Rootfinding of nonlinear Equations

Problem to Solve

Here, the problems we will solve are of the form:

> Given $f : I = (a, b) \subseteq R \to R$, find $\alpha \in I$ such that $f(a) = 0$

In other words, we want to find numerical approximations of the zeros of a real-valued function of one variable in a given interval. This is done by generating a sequence of values $x^{(k)}$ such that:

$$\lim_{k \to \infty} x^{(k)} = \alpha$$

Notes on Convergence:

The roots of the function we are trying to find may not be real. In these cases, we cannot find a solution with these methods. We will not be able to find the convergent sequence we have previously defined (the method will not converge). The convergence of the method is defined as:

> A sequence $\left\{x^{(k)}\right\}$ generated by a numerical method is said to converge to α with order $p \geq 1$ if:
>
> $$\exists C > 0 : \ \frac{|x^{(k+1)} - \alpha|}{|x^{(k)} - \alpha|^p} \leq C, \ \forall k \geq k_0$$

This condition basically requires that the error at step $k + 1$ is smaller than the error at step k, starting from a certain finite k_0. The degree of "strictness" of this condition is measured by p, which is the **order of convergence** of the method. The higher the p, the better convergence the method has (we reach the solution faster).

In the special case where $p = 1$, the additional condition is needed that $C < 1$ to make sure that the method converges.

It is also important to point out that the convergence of the method depends also on the initial datum $x^{(0)}$. If we start the method far away from α, the method is less likely to converge as quickly (if at all). In the special case where a method converges for any choice of $x^{(0)}$, it is said that the method is **globally convergent**.

Notes on Conditioning of Nonlinear Equations:

Definition of the Resolvent:

> If a problem admits a unique solution, then there exists a mapping G, called the resolvent, between the data d and the solutions set x such that:
>
> $$x = G(d)$$
>
> Then:

> $$F(x,d) = F(G(d),d) = 0$$

Definition of the Relative Condition Number:

> If the data of the problem with solution x is d, the **relative condition number** is defined as:
>
> $$K(d) = \sup_{\delta d \in D} \frac{||\delta x||/||x||}{||\delta d||/||d||}$$

Definition of the Absolute Condition Number:

> If the data of the problem with solution x is d, the **absolute condition number** is defined as:
>
> $$K_{abs}(d) = \sup_{\delta d \in D} \frac{||\delta x||}{||\delta d||}$$

We can approximate the condition numbres by using the taylor expansion of the resolvent:

> If G is the resolvent of a problem with datum d, the relative and absolute condition numbers are defined as:
>
> $$K(d) \approx ||G'(d)|| \frac{||d||}{||G(d)||}$$
>
> $$K_{abs}(d) \approx ||G'(d)||$$

Consider the equation $f(x) = \phi(x) - d = 0$, where f is continuous and differentiable. In this case, the (only) root is $\alpha = \phi^{-1}(d)$ (assuming that ϕ is invertible, otherwise the problem is ill conditioned). From the definition of the *Resolvent*, we can tell that here $G(x) = \phi^{-1}(x)$. Also, $(\phi^{-1})'(d) = 1/\phi'(\phi^{-1}(d)) = 1/\phi'(\alpha)$ and $f'(x) = \phi'(x)$ (d is constant) so we can compute the *Condition Numbers*:

$$K(d) \approx ||(\phi^{-1})'(d)|| \frac{||d||}{||\phi^{-1}(d)||} = \frac{1}{|f'(\alpha)|} \frac{|d|}{|\alpha|} = \frac{|d|}{|\alpha||f'(\alpha)|}$$

$$K_{abs}(d) \approx ||(\phi^{-1})'(d)|| = \frac{1}{|f'(\alpha)|}$$

The *Condition Numbers* give us information about how sensitive the method is to changes in the datum d when finding the roots of f. Here, we can see that if $f'(\alpha)$ is small, then K and K_{abs} will be very big, so the method will be ill conditioned, as a small change in the datum d will mean a large change in the solution x.

Therefore, the problem is **ill-conditioned if $f'(\alpha)$ is small** and **well-conditioned if $f'(\alpha)$ is large**.

In the more general case:

> If α is a root of f with multiplicity $m \geq 1$, we obtain the following formula for the **absolute condition number**:

> $$K_{abs}(d) \approx \left|\frac{m!\delta d}{f^{(m)}(\alpha)}\right|^{1/m} \frac{1}{|\delta d|}$$

Notes on Error Estimation:

Assume $d = 0$ and let α be a simple root of f. Moreover, for an approximation $\hat{\alpha}$ to the actual root α (where $\hat{\alpha} \neq \alpha$), we have that $f(\hat{\alpha}) = \hat{r}$, where $\hat{r}$ is called the *residual*. From the previous *Notes on Conditioning* we know that:

$$K_{abs}(0) \approx \frac{1}{|f'(\alpha)|}$$

and

$$K_{abs}(0) \leq \frac{||\delta x||}{||\delta d||}$$

Therefore, letting $\delta x = \hat{\alpha} - \alpha$ and $\delta d = \hat{r}$ and combining the previous expressions:

$$\frac{|\hat{\alpha} - \alpha|}{|\hat{r}|} \lesssim \frac{1}{|f'(\alpha)|}$$

Finally, we obtain the formula:

> For roots of multiplicity 1:
>
> $$\frac{|\hat{\alpha} - \alpha|}{|\alpha|} \lesssim \frac{|\hat{r}|}{|f'(\alpha)||\alpha|}$$

And, similarly:

> For roots of multiplicity $m \geq 1$:
>
> $$\frac{|\hat{\alpha} - \alpha|}{|\alpha|} \lesssim \left(\frac{m!}{|f^{(m)}(\alpha)||\alpha|^m}\right)^{1/m} |\hat{r}|^{1/m}$$

Geometric Approaches to Rootfinding

Bisection Method

This method is based on the *Theorem of Zeros for Continuous Functions (Bolzano's Theorem)*:

> **Bolzano's Theorem**
> Given a continuous function $f : [a, b] \to R$, such that $f(a)(b) < 0$, then $\exists \alpha \in (a, b)$ such that $f(\alpha) = 0$

Starting from an initial interval, the *Bisection Method* creates a series of subintervals by halving at each iteration. *Bolzano's Theroem* is then applied to both subintervals to determine in which of the two the zero is contained. The subinterval that contains the zero is the one that is chosen for halving in the next iteration. The method ends when the length of the interval is double the maximum allowed error. Then, the solution is chosen to be the midpoint of such interval.

> **Bisection Method**
>
> 1. An initial interval $I_0 = [a, b]$ containing a zero of the function $f(x)$ is defined. It must verify that $f(a)f(b) < 0$.
> 2. In the first iteration, set $a^{(0)} = a$, $b^{(0)} = b$ and $x^{(0)} = (a^{(0)} + b^{(0)})/2$.
> 3. For each iteration:
>   - If $f(x^{(k)})f(a^{(k)}) < 0$, set $a^{(k+1)} = a^{(k)}$ and $b^{(k+1)} = x^{(k)}$.
>   - If $f(x^{(k)})f(b^{(k)}) < 0$, set $a^{(k+1)} = x^{(k)}$ and $b^{(k+1)} = b^{(k)}$.
>   - Finally, set $x^{(k+1)} = (a^{(k+1)} + b^{(k+1)})/2$.
>
> 4. Stop the iterations when $b^{(k)} - a^{(k)} < \epsilon$, where ϵ is the maximum allowed error.

There are several considerations to take into account when using this method:

1. Specifying the initial interval can often be inconvenient, as we need to know more or less where the function has a zero beforehand.
2. An advantage of this method is it *does not require any derivative information*.
3. A disadvantage of this method is that it *cannot be used to find even order roots*, as the function does not change sign on either side of them.

The error in each step of this method is half the length of the k-th subinterval. Therefore:

> In the k-th step of the *Bisection Method* the error is:
>
> $$e^{(k)} \leq \frac{b-a}{2^k}$$
>
> From this, we see that $\lim_{k\to\infty} |e^{(k)}| = 0$ independently of the choice of a and b, so **the Bisection Method is globally convergent**.

Moreover, if we want to find out the number of iterations needed to obtain an error smaller than ϵ, we need:

$$e^{(m)} \leq \frac{b-a}{2^m} \leq \epsilon \to m \geq \log_2\left(\frac{b-a}{\epsilon}\right) = \frac{\log((b-a)/\epsilon)}{\log(2)}$$

This singles out the Bisection Method as an algorithm of **slow but certain convergence**. Finally, by its nature, the error of the Bisection Method does not decrease monotonically. The *maximum error* does, but the *actual error* does not, as the distance between the actual root and the center of two subsequent subintervals can may increase slightly, even though the general tendency is decreasing. For this reason, it *cannot be really classed as an order 1 method*.

Chord, Secant, Regula Falsi and Newton Methods

In order to achieve better convergence, we need more information about f or its derivative f'. Expanding f up to first order around its root α, we obtain, for some $\phi \in (\alpha, x)$:

$$f(x) = f(\alpha) + (x - \alpha)f'(\phi)$$

> The problem of finding a root $\alpha \in C$, for $f : (a, b) \subseteq R \to R$, such that $f(\alpha) = 0$ can be linearized as:
>
> $$0 = f(\alpha) = f(x) + (\alpha - x)f'(\phi)$$

Which prompts the following iterative method:

> For any $k \geq 0$, given $x^{(k)}$, determine $x^{(k+1)}$ by solving the equation:
>
> $$f\left(x^{(k)}\right) + \left(x^{(k+1)} - x^{(k)}\right) q_k = 0$$
>
> where q_k is a suitable approximation of $f'\left(x^{(k)}\right)$. More conveniently:
>
> $$x^{(k+1)} = x^{(k)} - q_k^{-1} f(x^{(k)}),\ \forall k \geq 0$$

The only difference between the four metods shown in this section is the way in which they approximate q_k.

The Chord Method

In the **Chord Method**, q_k is computed as the slope of the line joining the function evaluated at the two ends of the interval (a, b) where we are looking for a solution. These are the points $(a, f(a))$ and $(b, f(b))$

> For the **Chord Method**:
>
> $$q_k = \frac{f(b) - f(a)}{b - a},\ \forall k \geq 0$$
>
> So, the recursive formula is:
>
> $$x^{(k+1)} = x^{(k)} - \frac{b - a}{f(b) - f(a)} f(x^{(k)}),\ \forall k \geq 0$$

The order of convergence of this method is $p = 1$.

The Secant Method

In the **Secant Method**, q_k is computed as the slope of the line joining the function evaluated at the two previous approximations of the zero. These are the points $(x^{(k)}, f(x^{(k)}))$ and $(x^{(k-1)}, f(x^{(k-1)}))$:

> For the **Secant Method**:
>
> $$q_k = \frac{f(x^{(k)}) - f(x^{(k-1)})}{x^{(k)} - x^{(k-1)}},\ \forall k \geq 0$$
>
> So, the recursive formula is:

> $$x^{(k+1)} = x^{(k)} - \frac{x^{(k)} - x^{(k-1)}}{f(x^{(k)}) - f(x^{(k-1)})} f(x^{(k)}),\ \forall k \geq 0$$

The *Secant Method* computes a new q_k at every iteration, whereas the *Chord Method* only computes it once. This means that it has a higher computational cost. However, it also has a higher order of convergence, which pays off:

> **Local Convergence Theorem (I)**
> Let $f \in C^2(J)$, J being a suitable neighbourhood of the root α and assume that $f''(\alpha) \neq 0$. Then, if the initial data $x^{(-1)}$ and $x^{(0)}$ are chosen in J sufficiently close to α, the secant method converges to α with order $p = \frac{1+\sqrt{5}}{2} \approx 1.63$

One thing to note about this method is that some of the iterates may fall outside the initially set interval.

The Regula Falsi Method

The **Regula Falsi Method** is a variation of the *Secant Method* in which q_k is also computed in each step. However, instead of computing q_k as the slope of the line joining the function evaluated at the previous two approximations of the zero, it is computed as the line joining the points $\left(x^{(k)}, f(x^{(k)})\right)$ and $\left(x^{(k')}, f(x^{(k')})\right)$, where k' is the largest $k' < k$ such that the function evaluated at $x^{(k')}$ is of the opposite sign as the function evaluated at $x^{(k)}$ $\left(f(x^{(k)})f(x^{(k')}) < 0\right)$. Then:

> For the **Regula Falsi Method**:
>
> $$q_k = \frac{f(x^{(k)}) - f(x^{(k')})}{x^{(k)} - x^{(k')}},\ \forall k \geq 0$$
>
> Where k' is the largest index smaller than k such that $f(x^{(k)})f(x^{(k')}) < 0$. So, the recursive formula is:
>
> $$x^{(k+1)} = x^{(k)} - \frac{x^{(k)} - x^{(k')}}{f(x^{(k)}) - f(x^{(k')})} f(x^{(k)}),\ \forall k \geq 0$$

Note that the sequence of chosen k' will always be nondecreasing. Therefore, it is enough to stop at the previous k' when sweeping back to find the next k'.

The order of convergence of this method is $p = 1$. Even though it is of the same complexity as the *Secant Method*, it has a lower order of convergence. However, it has the advantage of having all the iterates $x^{(k)}$ inside the starting interval $\left[x^{(-1)}, x^{(0)}\right]$.

Newton's Method

Newton's Method introduces the use of the derivative of the function f in order to achieve a higher order of convergence.

> For **Newton's Method**:
>
> $$q_k = f'(x^{(k)}),\ \forall k \geq 0$$

> So, the recursive formula is:
>
> $$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})},\ \forall k \geq 0$$

The order of convergence of this method is $p = 2$, at the price of a higher computational cost, as the two functional evaluations can be demanding especially for complex functions.

Stopping Criteria

There are two possible ways of deciding when to stop a method's iteration depending on the precision we want to achieve:

1. *Stopping based on the residual:* The iterative process terminates at the first step k such that $|f(x^{(k)})| < \epsilon$.

    - This criteria can lead to big errors if the function has a *small derivative close to the root* (functions with a small derivative close to the root are ill conditioned).
    - *If the derivative is large*, this criteria is too restrictive, and we will have an error much smaller than the required ϵ (this is also not good, as we are wasting computational resources and time to obtain a precision that we do not need).
    - This test is best suited to the cases where the *derivative of the function around the root is close to 1*. Then, the actual error is around the required value ϵ.

2. *Stopping test based on the increment:* the iterative process terminates as soon as $\left|x^{(k+1)} - x^{(k)}\right| < \epsilon$. If $\left\{x^{(k)}\right\}$ is generated by the fixed-point method $x^{(k+1)} = \phi(x^{(k)})$:

    - This criteria is unsatisfactory if $\phi'(\alpha) \approx 1$.
    - Is an optimal critaria for methods of order 2 where $\phi'(\alpha) = 0$ (like *Newton's Method*).
    - Is still satisfactory when $-1 < \phi'(\alpha) < 0$.