

Dérivation/intégration

I. Dérivation numérique

1 - Calcul approché d'une dérivée

a) Présentation

Soit $f : I \rightarrow \mathbb{R}$ de classe \mathcal{C}^2 sur I . Soit $x_0 \in I$.

Alors d'après le théorème de Taylor-Young, on a :

$$f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2}{2}f''(x_0) + h^2\varepsilon(h) \text{ où } \lim_{h \rightarrow 0} \varepsilon(h) = 0$$

$$\text{c'est-à-dire } \frac{f(x_0 + h) - f(x_0)}{h} = f'(x_0) + \frac{h}{2}f''(x_0) + h\varepsilon(h)$$

On dit alors que $h \mapsto f(x_0 + h)$ admet un développement limité d'ordre 2 au voisinage de 0.

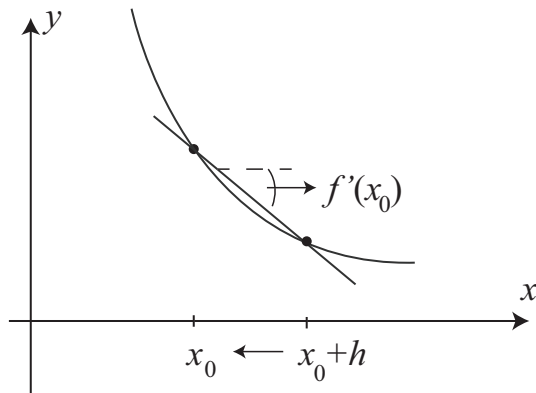


FIGURE 1 – Taux d'accroissement



Propriété :

$$f'(x_0) = \lim_{h \rightarrow 0} \frac{f(x_0 + h) - f(x_0)}{h}$$



Remarque 1 :

En théorie, il suffit de prendre h très petit pour obtenir une grande précision dans le calcul de la dérivée. Toutefois, en prenant une valeur de h trop petite, l'erreur relative sur le calcul de $f(x_0 + h) - f(x_0)$ peut être grande et conduire à une mauvaise approximation de $f'(x_0)$. Si la fonction est de l'ordre de l'unité, alors la valeur minimale de h est 10^{-15} pour un codage en 64 bits.

b) Dérivée à droite, dérivée à gauche, dérivée centrée



Définition :

On appelle approximation de la dérivée à **gauche** la relation :

$$f'(x_0) \approx \frac{f(x_0) - f(x_0 - h)}{h}$$

On appelle approximation de la dérivée à **droite** la relation :

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h}$$

Ces approximations sont dites d'ordre 1 car l'erreur commise sur la vraie valeur de $f'(x_0)$ est proportionnelle à h . Il est possible d'avoir une meilleure approximation du nombre dérivée par une approximation dite **centrée**. Soit $f : I \rightarrow \mathbb{R}$ de classe \mathcal{C}^3 sur I . Soit $x_0 \in I$. La formule de Taylor-Young conduit aux deux relations :

$$f(x+h) = f(x) + \sum_{n=1}^3 \frac{h^n}{n!} u^{(n)}(x) + h^3 \varepsilon_1(x, h)$$

$$f(x-h) = f(x) + \sum_{n=1}^3 \frac{(-h)^n}{n!} u^{(n)}(x) + h^3 \varepsilon_2(x, h)$$

où les deux fonctions $\varepsilon_i(x, h)$ convergent vers 0 avec h . Par conséquent

$$\frac{f(x+h) - f(x)}{h} = f'(x) + \frac{h}{2} f''(x) + \frac{h^2}{3!} f^{(3)}(x) + h^2 \varepsilon_1(x, h)$$

$$\frac{f(x-h) - f(x)}{h} = -f'(x) + \frac{h}{2} f''(x) - \frac{h^2}{3!} f^{(3)}(x) + h^2 \varepsilon_2(x, h)$$

correspondent à deux approximations de $f'(x)$ du 1er ordre en h .

En soustrayant les développements précédents, ce qui revient à faire la moyenne des deux différences finies antérieure et postérieure à $f(x)$, on obtient

$$\frac{f(x+h) - f(x-h)}{2h} = f'(x) + \frac{h^2}{6} f^{(3)}(x) + h^2 \varepsilon_3(x, h)$$

qui est une approximation de $f'(x)$ du 2^e en h .

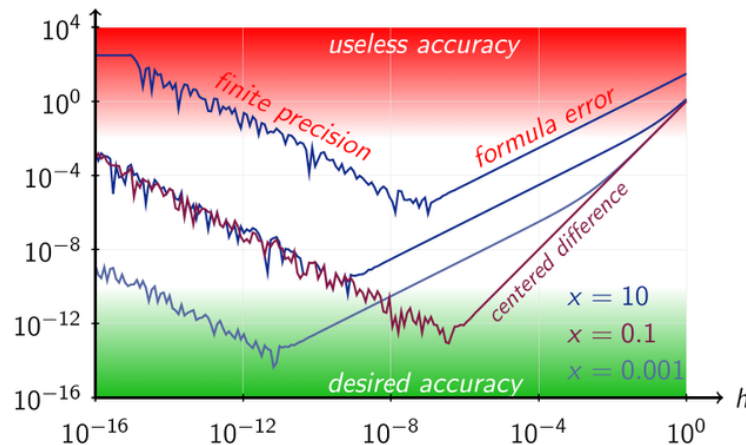


FIGURE 2 – Erreur Numérique pour une dérivée

2 - Application

Il peut être utile à partir d'une fonction mathématiques de connaître sa dérivée. Pour une modélisation physique, on peut ainsi avoir accès à la vitesse à partir de la position.

```
1 def deriv(f,x):
2     h=1e-6 % Valeur arbitrairement petite
3     f_prime=(f(x+h)-f(x))/h
4     return f_prime
```

La valeur de h peut être arbitrairement petite tout en assurant que le calcul est dans les limites de la représentation des des nombres de python. En effet, Python possède 16 chiffres significatifs. Pour une fonction cos dont les valeurs sont comprises entre -1 et 1 , la valeur de $h = 1e - 6$ est choisie pour que la dérivée soit une approximation à $16 - 6 = 10$ CS.

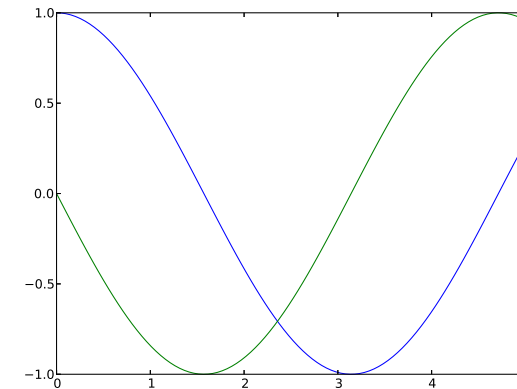


FIGURE 3 – Fonction cos et sa dérivée

3 - Dérivation à partir de données

a) Utilisation de listes

En général, un capteur de position fourni une grandeur analogique qui peut être récupérée dans un fichier. Ces données se présentent sous la forme de

listes, les mesures sont généralement prises à intervalle de temps fixe T_e appelée période d'échantillonnage. En reprenant les expressions précédentes, pour une liste de valeur X , l'approximation du nombre dérivée V s'exprime alors sous la forme suivante

```
1 #dérivée à droite
2 V[n]=(X[n+1]-X[n])/Te
3 #dérivée à gauche
4 V[n]=(X[n]-X[n-1])/Te
5 #dérivée centrée
6 V[n]=(X[n+1]-X[n-2])/(2*Te)
```

b) Dérivation de données

Considérons une liste : $X = [x_1, x_2, \dots, x_N]$ décrivant la position X d'un mobile à intervalle de temps T_e , on peut définir une liste V contenant la vitesse du mobile en utilisant

- pour le premier point, la dérivée à droite ;
- pour les points 2 à $N-1$, la dérivée centrée ;
- pour le point N , la dérivée à gauche.

```
1 V=len(X)*[0]
2 V[0]=(X[1]-X[0])/Te
3 for n in range(2,N-1)
4     V[n]=(X[n+1]-X[n-2])/(2*Te)
5 V[N-1]=(X[N-1]-X[N-2])/Te
```

Remarque 2 :

La dérivée de données numériques s'accompagne généralement d'une augmentation sensible du bruit de mesure. En effet, l'aggrandissement sur la courbe ci-dessous révèle des fluctuations de X qui peuvent être importantes même si le bruit de mesure paraît faible au regard de l'évolution de x .

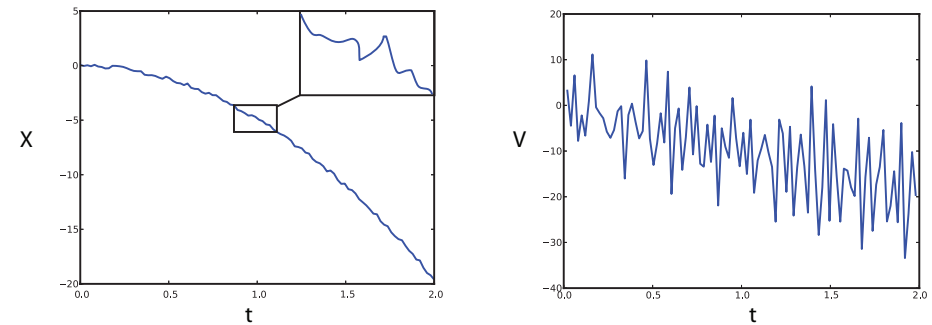


FIGURE 4 – Position et dérivée numérique.

4 - Calcul approché d'une dérivée seconde à l'ordre 2

Soit $f : I \rightarrow \mathbb{R}$ de classe \mathcal{C}^4 sur I . Soit $x_0 \in I$.

Alors d'après le théorème de Taylor-Young, on a :

$$f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2}{2}f''(x_0) + \frac{h^3}{6}f^{(3)}(x_0) + \frac{h^4}{24}f^{(4)}(x_0) + h^4\varepsilon_1(h)$$

où $\lim_{h \rightarrow 0} \varepsilon_1(h) = 0$

On dit alors que $h \mapsto f(x_0 + h)$ admet un développement limité d'ordre 3 au voisinage de 0.

$$\text{et } f(x_0 - h) = f(x_0) - hf'(x_0) + \frac{h^2}{2}f''(x_0) - \frac{h^3}{6}f^{(3)}(x_0) + \frac{h^4}{24}f^{(4)}(x_0) + h^4\varepsilon_2(h)$$

où $\lim_{h \rightarrow 0} \varepsilon_2(h) = 0$

$$\text{D'où } \frac{f(x_0 + h) + f(x_0 - h) - 2f(x_0)}{h^2} = f''(x_0) + \frac{h^2}{12}f^{(4)}(x_0) + h^2\varepsilon(h) \text{ où}$$

$\lim_{h \rightarrow 0} \varepsilon(h) = 0$



Propriété :

Le nombre dérivé $f''(x_0)$ peut être approximé à l'ordre 2 par $\frac{f(x_0 + h) + f(x_0 - h) - 2f(x_0)}{h^2}$ pour h suffisamment proche de 0.

Remarque 3 :

$$\frac{f(x_0+h)+f(x_0-h)-2f(x_0)}{h^2} = \frac{\frac{f(x_0+h)-f(x_0)}{h} - \frac{f(x_0)-f(x_0-h)}{h}}{h}$$

Remarque 4 :

Il faudra être très vigilant sur le choix du nombre de points par rapport au nombre d'intervalles ! Pour N points il y a N-1 intervalles, pour N intervalles, il faut N+1 points !

5 - Utilisation de listes

Considérons une liste : $X = [x_1, x_2, \dots, x_N]$ décrivant la position X d'un mobile par intervalle de temps T_e . On obtient l'accélération par le calcul suivant :

```
1 A[N] = (X[N+1] + X[N-1] - 2X[n]) / Te**2
```

II. Intégration numérique**1 - Méthode des rectangles****a) Représentation numérique d'une fonction**

Pour une fonction quelconque, la représentation numérique d'une fonction est associée à un domaine de définition du type $I = [a; b]$. On évalue cette fonction sur une « grille » de N valeurs de x prise entre a et b . On utilise à ce titre le plus souvent la fonction `linspace` (avec le package Numpy ou pylab)

```
1 from numpy import *
2 a=0 # borne inf
3 b=2 # borne sup
4 N=100 # nombre de points
5 x=linspace(a,b,N)
```

Cette fonction permet de créer une liste de valeurs de x . Le pas de discrétisation en x est donné par

$$\Delta x = \frac{b-a}{N-1}$$

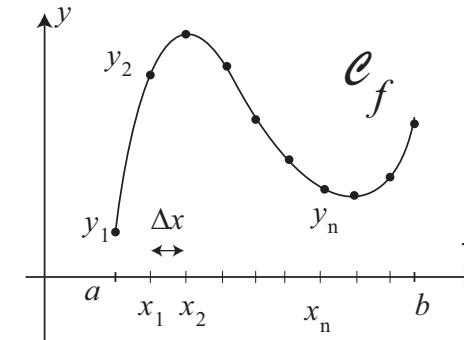


FIGURE 5 – Représentation numérique d'une fonction

La représentation d'une fonction mathématique, même continue est une succession de coordonnées de points $(x_i, y_i = f(x_i))$. Cette représentation discrète (point par point) est utilisée par tous les instruments numériques traitant de signaux (capteur de vitesse, de température...).

Un signal analogique noté $s(t)$ est un signal **continu**. La représentation numérique d'un signal analogique est constitué de deux liste T et S. En général, T est de la forme $k \times T_e$ où T_e est la période d'échantillonnage du signal.

On a alors la représentation suivante :

$$s(T[k]) = S_k$$

En présence d'accéléromètre, il peut être intéressant d'intégrer le signal fourni par l'accéléromètre pour obtenir la vitesse puis la position.

b) Principe

La méthode des rectangles consiste à approximer la fonction f par une fonction en escalier. On considère un entier n et un pas de subdivision $\frac{b-a}{n}$. Pour

tout entier k de $\llbracket 0, n \rrbracket$, on pose $a_k = a + k \frac{b-a}{n}$. Sur l'intervalle $[a_k, a_{k+1}]$, on approxime f par la fonction constante égale à $f(a_k)$ (respectivement $f(a_{k+1})$).

On prend comme valeur approchée de l'intégrale de f sur $[a, b]$ l'intégrale de la fonction en escalier ainsi construite, c'est-à-dire la somme des aires des rectangles qui vaut S_n (respectivement S'_n) : cf figures ci-dessous.

▲ Définition : sommes de Riemann

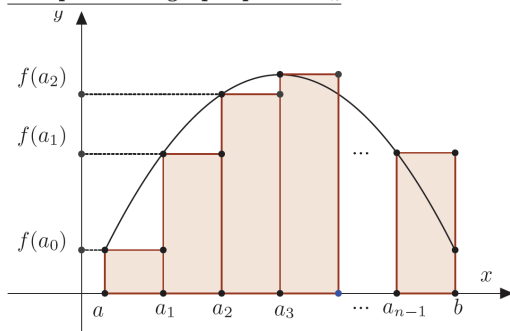
Soit $f : [a, b] \rightarrow \mathbb{R}$ continue sur $[a, b]$.

On appelle sommes de Riemann les suites (S_n) et (S'_n) définies pour tout $n \in \mathbb{N}^*$ par :

$$S_n = \frac{b-a}{n} \sum_{k=0}^{n-1} f\left(a + k \frac{b-a}{n}\right) = \frac{b-a}{n} f(a) + \frac{b-a}{n} f\left(a + \frac{b-a}{n}\right) + \dots + \frac{b-a}{n} f\left(a + (n-1) \frac{b-a}{n}\right)$$

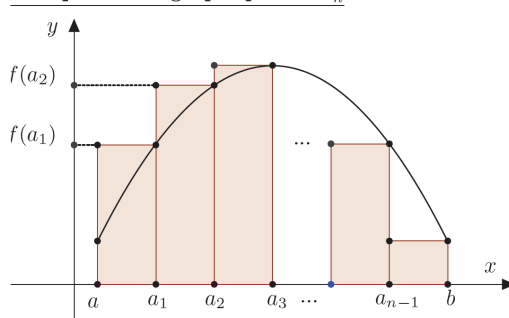
$$\text{et } S'_n = \frac{b-a}{n} \sum_{k=1}^n f\left(a + k \frac{b-a}{n}\right) = \frac{b-a}{n} f\left(a + \frac{b-a}{n}\right) + \frac{b-a}{n} f\left(a + 2 \frac{b-a}{n}\right) + \dots + \frac{b-a}{n} f(b)$$

interprétation graphique de S_n :



La somme des aires des rectangles est S_n .
avec $a_k = a + k \frac{b-a}{n}$ pour tout $k \in \llbracket 0, n-1 \rrbracket$

interprétation graphique de S'_n :



La somme des aires des rectangles est S'_n .
avec $a_k = a + k \frac{b-a}{n}$ pour tout $k \in \llbracket 0, n-1 \rrbracket$

📌 Remarque 5 :

Pour une intégration « à gauche » (en prenant le premier point), la méthode des rectangles minore (resp. croissante) l'intégrale vraie pour une courbe croissante (resp. décroissante).

📌 Propriété :

Soit $f : [a, b] \rightarrow \mathbb{R}$ continue sur $[a, b]$.

Alors les sommes de Riemann (S_n) et (S'_n) convergent vers $\int_a^b f(t) dt$.

📌 Propriété : vitesse de convergence et majoration de l'erreur

Soit $f \in \mathcal{C}^1([a, b], \mathbb{R})$. Posons $M = \max_{x \in [a, b]} |f'(x)|$.

Alors $\left| \int_a^b f(t) dt - S_n \right| \leq \frac{M(b-a)^2}{2n}$ ce qui implique $\int_a^b f(t) dt - S_n = O_{+\infty}\left(\frac{1}{n}\right)$.

📌 Remarque 6 :

La dernière inégalité permet d'obtenir une majoration de l'erreur commise en approximant $\int_a^b f(t) dt$ par S_n . Ce résultat reste valable si on remplace S_n par S'_n .

**Exemple 1 :**

Voici un code Python donnant une valeur approchée de $\int_0^\pi \sin t \, dt$ avec 100 rectangles. On suppose que le module pylab a été importé.

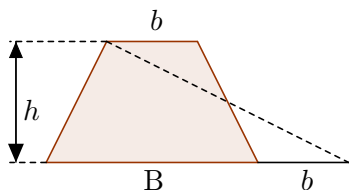
```
1 N=100
2 dt=pi/N
3 t=linspace(0,pi,N+1)
4 S=0
5 for k in range(N):
6     S+=sin(t[k])*dt
7 #ou S+=sin(t[k+1])*dt
```

2 - Méthode des trapèzes**a) Principe**

La méthode des trapèzes consiste à approximer la fonction f par une fonction continue affine par morceaux. On considère un entier n et un pas de subdivision $\frac{b-a}{n}$. Pour tout entier k de $\llbracket 0, n \rrbracket$, on pose $a_k = a + k \frac{b-a}{n}$. Sur l'intervalle $[a_k, a_{k+1}]$, on approxime f par la fonction affine qui joint les points $(a_k, f(a_k))$ et $(a_{k+1}, f(a_{k+1}))$.

On prend comme valeur approchée de l'intégrale de f sur $[a, b]$ l'intégrale de la fonction continue affine par morceaux, c'est-à-dire une somme d'aires de trapèzes notée T_n .

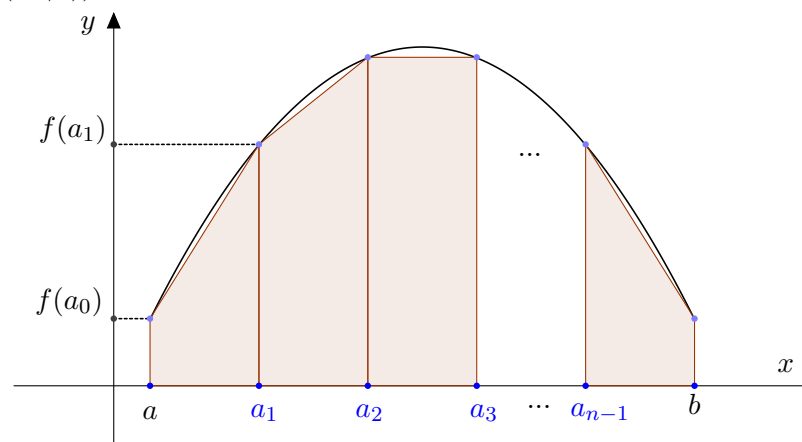
L'aire d'un trapèze de petite base b , de grande base B et de hauteur h est ...



interprétation graphique de T_n :

Au lieu d'une somme de rectangles, on effectue la somme T_n d'aires de trapèzes de sommets $(a_k, 0)$, $(a_{k+1}, 0)$, $(a_k, f(a_k))$ et $(a_{k+1}, f(a_{k+1}))$ avec $a_k = a + k \frac{b-a}{n}$ pour tout $k \in \llbracket 0, n-1 \rrbracket$

Chaque trapèze a pour aire : $(a_{k+1} - a_k) \frac{f(a_k) + f(a_{k+1})}{2} = \frac{(b-a)}{2n} (f(a_k) + f(a_{k+1}))$

**Remarque 7 :**

La méthode des trapèzes minore (resp. majore) l'intégrale vraie si la fonction est concave (resp. convexe)

**Propriété :**

Soit $f : [a, b] \rightarrow \mathbb{R}$ continue sur $[a, b]$.

Soit $n \in \mathbb{N}^*$. On pose $a_k = a + k \frac{b-a}{n}$ pour tout $k \in \llbracket 0, n-1 \rrbracket$.

La somme des aires des trapèzes est :

$$T_n = \frac{b-a}{n} \sum_{k=0}^{n-1} \frac{f(a_k) + f(a_{k+1})}{2}$$

La suite (T_n) converge vers $\int_a^b f(t) \, dt$.

**Exemple 2 :**

Voici un code Python donnant une valeur approchée de $\int_0^\pi \sin t \, dt$ avec 100 trapèzes. On suppose que le module pylab a été importé.

```
1 N=100
2 dt=pi/N
3 t=linspace(0,pi,N+1)
4 S=0
5 for k in range(N):
6     S+=(sin(t[k])+sin(t[k+1]))*dt/2
```

b) Convergence**Propriété : vitesse de convergence et majoration de l'erreur**

Soit $f \in \mathcal{C}^2([a,b], \mathbb{R})$. Posons $M_2 = \max_{x \in [a,b]} |f''(x)|$.

Alors $\left| \int_a^b f(t) \, dt - T_n \right| \leq \frac{M_2(b-a)^3}{12n^2}$ ce qui implique $\int_a^b f(t) \, dt -$

$T_n = O_{+\infty}\left(\frac{1}{n^2}\right)$. La méthode des trapèzes converge beaucoup plus vite que la méthode des rectangles.

III. Résolution EDO 1° ordre

1 - Problème de Cauchy

En analyse, un problème de Cauchy est un problème constitué d'une équation différentielle dont on recherche une solution vérifiant une certaine condition

initiale. Le théorème de Cauchy-Lipschitz assure l'existence et l'unicité d'une solution au problème de Cauchy.

**Définition :**

Les équations différentielles linéaires d'ordre 1 de la forme

$$y' + a(x)y = \delta(x)$$

où a et δ sont des fonctions continues sur I .

Ces équations se retrouvent très fréquemment en physique : chute avec frottement, charge ou décharge d'un condensateur, d'une bobine, évolution thermique...

**Exemple 3 :**

Résoudre sur \mathbb{R} : $y' + ay = 0$, avec $y(0) = 1$. a est une constante réelle.

2 - Approximation numérique

Le principe de la méthode d'Euler repose sur une résolution pas à pas de l'équation différentielle à partir de la condition initiale. Tentons de résoudre numériquement l'équation suivante :

$$y' + ay = 0$$

avec $y(0)$ connu.

Utilisons l'approximation du nombre dérivée pour les points $x_0 = 0$ et $x_1 = 0 + \Delta x$. En remplaçant dans l'équation différentielle, on obtient :

$$y'(0) = -ay(0) \approx \frac{y(0 + \Delta x) - y(0)}{\Delta x}$$

Il suffit maintenant d'isoler $y(\Delta x)$ pour obtenir :

$$y(\Delta x) = y(0) + \Delta x \times (-ay(0))$$

Il est important de remarquer que la valeur $y(\Delta x)$ n'est obtenue qu'avec des informations sur les valeurs "antérieures" en $x = 0$. Une fois connue la valeur de y en $x = \Delta x$, il est possible de déterminer le terme en $x = 2\Delta x$ et de proche en proche de déterminer les points de la fonction :

$$y(2\Delta x) = y(\Delta x) + \Delta x \times (-ay(\Delta x))$$

$$y(3\Delta x) = y(2\Delta x) + \Delta x \times (-ay(2\Delta x))$$

...

La structure de la solution approchée est celle d'une suite avec $u_n = y(n\Delta x)$:

$$u_0 = y(0)$$

$$u_{n+1} = u_n + \Delta x \times (-au_n)$$

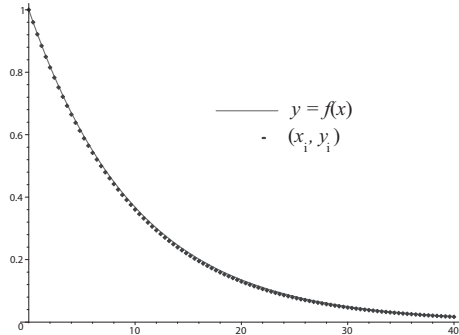


FIGURE 6 – Résolution numérique de $y' + 0,1 \times y = 0$, avec $y(0) = 1$



Exemple 4 :

Écrire un code python permettant de résoudre l'équation différentielle $y' + y = 0$, avec $y(0) = 1$ pour $x \in [0, 5]$ avec $N = 100$ points.

```
1  xmin,xmax=0,5
2  N=100
3  X=linspace(xmin,xmax,N)
4  h=X[1]-X[0]
5  Y=N*[0]
6  for k in range(0,N-1):
7      Y[k+1]=Y[k]+h*(-Y[k])
```

3 - Convergence de la solution

a) Ecart à la solution réelle

L'écart entre la solution numérique et la « vraie » solution dépend du nombre de points dans l'intervalle de résolution.



Définition :

On peut montrer que l'erreur est proportionnelle au pas Δx , on parle alors de méthode du **premier ordre**.

Il existe des méthodes quantitatives pour s'assurer qu'une solution numérique soit proche d'une solution analytique. On parle alors de solution **convergente**. L'intérêt de la méthode d'Euler est de résoudre des cas non-analytique, c'est à dire dont nous ne connaissons pas la solution. Afin de s'assurer que le nombre de points choisi est suffisant, nous effectuerons donc un test « à l'oeil » : c'est à dire que nous considérerons que le nombre de points est suffisant lorsque l'allure de la solution n'est plus modifiée :



Définition :

On dit qu'une équation différentielle discrétisée est consistante par rapport à l'équation différentielle réelle si l'écart des solutions tend vers 0 lorsque le pas de discrétisation tend vers 0

b) Influence de l'équation différentielle

Avec un même pas, deux équations différentes peuvent présenter des écarts importants selon que la solution finale varie rapidement ou non. En effet, la

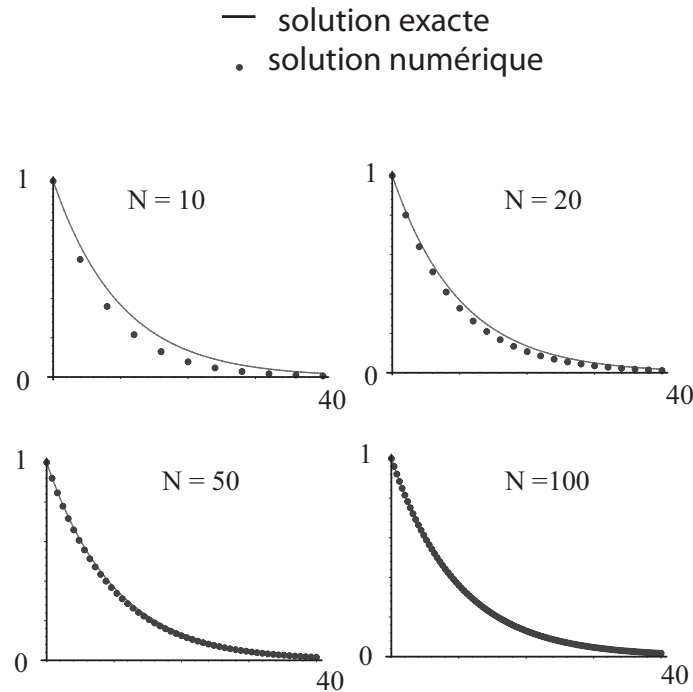
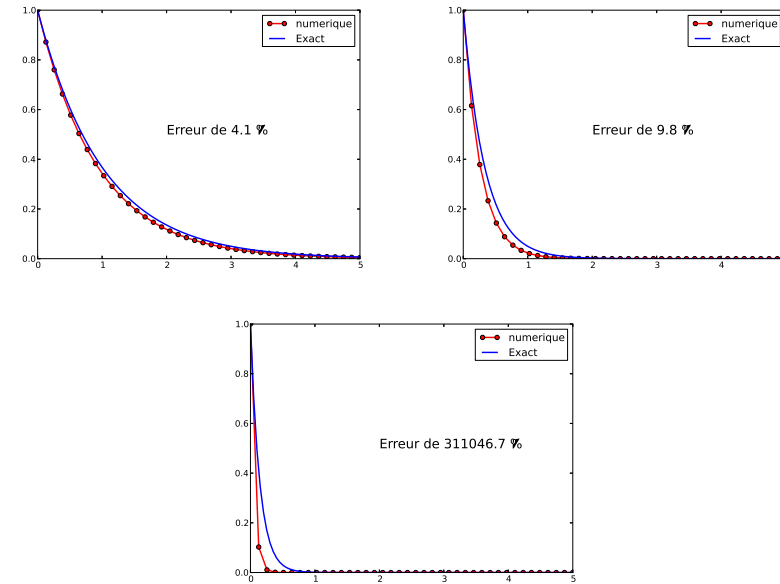


FIGURE 7 – Convergence de la solution numérique

méthode d'Euler utilisant une approximation du nombre dérivée, plus celui-ci est important, plus les écarts seront importants. Résolvons l'équation différentielle $y' = -a \times y$ avec $y(0) = 1$. L'écart à la solution numériques augmente rapidement avec la valeur de a .

FIGURE 8 – Convergence de la solution numérique de l'équation $y' + ay = 0$ pour $a = 1$, $a = 3$ et $a = 7$.

c) instabilité

Si le pas est trop faible, la solution peut même diverger. C'est parfois le cas lorsque la solution doit présenter des oscillations.

$$y'(x) = -y(x)$$

On remarque que la relation de récurrence, pour une équation de ce type est de la forme :

$$Y[k+1] = Y[k] + h * (-aY[k]) = Y[k](1 - ah)$$

Les éléments $Y[k]$ suivent une relation de récurrence d'une suite géométrique de raison $(1 - ah)$. On en déduit que la solution diverge si $|1 - ah| > 1$.



Définition :

Pour des équations du premier ordre, la solution est stable si l'on peut trouver une constante V telle que

$$-1 \leq \frac{X[k+1] - V}{X[k] - V} \leq 1$$

4 - Application en physique non linéaire

a) Présentation

Les problèmes du prix du millénaire comptent sept défis mathématiques réputés insurmontables posés par le Clay Mathematical Institute en 2000. La résolution de chacun des problèmes est dotée d'un prix d'un million de dollars américains offert par l'institut. À ce jour, six des sept problèmes demeurent non résolus. L'un d'eux concerne la mécanique des fluides avec l'équation de Navier Stokes :

$$\rho \left[\frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \vec{\nabla}) \vec{v} \right] = -\vec{\nabla} p + \mu \left[\nabla^2 \vec{v} + \frac{1}{3} \vec{\nabla} (\vec{\nabla} \cdot \vec{v}) \right] + \rho \vec{f}$$

La difficulté de cette équation est l'apparition de produit de dérivée avec d'autres fonctions. Ces équations sont fondamentales pour expliquer le comportement des fluides. Il existe des solutions partielles, mais aucune solution générale n'est encore proposée.

b) Chute à haute vitesse

Considérons la chute d'un parachutiste, ce dernier est soumis à son poids $\vec{P} = m\vec{g}$ et à une force de frottement de l'air. Une force de frottement dans l'air est modélisée par une force proportionnelle au carré de la vitesse :

$$\vec{f} = -av^2 \frac{\vec{v}}{|v|}$$

La 2^e loi de Newton donne

$$m\vec{a} = m\vec{g} + \vec{f}$$

En projetant selon l'axe Oz , on en déduit que

$$m \frac{dv}{dt} = mg - av^2$$

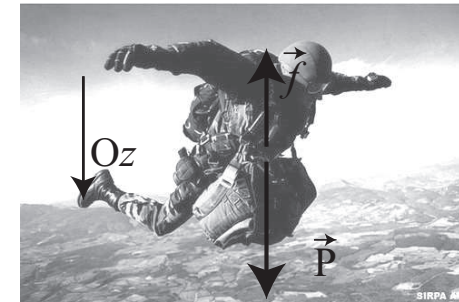


FIGURE 9 – Bilan des forces sur un parachutiste



Exemple 5 :

En décomposant le temps sur une grille de pas Δt , écrire la relation de récurrence entre $u_{n+1} = v(n\Delta t)$ et $u_{n+1} = v((n+1)\Delta t)$



Remarque 8 :

Dans le cas présenté avec une chute parfaitement verticale, les puristes pourront montrer que la solution de cette équation est analytique et fait intervenir la fonction **arctanh**...

IV. Résolution d'une équation différentielle du second ordre

1 - Présentation



Définition :

En mathématiques : les équations différentielles linéaires d'ordre 2 à coefficients constants

$$ay'' + by' + cy = \delta(x),$$

où a, b, c sont des éléments de \mathbb{R} , $a \neq 0$, et δ une fonction continue sur I .

En physique, on rencontre fréquemment ces équations pour des systèmes oscillants avec ou sans amortissement. L'équation différentielle canonique est alors donnée par :

$$\frac{d^2x}{dt^2} + \frac{\omega_o}{Q} \frac{dx}{dt} + \omega_0^2 x = 0$$

On remarquera que, par rapport aux mathématiques, il y a simplement :

$$\omega_0/Q = b/a \quad \text{et} \quad \omega_0^2 = c/a$$

2 - Résolution numérique



Exemple 6 :

Résoudre par la méthode d'Euler l'équation de l'oscillateur harmonique :

$$\ddot{x} + \omega_0^2 x = 0$$

, avec $\omega_0 = 1 \text{ rad.s}^{-1}$, pendant 10 s avec 100 points.

On peut donc décomposer l'équation différentielle précédente du 2° ordre en deux équations du 1° ordre :

$$\frac{dx}{dt} = \dot{x}$$

$$\frac{d\dot{x}}{dt} = -\omega_0^2 x$$

```

1 Tmin=0
2 Tmax=10
3 N=100
4 T=linspace(Tmin,Tmax,N)
5 h=(Tmax-Tmin)/(N-1)
6 X=N*[0]
7 V=N*[0]
8 X[0]=1
9 V[0]=0
10 wo=1
11 for k in range(0,N-1):
12     X[k+1]=X[k]+h*V[k]
13     V[k+1]=V[k]+h*(- wo**2 X[k])

```

On peut donc décomposer l'équation différentielle précédente du 2° ordre en deux équations du 1° ordre :

$$\frac{dx}{dt} = \dot{x}$$

$$\frac{d\dot{x}}{dt} = -\frac{\omega_0}{Q} \dot{x} - \omega_0^2 x$$



Exemple 7 :

Résoudre par la méthode d'Euler l'équation de l'oscillateur harmonique :

$$\ddot{x} + \frac{\omega_0}{Q} \dot{x} + \omega_0^2 x = 0$$

, avec $\omega_0 = 1 \text{ rad.s}^{-1}$, $Q = 5$, pendant 4 s avec 100 points. Les conditions initiales, sont $x(0) = 1$, et $\dot{x}(0) = 0$.

Afin de résoudre ces équations, on considère deux listes dont les relations de récurrence entre les différents éléments consistent en des suites imbriquées définies par la position et la vitesse :

$$X[k+1] = X[k] + h * V[k]$$

$$V[k+1] = V[k] + h * (-wo/Q * V[k] - wo * 2X[k])$$

```

1 Tmin=0
2 Tmax=4
3 N=100
4 T=linspace(Tmin,Tmax,N)
5 h=(Tmax-Tmin)/(N-1)
6 X=N*[0]
7 V=N*[0]
8 X[0]=1
9 V[0]=0
10 wo=1
11 Q=5
12 for k in range(0,N-1):
13     X[k+1]=X[k]+h*V[k]
14     V[k+1]=V[k]+h*(-wo/Q*V[k] - wo**2 X[k])

```

3 - Pas fixe

```

1 Tmax=10
2 h=0.1
3 X[0]=[1]
4 V[0]=[0]
5 t=0
6 T=[t]
7 k=0
8 while t<Tmax:
9     X.append(X[k]+h*V[k])
10    V.append(V[k]+h*(-wo/Q*V[k] - wo**2 X[k]))
11    k+=1
12    t+=h
13    T.append(t)

```

4 - Résolution d'une équation différentielle 2D

a) Équations

La trajectoire d'un corps dans l'espace peut souvent être ramené à un plan. Le principe fondamental de la dynamique donne les équations différentielles régissant l'évolution de la vitesse en fonction des forces en présence. Pour une chute libre, sans frottement, les équations sont :

$$\begin{cases} \dot{v}_x = 0 \\ \dot{v}_y = -g \end{cases} \quad (1)$$

avec $v_x(0) = 0$ et $v_y(0) = V_0$.

Le découplage des équations en vue d'une résolution numérique conduit à un système de 4 équations :

$$\begin{cases} \dot{v}_x = 0 \\ \dot{v}_y = -g \\ \dot{x} = v_x \\ \dot{y} = v_y \end{cases} \quad (2)$$

b) Code Python correspondant

```

1 Tmin=0
2 Tmax=10
3 N=100
4 T=linspace(Tmin,Tmax,N)
5 h=(Tmax-Tmin)/(N-1)
6 X,Y,Vx,Vy=N*[0],N*[0],N*[0],N*[0]
7 X[0],Y[0],Vx[0],Vy[0]=0,0,1,1
8 for k in range(0,N-1):
9     Vx[k+1]=Vx[k]+0
10    Vy[k+1]=Vx[k]+h*(-g)
11    X[k+1]=X[k]+h*Vx[k]
12    Y[k+1]=Y[k]+h*Vy[k]

```