

QUELQUES FONCTIONS APL SIMPLIS POUR LA RESOLUTION NUMERIQUE DES EQUATIONS ET SYSTEMES DIFFERENTIELS

CLAUDE CHACHATY
claude.chachaty@wanadoo.fr.

Cet article m'a été suggéré par la lecture de l'excellent ouvrage de P. Hellings sur l'initiation aux calculs numériques en astrophysique (1). Le premier chapitre de cet ouvrage traite en grande partie des méthodes classiques de résolution des équations et systèmes différentiels, également décrites avec leurs variantes dans la réf. (2). Il m'a semblé utile de montrer que ces méthodes permettant de traiter une multitude de problèmes parfois compliqués, peuvent se traduire en quelques lignes d'APL comme on le voit sur la fonction suivante :

```

EQUADIF
'Equation dy/dx = F(x,y) : ' ⋄ DIMF←ρFAB←FX←□
  A
  A - Substitution de x et y par A et B dans FAB:
  A
  FAB[(FAB='x')/⊔DIMF]←'A' ⋄ FAB[(FAB='y')/⊔DIMF]←'B'
'Methodes: Euler [1], point-milieu [2]'
'prévision-correction [3], Runge-Kutta [4]' ⋄ METH←□
'Valeurs initiales x0 et y0 : ' ⋄ (x y)←(x0 y0)←□
'Nombre d'itérations et incrément dx: '
(NIT dx)←□ ⋄ X←Y←⊔I←0
  A
L1: I←I+1
⊔(METH=1)/'Y←Y,y←y+dx×⊔FX ⋄ x←x+dx'
⊔(METH=2)/'A←x+0.5×dx ⋄ B←y+0.5×dx×⊔FX'
⊔(METH=2)/'Y←Y,y←y+dx×⊔FAB ⋄ x←x+dx'
⊔(METH=3)/'A←x+dx ⋄ B←y+dx×fxy←⊔FX'
⊔(METH=3)/'Y←Y,y←y+0.5×dx×fxy+⊔FAB ⋄ x←A'
→(METH≠4)/L2 ⋄ A←x+0.5×dx ⋄ K1←dx×⊔FX
B←y+0.5×K1 ⋄ K2←dx×⊔FAB ⋄ B←y+0.5×K2 ⋄ K3←dx×⊔FAB
x←A-x+dx ⋄ B←y+K3 ⋄ K4←dx×⊔FAB
Y←Y,y←y+(K1+K4+2×K2+K3)÷6
  A
L2: X←X,x ⋄ →(I<NIT)/L1
  
```

La méthode d'Euler (1) est une simple itération de la formule des accroissements finis. Les méthodes du point-milieu (2) et de prévision-correction (3), représentent deux modes d'interpolation des valeurs de f(x,y) obtenues selon la méthode 1. La méthode de Runge-Kutta (4) correspond à l'approximation de la fonction f(x,y) par un développement limité en série de Taylor. La précision de ces méthodes peut être évaluée par comparaison avec les valeurs obtenues pour une équation différentielle dont la solution analytique est connue; elle est en général dans l'ordre $1 \ll 2 \sim 3 < 4$. La méthode d'Euler, peu précise, n'est citée ici que pour mémoire. La méthode 4, la plus précise, est deux à trois fois plus lente que 2 et 3, inconvénient mineur si l'on considère les performances actuelles des microordinateurs.

La fonction SYSEQUA2, de structure analogue à EQUADIF, s'applique à deux équations différentielles du premier ordre couplées ou à une équation différentielle du deuxième ordre écrite sous la forme $Y' = Z, Z' = f(x, y, z)$

```

SYSEQUA2
'Equation dy/dx = F(x,y,z) : ' ⋄ DIMF←ρAC←FXYZ←,⊔
sysequa2 ⋄ FAB←AC
''
'Equation dz/dx = G(x,y,z) : ' ⋄ DIMF←ρAC←GXZY←,⊔
sysequa2 ⋄ GABC←AC asysequa2 substitue A, B et C
à x, y et z dans FAB et GABC
'Méthodes : Point-milieu [1], prévision-correction [2], 'Runge-
  
```

```

Kutta [3]'
METH←□
''
'Valeurs initiales x0, y0, z0 : '
X←Y←Z←1I←0 ◇ (x y z)←(x0 y0 z0)←□
''
'Nombre d'itérations et incrément dx: '
(NIT dx)←□
L1: I←I+1
→(METH≠1)/L2 ◇ A←x+0.5×dx ◇ B←y+0.5×dx×ϕFXYZ
C←z+0.5×dx×ϕGXYZ ◇ Y←Y,y←y+dx×ϕFABC
Z←Z,z←z+dx×ϕGABC ◇ x←x+dx
    A
L2:→(METH≠2)/L3 ◇ 'A←x+dx ◇ B←y+dx×fxyz←ϕFXYZ
C←z+dx×gxyz←ϕGXYZ ◇ Y←Y,y←y+0.5×dx×fxyz+ϕFABC
Z←Z,z←z+0.5×dx×gxyz+ϕGABC ◇ x←A
    A
L3:→(METH≠3)/L4 ◇ A←x+0.5×dx ◇ K1←dx×ϕFXYZ
B←y+0.5×K1 ◇ M1←dx×ϕGXYZ
C←z+0.5×M1 ◇ K2←dx×ϕFABC ◇ M2←dx×ϕGABC
B←y+0.5×K2 ◇ C←z+0.5×M2 ◇ K3←dx×ϕFABC
M3←dx×ϕGABC ◇ x←A←x+dx ◇ B←y+K3
C←z+M3 ◇ K4←dx×ϕFABC ◇ M4←dx×ϕGABC
Y←Y,y←y+(K1+K4+2×K2+K3)÷6
Z←Z,z←z+(M1+M4+2×M2+M3)÷6
    A
L4: X←X,x ◇ →(I<NIT)/L1

```

Le même procédé de calcul peut être facilement étendu à un plus grand nombre d'équations.

EXEMPLES D'APPLICATIONS

Evolution thermique d'une réaction en chaîne.

La cinétique de la plupart des réactions en chaîne comme la combustion, la décomposition thermique, la polymérisation, dépend d'une manière critique de la différence entre les vitesses de dégagement et d'évacuation de la chaleur produite en milieu confiné. Ce problème a fait l'objet d'un nombre considérable de travaux dont l'un des plus importants est la réf. (3).

Prenons l'exemple d'une polymérisation où la terminaison des chaînes ne s'effectue que par consommation du monomère. La vitesse de formation du polymère est :

$$\frac{dx}{dt} = R_0 C_0 \left(1 - \frac{x}{C_0}\right) A_p \exp\left(-\frac{E_p}{RT}\right) \quad [1]$$

R_0 : concentration du catalyseur, C_0 : concentration initiale de monomère.

A_p et E_p : facteur préexponentiel et énergie d'activation de propagation.

L'équation d'évolution de la température est la somme de deux termes correspondant respectivement au dégagement et à l'évacuation de la chaleur produite par la réaction :

$$\frac{dT}{dt} = \frac{1}{c_s \rho} \left(Q_p \frac{dx}{dt} - \frac{\chi S}{V} (T - T_e) \right) \quad [2]$$

Q_p , c_p et : enthalpie de réaction, chaleur spécifique et densité de la solution.

T_e : température externe du réacteur de surface S et de volume V.

: coefficient d'échange thermique.

La figure 1 représente l'évolution de la température calculée pour un réacteur contenant une solution de styrène en cours de polymérisation. Le système d'équations [1, 2] qui n'a pas de solution analytique, est résolu avec la fonction SYSEQUA2 par la méthode de Runge-Kutta. La concentration initiale des réactifs, le rapport surface/volume du réacteur et la température externe de celui-ci ont une influence cruciale sur le régime thermique de la réaction. Cette figure montre par exemple que l'augmentation d'une dizaine de degré de

la température externe provoque la transition d'un régime thermique quasi-stationnaire ou à variation lente, à une régime transitoire qui se manifeste par une brusque élévation de la température interne suivie d'une décroissance rapide due à la consommation complète du monomère.

Les équations [1] et [2] sont d'un intérêt tout à fait général; elles s'appliquent avec quelques variantes à une réaction en chaîne comme la polymérisation, à l'inflammation d'une allumette, à la détonation d'un explosif, à la surchauffe due à la fermentation des graines dans un silo etc...

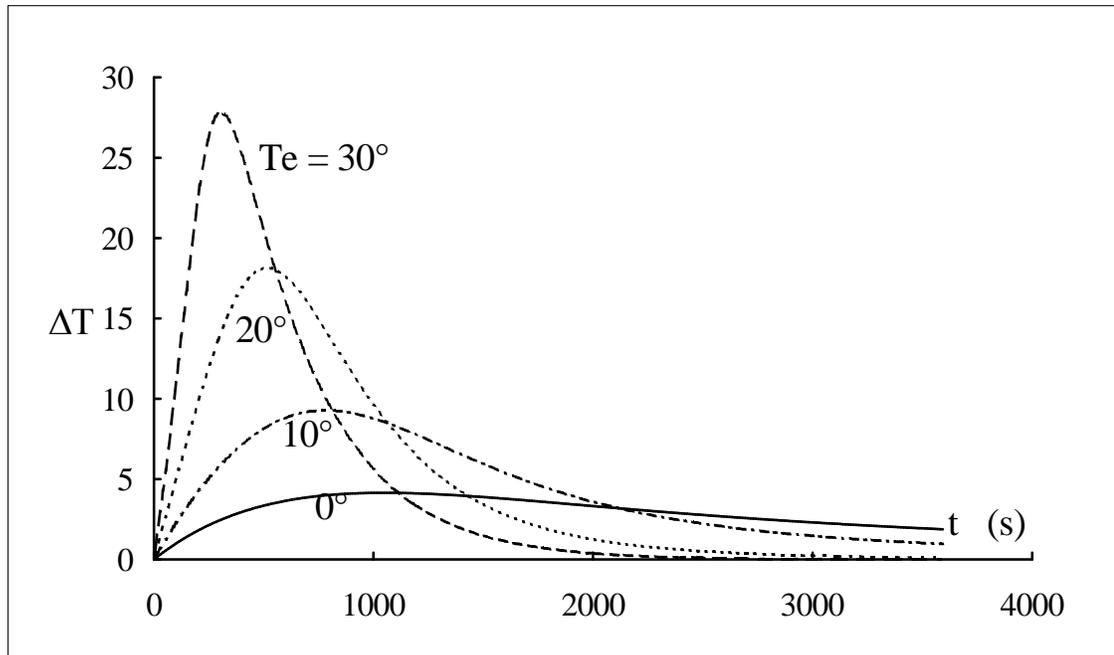


Figure 1. Différence entre les températures interne et externe du réacteur.

Paramètres : $C_0 \approx 0.9 \text{ mole l}^{-1}$, $R_0 / C_0 \approx 10^{-4}$

$Q_p = 3.6 \text{ kJ mole}^{-1}$, $\chi \approx 2.5 \times 10^{-2} \text{ kJ m}^{-1} \text{ s}^{-1} \text{ deg}^{-1}$,

$\frac{S}{V} \approx 7 \text{ m}^{-1}$, $E_p = 2.9 \text{ kJ mole}^{-1}$ et $A_p = 10^{10} \text{ l mole}^{-1} \text{ s}^{-1}$

Trajectoire d'un objet dans un champ gravitationnel.

L'exemple suivant concerne le problème à trois corps dont une version simplifiée est donnée dans la réf. (1). On considère deux astres en interaction gravitationnelle. L'astre le moins massif (II) est supposé parcourir une orbite quasi-circulaire à la vitesse angulaire $\omega = 2\pi / T$ autour de l'astre I. La trajectoire du troisième corps (III) dont la masse est négligeable devant celle de I et II, est exprimée dans un référentiel tournant à la vitesse autour du centre de gravité du système I-II pris comme origine. L'unité de temps est la période T, l'unité de longueur est la distance I-II et l'abscisse de l'astre II est choisie négative.

La trajectoire de l'objet III est obtenue par résolution de deux équations différentielles du second ordre couplées, que l'on peut remplacer par le système de quatre équations du premier ordre :

$$x' = u, \quad y' = v \quad [3a]$$

$$u' = -m_1 \frac{x - x_1}{((x - x_1)^2 + y^2)^{3/2}} - m_2 \frac{x - x_2}{((x - x_2)^2 + y^2)^{3/2}} + x + 2v \quad [3b]$$

$$v' = -m_1 \frac{y}{((x - x_1)^2 + y^2)^{3/2}} - m_2 \frac{y}{((x - x_2)^2 + y^2)^{3/2}} + y - 2u \quad [3c]$$

m_1 et m_2 sont les masses réduites des astres I et II ($m_1 + m_2 = 1$, $m_1 \geq m_2$) dont les coordonnées sont $x_1, y_1 = m_2, 0$ et $x_2, y_2 = -m_1, 0$. Les deux premiers termes des équations 3b et 3c correspondent à la loi de gravitation de Newton, les autres sont introduits pour tenir compte de la rotation du référentiel XY (voir réf. (1)).

Le système [3a,b,c] a été résolu par la méthode de Runge-Kutta au moyen de la fonction SYSEQUA4, analogue à SYSEQUA2, valable pour quatre équations du premier ordre. Les trajectoires représentées sur la figure 2 ont été calculées sur 1200 points avec un pas de $0.01 T$, soit douze révolutions de la Lune autour de la Terre. Pour le système Terre-Lune, la période T correspond à 27.3 jours et l'unité de longueur, à 384600 km. La vitesse initiale v_0 dans l'exemple de la figure 2 est donc de l'ordre de 600 km/h dans le référentiel tournant

La fonction NEWTON de Chaitin (4) permet la résolution numérique du problème à N corps dans toute sa généralité et ne nécessite donc pas les simplifications mentionnées plus haut. Que l'on utilise SYSEQUA4 ou NEWTON, on constate que les formes des trajectoires sont extraordinairement dépendantes des conditions initiales et donc difficilement prévisibles.

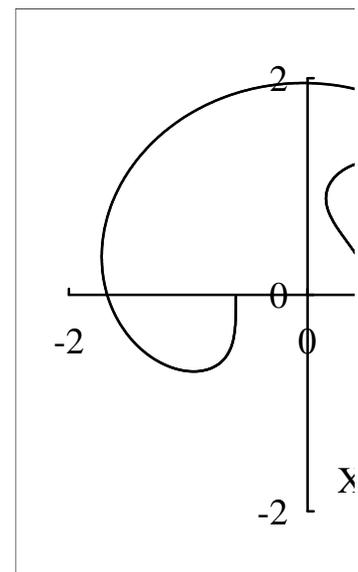
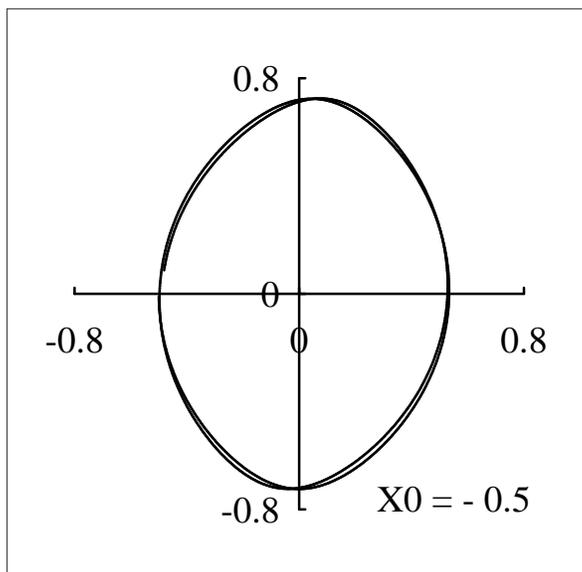
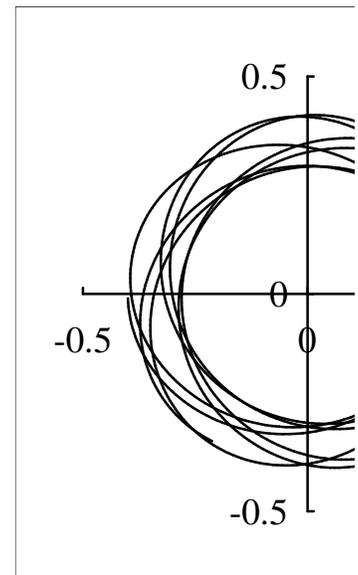
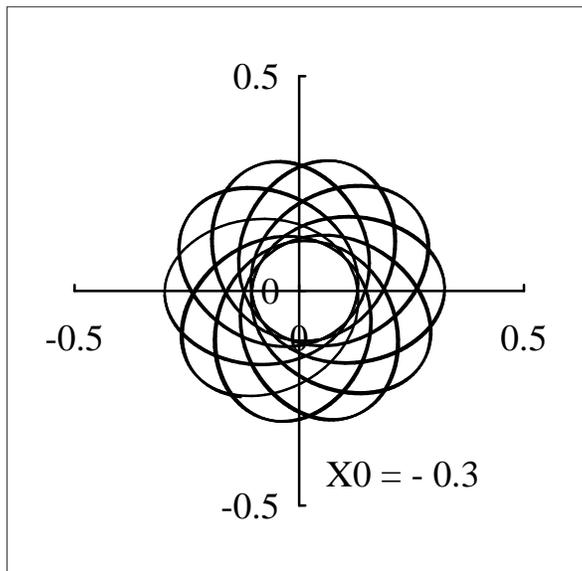


Figure 2. Trajectoire d'un objet de faible masse dans le système Terre ($m_1=0.98786$, $x_1= 0.01214$) - Lune ($m_2 = 0.01214$, $x_2= - 0.98786$) . Positions initiales : $y_0 = 0$, x_0 variable. Vitesses initiales : $u_0 = 0$, $v_0 = - 1$. Durée totale 12 T.

Comparaison des logiciels APL utilisés.

Les fonctions APL décrites dans cet article, comme la plupart de celles que j'utilise, existent en APL*PLUS II version 5.1 pour DOS et en deux applications Windows: APL*PLUS III version 1.2 et APL2win (IBM APL2 beta level 7). Les trajectoires de la figure 2 par exemple, ont été calculées avec ces logiciels en 10.9, 9.3 et 11.3 s respectivement, sur un PC Compaq 133 MHz avec processeur Pentium. Ces logiciels ont donc des performances à peu près équivalentes dans le cas d'une fonction opérant en boucle avec un grand nombre d'itérations. Pour ce qui est des fonctions ne comportant que des opérations sur des vecteurs et matrices réels, APL2win est environ deux fois plus rapide que les APL*PLUS II et III. Dans les opérations sur les complexes, en particulier pour les inversions de matrices, les vitesses relatives de calcul sont en moyenne dans l'ordre 3, 1.5 et 1 respectivement pour APL2win, APL*PLUS III et APL*PLUS II. Les opérations sur les variables complexes sont plus simples et plus rapides en APL2 IBM qu'en APL*PLUS du fait qu'elles sont considérées comme de même rang que les variables réelles équivalentes et font appel aux mêmes opérateurs, à l'exception du domino, remplacé par une fonction. Ainsi un vecteur à N composantes complexes est un vecteur de dimension N en APL2 IBM et un tableau de dimension 2,N en APL*PLUS; de même le produit scalaire de deux vecteurs complexes s'écrit $V+.W$ en APL2 IBM et $V CXMATTIMES W$ en APL*PLUS etc...

Un autre avantage d'APL2win sur les deux autres APL dont il est question ici, est l'existence de la zone (alias workspace) GRAPHPAK qui contient diverses fonctions d'utilisation facile, pour graphiques bi et tridimensionnels. Pour une impression de bonne qualité des graphiques, il est cependant nécessaire de transférer les tableaux numériques d'APLwin à EXCEL. Pour le moment je n'ai réussi ce transfert que par la séquence APL2win fichier ASCII sous DOS reformater en APL*PLUS III EXCEL. Je n'ai pas trouvé non plus en APL2win, l'équivalent des commandes)CMD ou CMD qui en APL*PLUS permettent de communiquer rapidement avec DOS en cours de session APL.

En APL*PLUS II seulement, CMD [instruction DOS] dans une fonction APL, permet d'exécuter un fichier *.exe et de récupérer automatiquement les tableaux calculés. APL*PLUS II a également l'avantage de permettre de disposer de zones plus grandes qu'en APL sous Windows. Ainsi pour 16 Mo de RAM, on dispose de zones de 14 Mo en APL*PLUS II alors qu'en APL*PLUS III et APL2win, il est préférable de limiter leur taille à 5 Mo pour ne pas compromettre la rapidité des calculs.

Il est donc avantageux d'utiliser APL2win pour les calculs longs que l'on rencontre souvent dans les problèmes d'optimisation, APL*PLUS III si l'on souhaite l'édition immédiate et de bonne qualité de données numériques et de graphiques sous EXCEL, APL*PLUS II pour des fonctions générant des variables de grande taille ou faisant appel à des fichiers DOS en cours d'exécution.

Références

1 - P. Hellings, *Astrophysics with a PC, an Introduction to Computational Astrophysics*. Willmann-Bell, Richmond, Virginia (USA) 1994. ISBN 943396-43-3.

2 - W.H. Press, S. A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes in Fortran, the Art of Scientific Computing*, seconde édition, chap. 16, Cambridge University Press, Cambridge 1992. ISBN 0 - 521 - 43064 - X.

3 - D.A. Frank-Kamenetzky, *Diffusion and Heat Exchange in Chemical Kinetics*, chap. 6 et 7, Princeton University Press, Princeton 1955.

4 - G.J. Chaitin, *An APL2 Gallery of Mathematical Physics. A course Outline*. Proceedings Japan 85 APL Symposium, pp 1-56, IBM Japan 1985.