INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

3 GM

Année universitaire 2016-2017
TP-Examen Méthodes Numérique
Vendredi 19 mai 2017 — durée : 2h

1 Intégration numérique

Soit $f \in C^4(\mathbb{R},\mathbb{R})$ et $x_0 \in \mathbb{R}$. Pour $h > 0$, on cherche à déterminer une valeur approchée de $I(x_0,h) = \int_{x_0-h}^{x_0+h} \varphi(t)\,dt$. Pour ce faire, on remplace la fonction φ par son interpolant de Lagrange aux point $x_0 - h$, x_0 et $x_0 + h$. Ainsi l'approximation de $I(x_0,h)$ est

$$J(x_0,h) = \frac{h}{3}(\varphi(x_0-h) + 4\varphi(x_0) + \varphi(x_0+h)). \tag{1}$$

1. Écrire une fonction `MATLAB`, `phi.m` calculant $\varphi(\boldsymbol{x})$ pour un tableau $\boldsymbol{x}$ avec $\varphi(t) = \dfrac{1}{1+t^2}$.

```
>> >> phi([0;1])
ans =
    1.0000
    0.5000
```

2. Écrire une fonction `MATLAB`, `Iapprox.m` : `J = Iapprox(nomf,x0,h)` calculant $J(x_0,h)$

```
>> Iapprox('phi',0,0.2)
ans =
    0.3949
```

3. En faisant varier $h = 1/n$, étudier l'erreur $err = |I(x_0,h) - J(x_0,h)|$. Écrire un programme script `MATLAB`, `VI1.m` ("VI" sont vos initiales) qui pour $x_0 = 0$, part du tableau de valeurs de n, $arrn = 100 : 110$ pour constuire le tableau $arrerr$[1] puis dessine $\ln(arrerr)$ en fonction de

[1] $\int_{-h}^{h} \frac{1}{1+t^2}\,dt = 2\arctan(h)$

$\ln(arrn)$. Conclure en commentaire sur l'ordre de la méthode. Exemple $\varphi(t) = \dfrac{1}{1+t^2}$, $x_0 = 0$.

>> VI1

2 Équation différentielle

Soit f une fonction définie et continue de $[t_1, t_1+T] \times \mathbb{R}^p \to \mathbb{R}^p$ lipschizienne par rapport à la seconde variable et soit $\boldsymbol{y} : [t_1, t_1+T] \to \mathbb{R}^p$, la solution du problème de Cauchy suivant :

$$\begin{cases} \boldsymbol{y}'(t) = f(t, \boldsymbol{y}(t)),\ t \in [t_1, t_1+T] \\ \boldsymbol{y}(t_1) = \boldsymbol{\eta}, \qquad \text{où} \quad \eta \in \mathbb{R}^p. \end{cases} \tag{2}$$

Pour $n \in \mathbb{N}^*$, on pose $h = \dfrac{T}{n}$ et $t_i = t_1 + (i-1)h$, pour $i = 1, \ldots, n+1$, une subdivision uniforme de $[t_1, t_1+T]$. Pour résoudre le problème (2), nous considérons le schéma implicite suivant

initialisation :

$$\begin{aligned} \boldsymbol{u}_1 &= \boldsymbol{\eta}, \\ &\quad \boldsymbol{k}_1 = hf(t_1, \boldsymbol{u}_1),\ \boldsymbol{k}_2 = hf(t_1 + h/2, \boldsymbol{u}_1 + \boldsymbol{k}_1/2), \\ &\quad \boldsymbol{k}_3 = hf(t_1 + h/2, \boldsymbol{u}_1 + \boldsymbol{k}_2/2), \boldsymbol{k}_4 = hf(t_2, \boldsymbol{u}_1 + \boldsymbol{k}_3), \\ \boldsymbol{u}_2 &= \boldsymbol{u}_1 + (\boldsymbol{k}_1 + 2\boldsymbol{k}_2 + 2\boldsymbol{k}_3 + \boldsymbol{k}_4)/6, \end{aligned}$$

puis pour $i = 2, \ldots, n$

$$\boldsymbol{u}_{i+1} = \boldsymbol{u}_{i-1} + \frac{h}{3}(f(t_{i-1}, \boldsymbol{u}_{i-1}) + 4f(t_i, \boldsymbol{u}_i) + f(t_{i+1}, \boldsymbol{u}_{i+1})). \tag{3}$$

$\boldsymbol{u}_i \in \mathbb{R}^m$ est la valeur approchée de $\boldsymbol{y}(t_i)$.

2.1 Équation linéaire

Dans (2), on prend $p = 1$, $f(t, u) = -u + t + 1$, $t_1 = 0$, $T = 1$, $\eta = 1$; ainsi la solution de (2) est $y(t) = e^{-t} + t$

1. Écrire une fonction `f1.m` calculant $f(t, u)$ et une fonction `solex.m` calculant $y(t)$.

2. Comme f est linéaire, (3) s'écrit

$$u_{i+1} = \frac{(1 - h/3)u_{i-1} - 4h/3u_i + 2h(t_i + 1)}{1 + h/3}.$$

Écrire une fonction `schema1.m` qui étant donné n calcule la subdivision uniforme $\boldsymbol{t}$ et les valeurs approchées $\boldsymbol{u} = [u_1, \ldots, u_{n+1}]$. Test avec $n = 4$.

```
>> [t,u]=schema1(4)
t =
         0    0.2500    0.5000    0.7500    1.0000
u =
    1.0000    1.0288    1.1065    1.2224    1.3679
```

3. Écrire un programme script `VI2.m` qui pour $n \in \mathbb{N}^*$ calcule les solutions exactes et approchées, les dessine et calcule l'erreur $err = \max_{i=1,\ldots,n+1} |y(t_i) - u_i|$. Test avec $n = 4$.

```
>> VI2
n =
     4
err =
   1.4526e-05
```

4. Écrire un programme script `VI3.m` qui étudie l'erreur quand n varie en partant d'un tableau $arrn = [50 : 10 : 200]$. Conclure sur l'ordre en commentaire.

2.2 Équation non linéaire du second ordre

On considère maintenant le problème

$$\begin{cases} y''(t) = -\sin(y(t)) + \sin(\sin(\pi(t + 1/2))) - \pi^2 \sin(\pi(t + 1/2)),\ t \in [0, 1] \\ y(0) = 1,\ y'(0) = 0, \end{cases} \tag{4}$$

dont la solution est $y(t) = \sin(\pi(t + 1/2))$.

1. Transformer le problème en un système du premier ordre avec $p = 2$:

$$\begin{cases} \boldsymbol{y}'(t) = f2(t, \boldsymbol{y}(t)),\ t \in [0, 1] \\ \boldsymbol{y}(0) = \boldsymbol{\eta}, \end{cases} \quad \text{où } \boldsymbol{y}(t) = \begin{bmatrix} y(t) \\ y'(t) \end{bmatrix}. \tag{5}$$

puis écrire une fonction `f2.m` définissant $\boldsymbol{v} = f2(t, \boldsymbol{u})$ pour un vecteur $\boldsymbol{u} \in \mathbb{R}^2$

```
>> v=f2(0.5,[1;-1])
```

L'équation (3) n'est pas linéaire et va être résolue par une méthode itérative supposée convergente au bout d'un certain nombre d'itérations ni. L'algorithme est alors

- Calcul explicite de $\boldsymbol{u}_1$ et $\boldsymbol{u}_2$,
- Pour $i = 2, \ldots, n$
  - Calcul de $\boldsymbol{a} = \boldsymbol{u}_{i-1} + h/3(f2(t_{i-1}, \boldsymbol{u}_{i-1}) + 4f2(t_i, \boldsymbol{u}_i))$,
  - On pose $\boldsymbol{x}_0 = [0, 0]^T$,
  - Mise en oeuvre du processus itératif sur ni itérations : $\boldsymbol{x}_1 = \boldsymbol{a} + h/3f2(t_{i+1}, \boldsymbol{x}_0)$ puis mis à jour $\boldsymbol{x}_0 = \boldsymbol{x}_1$
  - Sortie des itérations avec $\boldsymbol{u}_{i+1} = \boldsymbol{x}_0$.
- fin i

2. Écrire une fonction `schema2.m` qui étant donné n calcule la subdivision uniforme $\boldsymbol{t}$ et les valeurs approchées $\boldsymbol{u} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{n+1}] \in \mathbb{R}^{2\times(n+1)}$. On prendra $ni = 20$. Test :

```
>> [t,u]=schema2(4);
```

3. Écrire un programme script `VI4.m` qui pour $n \in \mathbb{N}^*$ calcule les solutions exactes et approchées, les dessine et calcule l'erreur $err = \max_{i=1,\ldots,n+1} |y(t_i) - u_{1,i}|$. Test avec $n = 10$, affichage de l'erreur.

4. Écrire un programme script `VI5.m` qui étudie l'erreur quand n varie en partant d'un tableau $arrn = [50 : 10 : 200]$. Conclure sur l'ordre en commentaire.