MATLAB - TP n°5

Exercice 1

Ecrire les expressions symboliques des fonctions suivantes à la ligne de commande de Matlab, puis calculer leurs dérivées :

1) $c^{a.x}$
2) e^x
3) ln(x)
4) sin(x)
5) cos(x)
6) tan(x)
7) cot(x)
8) sec(x)
9) csc(x)

Exercice 2

Executer les commandes Matlab suivantes :

```
>> syms x
>> expand((x-5)*(x+9))
>> expand((x+2)*(x-3)*(x-5)*(x+7))
>> expand(sin(2*x))
>> expand(cos(x+y))
>> collect(x^3 *(x-7))
>> collect(x^4*(x-3)*(x-5))
>> factor(x^3 - y^3)
>> factor([x^2-y^2,x^3+y^3])
>> simplify((x^4-16)/(x^2-4)
```

Exercice 3

Soit la fonction $y = 3\sin x + 7\cos(5x)$. Est-ce que $y" + y = -5\cos(2x)$ est vraie ? Pour y répondre, créer et exécuter le script suivant :

```
syms x
y = 3*sin(x)+7*cos(5*x);   % définition de la fonction y
lhs = diff(y,2)+y;         % évaluation du lhs de l'équation
rhs = -5*cos(2*x);         % évaluation du rhs de l'équation
if(isequal(lhs,rhs))
    disp('Oui, l''equation est vérifiée');
else
    disp('Non, l''equation n'est pas vérifiée');
end
disp('Valeur du LHS est: '), disp(lhs);
```

Exercice 4

A l'aide de la fonction *limit* de Matlab, vérifier les propriétés élémentaires suivantes des limites :

$$\lim_{x\to p}[f(x)+g(x)] = \lim_{x\to p} f(x) + \lim_{x\to p} g(x) \qquad \lim_{x\to p}[f(x)\cdot g(x)] = \lim_{x\to p} f(x) \cdot \lim_{x\to p} f(x)$$

$$\lim_{x\to p}[f(x)-g(x)] = \lim_{x\to p} f(x) - \lim_{x\to p} g(x) \qquad \lim_{x\to p}[f(x)/g(x)] = \lim_{x\to p} f(x) \,/ \lim_{x\to p} g(x)$$

appliquées aux fonctions ci-dessous :

- $f(x) = (3x+5)/(x-3)$
- $g(x) = x^2 + 1$

Exercice 5

Créer les variables symboliques a, b, x, n, t et theta (substitue de la lettre grecque θ) dans Matlab, puis à l'aide de la fonction *diff* calculer les dérivées des fonctions ci-dessous :

1) $2x^n$
2) $\sin(a\,t^2 + b)$
3) $e^{(i\,\theta^2)}$

Noter que le symbole 'i' n'est pas déclaré : il s'agit du nombre complexe $i^2=-1$.
Si nécessaire, utiliser la fonction *simplify* pour simplifier le résultat obtenu.

Exercice 6

La fonction *diff* peut prendre une matrice symbolique comme argument ; dans ce cas, la différenciation est donnée élément par élément. Créer puis exécuter le script ci-dessous :

```
syms a x
A = [cos(a*x),sin(a*x);-sin(a*x),cos(a*x)]
diff(A)
```

Exercice 7

Executer la fonction *simple* sur chacune des fonctions suivantes :

1) cos(x)^2 + sin(x)^2
2) 2*cos(x)^2 – sin(x)^2
3) cos(x)^2 – sin(x)^2
4) cos(x) + (-sin(x)^2)^(1/2)
5) cos(x) + i*sin(x)
6) cos(3*acos(x))

Exercice 8

a) Vérifier que la somme S_1 suivante

$$S_1 = 1 + \frac{1}{2^2} + \frac{1}{2^3} + \cdots$$

est égale à $\pi^2/6$ à l'aide des commandes :

```
>> syms x k
>> S1 =symsum(1/k^2, 1,
inf)
```

b) Idem pour

$$S_2(x) = 1 + x + x^2 + \cdots$$

```
>> S2 = symsum(x^k, k, 0,
inf)
```

c) Obtenir le developpement en serie de Taylor d'ordre 8 de la fonction

$$f(x) = \frac{1}{5 + 4\cos(x)}$$

à l'aide des instructions suivantes :

```
>> sym x
>> f = 1 / (5 + 4 * cos(x)
) ;
>> T = taylor(f,8)
>> pretty(T)  % afin
d'avoir un affichage plus
lisible
```

Exercice 9

Ecrire puis executer le script suivant :

```
f = x^2*cos(x);
ezplot(f, [-4,9])
a = int(f, -4, 9)
disp('Surface = '),
disp(double(a));
```

Exercice 10

Résoudre les équations suivantes avec la fonctions *solve* de Matlab:

1) $x^2 - ax + b = 0$
2) $(x - c)^2(x - d) = 0$

Exercice 11

Exécuter les commandes suivantes :

```
>> syms x y
>> f = x^2 + 2*x*y - 9
>> solve(f)
```

```
>> f1 = x*y - 5 ;
>> f2 = x^2 - y + 6 ;
>> [x,y] = solve(f1,f2)
```