Matmeca 1ère année - TP de programmation en Fortran
année 2015-2016

TP 1 : méthode numériques pour quelques edo

où l'on verra que des méthodes d'ordre élevé peuvent être utiles ...

---

1 Un problème de tir

À $t = 0$, on lance un projectile (assimilé à un point de masse m) depuis un point repéré par son altitude $y = 0$ et sa position $x = 0, z = 0$. La vitesse initiale du projectile est $v = (v_{x0}, v_{y0}, 0)$. En négligeant les effets de rotation de la terre (force de Coriolis), la trajectoire du projectile s'effectue dans le plan définit par $z = 0$. En négligeant en outre les effets de frottement dus à l'air, on trouve l'équation du mouvement suivante (donnée par les lois de la dynamique de Newton) :

$$
\begin{aligned}
x'(t) &= v_x(t) \\
y'(t) &= v_y(t) \\
v_x'(t) &= 0 \\
v_y'(t) &= \mathbf{g}
\end{aligned}
$$

où $\mathbf{g} = (0, 0, -g)$ est l'accélération gravitationnelle. La solution de ce système est facile à calculer, et l'on trouve

$$(x(t), y(t), v_x(t), v_y(t)) = (v_{x0}t, v_{y0}t - g\frac{t^2}{2}, v_{x0}, v_{y0} - gt),$$

ce qui montre que le projectile suit une parabole et retombe au sol au temps $t_{max} = 2v_{y0}/g$.

Question 1. Écrivez un programme contenant tous les paramètres physiques du problème qui trace la solution exacte dans un fichier (pour 100 valeurs de t de 0 à t_{max}). Prendre les données suivantes : $m = 1$, $v_{x0} = 10$, $v_{y0} = 10$, $g = 10$.

Question 2. Mettre le système différentiel sous la forme $X'(t) = F(X(t))$, et programmez le schéma d'Euler explicite pour le résoudre. Le pas de temps sera calculé en fixant un nombre d'itérations n_{max}, puis en posant $\Delta t = t_{max}/n_{max}$ de façon à atteindre le temps final auquel le projectile est censé toucher le sol. Écrivez la solution numérique dans un autre fichier, et vérifiez qu'elle est assez proche de la solution exacte. Vérifiez ensuite que cette solution numérique est d'autant plus précise que Δt est petit. Trouvez enfin un pas de temps qui soit un bon compromis entre précision (qui doit être suffisamment bonne) et nombre de pas de temps (qui ne doit pas être trop grand).

Question 3. Si vous n'avez pas écrit votre programme de façon modulaire, c'est le moment ! Insérez le calcul du second membre $F(X)$ dans une fonction ou une subroutine, et encapsulez-la dans un module. Cette façon de programmer simplifie considérablement le travail à faire dans les questions suivantes.

Question 4. Programmez à présent un schéma de Runge Kutta d'ordre 2, par exemple celui du point milieu, donné par

$$\begin{aligned} k_1 &= F(X) \\ k_2 &= F(X + \frac{\Delta t}{2} k1) \\ X_{n+1} &= X_n + \Delta t k_2. \end{aligned}$$

Constatez le gain énorme en terme de précision et de coût qu'apporte ce schéma par rapport à Euler explicite.

Question 5. On tient compte à présent des frottements subis par le projectile. On suppose que l'air exerce sur le solide une force $f = -kv$ qui s'oppose donc au mouvement. Si le coefficient de frottement k est constant, on peut encore résoudre le problème analytiquement. En réalité, le coefficient de frottement est plutôt *non linéaire*, par exemple $k(v) = \mu\|v\|$. Il est alors beaucoup plus difficile de résoudre le problème analytiquement. Écrivez le nouveau système différentiel, et modifiez le programme précédent pour calculer une solution numérique avec le schéma d'Euler explicite. On prendra le paramètre $\mu = 0.5$ et on arrêtera le calcul quand le projectile touche le sol. Vérifiez que la solution obtenue est cohérente avec l'idée que vous vous faites du phénomène physique.

Question 6. On augmente maintenant les frottements en prenant $\mu = 4$. Constatez que le schéma d'Euler explicite donne une solution non physique si Δt est tel que $n_{max} = 60$ (le projectile repart en arrière). Constatez que ceci disparaît en diminuant le pas de temps. Constatez aussi que le schéma RK2 fonctionne correctement avec $n_{max} = 60$. Voir en fin de TP pour une explication de ce phénomène.

2 Mécanique céleste : orbites planétaires

Le but de cet exercice est de calculer numériquement les trajectoires autour du soleil de 5 planètes du système solaire : Jupiter, Saturne, Uranus, Neptune, et Pluton. Il est tiré du polycopié *Introduction à l'Analyse Numérique* de Ernst Hairer, disponible sur `http://www.unige.ch/~hairer/polycop.html`.

On considère que chacune de ces planètes n'est soumise qu'à l'attraction gravitationnelle exercée par toutes les autres, plus celle du soleil. On rappelle que la force gravitionnelle exercée par un corps B de masse m_B situé en $x_B \in \mathbb{R}^3$ sur un corps A de masse m_A situé en $x_A \in \mathbb{R}^3$ est

$$F_{B\to A} = -G m_B m_A \frac{x_A - x_B}{\|x_A - x_B\|^3},$$

où G est la constante de gravitation universelle. La loi fondamentale de la dynamique nous permet alors d'écrire les équations suivantes :

$$x_i''(t) = -G \sum_{j=0,\neq i}^{5} m_j \frac{x_i - x_j}{\|x_i - x_j\|^3} \tag{1}$$

pour $i = 1$ à 5, où x_i, m_i sont les position et masse de Jupiter ($i = 1$), Saturne ($i = 2$), Uranus ($i = 3$), Neptune ($i = 4$), et Pluton ($i = 5$). Le mouvement de ces planètes est décrit dans le référentiel lié au soleil qui a donc la position fixe $x_0 = 0$. Les inconnues du problème sont donc les positions $x_i(t)$ pour $i = 1$ à 5.

En notant $v_i = x_i'$ la vitesse de la planète numéro i, on peut réécrire les relations précédentes sous forme d'un système d'équations différentielles du premier ordre :

$$\begin{aligned} x_i'(t) &= v_i(t) \\ v_i'(t) &= -G \sum_{j=0,\neq i}^{5} m_j \frac{x_i - x_j}{\|x_i - x_j\|^3} \end{aligned} \tag{2}$$

pour $i = 1$ à 5.

La masse du soleil est $m_0 = 1.00000597682$ (modifiée pour tenir compte des planètes proches) et la constante de gravitation universelle est $G = 2.95912208286e - 4$. Les unités de masse cont relatives à celle du soleil, les distance sont en unités astronomiques (1 AU = 149 597 870 km), et le temps est compté en jours terrestres. Pour les 5 planètes, les positions et vitesses initiales calculées le 5 septembre 1994 sont données dans la table ci-dessous :

x_1	-3.5023653	-3.8169847	-1.5507963
x_2	9.0755314	-3.0458353	-1.6483708
x_3	8.3101420	-16.2901086	-7.2521278
x_4	11.4707666	-25.7294829	-10.8168456
x_5	-15.5387357	-25.2225594	-3.1902382
v_1	0.00565429	-0.00412490	-0.00190589
v_2	0.00168318	0.00483525	0.00192462
v_3	0.00354178	0.00137102	0.00055029
v_4	0.00288930	0.00114527	0.00039677
v_5	0.00276725	-0.00170702	-0.00136504

Les masses sont :

$$m_1 = 0.000954786104043, \quad m_2 = 0.000285583733151, \quad m_3 = 0.0000437273164546,$$
$$m_4 = 0.0000517759138449 \quad m_5 = 1/(1.3 \times 10^8).$$

Le temps maximum de simulation est $tmax = 100\,000$, temps au bout duquel toutes les planètes ont fait au moins une révolution complète autour du soleil.

On demande de résoudre numériquement ce système avec les méthodes d'Euler explicite, puis la méthodes RK2 (Heunh) et la méthode RK4 classique, en traçant à chaque fois les trajectoires obtenues (un fichier par planète et par schéma). Le pas de temps utilisé sera $\Delta t = 300$. Des suggestions de programmation sont données ci-dessous.

On reprendra la même structure de programmation que pour le problème de tir, en adaptant la structure de donnée à ce nouveau problème, un peu plus complexe. Ainsi, les positions et les vitesses peuvent être stockées dans un unique tableau `x(i=1:10,j=1:3)` de sorte que les 3 coordonnées de la planète numéro i soient `x(i,1:3)` et les 3 composantes de sa vitesse soient `x(i+5,1:3)`, avec i entre 1 et 5.

Pour la visualisation, utilisez un logiciel de visualisation dans l'espace, par exemple gnuplot. Vous pouvez utiliser le script gnuplot `commandes_gnuplot_planetes.txt` disponible sur :

`www.math.u-bordeaux1.fr/~lmieusse/PAGE_WEB/enseignement.html/commandes_gnuplot_planetes.txt`

Question 7. Programmez le schéma d'Euler explicite, et constatez qu'il est instable, quel que soit le pas de temps choisi (essayez $\Delta t = 300$ puis $\Delta t = 10$) : toutes les planètes quittent leur orbite (et leur trajectoire part plus ou moins vite vers l'infini). Cela est d'autant plus visible que les planètes font beaucoup de tours autour du soleil (donc celles qui en sont le plus proche, comme Jupiter et Saturne).

Question 8. Programmez ensuite un schéma RK2, puis RK4, et commentez.

3 Pour aller plus loin

Pourquoi le projectile repart-il en arrière ? Cela est dû un défaut de préservation de la positivité du schéma lié au terme homogène du système (le terme de frottement). En effet, physiquement, le projectile doit toujours aller vers l'avant, c'est-à-dire que sa vitesse horizontale doit rester positive. Cela peut aussi se vérifier mathématiquement sur l'équation (sans la résoudre). On peut aussi écrire le schéma d'Euler explicite sur le système et trouver facilement une condition nécessaire et suffisante sur Δt pour que la composante horizontale de la vitesse reste positive (cette analyse est similaire à celle menée pour vérifier la A-stabilité d'un schéma). Essayez de trouver cette condition et vérifiez qu'elle permet bien d'obtenir la propriété voulue avec votre programme.

Pourquoi les orbites divergent-elles avec Euler explicite ? Le comportement du schéma d'Euler explicite peut se comprendre avec un système plus simple, celui du mouvement circulaire uniforme, qui est

$$x'(t) = -y(t) \quad \text{et} \quad y'(t) = x(t)$$

Vous pouvez écrire le schéma d'Euler explicite associé à ce système et prouver que la distance à l'origine $\sqrt{x_n^2 + y_n^2}$ est strictement croissante (quel que soit le pas de temps), ce qui ressemble au phénomène constaté sur les planètes. Une autre façon de comprendre cette instabilité est de calculer les valeurs propres du système différentiel : elles sont imaginaires pures, et il n'est pas difficile de prouver que le schéma d'Euler explicite est inconditionnellement A-instable pour de telles valeurs propres. Les schémas RK2 et RK4 sont en revanche A-stable sous une condition sur Δt (ce qui peut aussi se prouver facilement).

Outre la A-stabilité, il est souvent souhaitable pour de tels problèmes (dits « hamiltoniens ») d'utiliser des schémas qui préservent les invariants du système (ici, la distance à l'origine, et pour les planètes, l'énergie totale du système). De tels schémas existent, on les appelle schémas *symplectiques*. Un exemple très simple, mais peu précis, est le schéma d'Euler symplectique. Pour un système assez général du type

$$x'(t) = F_x(x(t), v(t)) \quad \text{et} \quad v'(t) = F_v(x(t), v(t)),$$

ce schéma s'écrit

$$x_{n+1} = x_n + \Delta t F_x(x_n, v_n) \quad \text{et} \quad v_{n+1} = v_n + \Delta t F_v(x_{n+1}, v_n).$$

Le système des planètes ou celui du mouvement circulaire peuvent être approchés de cette façon. Vous pouvez tester ce schéma vous-même.

Pour plus d'informations sur ce sujet, un livre de référence est

```
Ernst Hairer, Gerhard Wanner, Christian Lubich
Geometric Numerical Integration, Structure-Preserving Algorithms for Ordinary Differential Equations
Springer Series in Computational Mathematics
Volume 31 2006
```

Des polycopiés issus de ce livre sont disponibles sur cette page :

`http://www.unige.ch/~hairer/polycop.html`