Newton method

Kazufumi OZAWA

1 Fixed-point iteration

Consider the sequence $\{x_k\}$ given by

$$x_{k+1} = F(x_k), \qquad k = 0, 1, \ldots, \tag{1}$$

where $F : \mathbb{R} \to \mathbb{R}$. This iteration proceeds as in Figure 1. For the convergence of $\{x_k\}$ generated by (1), we have the following theorem (the fixed-point theorem):

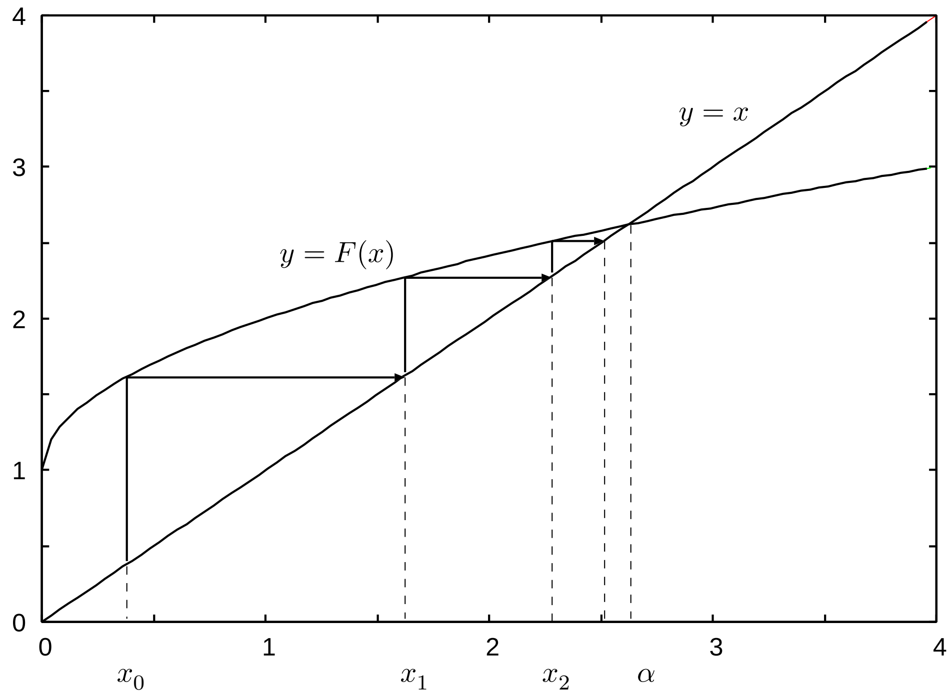

Figure 1: Fixed-point iteration

Theorem 1 *Let us assume that the function F satisfies the following conditions:*

1. *$F(x)$ is contiguous on the interval $I = [a, b]$.*
2. *For all $x \in I$, $F(x) \in I$.*
3. *For any x, $y \in I$, there exists a constant $0 \le L < 1$, such that*

   $$|F(x) - F(y)| < L\,|x - y|,$$

   where L is independent of x and y.

Then the sequence $\{x_k\}$ generated by (1) with $x_0 \in I$ converges to the unique point $\alpha \in I$ which satisfies

$$F(\alpha) = \alpha. \tag{2}$$

The point given by (2) is called the *fixed-point*, and the iteration method given by (1) is called the *fixed-point iteration.*

Proof From the second assumption, we have $x_k \in I\,(k = 1, 2, \ldots)$, if $x_0 \in I$. Therefore, from the third assumption, we have

$$|x_{k+1} - x_k| = |\,F(x_k) - F(x_{k-1})\,| < L\,|x_k - x_{k-1}|. \tag{3}$$

Using this, we have for any $m > 0$

$$\begin{aligned} 0 \le |x_{k+m} - x_k| &\le |x_{k+m} - x_{k+m-1}| + |x_{k+m-1} - x_{k+m-2}| + \cdots + |x_{k+1} - x_k| \\ &\le (L^{k+m-1} + \ldots + L^k)\,|x_1 - x_0| \\ &= \frac{L^m - 1}{L - 1}\,L^k\,|x_1 - x_0| \to 0, \qquad k \to \infty. \end{aligned} \tag{4}$$

Thus the sequence $\{x_k\}$ is a Cauchy sequence on I, which has a limit in I. Let α be the limit of the sequence $\{x_k\}$, then we have

$$\alpha = \lim_{k\to\infty} x_{k+1} = \lim_{k\to\infty} F(x_k) = F(\lim_{k\to\infty} x_k) = F(\alpha), \tag{5}$$

since F is continuous. This means α is a fixed-point of F. The point α is unique, since if this is not the case, then for another fixed-point β

$$|\alpha - \beta| = |F(\alpha) - F(\beta)| < L|\alpha - \beta| < |\alpha - \beta|,$$

which is a contradiction. Q.E.D

Corollary Assume that $F(x)$ is differentiable on (a, b) and $|F'(x)| < 1$ on I. Then the sequence $\{x_k\}$ generated by (1) with $x_0 \in I$ converges to α, if the first two conditions of Theorem 1 are satisfied.

2 Newton method

2.1 Newton method as a fixed-point iteration

Consider the problem of solving the nonlinear equation

$$f(x) = 0, \tag{6}$$

where $f : \mathbb{R} \to \mathbb{R}$. Let $F(x)$ be

$$F(x) = x - f(x)/f'(x), \tag{7}$$

then the solution α of the equation (6) is also the fixed-point of $F(x)$. Since the derivative of $F(x)$ at α is given by

$$F'(\alpha) = 1 - \left.\frac{(f'(x))^2 - f(x)\,f'(x)}{(f'(x))^2}\right|_{x=\alpha} = 0, \tag{8}$$

the condition of the corollary is satisfied in the neighbor of α, and therefore the sequence

$$x_{k+1} = x_k - f(x_k)/f'(x_k), \qquad k = 0, 1, \ldots \tag{9}$$

converges to the solution of $f(x) = 0$, when the starting value x_0 is located at the neighbor of α. The method defined by (9) is called the *Newton method*.

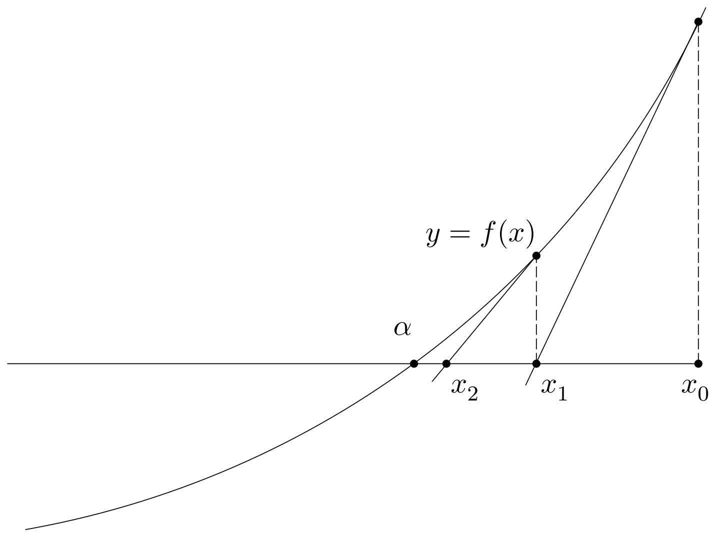

Figure 2: Geometrical interpretation of the Newton method.

2.2 Convergence rate

Let e_k be the error of x_k, i.e., $e_k = x_k - \alpha$, then from the Taylor expansion of $F(x)$, we have

$$\begin{aligned} e_{k+1} = x_{k+1} - \alpha &= F(x_k) - F(\alpha) \\ &= F'(\alpha)\,(x_k - \alpha) + \frac{1}{2!}\,F''(\xi)\,(x_k - \alpha)^2 \end{aligned} \tag{10}$$

where ξ is some value between α and x_k. From this we have $|\, e_{k+1} \,| \leq c|\, e_k^2 \,|$, where $c = \max_x |\, F''(x) \,|/2$.

Example 1 Consider the equation

$$x^3 - 2\,x + 2 = 0. \tag{11}$$

We solve the equation by the Newton method (9) with $x_0 = -1.1$.

2.3 Simultaneous equation

Consider the system of the equations

$$\begin{cases} f_1(x_1, x_2, \ldots, x_n) = 0 \\ f_2(x_1, x_2, \ldots, x_n) = 0 \\ \qquad\qquad \cdots \\ f_n(x_1, x_2, \ldots, x_n) = 0 \end{cases} \tag{12}$$

Table 1: Newton method applied to the equation $x^3 - 2x + 2 = 0$.

k	x_k	$f(x_k)$
0	-1.100000000000000e+00	2.869e+00
1	-2.860122699386503e+00	-1.568e+01
2	-2.164657223087728e+00	-3.814e+00
3	-1.848356485722793e+00	-6.181e-01
4	-1.773434389574454e+00	-3.071e-02
5	-1.769304621075152e+00	-9.067e-05
6	-1.769292354346692e+00	-7.987e-10
7	-1.769292354238631e+00	8.327e-17

and the Newton method for solving the equation. Denoting the k th approximations by $x_1^{[k]}, \ldots, x_n^{[k]}$, the Newton method for solving (12) is given by

$$\begin{pmatrix} x_1^{[k+1]} \\ x_2^{[k+1]} \\ \vdots \\ x_n^{[k+1]} \end{pmatrix} = \begin{pmatrix} x_1^{[k]} \\ x_2^{[k]} \\ \vdots \\ x_n^{[k]} \end{pmatrix} - J^{-1} \begin{pmatrix} f_1(x_1^{[k]}, x_2^{[k]}, \ldots, x_n^{[k]}) \\ f_2(x_1^{[k]}, x_2^{[k]}, \ldots, x_n^{[k]}) \\ \vdots \\ f_n(x_1^{[k]}, x_2^{[k]}, \ldots, x_n^{[k]}) \end{pmatrix}, \qquad k = 0, 1, \ldots, \tag{13}$$

where J is the Jacobian matrix given by

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}.$$

In computing (13), we never calculate the inverse matrix J^{-1}, but instead solve the simultaneous linear equation

$$J \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix} = - \begin{pmatrix} f_1(x_1^{[k]}, x_2^{[k]}, \ldots, x_n^{[k]}) \\ f_2(x_1^{[k]}, x_2^{[k]}, \ldots, x_n^{[k]}) \\ \vdots \\ f_n(x_1^{[k]}, x_2^{[k]}, \ldots, x_n^{[k]}) \end{pmatrix} \tag{14}$$

by the Gaussian elimination, and after that we calculate $x_i^{[k+1]} = x_i^{[k]} + d_i \, (i = 1, \ldots, n)$.

In computing the Newton method, the computational costs for the Jacobian matrix J and for solving (14) are proportional to n^2 and n^3, respectively. To reduce these costs, particularly when n is large, we usually use the quasi Newton method, in which the Jacobian matrix is calculated only for $x_i^{(0)} \, (i = 1, \ldots, n)$ and is fixed until convergence.

Next we consider the convergence of the Newton method for multi-dimensional cases. Let α_i be the i th solution, that is the value $\boldsymbol{f}(\alpha_1, \ldots, \alpha_n) = 0$, where $\boldsymbol{f} \in \mathbb{R}^n$, and $x_i^{[k]}$ be the k th approximation to α_i. Here, we use the vector notations

$$\begin{aligned}
&\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^T, \quad \boldsymbol{x}^{[k]} = (x_1^{[k]}, \ldots, x_n^{[k]})^T, \\
&\boldsymbol{f}(\boldsymbol{x}^{[k]}) = (f_1(x_1^{[k]}, \ldots, x_n^{[k]}), \ldots, f_n(x_1^{[k]}, \ldots, x_n^{[k]}))^T, \\
&J((x_1^{[k]}, \ldots, x_n^{[k]}) = J(\boldsymbol{x}^{[k]}).
\end{aligned}$$

Then the error of $\boldsymbol{x}^{[k]}$, that is $\boldsymbol{e}^{[k]} = \boldsymbol{x}^{[k]} - \boldsymbol{\alpha}$, is given by

$$\boldsymbol{e}^{[k+1]} = \boldsymbol{e}^{[k]} - J(\boldsymbol{x}^{[k]})^{-1}\, \boldsymbol{f}(\boldsymbol{x}^{[k]}). \tag{15}$$

From the Taylor expansion, we have

$$\begin{aligned}
\boldsymbol{f}(\boldsymbol{x}^{[k]}) &= \boldsymbol{f}(\boldsymbol{\alpha}) + J(\boldsymbol{x}^{[k]})\,(\boldsymbol{x}^{[k]} - \boldsymbol{\alpha}) + O(\|\boldsymbol{x}^{[k]} - \boldsymbol{\alpha}\|^2) \\
&= J(\boldsymbol{x}^{[k]})\,\boldsymbol{e}^{[k]} + O(\|\boldsymbol{e}^{[k]}\|^2).
\end{aligned} \tag{16}$$

Substituting this into (15), we have

$$\|\boldsymbol{e}^{[k+1]}\| = O(\|\boldsymbol{e}^{[k]}\|^2), \tag{17}$$

which means that the Newton method (13) converges quadratically.

Example 2 Consider the simultaneous nonlinear equation

$$\begin{cases} f(x,y) = -\dfrac{x^2}{4} - \dfrac{y^2}{9} + 1 = 0, \\ g(x,y) = x^2 - y = 0. \end{cases} \tag{18}$$

The Newton method for this equation is

$$\begin{cases} x_{k+1} = x_k + \dfrac{-g_y\, f + f_y\, g}{f_x\, g_y - f_y\, g_x}, \\ y_{k+1} = y_k + \dfrac{g_x\, f - f_x\, g}{f_x\, g_y - f_y\, g_x}, \end{cases} \qquad k = 0, 1, \ldots \tag{19}$$

where

$$\begin{aligned}
f_x &= \frac{\partial f}{\partial x}, \qquad f_y = \frac{\partial f}{\partial y}, \\
g_x &= \frac{\partial g}{\partial x}, \qquad g_y = \frac{\partial f}{\partial y},
\end{aligned}$$

and f, g, f_x, f_y, g_x and g_y are evaluated at $(x_k,\, y_k)$. We find the solution in the region $x > 0$, $y > 0$ by the Newton method. The exact solution in the region is

$$x = \sqrt{\frac{-9 + \sqrt{657}}{8}} = 1.441874268\cdots, \qquad y = \frac{-9 + \sqrt{657}}{8} = 2.079001404\cdots$$

Here is the C program of the Newton method.

```c
/*
   Newton method for the equation
     f(x,y)=0
     g(x,y)=0
*/
#include <stdio.h>
#include <math.h>

#define eps 1.0e-15

void func(x,y,f,g,fx,fy,gx,gy)
     double x,y,*f,*g,*fx,*fy,*gx,*gy;
{
  *f=-x*x/4-y*y/9+1; *fx=-x/2; *fy=-2*y/9;
  *g=x*x-y; *gx=2*x; *gy=-1;
}

main()
{
  double d,e,x0,y0,x1,y1;
  double f,g,fx,fy,gx,gy;
  int k=0;

  x0=1; y0=1;
  func(x0, x0, &f, &g, &fx, &fy, &gx, &gy);
  e=fabs(f)+fabs(g);
  printf(" %3d  %12.8f  %12.8f  %12.4e \n",k,x0,y0,e);

  do {
    d=fx*gy-fy*gx;
    x1=x0+(-gy*f+fy*g)/d;
    y1=y0+( gx*f-fx*g)/d;

    x0=x1; y0=y1; k++;
    func(x0, y0, &f, &g, &fx, &fy, &gx, &gy);
    e=fabs(f)+fabs(g);
    printf(" %3d  %12.8f  %12.8f  %12.4e \n",k,x0,y0,e);
  } while (e>eps);
}
```

Table 2: Newton method applied to equation (18).

k	x_k	y_k	$\|f(x_k,y_k)\|+\|g(x_k,y_k)\|$
0	1.00000000	1.00000000	6.3889e-01
1	1.67647059	2.35294118	7.7540e-01
2	1.46150594	2.08978983	6.5457e-02
3	1.44201231	2.07901951	4.8789e-04
4	1.44187427	2.07900140	2.3854e-08
5	1.44187427	2.07900140	3.1499e-16