then** ▷ compare f_2 with f_1
      Set $a \leftarrow x_1$.
      $L_j^* \leftarrow \frac{F_{n-j}}{F_{n-(j-2)}} \times L_0$. ▷ new interval
    else
      Set $a \leftarrow x_1$ and $b \leftarrow x_2$.
      Compute $L_j^* \leftarrow \frac{F_{n-j}}{F_{n-(j-2)}} \times (b - a)$ ▷ new interval
    end if
  end if
end for
if $f_1 \leq f_2$ **then** ▷ stopping criterion
  Print $x^* \leftarrow x_1,\ f(x^*) \leftarrow f(x_1)$ ▷ best solution
else
  Print $x^* \leftarrow x_2,\ f(x^*) \leftarrow f(x_2)$ ▷ best solution
end if

lec5(Fibonacci search method):

Example 5.3 Apply the Fibonacci search method to minimize $f(x) = x^2 + 2x$ on the interval $[-3\ ,\ 4]$. Obtain the optimal value within 5% exact value.

We have

$$\frac{\text{Length of final interval of uncertainty}}{2 \times \text{Length of initial interval of uncertainty}} \leq \frac{5}{100},$$

that is,

$$\frac{L_n}{2} \leq \frac{1}{20} L_0,$$