Dans ce TD on va revenir sur l'étude des fonctions et des équations différentielles. La première partie a pour but de vous familiariser avec quelques fonctions Maple de base pour étudier rapidement une fonction. Cette partie ne devrait pas vous poser de problème, essayez d'aller vite. La seconde partie revient sur la résolution d'équations différentielles et sur la méthode d'Euler, que nous comparerons à une méthode plus avancée.

1 Étude de fonctions : quelques outils

Tout plein de choses simples, mais parfois utiles. À faire vite normalement, utilisez l'aide ! !

Exercice 1

1. *Taper la ligne suivante, et la comprendre !*
   `[ > f := x->piecewise(x<1,-1,x<2,x,2); plot(f,0..3);`
2. *Définir la fonction valeur absolue avec* `piecewise`*, et la tracer sur l'interval* $[-\frac{2\pi}{5}, \frac{2\pi}{5}]$.
3. *Définir le prolongement par continuité sur* $\mathbb{R}$ *de la fonction* $x \mapsto \frac{\sin x}{x}$.
4. *Vous souvenez-vous de ce qui s'affiche pour l'instruction suivante ?*
   `[> limit(arctan(x),x=infinity);`

Consulter l'aide à propos des opérateurs D et diff. Quelle est la différence ?

Exercice 2 *Calculer la dérivée* q *de la fonction* $r : x \mapsto \sqrt{1+\sqrt{2+\sqrt{x}}}$. *Tracer* q *et étudier* $\lim_{+\infty} q$.

Exercice 3 *Soit* f *la fonction définie sur* $\mathbb{R}_*^+$ *par* $f : x \mapsto \frac{\sqrt{x}}{1+x^2}$. *Calculer la dérivée de* f *en utilisant l'opérateur de dérivation D. Résoudre* $f'(x) > 0$, *en déduire le tableau de variation de* f. *Donner le maximum de* f *sur* $\mathbb{R}_*^+$.

Exercice 4 *Résoudre l'équation différentielle* $y' + 2y = t \cos t$ *(*`dsolve`*).*

2 Résolution approchée d'équations différentielles

Nous revenons sur la résolution approchée d'équations différentielles. On se place dans le cas d'équations différentielles d'ordre 1 sous forme résolue $y'(t) = f(t, y(t))$, avec la condition initiale $y(t_0) = y_0$. Vous avez déjà vu la méthode d'Euler. Nous y revenons pour en montrer les limites, puis nous introduisons la méthode RK4, plus précise. Voici les 3 équations différentielles que nous utiliserons tout au long du TP.

(T_1) $y'(t) = \frac{\cos(t)-y(t)}{t+1}, y(0) = -\frac{1}{4}$
(T_2) $y'(t) = 2ty(t), y(0) = 1$
(T_3) $y'(t) = 3\frac{y(t)}{t} - \frac{5}{t^3}, y(1) = 1$

2.1 Rappels sur Euler

Nous avons besoin de reprogrammer Euler pour les comparaisons entre méthodes. N'hésitez pas à réutiliser vos anciens TPs ! !

Le principe est d'approcher la solution $y(t)$ sur $[a, b]$ par une fonction affine par morceaux, en opérant une discrétisation du paramètre t. On pose $t_n = a + nh$ (on a donc $N = \frac{b-a}{h}$ abscisses t_n), et y_n est l'approximation de y en t_n. La fonction affine par morceaux joindra donc les points de coordonnées (t_n, y_n), et il s'agit de proposer un algorithme pour construire les y_n enconnaissant y_0. Sur chaque intervalle $[t_n, t_{n+1}]$ on prend pour pente du segment affine celle que suggère l'équation différentielle, soit $f(t_n, y_n)$. On en déduit y_{n+1} à partir de y_n, par la formule $y(t_n + h) = y(t_n) + h \times f(t_n, y(t_n))$, soit aussi

$$y_{n+1} = y_n + h \times f(t_n, y_n) \qquad \text{(E1)}$$

Exercice 5 *Écrire une procédure* `procEuler` *prenant en entrée : le point initial* $(a, y(a))$, *l'extrémité* b *de l'intervalle de définition, le pas* h *de la méthode et la fonction* f *de l'équation différentielle, et retournant : la liste des points* $(t_0, y_0), \dots (t_N, y_N)$.

Maintenant vous allez tester votre procédure et comparer son résultat au résultat exact.

Exercice 6 *Pour chacun des problèmes suivants, tracer sur le même graphe la solution exacte (obtenue par* `dsolve`*) et la solution retournée par* `procEuler`*.*

1. *Sur l'exemple* T_1, *paramètres :* $a = 0, b = 10, h = 0.1$. *Prenez maintenant* $h = 0.25$. *Commentez.*
2. *Sur l'exemple* T_2, *paramètres :* $a = 0, b = 2, h = 0.1$. *Quelle est la qualité de l'approximation ? Pouvez vous expliquer (qualitativement) pourquoi on n'obtient pas la même précision que précédemment ?*
3. *Sur l'exemple* T_3, *paramètres :* $a = 0, b = 5, h = 0.1$. *Commentez. Passez à* $h = 0.01$. *Vous attendiez-vous à ce résultat ? Pourriez-vous l'expliquer (indications : utiliser* `DEplot` *du package* `DEtools`*, ou trouver la solution générale) ?*

2.2 Méthodes de Runge-Kutta

Nous allons maintenant présenter une méthode plus précise que celle d'Euler. Les méthodes de Runge-Kutta sont une famille de méthodes toutes basées sur la même idée. Euler en est un cas particulier. On présente ici RK4, la plus connue des méthodes de Runge-Kutta, très utilisée en pratique car d'un excellent rapport qualité-prix.

L'idée de base reprend Euler : on part d'un point initial (t_0, y_0), puis on calcule de proche en proche des points $(t_0 + nh, y_n)$ où h est un pas fixé. Les y_n obéissent à une récurrence du type $y_{n+1} = y_n + h \times g(x_n, y_n, h)$, avec g une fonction à définir. Par exemple, pour Euler, $g(x_n, y_n, h) = f(x_n, y_n)$.

Si on revient au développement de Taylor-Young, et en utilisant $y'(t) = f(t, y(t))$, on arrive à

$$y(t_n + h) = y(t_n) + \int_{t_n}^{t_n+h} f(u, y(u))du$$

Le fond du problème revient donc à approximer l'intégrale. Dans Euler, on utilise une approximation "par rectangle" : f est considérée constante sur $[t_n, t_{n+h}]$, valant $f(t_n, y(t_n))$. D'où $y(t_n+h) = y(t_n)+h\times f(t_n, y(t_n))$.

Pour les méthodes de Runge-Kutta, on aura une forme générale $y(t_n+h) = y(t_n)+h\times(\sum_1^m a_k f(t_{n,k}, y_{n,k})$, où les a_k sont des coefficients dont la somme vaut 1, et les $t_{n,k}$ une subdivision de $[t_n, t_{n+1}]$. Si les a_k et les $t_{n,k}$ sont bien choisis[1], on obtient une bonne approximation. Pour le cas particulier de RK4, on prend la formule suivante :

$$y_{n+1} = y_n + h \times \frac{f(p_1) + 2f(p_2) + 2f(p_3) + f(p_4)}{6} \qquad \text{(E2)}$$

avec

$p_1 = (t_n, y_n)$
$p_2 = (t_n + \frac{h}{2}, y_n + \frac{h}{2}f(p_1))$
$p_3 = (t_n + \frac{h}{2}, y_n + \frac{h}{2}f(p_2))$
$p_4 = (t_n + h, y_n + hf(p_3))$

L'idée est que y_{n+1} est approximé par la somme de la valeur actuelle y_n et du produit de la taille de l'intervalle h par la pente estimée. La pente est obtenue par une moyenne pondérée des pentes suivantes :

[1] C'est là que c'est difficile ! !

- $f(p_1)$ est la pente au début de l'intervalle ;
- $f(p_2)$ est la pente au milieu de l'intervalle, en utilisant la pente k_1 pour calculer la valeur de y au point $t_n + \frac{h}{2}$ par le biais de la méthode d'Euler ;
- $f(p_3)$ est de nouveau la pente au milieu de l'intervalle, mais obtenue cette fois en utilisant la pente k_2 pour calculer y ;
- $f(p_4)$ est la pente à la fin de l'intervalle, avec la valeur de y calculée en utilisant k_3.

RK4 est une méthode d'ordre 4, ce qui signifie que l'erreur commise à chaque étape est de l'ordre de h^5, alors que l'erreur totale accumulée est de l'ordre de h^4. Euler est d'ordre 2.

Exercice 7

1. *Écrire une procédure `nextValue`, prenant en entrée t_n, y_n, h, f et retournant y_{n+1} par la formule (E2).*
2. *Écrire une procédure `procRK4`, ayant mêmes entrées-sorties que `procEuler`.*
3. *Reprendre les exemples T_2 et T_3 avec $h = 0.1$. Que constatez-vous ?*
4. *Prenez T_3 (mêmes paramètres). Tracez la courbe d'erreur entre la solution idéale et celle retournée. Quelle est l'erreur maximale ?*

2.3 RK4 avec adapation du pas

Nous allons voir dans cette section comment améliorer grandement la précision de RK4 pour un coût raisonnable. On peut penser à diminuer le pas pour diminuer l'erreur. Cependant cela s'avère souvent trop coûteux en pratique : comme le pas est uniforme, on calcule autant de points dans les zones "faciles" de la fonction (peu de variations) que dans les zones "dangereuses" (variations rapides). Nous ne considérons donc plus un pas fixe h, et nous autorisons le pas à varier pour s'adapter à la fonction à approximer.

Nous prenons ici l'exemple du passage d'un point (t_n, y_n) à un point (t_{n+1}, y_{n+1}), avec un pas de h. L'erreur E_h à chaque de calcul est en $\beta \cdot h^5$. En fait on peut montrer que

$$E_h \approx \frac{16(y_{n+1} - y''_{n+1})}{15} \qquad \text{(E4)}$$

avec

$$\begin{aligned} y_{n+1} &= \text{nextValue}(t_n, y_n, h) \qquad (y_{n+1} \text{ est la valeur qu'aurait calculée RK4}) \\ y'_{n+1} &= \text{nextValue}(t_n, y_n, h/2) \\ y''_{n+1} &= \text{nextValue}(t_n + h/2, y'_{n+1}, h/2) \end{aligned}$$

On va s'imposer une erreur maximale en un pas E_m, cette erreur corrrespondant à un pas idéal h_m. On a alors

$$\left|\frac{E_h}{h^5}\right| = \left|\frac{E_m}{h_m^5}\right| = |\beta|$$

d'où

$$|h_m| = |h| \left|\frac{E_m}{E_h}\right|^{1/5} \qquad \text{(E5)}$$

On modifie alors l'algorithme RK4 comme suit : pour passer d'un point (t_n, y_n) à un point (t_{n+1}, y_{n+1}), on commence par calculer $|E_h|$ grâce à (E4)[2]. Deux cas se présentent :

1. $|E_h| > |E_m|$, dans ce cas on reprend le calcul avec h_m donné par (E5) ; on pourra diminuer encore un peu h_m pour se donner de la marge ;
2. $|E_h| \leq |E_m|$, dans ce cas le calcul est validé, et on peut augmenter un peu h pour se rapprocher de h_m.

[2]Attention aux erreurs d'arrondis : si il est très bas, $|E_h|$ peut être approximé par 0.

La procédure `procRK4bis` avec adaptation du pas prend maintenant comme quatrième argument l'erreur maximale en un pas E_m, et le pas est une variable locale pouvant être modifée. Pour le premier pas de calcul, on prendra $h = 0.01 \times |b - a|$ (convention).

Exercice 8 *Écrivez la procédure `procRK4bis`. Prenez (T_3) avec $b = 10$. Tracez graphiquement `procRK4bis` ($E_m = 10^{-6}$) point par point (voir les options de `plot`) pour voir comment le pas est adapté. Combien de points sont calculés ? Quelle est l'erreur totale obtenue ? Comparez avec l'erreur maximale permise en un pas. Tracez graphiquement les erreurs obtenues avec `procRK4` ($h = 0.1$ puis $h = 0.01$) et `procRK4bis`. Comparez la précision et le nombre de points calculés.*