TD Maple 2 : Suites et convergence

Michael Monerau

13 novembre 2010

1 Accélération de convergence

Nous allons voir des techniques qui visent à améliorer la vitesse de convergence de certaines familles de suites.

Ainsi, à partir des premiers termes d'une suite S_n, on va construire les premiers termes d'une suite T_n qui converge *beaucoup* plus vite que la première et vers la même limite.

Formellement, on se place donc dans le cas où on connaît une suite réelle $(S_n)_{n\in\mathbb{N}}$ qui converge vers une valeur (inconnue) S_∞.

L'objectif est alors de trouver une transformation $\Phi : \mathbb{R}^\mathbb{N} \longrightarrow \mathbb{R}^\mathbb{N}$ telle que si on pose $T_n = \Phi(S_n)$, on a $T_n - S_\infty = o(S_n - S_\infty)$.

1.1 Méthodes linéaires

1.1.1 Méthode d'Euler

On suppose dans cette section que la suite S_n a la forme suivante :

$$S_n = S_\infty + r^n\,\varphi(n)$$

avec $\frac{\varphi(n+1)}{\varphi(n)} \longrightarrow 1$ et $0 < r < 1$ connu.

On pose alors la suite :

$$T_n = \frac{S_{n+1} - rS_n}{1-r}$$

Question 1.1. Preuve
Montrer (sur une feuille) qu'on a alors

$$T_n - S_\infty = o(S_n - S_\infty)$$

Question 1.2. Exemple Supposons qu'on ait la suite $S_n = S_\infty + a_1\,0.95^n + \frac{a_2}{\sqrt{n}}$, avec S_∞, a_1 et a_2 inconnus. On veut accéder à la valeur de S_∞, mais on ne dispose que de 10 valeurs de la suite, qui ne sont pas particulièrement parlantes :

n	1	2	3	4	5
S_n	13.85	13.41	13.14	12.94	12.77
n	6	7	8	9	10
S_n	12.61	12.47	12.34	12.22	12.11

En utilisant le procédé décrit ci-dessus, conjecturer la limite de la suite.

Faire quelques essais avec des suites que vous générez vous-même.

1.1.2 Convergence plus lente

On n'a pas toujours la chance d'avoir une convergence géométrique. Cependant, si on prend plus de termes, on peut avoir une technique équivalente.

Dans le cas important où :

$$S_n = S_\infty + \frac{\alpha}{n} + \frac{\beta}{n^2} + \ldots$$

On pose $\widetilde{S_n} = S_{2^n}$ et on applique la méthode d'Euler à $\widetilde{S_n}$ avec $r = 1/2$.

1.1.3 Méthode de Richardson

On se place maintenant dans le cas où :

$$S_n = S_\infty + c_1\,{\rho_1}^n + \ldots + c_k\,{\rho_k}^n$$

avec $1 > \rho_1 > \rho_2 > \ldots > \rho_k > 0$.

Le principe est ici d'éliminer les ρ_i les uns après les autres. Pour ceci, on suit un procédé de construction itératif :

$$\begin{aligned} T_n^{(0)} &= S_n \\ &\vdots \\ T_n^{(i)} &= \frac{T_{n+1}^{(i-1)} - \rho_i\,T_n^{(i-1)}}{1-\rho_i} \\ &\vdots \\ T_n^{(k)} &= \frac{T_{n+1}^{(k-1)} - \rho_k\,T_n^{(k-1)}}{1-\rho_k} \end{aligned}$$

Le terme $T_n^{(k)}$ est appelé *transformée de Richardson* de S_n. $T_n^{(k)}$ converge (en n) vers S_∞ plus vite que S_n puisque les termes résiduels ont été amortis un à un. Plus la valeur de k est élevée, plus l'accélération est rapide (on a amorti plus de termes).

Question 1.3. Richardson

Écrire une procédure prenant en entrée la suite S_n, la liste des ρ_i et retournant une matrice avec les $T_n^{(k)}$.

Indication : On pourra consulter l'aide sur `Array` (avec majuscule) et regarder les exemples d'utilisation en bas de la page.

Question 1.4. Exemple 1

Générer une suite correspondant à ce schéma avec des valeurs que vous choisirez, et tester votre fonction.

Question 1.5. Exemple 2

On suppose qu'on dispose d'une suite S_n ayant pour premières valeurs :

n	1	2	3	4	5
	12.091	8.0443	7.1401	6.6946	6.4171
n	6	7	8	9	10
	6.2238	6.0799	5.9680	5.8781	5.8040
n	11	12	13	14	15
	5.7419	5.6890	5.6433	5.6034	5.5683
n	16	17	18	19	20
	5.5371	5.5093	5.4842	5.4615	5.4408
n	21	22	23	24	25
	5.4220	5.4047	5.3888	5.3741	5.3605
n	26	27	28	29	30
	5.3478	5.3360	5.3250	5.3147	5.3051
n	31	32	33	34	35
	5.2960	5.2874	5.2793	5.2717	5.2645

On sait d'autre part qu'elle est de la forme :

$$S_n = S_\infty + \ln\left(1 + \frac{a_1}{n} + \frac{a_2}{n^2} + \frac{a_3}{n^3} \dots\right)$$

Conjecturer la valeur de S_∞.

1.2 Méthode non-linéaires

1.2.1 Méthode Δ^2 d'Aitken (1925)

Ici, on suppose toujours que

$$S_n = S_\infty + r^n\,\varphi(n)$$

avec $\frac{\varphi(n+1)}{\varphi(n)} \longrightarrow 1$ et $r < 1$. Mais cette fois, r n'est pas supposé connu et on ne peut donc pas utiliser la méthode d'Euler telle quelle.

On pose :

$$\begin{aligned} \Delta S_n &= S_{n+1} - S_n \\ &= r^n(r\,\varphi(n+1) - \varphi(n)) \end{aligned}$$

Question 1.6. Théorique

Montrer que $\frac{\Delta S_{n+1}}{\Delta S_n} \longrightarrow r$.

On a donc en quelque sorte retrouvé l'accès à l'inconnue r.

La *transformée d'Aitken* consiste naturellement à faire la méthode d'Euler, mais en remplaçant r par sa valeur approchée donnée par $\frac{\Delta S_{n+1}}{\Delta S_n}$:

$$\begin{aligned} T_n &= \frac{S_{n+1} - \frac{\Delta S_{n+1}}{\Delta S_n} S_n}{1 - \frac{\Delta S_{n+1}}{\Delta S_n}} \\ &= S_n - \frac{(\Delta S_n)^2}{\Delta(\Delta S_n)} \\ &= S_n - \frac{(\Delta S_n)^2}{\Delta^2 S_n} \end{aligned}$$

Question 1.7. Méthode d'Aitken

Écrire une procédure prenant en entrée une liste des premiers éléments d'une suite (qu'on suppose vérifier les hypothèses), et qui renvoie la liste des premiers éléments de la suite accélérée T_n.

Tester sur une suite créée pour l'occasion. Comparer la précision avec la méthode d'Euler.

1.2.2 Méthode de Shanks (1949)

La méthode de Shanks est à la méthode d'Aitken ce que la méthode de Richardson est à la méthode d'Euler : on cherche à enlever plusieurs termes perturbateurs, mais cette fois-ci on ne connaît pas les ρ_i.

Elle est un petit peu délicate dans la mesure où il faut définir le déterminant suivant, appelé déterminant de Hankel :

$$H_n^{(k)}(u_n) = \begin{vmatrix} u_n & \Delta u_n & \dots & \Delta^{k-1} u_n \\ \Delta u_n & \Delta^2 u_n & \dots & \Delta^k u_n \\ \vdots & \vdots & & \vdots \\ \Delta^{k-1} u_n & \Delta^k u_n & \dots & \Delta^{2k-2} u_n \end{vmatrix}$$

Puis on l'utilise pour calculer les termes :

$$T_n^{(k)} = \frac{H_n^{(k+1)}(S_n)}{H_n^{(k)}(\Delta^2 S_n)}$$

Et, comme ci-dessus, la vitesse de convergence augmente toujours avec k. En pratique, on prend donc k le plus grand possible selon le nombre de termes numériques dont on dispose au départ sur la suite à accélérer.

Question 1.8. Shanks

Implémenter l'algorithme de Shanks et le tester sur une suite que vous créerez.

1.2.3 Algorithme ε de Wynn

En pratique, il est coûteux de calculer autant de déterminants. On préfère donc utiliser l'algorithme de Wynn qui donne accès aux mêmes valeurs mais de manière beaucoup plus simple.

On pose :

$$\begin{aligned} \varepsilon_{-1}^{(n)} &= 0 \\ \varepsilon_{0}^{(n)} &= S_n \\ \varepsilon_{k+1}^{(n)} &= \varepsilon_{k-1}^{(n+1)} + \frac{1}{\varepsilon_{k}^{(n+1)} - \varepsilon_{k}^{(n)}} \end{aligned}$$

On a alors le :

Théorème 1.1. *$\varepsilon_{2k}^{(n)}$ est le $T_n^{(k)}$ de la méthode de Shanks.*

Question 1.9. Shanks + Wynn

Implémenter cette nouvelle version de l'algorithme de Shanks, puis la tester sur une suite.

2 Une suite implicite

Nous allons voir dans cette partie comment utiliser Maple pour réaliser le développemente asymptotique d'une suite implicite.

L'idée générale est de trouver une forme "facile" de la suite, du type $u_n = \ell + f(n)$ où f est inconnue. On réinjecte alors cette expression dans l'équation de définition de la suite implicite pour une équation asymptotique sur $f(n)$ et ainsi en trouver ses premiers termes. On réinjecte alors à son tour réinjecter cette expression, et ainsi de suite. À la fin, on obtient un développement asymptotique de u_n.

Considérons ici la suite réelle positive $(u_n)_{n\in\mathbb{N}}$ définie implicitement par

$$\forall n \geq 2, \quad u_n^n - nu_n = 1 \tag{1}$$

Question 2.1. Définition

Cette suite est-elle correctement définie ?

Question 2.2. Limite

Tracer la fonction $x \mapsto x^n - nx - 1$ pour plusieurs valeurs de n (sur un même graphe). Que dire du comportement à l'infini de u_n ?

Indication : Rappel de la dernière fois : pour tracer tout sur un graphe, utiliser `plot({seq(.....)})`.

Montrer à la main que u_n converge effectivement, et trouver sa limite ℓ.

Question 2.3. Premier terme

En écrivant $u_n = \ell + \varepsilon(n)$ où $\varepsilon(n) = o(1)$, injecter dans l'équation (1) et trouver le premier élément du développement asymptotique de ε.

Question 2.4. Développement asymptotique

On constate que $u_n = \exp(\frac{1}{n}\ln(1 + nu_n))$.

En demandant à Maple de développer le terme de droite à un ordre de plus, trouver un terme de plus du développement asymptotique.

Indication : On pourra utiliser la fonction `series`.

En itérant le procédé, écrire une procédure qui réitère ce processus et trouve un développement asymptotique de u_n à un ordre arbitraire passé en argument.