Les suites avec MAPLE

Définir une suite

Une suite étant une application sur des entiers, on la définira comme une fonction à une variable, sauf que l'on n'oubliera pas que la variable est entière :

Exemple 1: Définir la suite de terme général $u(n) = 2n\text{^}2 + 2$, avec un appel paramétrique et un appel par valeur. Puis construire le graphe de u pour n entre 1 et 20

>restart;

$>u := n -> 2 * n\text{^}2 + 2;$

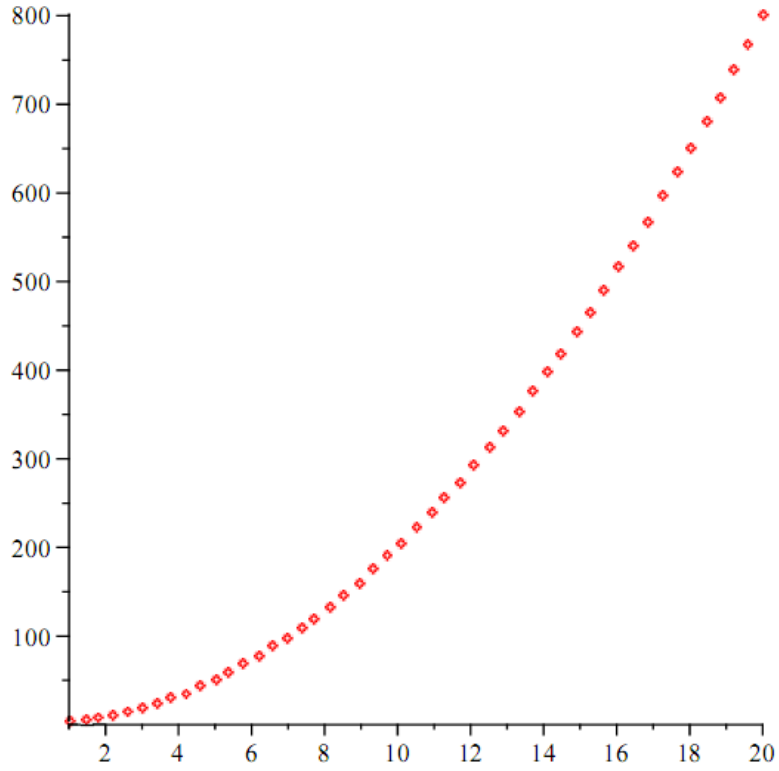
$$u := n \to 2n^2 + 2$$

$>u(n);$

$$2n^2 + 2$$

$>u(5);$

>plot(u,1..20, style=point);



Exemple 2: Définir une suite récurrente de type $u(n) = f(u(n-1))$, avec

$u(0) = 3$ et $f: x -> 1/(x + 1)$, puis donner la valeur de $u(12)$.

>restart;

$>f := x -> 1/(x + 1);$

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

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

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

$>u(0) := 3; u(12);$

u0 := 3

$$\frac{411}{665}$$

Représentation des termes d'une suite

La fonction plot permet aussi de représenter graphiquement les termes d'une suite et d'en déduire ainsi le comportement de la suite à l'infini.

Exemple 3: Représenter graphiquement la suite récurrente de type u(n)=f(u(n-1)), avec u(0)=1 et f:x->1/(x+1) , pour n entre 0 et 20.

>restart;

$>f := x -> 1/(x + 1);$

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

u(0):=1;

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

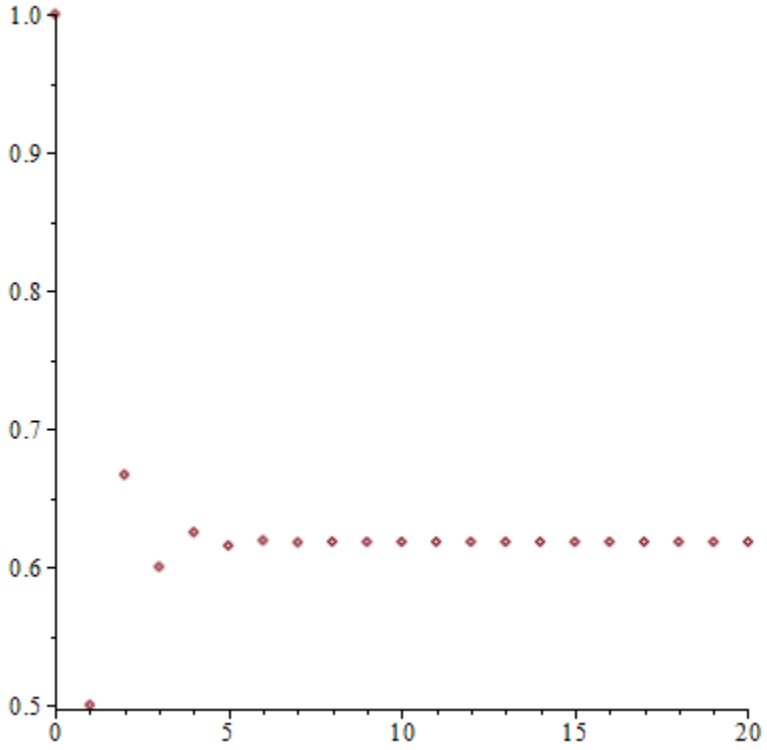
u := n→f (u(n-1))

u(0) := 1

>seq([n,u(n)], n=0..20);

$$[0, 1], \left[1, \frac{1}{2}\right], \left[2, \frac{2}{3}\right], \left[3, \frac{3}{5}\right], \left[4, \frac{5}{8}\right], \left[5, \frac{8}{13}\right], \left[6, \frac{13}{21}\right], \left[7, \frac{21}{34}\right], \left[8, \frac{34}{55}\right], \left[9, \frac{55}{89}\right], \left[10, \frac{89}{144}\right], \left[11, \frac{144}{233}\right], \left[12, \frac{233}{377}\right], \left[13, \frac{377}{610}\right], \left[14, \frac{610}{987}\right], \left[15, \frac{987}{1597}\right], \left[16, \frac{1597}{2584}\right], \left[17, \frac{2584}{4181}\right], \left[18, \frac{4181}{6765}\right], \left[19, \frac{6765}{10946}\right], \left[20, \frac{10946}{17711}\right]$$

$$plot([seq([n, u(n)], n = 0..20)], style = point);$$



Calcul des termes d'une suite

On peut calculer simplement la valeur des termes d'une suite et même le calcul de plusieurs d'entre eux (à l'aide de la demande de boucle **seq** par example).

On notera par ailleurs qu'il peut être bon de travailler avec des nombres en virgule flottante ("1." et non par "1") , afin que Maple ne travaille pas de manière symbolique.

>restart;

$>f := x -> 1/(x + 1);$

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

u(0):=1.;

$$f := x \to \frac{1}{x+1}$$

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

$$u(0) := 1.$$

>u(1);u(3);u(5);

$$0.5000000000$$

$$0.5999999999$$

$$0.6153846154$$

>seq(u(n), n=1..10);

0.5000000000, 0.6666666667, 0.5999999999, 0.6250000000, 0.6153846154, 0.6190476192, 0.6176470588, 0.6181818181, 0.6179775282, 0.6180555556

Les suites récurrentes simples du type u_n=f(u_{n-1}) :

Définition d'une telle suite

Les deux paragraphes précédents donnent des exemples complet pour définir une suite récurrente simple (suite où u_n dépend de u_{n-1}).

Calculs de termes généraux :

La fonction *rsolve* :

La fonction *rsolve* permet de calculer le terme général d'une suite dés lors que celle-ci est définie par une expression de récurrence.

Exemple : Déterminer le terme général des suites qui vérifient :

$v(n+1) = R * v(n)$ avec $v(0) = 2$

> $restart;\ f := x \to R*x : v := n \to f(v(n-1)) : v(0) := 2 :\ v(n);$

Error, (in v) too many levels of recursion

Maple ne peut pas donner l'expression de $v(n)$ en fonction de n, pour cela il faut utiliser *rsolve,*

$rsolve(v(n) = R*v(n-1), v(n));$

$$v(0)\ R^n$$

La fonction *rsolve* s'utilise de la même manière que *dsolve* pour la résolution d'équations différentielles .

> $rsolve(\{v(n) = R*v(n-1),\ v(0) = 2\}, v(n));$

$$2\ R^n$$

On notera que l'appel $v(n)$ reste infructueux, bien que l'on espérait faire apparaître le terme général.

> $v(n);$

$$v(n)$$

On peut alors récupérer ces solutions et travailler avec :

Soit $v(n)$ la suite définie par $v(n+1) = R * v(n)$ et $v(0) := 2$.

Exprimer $v(n)$ en fonction de n en utilisant *rsolve*

```
> restart : rsolve( {v(n) = R * v(n-1), v(0) = 2}, v(n) ); v(n);
```

$$2\,R^{n}$$

$$v(n)$$

```
> w := unapply(rsolve( {v(n) = R * v(n-1), v(0) = 2}, v(n) ), n);
```

$$w := n \rightarrow 2\,R^{n}$$

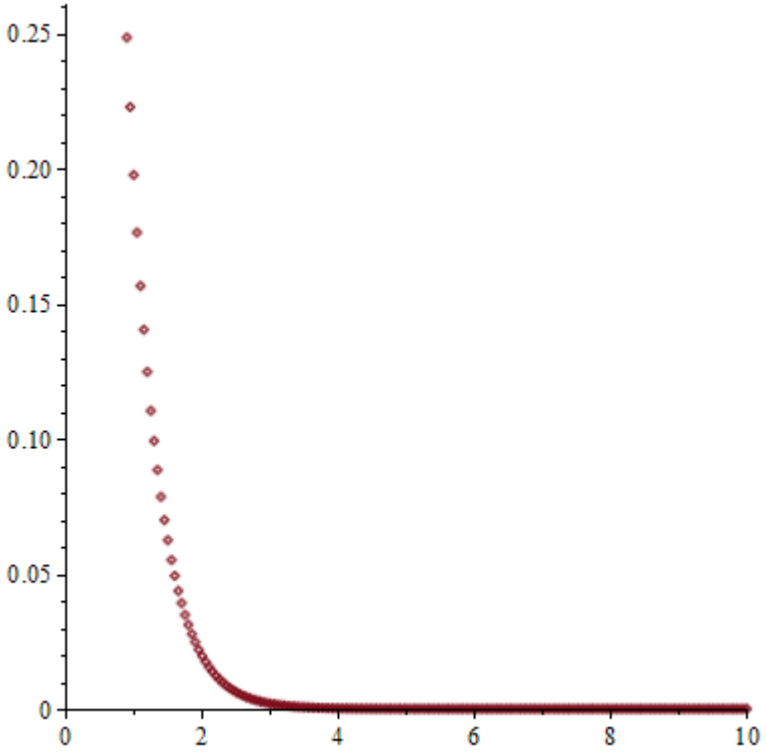
```
> w(n);
```

$$2\,R^{n}$$

```
> v(0) := 2 : R := 0.1 :
> seq(w(i), i = 1 ..5);
```

$$0.2,\ 0.02,\ 0.002,\ 0.0002,\ 0.00002$$

```
> plot(w, 0 ..10, style = point);
```



Dans un autre exemple, on va donner l'expression du terme générale de la suite définie par

$$v(n+1) = r + v(n), v(0) = 2$$

```
> restart : simplify(rsolve( {v(n) = v(n-1) + r, v(0) = 2}, v(n) ));
```

$$n\,r + 2$$

```
> w := unapply(simplify(rsolve( {v(n) = v(n-1) + r, v(0) = 2},
      v(n) )), n);
```

$$w := n \to n\,r + 2$$

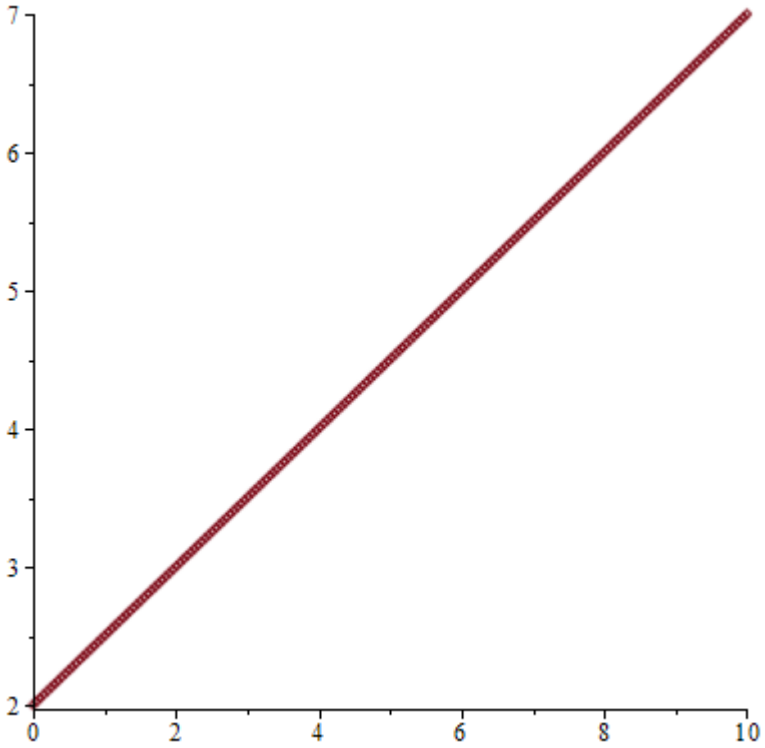
> $w(n);$

$$n\,r + 2$$

$v(0) := 2.; r := 0.5; seq(w(i), i = 1..10);$

$$2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0$$

> $plot(w, 0..10, style = point);$



Exemple :

La suite de Fibonacci est donnée par les équations :

$$\begin{cases} u(0) = 0 \\ u(1) = 1 \\ u(n) = u(n-1) + u(n-2), \quad n \geq 2 \end{cases}$$

> $rsolve(\{F(n+1) = F(n) + F(n-1), F(0) = 0, F(1) = 1\}, F(n));$

$$-\frac{1}{5}\sqrt{5}\left(-\frac{1}{2}\sqrt{5} + \frac{1}{2}\right)^n + \frac{1}{5}\sqrt{5}\left(\frac{1}{2}\sqrt{5} + \frac{1}{2}\right)^n$$

Extremums d'une fonction

Trois commandes de base sont disponibles pour trouver les extremums d'une fonction d'une ou de plusieurs variables :

Extrema (expression, {contrainte}, {variable})

Minimize (expression, option1, option2, …, option3)

Maximize (expression, option1, option2, …, option3)

Exemple 1: Trouver le minimum de $(x-2)^2$ sur l'intervalle $[1,5]$

```
>restart ;
 fonction := (x-2)^2 ;
 minimize(fonction, x=1..5,location) ;
```

location indiquera quelles sont les coordonnées du point où se trouve ce minimum.

Exemple 2: trouver le minimum et le maximum de $f(x,y) = x^2 + y^2$ sur le domaine $[-3,3] \times [-4,4]$

```
>restart ;
 fonction := x^2 + y^2 ;
 minimize(fonction, x = -3..3, y = -4..4, location) ;
 maximize(fonction, x = -3..3, y = -4..4, location) ;
```

Exemple 3 : soit une sphère centrée à l'origine de rayon 2 et dont l'équation est

$$x^2 + y^2 + z^2 = 2.$$

Sachant que la température sur la sphère est donnée par la fonction :

$T(x,y,z) = x^2 + y^2 + z^2$, trouver les extremums de $T(x,y,z)$.

```
>restart ;
 fonction := x^2 + y^2 + z^2 ;
 extrema (fonction, x^2+y^2+z^2=2, {x, y, z})
```

Tables et tableaux

> *restart :*

Elle permet de vider toutes les assignations faites avant la déclaration de cette commande.

```
> t := table([10, 11, 12, 13, 14, 15]) ; whattype(%) ; t[1] ;
```

$$t := table([1 = 10, 2 = 11, 3 = 12, 5 = 14, 4 = 13, 6 = 15])$$

$$table$$

$$10$$

Pour atteindre une valeur du tableau, on utilise la notation [].

On peut aussi assigner un tableau de variables quelconques.

```
> t := array(1..6); whattype(%) ; eval(t) ;
```

$$t := array(1..6, [\,])$$

$$array$$

$$\left[\grave{?}_1 \quad \grave{?}_2 \quad \grave{?}_3 \quad \grave{?}_4 \quad \grave{?}_5 \quad \grave{?}_6 \right]$$

Si on met la valeur des cellules en deuxième argument, le tableau est rempli :

```
> t := array(1..6, [10, 11, 12, 13, 14, 15])
```

$$t := \left[10 \quad 11 \quad 12 \quad 13 \quad 14 \quad 15 \right]$$