TP Maple 5 | Dérivées, études locales et EDO

Le logiciel va nous permettre, au-delà des outils graphiques, de calculer des limites, des développements limités et des développements asymptotiques, d'effectuer des calculs de dérivées et de résoudre des équations différentielles (de manière exacte ou approchée).

1. Dérivées totales et partielles

Comme toujours, on distinguera les fonctions et les expressions. Cela se traduira par deux syntaxes distinctes.

1.1. Expressions : les opérateurs diff et Diff

Afin de présenter avec élégance des séquences de calcul différentiel, Maple dispose d'un opérateur *effectif* de différenciation et d'un opérateur *inerte* de différenciation. L'opérateur *effectif* **diff** effectuera et affichera directement les calculs demandés alors que sa version *inerte* **Diff** effectuera les calculs sans les afficher. Il faudra forcer l'affichage du résultat par la commande **value**.

Dérivations formelles successives d'une expression E

Opérateur effectif : **diff(**E**,**L**)** où L est la liste des variables par rapport auxquelles on dérive l'expression E.
Opérateur inerte : **Diff(**E**,**L**)** où L est la liste des variables par rapport auxquelles on dérive l'expression E.

Dérivées successives d'une fonction à une variable

```
> E:=x^3*sin(x): diff(E,x), Diff(E,x);
```

$$3x^2\sin(x)+x^3\cos(x),\ \frac{d}{dx}(x^3\sin(x))$$

```
> value(Diff(E,x));
```

$$3x^2\sin(x)+x^3\cos(x)$$

On pourra donc écrire :

```
> Diff(E,x)=diff(E,x);
```

$$\frac{d}{dx}(x^3\sin(x))=3x^2\sin(x)+x^3\cos(x)$$

Bien-sûr, les dérivées successives sont également calculables, on notera les différentes syntaxes :

```
> diff(E,x,x); diff(E,x,x,x);
```

$$6x\sin(x)+6x^2\cos(x)-x^3\sin(x),\ 6\sin(x)+18x\cos(x)-9x^2\sin(x)-x^3\cos(x)$$

```
> diff(E,x$3);
```

$$6\sin(x)+18x\cos(x)-9x^2\sin(x)-x^3\cos(x)$$

```
> diff(E,[]), diff(E,[x]), diff(E,[x,x]);
```

$$x^3\sin(x),\ 3x^2\sin(x)+x^3\cos(x),\ 6x\sin(x)+6x^2\cos(x)-x^3\sin(x)$$

Seule la dernière syntaxe (ie **diff(E,[...])**) autorise la dérivation à l'ordre zéro.

Dérivées partielles succesives

```
> F:=x/(x^2+y^2): Diff(F,x,y)=diff(F,x,y);
```

$$\frac{\partial^2}{\partial y \partial x}\left(\frac{x}{x^2+y^2}\right) = -\frac{2y}{(x^2+y^2)^2} + 8\frac{x^2 y}{(x^2+y^2)^3}$$

```
> Diff(F,y$2,x)=diff(F,y$2,x);
```

$$\frac{\partial^3}{\partial x \partial^2 y}\left(\frac{x}{x^2+y^2}\right) = 8\frac{y^2}{(x^2+y^2)^3} - 48\frac{x^2 y^2}{(x^2+y^2)^4} - \frac{2}{(x^2+y^2)^2} + 8\frac{x^2}{(x^2+y^2)^3}$$

1.2. Fonctions : l'opérateur D

Dans le cas d'une fonction f, c'est l'opérateur **D** qui permettra de calculer les dérivées successives de f.

Dérivations formelles successives d'une fonction f

- **D(f)** : renvoie la fonction dérivée de f.
- **D@@n(f)** : renvoie la fonction dérivée n-ième de f pour $n \in \mathbb{N}^*$.
- **D**$[i_i, i_2, \dots, i_n](f)$ retourne la fonction $\dfrac{\partial^n f}{\partial x_{i_n} \dots \partial x_{i_2} \partial x_{i_1}}$

```
> f:=x->exp(x)*sin(x): D(f);
```

$$x \to e^x \sin(x) + e^x \cos(x)$$

```
> (D@@2)(f);
```

$$x \to 2e^x \cos(x)$$

```
> g:=(x,y,z)->z*cos(x*y): D[1,1,1](g), D[2](g), D[1,2](g);
```

$$(x,y,z) \to z\sin(xy)y^3,\quad (x,y,z) \to -z\sin(xy)x,\quad (x,y,z) \to -z\cos(xy)yx - z\sin(xy)$$

```
> D[1,1,1](g)(Pi,1/2,1);
```

$$\frac{1}{8}$$

```
> D[](g);
```

$$g$$

Signalons que l'on peut aussi dériver une fonction f en appliquant l'opérateur **diff** à l'expression[1] $f(x)$.

2. Etude locale des fonctions numériques

2.1. Limites

Le logiciel dispose d'un opérateur inerte (**Limite**) et effectif (**limit**) de calcul des limites. Pour calculer

$$\lim_{x \to a} f(x)$$

où $f(x)$ est une expression, on suivra la syntaxe décrite ci-dessous.

Calcul d'une limite par la commande limit

limite(f(x), x=a, *right* ou *left*)

le troisième argument (pour une limite à droite ou à gauche) étant optionel.

1. Ceci n'est bien entendu valable que si la variable x est libre.

```
> Limit(sin(x)/x,x=0): %=value(%);
```

$$\lim_{x\to 0}\frac{\sin(x)}{x} = 1.$$

On écrira $a = \pm$infinity pour calculer des limites en $\pm\infty$.

```
> Limit(exp(x)/x,x=infinity): %=value(%);
```

$$\lim_{x\to+\infty}\frac{e^x}{x} = \infty$$

Le logiciel renvoie le message *undefined* lorsqu'il manque de données pour calculer ce qui lui est demandé. C'est par exemple le cas lorsque la limite n'existe pas car les limites à gauche et à droite sont distinctes. L'utilisateur pourra spécifier *right* ou *left* en option dans la commande **limit** pour calculer respectivement des limites à droite et à gauche.

```
> Limit(exp(1/x),x=0): %=value(%);
```

$$\lim_{x\to 0} e^{\frac{1}{x}} = \mathit{undefined}$$

```
> Limit(exp(1/x),x=0, right): %=value(%);
```

$$\lim_{x\to 0+} e^{\frac{1}{x}} = \infty$$

2.2. Développements de Taylor

Maple calcule des développements de Taylor dont les restes sont des O (« *grand o* ») ; par exemple

$$\sin(x) = x + O(x^3)$$

Calcul d'un développement limité de $f(x)$ en un point a par la commande **taylor**

taylor($f(x)$**,** $x = a$**,** n**)**

Il faut bien comprendre la signification de n ci-dessus : il ne s'agit pas de l'ordre du développement calculé mais l'ordre auquel le logiciel développera les fonctions composant l'expression $f(x)$. Le lecteur méditera les exemples suivants :

```
> taylor(sin(x),x=0,4);
```

$$x - \frac{1}{6}x^3 + O(x^4)$$

```
> taylor(sin(x)/ln(1+x),x=0,4);
```

$$1 + \frac{1}{2}x - \frac{1}{4}x^2 + O(x^3)$$

2.3. Développements asymptotiques

Plus généralement, l'utilisateur obtiendra des développements asymptotiques[2] au moyen de la commande **series**.

Calcul d'un développement asymptotique de $f(x)$ en un point a par la commande **series**

series($f(x)$**,** $x = a$**,** n**)**

La remarque précédente sur la signification de n est encore valable pour la commande **series**. Ce troisième argument est cependant optionnel. Maple dispose d'une variable d'environnement **Order** qui par défaut vaut 6.

```
> series(1/sin(x),x=0,4);
```

$$x^{-1} + \frac{1}{6}x + O(x^2)$$

```
> Order:=2:series(sin(x),x=0);
```

$$x + O(x^2)$$

La commande **taylor** n'est qu'une spécialisation de la commande **series**. On pourra donc également obtenir des *DL* au moyen de **series**.

```
> series(sin(x),x=0,4);
```

$$x - \frac{1}{6}x^3 + O(x^4)$$

On pourra aussi effectuer des développements en $+\infty$...

2. Ce type de développement généralise les *DL*. Le lecteur est renvoyé à son cours de Mathématiques.

```
> series(1/(1+x),x=infinity,4);
```

$$\frac{1}{x}-\frac{1}{x^2}+\frac{1}{x^3}+O\left(\frac{1}{x^4}\right)$$

2.4. Extraction de la partie principale

Le résultat renvoyé par la commande **series** n'est pas utilisable tel quel pour un tracé ou pour effectuer des calculs algébriques (produits ou sommes de DL) car il n'est pas du type *polynom* (au sens large). On pourra cependant *extraire* la partie principale d'un développment en utilisant la commande **convert**.

Extraction de la partie principale d'un développement

convert (*résultat d'une commande series* , polynom)

```
> series(60*ln(1+x), x = 0); convert(%,polynom);
```

$$60x-30x^2+20x^3-15x^4+12x^5+O(x^6)$$
$$60x-30x^2+20x^3-15x^4+12x^5$$

2.5. Obtention d'un équivalent

On obtiendra un équivalent d'une expression au voisinage d'un point en utilisant la command **leadterm**.

Détermination d'un équivalent de x en a par **leadterm**

series(leadterm($f(x)$), $x=a$)

```
> series(leadterm(sqrt(1+sqrt(x))),x=infinity);
```

$$\frac{1}{\left(\frac{1}{x}\right)^{\frac{1}{4}}}$$

2.6. Exercices

Exercice 1.

Pour tout entier naturel n, on pose

$$\forall x \in \mathbb{R}, \quad f_n(x) = \sin(x)\cos^n(x).$$

1. Représenter sur un même graphique les fonctions f_n pour n variant de 0 à 15 sur $[0, 2\pi]$.

2. Caculer en utilisant le logiciel le maximum u_n de f_n sur $[0, 2\pi]$.

3. Déterminer la limite de u_n quand n tend vers $+\infty$ puis un équivalent de u_n.

Exercice 2.

Trouver la limite puis un équivalent du terme général de la suite définie par

$$\forall n \geqslant 2, \quad u_n = \cos(n^2 \pi \,\ell\text{n}(1 - 1/n)).$$

Exercice 3.

Pour tout entier naturel n non nul, on pose

$$u_n = (1 - \tanh(n))^{\tanh(1/n)}.$$

1. Déterminer la limite ℓ de u_n lorsque n tend vers $+\infty$ en utilisant le logiciel. Démontrer ce résultat.

2. Trouver un équivalent de $u_n - \ell$ en utilisant Maple pour les calculs intermédiaires.

3. Equations différentielles

Le logiciel peut résoudre explicitement de nombreuses équations différentielles. En cas d'impossibilité, l'utilisateur pourra se rabattre sur des solutions de rechange comme une résolution approchée ou la recherche de solutions développables en série.

3.1. La commande dsolve

La commande **dsolve** renvoie les solutions d'une équation différentielle (écrite au moyen de la commande **diff**) en écrivant les constantes sous la forme $\_C1$, $\_C2$, etc.

```
> equ:=diff(y(x),x)+y(x)=exp(-x): dsolve(equ,y(x));
```

$$y(x) = (x + \_C1)e^{-x}$$

On pourra aussi préciser des conditions initiales :

```
> dsolve({equ,y(0)=0},y(x));
```

$$y(x) = e^{-x}x$$

```
> eq:=diff(y(x),x$2)+diff(y(x),x)+y(x)=0: dsolve({eq,y(0)=0,D(y)(0)=1},y(x));
```

$$y(x) = \frac{2\sqrt{3}}{3}e^{-\frac{x}{2}}\sin\left(\frac{\sqrt{3}x}{2}\right)$$

L'utilisateur prendra garde à ce que les séquences précédents n'ont nullement affecté les variables y et $y(x)$: la solution n'a été retenue ni sous forme d'une fonction ni sous forme d'une expression.

```
> y(0);
```

$$y(0)$$

S'il souhaite conserver la solution comme une fonction y, l'utilisateur suivra la syntaxe suivante :

Résolution formelle d'une équation différentielle

- *Solution générale (avec constante)* : **dsolve(*équation*)**
- *Solution à un problème de Cauchy* : **dsolve({*équation, cond. init.*})**
- *Enregistrer la solution comme une fonction* :

  sol :=dsolve({*equation, cond. init.*},y(x)) : assign(sol) : y :=unapply(y(x),x) :
- *Enregistrer la solution comme une expression* :

  sol :=dsolve({*equation, cond. init.*},y(x)) : y :=rhs(sol) :

```
> eq:=diff(y(x),x)-y(x)=exp(x): sol:=dsolve({eq,y(0)=1},y(x)): assign(sol):
  y:=unapply(y(x),x): y(0);
```

$$1$$

Signalons que Maple ne peut pas résoudre toutes les équations !...

```
> equ:=diff(y(x),x$2)+x*diff(y(x),x)+(x^2+1)*y(x)=0;
```

$$\frac{d^2}{dx^2}y(x)+x\left(\frac{d}{dx}y(x)\right)+(x^2+1)\,y(x)=0$$

```
> sol:=dsolve({equ,y(0)=1,D(y)(0)=1},y(x));
```

$$\mathrm{sol}:=$$

Le logiciel reste muet ; il faudra alors envisager d'autres moyens...

3.2. Système d'équations différentielles

La syntaxe précédente se généralise sans peine à un système d'équations différentielles de la manière suivante :

Résolution formelle d'un système d'équations différentielles

dsolve({*équation*$_1$, *équation*$_2$, ..., *équation*$_n$, *conditions initiales*})

On pourra également extraire les solutions en tant que fonctions. Voici l'exemple d'un système linéaire homogène d'ordre deux.

```
> eqq:={diff(x(t),t)+x(t)+2*y(t)=0,diff(y(t),t)+x(t)-y(t)=0,x(0)=0,y(0)=1}:
  sol:=dsolve(eqq);
```

$$\mathrm{sol}:=\left\{x(t)=-\frac{1}{3}\sqrt{(3)}e^{\sqrt{(3)}\,t}+\frac{}{3})\sqrt{3}e^{-\sqrt{3}t},\ \ y(t)=\frac{1}{2}e^{\sqrt{3}t}+\frac{1}{2}e^{-\sqrt{(3)}\,t}+\frac{1}{6}\sqrt{(3)}e^{\sqrt{(3)}\,t}-\frac{1}{6}\sqrt{(3)}e^{-\sqrt{(3)}\,t}\right\}$$

```
> x:=unapply(subs(sol,x(t)),t);
```

$$x:=t\to-\frac{1}{3}\sqrt{(3)}e^{\sqrt{(3)}\,t}+\frac{}{3})\sqrt{3}e^{-\sqrt{3}t}$$

3.3. Développement limité d'une solution

Le logiciel peut trouver un *DL* de la solution à une équation différentielle donnée au voisinage d'une condition initiale fixée. L'utilisateur explicitera l'option *series* en troisième argument de la commande **dsolve**.

```
> equ:=diff(y(x),x$2)+x*diff(y(x),x)+(x^2+1)*y(x)=0:
  sol:=dsolve({equ,y(0)=1,D(y)(0)=1},y(x),series);
```

$$\mathrm{sol} := y(x) = 1 + x - \frac{1}{2}x^2 - \frac{1}{3}x^3 + \frac{1}{24}x^4 + \frac{1}{60}x^5 + O(x^6)$$

L'ordre du développement [3] est par défaut égal à 6. On pourra le changer en modifiant la valeur de la variable globale **Order** avant d'utiliser la commande **dsolve**.

```
> Order:=8: sol:=dsolve({equ,y(0)=1,D(y)(0)=1},y(x),series);
```

$$\mathrm{sol} := y(x) = 1 + x - \frac{1}{2}x^2 - \frac{1}{3}x^3 + \frac{1}{24}x^4 + \frac{1}{60}x^5 + \frac{7}{720}x^6 + \frac{1}{180}x^7 + O(x^8)$$

3.4. Résolution numérique

Lorsque le logiciel s'avérera incapable d'exprimer formellement une solution, on pourra tout de même obtenir une solution approchée obtenue par une méthode numérique choisie par Maple [4]. Il faudra alors préciser l'option *numeric* en troisième argument de la commande **dsolve**.

```
> equ:=diff(y(x),x$2)+x*diff(y(x),x)+(x^2+1)*y(x)=0;
  sol:=dsolve({equ,y(0)=1,D(y)(0)=1},y(x),numeric);
```

$$\mathrm{sol} := \mathbf{proc}(xrkf45)...\mathbf{end\ proc}$$

La variable *sol* se comporte comme une fonction de la variable x et donc la valeur en x est une liste de la forme

$$[x,\ \textit{valeur approchée de } y(x),\ \textit{valeur approchée de } y'(x),\ \ldots,\ \textit{valeur approchée de } y^{(n-1)}(x)]$$

dans le cas d'une équation d'ordre n. Ainsi, dans l'exemple exposé ci-dessus :

3. Il s'agit d'un *DL* en $\mathcal{O}$, attention au décallage par rapport au cours de Mathématiques !
4. Citons par exemple la méthode de *Runge-Kutta-Fehlberg*, *rkf* en abrégé.

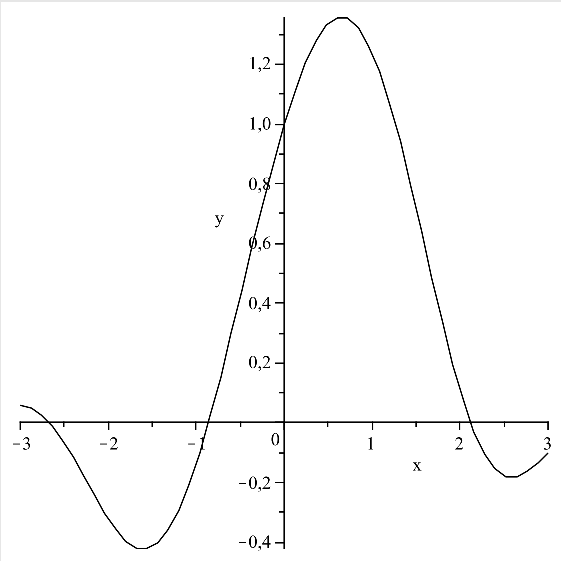
```
> sol(0);
```

$$\left[x = 0.,\ \ y(x) = 1.,\ \ \frac{d}{dx}y(x) = 1.\right]$$

Représentation graphique

La fonction **odeplot** de la bibliothèque **plots** permettra au lecteur de représenter graphiquement la solution approchée calculée par Maple.

```
> with(\verb|plots|): odeplot(sol,[x,y(x),color=black],-3..3);
```



On a précisé $[x, f(x), \ldots]$ en deuxième argument pour forcer le logiciel à tracer $y(x)$ en fonction de x.

Etraction de la solution approchée sous forme d'une fonction

Pour cette opération, on commence par forcer le logiciel à créer n procédures à partir de la procédure définissant *sol*. Il suffit d'indiquer l'otions *output=listprocedure* en quatrième argument de **dsolve**. On peut alors extraire la fonction $y(x)$ par une commande **subs**.

```
> lst:=dsolve({equ,y(0)=1,D(y)(0)=1},y(x),numeric,output=listprocedure);
```

$$\mathrm{lst} := \left[x = \textbf{proc}(x)\ldots\textbf{end proc},\ \ y(x) = \textbf{proc}(x)\ldots\textbf{end proc},\ \ \frac{d}{dx}y(x) = \textbf{proc}(x)\ldots\textbf{end proc}\right]$$

```
> f:=subs(lst,y(x));
```

$$f := \textbf{proc}(x)\ldots\textbf{end proc}$$

La fonction ainsi définie est utilisable par exemple pour des tracés, des équations, etc.

```
> fsolve(f(x)=0,x=2..3);
```

$$2.120658201$$

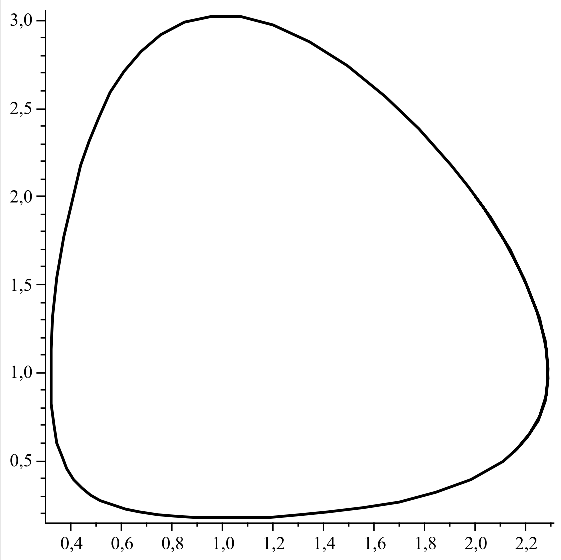
Systèmes d'équations différentielles

La commande **dsolve** permet également la résolution approchée des systèmes d'équations différentielles. La syntaxe s'inspire largement de ce qui précède et de ce qui a été déjà vu lors de l'étude de la résolution formelle des systèmes d'équations.

Voici l'exemple du sytème *proie-prédateurs*[5]. On observe classiquement une *oscillation* du système : ce dernier est stable.

5. les fonctions x et y représentent respectivement le nombre de prédateurs et de proies en fonction du temps. On a modélisé leur coexistence en fonction du temps t par le sytème différentiel $\dot{x}(t) = 0.5x(t)(y(t)-1),\quad \dot{y}(t) = y(t)(1-x(t))$.

```
> restart:equ:={diff(x(t),t)=0.5*x(t)*(y(t)-1),diff(y(t),t)=y(t)*(1-x(t)),
  x(0)=2,y(0)=2}:
  sol:=dsolve(equ,numeric,output=listprocedure):
  x:=subs(sol,x(t)):y:=subs(sol,y(t)):
  plot([x,y,0..11],color=black,thickness=2);
```



3.5. Exercices

Exercice 4.

Soit

$$(\mathbf{E})\ :\ x^2y'' + 4xy' + 2y = \ell n(1+x).$$

1. Résoudre $(\mathbf{E})$ sur $]0,+\infty[$ en utilisant Maple.

2. Déterminer l'unique solution de $(\mathbf{E})$ bornée au voisinage 0. On la notera f.

3. Tracer le graphe de f sur $[0,+\infty[$.

Exercice 5.

Soit

$$(\mathbf{E})\ :\ y'' + \frac{y}{x^2} = 0.$$

Résoudre $(\mathbf{E})$ au moyen du logiciel sur $[1,+\infty[$. Déduire de la forme des solutions une méthode de résolution de $(\mathbf{E})$.

Exercice 6.

Pour I intervalle de $\mathbb{R}$ et f continue sur I à valeurs dans $\mathbb{R}$, on considère l'équation différentielle

$$(\mathbf{E_f})\ :\ y' - y + f(x) = 0.$$

1. Désoudre $(\mathbf{E_f})$ au moyen de Maple dans le cas où $I = \mathbb{R}$ et $f(x) = e^{-2x}\cos^3(x)$.

2. Dans cette question, $I = \mathbb{R}_+^*$ et $f(x) = \frac{1}{x}$.

2.a. Que donne Maple pour la résolution de cette équation ?

2.b. Représenter sur un même graphique les solutions des problèmes de Cauchy suivants :

$$\begin{cases} y' - y + \frac{1}{x} = 0 \\ y(1) = \alpha \end{cases}, \qquad \alpha \in \{-15, -14, \dots, 14, 15\}.$$

2.c. Rechercher au moyen du logiciel les solutions de $(\mathbf{E_f})$ bornées au voisinage de $+\infty$.

2.d. Déterminer avec Maple un équivalent simple en 0+ des solutions de $(\mathbf{E_f}$.

2.e. Déterminer sans le logiciel l'ensemble $\mathscr{H}$ des points (a, b) du demi-plan d'inéquation $x > 0$ par lesquels passe le graphe d'une solution $x \mapsto y(x)$ de $(\mathbf{E_f})$ telle que $y'(a) = 0$.

2.f. Déterminer sans le logiciel l'ensemble $\mathscr{L}$ des points (a, c) du demi-plan d'inéquation $x > 0$ par lesquels passe le graphe d'une solution $x \mapsto y(x)$ de $(\mathbf{E_f})$ telle que $y''(a) = 0$.

2.g. Représenter simultanément $\mathscr{H}, \mathscr{L}$, les solutions de $(\mathbf{E_f})$ trouvées à la question **2.b.** ainsi que l'unique solution de $(\mathbf{E_f})$ bornée au voisinage de $+\infty$.