⧫ TP13 – Préparation à l'oral

I. — Utilisation de numpy pour la diagonalisation

Le module `numpy` contient la structure de données `matrix` qui permet de calculer simplement sur des matrices.

On importe le module `numpy` sous la forme :

```
from numpy import *
```

1) Création d'une matrice

On utilise l'instruction `matrix()` : entre les parenthèses, on indique la liste de lignes de la matrice, chaque ligne étant elle-même implémentée sous forme de liste.

Exemple. Considérons les matrices

$$A = \begin{pmatrix} 1 & 2 & -1 \\ 3 & 0 & 4 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & -0{,}4 \\ 0{,}5 & 1 \end{pmatrix}, \quad C = \begin{pmatrix} 2 \\ -1 \end{pmatrix}, \quad D = \begin{pmatrix} 2 & -1 \end{pmatrix}.$$

Pour saisir ces matrices, on écrit :

```
A = matrix([[1, 2, -1], [3, 0, 4]])
B = matrix([[1, -0.4], [0.5, 1]])
C = matrix([[2], [-1]])
D = matrix([[2, -1]])
```

Remarque. Après l'instruction `matrix`, il y a donc toujours une parenthèse et deux crochets.

2) Opérations sur les matrices

- Si A et B sont deux matrices de même taille, on calcule $A + B$ par l'instruction `A+B`.
- Si A est une matrice et k est un scalaire, on calcule kA par l'instruction `k*A`.
- Si A et B sont deux matrices telles que le produit AB soit défini, on calcule AB par l'instruction `A*B`.
- Si A est une matrice carrée et $n \in \mathbb{N}$, on calcule A^n par l'instruction `A**n`.
- Si A est une matrice carrée inversible, on calcule son inverse A^{-1} par l'instruction `A**(-1)`.

Exemple. Considérons les matrices

$$A = \begin{pmatrix} 1 & 2 \\ 3 & -4 \end{pmatrix}, \quad B = \begin{pmatrix} -1 & 1 \\ -2 & 3 \end{pmatrix}.$$

Implémenter ces matrices puis calculer à l'aide de `numpy`,

$$A + B \qquad 3A \qquad A \times B \qquad A^2 \qquad A^{-1}$$

Solution.

```
A=matrix([[1,2],[3,-4]])
B=matrix([[-1,1],[-2,3]])
```

```
>>> A+B
matrix([[ 0,  3],
        [ 1, -1]])
>>> 3*A
matrix([[  3,   6],
        [  9, -12]])
>>> A*B
matrix([[-5,  7],
        [ 5, -9]])
>>> A**2
matrix([[ 7, -6],
        [-9, 22]])
>>> A**(-1)
matrix([[ 0.4,  0.2],
        [ 0.3, -0.1]])
```

3) Résolution de systèmes linéaires

Soit $A \in \mathscr{M}_n(\mathbb{R})$ une matrice carrée inversible, $X \in \mathscr{M}_{n,1}(\mathbb{R})$ et $B \in \mathscr{M}_{n,1}(\mathbb{R})$. Pour résoudre le système matriciel $AX = B$, on utilise l'instruction :

```
linalg.solve(A, B)
```

Exemple. Considérons le système :

$$\begin{cases} 2x + y - z = 7 \\ x + 2y + z = 5 \\ 3x - y + 2z = -4 \end{cases}$$

On peut l'écrire sous forme matricielle $AX = B$ avec :

$$A = \begin{pmatrix} 2 & 1 & -1 \\ 1 & 2 & 1 \\ 3 & -1 & 2 \end{pmatrix}, \quad X = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad \text{et} \quad B = \begin{pmatrix} 7 \\ 5 \\ -4 \end{pmatrix}.$$

Dans l'interpréteur, on saisit

```
A = matrix([[2, 1, -1], [1, 2, 1], [3, -1, 2]])
B = matrix([[7], [5], [-4]])
linalg.solve(A, B)
```

et on obtient

```
matrix([[ 1.],
        [ 3.],
        [-2.]])
```

Ainsi, l'unique solution de (S) est $(x, y, z) = (1, 3, -2)$.

4) Valeurs propres et vecteurs propres

Soit A une matrice carrée à coefficients réels ou complexes. Les valeurs propres et les vecteurs propres de A peuvent s'obtenir par l'instruction

```
linalg.eig(A)
```

Exemple. On considère la matrice

$$M = \begin{pmatrix} 1 & -1 \\ 2 & 4 \end{pmatrix}.$$

Dans l'interpréteur, on saisit :

```
M = matrix([[1, -1], [2, 4]])
linalg.eig(M)
```

et on obtient :

```
(array([2., 3.]), matrix([[-0.70710678,  0.4472136 ],
        [ 0.70710678, -0.89442719]]))
```

Le tableau `array[2., 3.]` contient les valeurs propres : ici, il s'agit donc de 2 et 3 et la matrice `matrix([[-0.70710678, 0.4472136 ], [ 0.70710678, -0.89442719]]` contient, en colonne, des vecteurs propres associés à ces deux valeurs propres.

Remarques sur l'affichage.

— Les calculs sont approchés. On peut donc observer des arrondis. Par exemple, si une valeur propre vaut 2, Python peut afficher 2.000000002 ou 1.99999998.

— Si Python affiche un résultat très proche de zéro, comme par exemple $8.26466798 \times 10^{-9}$, il est possible qu'il s'agisse d'une erreur d'arrondi et que le résultat soit nul.

— Python donne des vecteurs propres unitaires, ce qui est peu pratique en raison des arrondis. Ainsi, on ne s'en servira pas pour déterminer des vecteurs propres. Cela peut cependant aider à vérifier un résultat. Ainsi, on peut montrer qu'un vecteur propre associé à la valeur propre 2 pour la matrice M est $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$ et on peut constater que la première colonne de la matrice fournit le vecteur $\begin{pmatrix} -0{,}70710678 \\ 0{,}70710678 \end{pmatrix}$ qui est bien colinéaire à ce vecteur.

— Python peut donner des résultats dans $\mathbb{C}$. En Python, on utilise la lettre `j` pour désigner le nombre complexe i. Par exemple, l'écriture `2+3j` désigne le nombre complexe $2 + 3\mathrm{i}$.

5) Résumé

Dans le tableau ci-dessous, A et B sont des matrices de dimensions appropriées, k est un scalaire et n un entier naturel.

Notation mathématique	Instruction Python
Définir une matrice $A = \begin{pmatrix} 1 & 2 \\ -3 & 4 \end{pmatrix}$	`A = matrix([[1, 2], [-3, 4]])`
Somme de deux matrices : $A + B$	`A+B`
Produit d'une matrice par un scalaire : kA	`k*A`
Produit de deux matrices : AB	`A*B`
Puissance : A^n	`A**n`
Inverse d'une matrice inversible : A^{-1}	`A**(-1)`
Résolution du système $AX = B$	`linalg.solve(A, B)`
Valeurs propres et vecteurs propres de A	`linalg.eig(A)`

6) Exercices

Exercice 1. On considère la matrice $A = \begin{pmatrix} -2 & 1 \\ 1 & -3 \end{pmatrix}$.

1. À la main, justifier que A est inversible et donner A^{-1}.

2. À l'aide de `numpy`, calculer A^{-1}

Solution.

1. Comme $\det(A) = (-2) \times (-3) - 1 \times 1 = 5 \neq 0$, A est inversible et $A^{-1} = \frac{1}{5}\begin{pmatrix} -3 & -1 \\ -1 & -2 \end{pmatrix} = \begin{pmatrix} -0{,}6 & -0{,}2 \\ -0{,}2 & -0{,}4 \end{pmatrix}$.

2. On saisit

```
A=matrix([[-2,1],[1,-3]])
A**(-1)
```

et on obtient :

```
matrix([[-0.6, -0.2],
        [-0.2, -0.4]])
```

Exercice 2. On considère les matrices :

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 2 & 1 & 2 \\ 0 & 1 & 1 \end{pmatrix} \quad \text{et} \quad P = \begin{pmatrix} 1 & 1 & 1 \\ 0 & -2 & 2 \\ -1 & 1 & 1 \end{pmatrix}.$$

1. À l'aide de `numpy`, calculer $Q = P^{-1}$. Vérifier avec `numpy` que $PQ = I_3$ et $QP = I_3$.

2. À l'aide de `numpy`, calculer $D = P^{-1}AP$. Que peut-on en déduire ?

Solution.

1. On saisit

```
P=matrix([[1,1,1],[0,-2,2],[-1,1,1]])
Q=P**(-1)
print(Q)
```

et on obtient :

```
[[ 0.5    0.    -0.5 ]
 [ 0.25 -0.25   0.25]
 [ 0.25  0.25   0.25]]
```

puis

```
>>> P*Q
matrix([[1., 0., 0.],
        [0., 1., 0.],
        [0., 0., 1.]])
>>> Q*P
matrix([[1., 0., 0.],
        [0., 1., 0.],
        [0., 0., 1.]])
```

Ainsi, on a bien $PQ = QP = I_3$.

2. On saisit

```
A=matrix([[1,1,0],[2,1,2],[0,1,1]])
D=Q*A*P
print(D)
```

et on obtient :

```
[[ 1.   0.   0.]
 [ 0.  -1.   0.]
 [ 0.   0.   3.]]
```

Il s'agit d'une matrice diagonale diagonale donc A est diagonalisable et les valeurs propres de A sont 1, -1 et 3.

Exercice 3. On considère le système

$$(S)\begin{cases} 2x+2y+5z=14 \\ 4x-y+z=6 \\ x+\frac{1}{2}y+\frac{3}{2}z=\frac{9}{2} \end{cases}$$

1. Déterminer les matrices A et B telles que l'écriture matricielle de (S) soit $AX = B$. Saisir les matrices A et B dans l'interpréteur.

2. Résoudre le système à l'aide de l'instruction `linalg.solve` de `numpy`.

3. Calculer $X = A^{-1}B$ à l'aide de `numpy` et vérifier que l'on retrouve la solution de (S).

Solution.

1. L'écriture matricielle de (S) est $AX = B$ avec $A = \begin{pmatrix} 2 & 2 & 5 \\ 4 & -1 & 1 \\ 1 & \frac{1}{2} & \frac{3}{2} \end{pmatrix}$ et $B = \begin{pmatrix} 14 \\ 6 \\ \frac{9}{2} \end{pmatrix}$.

2. On saisit

```
A=matrix([[2,2,5],[4,-1,1],[1,1/2,3/2]])
B=matrix([[14],[6],[9/2]])
linalg.solve(A,B)
```

et on obtient :

```
matrix([[ 0.5],
        [-1. ],
        [ 3. ]])
```

Ainsi, l'unique solution de (S) est $(\frac{1}{2}, -1, 3)$.

3. On saisit

```
A**(-1)*B
```

et on obtient :

```
matrix([[ 0.5],
        [-1. ],
        [ 3. ]])
```

On retrouve que l'unique solution de (S) est $(\frac{1}{2}, -1, 3)$.

Exercice 4. On considère les matrices

$$M = \begin{pmatrix} 3 & 1 & -1 \\ 1 & 3 & -1 \\ 1 & 1 & 1 \end{pmatrix}, \quad P = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

On peut démontrer que

- M possède deux valeurs propres : 2 et 3.
- le sous-espace propre associé à 2 est :

$$E_2(M) = \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \;\middle|\; z = x + y \right\} = \mathrm{Vect}(u, v) \quad \text{avec} \quad u = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \quad \text{et} \quad v = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}.$$

- le sous-espace propre associé à 3 est $E_3(M) = \mathrm{Vect}(w)$ avec $w = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$.

1. À l'aide de `numpy`, déterminer les valeurs propres de M.

2. Vérifier que les vecteurs propres donnés par `numpy` sont cohérents avec les espaces propres donnés par l'étude mathématique.

3. On note $D = P^{-1}MP$. Sans calculs, que vaut D ? Vérifier à l'aide de `numpy`.

4. On note Q la matrice des coordonnées des vecteurs propres donnée par `numpy`. On note $D' = Q^{-1}MQ$. Sans calculs, que vaut D' ? Vérifier à l'aide de `numpy`.

Solution.

1. On saisit

```
M=matrix([[3,1,-1],[1,3,-1],[1,1,1]])
linalg.eig(A)
```

et on obtient :

```
(array([3., 2., 2.]), matrix([[ 5.77350269e-01, -3.20493781e-16,  8.06898221e-01],
        [ 5.77350269e-01, -7.07106781e-01, -2.95345247e-01],
        [ 5.77350269e-01, -7.07106781e-01,  5.11552974e-01]])
)
```

On en déduit que le spectre de M est $\{3 ; 2\}$.

2. Le vecteur propre associé à 3 est (environ) $\begin{pmatrix} 0{,}577350269 \\ 0{,}577350269 \\ 0{,}577350269 \end{pmatrix}$ qui est bien colinéaire à w.

En réalité, la valeur exacte du vecteur propre est $\dfrac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$.

Les vecteurs propres associés sont (environ) $\begin{pmatrix} 0 \\ -0{,}707106781 \\ -0{,}707106781 \end{pmatrix}$ et $\begin{pmatrix} 0{,}806898221 \\ -0{,}295345247 \\ 0{,}511552974 \end{pmatrix}$ et leurs coordonnées vérifient $z = x + y$, ce qui est cohérent. En fait, la valeur exacte du premier vecteur est $\dfrac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}$. Celle du deuxième est plus mystérieuse...

3. Comme P est la matrice de passage de la base canonique de $\mathscr{M}_{3,1}(\mathbb{R})$ à (u, v, w), $D = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}$.
Si on ajoute, dans l'interpréteur,

```
P=matrix([[1,0,1],[0,1,1],[1,1,1]])
print(P**(-1)*M*P)
```

et on obtient :

```
[[2. 0. 0.]
 [0. 2. 0.]
 [0. 0. 3.]]
```

comme annoncé.

4. Pour la même raison, on va obtenir $D' = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}$ et on ajoute, dans l'interpréteur,

```
Q=matrix([[ 5.77350269e-01, -3.20493781e-16,  8.06898221e
   -01],
        [ 5.77350269e-01, -7.07106781e-01, -2.95345247e-01],
        [ 5.77350269e-01, -7.07106781e-01,  5.11552974e-01]])
print(Q**(-1)*M*Q)
```

et on obtient :

```
[[ 3.00000000e+00  0.00000000e+00  4.44089210e-16]
 [ 4.44089210e-16  2.00000000e+00  2.22044605e-16]
 [-2.22044605e-16  4.44089210e-16  2.00000000e+00]]
```

ce qui, aux erreurs d'arrondis près, confirme la valeur de D'.

II. — Utilisation de GeoGebra en analyse

GeoGebra est un logiciel qui permet de construire des figures géométriques planes ainsi que des courbes représentatives de fonctions. Il dispose également d'un tableur et d'un module de calcul formel.

1) Présentation

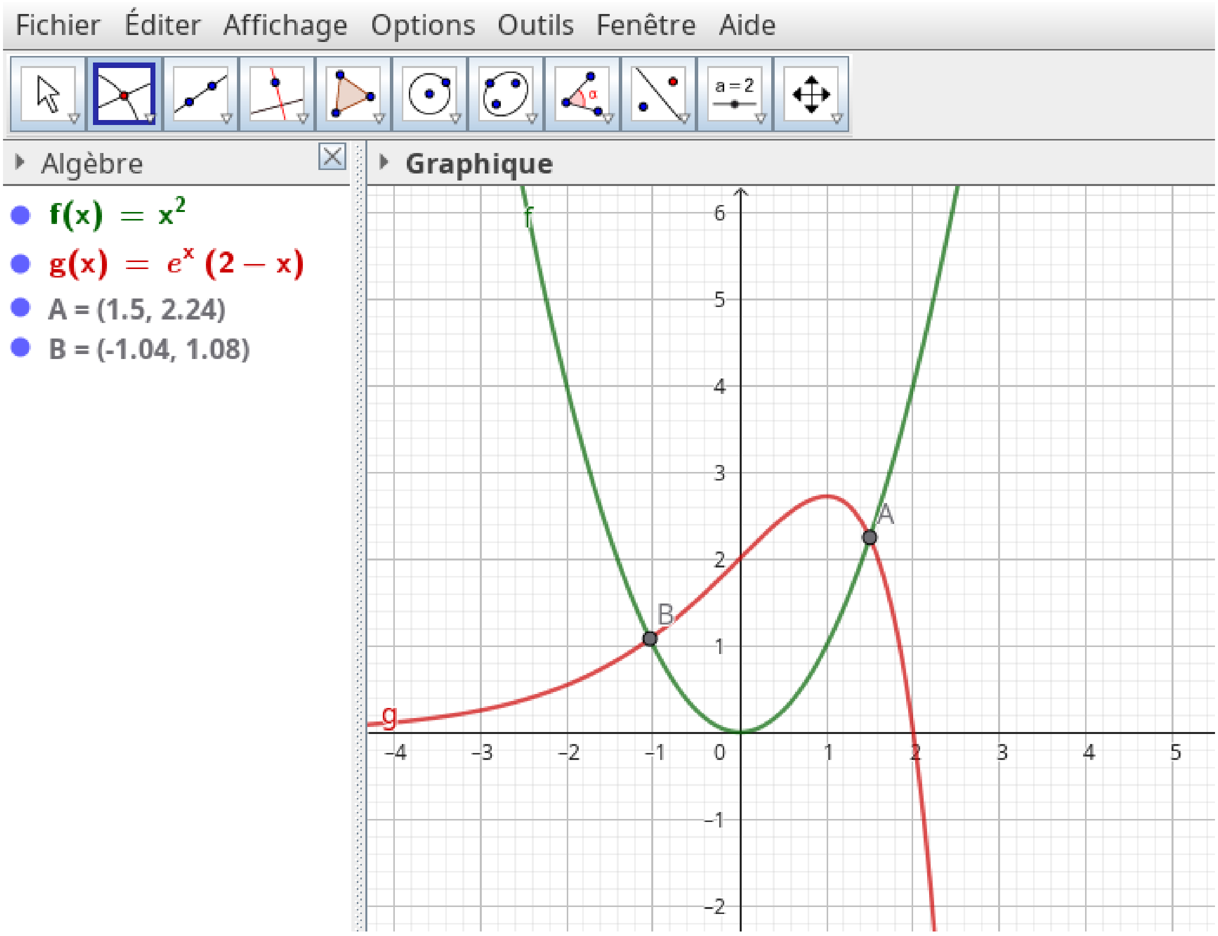
Lorsqu'on démarre GeoGebra, une fenêtre apparaît avec
— à droite : la partie Graphique pour faire des figures ;
— à gauche : la partie Algèbre où apparaissent les coordonnées des points ainsi que les expressions des fonctions représentées ;
— en haut : en dessous de la barre de menus, des boutons qui correspondent à d'autres menus déroulants ;
— en bas : un champ de saisie où l'on peut écrire directement les points, les fonctions, etc.

2) Représentation graphique d'une fonction

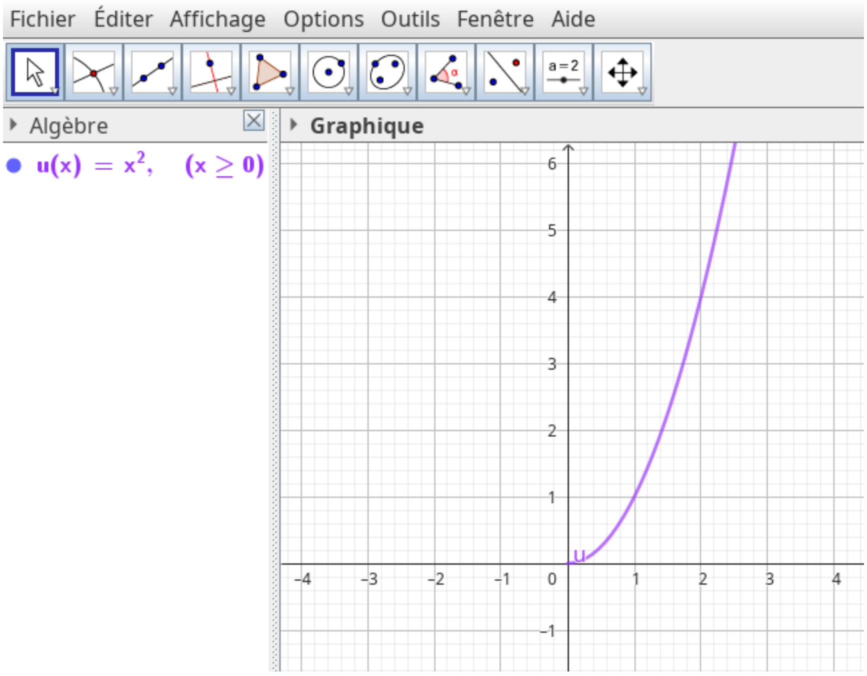
Considérons les fonctions f et g définies sur $\mathbb{R}$ par

$$f(x) = x^2 \quad \text{et} \quad g(x) = \mathrm{e}^x(2 - x).$$

1. Dans le champ de saisie, taper `f(x)=x^2` puis appuyer sur $\boxed{\text{Entrée}}$. La courbe représentative de f s'affiche. On peut aussi taper `x^2` dans le champ de saisie et GeoGebra trace la fonction en lui attribuant un nom. Il est possible d'utiliser une autre variable que x (par exemple : `f(t)=t^2`).
2. Taper de même `g(x)=exp(x)*(2-x)` et appuyer sur $\boxed{\text{Entrée}}$. La courbe représentative de g s'affiche.
3. **a.** Afficher le menu déroulant du deuxième bouton en cliquant sur la petite flèche (qui devient rouge) en bas à droite du carré qui sert d'icône.
   b. Sélectionner l'outil « Intersection ».
   c. Sur le graphique, cliquer sur le point d'intersection le plus à droite. Ses coordonnées apparaissent dans la fenêtre Algèbre. On obtient $A(1{,}5\,;2{,}24)$.
   d. Cliquer de même sur le point d'intersection le plus à gauche. On obtient $B(-1{,}04\,;1{,}08)$.



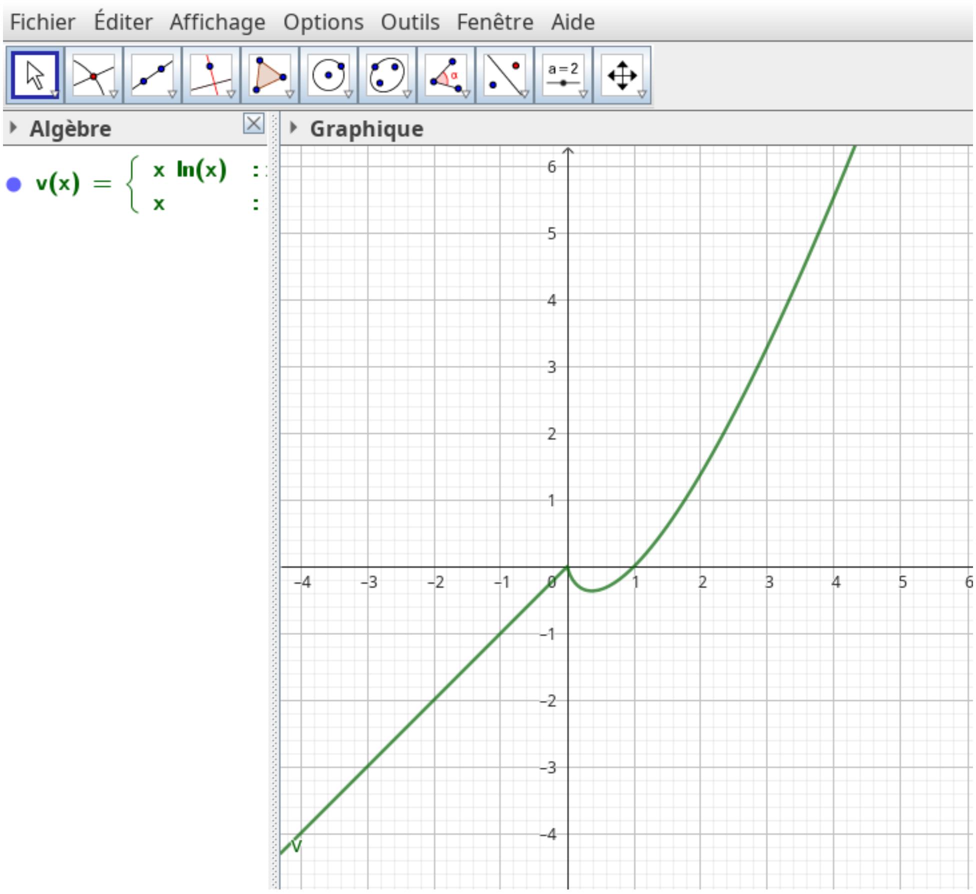
4. Il est possible de tracer la courbe d'une fonction sur un intervalle donné ou d'une fonction définie par morceaux.
   a. Pour tracer la courbe représentative de la fonction u définie sur $[0\,;+\infty[$ par $u(x) = x^2$, saisir `u(x)=Si(x>=0,x^2)` et appuyer sur $\boxed{\text{Entrée}}$.



b. Pour tracer la courbe représentative de la fonction v définie sur $\mathbb{R}$ par

$$v(x) = \begin{cases} x\ln(x) & \text{si } x > 0, \\ x & \text{si } x \leqslant 0. \end{cases}$$

saisir `v(x)=Si(x>0,x*ln(x),x)` et appuyer sur $\boxed{\text{Entrée}}$.



3) Calcul formel

GeoGebra possède un outil de calcul formel. On l'active en cochant Calcul formel dans le menu Affichage.

1. Pour calculer la valeur de $\displaystyle\int_0^1 x^2\,\mathrm{d}x$, saisir `Intégrale(x^2,0,1)` puis appuyer sur $\boxed{\text{Entrée}}$. On obtient $\frac{1}{3}$.

Il est possible de calculer des intégrales impropres. Par exemple, pour calculer $\displaystyle\int_0^{+\infty} \mathrm{e}^{-x^2}\,\mathrm{d}x$, saisir `Intégrale(exp(-x^2),0,+∞)` puis appuyer sur $\boxed{\text{Entrée}}$. Le symbole ∞ est disponible en cliquant sur l'icône $\boxed{\alpha}$ à droite de la cellule. On obtient $\frac{1}{2}\sqrt{\pi}$.

2. Pour déterminer une primitive de ln sur $]0\,;+\infty[$, on peut soit saisir `Intégrale(ln(x))` soit saisir `ln(x)` et cliquer sur Primitive dans le menu déroulant $\boxed{\texttt{f'}}$. On obtient, dans le deux cas, $x\ln(x) - x + c_1$.

3. De la même façon, on peut calculer la dérivée de $x \longmapsto \ln(x^2+1)$ en saisissant `Dérivée(ln(x^2+1))` ou en utilisant le bouton $\boxed{\texttt{f'}}$. On obtient $2\cdot\frac{x}{x^2+1}$.

4. GeoGebra permet également de calculer des limites. Par exemple,

- **a.** pour obtenir $\lim\limits_{x\to+\infty} \frac{2x}{x-1}$, on saisit `Limite(2*x/(x-1),+∞)` qui affiche 2.
- **b.** Il est également possible de calculer des limites à droite ou à gauche en remplaçant `Limite` par `LimDroite` ou `LimGauche`.

Remarque. Si on doit effectuer plusieurs calculs sur une même fonction f, on a intérêt à la définir en tapant dans le champ de saisie, par exemple, `f(x)=2*x/(x-1)`. On peut ensuite calculer sa dérivée, ses limites en saisissant `Dérivée(f(x))`, `Limit(f(x),−∞)`, `LimGauche(f(x),1)`, etc.

4) Exercices

Chaque exercice se fera dans une fenêtre GeoGebra distincte.

Exercice 5. On considère la fonction f définie sur $\mathbb{R}$ par

$$f(x) = x^3 + x^2 + x + 2.$$

1. Tracer la courbe représentative de f à l'aide de GeoGebra.

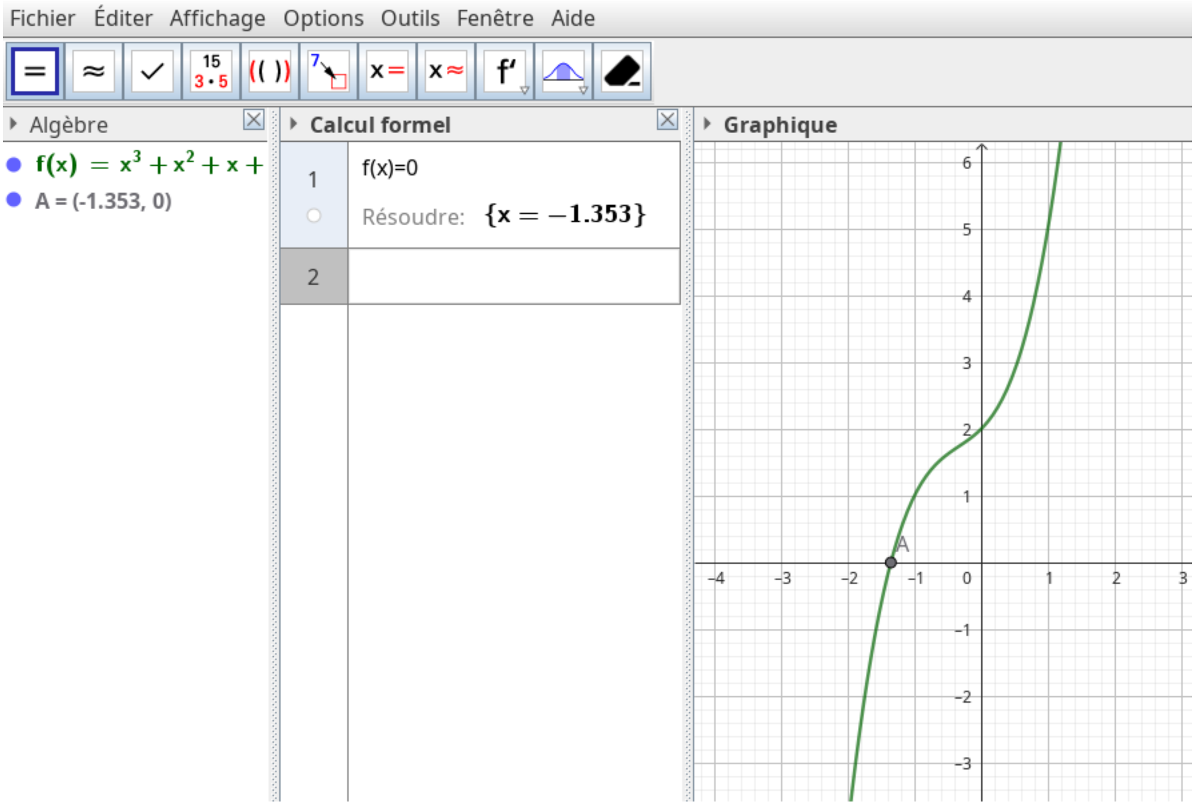
2. On peut démontrer que l'équation $f(x) = 0$ admet une unique solution α dans $\mathbb{R}$. À l'aide de GeoGebra, donner une valeur approchée de α à 10^{-3} près :

- **a.** en utilisant la courbe de f ;
- **b.** en utilisant le bouton $\boxed{\text{x=}}$ de l'outil de calcul formel.

Remarque. On peut régler la précision d'affichage des valeurs numériques en sélectionnant « Arrondi » dans le menu Options.

Solution.

1. On obtient la courbe suivante :



2. **a.** On construit le point d'intersection entre la courbe de f et l'axe des abscisses. L'abscisse de ce point est une valeur approchée de α. On trouve $\alpha \approx -1{,}353$.

   b. En utilisant l'outil de Calcul formel, on entre l'équation `f(x)=0` et on clique sur l'icône $\boxed{\text{x=}}$.

Exercice 6. Soit f et g les fonctions définies sur $\mathbb{R}$ par

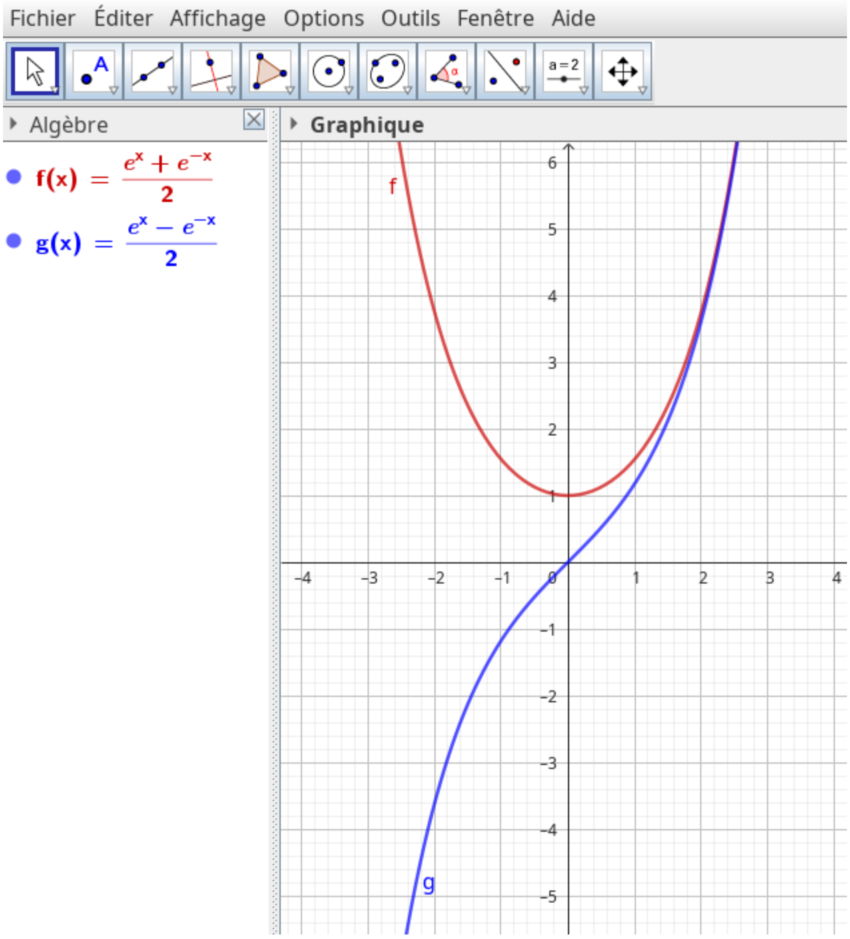
$$f(x) = \frac{\mathrm{e}^x + \mathrm{e}^{-x}}{2} \quad \text{et} \quad g(x) = \frac{\mathrm{e}^x - \mathrm{e}^{-x}}{2}.$$

On note C_1 et C_2 leurs courbes représentatives dans un repère orthonormé.

1. Tracer C_1 et C_2 à l'aide de GeoGebra.
2. Que peut-on conjecturer graphiquement concernant $\lim\limits_{x \to +\infty} [f(x) - g(x)]$?

Solution.

1. Voir page suivante.
2. À partir du graphique, on peut conjecturer que $\lim\limits_{x \to +\infty} [f(x) - g(x)] = 0$.



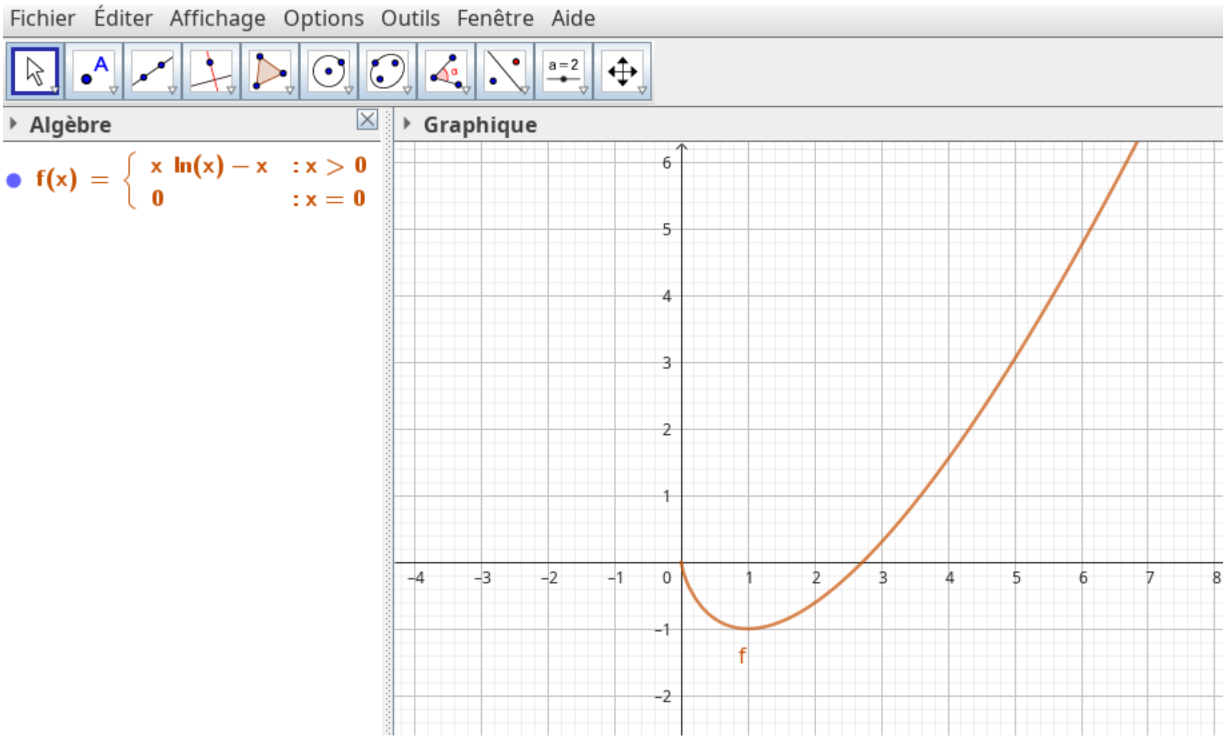
Exercice 7. Soit f la fonction définie sur $[0\,;+\infty[$ par

$$f(x)=\begin{cases} x\ln(x)-x, & \text{si } x>0,\\ 0 & \text{si } x=0.\end{cases}$$

1. Tracer la courbe représentative C de f. La fonction f semble-t-elle continue en 0 ?
2. Soit S l'aire, en unité d'aire, du domaine limité par C, l'axe des abscisses et les droites d'équation $x=1$ et $x=e$.
   a. Exprimer S à l'aide d'une intégrale.
   b. Calculer la valeur exacte de S à l'aide de l'outil de calcul formel.

Solution.

1. On saisit `f(x)=Si(x>=0,Si(x>0, x*ln(x)-x),0)` et on obtient la courbe suivante de la page suivante. Graphiquement, f semble continue.
2. a. Graphiquement, il semble de f soit négative sur $[1\,;e]$. C'est bien le cas car, pour tout $x>0$, $f(x)=x(\ln(x)-1)$ et si $x\in[1\,;e]$, $x>0$ et $\ln(x)\leqslant 1$ donc $f(x)\leqslant 0$. Ainsi, $S=\displaystyle\int_1^e -f(x)\,\mathrm{d}x$.
   b. On saisit `Intégrale(-(x*ln(x)-x),1,exp(1))` dans l'outil de Calcul formel et on trouve que $S=\dfrac{e^2-3}{4}$.
   Remarque. Il semble que la double condition « Si » dans la définition de f empêche la syntaxe `Intégrale(-f(x),1,exp(1))` de fonctionner ici.



Exercice 8. Soit f la fonction définie sur $\mathbb{R}$ par $f(x) = \dfrac{2x}{x^2+1}$.

À l'aide de GeoGebra, déterminer

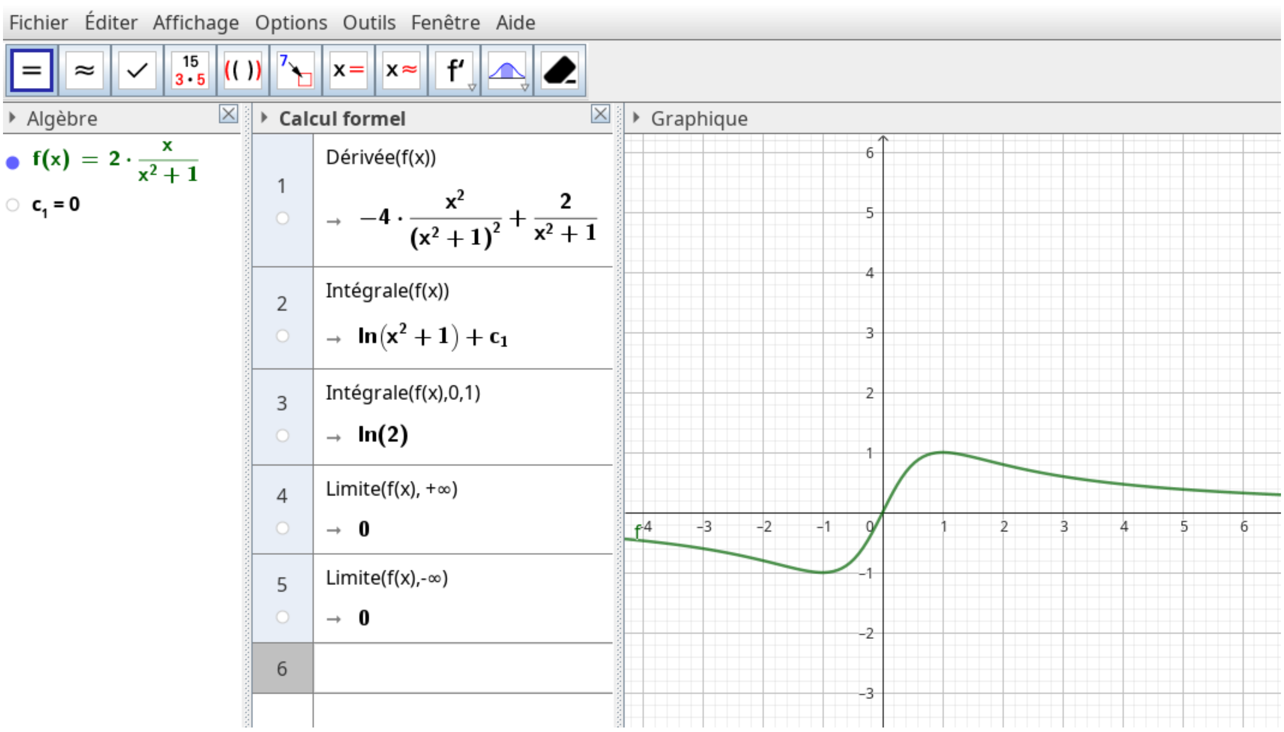
1. la dérivée de f sur $\mathbb{R}$;

2. une primitive de f sur $\mathbb{R}$;

3. la valeur exacte de $\displaystyle\int_0^1 f(x)\,\mathrm{d}x$;

4. déterminer le comportement de $f(x)$ quand x tend vers $+\infty$ puis vers en $-\infty$.

Solution. On obtient les résultats suivants :



qui permettent de dire que :

1. pour tout réel x, $f'(x) = \dfrac{-4x^2}{(x^2+1)^2} + \dfrac{2}{x^2+1}$

2. la fonction $F : x \longmapsto \ln(x^2+1)$ est une primitive de f sur $\mathbb{R}$;

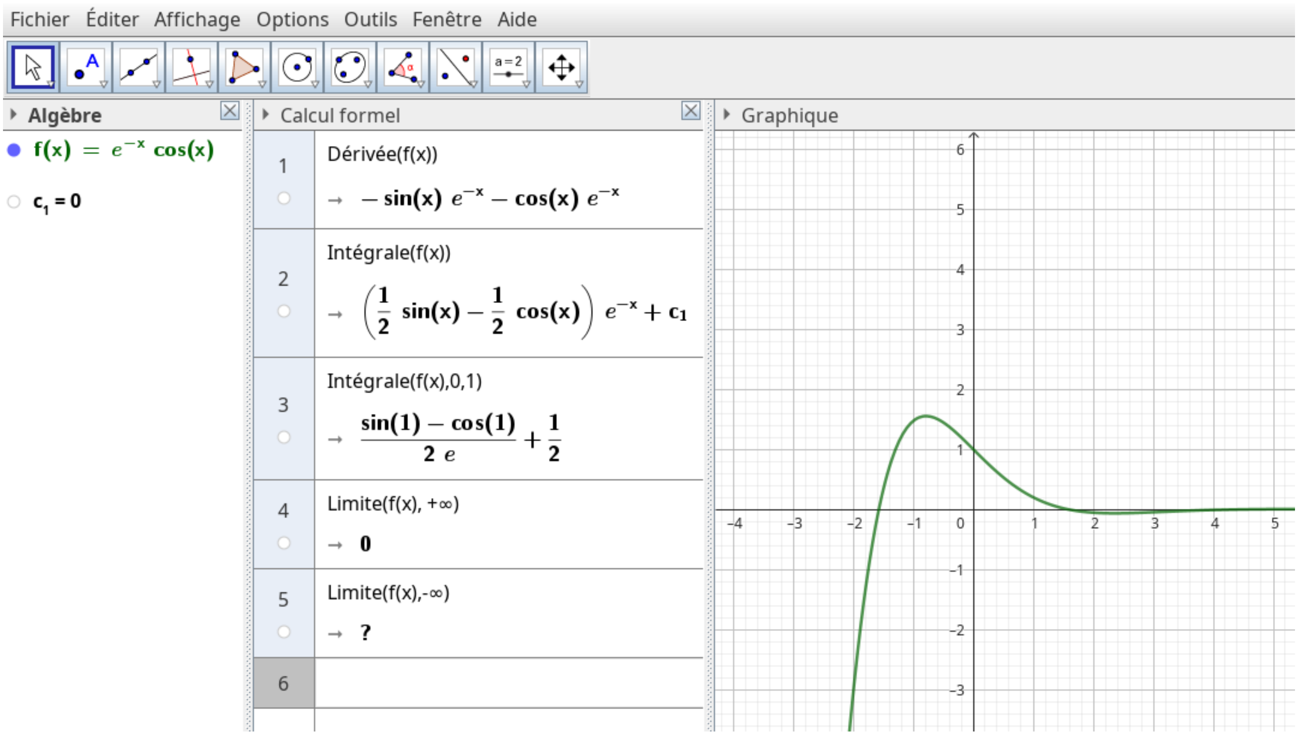
3. $\displaystyle\int_0^1 f(x)\,\mathrm{d}x = \ln(2)$;

4. $\displaystyle\lim_{x\to+\infty} f(x) = \lim_{x\to-\infty} f(x) = 0$.

Exercice 9. Reprendre les questions de l'exercice précédent avec la fonction f définie sur $\mathbb{R}$ par $f(x) = \exp(-x)\cos(x)$.

Que constate-t-on concernant le comportement de f au voisinage de $-\infty$? Expliquer.

Solution. On obtient les résultats suivants :



qui permettent de dire que :

1. pour tout réel x, $f'(x) = -\sin(x)\mathrm{e}^{-x} - \cos(x)\mathrm{e}^{-x}$

2. la fonction $F : x \longmapsto \left(\dfrac{1}{2}\sin(x) - \dfrac{1}{2}\cos(x)\right)\mathrm{e}^{-x}$ est une primitive de f sur $\mathbb{R}$;

3. $\displaystyle\int_0^1 f(x)\,\mathrm{d}x = \frac{\sin(1)-\cos(1)}{2e} + \frac{1}{2}$;

4. $\displaystyle\lim_{x\to+\infty} f(x) = 0$.

On remarque que GeoGebra ne parvient pas à déterminer la limite de f en $-\infty$. Cela est rassurant car cette limite n'existe pas. En effet, pour tout entier $n \in \mathbb{N}$, $f(-2n\pi) = \exp(2n\pi)$ tend vers $+\infty$ quand n tend vers $+\infty$ et $f(-(2n+1)\pi) = -\exp((2n+1)\pi)$ tend vers $-\infty$ quand n tend vers $+\infty$. Or, comme les deux suites $(-2n\pi)$ et $(-(2n+1)\pi)$ tendent vers $-\infty$, si f avait une limite L (finie ou infinie) en $-\infty$ alors, par composition, on aurait $\displaystyle\lim_{n\to+\infty} f(-2n\pi) = \lim_{n\to+\infty} f(-(2n+1)\pi)$.

III. — Quelques exemples d'étude de suites

Dans cette partie, on utilise Python pour effectuer certains calculs sur les suites.

Exercice 10. Soit (u_n) la suite définie par

$$\forall n \in \mathbb{N}, \quad u_n = n^3 - 4n.$$

1. Écrire une fonction Python `suite1`, d'argument $n \in \mathbb{N}$, qui renvoie la valeur de u_n.

2. Donner les valeurs exactes de u_{10}, u_{20} et u_{30}.

Solution.

```
def suite1(n):
    return n**3 - 4*n
```

`suite1(10)` donne $u_{10} = 960$, `suite1(20)` donne $u_{20} = 7920$ et `suite1(30)` donne $u_{30} = 26880$.

Exercice 11. Soit (u_n) la suite définie par $u_0 = 2$ et

$$\forall n \in \mathbb{N}, \quad u_{n+1} = \frac{2u_n + 1}{u_n + 1}.$$

1. Écrire un programme qui affiche les 10 premiers termes de la suite (de u_0 à u_9).

2. Écrire une fonction `suite2`, d'argument $n \in \mathbb{N}$, qui calcule et renvoie u_n.

Solution.

1.

```
u = 2
for k in range(10):
    print(u)
    u = (2*u + 1) / (u + 1)
```

2.

```
def suite2(n):
    u = 2
    for k in range(n):
        u = (2*u + 1) / (u + 1)
    return u
```

Exercice 12. Soit (u_n) la suite définie par $u_0 = 1$ et

$$\forall n \in \mathbb{N}, \quad u_{n+1} = u_n^2 - n.$$

Écrire une fonction `suite3`, d'argument un entier $n \in \mathbb{N}$, qui renvoie la valeur de u_n.

Solution.

```
def suite3(n):
    u = 1
    for k in range(n):
        u = u*u - k
    return u
```

Exercice 13. On considère les suites (u_n) et (v_n) définies par $u_0 = 2$ et $v_0 = 0$

$$\forall n \in \mathbb{N}, \quad \begin{cases} u_{n+1} = u_n + v_n \\ v_{n+1} = 2u_n. \end{cases}$$

Écrire une fonction `suite4`, d'argument $n \in \mathbb{N}$, qui calcule et renvoie u_n et v_n.

Solution.
Première méthode :

```
def suite4(n):
    u = 2
    v = 0
    for k in range(n):
        w = u
        u = u + v
        v = 2 * w
    return u, v
```

Seconde méthode :

```
def suite4(n):
    u, v = 2, 0
    for k in range(n):
        u, v = u + v, 2 * u
    return u, v
```

Exercice 14. On considère la suite (u_n) définie par $u_0 = 1$, $u_1 = 3$ et

$$\forall n \in \mathbb{N}, \quad u_{n+2} = u_{n+1} + u_n.$$

Écrire une fonction `suite5`, d'argument un entier $n \in \mathbb{N}$, qui calcule et renvoie u_n.

Solution.
Première méthode :

```
def suite5(n):
    u = 1
    v = 3
    for k in range(n):
      w = v
      v = u + v
      u = w
    return u
```

Seconde méthode :

```
def suite5(n):
    u, v = 1, 3
    for k in range(n):
        u, v = v, u + v
    return u
```

Exercice 15. Soit (u_n) la suite définie par

$$\begin{cases} u_0 = 1 \\ \forall n \in \mathbb{N} \quad u_{n+1} = u_n + \dfrac{4}{u_n}. \end{cases}$$

On admet que $\lim\limits_{n \to +\infty} u_n = +\infty$.

1. Déterminer le plus petit entier naturel n tel que $u_n > 100$.

2. Soit $A \in]0\,;+\infty[$. Écrire une fonction `seuil` prenant en argument un réel A et qui renvoie le plus petit entier naturel n tel que $u_n > A$.

Solution.

1. On peut utiliser le programme suivant :

```
u = 1
n = 0
while u <= 100:
    u = u + 4/u
    n += 1
print(n)
```

Le programme donne la valeur $n = 1247$.

2.

```
def seuil(A):
    u = 1
    n = 0
    while u <= A:
        u = u + 4/u
        n += 1
    return n
```

Exercice 16. Pour tout $n \in \mathbb{N}^*$, on pose

$$S_n = \sum_{k=1}^{n} \frac{1}{k}.$$

1. Écrire une fonction `somme` prenant en argument un entier naturel n non nul et qui renvoie la valeur de S_n.

2. Donner des valeurs approchées à 10^{-3} près de S_{10}, S_{100} et S_{1000}.

3. Justifier l'existence d'un entier N tel que $S_N > 10$. Déterminer le plus entier N vérifiant cette inégalité.

Solution.

1.

```python
def somme(n):
    S = 0
    for k in range(1, n+1):
        S = S + 1/k
    return S
```

2. On trouve les valeurs $S_{10} \approx 2{,}929$, $S_{100} \approx 5{,}187$ et $S_{1000} \approx 7{,}485$.

3. La série harmonique diverge vers $+\infty$ donc $\lim\limits_{n \to +\infty} S_n = +\infty$ donc il existe un entier N tel que $S_N > 10$. En utilisant le programme

```python
N = 1
while somme(N) <= 10:
    N += 1
print(N)
```

on trouve $N = 12367$.
On peut noter que ce programme n'est pas optimal car il réitère des calculs déjà effectué à chaque tour alors qu'on peut tirer parti du fait que, pour tout $n \in \mathbb{N}^*$, $S_{n+1} = S_n + \dfrac{1}{n+1}$ pour écrire un programme plus efficace :

```python
S = 1
N = 1
while S <= 10:
    N += 1
    S + = 1/(N+1)
print(N)
```