

ANALYSE NUMÉRIQUE

Corrigés des Travaux Pratiques 2024 – 2025

Séance 10

Les codes Octave des méthodes d'Euler progressive (fonction `eulerp`), Euler rétrograde (fonction `eulerr`) et Heun (fonction `heun`) sont donnés en annexe. Les trois fonctions ont la forme suivante

```
y = nom(f,y0,t);
```

où f est une fonction de deux variables t et y , y_0 est la condition initiale et t le vecteur d'abscisses où on veut calculer la solution approchée y . Notons que la fonction pour la méthode d'Euler rétrograde (la seule méthode implicite parmi les trois) utilise l'instruction `fsolve`.

- 1.a) La fonction $f(t, y) = y \cos(t) + \sin(y)$ est continue (pourquoi?) et lipschitzienne en y car

$$\left| \frac{\partial f}{\partial y} \right| = |\cos(t) + \cos(y)| \leq 2.$$

Donc, par Théorème 1 du Chapitre 8, il existe une et une seule solution.

- 1.b) Une manière d'obtenir la solution avec les méthodes d'Euler progressive et rétrograde est comme suit :

```
clear; close;
f = @(t,y) y.*cos(t) + sin(y);
y0 = 1;
h = 0.01;    t = 0:h:10;
yp = eulerp(f, y0, t);
yr = eulerr(f, y0, t);
% visualisation graphique
plot(t, yp, 'b');
hold on; plot(t, yr, 'r'); hold off
```

Notons que le choix de h peut se baser sur le fait que les solutions approchées (celles d'Euler progressive et Euler rétrograde) sont suffisamment proches. Bien entendu cela ne donne aucune garantie sur l'erreur qu'on commet.

- 1.c) On peut comparer la solution pour h et $2h$ avec une solution plus précise (qu'on considérera comme proche de la solution exacte). Exemple pour Euler progressive :

```
clear; close;
f = @(t,y) y.*cos(t) + sin(y);
y0 = 1;
h = 0.1;
th = 0:h:10;
yh = eulerp(f, y0, th);
h2 = h/2;
t2h = 0:h2:10;
y2h = eulerp(f, y0, t2h);
h16 = h/16;
t16h = 0:h16:10;
y16h = eulerp(f, y0, t16h);
```

```

erh = y16h(1:16:end)-yh;
er2h = y16h(1:8:end)-y2h;
% visualisation graphique
plot(th, erh, 'b');
hold on; plot(t2h, er2h, 'r');
plot(t2h, zeros(length(t2h),1), 'k'); hold off
% estimation numérique: quotient des normes sup
norm(erh, 'inf')/norm(er2h, 'inf')

```

Rappelons que la norme sup d'un vecteur $\mathbf{v} = (v_i)$, calculée via `norm(v, 'inf')` en Octave, vaut $\max_k |v_k|$. Comme les vecteurs de l'erreur ont comme composantes (approximativement) $y_k - y(t_k)$, leur norme sup est $\max_k |y_k - y(t_k)|$, ce qui permet d'avoir la même mesure de l'erreur que celle utilisée dans le Théorème 2 du Chapitre 8.

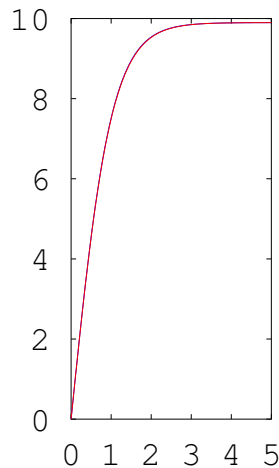
2. Une résolution possible est

```

clear; close;
f = @(t,y) 9.8 - 0.1.*abs(y).*y;
h = 0.01;
t = 0:h:5;
y0 = 0;
yp = eulerp(f, y0, t);
yh = heun(f, y0, t);
plot(t, yp, 'b');
hold on; plot(t, yh, 'r'); hold off

```

et donne une solution approchée comme suit



Notons qu'en absence de la résistance de l'air la solution est une droite dont la pente est 9.8. L'effet de la résistance de l'air est opposé à celui de la pesanteur, les deux termes étant asymptotiquement égaux en valeur absolue. On en conclue que $v(t) \leq v_\infty = \sqrt{g/\gamma}$.

A titre de digression, notons que la fonction $f(t, v) = g - \gamma v(t)|v(t)|$ est continue et, comme la valeur de la vitesse reste bornée par v_∞ , on a¹

$$|f(t, v_1) - f(t, v_2)| \leq 2\gamma v_\infty |v_1 - v_2|.$$

1. L'inégalité s'obtient, par exemple, en considérant séparément le cas où v_1, v_2 sont de même signe et de signe opposé.

La fonction est donc lipschitzienne (même si elle n'est pas dérivable en 0), et la solution est unique.

3. L'étude de stabilité se fait sur base du problème de Cauchy linéaire

$$\begin{cases} \frac{dy}{dt}(t) = -\beta y(t), & t > 0, \\ y(0) = y_0. \end{cases}$$

Dans ce cas, la méthode de Crank-Nicolson avec le pas d'intégration h constant devient

$$y_{k+1} = y_k - \frac{1}{2}h\beta(y_k + y_{k+1}),$$

ce qui donne

$$y_{k+1} = y_k \frac{1 - h\beta/2}{1 + h\beta/2}.$$

Comme $h\beta > 0$ (pourquoi?), la valeur absolue du facteur $(1 - h\beta/2)/(1 + h\beta/2)$ est inférieure à 1 et donc

$$y_k \rightarrow 0 \quad \text{pour} \quad k \rightarrow \infty.$$

On en conclut que la méthode de Crank-Nicolson est absolument stable pour tout h .

NOTE : De manière facultative, on peut étendre l'étude de stabilité aux systèmes d'équations différentielles en considérant le cas où β est complexe. La stabilité est alors assurée pour les valeurs de $\text{Re}(\beta) > 0$ ($\equiv$ condition de stabilité du problème continu) et $h > 0$ telles que

$$\left| \frac{1 - h\beta/2}{1 + h\beta/2} \right| = \left(\frac{(1 - h\text{Re}(\beta)/2)^2 + (h\text{Im}(\beta)/2)^2}{(1 + h\text{Re}(\beta)/2)^2 + (h\text{Im}(\beta)/2)^2} \right)^{1/2} < 1.$$

Or, comme $h\text{Re}(\beta) > 0$ implique que $(1 - h\text{Re}(\beta)/2)^2 < (1 + h\text{Re}(\beta)/2)^2$, l'inégalité est satisfaite d'office. Par conséquent, la méthode de Crank-Nicolson est absolument stable pour tout h même dans le cas où β est complexe.

Annexe

La code des fonctions `eulerp`, `eulerr` et `heun` :

```
function y = eulerp(f,y0,t)
% Y = eulerp(F,Y0,T)
%   Calcule la solution approchée Y du problème de Cauchy
%
%       dY/dt = F(t,Y)
%       Y(0) = Y0
%
%   aux points T en utilisant la méthode d'Euler progressive.
%
% arguments:
%   F - la fonction de deux variables Y et t;
%   t - un scalaire
%   Y - la solution en t
%   Y0 - condition initiale au point T(1)
%   T - le vecteur des points où la solution du problème de
%       Cauchy doit être approchée
```

```

% sortie:
%   Y - solution approchée du problème de Cauchy
y(1) = y0;
for i = 1:length(t)-1
    h = t(i+1)-t(i);
    y(i+1) = y(i) + f(t(i),y(i))*h;
end

function y = eulerr(f,y0,t)
% Y = eulerp(F,Y0,T)
%   Calcule la solution approchée Y du problème de Cauchy
%
%        $dY/dt = f(t,Y)$ 
%        $Y(0) = Y0$ 
%
%   aux points T en utilisant la méthode d'Euler rétrograde.
%
% arguments:
%   F - la fonction de deux variables Y et t;
%   t - un scalaire
%   Y - la solution en t
%   Y0 - condition initiale au point T(1)
%   T - le vecteur des points où la solution du problème de
%       Cauchy doit être approchée
% sortie:
%   Y - solution approchée du problème de Cauchy
y(1) = y0;
for i = 1:length(t)-1
    h = t(i+1)-t(i);
    g = @(x) y(i) + f(t(i+1),x) * h - x;
    y(i+1) = fsolve(g,y(i));
end

function y = heun(f,y0,t)
% Y = eulerp(F,Y0,T)
%   Calcule la solution approchée Y du problème de Cauchy
%
%        $dY/dt = f(t,Y)$ 
%        $Y(0) = Y0$ 
%
%   aux points T en utilisant la méthode de Heun.
%
% arguments:
%   F - la fonction de deux variables Y et t;
%   t - un scalaire
%   Y - la solution en t
%   Y0 - condition initiale au point T(1)
%   T - le vecteur des points où la solution du problème de
%       Cauchy doit être approchée

```

```

% sortie:
%      Y - solution approchée du problème de Cauchy
y(1) = y0;
for i = 1:length(t)-1
    h = t(i+1)-t(i);
    fi = f(t(i),y(i));
    y(i+1) = y(i) + (fi + f(t(i+1),y(i)+h*fi)) * h/2;
end

```