Université Partis-Sud — **Année universitaire 2002-2003**

DEUG SMPE — S3 — **Stage Mathematica**

Séance n°5

Mardi 24 septembre 2002

Compléments divers et révisions

Résumé et rappel du cours

Rappelons quelques points.

- Les valeurs auxquelles les fonctions sont évaluées doivent être placées entre crochets.
- Les parenthèses servent à séparer les termes d'un produit et à délimiter les expressions.
- Les accolades servent de délimiteurs de listes.
- Les variables, les éléments d'une liste sont séparés par des virgules. Le séparateur décimal est le point.
- Les instruction ne doivent pas être terminées par un ";".
- Le résultat de la dernière instruction évaluée par Mathematica peut être désigné par %. Attention : il ne s'agit pas forcément de l'instruction figurant juste au-dessus dans la feuille de travail.
- Les fonctions internes à Mathematica commencent toutes par des majuscules ; lorsque le nom d'une fonction est composé de plusieurs mots, on met une majuscule à chacun des mots composant le nom de l'instruction ; par exemple, TableForm.
- Les fonctions définies par l'utilisateur doivent commencer par une minuscule.
- Le nom d'une fonction ne doit pas comporter d'espace. Ainsi, *ab* sera interprété comme le nom d'une fonction ou d'un objet, tandis que *a b* (avec une espace entre le "*a*" et le "*b*") sera interprété comme le produit de *a* par *b*. Pour un produit, il est inutile d'écrire *a***b*.
- Le symbole "=" engendre une évaluation immédiate d'une expression, tandis que le symbole ":=" engendre une évaluation différée. Pour bien comprendre la différence, voici deux exemples :

```
a = 2

2

g[x_] = a x + 3

3 + 2 x

g[1]

5

a = 3

3

g[1]

5
```

A gauche, la fonction *g* est définie par le symbole "=". Elle est donc évaluée immédiatement, avec la valeur courante de *a*, c'est-à-dire 2. L'évaluation immédiate se traduit dans la réponse par le 3+2*x*. Lorsqu'on change la valeur de *a*, *g* ne change pas d'expression.

A droite, la fonction *f* est définie par le symbole ":=". Son expression n'est donc pas évaluée immédiatement. Il n'y a pas de réponse suivant l'instruction de définition de *f*. La définition de *f* que Mathematica garde en mémoire est celle qui a été donnée. Lorsqu'on change la valeur de *a*, l'évaluation de *f* en un point tient compte du changement de valeur de *a*.

```
a = 2

2

f[x_] := a x + 3

f[1]

5

a = 3

3

f[1]

6
```

‣ On peut définir une fonction de trois façons différentes. Il ne faut pas les mélanger : Lorsqu'on définit une fonction par son expression, on écrit $f[x\_]$:=... Ne pas oublier le "_" après le nom de la variable, et ne pas faire suivre la définition de la fonction du symbole "&". Au contraire, lorsqu'on définit une fonction pure, on écrit f :=...&. Dans l'expression placée entre := et &, on remplace la variable par #, et dans le cas d'une fonction de plusieurs variables la première variable par #1, la deuxième par #2 etc.

‣ On a vu que pour les fonctions de plusieurs variables, il convient de distinguer la fonction f définie par $f[x\_,y\_]$:=... de la fonction ff définie par $ff[\{x\_, y\_\}]$:=...Lorsqu'on veut composer f avec une fonction g dont l'image est un couple, cela ne marche pas, comme on peut le voir ci-contre, alors que la composition $ff \circ g$ est possible. La fonction Apply permet de contourner la difficulté. On a en effet par exemple

$$\text{Apply}[f, \{a, b, c\}] = f[a, b, c].$$

```
f := (#1 + #2) &

g := {2 #1, #2} &

f@g[x, y]

Function::slotn :
 Slot number 2 in #1 + #2 & cannot be
   filled from (#1 + #2 &) [{2 x, y}].

{2 x + #2, y + #2}

Apply[f, g[x, y]]

2 x + y
```

```
l1 := {a, c, b}

l2 := {c, d}

Join[l1, l2]

{a, c, b, c, d}

Union[l1, l2]

{a, b, c, d}

Position[l1, c]

{{2}}

Position[Join[l1, l2], c]

{{2}, {4}}

Extract[Join[l1, l2], {{1}, {3}, {5}}]

{a, b, d}

Flatten[Position[Join[l1, l2], c]]

{2, 4}

Position[Join[l1, l2], c] // TableForm

2
4

Transpose[Position[Join[l1, l2], c]] // TableForm

2     4
```

‣ L'instruction Join[ℓ_1, ℓ_2] concatène les deux listes ℓ_1 et ℓ_2, tandis que Union[ℓ_1, ℓ_2] donne la réunion des ensembles sous-jacents aux listes ℓ_1 et ℓ_2, en triant les éléments de cet ensemble. Le tri est effectué selon l'ordre lexicographique. Par exemple, si deux sous-listes {1, 2} et {1, 1} doivent être triées, {1, 1} sera placé avant {1, 2}.

‣ L'instruction Position[ℓ_1, c] donne une liste des positions de l'élément c dans la liste ℓ_1.

‣ L'instruction Extract[ℓ_1, ℓ_3] est en quelque sorte l'instruction opposée de Position. Si ℓ_3 est une liste d'emplacements, elle donne en effet les éléments de ℓ_1 placés aux éléments de ℓ_3.

‣ $\ell_1[[x]]$ fournit le x-ième élément de la liste ℓ_1.

‣ La commande Flatten[*liste*] crée une liste "plate" constituée des éléments de *liste* placés bout à bout, en enlevant les sous-niveaux de *liste*.

‣ *xxx* // f est une autre forme de f[*xxx*]. Par exemple, *liste* // TableForm est équivalent à TableForm[*liste*], demande l'affichage sous forme de tableau de *liste*.

‣ Transpose[*liste*] inverse les lignes et les colonnes de *liste*.

‣ Dans une liste de la forme $\ell = \{\{\ell_1\},\{\ell_2\}\}$, ℓ_1 est la première ligne, et ℓ_2 la deuxième ligne.

‣ Lorsqu'on a déjà travaillé un moment en Mathematica, et qu'on commence à étudier un nouveau problème, il est prudent d'effacer de la mémoire courante les définitions qui ont été utilisées auparavant, au moins pour tous les symboles que l'on compte utiliser dans la nouvelle étude. Pour cela, utiliser Clear; Par exemple, Clear[f, g, x] efface les définitions courantes de f, g, et x.

```
f := (# - 1)/(# + 1) &

Solve[Denominator[f[x]] == 0, x]

{{x -> -1}}

f'[x]

-(-1 + x)/(1 + x)^2 + 1/(1 + x)

Together[f'[x]]

2/(1 + x)^2

bornes := {-Infinity, -1, Infinity}

{bornes, Table[Limit[f[x], x -> bornes[[k]]],
   {k, 1, 3}]} // TableForm
```

//TableForm=

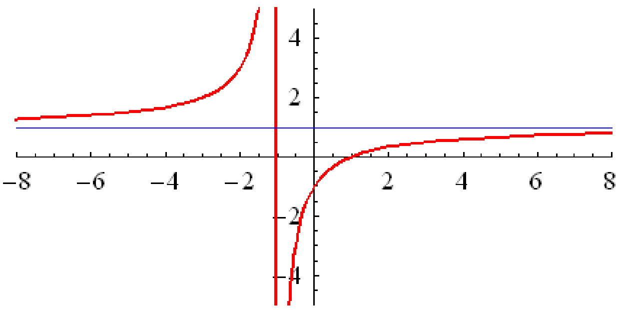
$-\infty$	-1	∞
1	$-\infty$	1

```
Limit[f[-x], x -> 1]

∞
```

- Le numérateur d'une fraction *fraction* peut être manipulé grâce à l'instruction Numerator[*fraction*]. Le dénominateur est appelé par Denominator[*fraction*].
- Une expression *expression* comportant plusieurs fractions peut être mise au même dénominateur grâce à la commande Together[*expression*].
- La dérivée d'une fonction d'une variable *f* se note *f*′.
- Pour regrouper les valeurs prises par une fonction en certains points remarquables, ou les limites, penser à faire un tableau, toujours plus facile à lire.
- Mathematica manipule sans problème l'infini, noté ∞ ou Infinity.
- La limite d'une fonction *f* en un point fini ou infini *a* s'obtient par la commande Limit[*f* [*x*], *x* → *a*]. Noter que lorsque *f* possède une limite à gauche et une limite à droite en *a*, la commande Limit donne la limite à droite. Pour avoir la limite à gauche, demander Limit[*f* [—*x*], *x* → –*a*].

```
Plot[{f[t], 1}, {t, -8, 8},
 PlotRange -> {{-8, 8}, {-5, 5}}, AspectRatio -> 1/2,
 PlotStyle -> {{RGBColor[1, 0, 0], Thickness[.005]},
   {RGBColor[0, 0, 1], Thickness[0.003]}},
 TextStyle -> {FontFamily -> "Times", FontSize -> 12}]
```



- Le tracé d'une fonction *f* sur un intervalle [*a*, *b*] s'obtient par
  Plot[*f* [*x*], {*x*, *a*, *b*}].
  Si l'on veut tracer deux fonctions *f* et *g* sur un même graphique, utiliser
  Plot[{*f* [*x*], *g*[*x*]}, {*x*, *a*, *b*}].
  Le contrôle de l'aspect du graphique obtenu se fait par l'ajout d'options placées entre virgules, avant le crochet de fermeture de la commande Plot. Voici les plus courantes : les autres se trouvent expliquées dans l'aide de Mathematica.

— PlotRange → {{a_1, a_2}, {b_1, b_2}} défini le rectangle représenté, $[a_1, a_2] \times [b_1, b_2]$.

— AspectRatio → α défini le rapport hauteur du graphique/largeur du graphique.

La combinaison de ces deux dernières instructions permet de traduire la notion de repère orthonormé, et de s'assurer par exemple, qu'un cercle apparaîtra sur le dessin comme un cercle plutôt que comme une ellipse.

— PlotStyle → {*liste de primitives graphiques*} précise les primitives graphiques qui doivent être utilisées pour les différents éléments du graphique. Par exemple, RGBColor[*u*, *v*, *w*], avec $u + v + w = 1$ donne la couleur de graphe de *f*, exprimée avec la combinaison barycentrique de coefficients *u*, *v* et *w* des trois couleurs fondamentales Rouge, Vert (Green) et Bleu. Thickness[*x*] défini l'épaisseur du trait du graphe de *f* , exprimé en proportion de l'image. Lorsqu'on trace plusieurs fonctions, on dresse une liste d'options graphiques, la première sous liste d'options concernant la première fonction, le deuxième le deuxième fonction, et ainsi de suite. Ainsi, au-dessus, on a représenté *f* et son asymptote d'équation $y = 1$, la première en rouge, la deuxième en bleu.

▸ Les autres commandes graphiques d'usage courant sont ListPlot pour représenter des listes de points dans le plan, ParametricPlot pour les courbes paramétrées, Plot3D pour les surfaces.

▸ On peut ramener l'étude de toute suite récurrente d'ordre ≥ 2, fixe ou variable à celle d'une suite récurrente d'ordre 1 en passant à une suite vectorielle. Par exemple, si u_n est définie par $u_{n+2} = u_{n+1} + u_n$, en posant

$U_n = (u_n, u_{n+1})$,

on obtient

$U_{n+1} = (u_{n+1}, u_{n+2})$

$= (u_{n+1}, u_{n+1} + u_n) = f(U_n)$

avec $f(x, y) = (y, x + y)$. Soit (v_n) la suite définie par

$v_{n+1} = 2n + 3v_n$.

Si $V_n = (n, v_n)$, on a

$V_{n+1} = (n+1, v_{n+1})$

$= (n+1,\ 2n + 3v_n) = g(V_n)$,

avec $g(x, y) = (x+1, 2x + 3y)$. Soit (w_n) la suite définie par

$w_{n+2} = 2n + 3u_{n+1} + 4u_n$.

Si $W_n = (n, w_n, w_{n+1})$, on a

$W_{n+1} = (n+1, w_{n+1}, w_{n+2})$

$= (n+1, w_{n+1}, 2n+3w_{n+1}+4w_n)$

$= h(W_n)$,

avec

$h(x, y, z) = (x+1, z, 2x + 3z + 4y)$.

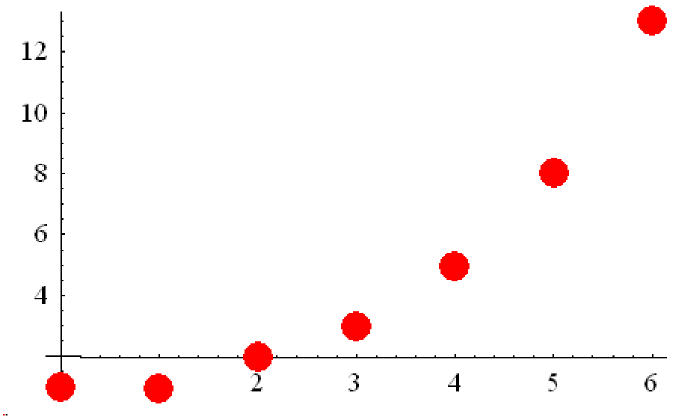
Noter qu'en Mathematica, pour pouvoir itérer commodément une fonction f de deux variables, à valeurs dans $\mathbb{R}^2$, il faut la définir non pas comme *f* [*x*_, *y*_] :=..., mais comme *f* [{*x*_, *y*_}] := ...

```
f[{x_, y_}] := {y, x + y}

NestList[f, {1, 1}, 5] // TableForm
```

1	1
1	2
2	3
3	5
5	8
8	13

```
ListPlot[Table[{k, Nest[f, {1, 1}, k][[1]]},
  {k, 0, 6}],
 PlotStyle → {RGBColor[1, 0, 0], PointSize[.05]},
 TextStyle → {FontFamily → "Times", FontSize → 12}]
```



▸ L'instruction NestList[*f*, *x*, *n*] donne la liste des itérées successives de *f* appliquées à *x* de 0, à *n*, c'est-à-dire $\{x, f(x), f^2(x), \cdots, f^n(x)\}$.

Exercices

1. Etudier puis représenter la fonction *f* définie par $f(x) = \dfrac{\sqrt{x^2 + 3x + 1}}{x - 1}$.

2. Soit (u_n) la suite définie par $u_1 = 4$, et, pour tout $n > 1$, $u_{n+1} = \dfrac{u_n}{3} + \sqrt{n}$. Représenter les 100 premiers points de cette suite, et dresser la liste des 30 premiers points.

3. Construire une fonction *m* donnant, pour tout entier naturel $n > 1$ la matrice de taille *n* dont le coefficient d'ordre $i \times j$ est le pgcd de *i* et de *j*. Afficher *m*(10). Extraire de cette matrice la sous-matrice constituée des lignes 3 et 4 de *m*(10). Dresser la liste des coefficients de *m*(10). Combien y-a-t-il dans *m*(10) de coefficients égaux à 1 ?

DEUG SMPE - 2002-2003

Stage Mathematica

Séance n°5

24 septembre 2002

Révisions et compléments

Résumé du cours

Principes de base de Mathematica

= versus :=

```
a = 2
```

2

```
g[x_] = a x + 3
```

3 + 2 x

```
g[1]
```

5

```
a = 3
```

3

```
g[1]
```

5

```
Clear[a, g]
```

```
a = 2
```

```
2
```

```
f[x_] := a x + 3
```

```
f[1]
```

```
5
```

```
a = 3
```

```
3
```

```
f[1]
```

```
6
```

■ **Fonctions de plusieurs variables**

```
f := (#1 + #2) &
```

```
g := {2 #1, #2} &
```

```
f@g[x, y]
```

— Function::slotn :
Slot number 2 in #1 + #2 & cannot be filled from (#1 + #2 &)[{2 x, y}].

```
{2 x + #2, y + #2}
```

```
Apply[f, g[x, y]]
```

```
2 x + y
```

■ Listes

```
Clear[a, b, c, d, e, f, g, l1, l2]
```

```
l1 := {a, c, b}
```

```
l2 := {c, d}
```

```
Join[l1, l2]
```

```
{a, c, b, c, d}
```

```
Union[l1, l2]
```

```
{a, b, c, d}
```

```
Position[l1, c]
```

```
{{2}}
```

```
Position[Join[l1, l2], c]
```

```
{{2}, {4}}
```

```
Extract[Join[l1, l2], {{1}, {3}, {5}}]
```

```
{a, b, d}
```

```
Flatten[Position[Join[l1, l2], c]]
```

```
{2, 4}
```

```
Position[Join[l1, l2], c] // TableForm
```

```
2
4
```

```
Transpose[Position[Join[l1, l2], c]] // TableForm
```

2	4

Fonctions

```
Clear[f]
```

$$f := \frac{\# - 1}{\# + 1} \&$$

```
Solve[Denominator[f[x]] == 0, x]
```

$\{\{x \to -1\}\}$

```
f'[x]
```

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

```
Together[f'[x]]
```

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

```
bornes := {-Infinity, -1, Infinity}
```

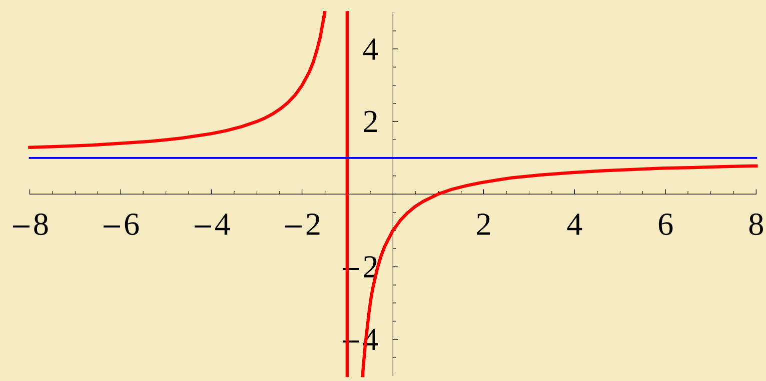
```
{bornes, Table[Limit[f[x], x → bornes[[k]]],
   {k, 1, 3}]} // TableForm
```

$-\infty$	-1	∞
1	$-\infty$	1

```
Limit[f[-x], x → 1]
```

∞

```
Plot[{f[t], 1}, {t, -8, 8},
 PlotRange → {{-8, 8}, {-5, 5}},
 AspectRatio → 1 / 2,
 PlotStyle → {{RGBColor[1, 0, 0], Thickness[.005]},
   {RGBColor[0, 0, 1], Thickness[0.003]}},
 TextStyle → {FontFamily → "Times", FontSize → 12}]
```




```
- Graphics -
```

■ Suites

```
Clear[f, u, v, w, x, y, z]
```

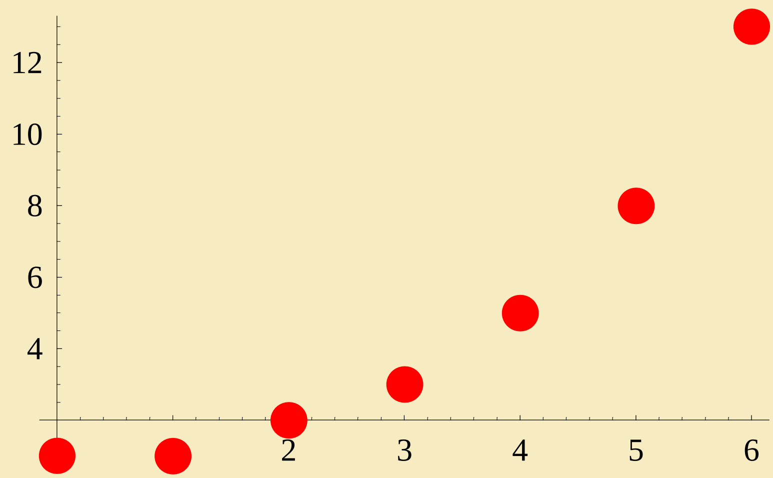
In[83]:=
```
f[{x_, y_}] := {y, x + y}
```

In[84]:=
```
NestList[f, {1, 1}, 5] // TableForm
```

Out[84]//TableForm=

1	1
1	2
2	3
3	5
5	8
8	13

In[92]:=
```
ListPlot[
 Table[{k, Nest[f, {1, 1}, k][[1]]}, {k, 0, 6}],
 PlotStyle → {RGBColor[1, 0, 0], PointSize[.05]},
 TextStyle → {FontFamily → "Times", FontSize → 12}]
```



Out[92]= - Graphics -

Exercices

Exercice 1 - Etude d'une fonction

In[94]:=
```
Clear[f, x]
```

In[95]:= $f := \dfrac{\sqrt{\#^2 + 3\,\# + 1}}{\# - 1}\ \&$

In[98]:=
```
Solve[Numerator[f[x]] == 0, x]
```

Out[98]= $\left\{\left\{x \to \frac{1}{2}\left(-3 - \sqrt{5}\right)\right\}, \left\{x \to \frac{1}{2}\left(-3 + \sqrt{5}\right)\right\}\right\}$

In[99]:=
```
N[%]
```

Out[99]= {{x → -2.61803}, {x → -0.381966}}

In[100]:=

```
f'[x]
```

Out[100]=

$$\frac{3+2x}{2(-1+x)\sqrt{1+3x+x^2}}-\frac{\sqrt{1+3x+x^2}}{(-1+x)^2}$$

In[101]:=

```
Together[f'[x]]
```

Out[101]=

$$-\frac{5(1+x)}{2(-1+x)^2\sqrt{1+3x+x^2}}$$

In[102]:=

$$\text{bornes} := \left\{-\infty,\ \frac{1}{2}\left(-3-\sqrt{5}\right),\ \frac{1}{2}\left(-3+\sqrt{5}\right),\ 1,\ \infty\right\}$$

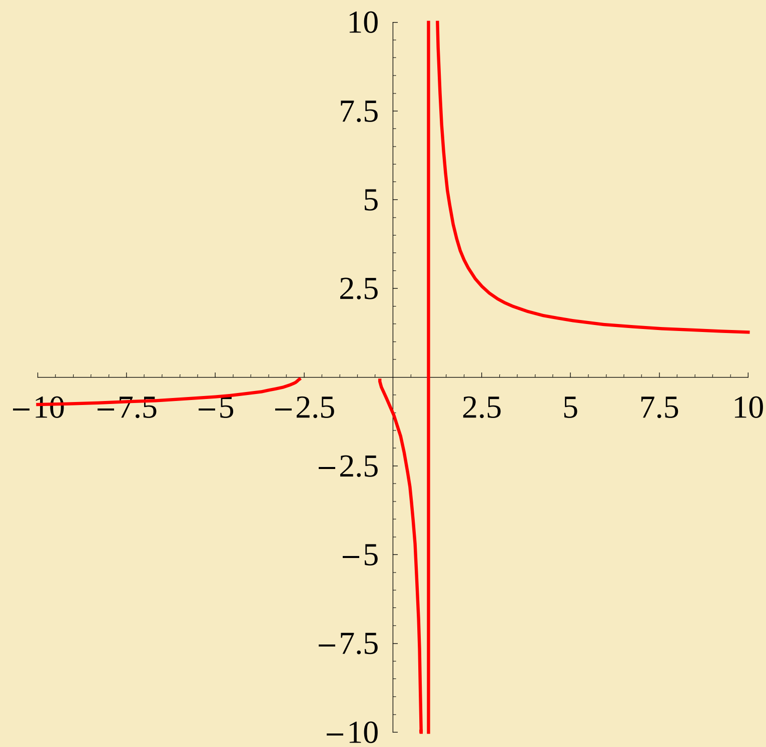
In[103]:=

```
{bornes, Table[Limit[f[x], x → bornes[[k]]],
   {k, 1, 5}]} // TableForm
```

Out[103]//TableForm=

$-\infty$	$\frac{1}{2}\left(-3-\sqrt{5}\right)$	$\frac{1}{2}\left(-3+\sqrt{5}\right)$	1	∞
-1	0	0	∞	1

In[97]:=
```
Plot[f[x], {x, -10, 10},
 PlotRange → {{-10, 10}, {-10, 10}},
 AspectRatio → 1,
 PlotStyle → {{RGBColor[1, 0, 0], Thickness[.005]},
   {RGBColor[0, 0, 1], Thickness[0.003]}},
 TextStyle → {FontFamily → "Times", FontSize → 12}]
```



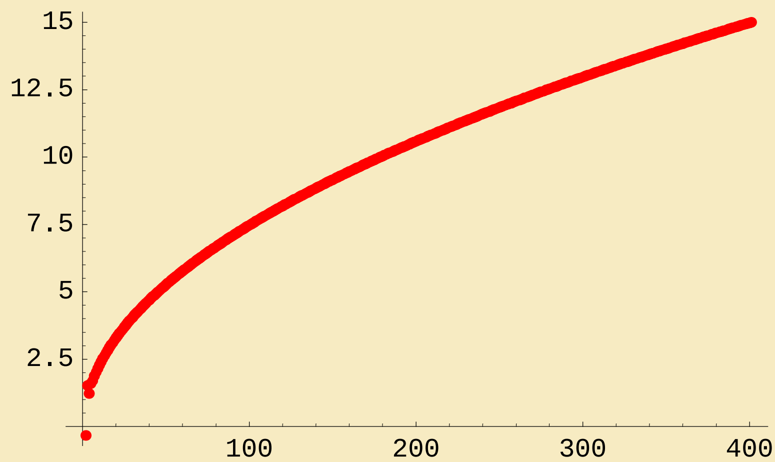
Out[97]= - Graphics -

Exercice 2 - Etude d’une suite

```
u[n + 1] := u[n] / 3 + √n
```

$f_2[\{x\_, y\_\}] := \left\{x + 1, -\frac{y}{3} + \sqrt{x}\right\}$

```
ListPlot[Table[Nest[f2, {1., 4.}, k], {k, 1, 400}],
 PlotStyle → {RGBColor[1, 0, 0], PointSize[.015]}]
```




```
- Graphics -
```

```
N[NestList[f2, {1, 4}, 100]] // TableForm
```

1.	4.
2.	−0.333333
3.	1.52532
4.	1.22361
5.	1.59213
6.	1.70536
7.	1.88104
8.	2.01874
9.	2.15551
10.	2.2815
11.	2.40178
12.	2.51603
13.	2.62542
14.	2.73041
15.	2.83152
16.	2.92914
17.	3.02362
18.	3.11523
19.	3.20423
20.	3.29082
21.	3.3752
22.	3.45751
23.	3.53791
24.	3.61653
25.	3.69347
26.	3.76884
27.	3.84274

28.	3.91524
29.	3.98642
30.	4.05636
31.	4.12511
32.	4.19273
33.	4.25928
34.	4.3248
35.	4.38935
36.	4.45296
37.	4.51568
38.	4.57754
39.	4.63857
40.	4.69881
41.	4.75829
42.	4.81703
43.	4.87506
44.	4.93242
45.	4.98911
46.	5.04517
47.	5.10061
48.	5.15545
49.	5.20972
50.	5.26343
51.	5.31659
52.	5.36923
53.	5.42136
54.	5.47299
55.	5.52414
56.	5.57482
57.	5.62504
58.	5.67482
59.	5.72417
60.	5.77309
61.	5.8216
62.	5.86972
63.	5.91744
64.	5.96478
65.	6.01174
66.	6.05834
67.	6.10459
68.	6.15049
69.	6.19605
70.	6.24127
71.	6.28618
72.	6.33076
73.	6.37503
74.	6.41899
75.	6.46266
76.	6.50603
77.	6.54912
78.	6.59192
79.	6.63445
80.	6.67671

81.	6.7187
82.	6.76043
83.	6.80191
84.	6.84313
85.	6.88411
86.	6.92484
87.	6.96534
88.	7.0056
89.	7.04563
90.	7.08544
91.	7.12502
92.	7.16439
93.	7.20353
94.	7.24247
95.	7.2812
96.	7.31973
97.	7.35805
98.	7.39617
99.	7.4341
100.	7.47184
101.	7.50939

Exercice 3 - Travail avec des listes

In[105]:=

```
m[n_] := Table[GCD[i, j], {i, 1, n}, {j, 1, n}]
```

In[107]:=

```
m[10] // TableForm
```

Out[107]//TableForm=

1	1	1	1	1	1	1	1	1	1
1	2	1	2	1	2	1	2	1	2
1	1	3	1	1	3	1	1	3	1
1	2	1	4	1	2	1	4	1	2
1	1	1	1	5	1	1	1	1	5
1	2	3	2	1	6	1	2	3	2
1	1	1	1	1	1	7	1	1	1
1	2	1	4	1	2	1	8	1	2
1	1	3	1	1	3	1	1	9	1
1	2	1	2	5	2	1	2	1	10

In[109]:=

```
{m[10][[3]], m[10][[4]]} // TableForm
```

Out[109]//TableForm=

1	1	3	1	1	3	1	1	3	1
1	2	1	4	1	2	1	4	1	2

Voici la liste des coefficients de m(10) :

In[113]:=
```
Union[Flatten[m[10]]]
```

Out[113]=
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

In[115]:=
```
Count[Flatten[m[10]], 1]
```

Out[115]=
```
63
```