TD 1 : Etudes de suites

Jean-Sébastien ROY, 1999

Ex 1 : Outils élémentaires d'étude d'une suite

On considère la suite u(i)=1/i^2, pour i>0

```
> restart;
```

Définition

```
> u := i -> 1/i^2;
```

$$u := i \to \frac{1}{i^2}$$

Calcul des premiers termes

```
> seq(u(i),i=1..10);
```

$$1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}, \frac{1}{25}, \frac{1}{36}, \frac{1}{49}, \frac{1}{64}, \frac{1}{81}, \frac{1}{100}$$

Monotonie

```
> d:=simplify(u(i)-u(i-1));
```

$$d := -\frac{2\,i-1}{i^2\,(i-1)^2}$$

A noter : assume / is ne marchent quasiment JAMAIS. Ici, c'est juste pour le principe.

```
> assume(i>=1);
> is(d<0);
```

true

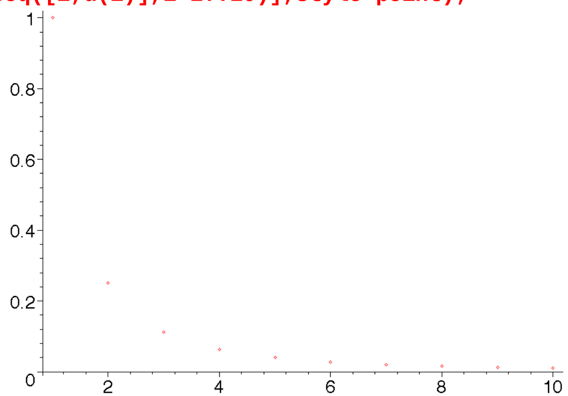
On efface l'assume

```
> i:='i';
```

$$i := i$$

Trace l'evolution de la suite (bien noter la position des crochets)

```
> plot([seq([i,u(i)],i=1..10)],style=point);
```



Limite de la suite

```
> limit(u(i),i=infinity);
```

$$0$$

Ex 2 : Convergence d'une série

```
> restart:
```

Tout d'abord, on définit une fonction renvoyant le terme général du DSE de ln(1+x) :

```
> terme := (i,x) -> (-1)^(i+1)*x^i/i;
```

$$terme := (i, x) \to \frac{(-1)^{(i+1)}\,x^i}{i}$$

Puis on défini une fonction renvoyant la somme des ***n*** premiers du DSE en ***x*** :

```
> approxLn := proc(n,x) local
  i;add(evalf(terme(i,x)),i=1..n);end;
```

approxLn := **proc**(*n*, *x*) **local** *i*; add(evalf(terme(*i*, *x*)), *i* = 1 .. *n*) **end**

A noter : on a utilisé **add** et non pas **sum**. Le premier effectue la somme numérique terme a terme, alors que le second est adapté aux calculs symboliques (quand ***n*** n'est pas connu par exemple).
De plus, on effectue un **evalf** sur le terme à ajouter, afin de ne jamais conserver de fractions, qui ralentiraient le calcul.
Enfin, l'utilisation d'une variable locale (**local**) n'est pas obligatoire, mais préférable avec **add** (raisons obscures).

L'approximation de ln(2) fournie par Maple :

```
> evalf(ln(2));
```

.6931471806

Puis la notre avec 10000 termes :

```
> approxLn(10000,1);
```

.6930971837

Utiliser le DSE de ln((1+x)/(1-x)) revient a utiliser le DSE de ln(1+x) et ln(1-x) en x tq (1+x)/(1-x)=2
On resoud :

```
> solve((1+x)/(1-x)=2);
```

$$\frac{1}{3}$$

On applique notre formule avec 17 termes :

```
> approxLn(17,1/3)-approxLn(17,-1/3);
```

.6931471804

Le resultat est bien meilleur. L'explication tient dans le terme en x^i du DSE. Quand x=1/3, il tend très rapidement vers 0, ce qui n'est pas le cas quand x=1.

Ex 3 : Récurrence linéaire

```
> restart;
```

Definition d'une suite recurrente :

'option remember' rends le calcul beaucoup plus rapide, en memorisant toutes les valeurs calculees de la suite, ce qui, pour une suite doublement récursive, évite de calculer de nombreuses fois chaque valeur.

Attention à l'ordre et à la syntaxe ; d'abord f:=proc ... et ensuite la definition des valeurs initiales.

```
> f:=proc(n) option remember; 10*f(n-1)-16*f(n-2); end:
> f(1):=20:f(2):=-8:
```

Calcul des premiers termes

```
> seq(f(i),i=1..5);
```

20, -8, -400, -3872, -32320

Resolution de la recurence (utiliser un nom different pour la suite (ici u à la place de f), sinon maple cherche a calculer sa valeur symboliquement)

```
> rsolve({u(n)=10*u(n-1)-16*u(n-2),u(1)=20,u(2)=-8},u(n));
```

$$-8^n + 14\,2^n$$

Definition non recursive (ne pas mettre de -> quand on utilise unapply)

```
> u:= unapply(%,n);
```

$$u := n \rightarrow -8^n + 14\,2^n$$

Calcul de la limite

```
> limit(u(n),n=infinity);
```

$$-\infty$$

Calcul d'un equivalent en +infinity (ne pas oublier de mettre un ordre)

```
> asympt(u(n),n,2);
```

$$-8^n + O(2^n)$$

Ex 4.1 : Récurence du type u(n+1)=f(u(n)), exemple 1

```
> restart;
> f:= x -> sqrt(3*x-2);
```

$$f := x \rightarrow \sqrt{3\,x-2}$$

Ici 'option remember' n'est pas vraiment necessaire (la suite n'est pas doublement recursive)

```
> u:= n -> f(u(n-1));
```

$$u := n \rightarrow \mathrm{f}(\mathrm{u}(n-1))$$

```
> u(0):=1.5;
```

$$\mathrm{u}(0) := 1.5$$

```
> u(3);
```

1.723072433

```
> seq(u(i),i=1..10);
```

1.581138830, 1.656326203, 1.723072433, 1.780229564, 1.827755096, 1.866350794, 1.897116860, 1.921288781, 1.940068644, 1.954534710

La fonction marche renvoie les extremites d'un petit segment vertical.

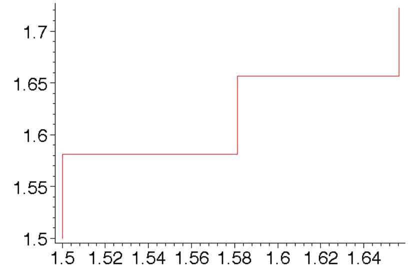
```
> marche:= i -> ([u(i),u(i)],[u(i),u(i+1)]):
```

En tracant plusieurs marche on cree l'escalier. (bien noter la position des crochets, comme toujours)

```
> seq(marche(i),i=0..2);
```

[1.5, 1.5], [1.5, 1.581138830], [1.581138830, 1.581138830], [1.581138830, 1.656326203], [1.656326203, 1.656326203], [1.656326203, 1.723072433]

```
> plot([%]);
```

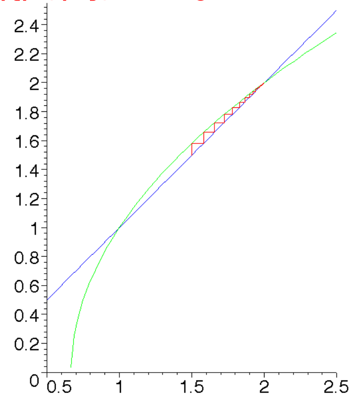



En nomant chacun des graphes (avec un ':' a la fin des expressions pour eviter l'horrible charabia de maple). Et en utilisant plots[display] on arrive a superposer des graphes.

```
> p1:=plot([seq(marche(i),i=0..20)]):
```

Avec [f(x),x] on trace simultanement y=f(x) et y=x.

```
> p2:=plot([f(x),x],x=0.5..2.5,color=[green,blue]):
> plots[display]({p1,p2},scaling=constrained);
```



Les points fixes :

```
> solve(f(x)=x,x);
```

1, 2

D(f) represente la fonction derivee de la fonction f.

```
> D(f);
```

$$x \rightarrow \frac{3}{2}\frac{1}{\sqrt{3\,x-2}}$$

```
> evalf(D(f)(2));
```

.7500000000

Le point x=2 est attractif.

```
> evalf(D(f)(1));
```

1.500000000

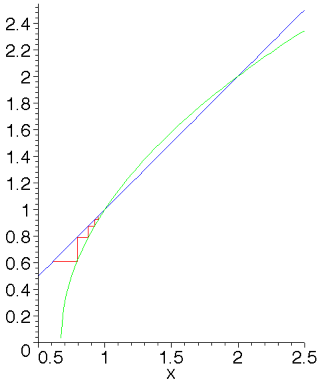
Et x=1 est repulsif.

On choisit une nouvelle valeur initiale pour la suite.

```
> u(0):=0.95;
```

$$\mathrm{u}(0) := .95$$

```
> p3:=plot([seq(marche(i),i=0..20)]):
> plots[display]({p3,p2},scaling=constrained);
```



A partir de quelques termes, la suite n'est plus définie.

Ex 4.2 : Récurrence du type u(n+1)=R*u(n)*(u(n)-1)

Pour le corrigé de cet exercice, on se référera au corrigé du TD sur les suites de 1998 (voir les 5/2).

Ex 4.3 : Récurrence du type u(n+1)=f(u(n))

```
> restart;
> f:=x->(1+x^2)/2;
```

$$f := x \rightarrow \frac{1}{2} + \frac{1}{2}x^2$$

```
> u:= n -> f(u(n-1));
```

$$u := n \rightarrow f(u(n-1))$$

Quelles sont les limites possibles de la suite si elle converge ?

```
> solve(f(x)=x,x);
```

$$1, 1$$

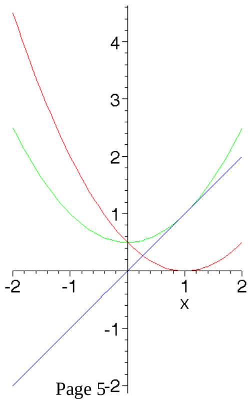
1 est donc la seule limite possible.

```
> factor(f(x)-x);
```

$$\frac{1}{2}(x-1)^2$$

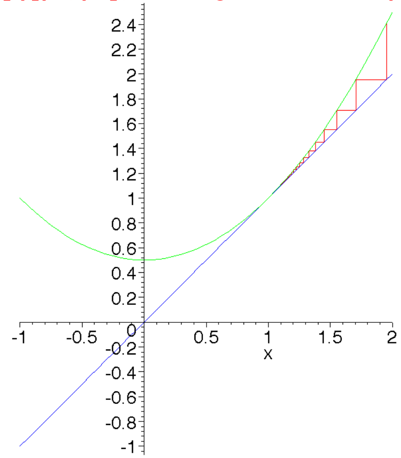
f(x)>x donc la suite est croissante.

```
> plot([x,f(x),f(x)-x],x=-2..2,scaling=constrained,color=[bl
ue,green,red]);
```



Si u(0)>1

```
> u(0):=1.1:
> marche:= i -> ([u(i),u(i)],[u(i),u(i+1)]):
> p1:=plot([seq(marche(i),i=0..20)]):
> p2:=plot([f(x),x],x=-1..2,color=[green,blue]):
> plots[display]([p2,p1],scaling=constrained);
```



La suite est croissante, elle ne peut etre majoree car sinon elle convergerait vers une limite superieure a 1 (ce qui est impossible)

Autres cas :

```
> assume(x<-1):is(f(x)>1);
```

true

```
> x:='x':
```

Donc si u(0)< -1, u(1)>1, on est donc ramene au cas precedent.

Si u(0)=1 ou u(0)=-1, la suite est constante egale a 1 a partir de u(1).

Si 0<u(0)<1 :

```
> assume(x>-1,x<1):is(f(x)<1);
```

true

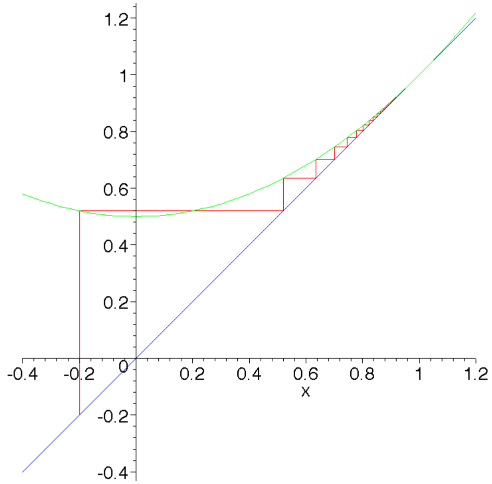
```
> is(f(x)>0);
```

true

```
> x:='x':
```

Donc pour tout n > 1 , u(n) est compris entre 0 et 1. La suite est croissante majoree par 1, elle converge, vers 1 car c'est la seule limite possible.

```
> u(0):=-0.2:
> p3:=plot([seq(marche(i),i=0..20)]):
> p4:=plot([f(x),x],x=-0.4..1.2,color=[green,blue]):
> plots[display]([p4,p3],scaling=constrained);
```



Ex 5 : Recurence à solution polynomiale

```
> restart;
> eq1:=n*u(n+2)-5*u(n+1)-(n+1)*u(n);
```

$$eq1 := n\,\mathrm{u}(n+2) - 5\,\mathrm{u}(n+1) - (n+1)\,\mathrm{u}(n)$$

L'operateur '.' concatene les elements autour de lui.

```
> a . 2;
```

$$a2$$

```
> add((a.i)*n^i,i=0..3);
```

$$a0 + a1\,n + a2\,n^2 + a3\,n^3$$

```
> p:=unapply(%,n);
```

$$p := n \to a0 + a1\,n + a2\,n^2 + a3\,n^3$$

Ne pas se tromper sur l'ordre des parametres de la fonction subs.
Ici on substitue une fonction (u) par une autre (p).

```
> subs(u=p,eq1);
```

$$n\,\mathrm{p}(n+2) - 5\,\mathrm{p}(n+1) - (n+1)\,\mathrm{p}(n)$$

```
> collect(%,n);
```

$$(-2\,a2 - 3\,a3)\,n^2 + (-4\,a1 - 7\,a3 - 6\,a2)\,n - 6\,a0 - 5\,a3 - 5\,a2 - 5\,a1$$

Attention. si on avait divisé eq1 par n, coeffs ne marcherait pas car l'expresion ne serait pas vraiment un polynome.

```
> coeffs(%,n);
```

$$-4\,a1 - 7\,a3 - 6\,a2,\ -2\,a2 - 3\,a3,\ -6\,a0 - 5\,a3 - 5\,a2 - 5\,a1$$

```
> solve({%});
```

$$\{a3 = 2\,a1,\ a1 = a1,\ a2 = -3\,a1,\ a0 = 0\}$$

```
> subs(%,p(n));
```

$$a1\,n - 3\,a1\,n^2 + 2\,a1\,n^3$$

```
> u:=unapply(%,n);
```

$$u := n \to a1\,n - 3\,a1\,n^2 + 2\,a1\,n^3$$

```
> simplify(eq1);
```

$$0$$

Ex 6 : Suite de fonctions

```
> restart;
```

Etudier la convergence simple sur [0,1] de

```
> f:= (x,n) -> n*(x^3+x)*exp(-x)/(n*x+1);
```

$$f := (x, n) \to \frac{n\,(x^3 + x)\,\mathrm{e}^{(-x)}}{n\,x + 1}$$

Appellons l la limite sur]0;1]

```
> l:=unapply(limit(f(x,n),n=infinity),x);
```

$$l := x \to (x^2 + 1)\,\mathrm{e}^{(-x)}$$

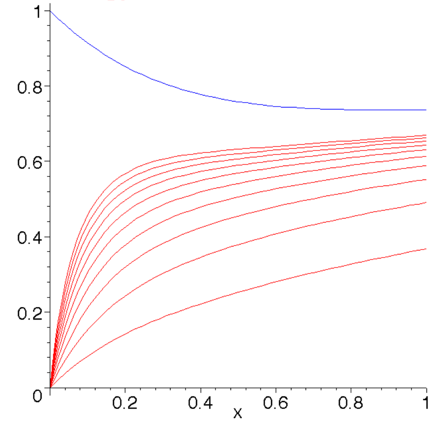
aille ! Maple est rapide mais peu rigoureux car :

```
> f(0,n);
```

$$0$$

Tracons les graphes

```
> plot([seq(f(x,n),n=1..10),l(x)],x=0..1,scaling=constrained
  ,color=[red$10,blue]);
```



Puis la convergence uniforme sur [a,1], a>0

```
> simplify(l(x)-f(x,n));
```

$$\frac{(x^2 + 1)\,\mathrm{e}^{(-x)}}{n\,x + 1}$$

Pas d'espoir de maximiser sur un intervale de type [a,1] : maple ne l'authorise pas.
On se rabbat sur la maximisation du numerateur seulement (et on fait le denominateur a la main)

```
> maximize(numer(%),x,0..1);
```

$$1$$

Donc l(x)-f(x,n)<1/(na+1) et la convergence uniforme

Non-convergence uniforme sur]0,1], par exemple avec x=1/n

```
> limit(l(1/n)-f(1/n,n),n=infinity);
```

$$\frac{1}{2}$$

D'ou le resultat souhaite.