Analyse Numérique
Corrigés des Travaux Pratiques 2024 – 2025
Séance 11

1.a) Le problème de Cauchy peut s'écrire sous la forme vectorielle

$$\begin{cases} \frac{d\mathbf{y}}{dt} = -A\mathbf{y}\,, \\ \mathbf{y}(0) = \mathbf{y}_0\,, \end{cases}$$

avec

$$\mathbf{y} = \begin{pmatrix} N_{\mathrm{Te}} \\ N_{\mathrm{I}} \\ N_{\mathrm{Xe}} \end{pmatrix}, \quad A = \begin{pmatrix} \lambda_{\mathrm{Te}} & 0 & 0 \\ -\lambda_{\mathrm{Te}} & \lambda_{\mathrm{I}} & 0 \\ 0 & -\lambda_{\mathrm{I}} & \lambda_{\mathrm{Xe}} \end{pmatrix} \quad \text{et} \quad \mathbf{y}_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

b) Comme la matrice est triangulaire, ses valeurs propres se trouvent sur la diagonale (en effet, $A - \beta I$ est aussi triangulaire et $\det(A - I\beta) = \Pi_{i=1}^{3}(a_{ii} - \beta) = 0$ pour $\beta = a_{ii}$). Pour la matrice en question les valeurs propres sont donc λ_{Te}, λ_{I} et λ_{Xe}. Comme toutes les valeurs propres sont strictement positives, la solution exacte du problème est bornée, comme l'indique le résultat de la page 22 des transparents du Chapitre 8. Le même résultat indique que la méthode d'Euler progressive est stable si $|1 - \lambda_X h| < 1$ pour tout X, ce qui dans notre cas revient à

$$h < \frac{2}{\lambda_{\mathrm{Te}}},$$

la valeur propre λ_{Te} étant la plus grande et donc la plus restrictive. La méthode d'Euler rétrograde est quant à elle inconditionnellement stable.

c) Pour commencer nous encodons les données du problème :

```
clear; close;
lambda_1 = log(2)/19;
lambda_2 = log(2)/(6.7*3600);
lambda_3 = log(2)/(9.2*3600);
f = @(t,y) [ -lambda_1*y(1); ...
             lambda_1*y(1)-lambda_2*y(2); ...
             lambda_2*y(2)-lambda_3*y(3)];
% ce qui correspond à
%    f= @(t,y) -A*y
% avec
%    A = [ lambda_1   0          0; ...
%         -lambda_1   lambda_2   0; ...
%          0         -lambda_2   lambda_3 ];
y0 = [1;0;0];
% pas à la limite de stabilité
hstab = 2/lambda_1;
Tmax = 12*3600;
```

La stabilité de la méthode d'Euler progressive est illustrée par le code suivant. Les fonctions `eulerp` et `eulerr` sont données en annexe ; il s'agit des versions vectorielles des fonctions de la séance précédente.

```
% Euler progressive
h = 1.01*hstab; % instable
T = 0:h:Tmax;
yp = eulerp(f,y0,T);
plot(T,yp(3,:),'r');
h = 0.4*hstab; % stable
T = 0:h:Tmax;
yp = eulerp(f,y0,T);
[ypmax indp] = max(yp(3,:));
tpmax = T(indp);
figure; plot(T,yp(3,:),'b');
% Euler rétrograde : toujours stable
h = 1.01*hstab;
T = 0:h:Tmax;
yp = eulerr(f,y0,T);
figure; plot(T,yp(3,:),'r');
```

d) La valeur maximale de pas $h_{5\%}$ pour la méthode d'Euler progressive qui permet de garder les erreurs sur la valeur maximale de N_{Xe} et le temps correspondant $t_{\max}$ inférieures à 5% est

$$h_{5\%} \approx h_{\mathrm{stab}} .$$

Pour ce qui est d'Euler rétrograde, la détermination du pas $h_{5\%}$ peut se faire par essais et erreurs. Si on considérait le pas maximal $h_{5\%}$ tel que l'erreur sur la valeur maximale $N_{\mathrm{Xe}}(t_{\max})$ du nombre de moles est inférieure à 5% pour tout $h \leq h_{5\%}$, on trouverait

$$h_{5\%} \approx 78\, h_{\mathrm{stab}} .$$

Pour ce qui est de l'erreur sur la valeur du temps $t_{\max}$, cette dernière varie fortement avec le choix du pas. En considérant le pas $h_{5\%}$ qui est un multiple de h_{stab} on obtient

$$h_{5\%} \approx 38\, h_{\mathrm{stab}}$$

comme valeur maximale telle que pour tout pas $h \leq h_{5\%}$ on ne dépasse pas $t_{\max}$ de 5% ; des valeurs de h plus grandes peuvent néanmoins redonner une erreur inférieure à 5%. Dans tous ces cas, le pas est donc plus favorable avec la méthode d'Euler rétrograde, principalement parce que cette dernière n'a pas de contraintes de stabilité. Notons que cet avantage pour la méthode d'Euler rétrograde est nettement moins prononcé si on mesure le temps d'exécution, car chaque itération d'Euler rétrograde est plus coûteuse.

Le programme qui a servi à déterminer $h_{5\%}$ (avec les lignes 2 et 6 ajustées pour trouver les bonnes valeurs du pas) est

```
% Euler progressive : h_5% = hstab
h = hstab; Tp = 0:h:Tmax;
tic; yp= eulerp(f,y0,Tp); toc
OK(Tp,yp(3,:))
%  Euler rétrograde : h_5% = 38*hstab
h = 38*hstab; Tr = 0:h:Tmax;
tic; yr = eulerr(f,y0,Tr); toc
OK(Tr,yr(3,:))
```

```
    %
    h = 39*hstab; Tr = 0:h:Tmax;
    tic; yr = eulerr(f,y0,Tr); toc
    OK(Tr,yr(3,:))
    plot(Tp,yp(3,:),'b',Tr,yr(3,:),'r');
```

avec la fonction `OK` (qui vérifie si l'écart est inférieur à 5%) donnée par

```
function res = OK(t,y)
% y doit être un vecteur
    [ymax imax] = max(y);
    tmax = t(imax);
    err = max(abs(tmax-40606)/40606, ...
              abs(ymax-0.4275)/0.4275);
    if(err < 0.05)
        res = 'OK';
    else
        res = 'no';
    end
```

2. On peut procéder comme dans le code suivant, avec les fonctions `eulerp` et `eulerr` données en annexe :

```
   % définir f(t,u)
f = @(t,u) ([u(1)-0.5*u(1)*u(2); ...
             0.1*u(1)*u(2)-u(2)]);
   % conditions initiales
u0 = [1; 1];
   % pas de discrétisation
% h = 0.01; % erreur encore importante,
            % surtout pour t > 10
h = 0.001;  % erreur faible, mais
            % temps de calcul perceptible
T = 0:h:20;
   % résoudre
solp = eulerp(f, u0, T);
solr = eulerr(f, u0, T);
   % graphe de V & P
plot(T, solp(1,:), '--b'); hold on;
plot(T, solr(1,:), '--r')
plot(T, solp(2,:), '-b')
plot(T, solr(2,:), '-r'); hold off
```

En choisissant le pas de discrétisation `h=0.01` on obtient les solutions numériques représentées sur la Figure 1 (gauche) ; on constate qu'il y a une différence (qu'on peut estimer comme significative) entre les solutions données par Euler progressive (bleu) et rétrograde (rouge). Par contre, pour `h=0.001` les deux solutions sont indistinguables à l'oeil, comme on peut aussi le constater sur la Figure 1 (droite).

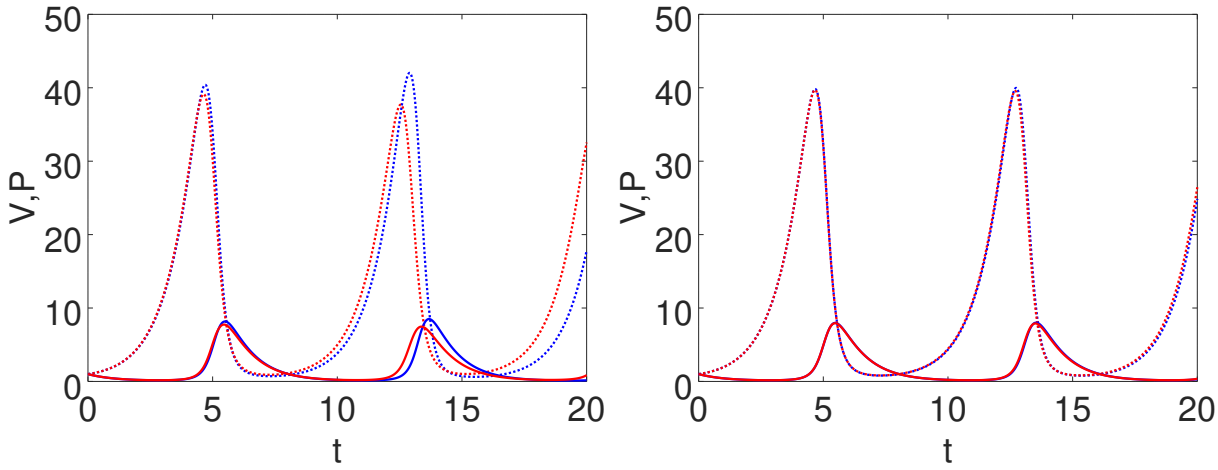

FIGURE 1 – Solutions P (trait interrmpou) et V (trait continu) obtenues avec les méthodes d'Euler progressive (bleu) et rétrograde (rouge) pour un pas `h=0.01` (figure gauche) et `h=0.001` (figure droite).

Annexe

Les codes des fonctions `eulerp` et `eulerr` pour les problèmes vectoriels sont comme suit. D'autres manières d'écrire ces fonctions sont possibles, à condition de faire attention aux dimensions des vecteurs et matrices que vous manipulez.

```
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.
%    L'instruction peut être utilisée avec une fonction F
%    vectorielle de dimension N; dans ce cas, Y0 est un
   vecteur
%    de taille N et le résultat Y sera une matrice N x Nt ,
%    où Nt est le nombre d'éléments dans T.
%
% arguments:
%    F - la fonction de deux variables y et t;
%        t - un scalaire
%        y - un vecteur de même dimension que Y0
%    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.
%    L'instruction peut être utilisée avec une fonction F
%    vectorielle de dimension N; dans ce cas, Y0 est un
   vecteur
%    de taille N et le résultat Y sera une matrice N x Nt ,
%    où Nt est le nombre d'éléments dans T.
%
% arguments:
%    F - la fonction de deux variables y et t;
%        t - un scalaire
%        y - un vecteur de même dimension que Y0
%    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
```