FSAB1104 : solution de l'examen de janvier 2017

1 Hillary perdue dans les différences finies

Pour estimer la dérivée seconde à l'origine d'une fonction $u(x)$, Hillary a trouvé la formule suivante :

$$u''(0) \approx \frac{2U_0 + \alpha U_h + \beta U_{2h} - U_{3h}}{h^2}$$

Malencontreusement, un slave facétieux lui a chapardé les valeurs des paramètres réels α et β qui donnent l'ordre de précision le plus élevé possible pour cette formule de différences finies.

1. Aider Hillary à retrouver les valeurs des deux paramètres.

Il suffit d'écrire les développements en série de Taylor comme suit :

$$\begin{aligned}
U_h &= U_0 + hU_0' + \frac{h^2}{2}U_0'' + \frac{h^3}{6}U_0''' + \frac{h^4}{24}U_0'''' \ldots \\
U_{2h} &= U_0 + 2hU_0' + \frac{4h^2}{2}U_0'' + \frac{8h^3}{6}U_0''' + \frac{16h^4}{24}U_0'''' \ldots \\
U_{3h} &= U_0 + 3hU_0' + \frac{9h^2}{2}U_0'' + \frac{27h^3}{6}U_0''' + \frac{81h^4}{24}U_0'''' \ldots
\end{aligned}$$

En substituant ces développements dans la formule d'Hillary, on obtient alors :

$$\frac{(2+\alpha+\beta-1)}{h^2}U_0 + \frac{(\alpha+2\beta-3)}{h}U_0' + \frac{(\alpha+4\beta-9)}{2}U_0'' + \frac{(\alpha+8\beta-27)}{6}hU_0''' + \frac{(\alpha+16\beta-81)}{24}h^2U_0'''$$

Pour que la formule d'Hillary aie du sens, il faut annuler les coefficients de U_0 et de U_0'. Il faut aussi exiger que le coefficient de U_0'' soit unitaire. Il y a trois conditions et deux inconnues : il est donc vraiment possible de vérifier que la réponse trouvée est bien correcte. C'est donc vraiment impardonnable de ne pas obtenir la solution exacte !

De la première condition sur U_0, on peut déduire que $\alpha = 1 - \beta$...

... et de la seconde condition sur U_0', on déduit immédiatement que :

$$\boxed{\begin{aligned} \alpha &= -5 \\ \beta &= 4 \end{aligned}}$$

2. Donner l'ordre de précision et calculer l'expression du terme d'erreur.

Ensuite, on substitue α et β dans la formule d'Hillary pour obtenir le terme d'erreur :-)

$$\underbrace{\frac{(2-5+4-1)}{h^2}}_{=0} U_0 + \underbrace{\frac{(-5+8-3)}{h}}_{=0} U_0' + \underbrace{\frac{(-5+16-9)}{2}}_{=1} U_0'' + \underbrace{\frac{(-5+32-27)}{6}}_{=0} hU_0''' + \underbrace{\frac{(-5+64-81)}{24}}_{=11/12} h^2 U_0'''$$

On conclut finalement : $\boxed{u'''0) \quad = \quad \frac{2U_0 + \alpha U_h + \beta U_{2h} - U_{3h}}{h^2} - \frac{11}{12} h^2 u^{(4)}(\zeta)}$

La méthode d'Hillary est donc d'ordre deux et correspond à une différence amont usuelle :-)

3. Calculer la valeur optimale de h afin de minimiser l'erreur totale, c'est-à-dire, les contributions conjointes de l'erreur de discrétisation et des erreurs d'arrondi dues au calcul en virgule flottante. Les valeurs U_0, U_h, U_{2h} et U_{3h} sont fournies avec une double précision ($\epsilon = 10^{-16}$). La dérivée quatrième de u est constante et vaut l'unité.

Il suffit de minimiser $E(h) \quad = \quad \frac{(2+5+4+1)}{h^2}\epsilon + \frac{11}{12}h^2.$

Attention : les erreurs ne font que s'additionner et donc on prend la valeur absolue de tous les coefficients de la formule d'Hillary et on prend la somme de l'erreur d'arrondi et de l'erreur de discrétisation même si cette dernière est négative !

On exige donc que $f'(h) \quad = \quad -\frac{24}{h^3}\epsilon + \frac{22h}{12} \quad = \quad 0$

$$\downarrow$$

$$h^4 \quad = \quad \frac{144}{11}\ \epsilon$$

On conclut que le pas optimal est donné par : $\boxed{h \quad = \quad \sqrt[4]{\frac{144}{11}}\ \epsilon}$

2 Donald veut calculer pi !

Pour obtenir une approximation de π, Donald utilise la méthode de Newton-Raphson pour la fonction :

$$f(x) = \cos\left(\frac{x}{2}\right)$$

En partant avec $x_1 = 3$, son programme MATLAB fournit les résultats :

```
x =  3.00000000000000 : Error 1.4159265e-01 at iteration 1
x =  3.14182968860530 : Error 2.3703502e-04 at iteration 2
x =  3.14159265358868 : Error 1.1097789e-12 at iteration 3
x =  3.14159265358979 : Error 0.0000000e+00 at iteration 4
```

1. Ecrire la relation entre x_{n+1} et x_n qui définit la suite générée par la méthode de Newton-Raphson.

   Il suffit d'écrire : $\boxed{x_{i+1} \quad = \quad x_i + 2\,\frac{\cos\left(\frac{x_i}{2}\right)}{\sin\left(\frac{x_i}{2}\right)}}$

2. Définir rigoureusement le taux de convergence d'une méthode itérative.
   Si les itérations convergent et s'il existe deux constantes positives $C < 1$ et $r \geq 1$ telles que

$$\lim_{i\to\infty} \frac{|e_i|}{|e_{i-1}|^r} = C$$

   on dit que la séquence converge avec un taux de convergence r.
   Il est aussi judicieux que la constante r soit supérieure à l'unité.

3. Démontrer que la méthode de Newton-Raphson converge avec un taux de trois pour $f(x) = \cos\left(\frac{x}{2}\right)$.

   Cela provient du fait que : $f''(\pi) = -\frac{1}{4}\ \cos\left(\frac{\pi}{2}\right) = 0$

   Pour effectuer la démonstration, il suffit tout simplement d'écrire

$$x_{i+1} \quad = \quad x_i - \frac{f(x_i)}{f'(x_i)}$$

$$\underbrace{x_{i+1} - x}_{e_{i+1}} \quad = \quad \underbrace{x_i - x}_{e_i} \; - \; \frac{f(x + e_i)}{f'(x + e_i)}$$

   $\downarrow$ *En effectuant un développement en série de Taylor pour* $g(x) = \frac{f(x)}{f'(xi)}$ *:*

$$e_{i+1} \quad = \quad e_i - g(x) - g'(x)\ e_i - g''(x)\ \frac{e_i^2}{2} + g'''(x)\ \frac{e_i^3}{6} + \ldots$$

   $\downarrow$ *En observant que $g(x) = g''(x) = 0$ et $g'(x) = 1$ car $f(x) = f''(x) = 0$*

$$e_{i+1} \quad = \quad g'''(x)\ \frac{e_i^3}{6} \qquad \square$$

Pour les sceptiques, les grincheux et les fans d'algèbre, la justification détaillée s'effectue comme suit :

$$
\begin{aligned}
g(x) &= \frac{f}{f'} = 0 \\
g'(x) &= \frac{f'}{f'} - \frac{ff''}{(f')^2} = 1 - \frac{ff''}{(f')^2} &&= 1 \\
g''(x) &= -\frac{f'f''}{(f')^2} - \frac{ff'''}{(f')^2} + \frac{2ff''f''}{(f')^3} &&= 0
\end{aligned}
$$

Pour cette question (difficile), plusieurs réponses et démonstrations étaient possibles. Pas mal d'étudiants ont la bonne intuition, mais peu arrivent à rédiger proprement une démonstration convaincante. Il est assez décevant d'observer qu'une immense majorité d'étudiants ne répondent pas du tout à cette question, alors que citer l'intuition que cela provenait du caractère particulier de la dérivée seconde de la fonction cosinus rapportait déjà pas mal de points !

4. Ecrire le programme MATLAB de Donald :

   ```
   function [x error] = cosinus(x0,tol,nmax)
   ```

   Les vecteurs `x` et `error` contiendront les itérations successives ainsi que l'erreur correspondante. Les arguments sont le nombre maximal d'itérations `nmax`, la tolérance maximale `tol` et le candidat initial `x0`. L'exécution de `[x error] = cosinus(3,1e-13,100)` fournit les résultats ci-dessus.

   Une implémentation possible est donnée par :

   ```
   function [x error] = cosinus(x0,tol,nmax)
     n=1; delta = tol + 1; x(1) = x0;
     while abs(delta) >= tol && n < nmax
         error(n) = abs(pi - x(n));
         delta = 2 * cot(x(n)/2.0);
         x(n+1) = x(n) + delta;
         n = n + 1;
     end
   end
   ```

5. Estimer l'ordre de convergence observé par Donald lors de l'exécution de son programme. Est-ce en accord avec la théorie ?

   Le taux de convergence observé est trois (et même un peu plus) car on triple (et même un peu plus) le nombre de chiffres significatifs à chaque itération.
   On pouvait aussi simplement noter que $(10^{-4})^3 = 10^{-12}$.

   Il ne fallait pas faire des calculs détaillés, mais une très brève justification était requise !
   Certains étudiants se lancent dans des calculs inutilement complexes et hasardeux...
   L'estimation (non demandée évidemment :-) via MATLAB

   ```
   [x error] = cosinus(3,1e-13,100);
   Order = log(error(2:end-1)./error(1:end-2))./log(error(1:end-2)) + 1
   ```

   fournit les valeurs suivantes : `Order = 4.2702 3.2977`
   C'est donc bien en accord avec la théorie !

3 Heun revisité par Barack

Pour obtenir une solution approchée d'une équation différentielle $u'(x) = f(x, u(x))$,
Barack propose d'utiliser une nouvelle méthode de Runge-Kutta définie comme suit :

$$
\begin{aligned}
U_{i+1} &= U_i + h\left(\frac{3K_1 + K_2}{4}\right) \\
K_1 &= f(X_i, U_i) \\
K_2 &= f(X_i + \alpha, U_i + \beta K_1)
\end{aligned}
$$

1. Calculer les valeurs de α et β afin que cette méthode soit du même ordre que la méthode de Heun.

   Il suffit d'effectuer le développement en série de Taylor :

$$
U_{i+1} = U_i + h\left(\frac{3}{4}F_i + \frac{1}{4}\left(F_i + \alpha\ \frac{\partial F_i}{\partial x} + \mathcal{O}(\alpha^2) + \beta\ \frac{\partial F_i}{\partial u}F_i + \mathcal{O}(\beta^2)\right)\right)
$$

$$
\downarrow
$$

$$
U_{i+1} = U_i + h\ F_i + \frac{h\alpha}{4}\frac{\partial F_i}{\partial x} + + \frac{h\beta}{4}\ \frac{\partial F_i}{\partial u}F_i + \mathcal{O}(h\beta^2) + \mathcal{O}(h\alpha^2)
$$

   Maintenant, écrivons la valeur que l'on obtiendrait avec une méthode de Taylor d'ordre deux :

$$
U_{i+1} = U_i + h\ F_i + \frac{h^2}{2}\ \frac{\partial F_i}{\partial x} + \frac{h^2}{2}\ \frac{\partial f}{\partial u}\ F_i + \mathcal{O}(h^3)
$$

   En identifiant terme à terme les deux développements, on obtient : $\boxed{\alpha = \beta = 2h}$

2. Donner l'expression analytique de la région de stabilité de la méthode de Barack.
   Esquisser cette région dans le plan complexe en définissant précisément les axes de la figure.
   Y indiquer aussi la zone de stabilité de la méthode d'Euler explicite.
   Ici, on considère évidemment le problème modèle habituel : $u'(x) = \lambda\ u(x)$.

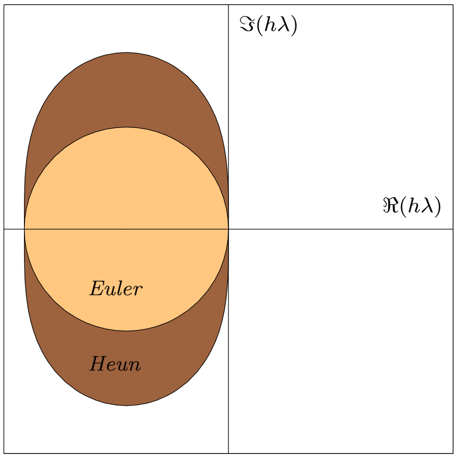
   On évalue simplement U_{i+1} pour le problème modèle :

$$
\begin{aligned}
K_1 &= \lambda U_i \\
K_2 &= \lambda\big(U_i + 2hK_1\big) = \lambda\big(U_i + 2h\lambda U_i\big)
\end{aligned}
$$

$$
\downarrow
$$

$$
U_{i+1} = U_i + \frac{h}{4}\Big(3K_1 + K_2\Big) = \left(1 + h\lambda + \frac{h^2\lambda^2}{2}\right)\ U_i
$$

   La zone de stabilité de la méthode de Barack est donc : $\boxed{\left|1 + h\lambda + \frac{h^2\lambda^2}{2}\right| \leq 1}$



Il est essentiel d'indiquer correctement les axes !

La zone de stabilité de la méthode de Barack est la même que celle de la méthode de Heun qui coïncide avec celle de Taylor d'ordre deux... Au passage, on pouvait obtenir très aisément β en exigeant que ces zones de stabilité soit identique...

3. Ecrire une fonction MATLAB[1]

   `function [X,U] =  barackIntegrate(n,h,U0,f)`

   qui retourne X_i et U_i obtenues en effectuant n pas `h` avec la méthode de Barack.
   Ce sont deux vecteurs de taille `n+1`. On commence le calcul avec `U0` en $x = 0$.
   Le dernier argument `f` est un pointeur vers la fonction `dudx = f(x,u)`.

   Une implémentation possible est donnée par :

```
function [X,U] =  barackIntegrate(n,h,U0,f)
  X = linspace(0,n*h,n+1);
  U = zeros(n+1,1);  U(1) = U0;
  for i=1:n
      K1 = f(X(i),U(i));
      K2 = f(X(i)+2*h,U(i)*2*h*K1);
      U(i+1) = U(i) + h * (3*K1 + K2)/4;
  end
end
```

   Ce programme était vraiment simple à écrire. Ne pas oublier de pré-allouer les deux vecteurs !
   Il fallait impérativement une fonction f avec deux arguments et ne pas bêtement recopier la solution d'un autre exercice du syllabus très proche de cette question d'examen :-)

Toutes mes félicitations aux 6 étudiants qui ont obtenu 20/20 : c'était donc vraiment faisable ! Les NURBS, l'intégration et les équations aux dérivées partielles, ce sera pour septembre : enfin, peut-être.... Et voilà faut pas toujours écouter Pierre-Alexandre qui se trompe parfois dans les prédictions sur les examens de méthodes numériques c'est comme tous ces sondages qui prédisaient la victoire d'Hillary : on se fait toujours avoir !

Bonnes vacances à tous : soyez prudents sur vos skis et ne buvez pas trop après :-)

[1] Il est parfaitement possible d'écrire la quasi-totalité du programme et d'obtenir l'ensemble des points qui y sont associé même si vous n'avez pas obtenu les valeurs de α et β à la première sous-question !