Résolution d'une équation différentielle non linéaire

L'équation de Blasius

1. Avant-propos

Cet article présente une méthode de résolution sous Matlab d'une équation différentielle non linéaire et possédant une condition de bout. L'équation de Blasius est obtenue à partir des équations de Navier-Stokes résolues le long d'une plaque. Pour plus de détails concernant la mise en équation voir les références [1] et [2].

2. Mise en équations

La vitesse d'un fluide de viscosité υ dans la couche limite est donnée par :

$$u = U.f'(\eta)$$

Où η est une variable sans dimension représentant la hauteur de couche limite, elle est calculée à partir des coordonnées géométriques : $\eta = y\left(\frac{U}{\upsilon.x}\right)^{0,5}$

Soit l'équation décrivant l'évolution de la couche limite au-dessus d'une plaque plane :

$$f''' + \frac{1}{2}f.f'' = 0$$

Les conditions aux limites sont :

$$\begin{cases} f(0) = 0 \\ f'(0) = 0 \\ f'(\infty) = 1 \end{cases}$$

3. Résolution numérique type Euler

Cette équation différentielle est résolue via une méthode d'intégration type Euler. Elle est décomposée en 3 équations différentielles couplées :

$$\begin{cases} \dfrac{dy_1}{d\eta} = y_2 \\ \dfrac{dy_2}{d\eta} = y_3 \\ \dfrac{dy_3}{d\eta} = -\dfrac{1}{2}y_1y_3 \end{cases}$$

La condition à la limite en l'infini $f'(\infty) = 1$ est transformée en condition initiale, une procédure itérative type dichotomie permet de trouver la condition initiale pour y_3 qui vérifiera que $y_2(\infty) = 1$. Les conditions initiales pour ce système sont donc :

$$\begin{cases} y_1(0) = 0 \\ y_2(0) = 0 \\ y_3(0) = x \end{cases}$$

La résolution de type Euler pour le système précédent s'écrit dans le cas général:

$$\frac{y_i^{n+1} - y_i^n}{d\eta} = f(y_1, y_2, y_3)$$

4. Résultats

D'après la littérature, la valeur en 0 pour y_3 est de 0,332.

Les résultats sont présentés ici pour une discrétisation de 20 puis de 100 points avec la résolution type Euler. Puis une comparaison des résultats est faite avec ceux obtenus par la routine Matlab® *ode45* (type Runge Kutta ordre 4/5). On remarque que dans les deux cas la condition à la limite en $f'(\infty) = 1$ est bien vérifiée. Néanmoins, les erreurs d'intégration intrinsèques à la méthode d'Euler conduisent à une mauvaise estimation de la valeur de la condition initial, cf. Tableau 1. Malgré tout, ce problème tend à s'atténuer avec l'augmentation du nombre de points d'intégration. On remarque également que la méthode RK4 converge vers la bonne solution même avec peu de points de discrétisation.

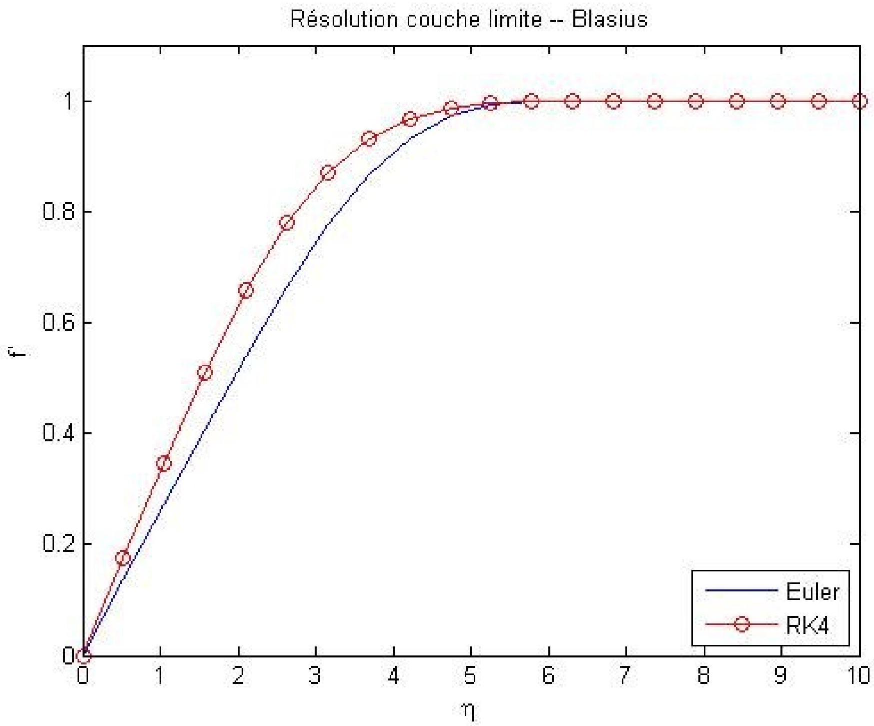

Figure 1 -- Résultats avec 20 points de discrétisation

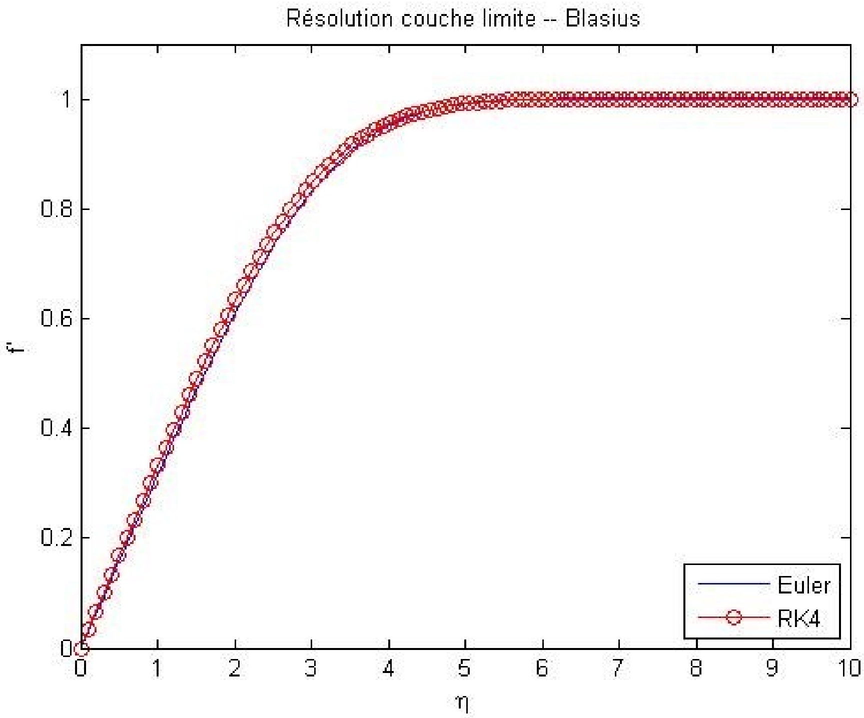

Figure 2 -- Résultats avec 100 points de discrétisation

Valeur de la condition initiale pour y3		
Nombre de points	**Euler**	**RK4 (Matlab)**
20	0.2573	0.3320
100	0.3154	0.3320

Tableau 1 -- Comparaison de la valeur de la condition initiale obtenue

5. Bibliographie

[1] : Fluid Mechanics, F. M. White, 2011 McGraw-Hill

[2] : Hydrodynamique Physique, L. Petit, 2001 EDP Sciences

6. Programme Matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  RESOLUTION DE L'EQUATION DE BLASIUS
%    COUCHE LIMITE AU DESSUS D'UNE PLAQUE PLANE
%
%        f''' + 1/2 * f *f'' = 0
%         f(0) = f'(0) = 0
%         f'(inf) = 1
%
%   RESOLUTION EULER/RK4 + DICCHOTOMIE
%
%        S. CHEVALIER 21/02/12
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
clc

%PARAMETRES
N=100;
l=10; %longueur addimentionnelle
deta=l/(N-1);
eta=0:deta:l;

%INITIALISATION
y1(1:N)=0;
y2(1:N)=0;
y3(1:N)=0;
ya=0; %constante à déterminer pour vérifier f'(inf) = 1
yb=1;
y2N=1;

while abs(y2(N)-y2N) > 0.001
    y3(1)=(ya+yb)/2;
    %RESOLUTION EULER
    for ii=1:N-1
        y1(ii+1)=y2(ii)*deta+y1(ii);
        y2(ii+1)=y3(ii)*deta+y2(ii);
        y3(ii+1)=-1/2*y1(ii)*y3(ii)*deta+y3(ii);
    end

    if y2(N)>y2N
        yb=y3(1);
    else
        ya=y3(1);
    end
end


%AFFICHAGE
clf
figure(1)
plot(eta,y2)
ylim([0 1.1])
disp(sprintf('f'' = %.4f avec résolution Euler',y3(1)))
%RESOLUTION RK4, ROUTINE ODE45
%INITIALISATION
y1(1:N)=0;
y2(1:N)=0;
y3(1:N)=0;
ya=0; %constante à déterminer pour vérifier f'(inf) = 1
yb=1;
y2N=1;
while abs(y2(N)-y2N) > 0.001
    y3(1)=(ya+yb)/2;
    %RESOLUTION EULER
    fun=@(t,y)[y(2);...
        y(3);...
        -1/2*y(1)*y(3)];
    [X,Y]=ode45(fun,eta,[y1(1);y2(1);y3(1)]);
    y1=Y(:,1);
    y2=Y(:,2);
    y3=Y(:,3);
    if y2(N)>y2N
```

```
        yb=y3(1);
    else
        ya=y3(1);
    end
end
%AFFICHAGE
hold on
figure(1)
plot(eta,y2,'r-o')
ylim([0 1.1])
hold off
legend('Euler','RK4','Location','SouthEast')
ylabel('f''')
xlabel('\eta')
title('Résolution couche limite -- Blasius')
disp(sprintf('f'' = %.4f avec résolution RK4',y3(1)))
```