Affectations conditionnelles (fonctions définies par morceaux)

Pour programmer la fonction

$$f(x) = \begin{cases} \exp(-x^2) & \text{si } x > 0 \\ 1 & \text{si } x \leq 0 \end{cases}$$

on peut utiliser la fonction `If` en écrivant

```
f[x_]:=If[x>0,Exp[-x^2],1]
```

On pourra préférer la syntaxe suivante :

```
f[x_/;condition]:=expression[x]
```

ou encore utiliser la fonction `Which` (voir l'aide pour la syntaxe)

Évaluer les entrées suivantes :

```
Clear[f]
f[x_/;x>0]:=Exp[-x^2];
f[x_/;x<=0]:=1;
Plot[f[x],{x,-5,5}]
```

Tracer le graphe sur $[-3,3]$ de la fonction f définie par

$$f(x) = \begin{cases} 0 & \text{si } x = 1 \text{ ou } -1 \\ e^{-|x|} & \text{si } x < -1 \\ e^{-|x|} & \text{si } x > 1 \\ x^2 & \text{si } -1 < x < 1 \end{cases}$$

Définition récursive

On dit qu'une fonction f est définie de manière *récursive* lorsque la définition de f fait appel à f.
La fonction factorielle, définie sur $\mathbb{N}$, peut être définie de manière récursive de la façon suivante :

$$\text{fact}(0) = 1 \qquad \text{et} \qquad \text{fact}(n) = n \cdot \text{fact}(n-1).$$

On pourra programmer cette fonction en écrivant :

```
fact[0]=1
fact[n_]:=n*fact[n-1]
```

1. Programmer de manière récursive la fonction $n \mapsto 1 + 2 + \cdots + n$.
2. Soit f la fonction définie sur $\mathbb{R}$, périodique de période 1 et définie sur $[0,1[$ par $f(x) = x(1-x)$.
   Programmer la fonction, de manière récursive, et tracer son graphe sur $[-3,3]$.
   ```
   Clear[f]
   f[x_/;(0<=x)&&(x<1)]:=x*(1-x)
   f[x_/;x<0]:=......
     ...  ...  ...
   ```

Vecteurs et Matrices

Pour *Mathematica* les vecteurs ou les matrices sont représentés par des *listes*. Ainsi une matrice est une liste de vecteurs de même longueur.
Évaluer, une à une, les entrées suivantes :

```
vecteur1={1,2,3,4}
vecteur2={e1,e2,e3,e4}
maMatrice={{1,2,3},{4,5,6},{7,8,9}}
MatrixForm[maMatrice]
maMatrice[[2]]
maMatrice[[2,1]]
```

La fonction `Dimensions[A]` donne le nombre de lignes et de colonnes de la matrice `A` tandis que `Length[A]` donne le nombre d'éléments de la liste, c'est-à-dire le nombre de lignes de la matrice `A`.
La fonction `MatrixForm` permet de voir la matrice sous sa forme traditionnelle de tableau.
Certaines matrices usuelles comme par exemple la matrice identité, les matrices diagonales ...sont directement accessibles par les fonctions `IdentityMatrix`, `DiagonalMatrix`.... (voir l'aide pour la syntaxe)
La fonction `Table` est précieuse pour la construction de certaines matrices.
Évaluer, une à une, les entrées suivantes :

```
Table[i,{i,1,3},{j,1,4}]//MatrixForm
Table[Random[],{i,1,3},{j,1,4}]//MatrixForm
```

Programmer des fonctions *Mathematica*, d'argument d'entrée n, permettant de construire les matrices suivantes :

1. la matrice $(a_{ij}) \in \mathcal{M}_n(\mathbb{R})$ où $a_{ij} = 1$.
2. La matrice de Hilbert $(\frac{1}{i+j-1})_{1\leq i,j\leq n} \in \mathcal{M}_n(\mathbb{R})$.
3. Les matrices $\begin{pmatrix} 1 & 1 & \dots & \dots & 1 \\ 0 & 1 & 1 & \dots & 1 \\ \vdots & & & & \vdots \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}$, $\begin{pmatrix} 1 & 2 & \dots & \dots & n \\ 0 & 1 & 2 & \dots & n-1 \\ \vdots & & & & \vdots \\ 0 & 0 & \dots & \dots & 1 \end{pmatrix}$ et $\begin{pmatrix} 0 & 1 & 1 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ 1 & 1 & 0 & \dots & 1 \\ & & & & \\ 1 & 1 & \dots & 1 & 0 \end{pmatrix} \in \mathcal{M}_n(\mathbb{R})$.

Opérations sur les vecteurs et les matrices

La plupart des fonctions mathématiques, en particulier `Plus`, `Times`, `Power`, sont *listables* : leur action sur une liste s'effectue sur chaque composante de la liste. Ceci s'applique en particulier aux matrices. Évaluer :

```
vecteur1 + vecteur2
c*vecteur2
l=Table[2*k*Pi/6,{k,1,6}]; Sin[l]
A=Table[1,{3},{3}]
A*A
```

Il en résulte que l'on peut ajouter (`A+B`) ou multiplier (`A*B`) des matrices de mêmes dimensions ; l'opération indiquée se fera terme à terme !
Noter que le produit matriciel entre deux matrices `A` et `B` est calculé en évaluant `A.B` ou `Dot[A,B]` et que `MatrixPower[A,n]` calcule A^n pour une matrice carrée A.
Lorsque la matrice A est inversible, les commandes `Inverse[A]` et `MatrixPower[A,-1]` sont équivalentes.

Attention à la fonction `MatrixForm` ! Après évaluation de `B=MatrixForm[A]`, *Mathematica* ne donne pas le statut de matrice à `B` et les fonctions du type `MatrixPower`, `Inverse`, ... deviennent alors inopérantes.

1. Soit $R = \begin{pmatrix} 5 & 2 \\ -4 & -1 \end{pmatrix}$. Vérifier que $R^2 - 4R = -3I$. En déduire, sans *Mathematica* que R est inversible et que R^{-1} peut s'exprimer à l'aide de R.
   Retrouver l'inverse de R à l'aide *Mathematica*.
2. Déterminer le « noyau » de $A = \begin{pmatrix} 1 & -2 & 3 & 2 \\ 0 & 1 & -1 & -1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$. (voir `NullSpace`)
   Noter l'abus de langage. On demande le noyau de l'application linéaire canoniquement associée à A.

Exercices divers au choix

1. *(Extrait écrit Centrale 2012)*
   Pour tout suite réelle $x = (x_k)_{k\geq 0}$ et tout entier naturel n non nul, on note $H_n(x)$ la matrice de $\mathcal{M}_n(\mathbb{R})$ définie par
   $$\forall (i,j) \in \{1,\dots,n\}^2, \big(H_n(x)\big)_{i,j} = x_{i+j-2}.$$
   On a par exemple
   $$H_2(x) = \begin{pmatrix} x_0 & x_1 \\ x_1 & x_2 \end{pmatrix}, \ H_3(x) = \begin{pmatrix} x_0 & x_1 & x_2 \\ x_1 & x_2 & x_3 \\ x_2 & x_3 & x_4 \end{pmatrix}, \ H_4(x) = \begin{pmatrix} x_0 & x_1 & x_2 & x_3 \\ x_1 & x_2 & x_3 & x_4 \\ x_2 & x_3 & x_4 & x_5 \\ x_3 & x_4 & x_5 & x_6 \end{pmatrix}$$
   On considère désormais la suite $x = (x_n)$ définie par
   $$x_0 = 1,\ x_1 = 1,\ x_2 = 1,\ x_3 = 0,\ \text{et } \forall n \in \mathbb{N},\ x_{n+4} = x_{n+3} - 2x_{n+1}.$$
   (a) Écrire une fonction *Mathematica* de paramètre un entier naturel n renvoyant la liste des x_k pour $0 \leq k \leq n$, puis une fonction de paramètre, n entier non nul, et renvoyant la matrices $H_n(x)$ associée.
   (b) Déterminer le rang de $H_n(x)$, pour $n = 1, 2, \dots, 10$. (`MatrixRank`)

(c) On modifie la suite (x_n), en posant désormais $x_0 = 1/2$ et conservant les autres définitions. Déterminer le rang de $H_n(x)$, pour $n = 1, 2, \dots, 10$.

2. *(Extrait Oraux Centrale 2012)*
Soit $(u_n)_{n\geq 0}$ la suite définie par
$$u_0 = 9 \text{ et } \quad \forall n \geq 0, u_{n+1} = 4u_n^3 + 3u_n^4.$$
Pour n entier naturel, on note c_n le nombre de 9 à la fin de l'écriture décimale de u_n
(a) Déterminer c_n pour $n \in \{0, 1, \dots, 13\}$.
(b) Conjecturer, puis déterminer la valeur de c_n.

3. Construire une fonction 3-périodique vérifiant simultanément les conditions
– La restriction à l'intervalle $[0, 3]$ est polynomiale de degré 3.
– La fonction f est de classe $\mathcal{C}^1$ sur $\mathbb{R}$.
– $f(0) = 0$ et $f'(3) = 1$.
Tracer son graphe sur $[-3, 3]$. La fonction est-elle classe $\mathcal{C}^2$ sur $\mathbb{R}$?

4. *Algorithme d'Euclide*
Pour deux entiers naturels a et b, le calcul du PGCD de a et b par l'algorithme d'Euclide, repose sur la relation $\mathrm{PGCD}(a, b) = \mathrm{PGCD}(b, r)$ où r est le reste de la division de a par b.
De même, pour $a > b \geq 0$, $\mathrm{PGCD}(a, b) = \mathrm{PGCD}(b, a - b)$.
Programmer une fonction récursive, `pgcd[a_,b_]`, qui calcule le PGCD de a et b. Vérifier vos résultats avec la commande *Mathematica* `GCD`.

5. En utilisant l'identité $\binom{n}{p} = \frac{n}{p}\binom{n-1}{p-1}$. Programmer de manière récursive le calcul de $\binom{n}{p}$.

6. *(Extrait Oraux Centrale 2008)*
Soit $A = \begin{pmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix} \in \mathcal{M}_3(\mathbb{R})$ et $B = {}^tA$. (voir `Transpose`)

(a) Soit $K = \mathrm{vect}(A, B)$. Calculer AB, BA, A^2 et B^2. En déduire que K est stable par produit.
(b) Montrer que K est un corps.
Indication : qui est l'élément neutre pour la multiplication ?