

Prenez le temps de lire ce sujet. Les questions sont indépendantes pour la plupart.

## 1 Systèmes de réactions chimiques (14 points)

Un *système de réactions chimiques* est constitué d'un ensemble de *réactions chimiques élémentaires*. Une réaction chimique élémentaire est la donnée d'un ensemble de *réactants*, d'une *constante cinétique* et d'un ensemble de *produits*. La figure 1 représente un système de réactions chimiques généralisées<sup>1</sup> composé de quatre réactions élémentaires. Chaque réaction correspond à une flèche. Les réactants apparaissent en partie gauche des flèches, les constantes cinétiques au-dessus des flèches et les produits en partie droite. Les *espèces* chimiques (les réactants et les produits) sont  $A$ ,  $B$  et  $C$ . Les constantes cinétiques sont  $k_1$ ,  $k_2$ ,  $k_3$  et  $k_4$ .

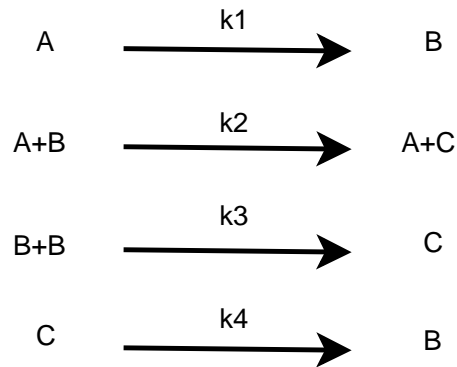


FIG. 1 – Un système de quatre réactions chimiques généralisées.

À tout système de réactions chimiques, on peut associer un système d'équations différentielles ordinaires. On introduit une fonction du temps pour chaque espèce chimique. Cette fonction représente la *concentration* de l'espèce. Sur l'exemple,  $A(t)$ ,  $B(t)$  et  $C(t)$  représentent les concentrations des espèces  $A$ ,  $B$  et  $C$ . Les membres droits des équations différentielles s'obtiennent par application de la loi d'action de masse. Le système d'équations différentielles obtenu sur l'exemple s'écrit :

$$\begin{aligned} \frac{dA}{dt}(t) &= -k_1 A(t), & \frac{dB}{dt}(t) &= k_1 A(t) - k_2 A(t) B(t) - 2k_3 B(t)^2 + k_4 C(t), \\ \frac{dC}{dt}(t) &= k_2 A(t) B(t) + k_3 B(t)^2 - k_4 C(t). \end{aligned}$$

<sup>1</sup>Une vraie réaction chimique est une réaction équilibrée (la masse totale des réactants est égale à celle des produits). Lorsqu'on s'affranchit de cette contrainte, la réaction est dite généralisée.

Ce système s'obtient par la formule :

$$V = N \cdot L$$

où  $V$  est le vecteur des *vitesses*,  $N$  est la matrice des *coefficients stœchiométriques* (une ligne par espèce et une colonne par réaction) et  $L$  le vecteur des *lois des réactions*. Sur l'exemple, voici  $V$ ,  $N$  et  $L$  :

> V, N, L;

```
[d      ]
[-- A(t)]
[dt      ] [ k1 A(t) ]
[      ] [-1  0  0  0] [      ]
[d      ] [      ] [k2 A(t) B(t)]
[-- B(t)], [ 1  -1  -2  1], [      ]
[dt      ] [      ] [      2 ]
[      ] [ 0  1  1  -1] [ k3 B(t) ]
[d      ] [      ] [      ]
[-- C(t)] [      ] [ k4 C(t) ]
[dt      ] [      ]
```

## Connaissance du logiciel

**Question 1** [1 pt]. Donner une commande MAPLE permettant de charger le paquetage *LinearAlgebra* en mémoire.

SOLUTION.

```
with (LinearAlgebra):
```

**Question 2** [1 pt]. Donner une commande MAPLE affectant à une matrice  $N$  la matrice donnée en exemple ci-dessus.

SOLUTION.

```
N := <<-1, 1, 0> | <0, -1, 1> | <0, -2, 1> | <0, 1, -1>>;
```

## Structures de données

On suppose les espèces chimiques enregistrées dans une liste, nommée *Especies*. Sur l'exemple, on a :

> Especies;

```
[A(t), B(t), C(t)]
```

Chaque réaction chimique est représentée par une liste à trois éléments : la liste des réactants, la constante cinétique et la liste des produits. Un système de réactions chimiques est une liste de réactions. Sur l'exemple, on a :

```

> Systeme;
[ [ [A(t)], k1, [B(t)] ],

  [ [A(t), B(t)], k2, [A(t), C(t)] ],

  [ [B(t), B(t)], k3, [C(t)] ],

  [ [C(t)], k4, [B(t)] ] ]

```

## Questions de programmation

Les questions qui suivent consistent à écrire une suite de fonctions MAPLE qui automatisent la construction de  $V$ ,  $N$  et  $L$ .

**Question 3** [2 pts]. Écrire une fonction MAPLE *vitesses*, paramétrée par une liste d'espèces, qui retourne le vecteur des vitesses.

```

> vitesses (Espèces);

          [d      ]
          [-- A(t)]
          [dt      ]
          [        ]
          [d      ]
          [-- B(t)]
          [dt      ]
          [        ]
          [d      ]
          [-- C(t)]
          [dt      ]

```

SOLUTION.

```

vitesses := proc (Espèces)
  local V, i;
  V := Vector[column] (nops (Espèces)):
  for i from 1 to nops (Espèces) do
    V [i] := diff (Espèces [i], t)
  od:
  V;
end:

```

**Question 4** [2 pts]. Écrire une fonction MAPLE *loi\_reaction*, paramétrée par une réaction chimique élémentaire et qui retourne la loi de la réaction, c'est-à-dire le produit de la constante cinétique avec tous les réactants.

```

> Systeme [2];
          [[A(t), B(t)], k2, [A(t), C(t)]]

> loi_reaction (Systeme [2]);
          k2 A(t) B(t)

```

SOLUTION.

```
loi_reaction := proc (reaction)
  local reactants, constante, resultat, i;
  reactants := reaction [1];
  constante := reaction [2];
  resultat := constante;
  for i from 1 to nops (reactants) do
    resultat := resultat * reactants [i]
  od;
  resultat
end:
```

**Question 5** [2 pts]. Écrire une fonction MAPLE *lois*, paramétrée par un système de réactions chimiques et qui retourne le vecteur des lois des réactions du système. Réutiliser la fonction *loi\_reaction*.

```
> lois (Systeme);
      [ k1 A(t) ]
      [          ]
      [k2 A(t) B(t)]
      [          ]
      [          2 ]
      [ k3 B(t) ]
      [          ]
      [ k4 C(t) ]
```

SOLUTION.

```
lois := proc (systeme)
  local i, L;
  L := Vector[column] (nops (systeme)):
  for i from 1 to nops (systeme) do
    L [i] := loi_reaction (systeme [i])
  od;
  L;
end:
```

### Matrice de stœchiométrie

La matrice de stœchiométrie  $N$  comporte une ligne par espèce et une colonne par réaction. Ligne  $\ell$ , colonne  $c$ , on trouve le *coefficient stœchiométrique* de la  $\ell$ -ème espèce dans la  $c$ -ième réaction. Ce coefficient est égal au nombre de molécules d'espèce  $\ell$  produites moins le nombre de molécules d'espèce  $\ell$  consommées, dans la réaction  $c$ . Calculons par exemple le coefficient stœchiométrique de  $B$  ( $\ell = 2$ ) dans la troisième réaction ( $c = 3$ ) du système donné en exemple. Nombre de molécules de  $B$  produites : 0. Nombre de molécules de  $B$  consommées : 2. On a donc  $N_{23} = 0 - 2 = -2$ .

On suppose pour commencer l'existence d'une fonction *coeff\_stoecho* paramétrée par une espèce, une réaction et qui retourne le coefficient stœchiométrique de l'espèce dans la réaction.

```

> Systeme [3];
                [[B(t), B(t)], k3, [C(t)]]

> coeff_stoechio (B(t), Systeme [3]);
                -2

```

**Question 6** [2 pts]. Écrire une fonction *matrice\_stoechio*, paramétrée par une liste d'espèces, un système de réactions chimiques et qui retourne la matrice de stœchiométrie du système. Utiliser la fonction *coeff\_stoechio*.

SOLUTION.

```

matrice_stoechio := proc (Especes, Systeme)
  local N, l, c;
  N := Matrix (nops (Especes), nops (Systeme)):
  for l from 1 to nops (Especes) do
    for c from 1 to nops (Systeme) do
      N [l,c] := coeff_stoechio (Especes [l], Systeme [c])
    od
  od:
  N;
end:

```

On s'intéresse maintenant à la fonction *coeff\_stoechio*. On suppose l'existence d'une fonction *nombre*, paramétrée par une espèce, une liste d'espèces et qui retourne le nombre de fois où l'espèce apparaît dans la liste.

**Question 7** [2 pts]. Écrire la fonction *coeff\_stoechio* paramétrée par une espèce, une réaction et qui retourne le coefficient stœchiométrique de l'espèce dans la réaction. Utiliser la fonction *nombre*.

SOLUTION.

```

coeff_stoechio := proc (espece, reaction)
  local reactants, produits;
  reactants := reaction [1];
  produits := reaction [3];
  nombre (espece, produits) - nombre (espece, reactants)
end:

```

**Question 8** [2 pts]. Écrire la fonction *nombre*, paramétrée par une espèce, une liste d'espèces et qui retourne le nombre de fois où l'espèce apparaît dans la liste.

SOLUTION.

```

nombre := proc (espece, liste)
  local i, resultat;
  resultat := 0;

```

```

for i from 1 to nops (liste) do
  if liste [i] = espece then
    resultat := resultat + 1
  fi
od;
resultat;
end:

```

## 2 Intégrateurs implicites d'équations différentielles (7 points)

La méthode d'Euler étudiée en cours est un intégrateur numérique d'équations différentielles *explicite*. Lorsque certaines constantes cinétiques sont beaucoup plus grandes que les autres, les systèmes d'équations différentielles obtenus à partir des systèmes de réactions chimiques peuvent causer de grandes difficultés aux intégrateurs explicites : il faut un pas  $h$  très petit (et donc un temps de calcul très grand) pour garantir une précision raisonnable.

Une famille d'intégrateurs numériques donne de bien meilleurs résultats sur de tels systèmes (systèmes *raides* ou *stiff*). Ce sont les intégrateurs *implicites*.

Soient  $f : \mathbb{R} \rightarrow \mathbb{R}$  une fonction d'une variable réelle,  $[a, b]$  un intervalle de temps,  $N$  un nombre de pas et  $h = (b - a)/N$  la longueur d'un pas. Considérons l'équation différentielle ordinaire (autonome, pour simplifier) avec condition initiale :

$$\frac{dx}{dt}(t) = f(x(t)), \quad x(a) = \alpha \in \mathbb{R}.$$

**Question 9** [1 pt]. Rappeler la suite  $(x_n)$  définie par la méthode d'Euler *explicite* étudiée en cours.

SOLUTION.  $x_{n+1} = x_n + h f(x_n)$ .

La méthode d'Euler *implicite* consiste à calculer les termes de la suite définie par :

$$x_{n+1} = x_n + h f(x_{n+1}).$$

À la différence de la méthode vue en cours, où  $x_{n+1}$  est une fonction de  $x_n$ , dans cette méthode—ci,  $x_{n+1}$  est défini *implicite*ment comme une racine de l'équation :  $x_{n+1} - x_n - h f(x_{n+1}) = 0$ . Pour éviter certaines confusions, notons  $x_{old}$  au lieu de  $x_n$  et  $x_{new}$  au lieu de  $x_{n+1}$ . Connaissant  $h$ ,  $f$  et  $x_{old}$ , on cherche donc à calculer  $x_{new}$  tel que

$$g(x_{new}) \stackrel{\text{def}}{=} x_{new} - x_{old} - h f(x_{new}) = 0.$$

La méthode de Newton résout ce problème.

**Question 10** [2 pts]. Donner la suite  $(y_n)$  définie par la méthode de Newton, qui permet de calculer  $x_{new}$ . Sachant que  $x_{new}$  est proche de  $x_{old}$ , quelle valeur prendre pour  $y_0$  ?

SOLUTION. La suite est définie par  $y_{n+1} = y_n - g(y_n)/g'(y_n)$  avec  $g'(y_n) = 1 - h f'(y_n)$ .

On peut prendre  $y_0 = x_{old}$ .

**Question 11** [2 pts]. En MAPLE, on suppose que des valeurs sont affectées aux variables  $xold$ ,  $f$  et  $h$ . On suppose que la variable  $epsilon$  contient un petit réel  $\varepsilon > 0$ . Donner une suite de commandes (pas une fonction), calculant le premier terme de la suite  $(y_n)$  tel que  $|g(y_n)| < \varepsilon$ .

SOLUTION.

```
g := xnew -> xnew - xold - h*f(xnew);
dg := D(g);
y := xold;
while abs (g(y)) >= epsilon do
  y := y - g(y)/dg(y)
od;
```

**Question 12** [2 pts]. Modifier la suite de commandes précédente pour que les calculs s'arrêtent dès que  $|g(y_n)| < \varepsilon$  ou  $n > 10$ .

SOLUTION.

```
g := xnew -> xnew - xold - h*f(xnew);
dg := D(g);
y := xold;
n := 0;
while abs (g(y)) >= epsilon and n <= 10 do
  y := y - g(y)/dg(y);
  n := n+1
od;
```