Limites de suites récurrentes

I. Cas général d'une suite récurrente

Objectif : calculer une approximation numérique de la limite d'une suite récurrente

$$x_{k+1} = f(x_k)$$

Données

La valeur initiale $x_0 = a$, la fonction f.
Le nombre de décimales exactes attendues N.

Description de l'algorithme

Implémentation MAPLE

```
> newton:=proc(a,N)

local x,y;
global f;

x:=a;

do                                                      #boucle sans condition
        y:=f(x);
        if abs(y-x)<10^(-N) then return evalf(x,N) end if;  #condition de sortie de la boucle
        x:=y
end do;

end proc;
```

II. Méthode de Newton

Objectif : calculer une approximation numérique d'un zéro d'une fonction dérivable

Données :

Deux valeurs du paramètre *a*, *b* telles que $f(a)f(b)<0$.
Le nombre de décimales exactes attendues N.

Description de l'algorithme

On calcule les valeurs de la suite définie par récurrence par

$$x_{k+1} = x_k - f(x_k)/f'(x_k)$$

L'algorithme s'arrête lorsque l'écart entre deux termes consécutifs est <10^(-N).

Implémentation MAPLE

```
> newton:=proc(a,N)
local x,y;
global f;

x:=a;

do  #boucle sans condition
        y:=x - f(x)/D(f)(x);
        if abs(y-x)<10^(-N) then return evalf(x,N) end if; #condition de sortie de la boucle
x:=y
end do;

end proc;

> f:=x->x^3-2*x^2+2;
```

$$f := x \rightarrow x^3 - 2\,x^2 + 2$$

```
> newton(-1,15);
```

$$-.839286755214161$$

```
> f(%);
```

$$0.$$