Exposé 85 : Exemples d'approximation d'une solution d'une équation différentielle par la méthode d'Euler. L'exposé pourra être illustré par un ou des exemples faisant appel à l'utilisation d'une calculatrice.

Prérequis[1] :
-Continuité, dérivabilité d'une fonction
-Etude des suites
-Equation différentielle linéaire du premier ordre
-Calcul intégral (intérêt : présenter les deux autres méthodes d'Euler[2])

N : terminal - programme complémentaire

De nombreux problèmes, issus de la physique (radioactivité, réseaux éléctriques), de l'écologie et de l'économie conduisent à étudier une relation entre une fonction et sa dérivée. Le but de cette leçon est d'étudier des exemples simples de telles relations et de décrire une méthode (élémentaire) d'approche de ces problèmes.

1 Problématique

Soit I un intervalle de $\mathbb{R}$, $t_0 \in I$, f une fonction définie et continue sur $I \times \mathbb{R} \to \mathbb{R}$
On considère l'équation différentielle $(E) : y' = f(t, y)$
On pourra prendre $f(t, y) = a(t)y + b(t)$, $a, b \in C^0$ résolvable en terminale : (E) a une unique solution : $y(t) = e^{A(t)}.y_0 + e^{A(t)} \int_{t_0}^{t} e^{-A(s)} b(s)ds, A(t) = \int_{t}^{t} a(s)ds$ Etant donné un réel y_0, on cherche à construire une solution approchée de (E) sur I tq. $y(t_0) = y_0$
Par la suite, on va supposer que (E) admet une solution unique (que l'on ne cherche pas à déterminer, mais juste à approcher). Approcher la solution de l'équation différentielle se ramène à un calcul d'intégrales :
$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} y'(t)dt = y(t_n) + \int_{t_n}^{t_{n+1}} f(t, y(t))dt$. C'est cette dernière intégrale que l'on va chercher à approcher.

2 Méthode d'Euler (explicite)

Soit $(E) : y' = f(t, y) = a(t)y + b(t)$, $t \in [t_0, t_n] = I$ L'idée de la méthode est simple : on découpe l'intervalle d'étude I en n intervalles isométriques, et on approche, sur chacun de ces sous-intervalles, la courbe solution par sa tangente à l'extrémité gauche de l'intervalle : $t_0 < t_1 ... < t_n = T$, pas $h = \dfrac{t_n - t_0}{n}$
$t_i = t_0 + ih$

[1] L'exposé a été présenté et tapé à Bordeaux(1) le 27/04/05 par Gwendal Haudebourg, corrigé par M.B. Mis à jour le 31/07/2007.
[2] Léonard Euler : mathématicien suisse (1701-1783)

On s'efforce donc de suivre la courbe solution à l'aide des tangentes : on définit ainsi une suite $(y_k)_k$ par : $y_{k+1} = y_k + hf(t_k, y_k)$, $k = 0...n-1$.
On a ainsi une suite de points $A_i(t_i, y_i)$ qui forment une ligne brisée approchant la solution de l'équation différentielle (E). Cette ligne brisée n'étant pas évidente à tracer, on met ce point de vue de côté, pour s'occuper surtout de l'approximation des intégrales par la méthode des rectangles. La ligne brisée qu'une illustration graphique de la méthode d'Euler.

preuve :
En effet, soit y solution de E. Alors : $y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} y'(t)dt = y(t_n) + \int_{t_n}^{t_{n+1}} f(t, y(t))dt$ or $\int_{t_n}^{t_{n+1}} f(t, y(t))dt = (t_{n+1} - tn)f(t_n, y(t_n)) + (t_{n+1} - t_n).\xi$ (par la méthode des rectangles gauches). On approche donc la solution y en posant comme suite de points : $y_{n+1} = y_n + (t_{n+1} - t_n)f(t_n, y_n)$.
L'erreur à l'issue du pas $n+1$ est $y_{n+1} - y(t_{n+1})$

Le shéma d'Euler explicite est une méthode constructive

Exemple :
Soit (E) : $y' = -y$ et $y_0 = 1$ (on prend donc $t_0 = 0$)
On sait bien sûr que la solution de (E) est $y(t) = e^{-t}$ et $\lim_{t\to\infty} e^{-t} = 0$
Par Euler, on contruit $y_n = y_{n-1} + hf(t_{n-1}, y_{n-1}) = y_{n-1} - h.y_{n-1} = y_{n-1}(1-h) = ...$ d'où $y_n = (1-h)^{n+1} = (1 - \frac{T}{n})^{n+1}$ qui converge bien vers la solution.
Mais lorsque le temp est très grand, y_n devient très grand : **pour** n **fixé**[3], $\lim_{T\to\infty}(1 - \frac{T}{n})^n \nrightarrow 0$ c'est à dire ne s'approche plus de la solution exacte.
Le shéma d'Euler explicite est donc un shéma instable, car on ne peut l'utiliser que sur des ensembles où le temps est fini.

Existe t-il une méthode proche de celle utilisée, qui soit stable pour des temps très grand ?

3 Méthode d'Euler (implicite)

On utilise le même principe, mais cette fois-ci, on va approcher $\int_t^t f(t, y(t))dt$ par la méthode des rectangles droits.
$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} y'(t)dt = y(t_n) + \int_{t_n}^{t_{n+1}} f(t, y(t))dt$ or
$\int_{t_n}^{t_{n+1}} f(t, y(t))dt = (t_{n+1} - tn)f(t_{n+1}, y(t_{n+1})) + (t_{n+1} - t_n).\xi$. On approche donc la solution y en posant comme suite de points : $y_{n+1} = y_n + (t_{n+1} - t_n)f(t_{n+1}, y_{n+1})$.

On a alors : $y_{n+1} = y_n + (t_{n+1} - t_n)(a(t_{n+1})y_{n+1} + b(t_{n+1}))$ donc $y_{n+1} = (1 - (t_{n+1} - t_n) * a(t_{n+1}))^{-1}.(y_n + (t_{n+1} - t_n)b(t_{n+1})$
Dans notre exemple, cela donne :
$y_{n+1} = (1 + \frac{T}{n})^{-1}.y_n = (1 + \frac{T}{n})^{-n}.y_0$ $(a = -1, b = 0)$, et $(y_n)_n$ converge bien vers la solution.
De plus, **pour n fixé**, $\lim_{T\to\infty} y_n = \lim_{T\to\infty}(1 + \frac{T}{n})^{-n} = 0$
La méthode d'Euler implicite reste fiable lorsque le temps est plus long.

On a gagné par cette méthode de la stabilité, mais le coût de l'algorithme est plus élevé, car il s'agit d'une méthode implicite : il y a des termes en y_{n+1} "de chaque côté de la suite", la méthode **n'est donc pas constructive**.

[3]Il faut bien voir que l'on peut faire varier n et T indépendament : on peut fixer n et faire varier T (le pas h variera)

4 Conclusion-Compléments

4.1 Conclusion

D'autre méthodes d'approximations des solutions d'une équations différentielles existent ; citons la méthode d'Euler amélioré, et Runge-Kutta.

4.1.1 Exemples à mettre dans l'exposé en conclusion

1) Un bon exemple pour comprendre l'utilisation des deux méthodes, et pour les comparer :
Soit (E) : $y' = -y + 1$, $y(0) = 0$, $t \in [0, 1]$. Solution exacte : $y(t) = 1 - e^{-t}$
On prend un pas $h = 0,1$ et $n = 10$ Euler explicite :$y_{n+1} = y_n.0,9 + 0,1$
Euler implicite : $y_{n+1} = y_n + 0,1.(1 - y_{n+1})$ soit $y_{n+1} = \dfrac{y_n + 0,1}{1,1}$
On rentre les deux suites dans la calculatrice (plus la solution exacte) pour les comparer. Sur des temps longs, Euler explicite s'éloigne de la solution.
Calculatrice : $u1(n) = u1(n-1).0,9 + 0,1$ et $ui1 = 0,1$
$u2(n) = \dfrac{u2(n-1) + 0,1}{1,1}$ et $ui2 = 0$
2) Exemple M.B. : $y'(t) = -1000y(t) + 100.t^2 + \dfrac{1}{5}$. Solution exacte : $y(t) = \dfrac{1}{10}.t^2 + \dfrac{1}{5}.t + \dfrac{1000}{3}$, $y(0) = \dfrac{1}{3}$.
Lorsque le temps devient très grand, Euler explicite oscille et s'éloigne complètement de la solution exacte.

4.1.2 Euler amélioré

Euler amélioré utilise la méthode du point milieu :

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(t, y(t))dt = y(t_n) + f(t_n + \frac{h}{2}, y(t_n + \frac{h}{2})) * (t_{n+1} - t_n) + truc$$

On approche $y(t_n + \frac{h}{2})$ par $y_n + \frac{h}{2}y'(t_n)$, soit par $y_n + \frac{h}{2} f(t_n, y(t_n))$.
On pose donc :
$y_{n+1} = y_n + hf(t_n + \frac{h}{2}, y_n + \frac{h}{2} f(t_n, y_n))$ (Euler amélioré explicite-ordre 1)
ou $y_{n+1} = y_n + hf(t_n + \frac{h}{2}, y_{n+1} - \frac{h}{2} f(t_{n+1}, y_{n+1}))$ (Euler amélioré implicite-ordre 2-stable)

exemple :
Soit $f(t, y) = ay$. Par Euler amélioré :
$y_{n+1} = y_n + h(a(y_n + \frac{h}{2}ay_n)) = y_n(1 + ah + a^2\frac{h^2}{2})$ d'où $y_n = (1 + ah + \frac{a^2.h^2}{2})^n$

4.2 Compléments

Les méthodes d'Euler sont des méthodes à un pas (ie y_{n+1} est calculée à partir de y_n et t_n).
Les shéma d'Euler sont stables, consistants, et d'ordre 1 pour euler explicite amélioré et implicite) ; ordre 2 pour euler amélioré implicite.

4.2.1 Méthode consistante

Le shéma général d'une méthode à un pas est, à partir d'une subdivision $x_{k+1} = x_k + h$ de l'intervalle : $y_{n+1} = y_n + h\Theta(x_n, y_n, h)$

La méthode sera dit consistante si la fonction Θ approche convenablement le véritable accroissement de la fonction cherchée, c'est à dire : pour tout y solution de (ED),

$$\lim_{h \to 0} max_n |\frac{y(x_{n+1}) - y(x_n)}{h} - \Theta(x_n, y(x_n), h)| = 0$$

4.2.2 Méthode stable

La méthode sera dite stable si le cumul des erreurs d'arrondis ne fait pas exploser l'écart entre le shéma théorique et le shéma réel

4.2.3 Ordre d'une méthode

On dira qu'une méthode est d'ordre p si $|\frac{y(x+h) - y(x)}{h} - \Theta(x, y(x), h)| = O(h^p)$ (grand Tô)

4.2.4 Théorèmes

Théorème : si la méthode est stable et consistante, elle est convergente, c'est à dire :
$\lim_{h \to 0}(max_n |y_n - y(x_n)|) = 0$ quelle que soit la valeur initiale y_0.
Critère 1 : si $\Theta(x, y, 0) = f(x, y)$, alors la méthode est consistante.
Critère 2 : si Θ vérifie une condition de Lipschitz en y, uniforme par rapport à x et à h, alors la méthode est stable.

Théorème (P)
Soient f continue en x et en t, sur $I \times \mathbb{R}$, où I intervalle fermé ; on suppose de plus qu'il existe t_0 et B une boule fermée dans $\mathbb{R}^n$ de rayon R, $x_0 \in B$
Alors $\exists$ solution X définie sur $\left[t_0 - \frac{R}{M}, t_0 + \frac{R}{M}\right]$ où $M = sup_{I \times B} |f(t, x)|$.
existence locale d'une solution

Théorème (C -L)
Soient $f : [a, b] \times \mathbb{R} \to \mathbb{R}$, continue par rapport à $(x, y) \in [a, b] \times \mathbb{R}$ et lipschitzienne par rapport à y (ie $\exists L > 0$ indépendante de x, telle que $|f(x, y_1) - f(x, y_2)| \leq L|y_1 - y_2|$ $, forall x, y_1, y_2$).
Alors le problème de Cauchy (E) :$y' = f(x, y(x))$, $y(a) = \alpha$, admet une solution $y(x)$ unique sur $[a, b]$.
existence et unicité d'une solution maximale

4.2.5 Calcul d'erreur-Ordre dans la méthode d'Euler

Si on se place au niveau secondaire :
Soit (E) : $y' = ay, y(0) = 1$. Par Euler explicite : $y_{n+1} = (1 + ah)^{n+1}$. Solution exacte : $y(t) = e^{at}$.
erreur absolue e pour t fixé, $h = \frac{t}{n}$, $e = |y(t) - y_n| = |(1 + \frac{at}{n})^n - e^{at}| = |e^{n.ln(1+\frac{at}{n})} - e^{at}| = |e^{n.\frac{at}{n} + O(\frac{1}{n^2})} - e^{at}|$
$= |e^{at}.(e^{n.O(\frac{1}{n^2})} - 1)| = e^{at}.|e^{O(\frac{1}{n})} - 1| = e^{at}.|1 + O(\frac{1}{n}) - 1| = e^{at}.O(\frac{1}{n}) = O(\frac{1}{n})$
erreur relative $\mathcal{E} = |\frac{y(t) - y_n}{y_n}| = O(\frac{1}{n})$ au moins d'ordre 1.
En travaillant plus précisément (par des équivalences par exemple), on montre qu'elle est exactement d'ordre 1.
Pour Euler implicite : idem (même calcul, même ordre).
Tout ce qui a été utilisé dans cette méthode est faisable niveau terminale : les DL utilisés peuvent être des petits exercices demandées auparavant.

Si on se place dans le programme complémentaire (utilisation des DL) :
On a : $\frac{y(x+h) - y(x)}{h} = y'(x + Oh) = y'(x) + \frac{h}{2} y''(x + \varepsilon h)$ et $\Theta(x, y, h) = f(x, y) = y'(x)$, donc l'ordre est 1. L'erreur est en $O(h)$.

M´ ’E 1

Programme pour la TI en langue anglaise : résolution graphique d'une équation différentielle par la méthode d'Euler (pris dans Initiation Voyage 200, P.7). Les "@**...**" sont des commentaires. Pour le mode français : remplacer Newlist par NouvList, et Zoomdata par ZoomDonn. Le programme qui suit est la méthode d'euler explicite, pour la fonction $f(x) = x + 1$, c'est à dire pour l'équation différentielle $y' = y + 1$, avec un pas de 0.1, et pour 40 points.

euler()
Prgm
Local f,x0,y0,p,n,lx,ly,i
"x+1"→ f :"0"→ x0 :"0"→ y0 :"0.1"→ p
"40"→ n
@ **la ligne précédente sert à mettre des valeurs par défaut pour le programme**
Dialog
Title "Entrée des données"
Request "f(x)",f
Request "x0 ",x0
Request "y0 ",y0
Request "Pas ",p
Request "Pts ",n
Endlog
expr(f)→ f : expr(x0)→ x0 : expr(y0)→ y0
expr(p)→ p : expr(n)→ n
Newlist(n)→ lx : Newlist(n)→ ly
For i,1,n
x0 → lx[i] :y0 → ly[i]
y0+p*(f| x=x0)→ y0 :x0+p→ x0
EndFor
lx → liste1 :ly → liste2
Newplot 1,2,liste1,liste2,,,,3
Zoomdata @ **pour zoomer sur les données. Pas nécessaire, et même parfois troublant**
EndPrgm

On trouve dans les manuels de TI une comparaison entre Euler et Runge-Kutta : P.11-19 pour le manuel de la TI 89, p.686 pour le manuel TI 89-Voyage 200.

M´ ’E 2

L'exemple qui suit est la méthode d'euler explicite, pour la fonction $f(t) = a(t)y + b(t)$. On peut tester par exemple $a(t) = -t$ et $b(t) = 1$, ie équation différentielle $y' = -y + 1$, avec un pas de 0.1, et pour 40 points.

euler2()
Prgm
Local a,b,x0,y0,p,n,lx,ly,i
Dialog
Title "Entrée des données"
Request "a(x)",a
Request "b(x)",b
Request "x0 ",x0
Request "y0 ",y0

```
Request ”Pas ”,p
Request ”Pts ”,n
EndDlog
expr(a)→ a : expr(b)→ b : expr(x0)→ x0 : expr(y0)→ y0
expr(p)→ p : expr(n)→ n
Newlist(n)→ lx : Newlist(n)→ ly
For i,1,n
x0 → lx[i] :y0 → ly[i]
(y0+p*(b| x=x0))/(1-p*(a|x=x0)→ y0 x0+p→ x0
EndFor
lx → liste3 :ly → liste4
Newplot 2,2,liste3,liste4,,,,3
```
@ **qques petits chgts pour ne pas écraser le travail réalisé avec Euler 1**
```
Zoomdata
EndPrgm
```

remarque : s'il y a un souci avec un programme, vérifier que la TI est dans la bonne langue, et que les liste liste1 et liste2 (ou liste3, etc.) sont bien créés (on le vérifie manuellement dans Apps/Stats List Editor) ; le cas échéant créer ces listes.