MI 101 Maple
Résolution d'équations
Feuille de TD numéro 7 - jeudi 22 novembre

Table des matières

1 Infos diverses

Examen intra confirmé le 29 novembre. Pensez à choisir votre horaire!!

L'examen portera sur les sujets des semaines 1 à 6, intégralement. Il durera 1 heure (ou 50 minutes...). Un ordinateur sera à votre disposition, mais le résultat de l'examen sera à rendre 'sur papier'. Vous pourrez disposez, comme pour chaque épreuve, de toutes les feuilles de TD des semaines 1 à 6 (à vous de les apporter) et de tout document manuscrit non photocopié. Et il comptera pour 30% de la note.

D'autre part, le plan de cours a encore été légèrement modifié (pour plus de cohérence, donc une meilleure qualité...); voir `mi101.fr.st`[1].

2 Résolution d'équations

On a déjà vu un certain type de résolution d'équations, en cherchant les racines d'un polynôme. En fait, la fonction `solve()` est polyvalente, et nous sera très pratique. La syntaxe-type est

`solve(`*`équation`*`,`*`variable`*`)`

où

- *`équation`* est l'équation à résoudre (ou le système d'équations); l'égalité `=0` est prise par défaut si aucune égalité n'est spécifiée.
- *`variable`* est la variable (ou les variables) à déterminer afin que l'équation soit vérifiée; si aucune variable n'est indiquée, Maple utilisera toutes les variables non afectées de l'équation.

Cette commande demande à Maple de déterminer la ou les valeurs de la variable pour laquelle (lesquelles) l'équation est vérifiée (ou les équations). En retour, Maple fournit la *séquence* des

[1] Il est possible que le site aie des problèmes d'accessibilité d'ici quelques semaines. N'hésitez pas à me prévenir par mail à `limare@altern.org`.

solutions, ou la solution si elle est unique, parfois simplifiée par des %1, %2,... qui sont des raccourcis résumant des termes apparaissant plusieurs fois, ou avec des `RootOf()` (racine de) dans le cas de résolutions longues et encombrantes, ou impossibles par calcul symbolique exact; si on note `S` le résultat d'une résolution faisant apparaitre des `RootOf()`, on peut 'insister' pour obtenir des résolutions explicites (si elles ont possibles) par l'instruction `allvalues(S);`. Et si il n'existe pas de méthode connue pour résoudre notre équation, on peut obtenir une valeur approchée, par la commande `fsolve()`, de syntaxe semblable.

Une équation peut être saisie explicitement (`solve(exp(t)=t+1,...)`), ou 'stockée' dans une variable (`E:=a*(b-2*a)/2; solve(E,...)`) (car une équation est encore un type d'objet pour Maple), ou bien exprimée à l'aide d'une fonction (`foo:=t->exp(-t)*sin(3*t); solve(foo(x)=0,x);`).

2.1 Equation simple

La résolution d'une équation simple ne pose pas de problème particulier. La principale difficulté consiste à comprendre et interpréter la réponse de Maple.

2.2 Système d'équations

On peut aussi chercher les solutions d'un système de plusieurs équations, c'est-à-dire les variables qui vérifient simultanément chacune de ces équations; c'est en fait, mathématiquement, comme si on résolvait une équation, avec une variable dans R^n. Pour cela, on regroupe les équations `E1`, `E2`,... et les variables `v1`, `v2`,... en un ensemble `{E1,E2,...}` et `{v1,v2,...}` (ici, la liste est inutile, car il n'y a pas de notion de position). Maple nous répond par des couples, des ensembles, ... On saisit donc, par exemple,

```
solve({E1,E2,E3},{v1,v2,v3});
```

Dans le cas des systèmes d'équations linéaires, il existe des commandes spécifiques, particulièrement efficaces, liées à la vision matricielle du sujet; on verra cela à la prochaine séance.

3 Interprétation graphique

On peut visualiser graphiquement les solutions d'une équation, avec la comande `implicitplot()`. Celle-ci doit être 'chargée' avant utilisation, car elle est dans la bibliothèque graphique de Maple, qu'on charge par `with(plots)` (ATTENTION : il est nécessaire de recharger cette bibliothèque après chaque `restart`). `implicitplot(E, x=a..b, y=c..d)` dessinera alors dans le plan l'ensemble des points du plan vérifiant `E`, si `E` est une équation à deux inconnues, et `implicitplot3d()` fera de même pour les équations à 3 inconnues.

Une solution d'un système d'équations doit vérifier chacune des équations, et réciproquement si une valeur de la variable est solution de chacune des équations, alors elle est solution du système. Ainsi, si $S1$ est l'ensemble des solution de l'équation $E1$, $S2$ pour $E2$,..., alors l'ensemble des solutions du système $\{E1, E2, \ldots\}$ est $S1 \cap S2 \cap \ldots$. Si `E1`, `E2` sont les équations à résoudre, leurs ensembles de solutions sont visibles par `implicitplot(E1, v1=a..b, v2=c..d)` (idem pour `E2`), et on peut observer les solutions du système, donc l'intersection des solutions ce chaque équation, en affichant simultanément les deux ensembles de solutions, par `implicitplot({E1,E2}, v1=a..b, v2=c..d)`.

L'interprétation géométrique des équations linéaires (plans et droites) sera détaillée ultérieurement.

4 Exercices

4.1 Equation simple

Exo 1
Résolvez l'équation $2ax^4 = b$*, sur la variable* x*, pour* a *et* b *indéterminées, puis en fixant des valeurs précises à* a *et* b*.*

Exo 2
Résolvez $\cos(x) = x$*.*

Exo 3
Résolvez $\frac{x^2}{a} + \frac{y^2}{b} = 1$*, pour* a *et* b *de votre choix, et affichez graphiquement l'ensemble des solutions. Que reconnaissez-vous?*

Exo 4
Construisez la liste des solutions pprochées de $\ln(y) = 1/n$*, pour* n *entre 1 et 100.*

Exo 5
Résolvez TOTALEMENT *l'équation* $ax^4 + bx^3 + cx^2 + dx + e = 0$*, avec* a*,* b*,* c*,* d *et* e *indéterminées, et donnez-en des valeurs approchées à 15 décimales pour* $a = b = c = d = e = 1$*.*

Exo 6
Définissez une procédure qui prend en arguments s *et* t *et qui renvoie l'ensemble des solutions de* $\cos(sx) = cos(tx)$*, puis testez-la avec* $t = s$*,* $t = 2s$*,* $t = s + \pi$*.*

Exo 7
Exprimez avec `solve()` *les solutions d'un polynôme de degré 2, de coefficients indéterminés. Puis, toujours avec* `solve()`*, trouvez les valeurs que doit prendre le coeficient de degré 2, en fonction des autres coefficients, pour que le discriminant soit nul. Combien y a t'il de valeurs du coefficient de degré 2, les autres étant fixés, telles que le polynôme aie une seule solution?*

4.2 Système d'équations

Exo 8
Résolvez le système $3x - 5y = 2, -x + 2y = 4$ *formellement, puis graphiquement.*

Exo 9
Résolvez le système $4x - 2y + z = 2, 2x + 2y - 5z = 1, -x - y - 3z = 0$ *formellement, puis graphiquement.*

Exo 10
Résolvez le système $x^2 + y^2 + z^2 = 1, x = |y|$ *formellement, puis graphiquement.*

Exo 11
Résolvez le système $xy = z, yz = x, zx = y$ *formellement, puis graphiquement.*

5 Corrections

sous réserve de vérification...

5.1 Equation simple

Exo 1

```
S:= solve(2*a*x**4-b,x);
a:=1: b:=3: S;
```

Exo 2

```
solve(cos(x)=x);
```

Exo 3

```
a:=3: b:=5: E:=x**2/a+y**2/b=1: solve(E,x,y);
with(plots): implicitplot(E,x=-2..2,y=-3..3);
```

Il s'agit d'une ellipse.

Exo 4

```
L:=[seq(fsolve(ln(y)=1/n),n=1..100)];
```

Exo 5

```
P:=a*x**4+b*x**3+c*x**2+d*x+e=0: solve(E,x);
S:=allvalues(%);
a:=1: b:=1: c:=1: d:=1: e:=1: evalf(S,15);
```

On peut aussi faire un

```
Digits:=15: fsolve(E,x);
```

Exo 6

```
sol:=proc(s,t)
S:=solve(cos(s*x)=cos(t*x),x);
S;
end;
```

puis

```
sol(s,s); sol(s,2*s); sol(s,s+Pi);
```

Exo 7

```
solve(a*x**2+b*x+c,x);
solve(b**2-4*a*c=0,a);
```

5.2 Système d'équations

Exo 8

```
solve(3*x-5*y=2,-x+2*y=4,x,y);
implicitplot(3*x-5*y=2,-x+2*y=4,x=-10..10,y=-10..10);
```

Exo 9

```
solve(4*x-2*y+z=2,2*x+2*y-5*z=1,-x-y-3*z=0,x,y,z);
implicitplot3d(4*x-2*y+z=2,2*x+2*y-5*z=1,-x-y-3*z=0,x=-10..10,y=-10..10,z=-10..10);
```

Exo 10

```
solve(x**2+y**2+z**2=1,x=abs(y),x,y,z);
implicitplot3d(x**2+y**2+z**2=1,x=abs(y),x=-2..2,y=-2..2,z=-2..2);
```

Exo 11

```
solve(x*y=z,y*z=x,z*x=y,x,y,z);
implicitplot3d(x*y=z,y*z=x,z*x=y,x=-2..3,y=-2..3,z=-2..3);
```