

Nous commencerons par quelques rappels sur les calculs de sommes, de produits, de limites, de développements de Taylor et de développements asymptotiques au moyen de Maple. Ces différentes techniques seront mises en œuvre pour l'étude de la convergence et, le cas échéant, le calcul de la somme d'une série donnée. Nous exposerons dans un dernier paragraphe plusieurs méthodes d'accélération de la convergence.

1	Sommes et produits	2
1.1	Les opérateurs sum et Sum	2
1.2	L'opérateur add	2
1.3	Les opérateurs product et Product	3
1.4	L'opérateur mul	3
2	Limites et développements	4
2.1	Calculs de limites	4
2.2	Développements de Taylor	5
2.3	Développements asymptotiques	5
2.4	Extraction de la partie principale	6
2.5	Obtention d'un équivalent	7
3	Suites et séries	7
3.1	Comportement asymptotique d'une suite ...	7
3.2	Suites récurrentes $u_{n+1} = f(u_n)$	8
3.3	Calcul de la somme d'une série convergente .	8
4	Accélération de la convergence	9
4.1	Typologie de la convergence	9
4.2	La méthode d'Euler	10
4.3	La méthode de Richardson	11
4.4	La méthode d'Aitken	13
4.5	Méthode de Shanks	14

1. Sommes et produits

1.1. Les opérateurs sum et Sum

L'utilisateur dispose de deux commandes pour le calcul des sommes ; l'opérateur *effectif* **sum** et l'opérateur *inerte* **Sum**. Pour calculer

$$\sum_{i=n}^m e_i$$

où e_i est une expression dépendant de la *variable libre*¹ i , on retiendra la syntaxe suivante :

Calcul formel d'une somme

1. opérateur *effectif* : **sum**(e_i , $i = n .. m$)

2. opérateur *inerte* : **Sum**(e_i , $i = n .. m$)

Le premier affichera le résultat du calcul alors que le second se contentera d'un affichage au moyen du symbole Σ ; on pourra cependant forcer le logiciel à effectuer le calcul au moyen de la commande **value**. L'utilisation de **Sum** relève donc surtout de l'esthétique ; d'une manière générale, les commandes inertes (voir **Diff**, **Int**, etc.) permettent la réalisation de documents plus lisibles, faisant apparaître les symboles mathématiques usuels (dérivée, intégrales, etc.).

```
> sum(i^3,i=1..n);
```

$$\frac{1}{4}(n+1)^4 - \frac{1}{2}(n+1)^3 + \frac{1}{4}(n+1)^2$$

```
> Sum(i^3,i=1..n)=value(Sum(i^3,i=1..n));factor(%);
```

$$\sum_{i=1}^n i^3 = \frac{1}{4}(n+1)^4 - \frac{1}{2}(n+1)^3 + \frac{1}{4}(n+1)^2$$

$$\frac{n^2(n+1)^2}{4}$$

1.2. L'opérateur add

Une autre solution est d'utiliser l'opérateur **add**. Ce dernier ne peut cependant pas effectuer de calculs formels. En revanche, l'indice de sommation pourra décrire une liste donnée d'expression.

1. Ce point est essentiel sous peine d'un message d'erreur !

```
> add(i^3,i=1..6);
```

$$441$$

```
> L:=[x,x^2,x-1,-x+1]: add(i,i=L), add(i,i in L);
```

$$x + x^2, \quad x + x^2$$

1.3. Les opérateurs **product** et **Product**

Comme pour les sommes, le logiciel dispose d'un opérateur inerte et d'un opérateur effectif pour le calcul des produits d'expressions e_i de la variable libre i :

$$\prod_{i=n}^m e_i.$$

La syntaxe de **product** est identique à celle de **sum**.

— Calcul formel d'un produit —

1. *opérateur effectif* : **product**(e_i , $i = n .. m$) 2. *opérateur inerte* : **Product**(e_i , $i = n .. m$)

```
> Product(i/(i+1),i=1..n): %=simplify(value(%));
```

$$\prod_{i=1}^n \frac{i}{i+1} = \frac{1}{n+1}$$

1.4. L'opérateur **mul**

Cet opérateur est à **Product** ce que **Add** est à **sum** : une version n'autorisant pas le calcul formel mais admettant une syntaxe parfois utile lorsque les termes du produits sont rangés dans une liste.

```
> mul((i+1)/i,i=1..78): %=simplify(value(%));
```

$$\prod_{i=1}^{78} \frac{i+1}{i} = 79.$$

2. Limites et développements

2.1. Calculs de limites

Le logiciel dispose d'un opérateur inerte² (**Limite**) et effectif (**limit**) de calcul des limites. Pour calculer

$$\lim_{x \rightarrow a} f(x)$$

où $f(x)$ est une expression, on suivra la syntaxe décrite ci-dessous.

Calcul d'une limite par la commande **limite**

limit(f(x), x=a, right ou left)

le troisième argument (pour une limite à droite ou à gauche) étant optionnel.

```
> Limit(sin(x)/x,x=0): %=value(%);
```

$$\lim_{x \rightarrow 0} \frac{\sin(x)}{x} = 1.$$

On écrira $a = \pm\text{infinity}$ pour calculer des limites en $\pm\infty$.

```
> Limit(exp(x)/x,x=infinity): %=value(%);
```

$$\lim_{x \rightarrow +\infty} \frac{e^x}{x} = \infty$$

Le logiciel renvoie le message *undefined* lorsqu'il manque de données pour calculer ce qui lui est demandé. C'est par exemple le cas lorsque la limite n'existe pas car les limites à gauche et à droite sont distinctes. L'utilisateur pourra spécifier *right* ou *left* en option dans la commande **limit** pour calculer respectivement des limites à droite et à gauche.

```
> Limit(exp(1/x),x=0): %=value(%);
```

$$\lim_{x \rightarrow 0} e^{\frac{1}{x}} = \text{undefined}$$

```
> Limit(exp(1/x),x=0, right): %=value(%);
```

$$\lim_{x \rightarrow 0^+} e^{\frac{1}{x}} = \infty$$

2. Voir page 2 pour la distinction entre opérateur inerte et opérateur effectif.

2.2. Développements de Taylor

MAPLE calcule des développements de Taylor dont les restes sont des O (« *grand o* ») ; par exemple

$$\sin(x) = x + O(x^3)$$

Calcul d'un développement limité de $f(x)$ en un point a par la commande **taylor**

taylor($f(x)$, $x = a$, n)

Il faut bien comprendre la signification de n ci-dessus : il ne s'agit pas de l'ordre du développement calculé mais l'ordre auquel le logiciel développera les fonctions composant l'expression $f(x)$. Le lecteur méditera les exemples suivants :

```
> taylor(sin(x), x=0, 4);
```

$$x - \frac{1}{6}x^3 + O(x^4)$$

```
> taylor(sin(x)/ln(1+x), x=0, 4);
```

$$1 + \frac{1}{2}x - \frac{1}{4}x^2 + O(x^3)$$

2.3. Développements asymptotiques

Plus généralement, l'utilisateur obtiendra des développements asymptotiques³ au moyen de la commande **series**.

Calcul d'un développement asymptotique de $f(x)$ en un point a par la commande **series**

series($f(x)$, $x = a$, n)

La remarque précédente sur la signification de n est encore valable pour la commande **series**. Ce troisième argument est cependant optionnel. Maple dispose d'une variable d'environnement **Order** qui par défaut vaut 6.

3. Ce type de développement généralise les *DL*. Le lecteur est renvoyé à son cours de Mathématiques.

```
> series(1/sin(x),x=0,4);
```

$$x^{-1} + \frac{1}{6}x + O(x^2)$$

```
> Order:=2:series(sin(x),x=0);
```

$$x + O(x^2)$$

La commande **taylor** n'est qu'une spécialisation de la commande **series**. On pourra donc également obtenir des *DL* au moyen de **series**.

```
> series(sin(x),x=0,4);
```

$$x - \frac{1}{6}x^3 + O(x^4)$$

On pourra aussi effectuer des développements en $+\infty$...

```
> series(1/(1+x),x=infinity,4);
```

$$\frac{1}{x} - \frac{1}{x^2} + \frac{1}{x^3} + O\left(\frac{1}{x^4}\right)$$

2.4. Extraction de la partie principale

Le résultat renvoyé par la commande **series** n'est pas utilisable tel quel pour un tracé ou pour effectuer des calculs algébriques (produits ou sommes de DL) car il n'est pas du type *polynom* (au sens large). On pourra cependant *extraire* la partie principale d'un développement en utilisant la commande **convert**.

Extraction de la partie principale d'un développement

convert (résultat d'une commande series, polynom)

```
> series(60*ln(1+x), x = 0); convert(%,polynom);
```

$$60x - 30x^2 + 20x^3 - 15x^4 + 12x^5 + O(x^6)$$

$$60x - 30x^2 + 20x^3 - 15x^4 + 12x^5$$

2.5. Obtention d'un équivalent

On obtiendra un équivalent d'une expression au voisinage d'un point en utilisant la command **leadterm**.

Détermination d'un équivalent de x en a par **leadterm**

series(leadterm($f(x)$), $x = a$)

```
> series(leadterm(sqrt(1+sqrt(x))),x=infinity);
```

$$\frac{1}{\left(\frac{1}{x}\right)^{\frac{1}{4}}}$$

3. Suites et séries

Une suite numérique étant une fonction de $\mathbb{N}$ dans $\mathbb{C}$, on pourra définir une suite $u = (u_n)_{n \in \mathbb{N}}$ par l'expression de son terme général u_n ou en tant que fonction (que l'on notera u).

3.1. Comportement asymptotique d'une suite

L'ensemble des résultats précédents (développements asymptotiques, calcul des limites par **series** et **limit**) reste valable dans ce cadre. Les opérateurs **sum** et **product** permettront de définir des suites, des sommes partielles, etc. En voici quelques exemples...

```
> u:=unapply(sin(n)/n,n): Limit(u(n),n=infinity): %=value(%);
```

$$\lim_{n \rightarrow +\infty} \frac{\sin(n)}{n} = 0.$$

```
> series(sum(1/k,k=1..n),n=infinity);
```

$$\ln(n) + \gamma + \frac{1}{2n} - \frac{1}{12n^2} + \frac{1}{120n^4} + O\left(\frac{1}{n^6}\right)$$

Exercice 1.

Pour tout entier naturel $n \geq 2$, on pose

$$u_n = \cos\left(n^2 \cdot \pi \cdot \ln\left(1 - \frac{1}{n}\right)\right)$$

Etudier la convergence de $\sum u_n$ en utilisant Maple pour les calculs intermédiaires.

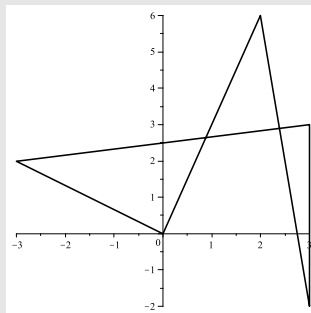
3.2. Suites récurrentes $u_{n+1} = f(u_n)$

Tracé d'une ligne brisée

plot(L, options de tracé)

où L est une liste de points donnés sous la forme d'une liste à deux éléments $[x, y]$, respectivement abscisse et ordonnée du point.

```
> plot([ [0,0], [2,6], [3,-2], [3,3], [-3,2], [0,0] ], thickness=2, color=black);
```



Exercice 2.

Soit $(u_n)_{n \geq 0}$ la suite définie par

$$\begin{cases} u_0 = \frac{\pi}{2} \\ \forall n \in \mathbb{N}, u_{n+1} = \sin(u_n) \end{cases}$$

1. Ecrire une procédure **Escalier** d'argument n construisant l'escalier d'itération à l'ordre n de la suite $(u_n)_{n \geq 0}$.

2. On s'intéresse dans cette question à la convergence de la série $\sum u_n$.

2.a. Trouver deux réels $\alpha < 0$ et λ tels que

$$\sin^\alpha(x) - x^\alpha = \lambda + o(1)$$

2.b. En déduire un équivalent de u_n puis la nature de la série $\sum u_n$.

3.3. Calcul de la somme d'une série convergente

On calculera la somme d'une série

$$\sum_{n \geq 0} u_n$$

au moyen de la commande **sum** (décrite dans le premier paragraphe de ce document) qui accepte une borne infinie.

Calcul de la somme d'une série numérique

`sum(un, n = 0..infinity)`

```
> Sum(x^k, k=0..infinity): %=value(%);
```

$$\sum_{k=0}^{+\infty} x^k = -\frac{1}{x-1}$$

4. Accélération de la convergence

Soit $(x_n)_{n \geq 0}$ une suite qui converge vers une valeur $\bar{x}$. On cherche une transformation de la suite $(x_n)_{n \geq 0}$ en une suite $(y_n)_{n \geq 0}$ telle que $(y_n)_{n \geq 0}$ tende aussi vers $\bar{x}$, mais beaucoup plus rapidement. Formellement, on cherche une transformation $T: \mathbb{R}^{\mathbb{N}} \rightarrow \mathbb{R}^{\mathbb{N}}$ telle que, si on pose $y_n = T((x_n)_{n \geq 0})$, alors on ait

$$y_n - \bar{x} = o(x_n - \bar{x})$$

4.1. Typologie de la convergence

Voici quelques types courants de convergence.

Définition 1. Convergence géométrique

On dit qu'une suite $(x_n)_{n \geq 0}$ converge (au moins) géométriquement vers $\bar{x}$ si

$$x_n - \bar{x} = O(k^n) \quad \text{où} \quad 0 < k < 1$$

On déduit le résultat suivant de la preuve du critère de D'Alembert pour les séries numériques :

Proposition 2. (Condition suffisante de convergence géométrique)

Si $(u_n)_{n \geq 0}$ est une suite à valeurs dans $\mathbb{R}_+^*$ telle que

$$\lim_{n \rightarrow +\infty} \frac{u_{n+1}}{u_n} = k \quad \text{où} \quad 0 < k < 1$$

alors $(u_n)_{n \geq 0}$ converge vers 0 géométriquement.

Définition 3. Convergence quadratique

On dit qu'une suite $(x_n)_{n \geq 0}$ converge (au moins) quadratiquement vers $\bar{x}$ si

$$u_n - \bar{x} = O(k^{2^n}) \quad \text{avec} \quad 0 < k < 1$$

Théorème 4. (Condition suffisante de convergence quadratique)

Soit $(u_n)_{n \geq 0}$ une suite de nombres strictement positifs de limite 0. On suppose que la suite

$$\left(\frac{u_{n+1}}{u_n^2} \right)_{n \geq 0}$$

a une limite finie (ou simplement qu'elle est bornée). Alors, la convergence de $(u_n)_{n \geq 0}$ est quadratique.

Il n'est pas facile de donner une définition de la convergence lente...

Définition 5. (Convergence lente)

Une suite $(u_n)_{n \geq 0}$ (positive) converge lentement vers 0 si, à partir d'un certain rang,

$$\frac{A}{n^\alpha} \leq u_n \quad \text{avec} \quad A, \alpha > 0$$

Exercice 3.

Convergence vers un point fixe attractif

On se propose d'éclaircir quelques points du paragraphe 4.1.

1. Démontrer le théorème 4.

2. Soient $f \in \mathcal{C}^1(I, I)$ où I est un segment non trivial de $\mathbb{R}$ et $(u_n)_{n \geq 0}$ définie par

$$\begin{cases} u_0 \in I \\ u_{n+1} = f(u_n) \end{cases}$$

pour tout entier naturel n .

2.a. Etablir que f admet au moins un point fixe α dans I .

2.b. On suppose que α est un point fixe attractif, i.e. $|f'(\alpha)| < 1$. Montrer que, si u_0 est suffisamment proche de α , $(u_n)_{n \geq 0}$ converge géométriquement vers α .

2.c. On suppose que f est classe $\mathcal{C}^2$ et que $f'(\alpha) = 0$. Etablir que, si u_0 est suffisamment proche de α , $(u_n)_{n \geq 0}$ converge quadratiquement vers α .

2.d. Lorsque $|f'(\alpha)| = 1$, montrer par un contre-exemple que la convergence (lorsqu'elle a lieu !) peut n'être que lente.

4.2. La méthode d'Euler

Les premières tentatives d'accélération de convergence remontent à l'époque classique. Euler proposa une méthode qui s'applique aux suites vérifiant

$$x_n = \bar{x} + \lambda \cdot \rho^n + o(\rho^n) \quad \text{avec} \quad |\rho| < 1 \quad \text{et} \quad \lambda \in \mathbb{R}$$

La convergence d'une telle suite est donc au moins géométrique. On pose

$$y_n = \frac{x_{n+1} - \rho \cdot x_n}{1 - \rho}$$

Proposition 6. (Euler) Avec les notations précédentes, $y_n - \bar{x} = o(x_n - \bar{x})$.

On remarquera que cette méthode nécessite la connaissance du nombre réel ρ .

Exemple 7. Cas de $x_n = \bar{x} + \lambda_1 \cdot (0.95)^n + \lambda_2 \cdot n^{-1} + o(n^{-3/2})$

On suppose que λ_1 et λ_2 sont inconnus. On veut accéder à la valeur de $\bar{x}$, mais on ne dispose que de 10 valeurs de la suite, qui ne sont pas particulièrement parlantes :

n	1	2	3	4	5	6	7	8	9	10	11	12
x_n	3.90	4.555	4.604	4.566	4.508	4.442	4.377	4.311	4.248	4.187	4.130	4.074

Une accélération de convergence par la méthode d'Euler, avec $\rho = 0.95$, donne ici le tableau de valeurs suivant :

n	1	2	3	4	5	6	7	8	9	10	11
y_n	17.00	5.54	3.84	3.41	3.19	3.14	3.06	3.05	3.03	3.05	3.01

On peut donc conjecturer que $\bar{x} = 3$ (et en l'occurrence, c'est bien la valeur avec laquelle la suite a été générée au départ).

Exercice 4.

Preuve de la méthode d'Euler et application.

1. Démontrer le théorème 6.

2. Soit $(u_n)_{n \geq 1}$ une suite vérifiant

$$u_n = \bar{x} + \frac{\lambda}{n^\alpha} + o\left(\frac{1}{n^\alpha}\right) \quad \text{avec} \quad \alpha > 0 \quad \text{et} \quad \lambda \in \mathbb{R}^*$$

2.a. Quel est le type de convergence de $(u_n)_{n \geq 1}$?

2.b. On pose $x_n = u_{2^n}$ pour tout $n \in \mathbb{N}$. Quel est le type de convergence de $(x_n)_{n \geq 0}$?

2.c. En déduire comment accélérer la convergence de $(u_n)_{n \geq 0}$.

4.3. La méthode de Richardson

Itération de la méthode d'Euler dans le cas où

$$x_n = \bar{x} + \lambda_1 \cdot \rho_1^n + \dots + \lambda_k \cdot \rho_k^n + o(\rho_k^n) \quad \text{avec} \quad 1 > |\rho_1| > \dots > |\rho_k| > 0$$

où les λ_i sont non nuls. Le principe est ici d'éliminer les ρ_i les uns après les autres par la méthode d'Euler. Pour ceci, on suit un procédé de construction itératif :

$$y_n^{(0)} = x_n, \quad y_n^{(i)} = \frac{y_{n+1}^{(i-1)} - \rho_i \cdot y_n^{(i-1)}}{1 - \rho_i}, \quad \dots, \quad y_n^{(k)} = \frac{y_{n+1}^{(k-1)} - \rho_k \cdot y_n^{(k-1)}}{1 - \rho_k}$$

Le terme $y_n^{(k)}$ est appelé transformée de Richardson de x_n . La suite $(y_n^{(k)})_{n \geq 0}$ converge vers $\bar{x}$ plus vite que $(x_n)_{n \geq 0}$ 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).

Exemple 8. Accélération d'une convergence lente

On suppose qu'on a une suite de la forme

$$u_n = \bar{x} + \frac{\lambda_1}{n} + \cdots + \frac{\lambda_k}{n^k} + o\left(\frac{1}{n^k}\right)$$

On est ici dans le cas d'une convergence lente : on considère la suite $x_n = u_{2^n}$. On a

$$x_n = \bar{x} + \lambda_1 \cdot (2^{-1})^n + \cdots + \lambda_k \cdot (2^{-k})^n + o\left((2^{-k})^n\right)$$

On peut donc appliquer la méthode de Richardson afin d'accélérer la convergence.

Voici une application numérique de cette méthode.

Exemple 9. Accélération de la convergence de $(1 + \frac{1}{n})^n$ vers e

Cette convergence est lente. Pour le voir, on écrit

$$u_n = \exp\left(n \cdot \ln\left(1 + \frac{1}{n}\right)\right)$$

et on utilise les développements limités du logarithme et de l'exponentielle au voisinage de 0 :

$$u_n = e \cdot \left(1 - \frac{1}{2n} + \frac{11}{24n^2} - \frac{21}{48n^3}\right) + o\left(\frac{1}{n^3}\right)$$

On applique la méthode de Richardson à la suite $x_n = u_{2^n}$:

$$x_n = e \cdot \left(1 - \frac{1}{2} \cdot (2^{-1})^n + \frac{11}{24} \cdot (2^{-2})^n - \frac{21}{48} \cdot (2^{-3})^n\right) + o\left((2^{-3})^n\right)$$

Les coefficients ρ_i qui interviennent étant successivement $\rho_1 = 2^{-1}$, $\rho_2 = 2^{-2}$ et $\rho_3 = 2^{-3}$ les trois suites à considérer sont

$$y_n^{(1)} = 2 \cdot x_{n+1} - x_n, \quad y_n^{(2)} = \frac{4 \cdot y_{n+1}^{(1)} - y_n^{(1)}}{3}, \quad y_n^{(3)} = \frac{8 \cdot y_{n+1}^{(2)} - y_n^{(2)}}{7}$$

On obtient les valeurs suivantes en calculant jusqu'à $y_5^{(0)}$:

$$y_5^{(0)} = x_5 = 2.676990129, \quad y_4^{(1)} = 2.716051761, \quad y_3^{(2)} = 2.718044855, \quad y_2^{(3)} = 2.718235685,$$

(rappelons que l'on a $e \approx 2.718281828$, de sorte que $y_2^{(3)}$ a 4 décimales exactes).

Exercice 5.

Application au calcul approché de la constante d'Euler

Pour tout entier naturel n non nul, on pose

$$H_n = \sum_{k=1}^n \frac{1}{k} \quad \text{et} \quad x_n = H_n - \ln(n)$$

1. Montrer que $(x_n)_{n \geq 1}$ converge vers un réel γ (appelé *constante d'Euler*).

2. Trouver un développement asymptotique de x_n en $o(1/n^4)$ au moyen de Maple. Comment justifier l'existence et la forme ce développement ?
3. Quel est le type de convergence de $(x_n)_{n \geq 1}$?
4. Appliquer la méthode de Richardson à $(x_n)_{n \geq 1}$. Quel est le premier terme de $(y_n^{(3)})_{n \geq 1}$ ayant les 5 premières décimales de γ ?

4.4. La méthode d'Aitken

C'est une variante de la méthode précédente qui permet de faire le calcul lorsqu'on sait qu'on a un développement asymptotique de la forme

$$x_n = \bar{x} + \lambda \cdot \rho^n + o(\rho^n) \quad \text{avec} \quad 0 < |\rho| < 1 \quad \text{et} \quad \lambda \in \mathbb{R}^*$$

mais dans lequel on ne connaît pas explicitement les quantités ρ et λ . L'astuce est de trouver ρ quand même (ou presque) avec la remarque suivante : x_n, x_{n+1}, x_{n-1} sont respectivement de l'ordre de $\lambda \cdot \rho^n, \lambda \cdot \rho^{n+1}, \lambda \cdot \rho^{n-1}$ et donc ρ n'est pas très différent de

$$c_n = \frac{x_{n+1} - x_n}{x_n - x_{n-1}}$$

(l'intérêt de considérer les différences $x_{n+1} - x_n$ et $x_n - x_{n-1}$ est de faire disparaître la limite $\bar{x}$ qui est évidemment inconnue puis λ – en formant le quotient – qui est également inconnue). On est alors amené à considérer la suite

$$y_n = \frac{x_{n+1} - c_n \cdot x_n}{1 - c_n} = \frac{x_n^2 - x_{n+1} \cdot x_{n-1}}{2 \cdot x_n - x_{n+1} - x_{n-1}}$$

En utilisant l'opérateur de différence suivant

$$\begin{aligned} \Delta : \quad \mathbb{R}^{\mathbb{N}} &\longrightarrow \mathbb{R}^{\mathbb{N}} \\ (u_n)_{n \geq 0} &\longmapsto \Delta u = (u_{n+1} - u_n)_{n \geq 0} \end{aligned}$$

on obtient une écriture plus compacte de y_n :

$$y_n = x_n - \frac{(\Delta x_n)^2}{\Delta^2 x_n}$$

La suite $(y_n)_{n \geq 0}$ est appelée transformée d'Aitken de $(x_n)_{n \geq 0}$. Contrairement aux procédés d'Euler et de Richardson, cette transformation n'est pas linéaire⁴. On a le théorème suivant

Théorème 10. (Aitken)

Avec les notations précédentes, si x_n admet un développement asymptotique du type

$$x_n = \bar{x} + \lambda \cdot \rho^n + o(\rho^n) \quad \text{avec} \quad 0 < |\rho| < 1 \quad \text{et} \quad \lambda \in \mathbb{R}^*$$

où ρ et λ sont inconnus, la suite $(y_n)_{n \geq 0}$ converge vers $\bar{x}$ et $y_n - \bar{x} = o(x_n - \bar{x})$.

4. C'est-à-dire que l'application $T : \mathbb{R}^{\mathbb{N}} \rightarrow \mathbb{R}^{\mathbb{N}}$ correspondante n'est pas linéaire.

Exercice 6.

Prouver le théorème 10.

Exercice 7.

Ecrire une procédure **Aitken (u,n)** calculant le n -ième terme de la transformée d'Aitken de la suite u . Utiliser cette procédure pour accélérer la convergence de la suite définie par

$$u_n = n \cdot \sin\left(\frac{\pi}{n}\right)$$

4.5. Méthode de Shanks

Il existe d'autres méthodes d'accélération de la convergence. La méthode de Shanks (1949) 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 ρ_i^n dans le développement

$$x_n = \bar{x} + \lambda_1 \cdot \rho_1^n + \cdots + \lambda_k \cdot \rho_k^n + o(\rho_k^n) \quad \text{où} \quad 1 > |\rho_1| > \cdots > |\rho_k| > 0$$

mais cette fois-ci on ne connaît pas les ρ_i . Cette méthode est réalisée au moyen de l'algorithme de Wynn. Nous n'exposerons pas cette technique ici. Le lecteur curieux recherchera quelques informations supplémentaires sur la page

http://fr.wikipedia.org/wiki/Transformation_de_Shanks