

[**Correction du contrôle du 21 octobre**

[**Exercice 1**

[> z:=1+sqrt(3)*I;

$$z := 1 + \sqrt{3} I$$

[> argument(z);

$$\frac{\pi}{3}$$

[Remarque : il existe une commande Maple qui se nomme *isqrt*. Dans la deuxième phrase à corriger, on calcule l'argument du nombre défini à la première ligne. Il me semble alors clair que la première ligne sert à définir un nombre complexe, et qu'il s'agissait de $1 + \sqrt{3}i$, et non pas de *isqrt*(3).

[**Exercice 2**

[Il y a un exercice similaire dans la feuille d'exercices n°2. Il s'agit de déterminer la "formule" qui se cache derrière les séquences, puis utiliser la commande *seq* avec cette formule.

[> seq(2*n+1, n=0..8);

$$1, 3, 5, 7, 9, 11, 13, 15, 17$$

[> seq(n*x/(y+n), n=1..5);

$$\frac{x}{y+1}, \frac{2x}{y+2}, \frac{3x}{y+3}, \frac{4x}{y+4}, \frac{5x}{y+5}$$

[**Exercice 3**

[1)

[> restart;

[> a:=exp(Pi*sqrt(163));

$$a := e^{(\pi\sqrt{163})}$$

[> evalf(a, 30);

$$0.26253741264076874400000000024 \cdot 10^{18}$$

[2)

[> n:=262537412640768744;

$$n := 262537412640768744$$

[> evalf(a-n);

$$0.$$

[En calculant une valeur approchée de a-n à 10 chiffres significatifs (c'est ce que fait par défaut *evalf*), on trouve 0. Il faut donc "pousser" plus loin.

[> evalf(a-n, 15);

$$0.$$

[En continuant, par exemple avec 30, on trouve :

[> evalf(a-n, 30);

$$0.24 \cdot 10^{-10}$$

[Ceci montre que $|a-n| \leq 10^{-10}$. Donc n est une approximation de a à 10^{-10} près.

[*Rappel.* Soient deux nombres a et n. On dit que n est une approximation (ou valeur approchée) de a à ϵ près si on a $|a-n| \leq \epsilon$

[**Exercice 4**

[1) Attention à bien définir une *fonction* comme demandé (par exemple à l'aide de \rightarrow).

[> restart;

f:=x->x^3*exp(x);

$$f := x \rightarrow x^3 e^x$$

[2) Comme f est une fonction, on s'attend à ce que g soit également une fonction.

[On peut procéder en plusieurs étapes :

[> h:=x->diff(f(x), x);

$$h := x \rightarrow \text{diff}(f(x), x)$$

[> g:=x->diff(h(x), x);

$$g := x \rightarrow \text{diff}(h(x), x)$$

[> g(x);

$$6x e^x + 6x^2 e^x + x^3 e^x$$

[ou en une seule :

[> g:=x->diff(diff(f(x), x), x);

$$g := x \rightarrow \text{diff}(\text{diff}(f(x), x), x)$$

[> g(x);

$$6x e^x + 6x^2 e^x + x^3 e^x$$

[ou encore avec D, au choix !

[> g:=D(D(f));

$$g := x \rightarrow 6x e^x + 6x^2 e^x + x^3 e^x$$

[> g(x);

$$6x e^x + 6x^2 e^x + x^3 e^x$$

[3) Pour calculer les valeurs approchées des racines :

[> solve(g(x)=0, x);

$$0, -3 + \sqrt{3}, -3 - \sqrt{3}$$

[> evalf(%);

$$0., -1.267949192, -4.732050808$$

[Attention, si vous aviez défini g à la question 2 comme une expression en x (et non pas comme une fonction), la syntaxe pour *solve* n'est pas la même. Exemple :

[> g:=diff(diff(f(x), x), x);

$$g := 6x e^x + 6x^2 e^x + x^3 e^x$$

[> solve(g(x), x);

proc(x) 0 end proc, RootOf((e^Z)X_Z), proc(x) RootOf(6+6*_Z+_Z^2) end proc

[Cela ne fonctionne pas parce que g(x) n'a pas de sens ! La preuve :

[> g(x);

$$6x(x)(e^x)(x) + 6x(x)^2(e^x)(x) + x(x)^3(e^x)(x)$$

[Il fallait alors utiliser *solve* comme ceci :

[> solve(g, x);

$$0, -3 + \sqrt{3}, -3 - \sqrt{3}$$

[Conclusion : attention à bien faire la différence entre **fonctions** et **expressions**. Elles ne se définissent pas et ne s'utilisent pas de la même façon.

[Enfin, pour résoudre la question, on pouvait essayer *fsolve*. Malheureusement, on ne trouve qu'une racine :

[> g:=D(D(f));

fsolve(g(x)=0, x=-100..100);

$$g := x \rightarrow 6x e^x + 6x^2 e^x + x^3 e^x$$

0.

En consultant l'aide, on apprend que pour une équation générale (non polynomiale), `fsolve` ne calcule qu'une racine réelle. Si on a déterminé les valeurs exactes des racines avec `solve`, on pouvait s'en sortir en choisissant des intervalles bien adaptés pour "attraper" les deux autres racines :

```
> fsolve(g(x)=0, x=-5..-2);
```

-4.732050808

```
> fsolve(g(x)=0, x=-2..-1);
```

-1.267949192

La solution recommandée était celle avec `solve`, puis `evalf`.

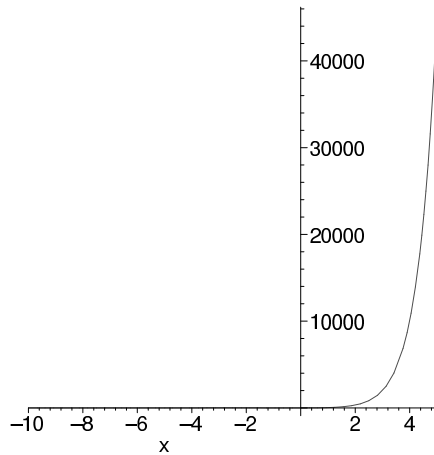
4) Pour afficher l'aide en question :

```
> ?plot, options
```

Cela amène sur la page spéciale consacrée aux options de la commande `plot` (elles sont très nombreuses). Au paragraphe consacré à l'option `numpoints`, on y lit : "Specifies the minimum number of points to be generated (the default is $n = 50$)". Donc Maple utilise 50 points par défaut pour tracer une courbe.

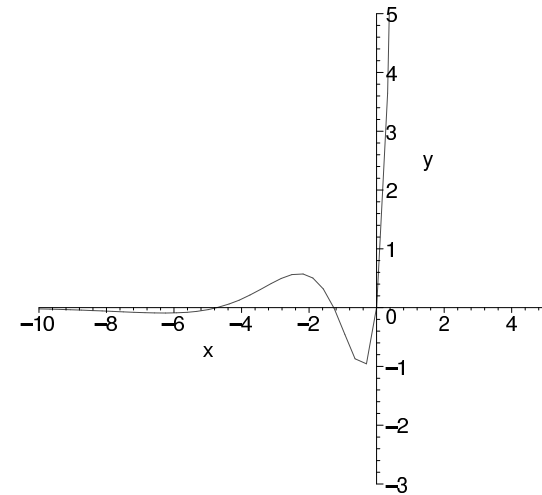
5) Comme intervalle adapté en abscisse, on choisit un intervalle faisant apparaître les racines de g , par exemple `-10..5`

```
> plot(g(x), x=-10..5);
```



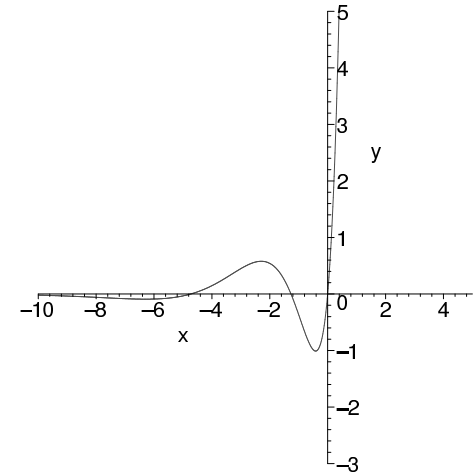
Du coup, il faut choisir un intervalle en y adapté, par exemple :

```
> plot(g(x), x=-10..5, y=-3..5);
```



Enfin, on demande à Maple de tracer la courbe avec 1000 points :

```
> plot(g(x), x=-10..5, y=-3..5, numpoints=1000);
```



(Remarquer que la courbe obtenue est plus "lisse" que la précédente)