

Nom :

Prénom :

Matricule :

## MATH2010-1 Logiciels mathématiques Examen final pratique – groupe A – 17 juin 2016

Répondre aux questions suivantes. Vous avez droit à vos notes, à internet, mais pas aux moyens de communication de tout type (courriel, téléphone, etc.) Vos réponses doivent être manuscrites (rédigées à la main) et utiliser Sympy sous Python 3 ou Mathematica. Vous devez écrire à la main la réponse ainsi que la démarche pour l'obtenir, c'est-à-dire le code ou les calculs que vous avez utilisés. Si vous utilisez Python, vous pouvez supposer que toutes les fonctions de sympy ont été importées et que les variables  $x$  et  $n$  ont été définies :

```
from sympy import *
from sympy.abc import x, n
```

Vous devez mentionner toutes les autres variables et fonctions que vous importez.

**Question 1** (2 pts). En exécutant le code ci-dessous en SymPy, qu'est-ce que la ligne `print(expr)` imprime à l'écran? Expliquer pourquoi.

```
from sympy.abc import x
expr = x + 1
x = 2
print(expr)
```

Réponse :  $x+1$

Explications : La variable `expr` contient la somme du *symbole*  $x$  et du nombre 1 et n'a pas de référence à la *variable*  $x$ . Modifier la *variable*  $x$  n'a donc pas d'effet sur le contenu de la variable `expr`.

Est-ce que Mathematica se comporte de la même façon ?

Réponse :

Non. Dans Mathematica, il n'y a pas de distinction entre une *variable* et un *symbole*.

**Question 2** (1 pts). Écrire l'expression suivante sous un dénominateur commun et factoriser ce dénominateur :

$$\frac{13}{1764(x+5)} - \frac{1}{36(x-1)} + \frac{1}{49(x-2)} + \frac{1}{42(x+5)^2}.$$

Démarche et réponse :

```
>>> from sympy.abc import x
>>> from sympy import ratsimp
>>> A = 13/(1764*(x+5)) + 1/(42*(x+5)**2) - 1/(36*(x-1)) + 1/(49*(x-2))
>>> ratsimp(A).factor()
      1
-----
      2
(x - 2)*(x - 1)*(x + 5)
```

**Question 3** (1 pts). Décomposer l'expression suivante en somme de fractions rationnelles :

$$f(x) = \frac{11x^4 - 143x^3 + 249x^2 + 2191x - 6790}{x^5 - 22x^4 + 147x^3 - 166x^2 - 1376x + 3360}$$

Démarche et réponse :

```
>>> from sympy.abc import x
>>> numerateur = 11*x**4 - 143*x**3 + 249*x**2 + 2191*x - 6790
>>> denominateur = x**5 - 22*x**4 + 147*x**3 - 166*x**2 - 1376*x + 3360
>>> f = numerateur / denominateur
>>> f.apart()
      1      3      7      5
----- + ----- + ----- + -----
  x + 3      2      x - 7      x - 10
            (x - 4)
```

**Question 4** (1 pts). Calculer  $\int_{-1}^1 f(x) dx$  pour la fonction rationnelle  $f(x)$  définie à la question précédente.

Démarche et réponse :

```
>>> from sympy import integrate
>>> integrate(f, (x,-1,1))
-7*log(8) - 5*log(11) - log(4) - 2/5 + log(2) + 5*log(9) + 7*log(6)
```

**Question 5** (2 pts). Calculer la série suivante, en fonction de  $x$  :

$$\sum_{n=0}^{\infty} \frac{(n+2)x^{n+3}}{n!}$$

Démarche et réponse :

```
>>> from sympy.abc import x, n
>>> from sympy import factorial, oo, summation
>>> summation((n+2)*x**n/factorial(n), (n, 0, oo))
      x      x
x*e  + 2*e
```

**Question 6** (2 pts). Le nombre 2520 est le plus petit nombre divisible par 1, 2, 3, ..., 10. Quelle est le plus petit nombre divisible par 1, 2, 3, ..., 20 ?

Démarche et réponse :

```
>>> from sympy import lcm
>>> lcm(range(1,11))
2520
>>> lcm(range(1,21))
232792560
```

```
>>> def plus_petit_commun_multiple(x, y):
...     multiple_de_x = x
...     while not multiple_de_x % y == 0:
...         multiple_de_x += x
...     return multiple_de_x
>>> R = 1
>>> for a in range(1,21):
...     R = plus_petit_commun_multiple(R,a)
>>> R
232792560
```

**Question 7** (1 pts). La somme des chiffres de  $2^{15} = 32768$  est égale à  $3 + 2 + 7 + 6 + 8 = 26$ . Quelle est la somme des chiffres de  $2^{1000}$  ?

Démarche et réponse :

```
>>> L = [int(i) for i in str(2**15)]
>>> sum(L)
26
>>> L = [int(i) for i in str(2**1000)]
>>> sum(L)
1366
```

**Question 8** (1 pts). Chercher l'ensemble des nombres complexes  $z$  tels que  $|z| = |z - 1|$ .

Démarche et réponse :

```
>>> from sympy import solve, I, symbols, Eq
>>> a,b = symbols('a b', real=True)
>>> z = a + b*I
>>> eq = Eq(abs(z), abs(z-1))
>>> eq
```

$$\sqrt{a^2 + b^2} = \sqrt{a^2 - 2*a + b^2 + 1}$$

```
>>> solve(eq, (a,b))
[{a: 1/2}]
```

**Question 9** (1 pts). Trouver les racines complexes du polynôme suivant :

$$g(x) = x^5 - 7ix^4 + 9x^3 - 9ix^2 - 10x + 16i.$$

Démarche et réponse :

```
>>> from sympy import I
>>> from sympy.abc import x
>>> B = x**5 - 7*I*x**4 + 9*x**3 - 9*I*x**2 - 10*x + 16*I
>>> sympy.factor(B, gaussian=True)
(x - 1)*(x + 1)*(x - 8*I)*(x - I)*(x + 2*I)
```

**Question 10** (2 pts). Résoudre le système d'équations linéaires suivant :

$$\begin{cases} x + y + z + 2w = 5 \\ 3x + 5y + 3w = 4 \\ x + 4y + z + 3w = 1 \\ x + 4y + 2w = 1 \end{cases}$$

Démarche et réponse :

```
>>> from sympy import Matrix
>>> M = Matrix([
[0, 1, 1, 2, 5],
[3, 5, 0, 3, 4],
[0, 4, 1, 3, 1],
[1, 4, 0, 2, 1]])
>>> M.rref()
(Matrix([
[1, 0, 0, 0, -2],
[0, 1, 0, 0, -11/2],
[0, 0, 1, 0, -29/2],
[0, 0, 0, 1, 25/2]]), [0, 1, 2, 3])
```

**Question 11** (2 pts). La conchoïde de Sluze est une famille de courbes planaires étudiées en 1662 par René François Walter, baron de Sluze. Elles sont définies par l'équation polaire

$$r = \sec \theta + a \cos \theta.$$

Choisissez trois valeurs réelles du paramètre  $a$  et dessiner la courbe en fonction de celles que vous avez choisies. Pour quelles valeurs de  $a$  est-ce que la courbe s'intersecte ?

Démarche et réponse :

```
>>> from sympy import cos, sin, sec, pi
>>> from sympy.abc import theta, a
>>> r = sec(theta) + a*cos(theta)
>>> x = (r*cos(theta)).simplify()
>>> y = (r*sin(theta)).simplify()
>>> x
a*cos(theta)**2 + 1
>>> y
a*sin(theta)*cos(theta) + tan(theta)
>>> from sympy.plotting import plot_parametric
>>> plot_parametric(x.subs(a, -1.5), y.subs(a, -1.5), (theta, -pi, pi),
                    ylim=(-2,2), xlim=(-2,2))
>>> plot_parametric(x.subs(a, -0.8), y.subs(a, -0.8), (theta, -pi, pi),
                    ylim=(-2,2), xlim=(-2,2))
>>> plot_parametric(x.subs(a, 0.8), y.subs(a, 0.8), (theta, -pi, pi),
                    ylim=(-2,2), xlim=(-2,2))
```

**Question 12** (2 pts). Soit la matrice

$$A = \begin{pmatrix} 62 & -110 & -82 & 52 \\ 19 & 33 & -12 & -38 \\ -3 & 0 & -23 & 6 \\ 36 & -55 & -43 & 16 \end{pmatrix}.$$

Calculer le polynôme caractéristique et les valeurs propres de  $A$ . Calculer le vecteur propre à droite associé à la valeur propre dont le module est le plus grand.

Démarche et réponse :

```
>>> from sympy import Matrix
>>> from sympy import init_printing
>>> init_printing(pretty_print=True, use_unicode=False)
>>> M = Matrix([
[62, -110, -82, 52],
[19, 33, -12, -38],
[-3, 0, -23, 6],
[36, -55, -43, 16]])
>>> M.eigenvals()
{-22: 1, -11: 1, 33: 1, 88: 1}
>>> M.eigenvects()
[(-22, 1, [[3/2]]), (-11, 1, [[2/3]]), (33, 1, [[2]]), (88, 1, [[2]])]
      [ ]      [ ]      [ ]      [ ]
      [1/2]    [2/3]    [1]      [0]
      [ ]      [ ]      [ ]      [ ]
      [3/2]    [1/3]    [0]      [0]
      [ ]      [ ]      [ ]      [ ]
      [ 1 ]    [ 1 ]    [1]      [1]
```

**Question 13** (2 pts). Le 3 avril 2016, le Consortium international des journalistes d'enquête (ICIJ) a publié les premiers articles au sujet des *Panama Papers*<sup>1</sup>. Au départ, seuls les journalistes travaillant sous l'égide de l'ICIJ ont pu fouiller les données, mais il s'agit d'une tâche titanesque. Le 9 mai 2016, le consortium a décidé de faire appel au public pour pousser la recherche encore plus loin en publiant une base de données contenant des informations provenant des Panama Papers (sans les courriels et autres données personnelles). Celle-ci contient cinq fichiers au format `.csv` :

<code>Addresses.csv</code>	(25M)	<code>Entities.csv</code>	(102M)	<code>Officers.csv</code>	(41M)
<code>all_edges.csv</code>	(41M)	<code>Intermediaries.csv</code>	(3,5M)		

Répondre aux questions suivantes.

- En utilisant `pandas`, ouvrir le fichier `http://www.math.ulg.ac.be/hidden_files/Officers.csv` dans un tableau de données.
- Combien de lignes possède ce tableau ?
- Filtrer ce tableau pour conserver que les lignes dont la colonne `country_codes` est relative à la Belgique (ignorer le cas ou plus d'un pays apparaît).
- En utilisant la méthode `value_counts` d'une série, afficher le nom des cinq officiers qui apparaissent le plus souvent dans le tableau filtré.

Démarche et réponse :

```
>>> import pandas as pd
>>> url = "http://www.math.ulg.ac.be/hidden_files/Officers.csv"
>>> df = pd.read_csv(url)

>>> len(df)
345594

>>> df_belgique = df[df.country_codes=='BEL']

>>> df_belgique.name.value_counts().head(5)
THE BEARER 36
MICHAEL GERHARD JOHANNES GLEISSNER 8
ALGONQUIN TRUST, PANAMA 2
Franck Robert WYSTYRK 2
Mr. Michael Bernard ATWOOD 2
Name: name, dtype: int64
```

---

1. Les *Panama Papers* désignent la fuite de plus de 11,5 millions de documents confidentiels issus du cabinet d'avocats panaméen Mossack Fonseca, détaillant des informations sur plus de 214 000 sociétés établies dans des paradis fiscaux ainsi que les noms de ceux qui y sont associés.