**Secant method**,

Secant method is an iterative tool of mathematics and numerical methods to find the approximate root of polynomial equations. During the course of iteration, this method assumes the function to be approximately linear in the region of interest.

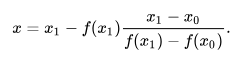
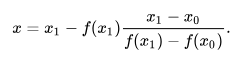
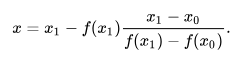
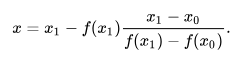
Although secant method was developed independently, it is often considered to be a finite difference approximation of [Newton’s method](https://www.codewithc.com/newton-raphson-method-matlab-program/). But, being free from derivative, it is generally used as an alternative to the latter method.

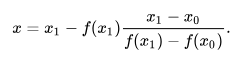
the **secant method** is a [root-finding algorithm](https://en.wikipedia.org/wiki/Root-finding_algorithm)that uses a succession of [roots](https://en.wikipedia.org/wiki/Root_of_a_function) of [secant lines](https://en.wikipedia.org/wiki/Secant_line) to better approximate a root of a [function](https://en.wikipedia.org/wiki/Function_(mathematics)) *f*. The secant method can be thought of as a [finite-difference](https://en.wikipedia.org/wiki/Finite-difference) approximation of [Newton's method](https://en.wikipedia.org/wiki/Newton%27s_method). However, the method was developed independently of Newton's method and predates it by over 3000 years.

Derivation of the method

Starting with initial values *x*0 and *x*1, we construct a line through the points (*x*0, *f*(*x*0)) and (*x*1, *f*(*x*1)), as shown in the picture above. In slope–intercept form, the equation of this line is

The root of this linear function, that is the value of *x* such that *y* = 0 is

{\displaystyle x=x\_{1}-f(x\_{1}){\frac {x\_{1}-x\_{0}}{f(x\_{1})-f(x\_{0})}}.}



We then use this new value of *x* as *x*2 and repeat the process,

if *f*(*x0*) and *f*(*x2*) have opposite signs, then the method sets *x*2as the new value for *x*1, and if *f*(*x*1) and *f*(*x*2) have opposite signs then the method sets *x*2as the new *x*0. ,so the method is applicable to this smaller interval

We continue this process, solving for *x*3, *x*4, etc., until we reach a sufficiently high level of precision (a sufficiently small difference between *xn*and *xn*−1):

A computational example

The secant method is applied to find a root of the function *f*(*x*) = *x*2 − 612. Here is an implementation in the [MATLAB](https://en.wikipedia.org/wiki/MATLAB" \o "MATLAB)language (from calculation, we expect that the iteration converges at *x* = 24.7386):

f=@(x) x^2 - 612;

x(1)=10;

x(2)=30;

**for** i=3:7

x(i) = x(i-1) - (f(x(i-1)))\*((x(i-1) - x(i-2))/(f(x(i-1)) - f(x(i-2))));

**end**

root=x(7)

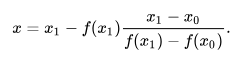
الطريقة الثانية secant method

في **التحليل العددي ، طريقة القاطع هو خوارزمية لتقصي جذور يستخدم سلسلة من الجذور من خط قاطع لتقريب أفضل جذر وظيفة و يمكن اعتبار الطريقة الثانية تقريبًا للفرق المحدود لطريقة نيوتن . ومع ذلك تم تطوير هذه الطريقة بشكل مستقل عن طريقة نيوتن وتسبقها بأكثر من 3000 عام.**

## الطريقة ]

يتم تعريف الطريقة الثانية بعلاقة التكرار

ما يتضح من علاقة التكرار ، فإن الطريقة الثانية تتطلب قيمتين أوليتين ، *x*0 و *x*1، والتي يجب اختيارها بشكل مثالي للاقتراب من الجذر



% Secant Method in MATLAB

a=input('Enter function:','s');

f=inline(a)

x(1)=input('Enter first point of guess interval: ');

x(2)=input('Enter second point of guess interval: ');

n=input('Enter allowed Error in calculation: ');

iteration=0;

for i=3:1000

x(i) = x(i-1) - (f(x(i-1)))\*((x(i-1) - x(i-2))/(f(x(i-1)) - f(x(i-2))));

iteration=iteration+1;

if abs((x(i)-x(i-1))/x(i))\*100<n

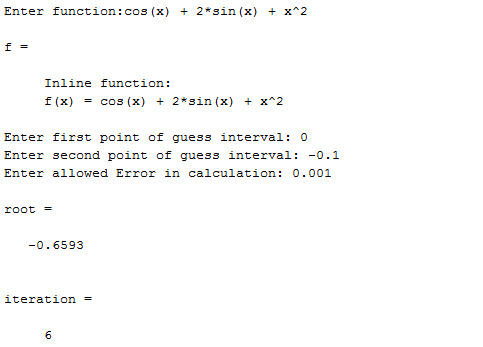
root=x(i)

iteration=iteration

break

end

end



{\ displaystyle x\_ {n} = x\_ {n-1} -f (x\_ {n-1}) {\ frac {x\_ {n-1} -x\_ {n-2}} {f (x\_ {n-1} }) - f (x\_ {n-2})}} = {\ frac {x\_ {n-2} f (x\_ {n-1}) - x\_ {n-1} f (x\_ {n-2}) } {و (X\_ {ن 1}) - و (X\_ {ن 2})}}}

Lets perform a numerical analysis of the above program of **secant method in MATLAB**. The same function f(x) is used here; x0 =0 and x1= -0.1 are taken as initial approximation, and the allowed error is 0.001.

Here,

f(*x*) = cos(*x*) + 2 sin(*x*) + *x*2x0 = 0  
x1 = -0.1

For **first iteration**,

f(x1) = cos(-0.1) + 2 sin(-0.1) + ( -0.1 )2 = 0.8053 and  
f(x0) = cos0 + 2 sin0 + 02= 1

As we know,

x2 = x1 – f(x1)  
x2= 0 + 0.8053 \* (-0.1-0)/(0.8053-1)  
x2 =0.4136

Similarly, x3 = – 0.5136, and so on…

The complete calculation and iteration of secant method (and MATLAB program) for the given function is presented in the table below:

## secant-method-matlab-iteration.png (623Ã268)

## Thus, the root of f(x) = cos(x) + 2 sin(x) + x2 as obtained from secant method as well as its MATLAB program is -0.6595.

## **Check**: f(-0.6585) = cos(-0.6585) + 2 sin(-0.6585) + (-0.6585)2 = 0.0002 (OK).

## Example2

As an example of the secant method, suppose we wish to find a root of the function f(*x*) = cos(*x*) + 2 sin(*x*) + *x*2. A closed form solution for *x* does not exist so we must use a numerical technique. We will use *x*0 = -0.1 and *x*1 = 0 as our initial approximations. We will let the two values εstep = 0.001 and εabs = 0.001 and we will halt after a maximum of *N* = 100 iterations.

We will use four decimal digit arithmetic to find a solution and the resulting iteration is shown in Table 1.

Table 1. The secant method applied to f(*x*) = cos(*x*) + 2 sin(*x*) + *x*2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***n*** | ***xn*− 1** | ***xn*** | ***xn*+ 1** | **|f(*xn*+ 1)|** | **|*xn*+ 1 - *xn*|** |
| 1 | 0.0 | -0.1 | -0.5136 | 0.1522 | 0.4136 |
| 2 | -0.1 | -0.5136 | -0.6100 | 0.0457 | 0.0964 |
| 3 | -0.5136 | -0.6100 | -0.6514 | 0.0065 | 0.0414 |
| 4 | -0.6100 | -0.6514 | -0.6582 | 0.0013 | 0.0068 |
| 5 | -0.6514 | -0.6582 | -0.6598 | 0.0006 | 0.0016 |
| 6 | -0.6582 | -0.6598 | -0.6595 | 0.0002 | 0.0003 |

## Thus, with the last step, both halting conditions are met, and therefore, after six iterations, our approximation to the root is -0.6595.

## References

* Kaw, Autar; Kalu, Egwu (2008), [*Numerical Methods with Applications*](http://www.autarkaw.com/books/numericalmethods/index.html) (1st ed.).
* Allen, Myron B.; Isaacson, Eli L. (1998). [*Numerical analysis for applied science*](https://books.google.com/books?id=PpB9cjOxQAQC). [*John Wiley & Sons*](https://en.wikipedia.org/wiki/John_Wiley_%26_Sons). pp. 188–195. [*ISBN*](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [*978-0-471-55266-6*](https://en.wikipedia.org/wiki/Special:BookSources/978-0-471-55266-6).