Open domain methods

- Newton's iteration.
- Method of secants.
- Muller's method.
- Fixed point iteration.

Open domain methods

- Methods do not restrict the root to remain trapped in a closed interval.

Open domain methods

- Methods do not restrict the root to remain trapped in a closed interval.
- As a result, these methods are not as *robust* as bracketing methods – these methods may not converge at all.

Open domain methods

- Methods do not restrict the root to remain trapped in a closed interval.
- As a result, these methods are not as *robust* as bracketing methods – these methods may not converge at all.
- They use information about the non linear function to refine the estimates of the root – can be considerably more efficient than bracketing methods.

Newton's method

- This method is also called the Newton–Raphson method.

Newton's method

- This method is also called the Newton–Raphson method.
- This method uses the derivatives $f'(x)$ of the function $f(x)$ to accelerate convergence for solving $f(x) = 0$.

Newton's method

- This method is also called the Newton–Raphson method.
- This method uses the derivatives $f'(x)$ of the function $f(x)$ to accelerate convergence for solving $f(x) = 0$.
- It always converges *if the initial approximation is sufficiently close to the root*. Its rate of convergence is quadratic!

Newton's method

- This method is also called the Newton–Raphson method.
- This method uses the derivatives $f'(x)$ of the function $f(x)$ to accelerate convergence for solving $f(x) = 0$.
- It always converges *if the initial approximation is sufficiently close to the root*. Its rate of convergence is quadratic!
- However, it not only needs $f(x)$, it also needs the derivative $f'(x)$.

Newton's method: Key idea

A continuous function $f(x)$ can be expanded around a point x is terms of a Taylor's series:

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{(x - x_0)^2}{2!}f''(x_0) + \dots$$

Newton's method: Key idea

A continuous function $f(x)$ can be expanded around a point x is terms of a Taylor's series:

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{(x - x_0)^2}{2!}f''(x_0) + \dots$$

As we want the solution for $f(x) = 0$, we can write the following:

$$f(x) = 0 = f(x_0) + (x - x_0)f'(x_0)$$

neglecting higher order terms.

Newton's method: Key idea

A continuous function $f(x)$ can be expanded around a point x is terms of a Taylor's series:

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{(x - x_0)^2}{2!}f''(x_0) + \dots$$

As we want the solution for $f(x) = 0$, we can write the following:

$$f(x) = 0 = f(x_0) + (x - x_0)f'(x_0)$$

neglecting higher order terms.
Using this we can write:

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

Newton's method: Algorithm

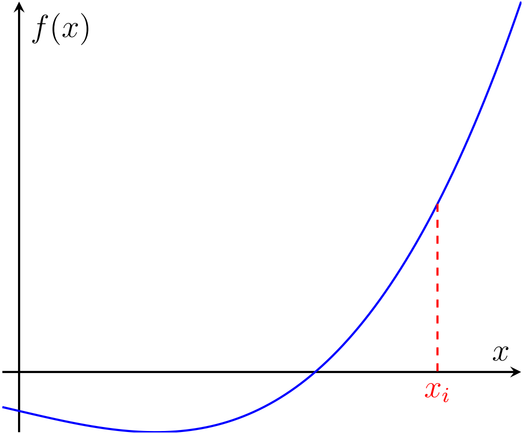
The iterations use $f(x)$ and $f'(x)$ to proceed in the following way:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Newton's method: Algorithm

The iterations use $f(x)$ and $f'(x)$ to proceed in the following way:

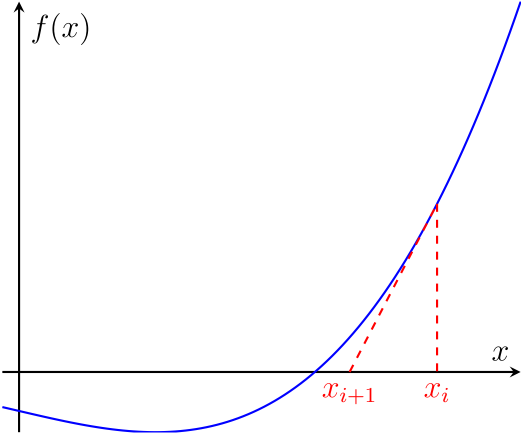
$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$



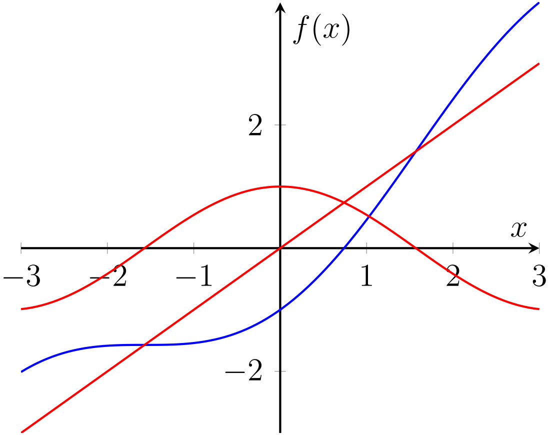
Newton's method: Algorithm

The iterations use $f(x)$ and $f'(x)$ to proceed in the following way:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$



Consider an example: $f(x) = x - \cos(x)$



Initial guess $x = 1.0$.

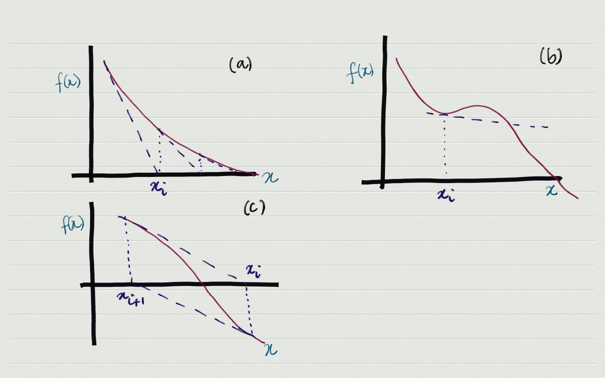
Newton's method: example

i	x_i	f(x_i)	f'(x_i)	x_{i+1}	f(x_{i+1})
1	1.000000	0.459698	1.841471	0.750364	0.018923
2	0.750364	0.018923	1.681905	0.739113	0.000046
3	0.739113	0.000046	1.673633	0.739085	0.000000

```
Root is: 0.739085133385
```

- False position used 4 iterations and bisection – 22 iterations!
- Newton's method has excellent local convergence properties.
- However, global convergence properties can be quite poor – due to neglect of higher order terms in the Taylor series expansion.

Possible problems



(a) Very slow approach to the root – $f'(x) \to 0$ near the root.
(b) Difficulty with local minima – may send the next iteration x_{i+1} very far from the root.
(c) Lack of convergence for asymmetric functions: $f(a + x) = -f(a - x)$

Newton's method: General comments

- It is an excellent method for polishing roots obtained from other methods!

Newton's method: General comments

- It is an excellent method for polishing roots obtained from other methods!
- The method not only requires the function, but also its derivative. Besides being an additional computational expense, in some cases, it might not be possible to calculate the derivative easily...

Newton's method: General comments

- It is an excellent method for polishing roots obtained from other methods!
- The method not only requires the function, but also its derivative. Besides being an additional computational expense, in some cases, it might not be possible to calculate the derivative easily...
- If the derivative becomes small – the method may end up not converging – as the next step might be far away from the root.

Method of secants

- The secant method is a variant of Newton's method – for the case when the evalulation of the derivative is difficult/impossible.

Method of secants

- The secant method is a variant of Newton's method – for the case when the evalulation of the derivative is difficult/impossible.
- The non linear function $f(x)$ is aproximated locally by the linear function $g(x)$, which is the secant to $f(x)$, and the root of $g(x)$ is taken as an improved approximation to the root of $f(x)$.

Method of secants: Algorithm

The derivative of the function $f(x)$ at point x_i can be approximated as:

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$$

Method of secants: Algorithm

The derivative of the function $f(x)$ at point x_i can be approximated as:

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$$

Then using Newton's method:

$$\begin{aligned} x_{i+1} &= x_i - \frac{f(x_i)(x_i - x_{i-1})}{f(x_i) - f(x_{i-1})} \\ &= \frac{x_{i-1} f(x_i) - x_i f(x_{i-1})}{f(x_i) - f(x_{i-1})} \end{aligned}$$

Method of secants: Algorithm

The derivative of the function $f(x)$ at point x_i can be approximated as:

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$$

Then using Newton's method:

$$\begin{aligned} x_{i+1} &= x_i - \frac{f(x_i)(x_i - x_{i-1})}{f(x_i) - f(x_{i-1})} \\ &= \frac{x_{i-1}f(x_i) - x_i f(x_{i-1})}{f(x_i) - f(x_{i-1})} \end{aligned}$$

- Needs two initial points to start.

Method of secants: Algorithm

The derivative of the function $f(x)$ at point x_i can be approximated as:

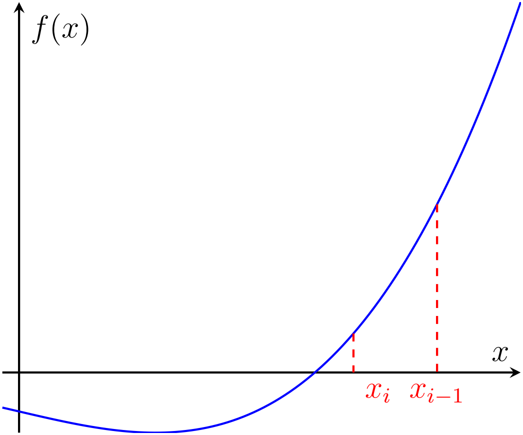
$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$$

Then using Newton's method:

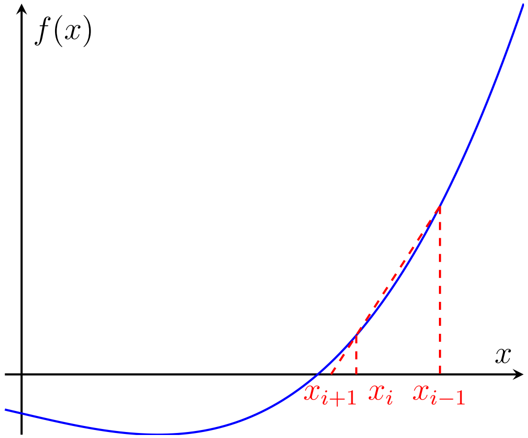
$$\begin{aligned} x_{i+1} &= x_i - \frac{f(x_i)(x_i - x_{i-1})}{f(x_i) - f(x_{i-1})} \\ &= \frac{x_{i-1} f(x_i) - x_i f(x_{i-1})}{f(x_i) - f(x_{i-1})} \end{aligned}$$

- Needs two initial points to start.
- Same as False position method – except for the points used!

Method of secants: Algorithm



Method of secants: Algorithm



Method of secants: example

i	x_i	f(x_i)	x_{i-1}	f(x_{i-1})	x_{i+1}	f(x_{i+1})
1	1.000000	0.459698	1.500000	1.429263	0.762936	0.040126
2	0.762936	0.040126	1.000000	0.459698	0.740264	0.001974
3	0.740264	0.001974	0.762936	0.040126	0.739091	0.000010
4	0.739091	0.000010	0.740264	0.001974	0.739085	0.000000

```
Root is: 0.73908513481
```

- False position used 4 iterations and bisection – 22 iterations!
- Newton's method took 3 iterations.
- However, the question as to which method is more efficient depends not just on the number of iterations – as in the Newton's iteration, one also has to evaluate the derivative!

Muller's method

- The Muller's method is a variant of method of secants.

Muller's method

- The Muller's method is a variant of method of secants.
- The non linear function $f(x)$ is aproximated locally by the *quadratic* function $g(x)$, and the root of $g(x)$ is taken as an improved approximation to the root of $f(x)$.

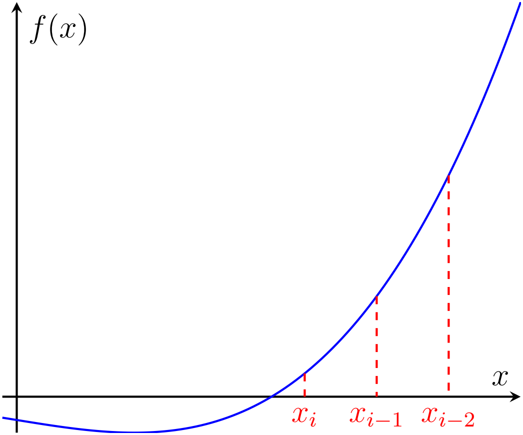
Muller's method

- The Muller's method is a variant of method of secants.
- The non linear function $f(x)$ is aproximated locally by the *quadratic* function $g(x)$, and the root of $g(x)$ is taken as an improved approximation to the root of $f(x)$.
- The only difference between Muller's method and method of secants is that the $g(x)$ is quadratic function in Muller's method and linear function in secant method!

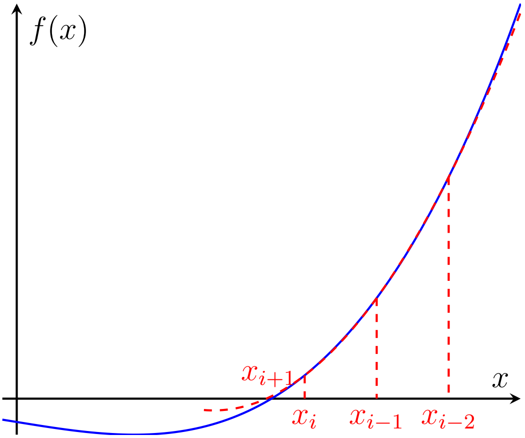
Muller's method

- The Muller's method is a variant of method of secants.
- The non linear function $f(x)$ is aproximated locally by the *quadratic* function $g(x)$, and the root of $g(x)$ is taken as an improved approximation to the root of $f(x)$.
- The only difference between Muller's method and method of secants is that the $g(x)$ is quadratic function in Muller's method and linear function in secant method!
- Three initial approximations are required to start the algorithm (as opposed to two in secant method).

Muller's method: Algorithm



Muller's method: Algorithm



Fixed point iteration

Suppose that you can bring an equation $g(x) = 0$ in the form $x = f(x)$. This fixed point equation, under certain conditions, can be solved using iteration.

Fixed point iteration

Suppose that you can bring an equation $g(x) = 0$ in the form $x = f(x)$. This fixed point equation, under certain conditions, can be solved using iteration.

- Start with an approximation x_0 of the root.

Fixed point iteration

Suppose that you can bring an equation $g(x) = 0$ in the form $x = f(x)$. This fixed point equation, under certain conditions, can be solved using iteration.

- Start with an approximation x_0 of the root.
- Calculate x_0, x_1 $\ldots$ x_n such that

$$x_1 = f(x_0)$$
$$x_2 = f(x_1)$$
$$x_3 = f(x_2)$$

Fixed point iteration

Suppose that you can bring an equation $g(x) = 0$ in the form $x = f(x)$. This fixed point equation, under certain conditions, can be solved using iteration.

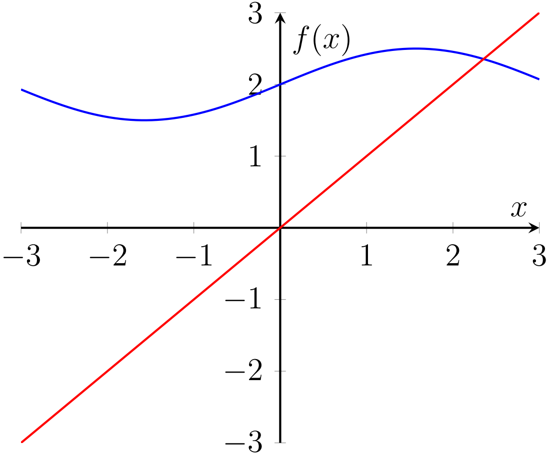
- Start with an approximation x_0 of the root.
- Calculate x_0, x_1 ... x_n such that

$$x_1 = f(x_0)$$
$$x_2 = f(x_1)$$
$$x_3 = f(x_2)$$

- If the sequence x_0, x_1 ... x_n belongs to an interval I, where $|f'(x)| < k < 1$ then the sequence has a limit L and L is the only root of $x = f(x)$ in the interval I.

Fixed point iteration: example

Consider an example: $f(x) = 2 + \frac{\sin(x)}{2}$
$|f'(x)| = |\frac{\cos(x)}{2}| < 1$.



Initial guess $x = 2.0$.

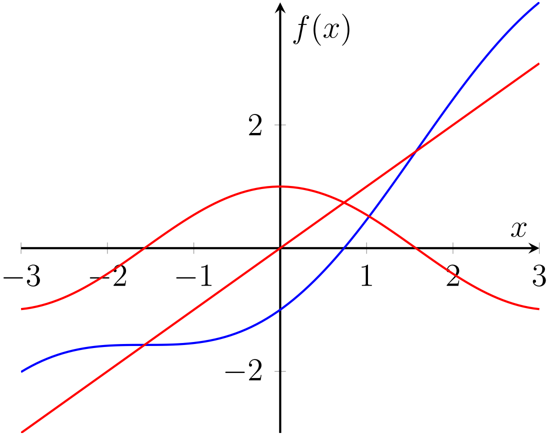
Fixed point iteration: example

i	x_i	f(x_i)
1	2.000000	2.454649
2	2.454649	2.317089
3	2.317089	2.367106
4	2.367106	2.349675
5	2.349675	2.355851
6	2.355851	2.353675
7	2.353675	2.354443
8	2.354443	2.354172
9	2.354172	2.354268
10	2.354268	2.354234
11	2.354234	2.354246
12	2.354246	2.354242
13	2.354242	2.354243

```
Root is: 2.35424314498
```

Fixed point iteration: Example

Consider an example: $f(x) = x - \cos(x)$



Initial guess $x = 1.0$.

Fixed point iteration: example

```
 i     x_i       f(x_i)
 1  1.000000  0.540302
 2  0.540302  0.857553
 3  0.857553  0.654290
 4  0.654290  0.793480
 5  0.793480  0.701369
 6  0.701369  0.763960
 7  0.763960  0.722102
 8  0.722102  0.750418
 9  0.750418  0.731404
10  0.731404  0.744237
11  0.744237  0.735605
12  0.735605  0.741425
13  0.741425  0.737507
14  0.737507  0.740147
15  0.740147  0.738369
16  0.738369  0.739567
17  0.739567  0.738760
18  0.738760  0.739304
19  0.739304  0.738938
20  0.738938  0.739184
21  0.739184  0.739018
22  0.739018  0.739130
23  0.739130  0.739055
24  0.739055  0.739106
25  0.739106  0.739071
26  0.739071  0.739094
27  0.739094  0.739079
28  0.739079  0.739089
29  0.739089  0.739082
30  0.739082  0.739087
31  0.739087  0.739084
32  0.739084  0.739086
33  0.739086  0.739085
Root is: 0.739084549575
```

Fixed point iteration

- We want a root of the function $g(x) = 0$ and we know a first approximation x_0 of the root.

Fixed point iteration

- We want a root of the function $g(x) = 0$ and we know a first approximation x_0 of the root.
- We write $x = x + rg(x)$

Fixed point iteration

- We want a root of the function $g(x) = 0$ and we know a first approximation x_0 of the root.
- We write $x = x + rg(x)$
- We choose an r-value such that $1 + rg'(x_0) \approx 0$. Let $r_0 = \text{the chosen value.}$

Fixed point iteration

- We want a root of the function $g(x) = 0$ and we know a first approximation x_0 of the root.
- We write $x = x + rg(x)$
- We choose an r-value such that $1 + rg'(x_0) \approx 0$. Let $r_0 = \text{the chosen value.}$
- We apply iteration on $x = x + r_0 g(x)$ starting with $x = x_0$.

Fixed point iteration: example

We rewrite our function as $x = x - 0.54 * (x - \cos(x))$.

```
i    x_i       f(x_i)
1  1.000000  0.751763
2  0.751763  0.740273
3  0.740273  0.739199
4  0.739199  0.739096
5  0.739096  0.739086
6  0.739086  0.739085

Root is: 0.73908523492
```

- Huge difference in terms of the efficiency!!! From 33 iterations to 6 iterations.
- Careful choice/massaging of the equation can yield remarkable gains.

Open domain methods: General remarks

- All these methods converge rapidly in the vicinity of the root. When the derivative is difficult to compute or is time-consuming, the secant method is very efficient.

Open domain methods: General remarks

- All these methods converge rapidly in the vicinity of the root. When the derivative is difficult to compute or is time-consuming, the secant method is very efficient.
- In sensitive cases, all these methods may misbehave! It may be required to use bracketing methods.

Open domain methods: General remarks

- All these methods converge rapidly in the vicinity of the root. When the derivative is difficult to compute or is time-consuming, the secant method is very efficient.
- In sensitive cases, all these methods may misbehave! It may be required to use bracketing methods.
- Plotting of the functions can help in indetifying such cases.

Open domain methods: General remarks

- All these methods converge rapidly in the vicinity of the root. When the derivative is difficult to compute or is time-consuming, the secant method is very efficient.
- In sensitive cases, all these methods may misbehave! It may be required to use bracketing methods.
- Plotting of the functions can help in indetifying such cases.
- *All of these methods can find complex roots simply by using complex arithmetic!*