Troisième année de licence M
Analyse numérique 2008/2009

Travaux dirigés sur ordinateur n°**7** :

Méthode de Runge-Kutta

Question 1 :Résolution théorique. Soit l'équation différentielle :

$$(E)\left\{\begin{array}{rcl} y'(t) & = & 1+y^2 \\ y(0) & = & 0 \end{array}\right.$$

avec

$$f(t,y)=1+y^2$$

Résoudre l'équation (E) sur $]-\frac{\pi}{2};\frac{\pi}{2}[$.

On rappelle que pour approcher numériquement la solution d'une équation différentielle sur l'intervalle $[a,b]$:

$$(E)\left\{\begin{array}{rcl} y'(t) & = & f(t,y(t)) \\ y(t_0) & = & y_0 \end{array}\right.$$

pour chaque N, on utilise une discrétisation de l'intervalle en posant $h=\frac{b-a}{N}$, $t_n=a+nh$, $0\leq n\leq N$ et

$$u_{n+1}=u_n+h\phi(t_n,u_n,h),\ 0\leq n\leq N-1$$

Le premier terme de la suite u_0 est donné (il peut être différent de y_0).

On utilise le schéma à un pas défini par $u_0=0$ et

$$\phi(t,u,h)=\frac{1}{6}\Big(l_1(t,u,h)+2l_2(t,u,h)+2l_3(t,u,h)+l_4(t,u,h)\Big)$$

$$\text{avec}\left\{\begin{array}{l} l_1(t,u,h)=f(t,u) \\ l_2(t,u,h)=f(t+\frac{h}{2},u+\frac{h}{2}l_1(t,u,h)) \\ l_3(t,u,h)=f(t+\frac{h}{2},u+\frac{h}{2}l_2(t,u,h)) \\ l_4(t,u,h)=f(t+h,u+hl_3(t,u,h)) \end{array}\right.$$

Aller chercher le code suivant Runge.sce sur la page d'Alexandre Mizrahi :

```
clf(); clear

function z=f(t,y) // définit la fonction f de l'énoncé.
  z=1+y^2
endfunction

y0=0; a=0; b=1.55;

function z=F(t,y,h) // définit la fonction Phi
    l1=f(t,y)
    l2=f(t+h/2,y+h/2*l1)
    l3=f(t+h/2,y+h/2*l2)
    l4=f(t+h,y+h*l3)
    // z=........................à compléter
endfunction

function Dessine // dessine la solution puis l'ecart entre la solution et la valeur ap
  N=100;
  x=[0:N]/N*(b-a);
  y=tan(x);
```

```
  plot2d(x,y,style=6) //rose
  //h=...............permet de définir h ;
  u(1)=y0;
  for i=2:N+1 u(i)=u(i-1)+h*F((i-2)*h,u(i-1),h); end;
  plot2d(x,w,style=-2) // trace des signes + aux points de la suite d'approximation
  xset("window",1)
  plot2d(x,u-y',style=2)  //bleu
endfunction

function e=erreur(N)
// calcul la différence entre la valeur de la solution en b et la valeur approchée
  h=(b-a)/N;
  w=y0;
  //for i=2:N+1 u=..........................; end;
  //à compléter pour qu'à chaque itération w passe de la valeur de w(n) à celle de w(n
  e=abs(w-tan(b))
endfunction

function DessineErreur
  n=100:5:200;
  T=[]
  for k=n
    T=[T,erreur(k)];
  end
  xset("window",3); //plot2d(.......à compléter pour tracer l'erreur en b en fonction
  xset("window",4); //plot2d(.......
      // à compléter pour tracer le log de l'erreur en b en fonction de log(N).
endfunction
```

Question 2 :

1. Compléter la fonction `F`, pour définir la fonction de trois variable Φ.
2. Compléter la fonction `Dessine`, puis faites la fonctionner, dans la fenêtre graphique (0) l'approximation semble-t-elle bonne ?
3. Dans la fenêtre graphique (1) que peut-on dire de l'erreur ?
4. Jusqu'à présent nous avons utilisé $N = 100$, combien de fois scilab a-t-il évalué la fonction `F`, la fonction `f` ?
5. La fonction `erreur` doit calculer la différence entre la valeur de la solution y en b : $y(b)$ et la valeur approchée de $y(b)$ à l'aide du schéma de Runge-Kutta qui est donc u_N, lorsque on a utilisé N itérations. Compléter le code de la fonction `erreur`.
6. La fonction `DessineErreur` permet de tracer l'erreur entre $y(b)$ et u_N, en fonction de N, puis idem pour les log.
   (a) Compléter la fonction `DessineErreur`.
   (b) Que peut-on dire de l'erreur à la vue de la fenêtre graphique (3).
   (c) A l'aide de la fenêtre graphique (4) conjecturer que l'erreur est un $O\big(N^{-\alpha}\big)$.

Question 3 : Dans la question précédente, nous étions dans un cas particulier car nous connaissions la solution, supposons que $f(t, y) = \sqrt(t) + y^2$, on voudrais pouvoir estimer l'erreur, mais comme on ne connaît pas à priori la solution, on peut juste étudier la vitesse de convergence de la suite, par exemple on suppose que $u_N - y(b) \sim \frac{K}{N^\beta}$, et l'on étudie (en le traçant par exemple) la suite $(u_{2N}^{2N} - u_N^N)$ en fonction de N. u_{2N}^{2N} représente la valeur approchée de $y(b)$ par la méthode avec $2N$ itérations et u_N^N avec N itérations.