Maîtrise d'informatique Paris 6
Module Calcul Formel

Renaud RIOBOO
Annick VALIBOUZE

Utilisation de *MAPLE*

MAPLE est un système de Calcul Formel facile à utiliser tournant sur des PC compatibles, Stations de travail, et macintoshes. Ce système réalise des calculs symboliques (calcul différentiel, intégration, calcul de limites, trigonométrie, matrice et calcul vectoriel, etc...), pour résoudre des problèmes numériques, pour tracer des courbes en deux ou trois variables et pour définir soit même de nouvelles fonctions. *MAPLE* est le logiciel de calcul formel le plus usuel offrant une large bibliothèque.

1. Principe généraux

MAPLE est un logiciel interactif, comme le sont la plupart des systèmes de calcul formel. Son apprentissage permettra d'utiliser d'autres systèmes avec un investissement supplémentaire négligeable.

Pour ouvrir une session *MAPLE* sur l'ibm2 il faut que /usr/local/bin soit dans votre path unix. Vous tapez alors simplement `maple` (ou le chemin complet `/usr/local/bin/maple` si vous n'avez pas le bon path).

Pour entrer une ligne à *MAPLE* il faut la faire suivre d'un retour chariot.

Une fois sous *MAPLE* pour en sortir il suffit de taper : `quit;`

1.1. initiation à l'aide en ligne.

- tapez `??`,
- lisez l'introduction,
- exécutez les commandes conseillées,
- cherchez comment sauver dans un fichier ce que vous avez fait, et comment charger un fichier de commandes *MAPLE* ensuite.
- examinez la librairie maple "linalg'; comment faire pour la charger dans *MAPLE*?

Le but de ces premiers TDs est de se familiariser avec les notions que vous venez de consulter. Quittez *MAPLE*, faites `man maple` pour avoir quelques renseignements supplémentaires.

2. Exercices d'initiation

Dans cette série d'exercices nous donnons volontairement des informations partielles afin de vous permettre d'explorer *MAPLE*. Nous vous donnons les informations essentielles afin de vous orienter dans votre recherche.

2.1. Simplification et dévellopement.

- Entrez le polynôme $P(X) = X^2 - X - 1$, affectez le à une variable sans l'afficher.
- substituez Y à X dans P
- résolvez l'équation $P(Y) = 0$,

- mettez la solution sous forme de liste, et extrayez les 2 racines du polynôme P,
- vérifiez que ces 2 nombres sont bien les racines de P.
- faites de même avec $P(X) = X^2 + aX + c$

2.2. Integration.

- Calculez la primitive de $\sin^2 x$,
- vérifiez que le résultat est correct en dérivant puis en simplifiant,
- faites de même avec des fonctions plus compliquées : $\sin 1/x$, $\frac{1}{P}$, $\frac{1}{(x^2+1)P}$

2.3. Algèbre linéaire.

- Entrez deux matrices A et B 3x3 de valeurs numériques.
- calculez A^2 puis le produit AB,
- calculez le déterminant de B,
- calculez le polynôme caractéristique de A,
- factorisez le polynôme,
- calculez le polynôme caractéristique de B,
- à quelle condition les matrices A et B ont des valeurs propres communes ? (**indication :** pgcd de deux polynômes).

Utilisation de *MAPLE*(suite)

Nous allons voir comment se comporte *MAPLE* de façon un peu plus précise.

3. L'évaluation sous *MAPLE*

Les expressions sous *MAPLE* sont complètement évaluées :

- tapez `x := y;`
- tapez `x ;`
- tapez `y := 3;`
- que vaut x et pourquoi ?
- faites `?quote`, trouvez comment on peut manipuler des symboles,
- désafectez x et y.

4. Les expressions *MAPLE*

- faites `?op`
- soit le polynôme $p(x) = (x - y + 1)^5$, quel est le troisième terme de l'expression " expandée " de p ?
- combien $p(x)$ a-t-il de termes ?
- quel est le type de $p(x)$, de factor($p(x)$) ?

MAPLE possède des structures complexes : les séquences, les listes, les emsembles :

- faites `?sequence` , `?$`
- cherchez les puissances troisièmes des entiers entre 10 et 15,
- cherchez le pgcd de 12 et de tous les entiers entre 1 et 30, est-ce juste ?
- des listes ou des ensembles de ces différents termes,
- créez la liste $[x1, x2, \ldots, x30]$ de la façon la plus coute possible (13 caractères).

MAPLE possède aussi les notions de tableaux et de tables :

- faites `?array`
- créez un tableau vide de nom $a$ayant dix termes numérotés de 1 a 10,
- que contient a ? imprimez a,
- mettez des valeurs polynomiales en x dans a,
- faites `?map`
- creez le tableau b des dérivées en x des termes de a.

5. Programmation *MAPLE*

- faites `?statement`, `?proc`, regardez les principaux éléments de la syntaxe *MAPLE*,
- faites une boucle " for " pour créez la liste $[x1, x2, \ldots, x30]$,
- quelle est la valeur de votre variable de boucle ? annulez cette valeur,
- écrivez dans un fichier un petite procédure *MAPLE* qui calcule la suite de Fibonacci,
- faites `?option`, `?local`, ré-écrivez votre fonction fib le plus efficacement possible.

- programmez la suite de Fibonacci a l'aide de techniques matricielles (pensez à la récursion linéaire vue en Math-Info) quel est le résultat.

Il est vraisemblablement l'heure ! Si vous avez encore un peu de temps faites `?time` pour comparer les différentes versions de la suite de Fibonacci.

6. Instabilité numérique

On veut tester si $e^{\pi\sqrt{163}}$ est nul. Réaliser la suite de commandes suivantes et commentez :

```
reel :=exp(Pi*sqrt(163) ; evalf(",35) ; evalf(ceil("),35) ;
evalf(""-",35) ; evalf(reel,25) ; ceil(")-"
```

7. Courbes

Soient une droite f croissante en dimension 2. Définir une fonction de 5 paliers qui approxime f. Ecrire les deux fonctions en maple, et les tracer avec `plot` sur un même graphe et sur deux graphes différents. Sauver les en Postcript et aller voir dans le fichier poscript pour en modifier l'impression.

Maîtrise d'informatique Paris 6
Module Calcul Formel

Renaud RIOBOO
Annick VALIBOUZE

Mise en oeuvre de programmes à partir d'une formule

Fonctions symétriques

Soient les formules de Girard-Newton vraies pour tout $n \in N$:

$$(1) \qquad p_n - e_1 p_{n-1} + e_2 p_{n-2} - \cdots + (-1)^n n e_n = 0 \qquad ,$$

où $p_1, \ldots, p_m, \ldots$ sont les fonctions puissances et $e_1, e_2, \ldots, e_m, \ldots$ sont les fonctions symétriques élémentaires. Nous supposerons qu'il existe un indice n_0 à partir duquel $e_m = 0$ pour tout $m > n_0$.

(1) Soient `m` un entier naturel et `les_elem` formé de n_0 et des i premières fonctions symétriques élémentaires $e_1, \ldots, e_i$ où $i = inf(m, n_0)$. Ecrire une fonction `ele2pui (m,les_elem)` qui rend n_0 et $p_1, \ldots, p_m$ en fonction de $e_1, \ldots, e_i$ avec la même structure de données que `les_elem`.
(2) Ecrire la fonction réciproque `pui2ele(m,les_pui)` permettant de calculer les fonctions symétriques élémentaires à partir des fonctions puissances.
(3) Vérifier que `ele2pui(m,pui2ele(m,les_pui))=les_pui` et que de même `pui2ele(m,ele2pui(m,les_ele))=les_ele`.
(4) Montrer que toutes les fonctions puissances dépendent algébriquement des n_0 premières fonctions puissances.
(5) Ecrire une fonction permettant d'exprimer une fonction puissance en fonction des n_0 premières fonctions puissances.
(6) Ecrire une fonction `puireduc(m,les_pui)` qui calcule les m premières fonctions puissances en fonction des n_0 premières données dans `les_pui`.
(7) Traduire matriciellement les formules de Girard-Newton. Vous remarquerez que les fonctions demandées peuvent donc s'écrire également avec des calculs matriciels. Programmez cette autre méthode et comparez.

Dans toutes ces fonctions si l'utilisateur ne donne pas toutes les fonctions symétriques élémentaires ou toutes les fonctions puissances, elles devront être remplacées par des variables `em` ou `pm`. Par exemple, en supposant que `les_elem = [`$n_0, e_1, e_2, \ldots, e_i$`]`, l'appel à `ele2pui(5,[4,2])` devra rendre le même résultat que l'appel à `ele2pui(5,[4,2,e2,e3,e4])`. Pour compléter `les_pui`, il y a deux possibilités : la première est de faire comme pour `les_elem` et la seconde est d'utiliser `puireduc`. Il faudra faire les deux.

Maîtrise d'informatique Paris 6
Module Calcul Formel

Renaud RIOBOO
Annick VALIBOUZE

Recherche d'une équation différentielles minimale

Soit $P(x, y)$ un polynôme en les variables x et y. Nous le noterons $P(y)$ quand nous le considèrerons comme polynôme en la variable y à coefficients dans les polynômes en la variable x. Nous supposerons que $P(y)$ *n'a pas de racine multiple.*

Nous rappelons qu'une équations différentielles linéaires est donnée par $L(y) = 0$ où L est de la forme :

$$L(y) = y + b_1 y' + b_2 y'' + \cdots + b_m y^{(m)}, \tag{2}$$

où $y^{(k)}$ est la dérivée k-ième de $y = y(x)$ en x et $b_1, b_2, \ldots, b_m$ sont des polynômes en x à coefficients dans Q. L'entier m est appelé l'*ordre* de L.

Le but de ce TD est de calculer une l'équation différentielle linéaire L d'ordre minimal tel que toute solution $y_0(x)$ à $P(y) = 0$ soit également solution de $L(y) = 0$. Pour cela nous allons employer la méthode ci-dessous :

Soit $y_0 = y_0(x)$ une solution quelconque de $P(y) = 0$ (i.e. $P(x, y_0(x)) = 0$).

1-
Calculer la dérivée du polynôme $P(x, y)$ en x

2-
En déduire y_0' en fonction de y_0

3-
Chercher, par identification, s'il existe une relation linéaire entre y_0 et y_0' : $y_0 + b_1 y_0' = 0$.
(On utilisera la relation trouvée en 2- pour ce ramener à une équation polynomiale en y_0, puis on utilisera l'hypothèse $P(y_0) = 0$ et enfin on remarquera qu'un polynôme de degré strictement plus petit que $P(y)$ ne peut avoir autant de racines de $P(y)$ sans être nul). Si on ne trouve pas de b_1 solution polynômiale en x alors on passe à l'étape suivante.

4-
Recommencer avec y" en dérivant P une seconde fois et chercher s'il existe b_1, b_2 des polynômes en x tels que $y + b_1 y' + b_2 y'' = 0$.

Ainsi de suite on finira par trouver une équation différentielle L comme en (2).

(1) Mettre en oeuvre l'algorithme à l'aide des polynômes $P_1(x, y) = y + x^2$ puis $P_2(x, y) = y^2 + (2x + 2)y + x - 1$ et $P_3(x, y) = y^4 + (3x + 1)y^3 - x^3 - 1$ en *MAPLE*.
(2) Décrire proprement l'algorithme.
(3) Ecrire le programme en *MAPLE*.

indications : Faire ?students, ?assign, ?coeff, ?changevar

A propos de Math-Info faire : ?rsolve

Maîtrise d'informatique Paris 6
Module Calcul Formel

Renaud RIOBOO
Annick VALIBOUZE

Le Calcul Modulaire

Le calcul modulaire permet de faire plusieurs "petits" calculs afin d'en éviter un "gros".

1. Rappel de cours

Soient $m_1, \ldots, m_n$ des entiers premiers entre eux deux à deux. Soit u un entier et $u_1, \ldots u_n$ donnés par $u = u_i \mod m_i$ pour $i = 1, \ldots, n$.

Pour $i, j \in [1, \ldots n]$, notons $B_{j,i}$ et $B_{i,j}$ des coefficients de Bézout de m_i avec m_j

$$B_{j,i} m_i + B_{i,j} m_j = 1 \tag{3}$$

Notons $p_i = \prod_{j \neq i} B_{i,j} m_j$ pour $i = 1, \ldots, n$, alors l'entier $l = u_1 p_1 + \cdots + u_n p_n$ est tel que $l = v \mod m_1 \cdots m_n$ ssi $v = u_i \mod m_i$ pour $i = 1, \ldots, n$.

Donc si on connait une borne pour u, et qu'on choisit $m_1, \ldots, m_n$ de telle sorte que leur produit soit supérieur à deux fois cette borne, alors on peut obtenir u à partir de $u_1, \ldots u_n$ qui ont pour avantage de ne pas dépasser la valeur $sup\{m_i \mid i = 1, \ldots n\}$.

2. Les bases

- Faites `?mod`, lisez la documentation, comment faire en sorte que les calculs soient fait de façon modulaire et pas en entier puis tronqués ensuite ?
- Les calculs modulaires nécessitent les **coefficients de Bezout** de nombres ou de polynômes, faites `?gcd`, trouvez comment obtenir les coefficients de Bezout pour deux polynômes, deux entiers.
- Exprimez un entier u avec $u_1, \ldots, u_n$ tels que $u = u_i \mod m_i$ où $i = 1, \ldots, n$ et où les m_i sont des entiers premiers entre eux deux à deux dont le produit $m_1 \cdots m_n$ est supérieur à deux fois une constante C donnée. Il faudra donc écrire une fonction `code_chinois` prenant u et C en argument et rendant $u_1, \ldots, u_n$ et $m_1, \ldots, m_n$.
- Connaissant $u_1, \ldots, u_n$ et $m_1, \ldots, m_n$ (premiers entre eux deux à deux), calculez l' entier u sachant que $|u| < C$ et $m_1 \cdots m_n > 2C$. Vous écrirez donc une fonction `decode_chinois`.
- Faites de même avec des polynômes.
- Vous disposez maintenant de deux nouvelles fonctions *MAPLE*permettant de coder et décoder un entier représenté modulairement.

3. Un exemple

Il s'agit ici de calculer le déterminant d'une matrice en utilisant des méthodes modulaires.

- Entrez une matrice 10x10 (faites un petit programme !) à coefficients numériques, calculez son déterminant.
- Programmez la méthode de Gauss, pour trianguláriser une matrice (on pourra supposer que les pivots intervenants sont non nuls).
- Calculez par récurrence un majorant du déterminant d'une matrice nxn en fonction de n et de la taille des coefficients de la matrice.
- Comme on l'a vu en programmant la méthode de Gauss, il est nécessaire de faire des divisions. Pour que les calculs modulaires se fassent dans un corps, il faut que le nombre modulo lequel on travaille soit premier. faites `?prime` pour savoir comment trouver et générer des nombres premiers.
- Reprogrammez la méthode de Gauss pour qu'elle soit modulaire, cette fois-ci on ne pourra plus supposer que les pivots intervenants sont non nuls (profitez en pour mettre à jour la fonction précédente !).
- Écrivez deux petites fonctions qui calculent le déterminant (normal et modulaire) en utilisant les formes triangulaires précédentes.
- Génerez des nombres premiers (3 ou 4) jusqu'à ce que leur produit dépasse une borne égale à deux fois celle du déterminant de votre matrice.
- Calculez les déterminants modulaires pour chacun de ces nombres.
- Il faut maintenant retrouver le résultat en se servant de ce qui précède, comme tout est défini modulo $\prod_{i=1}^{n} m_i$ si le déterminant calculé depasse la borne c'est qu'en fait il est négatif.

4. Interpolation

On va appliquer les méthodes modulaires sur les polynômes au lieu des entiers, le but est de calculer un polynôme de degré fixé à partir des valeurs qu'il prend sur un certain nombre de points.

Soit donc P un polynôme de degré 9 que l'on cherche ; nous savons qu'il prend les valeurs $p_1, \cdots p_{10}$ aux points $x_1, \cdots x_{10}$ que vous pouvez vous fixer arbitrairement.

- dire que $P(x_1) = p_1$, c'est dire en termes de polynômes que le polynôme $P(x)$ est congru à p_1 modulo $x - x_1$; de même pour les autres. Vérifiez ces relations en vous donnant P, et en l'évaluant en 10 points.
- Nous sommes donc dans les conditions du théorème chinois où nous connaissons une quantité modulo d'autres, nous savons que le degré du polynôme est plus petit que la somme des degrés des polynômes $x - x_i$. De la même manière que précédement les **polynômes de Lagrange** sont donnés par :

$$L_i(x) = \prod_{k \neq i} \frac{x - x_i}{x_k - x_i}.$$

- Donnez vous 10 points et calculez le polynôme associé.
- Ecrivez un petit programme *MAPLE* qui prenne une liste de n points et une liste de n valeurs et qui retourne le polynôme associé.

Maîtrise d'informatique Paris 6
Module Calcul Formel

Renaud RIOBOO
Annick VALIBOUZE

Représentation des données

Soit $p(x) = 5/2x^5+3xy^2z+2yz^2-y^2x^5+7zyx^2+8xyz$ un polynôme de $Q[x,y,z]$. Nous savons que les anneaux de polynômes suivant sont isomorphes :

$$Q[x,y,z], Q[x][y,z], Q[x,y][z], Q[x][y][z], \ldots$$

a) Nous désirons écrire mathématiquement p dans différents anneaux. Mais afin que chaque anneau ait une représentation canonique (en vue d'une écriture informatique) nous devons ordonner nos polynômes. Cet ordre étant arbitrairement choisi, nous en fixons un : Si A est un anneau, écrire un polynôme de $A[x_1, \ldots, x_n]$ signifiera que l'on donne un ordre $x_1 > x_2 > \cdots > x_n$ sur les variables. Chaque monôme commencera par x_1 puis x_2 ... puis x_n. Puis choisissons l'ordre lexicographique sur les exposants dans N^n pour ranger les monômes dans le polynôme.

A partir de cette convention d'ordre, écrire p dans

$$\begin{gathered} Q[x,y,z], \\ Q[x][y,z], Q[z][x,y], \\ Q[x,y][z], Q[z,y][x], \\ Q[x][y][z], Q[y][z][x], Q[z][y][x]. \end{gathered}$$

b) Compter le nombre d'anneaux isomorphes à $Q[x,y,z]$ d'abord sans considération d'ordre puis avec considération d'ordre (i.e. le nombre de représentations possibles). Puis généraliser à $Q[x_1, \ldots, x_n]$.

c) En déduire les représentations LISP correspondantes.

d) Donner un représentation distribuée de p dans

$$\begin{gathered} Q[x,y,z], \\ Q[x][y,z], Q[z][x,y]. \end{gathered}$$

e) Plaçons nous dans $Q[x,y]$.

- : Ecrire un programme de tri, `TRI_DIST`, qui change l'ordre des variables dans la représentation distibuée d'un polynôme de $Q[x,y]$. Si le tri est réalisé par dichotomie sur un polynôme de degré n il est en $nlogn$.
- : Ecrire un programme de tri , `DIS_RECU`, qui permette de passer de $Q[x][y]$ à $Q[x,y]$. Si ce tri est effectué directement il est en n^2.
- : Ecrire un programme, `RECU2DIST`, qui passe de la représentation récursive dans $Q[x][y]$ à la représentation distribuée dans $Q[y,x]$.
- : Ecrire un programme réciproque `DIST2RECU`.
- : Installer un compteur qui permette de comparer `DIS_RECU`, théoriquement en n^2, avec le même résultat obtenu par la composition de `RECU2DIST` avec `TRI_DIS` puis `DIST2RECU` qui est théoriquement en $nlogn$.

: Ecrire deux programmes permettant de passer de la représentation distribuée à la forme LISP dans $Q[x, y]$ et réciproquement.

: Ecrire deux programmes permettant de passer de la représentation récursive à la forme LISP dans $Q[x, y]$ et réciproquement.

Maîtrise d'informatique Paris 6
Module Calcul Formel

Renaud RIOBOO
Annick VALIBOUZE

Manipulations de Polynômes

Le but du TD est de réaliser les opérations simples sur les polynômes, en une ou plusieurs variables. Nous allons adopter une représentation distribuée creuse pour les polynômes: un **polynôme** sera une liste de **monômes** non nuls triés dans l'ordre lexicographique décroissant des exposants. La liste des variables est implicite (et n'est donc pas codée dans la représentation). Comment représente-t-on le polynôme nul ?

Un **monôme** sera représenté comme une liste à deux éléments où le premier terme est l'**exposant** et le deuxième est le **coefficient**.

Un **exposant** est une liste de longueur le nombre de variables contenant les degrés en chaque variable. Un **coefficient** est un objet *MAPLE* quelconque.

Ainsi nous avons la forme suivante pour un polynôme ayant m monômes non nuls sur n variables:

$$[\,[\,[\,\deg_{1,1}\,,\,\ldots\,,\,\deg_{n,1}\,]\,,\,\mathrm{coeff}_1\,],\ldots,\,[\,[\,\deg_{1,m}\,,\,\ldots\,,\,\deg_{n,m}\,]\,,\,\mathrm{coeff}_m\,]\,]$$

En particulier en une seule variable on aura:

$$[\,[\,[\,\deg_1\,]\,,\,\mathrm{coeff}_1\,],\ldots,\,[\,[\,\deg_m\,]\,,\,\mathrm{coeff}_m\,]\,]$$

Nous allons d'abord nous donner quelques utilitaires de programmation:

- Écrivez des procédures *MAPLE* `mainMon` et `redPol` qui retournent respectivement le monôme de tête et le polynôme formé de tous les autres monômes d'un polynôme (i.e privé du monôme de tête).
- Écrivez une fonction `consPol` à deux arguments (un monôme et un polynôme) et qui retourne le polynôme formé du monôme argument (en tête) et du polynôme argument. Faites de même une fonction `insereMon` qui mette le monôme argument "à sa place" dans le polynôme et non plus en tête.
- Écrivez des procédures *MAPLE* `exposant` et `terme` qui retournent respectivement la liste des degrés en les différentes variables, et le coefficient d'un monôme.
- Écrivez une fonction `lexGt` qui prenne deux monômes en arguments et qui retourne un **booléen** (`true` ou `false`) suivant que le premier monôme est ou non supérieur au second dans l'ordre lexicographique des exposants. Vous pourrez commencer par le cas simple d'une seule variable (c'est l'ordre usuel), puis de deux, puis généraliser.
- Écrivez une fonction `lexOrder` qui trie une liste de monômes pour retourner un polynôme et une fonction `makeCan` qui enlève d'une liste de monômes ceux dont le coefficient est nul.
- Écrivez une fonction `polToMaple` qui prenne en arguments un polynôme et une liste de variables (ayant le même nombre d'éléments que les listes des exposants des monômes) et qui retourne un objet *MAPLE* usuel.

Nous pouvons maintenant penser à implémenter les opérations arithmétiques, nous allons réaliser l'addition de 2 polynômes P_1 et P_2.

- Que retourner si l'un des 2 polynômes est nul ?
- Nous allons examiner les monômes de tête M_1 et M_2 de P_1 et de P_2. Que faut-il retourner si l'exposant de M_1 est plus grand (strictement pour l'ordre lexicographique), plus petit, égal à celui de M_2 ?
- Implémentez cet algorithme en *MAPLE* (attention au cas où les exposants sont égaux !).

Nous pouvons passer à la multiplication de P_1 et P_2.

- Comment multiplier un polynôme par un monôme? Implémentez une fonction *MAPLE* `monMul` qui multiplie un polynôme (passé en premier argument) par un monôme (passé en deuxième argument).
- Que reste-t-il à faire pour multiplier deux polynômes? Écrivez une fonction *MAPLE* qui multiplie deux polynômes.

Pourquoi n'a-t-on pas de notion de division euclidienne sur les polynômes en plusieurs variables ? Lorsqu'il n'y a qu'une seule variable nous pouvons écrire la division euclidienne en utilisant les opérations sur les coefficients.

- Comment calcule-t-on le premier terme du quotient de la division de P par Q ?
- Écrivez une fonction *MAPLE* `polDiv` à deux arguments (P et Q) qui calcule le quotient et le reste de la division euclidienne. Le résultat sera une liste de longueur 2 contenant le quotient et le reste.
- Écrivez deux fonctions *MAPLE* `polQuo` et `polRem` analogues à `polDiv` qui calculent directement le quotient et le reste.

Maintenant que nous avons la base, rajoutez quelques fonctions utilitaires ! Inspirez vous des fonctions déjà existantes en *MAPLE*.

Maîtrise d'informatique Paris 6
Module Calcul Formel

Renaud RIOBOO
Annick VALIBOUZE

Exercices d'application en Maple et comparaison avec Macsyma

La correction des exercices en Macsyma vous est fournie.

(1) *Dynamique de Populations*
C'est un exemple d'application en Electrochimie de P. Turq et M. Simonin.
On considère le système $(A(t), B(t), C(t))$ évoluant selon

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C \xrightarrow{k_3} A \qquad .$$

Ce système est régi par l'équation différentielle

$$\dot{V}(t) = MV(t) \tag{4}$$

où V est le vecteur colonne de composantes (A, B, C) et

$$M = \begin{pmatrix} -k_1 & 0 & k_3 \\ k_1 & -k_2 & 0 \\ 0 & k_2 & -k_3 \end{pmatrix} \qquad .$$

La solution initiale de (4) est $V(t) = e^{Mt}V_0$, où V caractérise l'état initial du système. De sorte que

$$V(t) = \sum_{i=1}^{3} e^{\lambda_i t} P_i V_0 \tag{5}$$

où les λ_i sont les valeurs propres de M et les P_i les projecteurs associés aux λ_i. L'équation (5) montre que les modes de relaxation du système sont donc les valeurs propres de M.

(a) *Détermination analytique des modes*
A l'aide de Macsyma ou maple, et forts des cours précédents et aussi en utilisant le manuel, procéder aux étapes suivantes :
- Entrer la matrice M (méthode de votre choix).
- Calculer son polynôme caractéristique (idem).
- Essayer de factoriser. Conclusion.
- Extraire le polynôme du second degré et calculer son discriminant.
- Considérer les deux cas limites où l'une des deux constantes est très petite (à la limite nulle : $k_3 = 0$), et où les trois constantes sont égales ($k_i = k$).
- Calculer explicitement les modes dans ce dernier cas. Extraire les parties réelles et imaginaires des solutions.

(b) *Calcul des fonctions* $A(t), B(t), C(t)$
On suppose que $A(t = 0) = A_0$, $B(t = 0) = C(t = 0) = 0$ et l'on se place dans le cas où les constantes k_i sont égales.

- Ecrire (sur le papier) la transformée de Laplace de (4). Introduire dans macsyma (plusieurs possiblités) le nouveau système. Le résoudre. Exprimer la T.L. de A.
- Décomposer cette dernière expression en éléments simples (contourner les problèmes éventuels) et calculer ainsi $A(t)$.

(2) *Résolution d'une équation différentielle* (P. Turq et M. Simonin) Trouver la solution générale de l'équation différentielle

$$y'' + y = \frac{1}{\text{sinus } (x)} \quad .$$

Quelle est la solution qui vérifie $y(\pi/3) = y'(\pi/3) = 1$?

(3) *Exercices d'algèbre linéaire*
Pour illustrer les manipulations sur les matrices réalisables sous un système de calcul formel, nous allons tout d'abord tester la diagonalisation de matrice avec le théorème suivant :

théorème 1. *Soient A une matrice carrée et d le pgcd des coefficients de la matrice adjointe de $A - xI$. Alors le polynôme minimal de A est égal au quotient de son polynôme caractéristique par d.*

On sait que la matrice est diagonalisable si ce polynôme minimal n'a que des racines simples.

(a) Ecrire le programme permettant de tester si une matrice est diagonalisable.
(b) Ecrire le programme qui diagonalise une matrice si c'est possible.
(c) Applications :

(6)
$$\begin{pmatrix} 7 & 12 & -12 \\ -2 & -3 & 4 \\ 2 & 4 & -3 \end{pmatrix}$$

(7)
$$\begin{pmatrix} -1 & -2 & 3 & 2 \\ 0 & 1 & 0 & 1 \\ -2 & -2 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

(4) *La matrice de Wilkinson*
Une matrice de Wilkinson d'ordre n peut être ainsi définie :

$$\begin{pmatrix} n & 1 & & & & & & \\ 1 & n-1 & 1 & & & & & \\ & 1 & n-2 & 1 & & & & \\ & & \ddots & \ddots & \ddots & & (0) & \\ & (0) & & 1 & 0 & 1 & & \\ & & & & \ddots & \ddots & \ddots & \\ & & & & & 1 & n-1 & 1 \\ & & & & & & 1 & n \end{pmatrix}$$

Ecrire le programme permettant d'obtenir une matrice de Wilkinson. Pour $n = 13$ donner une évaluation de ses valeurs propres. Que remarquez-vous ?

(5) *Polynôme de Wilkinson*
Créer (par une boucle simple) le polynôme

$$P(x) = (x-1)(x-2)\dots(x-20).$$

Modifier, dans ce polynôme, le coefficient du terme de degré 19 dans une proportion de 2^{-32} (changement du dernier bit seulement). Déterminer toutes les racines de ce nouveau polynôme.

(6) *Calcul d'une primitive*
Intégrer les fonctions suivantes :

$$f(x) = \frac{2x+1}{x^4+2x^3+3x^2+2x+2}$$

$$g(x) = \frac{4x^3+6x}{x^8+6x^6+15x^4+18x^2+10}$$