Introduction to Optimization and Nonlinear Equations

Zeyu Lu & Yuqiu Yang

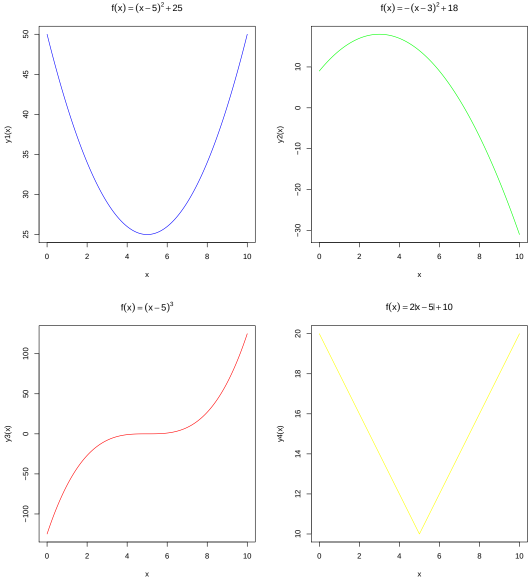
Safe Univariate Methods:

Optimization Problem:Definition

In mathematics, computer science and economics, an optimization problem is the problem of finding the best solution from all feasible solutions.–wiki

Such as finding the maximum/minimum value for a certain function that is defined on a discrete set/continuum

Optimization Problem:examples



Optimization Problem:Assumptions and efficiency

(1)Assumptions needed for each methods?

(2)which method is less restrictive?

(3)How to evaluate the efficiency of a method?

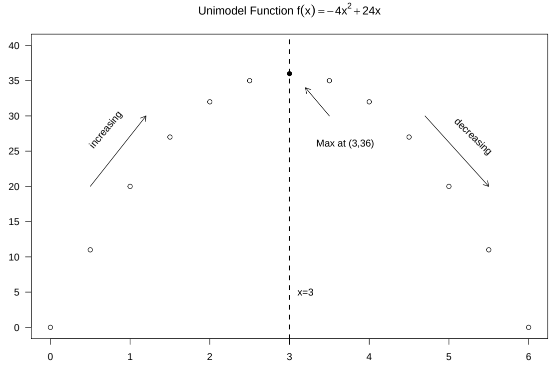
Lattice Search

Finding the maximum of a unimodel function f on a discrete set of points $1, 2, ..., m$ a lattice

unimodel function, A function $f(x)$ is said to be unimodal function if for some value m is monotonically increasing for $x \leq m$ and monotonically decreasing for $x \geq m$.

Lattice Search:Unimodel function on discrete points

Graph for $f(x) = -4x^2 + 24x$, this is a unimodel function.



for $x < 3$, function value $f(x)$ is monotonically increasing, and for $x \geq 3$, $f(x)$ is monotonically decreasing.

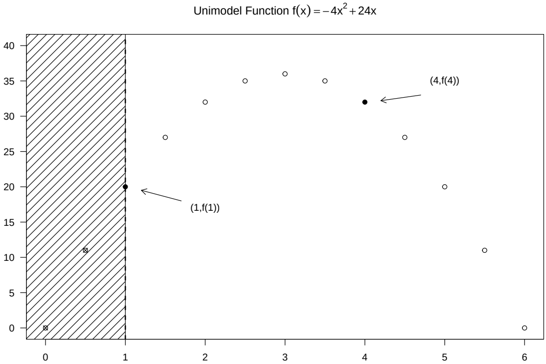
Lattice Search:Optimal Strategy

(i)finding good end strategies for finding the mode on a small set of points

(ii)employing backwards induction to start with the right strategy to match the optimal ending

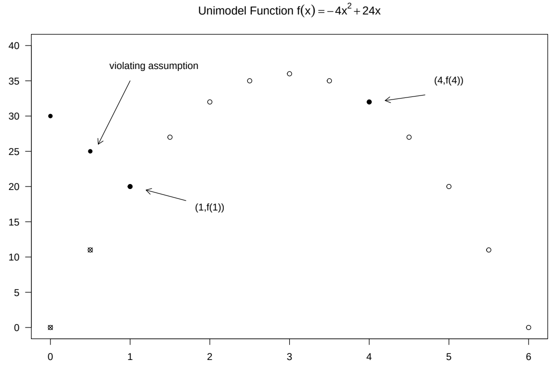
Basically, optimal strategy means the fewest evaluations of the function f that will solve all problems that meet the specifications, here is any strictly unimodel function.

Lattice Search:How to choose points to evaluate?



If $f(4) > f(1)$, then we immediately discard the points that are less than $x = 1$, otherwise it will violate the assumption of unimodel function

Lattice Search:How to choose points to evaluate?



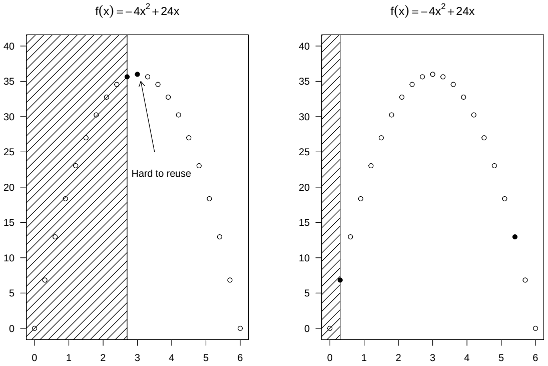
If $f(4) > f(1)$, then we immediately drop the points that are less than $x = 1$, otherwise it will violate the assumption of unimodel function

Lattice Search:How to choose points to evaluate?

If the points are too far away from each other, then only a few portion of
points can be discarded in one step, which decreases the efficiency.

If the two points are close to center of the domain, it's difficult to reuse any of them in the next step, and even we hope to reuse them in the following steps.

Lattice Search:How to choose points to evaluate?



Lattice Search:Fibonacci Numbers

The optimal strategy is by applying Fibonacci numbers

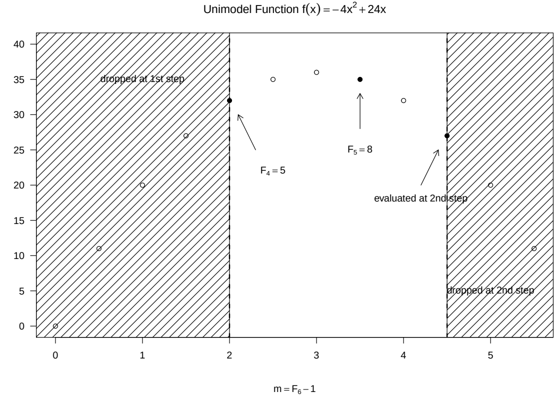
$F_n = \{1, 2, 3, 5, 8, 13, ..., F_n = F_{n-1} + F_{n-2}\}$

suppose we have a set of discrete points$\{1, 2, 3, ..., m = F_n - 1\}$, and we begin the searching by evaluating at the pointsF_{n-2}andF_{n-1}.

if $f(F_{n-2}) < f(F_{n-1})$, then the sub-problem is $\{F_{n-2} + 1, ..., F_n - 1\}$ with $f(F_{n-1})$ has already been evaluated, thus a problem with $F_n - 1$ elements needs $n - 1$ evaluations to solve.

Lattice Search:Fibonacci Numbers

$F_n = \{1, 2, 3, 5, 8, 13\}$



For $m = F_6 - 1 = 12$, 5 evaluations are enough to reach the maximal value.

after each step, we got a sub-problem with $F_{n-1} - 1$ points.

Lattice Search:Details

(i)If the values of the function are the same at F_{n-2} and F_{n-1}, the mode must be between the two points according to our assumption, then it doesn't matter which part is discarded.

(ii)If the number of points m is not one fewer than a Fibonacci number, then add some points at one side. where the value of additional points is $-\infty$.

Golden Section

A more common problem is searching for the maximum on a continuum. so without losing generality, we set the interval (0,1).

Divided the interval and use lattice search by placing m points in the interval, the set is

$$\{0, \frac{1}{m-1}, \frac{2}{m-1}, ..., 1\}.$$

First two points

$lim\frac{F_{n-2}-1}{F_n-1}$ and $lim\frac{F_{n-1}-1}{F_n-1}$

Golden Section

And since we knew that the lattice search is defined by the first two evaluations, let m goes to infinity, then

set$lim\frac{F_{n-1}}{F_n} = \phi$, so that $lim\frac{F_n}{F_{n-1}} = \frac{1}{\phi} = lim\frac{F_{n-2}+F_{n-1}}{F_{n-1}} = \phi + 1$

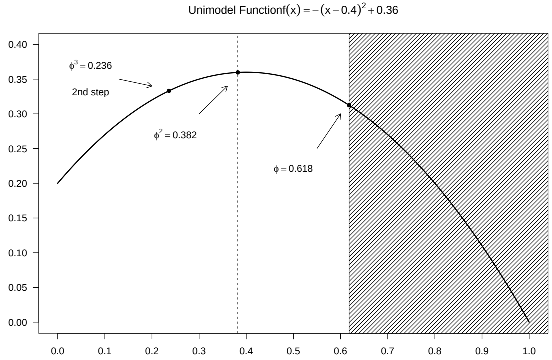
$\phi^2 + \phi + 1 = 0, \phi = \frac{\sqrt{5}-1}{2} \approx .0618$

which also known as the golden ratio. thus the starting points of the search are

$X_1 = \phi^2 \approx 0.382, X_2 = \phi \approx 0.618$

The limit of the lattice search is called the golden section search.

Golden Section



After the first step, the uncertainty of interval is $(0, \phi)$ and the point $\phi^2 = 0.382$ has already been evaluated. Noticed it is also the right point in the second step, which is very similar to lattice search, and we only need to evaluate one more point at $\phi^3 \approx 0.236$.

Bisection

Fibonacci search is less restrictive, since the derivative of the function f doesn't need to exist, but suppose the derivative of the function f is available, which would convert the problem from finding maximum of a unimodel function to finding the root of a monotone function g on the same interval

suppose $g(x)$ is defined on interval (a, b), and let$g(a) < 0 < g(b)$. with a single evaluation at $g(\frac{a+b}{2})$, the uncertainty of interval will be halved.

Bisection

Introduction to Optimization and Nonlinear Equations

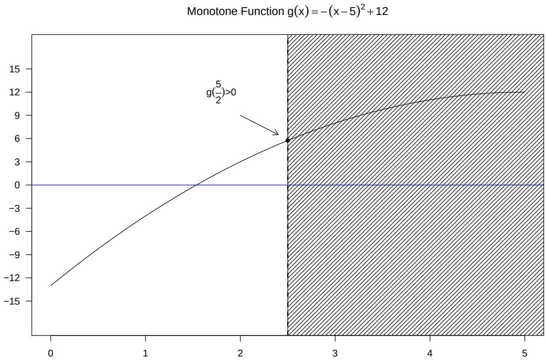
Zeyu Lu & Yuqiu Yang

Safe Univariate Methods:

Root finding

Stopping and Condition

Appendix (Details for order of convergence analysis)



Thus, after the first step, we reset the right endpoint as $\frac{a+b}{2}$, and repeat this procedure to get the root.

Comparison:Golden Section and Bisection

1. Golden Section:
   - less restrictive, requiring only a strictly unimodel function.
   - it reduces the interval of uncertainty to $(0, \phi)$ in each iteration.
2. Bisection:
   - more restrictive, requiring the derivative exist and be available.
   - it halves the interval of uncertainty, that is $\frac{1}{2}$.

Root finding

Newton's Method:Iteration Formula

The more common problem is finding a root for a single nonlinear equation $g(x) = 0$.

for function g, set its derivative as g', we have

$$g_t(x) = g(x_{old}) + g'(x_{old})(x - x_{old})$$

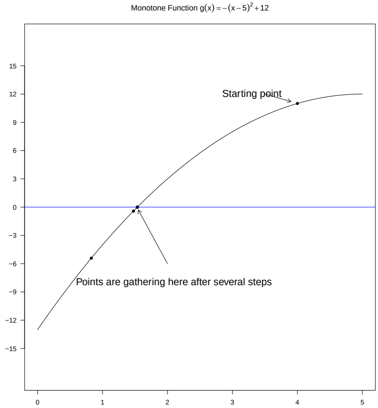
$g_t(x) = 0$ is at

$$x_{new} = x_{old} - \frac{g(x_{old})}{g'(x_{old})}$$

by using n, the iteration formula is:

$$x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)}$$

Newton's Method:Example



n	xn	gn
1	4.0000000	11.0000000
2	-1.5000000	-30.2500000
3	0.8269231	-5.4145710
4	1.4756735	-0.4208771
5	1.5353838	-0.0035653
6	1.5358983	-0.0000003
7	1.5358984	0.0000000
8	1.5358984	0.0000000
9	1.5358984	0.0000000
10	1.5358984	0.0000000

Newton's Method:Quadratic Convergence

If we denote the root by c and the error at iteration n by $e_n = x_n - c$

the relative error is $d_n = \frac{e_n}{c} = \frac{(x_n - c)}{c}$

By using Taylor expansion:

$$g(c) = 0 = g(x_n) + (c - x_n)g'(x_n) + (c - x_n)^2 \frac{g''(t)}{2}$$

where t lies between x_n and c.

Newton's Method:Quadratic Convergence

noticed $e_{n+1} = x_{n+1} - c = x_n - \frac{g(x_n)}{g'(x_n)} - c$

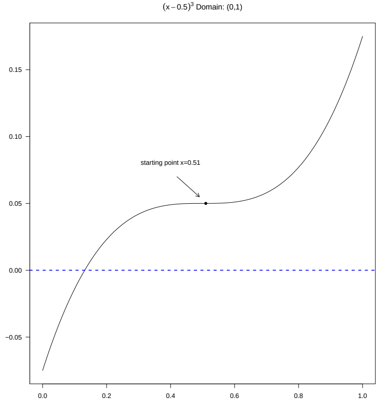
substitute into the equation.

$$x_n - c - \frac{g(x_n)}{g'(x_n)} = (x_n - c)^2 [\frac{g''(t)}{2g'(x_n)}]$$

$$e_{n+1} = e_n^2 [\frac{g''(t)}{2g'(x_n)}]$$

This expression reveals the quadratic convergence of Newton's Method.

Newton's Method:steep or flat?



n	xn	ratio
1	0.510	166.670
2	-166.160	-55.553
3	-110.607	-37.036
4	-73.571	-24.690
5	-48.881	-16.460
6	-32.421	-10.973
7	-21.447	-7.316
8	-14.131	-4.877
9	-9.254	-3.251
10	-6.003	-2.167

noticed for a flat point, $g'(x)$ could be very small so that the next point may leap far away from the true root.

Newton's Method:Pros and Cons

1. Pros:
   - Newton's method achieves the fastest rate of convergence(quadratic).
2. Cons:
   - the derivative function must be available, and finding it can be tedious or impossible.

The Secant Method

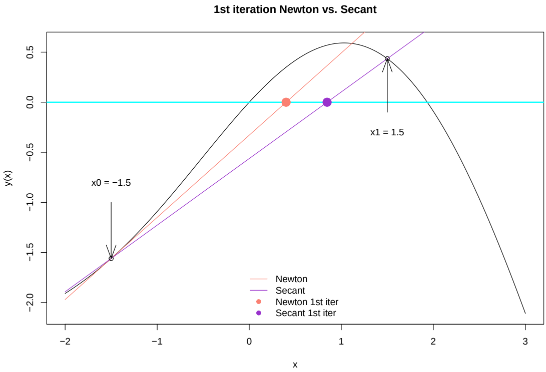
If g' is hard or even impossible to find, we can approximate $g'(x) \approx \dfrac{g(x+h) - g(x)}{h}$.

The iteration formula now becomes $x_{n+1} = x_n - g(x_n)\dfrac{x_n - x_{n-1}}{g(x_n) - g(x_{n-1})}$

Notice two initial points are required instead of one like the Newton's method.

The Secant Method: Geometrical Interpretation

Let $f(x) = sin(x) - (\frac{x}{2})^2$, $x_0 = -1.5$ and $x_1 = 1.5$

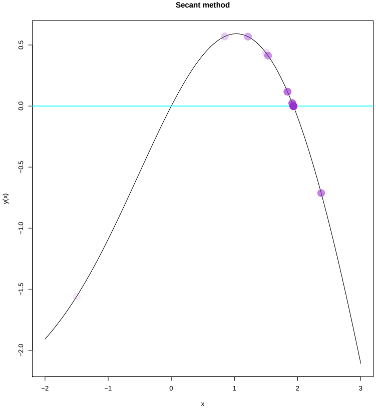


x_{n+1} is taken to be the abscissa of the point of intersection between the secant through $(x_{n-1}, f(x_{n-1}))$ and $(x_n, f(x_n))$ and the x-axis.

The Secant Method: An Example

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

$x_0 = -1.5$ and $x_1 = 1.5$



n	xn	fn
0	{-1.5,1.5}	-1.5600
1	{1.5,0.8459}	0.4350
2	{0.8459,3.6128}	0.5697
3	{3.6128,1.2136}	-3.7169
4	{1.2136,1.5319}	0.5687
5	{1.5319,2.3731}	0.4125
6	{2.3731,1.8403}	-0.7128
7	{1.8403,1.9155}	0.1172
8	{1.9155,1.9347}	0.0238
9	{1.9347,1.9337}	-0.0013
10	{1.9337,1.9338}	0.0000
11	{1.9338,NA}	0.0000

The Secant Method: Order of Convergence

Let the root the Secant Method approaches be c, then asymptotically, we will have

$$|\epsilon_{n+1}| = C|\epsilon_n||\epsilon_{n-1}|,$$

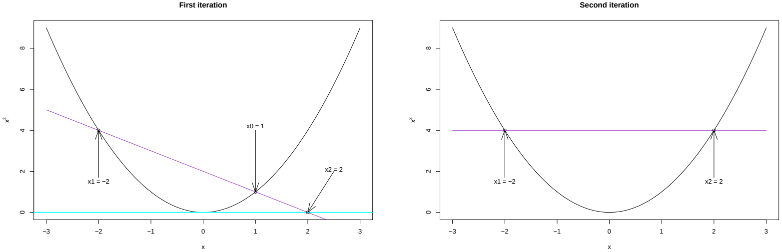
where $\epsilon_n = c - x_n$ and $C = |\dfrac{g''(c)}{2g'(c)}|$.

Based on this relationship, we can find $|\epsilon_n| = C|\epsilon_{n-1}|^{1.618}$

Since the exponent 1.618 lies between 1 (linear convergence) and 2 (quadratic convergence), the convergence rate of the Secant Method is called *superlinear*.

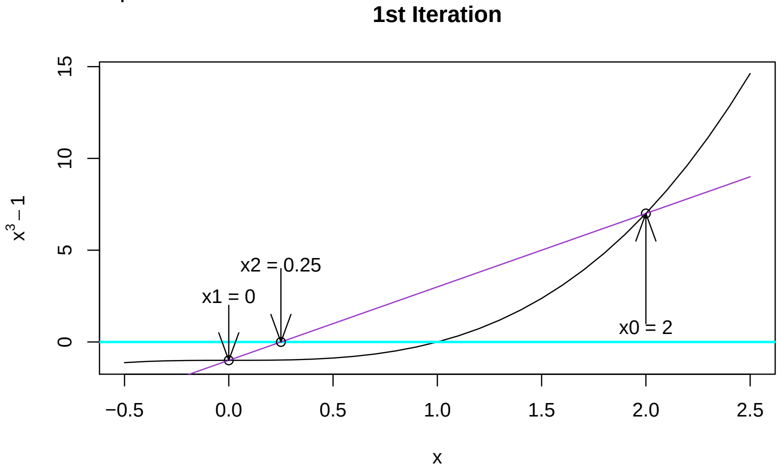
The Secant Method: Pros and Cons

1. Pros:
   - Superlinear convergence
   - No need to evaluate derivatives
2. Cons:
   - Convergence is not guaranteed
   - Not well behaved when g is relatively flat

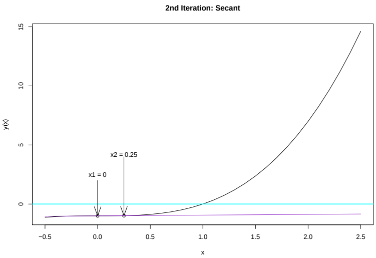


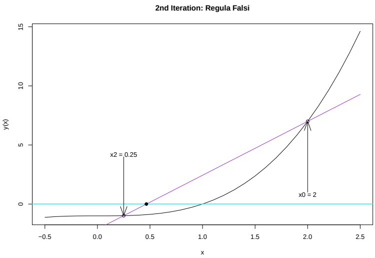
Regula Falsi: A Motivative Example

Borrowing the idea of the Bisection Method, what if we start with two points that straddle the root?



Regula Falsi: A Motivative Example



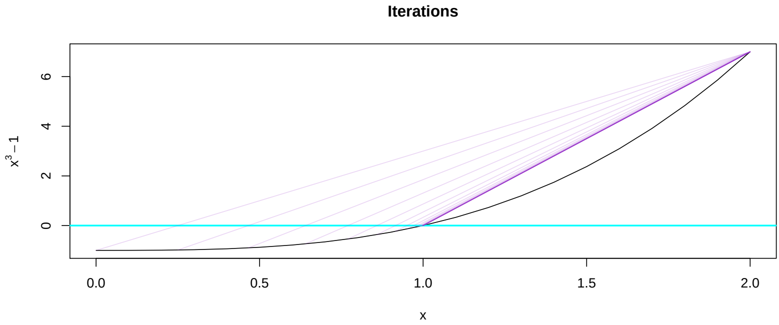


In this case, since the slope of the secant used in the Secant Method is so close to 0, the root is out of our scope.

However, by straddling the root, the Regula Falsi makes sure that the new root is always between the previous two values.

Regula Falsi

A variant of the Secant Method where instead of choosing the secant through $(x_n, g(x_n))$ and $(x_{n-1}, g(x_{n-1}))$, one finds the secant through $(x_n, g(x_n))$ and $(x_{n'}, g(x_{x'}))$ where $n' < n$ is the largest index for which $g(x_n)g(x_{n'}) < 0$.



Regula Falsi: Order of Convergence

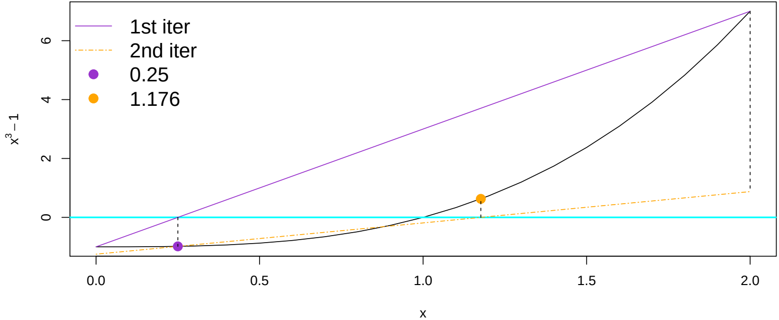
Like the Bisection Method, the Regula Falsi is "safe". However, from the previous example, we see that this method is in general a first-order method.
Especially, if $g(x)$ is convex on $[x_0, x_1]$, then

$$|\epsilon_{n+1}| \approx C|\epsilon_n||\epsilon_0| = C'|\epsilon_n|$$

where $C = \dfrac{g''(c)}{2g'(c)}$
The Regula Falsi Method tends to retain one end-point for several iterations. As a result, it can be a good "start" method or a part of a "hybrid" method, but it should not be used near a root.

Illinois Algorithm: Building on Regula Falsi

In the previous example, if we artificially create a shallower secant, then maybe the end-point will no longer be retained.

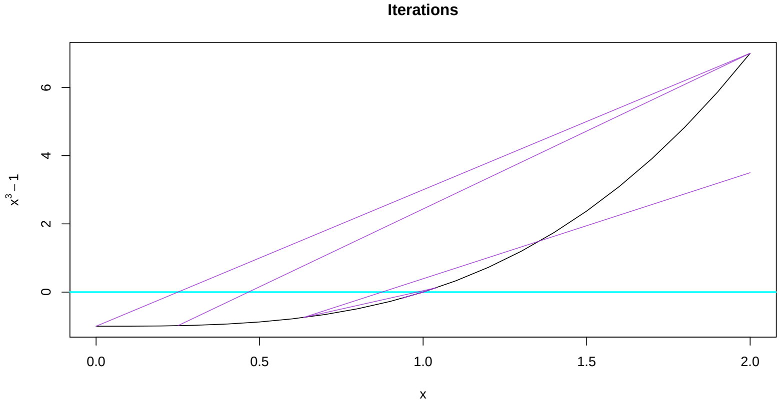


By dividing the function value at 2 by 8 and calculating the new secant, we find a new root on right of the root. In the next iternation, the new root 1.176 instead of 2 will be used.

Illinois Algorithm

1. During the Regula Falsi procedure, once we find one end-point has been retained more than once, we half the function value at that point, find the secant line and the new root.
2. If the point still retains, we repeat Step 1.
3. Once the point changes, we proceed with the Regula Falsi

Illinois Algorithm: An Example



Compared with the Regula Falsi Method, the Illinois Algorithm gets in a small neighborhood of the root in just 4 or 5 iterations.

Illinois Algorithm: Order of Convergence

The order of convergence of the Illinois Algorithm is found based on the following two observations:

$$\epsilon_{n+1} = \frac{g''(c)}{2g'(c)}\epsilon_n\epsilon_{n-1}$$

and asymptotically, we will perform the Illinois step (halving the function value) once every third time.

The order of convergence is found to be approximately 1.44.

Successive Parabolic Interpolation

Recall the Newton's Method for optimization can be written as

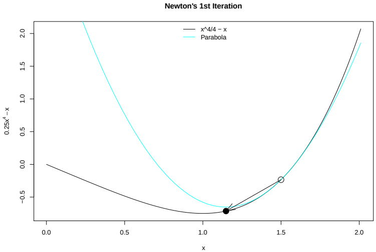
$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$$

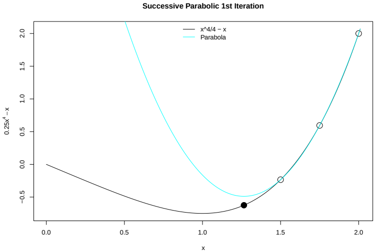
The essence of Newton's Method is locally approximating a function via a sequence of parabolas.

If f' or f'' is hard to find, the Successive Parabolic Interpolation Method can be used to find the extremum.

In each iteration, we fit a parabola to 3 unique points and replace the "oldest" one with the extremum of the fitted parabola.

Successive Parabolic Interpolation: vs. Newton's Method



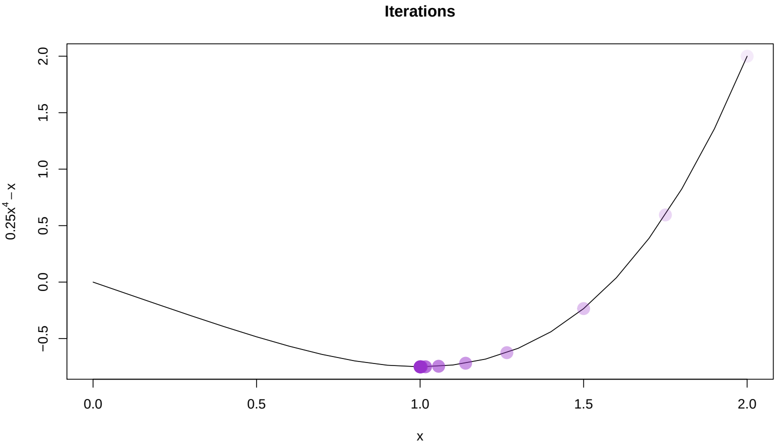


The parabola fitted in the Successive Parabolic Interpolation depends on the 3 points we chose.

In the next iteration, Newton's Method will fit a parabola based on the point 1.1481.

The Successive Parabolic Interpolation will fit a parabola based on 1.75, 1.5, and 1.2653.

Successive Parabolic Interpolation: An Example



The Order of Convergence of the Successive Parabolic Interpolation is approximately 1.3.

Summary: Convergence Rates

	Root Finding	Optimization
Linear	Bisection, Regula Falsi	Golden Section
Superlinear	Secant, Illinois	Parabolic Interpolation
Quadratic	Newton	Newton

Stopping and Condition

Three options for termination

1. Too many steps
   - Usually indicate a serious error in problem specification
2. No change in x or No change in the function values
   - Need to check if a root or an extremum is being approached
   - Root finding: in some cases, no x will produce $g(x)$ "close" to 0
   - Root finding: at some function value, there may appear to be multiple roots
   - Both absolute change $|x_{n+1} - x_n| < \epsilon_x$ and relative change $||x_{n+1} - x_n| < |x_n|\epsilon_x|$ can be used for x.
   - For g only absolute change for root finding. Relative change is appropriate with optimization.

Appendix (Details for order of convergence analysis)

The Secant Method: Several Definitions

Given $n+1$ distinct pairs $\{(x_0, g(x_0)), (x_1, g(x_1)), \ldots, (x_n, g(x_n))\}$, we will define:

$int(x_0, x_1, \ldots, x_n)$: the smallest interval that contains $x_0, \ldots, x_n$

The divided differences

$$g[x_0, x_1, \ldots, x_j, x] = \frac{g[x_0, x_1, \ldots, x_{j-1}, x] - g[x_0, x_1, \ldots, x_j]}{x - x_j}$$

, and

$$g[x_0, x] = \frac{g(x) - g(x_0)}{x - x_0}$$

The Secant Method: Newton's Interpolation Formula

Given $n+1$ distinct pairs $\{(x_0, g(x_0)), (x_1, g(x_1)), \dots, (x_n, g(x_n))\}$, we can interpolate these points using a polynomial $q(x)$ of degree n. Specifically,

$$q(x) = g(x_0) + \sum_{j=1}^{n} g[x_0, x_1, \dots, x_j] \prod_{i=0}^{j-1} (x - x_i)$$

, with the remainder

$$g(x) - q(x) = \frac{g^{n+1}(\xi) \prod_{i=0}^{n} (x - x_i)}{(n+1)!}$$

, where $\xi \in int(x_0, x_1, \dots, x_n, x)$

The Secant Method: Order of Convergence

Let $int(x_0, x_1, \ldots, x_n)$: the smallest interval that contains $x_0, \ldots, x_n$

According to Newton's interpolation formula, we have

$$g(x) = g(x_n) + (x - x_n)g[x_{n-1}, x_n] + \frac{1}{2}(x - x_n)(x - x_{n-1})g''(\xi)$$

where $g[x_{n-1}, x_n] = \dfrac{g(x_n) - g(x_{n-1})}{x_n - x_{n-1}}$, and $\xi \in int(x, x_n, x_{n-1})$

By the Secant Method, we have

$$x_{n+1} = x_n - g(x_n)\frac{x_n - x_{n-1}}{g(x_n) - g(x_{n-1})} \Rightarrow$$

$$0 = g(x_n) + (x_{n+1} - x_n)g[x_{n-1}, x_n]$$

The Secant Method: Order of Convergence

Let the root the Secant Method approaches be c, then

$$0 = g(c) - g(x_n) - (x_{n+1} - x_n)g[x_{n-1}, x_n] =$$

$$g[x_{n-1}, x_n](c - x_{n+1}) + \frac{1}{2}(c - x_n)(c - x_{n-1})g''(\xi)$$

By the mean value theorem, we have

$$g[x_{n-1}, x_n] = g'(\eta), \eta \in (x_{n-1}, x_n)$$

Let $\epsilon_n = c - x_n$, we get $0 = g'(\eta)\epsilon_{n+1} + \frac{1}{2}\epsilon_n\epsilon_{n-1}g''(\xi) \Rightarrow$
$\epsilon_{n+1} = \frac{g''(\xi)}{2g'(\eta)}\epsilon_n\epsilon_{n-1}$

The Secant Method: Order of Convergence

Now suppose the Secant Method converges, then when $n \to \infty$, $\xi \approx c$ and $\eta \approx c$. Let $C = |\frac{g''(c)}{2g'(c)}|$, then $|\epsilon_{n+1}| = C|\epsilon_n||\epsilon_{n-1}|$

To find the order of convergence, we find M and p such that $|\epsilon_{n+1}| \approx M|\epsilon_n|^p$, where p is the convergence order.
$M|\epsilon_n|^p = C(M|\epsilon_{n-1}|^p)|\epsilon_{n-1}| \Rightarrow |\epsilon_n| = C|\epsilon_{n-1}|^{(1+p)/p}$
This implies $p = (1+p)/p \Rightarrow p = 1 + \phi \approx 1.618$

Since the exponent 1.618 lies between 1 (linear convergence) and 2 (quadratic convergence), the convergence rate of the Secant Method is called *superlinear*.

Illinois Algorithm: Order of Convergence

Recall the errors of the Secant Method satisfy:

$$\epsilon_{n+1} = \frac{g''(\xi)}{2g'(\eta)}\epsilon_n\epsilon_{n-1}.$$

If g'' is continuous, then when the Illinois Algorithm gets into a sufficient small neighborhood of the root c, we can assume g' and g'' have constant sign.

This implies that $\frac{\epsilon_{n+1}}{\epsilon_n\epsilon_{n-1}}$ also has constant sign.

Since $g_{n-1}g_n < 0 \Rightarrow \epsilon_{n-1}\epsilon_n < 0$, we then necessarily have the sign of ϵ_n's follow one of the two schemes:

$$\cdots + - + + - + + - + \dots$$

or

$$\cdots + - - + - - + - - \dots$$

Illinois Algorithm: Order of Convergence

Previous analysis shows that asymptotically, an end-point will be retained twice in consecutive three iterations.
In other words, we will perform the Illinois step (halving the function value) once every third time.
Further asymptotic analysis shows that an Illinois step has
$\epsilon_{n+1} \approx -\epsilon_n$
Putting the pieces together, we have

$$\epsilon_n = -\epsilon_{n-1} \Rightarrow \epsilon_{n+1} = -C\epsilon_{n-1}^2 \Rightarrow \epsilon_{n+2} = C^2\epsilon_{n-1}^3$$

Via finding p such that $|\epsilon_n| = M|\epsilon_{n-1}|^p$, we get
$\frac{3}{p^2} = p \Rightarrow p \approx 1.44$