O Método de Newton

Este procedimento dá uma sequência que aproxima as raizes de uma função dada.

Sintaxe: Newton(f,x=a)

PARÂMETROS: f - a função cujas raizes procuramos

x -a variável independente de f`,

a - o candidato inicial para uma raiz

Resumo: se a sequência converge para um valor com 10 digitos de precisão o procedimento pára e mostra a sequência. Se a sequência falha na convergência com 10 iterações o procedimento pára e informa.

Execute o procedimento e faça os exemplos.

O Procedimento (execute-o)

```
> Newton := proc(func::{algebraic,procedure},start::name=constant)
> local var, pt, Func, n, last, root, dFunc;
> var := op(1,start);
> pt := op(2,start);
> if type(func,procedure) then
> Func := func
> else
> if member(var,indets(func,name)) then
> if nops(indets(func,name)) <> 1 then
> ERROR(`the first argument must have only one indeterminant.`) fi;
```

```
> Func := traperror(unapply(func,var));
> if Func=lasterror then
> ERROR(`unable to construct a function from the first argument.`) fi;
> else
> ERROR(`second argument variable not present in the first.`);
> fi;
> fi;
> dFunc := diff(Func(var),var);
> last := pt+5;
> root := pt;
> for n to 100 while evalf(abs(last-root))>10^(-8) do
> last := root;
> root := evalf(last-Func(last)/subs(var=last,dFunc));
> print(root);
> od;
> if n<100 then
> root
> else
> print(` O método de Newton falhou para convergência com 100 iterações.`)
> fi;
> end:
>
```

Exemplos

```
> Newton(exp(x)-3*x,x=2 );
```

1.683518263

1.543481972

1.025325929

1.000908452

1.000001235

1.000000000

1.512134551

> **Newton(cos(x)-3*x,x=2);**

1.000000000

.3169984800

.3167508376

.3167508288

.3167508288

> **Newton(2*sin(x)-x*cos(x),x=5);**

1.512134552

4.274826749

1.512134551

.358746819

.3169984800

> **#Newton(x^2+1,x=4);**

>