Solving Nonlinear Equations

1 Nonlinear equations

We say a function $f(\mathbf{x})$ is linear in a domain, if for any scalars α and β, and for any $\mathbf{x}$ and $\mathbf{y}$ in that domain,

$$f(\alpha\mathbf{x} + \beta\mathbf{y}) = \alpha f(\mathbf{x}) + \beta f(\mathbf{y}).$$

Otherwise, we say the function $f(\mathbf{x})$ is nonlinear, for example, $f(x) = x^2 - 1$, $f(x) = e^{\sin(x)}$, are nonlinear functions.

Let $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^n$, be a linear function of $\mathbf{x} \in \mathbb{R}^n$, and it maps any vector of n components to a vector of n components. Then we can see that $f(\mathbf{x})$ can be written as the standard form in terms of matrix and vector operations

$$f(\mathbf{x}) = A\mathbf{x} - \mathbf{b},$$

where A is a $n \times n$ matrix, and $\mathbf{b}$ is a n-dimensional column vector.

Given a function $f(\mathbf{x})$, we call $\mathbf{x}$ a root of the function if $f(\mathbf{x}) = \mathbf{0}$, i.e., if the image of the variable $\mathbf{x}$ under the map f is the zero vector.

If $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^n$ is a linear function, i.e., $f(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$, the roots of $f(\mathbf{x})$ are just the solutions of the system of linear equations

$$A\mathbf{x} = \mathbf{b}.$$

If A is invertible then the root is unique; if A is singular, the root may either not exist or there are an infinite number of roots, depending whether $\mathbf{b}$ is in the range of A or not. To find such a solution, we can use the Gaussian elimination for example.

The solution behaviors of nonlinear equations are much more complicated than linear equations. It may have no solution, e.g., $e^x = 0$; a unique solution, e.g., $e^x = 2$; any finite number of solutions, e.g., $(x-1)(x-2)\cdots(x-n) = 0$; or a infinite number of solutions $\sin(x) = 0$.

More complicated is to find the roots of nonlinear functions. Typically, there are not direct formulas for the roots of nonlinear functions. For example, in general, for polynomials of degrees larger or equal than five, there are no closed-form formulas for the roots of such polynomials. This means that any algorithms for solving nonlinear equations must be iterative. A sequence of iterates are generated in the algorithm and it is then expected that the sequence converge to the exact roots of the nonlinear functions.

Here we only discuss algorithms for solving nonlinear equations of a single variable.

2 Solving nonlinear equations of one variable

Given a function $f(x) : \mathbb{R} \to \mathbb{R}$, we consider to solve the nonlinear equation

$$f(x) = 0.$$

As mentioned earlier, in general it is impossible to make any global assertions about solution of nonlinear equations. However there are several local criteria that can be used to study the solutions of nonlinear equations. The simplest one is the intermediate value theorem for nonlinear equations in one dimension.

Theorem 1 (Intermediate Value Theorem) *Let $f(x) : \mathbb{R} \to \mathbb{R}$ be a continuous function. Given $x_1 \in \mathbb{R}$, $x_2 \in \mathbb{R}$, for any value y between $f(x_1)$ and $f(x_2)$, i.e.,* $\min\{f(x_1), f(x_2)\} \le y \le \max\{f(x_1), f(x_2)\}$*, there exists $x^* \in [x_1,\ x_2]$ such that $f(x^*) = y$.*

2.1 Interval bisection method

Based on Theorem 1, we know that for any function $f(x) : \mathbb{R} \to \mathbb{R}$, if $f(a)f(b) \le 0$, then there must exist $x^* \in [a,\ b]$ such that $f(x^*) = 0$. This gives the following interval bisection method to find a solution of $f(x) = 0$.

Algorithm: Interval bisection method for solving $f(x) = 0$

```
Given $a$, $b$, and a tolerance $\varepsilon$
if   $(f(a) * f(b) > 0)$,  Error(‘ Not bracketing ’) and return;
$k = 1$,   $x_k = (a+b)/2$
while   $|f(x_k)| > \varepsilon$
  if   $(f(a) * f(x_k) > 0)$
    $a = x_k$
  else
    $b = x_k$
  end
  $k = k + 1$,   $x_k = (a+b)/2$
end
```

We can see that this algorithm keeps shrinking by half the interval which always encloses one solution of $f(x) = 0$. At the end of the interval bisection method, x_k will be an approximation of the true solution x^*.

The interval bisection method is an iterative method solving the nonlinear equation $f(x) = 0$. It generates a sequence of iterates x_k, which converges to a true solution x^*. To study the rate of convergence, let us denote the difference between x_k and x^* by $e_k = x_k - x^*$, at each iteration step k, for $k = 1, 2, 3, \ldots$. Then we know that

$$|e_1| \le \frac{b-a}{2}, \quad |e_2| \le \frac{b-a}{2^2}, \quad \ldots \quad |e_k| \le \frac{b-a}{2^k}, \quad |e_{k+1}| \le \frac{b-a}{2^{k+1}}, \quad \ldots$$

In general we have

$$\frac{|e_{k+1}|}{|e_k|} \approx 0.5,$$

which implies that the error is reduced by half after each step. If the length of the initial interval $[a,\ b]$ is 1, then in order for the error $|e_k|$ to be less than 10^{-12}, it needs to take $\log_2 10^{12} \approx 40$ steps.

In general, for an iterative algorithm, let us denote the error between the iterate x_k and the true solution x^* by $e_k = x_k - x^*$, at each step k of the iteration. The convergence rate of the iterative algorithm can be determined by the reduction in the error,

$$\lim_{k\to\infty} \frac{\|e_{k+1}\|}{\|e_k\|^r} = C.$$

- if $r = 1$, and $C < 1$, then we say the convergence rate is *linear*,
- if $r > 1$, we call it *superlinear* convergence,
- if $r = 2$, *quadratic* convergence,
- if $r = 3$, *cubic* convergence.

Typically, a quadratic convergence is faster than the linear convergence, and the cubic convergence is faster than the quadratic convergence. For example, the following two sequences have linear and quadratic convergence rates respectively, and we can see the errors in the second sequences converge to zero faster,

$$e_1 = 10^{-1}, \quad e_2 = 10^{-2}, \quad e_3 = 10^{-3}, \quad e_4 = 10^{-4}, \quad e_5 = 10^{-5}, \quad \ldots$$
$$e_1 = 10^{-1}, \quad e_2 = 10^{-2}, \quad e_3 = 10^{-4}, \quad e_4 = 10^{-8}, \quad e_5 = 10^{-16}, \quad \ldots$$

We know that the convergence rate of the interval bisection method is linear with $C = 0.5$. We also know from earlier experience that the convergence rates of the power iteration and inverse iteration for finding the eigenvalues of a matrix are both linear, while the convergence rate of the Rayleigh quotient iteration is cubic.

In the following, we discuss algorithm with a quadratic convergence rate for solving nonlinear equations.

2.2 Fixed-point iteration

Given a function $g(x) : \mathbb{R} \to \mathbb{R}$, we say x^* is a fixed point of the function $g(x)$, if

$$x^* = g(x^*),$$

i.e., the image of x^* under the map (function) $g(x)$ is x^* itself.

The following algorithm can be used to find a fixed point of functions. We call it the fixed-point iteration.

Algorithm: fixed point iteration

Given a function $g(x)$ and an initial guess x_0

```
for $k = 0, 1, 2, 3, \ldots$
    $x_{k+1} = g(x_k)$
end
```

The fixed point iteration generates a sequence of iterates x_k, $k = 1, 2, 3, \ldots$. The question is whether this sequence converges to a fixed point of $g(x)$ or not. If it converge, how fast the convergence is. To study that, we have following theorem.

Theorem 2 *Let $g(x) : \mathbb{R} \to \mathbb{R}$ has continuous derivative in a neighborhood of its fixed point x^*, and let $|g'(x^*)| < C < 1$. For any initial guess x_0, which is close enough to x^*, the fixed point iteration $x_{k+1} = g(x_k)$, $k = 0, 1, \ldots$, converges to x^*, and the error at each iteration satisfies*

$$|e_{k+1}| \leq C|e_k|.$$

Furthermore, if $g'(x^*) = 0$, then the convergence is at least quadratic.

Proof: We denote the error at each iteration step by

$$e_k = x_k - x^*, \quad k = 0, 1, 2, \ldots$$

Let us look at the error after the first iteration. We have

$$e_1 = x_1 - x^* = g(x_0) - x^* = g(x_0) - g(x^*).$$

From the *Mean Value Theorem*, we know that there exits a point $\widehat{x}_0$ between x_0 and x^* such that

$$g(x_0) - g(x^*) = g'(\widehat{x}_0)(x_0 - x^*).$$

Therefore we have

$$e_1 = g'(\widehat{x}_0)(x_0 - x^*) = g'(\widehat{x}_0)e_0.$$

In general, we have,

$$e_{k+1} = g'(\widehat{x}_k)e_k, \quad \text{for} \quad k = 0, 1, 2, 3, \ldots$$

for a certain $\widehat{x}_k$ between x_k and x^*.

Since $g'(x)$ is continuous in the neighborhood of x^* and $g'(x^*) < C < 1$, we know that if the initial guess x_0 is close enough to x^*, then we have for all those $\widehat{x}_k$, $k = 0, 1, 2, \ldots$

$$|g'(\widehat{x}_k)| < C < 1.$$

Therefore we have

$$|e_{k+1}| \leq |g'(\widehat{x}_k)e_k| \leq C|e_k| \leq C^2|e_{k-1}| \leq \ldots \leq C^{k+1}|e_0| \to 0, \qquad \text{as} \qquad k \to \infty.$$

Therefore the fixed-point iteration $x_{k+1} = g(x_k)$ converges to x^*.

Furthermore, if $g'(x^*) = 0$, we will see that the convergence rate is at least quadratic. In this case, we have, from the Taylor series,

$$g(x_k) = g(x^*) + g'(x^*)(x_k - x^*) + g''(\widehat{x}_k)(x_k - x^*)^2/2,$$

which is

$$g(x_k) - g(x^*) = g''(\widehat{x}_k)(x_k - x^*)^2/2.$$

Therefore we have $\dfrac{|e_{k+1}|}{|e_k|^2} = \dfrac{g''(\widehat{x}_k)}{2} \to \dfrac{g''(x^*)}{2}$, as $k \to \infty$, and the convergence is quadratic. □

Using the fixed point iteration, we can consider solving nonlinear equations. If a solution of the nonlinear equation

$$f(x) = 0,$$

satisfies

$$x = g(x),$$

then we can see that to solve the nonlinear equation $f(x) = 0$ is equivalent to find a fixed point of the function $g(x)$. We can see that for a given nonlinear function $f(x)$, there are many different possible choices of $g(x)$, such that $x = g(x)$ is equivalent to $f(x) = 0$. For example, consider to solve

$$f(x) = x^2 - x - 2 = 0,$$

which can be equivalently written as, $x = g(x)$, where $g(x)$ can be chosen as

- $g_A(x) = x^2 - 2$
- $g_B(x) = \sqrt{x+2}$
- $g_C(x) = 1 + 2/x$
- $g_D(x) = (x^2 + 2)/(2x - 1)$

The question is which $g(x)$ to choose. In this example, $f(x)$ has two roots, $x = -1$ and $x = 2$. Here let us consider the convergence to the root $x = 2$, which is a fixed point of each choice of $g(x)$ listed above. Applying the theorem, we can see that

- $g'_A(2) = 4$
- $g'_B(2) = 1/4$
- $g'_C(2) = -1/2$
- $g'_D(2) = 0$

Therefore we can see that the convergence for using $g_B(x)$ and $g_C(x)$ is linear, and quadratic for $g_D(x)$. If we take the fixed-point iteration for $x = g_A(x)$, it will not converge to $x = 2$.

2.3 Newton's method

We see that the essential part in an iterative algorithm is to determine the new iterate, x_{k+1}, based on the current iterate, x_k, in each iteration step. In the interval bisection method, we keep cutting the interval by half and choose the mid-point of the bracketing interval as the new iterate. For the fixed-point iteration, we use $g(x_k)$ as the new iterate. The Newton's method can be regarded as a particular case of the fixed-point iteration.

At an iterate x_k, $k = 0, 1, 2, ...$, we can approximate the function $f(x)$ at $(x_k, f(x_k))$ by using its linear approximation, i.e., the tangent line of the curve $y = f(x)$, through the point $(x_k, f(x_k))$. The equation of the tangent line is given by

$$y = f(x_k) + f'(x_k)(x - x_k) \ =: \ M(x).$$

Here $M(x) = f(x_k) + f'(x_k)(x - x_k)$ is a linear approximation of $f(x)$. Since the goal is to find the root of $f(x)$, it is reasonable to choose the root of its linear approximation $M(x)$ as the new iterate x_{k+1}, i.e., we choose

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

This gives the following Newton's algorithm for solving nonlinear equations of one variable.

Algorithm: Newton's iteration for solving $f(x) = 0$

```
Give a function $f(x)$, and an initial guess $x_0$
for $k = 0, 1, 2, \ldots$
    $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$
end
```

The Newton's method can be regarded as a special fixed point iteration, where we take

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

We can see that the root of $f(x)$ is a fixed point of the function $g(x)$. In fact the last choice of $g_D(x) = (x^2 + 2)/(2x - 1)$ in the previous example for solving $x^2 - x - 2 = 0$ is determined by the Newton's iteration.

We also see that

$$g'(x) = \frac{f(x)f''(x)}{(f'(x))^2},$$

equals zero at the root of $f(x)$ (if $f'(x) \neq 0$, i.e., if the root is a simple root). It implies that the Newton's method is convergent and the convergence is at least quadratic, if the initial guess is sufficiently close to a simple root of $f(x)$.

Certainly it is an issue to find a good enough initial guess. Also when $f'(x_k)$ is very close to zero, the algorithm may perform badly too. It happens that the application of the above standard Newton's method may fail to converge or converges very slowly.

In the following, we consider a strategy which guarantees the convergence of the Newton's method and also preserve the fast convergence.

2.4 A globally convergent Newton's method

We know that the interval bisection method is guaranteed to converge, as long as the initial interval is *bracketing*. The strategy for a globally convergent Newton's method is to combine the interval bisection method with the original Newton's iteration. The basic idea is that at each iteration step, whenever the Newton's iterate is allowable, we choose the Newton's iterate as the new iterate, otherwise we choose the middle point of the interval as the new iterate.

More details are as following. We first choose an initial interval $[a,\ b]$, which satisfies $f(a) \cdot f(b) < 0$, i.e., which encloses at least one root of $f(x)$. We choose the initial guess x_0 as one end point of $[a,\ b]$, e.g., we choose $x_0 = a$. In the next iteration, we compute the Newton's iterate based x_0, i.e, we compute

$$x_N = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

If $x_N \in [a,\ b]$, then we take the new iterate $x_1 = x_N$, otherwise we take the $x_1 = (a+b)/2$. With the choice of x_1 already determined, the new interval $[a,\ b]$ is chosen either as $[a,\ x_1]$, or as $[x_1, b]$, depending on whether $f(a) \cdot f(x_1)$ is negative or not. With the new interval $[a,\ b]$ and the new iterate x_k chosen, the next steps follow in the same way. The algorithm is given as following

Algorithm: A globally convergent Newton's method for solving $f(x) = 0$

```
Given $a$, $b$, and a tolerance $\varepsilon$
if   $(f(a) * f(b) > 0)$,  Error " Not bracketing " and return;
$k = 0$,   $x_k = a$
while   $|f(x_k)| > \varepsilon$

   $x_N = x_k - \frac{f(x_k)}{f'(x_k)}$

   if   $x_N \in [a,\ b]$
      $x_{k+1} = x_N$            (take Newton iterate)
   else
      $x_{k+1} = (a+b)/2$         (take middle point)
   end

   if   $(f(a) \cdot f(x_{k+1}) > 0)$
      $a = x_{k+1}$
   else
      $b = x_{k+1}$
   end
   $k = k + 1$
end
```

This approach is able to guarantee the convergence of the Newton's method. Also when the iterates are close enough to the true solution, it will choose automatically the Newton's iterates and generate a quadratic convergence rate.

3 Solving system of nonlinear equations

We have discussed solving nonlinear equations of a single variable, i.e., for a given function $f(x) : \mathbb{R} \to \mathbb{R}$, we considered solving the equation

$$f(x) = 0.$$

The interval bisection method and the Newton's method have been discussed. For example, the iterates generated by the Newton's method are given by

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \ldots$$

Also to make the convergence of Newton's method robust, an interval bisection strategy can be added into the Newton's algorithm.

Now let us consider solving a system of nonlinear equations

$$F(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

where $F(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^n$ has n components and each component $f_j(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ is a function of $\mathbf{x}$, which contains n variables. Such solutions are also called the roots of $F(\mathbf{x})$.

Solutions of nonlinear equations, especially systems of nonlinear equations, are complicated. In general a system of nonlinear equations may have no solution, a unique solution, or any number of solutions. Here we discuss numerical approaches to solve such nonlinear equations, if such a solution exists. The basic approach is still the Newton's method. But the interval bisection approach to make it globally convergent is no longer applicable in the multidimensional case.

Given a function $F(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^n$, its Jacobian $J(x)$ is defined by

$$J(x) = \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \frac{\partial f_1(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial f_2(\mathbf{x})}{\partial x_1} & \frac{\partial f_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_2(\mathbf{x})}{\partial x_n} \\ & & \cdots & \\ \frac{\partial f_n(\mathbf{x})}{\partial x_1} & \frac{\partial f_n(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_n(\mathbf{x})}{\partial x_n} \\ & & & \end{bmatrix} = \left[\ \nabla f_1(\mathbf{x})\ \nabla f_1(\mathbf{x})\ \ldots\ \nabla f_n(\mathbf{x})\ \right]^T.$$

which essentially represents the first derivative information of the function $F(\mathbf{x})$.

The following theorem on Taylor's expansion is essentially the extension of the Taylor's expansion

$$f(x+h) \ = f(x) + \int_0^1 f'(x+th)h\ dt, \text{ i.e., } f(x+h) - f(x) \ = \ \int_0^1 f'(x+th)h\ dt,$$

to multi-dimensional case.

Theorem 3 *Let $F(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^n$ be continuously differentiable. For any $\mathbf{x}, \mathbf{x}+\mathbf{p} \in \mathbb{R}^n$,*

$$F(\mathbf{x}+\mathbf{p}) = F(\mathbf{x}) + \int_0^1 J(\mathbf{x}+t\mathbf{p})\mathbf{p}dt,$$

where $J(\mathbf{x})$ is the Jacobian of $F(\mathbf{x})$ and the integration is taken for each component of the vector.

Given a current iterate $\mathbf{x}_k$ in an iterative method for finding the root of $F(\mathbf{x})$, from Theorem **??**, we have

$$F(\mathbf{x}) = F(\mathbf{x}_k) + \int_0^1 J(\mathbf{x}_k + t(\mathbf{x}-\mathbf{x}_k))(\mathbf{x}-\mathbf{x}_k)dt.$$

Taking an approximation of the integral, the function $F(\mathbf{x})$ can be approximated in a neighborhood of $\mathbf{x}_k$ by a linear function

$$M(\mathbf{x}) = F(\mathbf{x}_k) + J(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k).$$

We can then take the root of the linear approximation $M(\mathbf{x})$ as the new iterate in the algorithm to find a root of $F(\mathbf{x})$, i.e., we take

$$\mathbf{x}_{k+1} = \mathbf{x}_k - J(\mathbf{x}_k)^{-1}F(\mathbf{x}_k).$$

We call such iteration the Newton's algorithm for solving the system of nonlinear equations $F(\mathbf{x}) = \mathbf{0}$.

Newton's algorithm for solving $F(x) = 0$

```
Given initial guess $x_0$
for $k = 0, 1, 2, \ldots$ until convergence
    solve $J(x_k)p_k = -F(x_k)$ for $p_k$
    $x_{k+1} = x_k + p_k$
end
```

If the Jacobian matrix is non-singular in a neighborhood of a root of $F(\mathbf{x})$ and the initial guess $\mathbf{x}_0$ is close enough to the root, then the Newton's method converge to the root quadratically.

In order to make the Newton's method for solving nonlinear equations globally convergent, we need to connect the solution of nonlinear equations with solving unconstrained minimization problems.

Since a root of $F(\mathbf{x})$ is a minimizer of the function

$$\frac{1}{2}\|F(\mathbf{x})\|_2^2,$$

which is

$$\frac{1}{2}F(\mathbf{x})^T F(\mathbf{x}),$$

we can consider solution strategies of the unconstrained minimization problem

$$\min_{\mathbf{x}} \ \frac{1}{2}\|F(\mathbf{x})\|_2^2.$$

References

[1] M. T. Heath, *Scientific Computing, An introductory Survey*, McGraw-Hill, 2002.