

# Chapitre 7

## Résolution numérique des équations différentielles ordinaires (EDO)

### 7.1 Introduction

**Définition 7.1.** Une **équation différentielle (ED)** est une équation pour laquelle la (ou les) inconnue(s) sont **des fonctions**. L'équation est une relation entre la (les) fonction(s) inconnue(s) et ses (leurs) **dérivées** (éventuellement partielles).

**Exemple 7.2.** En physique, l'équation de Newton

$$\vec{F} = m\vec{a}$$

fournit de nombreux exemples d'équations différentielles.  
Nous allons brièvement en évoquer trois.

1. Dans le cas de la chute libre d'une masse  $m$ , la seule force est le poids  $m\vec{g}$  et l'équation de Newton s'écrit, selon un axe vertical dirigé vers le bas,

$$mg = ma \equiv m\dot{v}.$$

Ainsi, on cherche dans ce cas à résoudre l'équation différentielle suivante :

$$\dot{v} = g,$$

où la fonction cherchée est la vitesse  $v = v(t)$ . Expérimentalement, on observe que la vitesse augmente linéairement dans le temps et on se convainc facilement, **par intégration**, que la solution est une fonction affine :

$$v(t) = A + gt,$$

où  $A$  est une constante fixée par une **condition** (par exemple, la vitesse **initiale** :  $v(t = 0) = A$ ).

2. Dans le cas d'une masse  $m$  accrochée horizontalement à un ressort de constante de rappel  $k$ , l'équation de Newton s'écrit

$$-kx = ma \equiv m\ddot{x}.$$

L'équation différentielle

$$\ddot{x} = -\frac{k}{m}x,$$

a pour solution la fonction

$$x = x(t) = A \sin \omega_0 t + B \cos \omega_0 t.$$

avec  $\omega_0 = \sqrt{k/m}$ , et  $A, B$  deux " constantes d'intégration".

La masse oscille autour de la position d'équilibre  $x = 0$ .

3. Une particule de masse  $m$  possédant une charge  $q$  et se déplaçant dans un champ magnétique  $\vec{B}$  subit une force appelée force de Lorentz. L'équation de Newton de cette particule s'écrit

$$q \vec{v} \wedge \vec{B} = m \vec{a} \equiv m \dot{\vec{v}}.$$

La vitesse de la particule satisfait donc l'équation différentielle vectorielle

$$\dot{\vec{v}} = \frac{q}{m} \vec{v} \wedge \vec{B}.$$

La trajectoire de la particule est une hélice.

◇

**Définition 7.3.** Une équation aux dérivées partielles est une équation différentielle avec des dérivées partielles d'au moins une fonction inconnue qui dépend de plusieurs variables.

**Exemple 7.4.** Supposons que la fonction  $T(x, y, z, t)$  représente la température, la notation

$$\frac{\partial}{\partial x} T(x, y, z, t)$$

signifie que l'on dérive  $T$  par rapport à  $x$  uniquement, en gardant  $y, z$ , et  $t$  constants. Autrement dit, on s'intéresse à la variation de  $T$  dans la direction  $x$  uniquement.

◇

**Définition 7.5.** L'ordre d'une équation différentielle est le degré le plus élevé de dérivation (partielle ou non) d'une fonction inconnue de l'équation différentielle.

**Définition 7.6.** Une équation différentielle ordinaire (EDO) est une équation différentielle qui ne fait intervenir qu'une seule variable. Il n'y a pas de dérivée partielle dans une équation différentielle ordinaire.

Dans ce chapitre, nous allons nous intéresser exclusivement à la résolution numérique des équations différentielles ordinaires.

**Définition 7.7.** Soit  $F$  une fonction (continue) d'une variable  $x$  (ou  $t$ ), de  $y = y(x)$  une fonction inconnue, et des dérivées  $y', y'', \dots$  de  $y$ .

Une équation de la forme

$$F(x, y, y', y'', \dots, y^{(n-1)}) = y^{(n)}$$

est une **EDO explicite d'ordre  $n$** .

Une équation de la forme

$$F(x, y, y', y'', \dots, y^{(n)}) = 0$$

est une **EDO implicite d'ordre  $n$** .

**Définition 7.8.** Si  $F$  ne dépend pas explicitement de  $x$ , autrement dit si  $F(x, y, y', y'', \dots, y^{(n)}) \equiv F(y, y', y'', \dots, y^{(n)})$ , l'ED est dite **autonome**.

**Définition 7.9. Résoudre une équation différentielle** revient à **trouver (toutes) les fonctions  $y$  solutions**. Une fonction solution  $y_{\text{sol.}}(x)$  est solution de l'EDO implicite  $F(x, y, y', y'', \dots, y^{(n)}) = 0$  si,  $\forall x$ , on a  $F(x, y_{\text{sol.}}, y'_{\text{sol.}}, y''_{\text{sol.}}, \dots, y^{(n)}_{\text{sol.}}) = 0$ .

**Exemple 7.10.** L'EDO

$$y'' + y = 0,$$

que l'on peut réécrire sous la forme

$$y'' = -y,$$

admet une infinité de solutions de la forme

$$y_{\text{sol.}}(x) = A \cos x + B \sin x, \text{ où } A, B \in \mathbb{R}.$$

Les constantes  $A$  et  $B$  peuvent être déterminées en imposant **deux conditions initiales**. Le nombre de constantes à imposer correspond à l'ordre de l'ED. Le Par  $\diamond$

Dans ce chapitre, nous allons nous limiter à la résolution d'**équations différentielles ordinaires du premier ordre**. Il est toujours possible de résoudre une EDO d'ordre supérieur en résolvant un système d'EDO du premier ordre.

## 7.2 Problème de Cauchy

Dans ce chapitre, nous allons chercher à résoudre numériquement le problème suivant :

**Définition 7.11. Problème de Cauchy** (pour une EDO du premier ordre) :

Trouver une fonction  $y : I \subset \mathbb{R} \rightarrow \mathbb{R}$  vérifiant

$$\begin{cases} y'(t) &= f(t, y(t)), \forall t \in I, \\ y(t_0) &= y_0, \end{cases}$$

avec  $f : I \times \mathbb{R} \rightarrow \mathbb{R}$  une fonction donnée.

### 7.3. Méthodes numériques à un pas

---

La première ligne du problème de Cauchy fournit l'**EDO du premier ordre à résoudre**. L'EDO est ici donnée sous forme explicite.

La seconde ligne correspond à la **condition de Cauchy** :  $t_0 \in I$  est le point (ou le moment) initial et  $y_0$  est la valeur (donnée) initiale.

**Remarque 7.12.** On note qu'en intégrant la première ligne du problème de Cauchy entre  $t_0$  et  $t$ , on obtient

$$\int_{t_0}^t y'(\tau) d\tau = \int_{t_0}^t f(\tau, y(\tau)) d\tau.$$

Ainsi, comme le terme de gauche n'est autre que  $y(t) - y(t_0)$ , le problème de Cauchy peut être écrit de manière équivalente sous forme intégrale :

$$y(t) = y(t_0) + \int_{t_0}^t f(\tau, y(\tau)) d\tau \equiv y_0 + \int_{t_0}^t f(\tau, y(\tau)) d\tau.$$

◇

La solution au problème de Cauchy est souvent appelée "**l'intégrale**" de l'EDO.

## 7.3 Méthodes numériques à un pas

Le déroulement d'une **méthode numérique à un pas** est le suivant :

- On choisit une **partition régulière** de  $I = [t_0, T]$  :  $t_0 < t_1 < t_2 < \dots < t_N = T$ , où  $N$  est le nombre de sous-intervalles et  $h = \frac{T-t_0}{N}$  est le **pas de discréétisation**.
- Pour chacun des **noeuds**  $t_n = t_0 + nh$ , avec  $1 \leq n \leq N$ , on cherche la valeur inconnue  $u_n$  qui approche  $y_n \equiv y(t_n)$ . L'ensemble des valeurs  $\{u_0 \equiv y_0, u_1, u_2, \dots, u_N\}$ , construit à partir de la condition initiale  $u_0 \equiv y_0$ , représente la **solution numérique**.
- On repète éventuellement la méthode en considérant des partitions de plus en plus fines, c'est-à-dire des partitions avec des pas  $h$  de plus en plus petits.

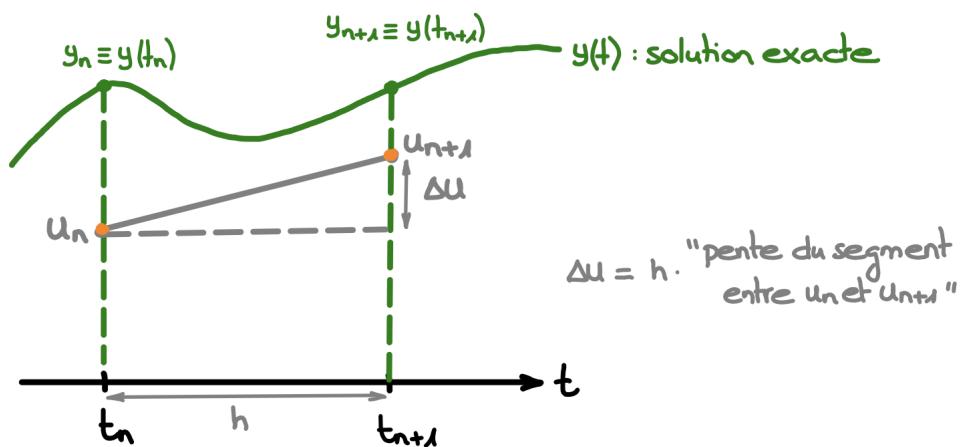
Pour chaque sous-intervalle  $[t_n, t_{n+1}]$ , on a l'égalité

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f(t, y(t)) dt.$$

La méthode numérique choisie permet d'obtenir une approximation de l'intégrale donnant  $y_{n+1} \equiv y(t_{n+1})$  à partir de  $y_n \equiv y(t_n)$ . Nous allons envisager six manières de calculer numériquement cette intégrale.

Chacune de ces méthodes permet d'obtenir, pour chaque instant  $t_{n+1}$  avec  $n = 0, \dots, N-1$ , c'est-à-dire pour chaque noeud de la partition de pas  $h$  choisie, **une approximation numérique**  $u_{n+1}$  de  $y_{n+1}$  :

$$u_0 \equiv y_0 \longrightarrow u_1 \longrightarrow u_2 \longrightarrow \dots \longrightarrow u_n \longrightarrow u_{n+1}$$



On obtient l'approximation  $u_{n+1}$  en  $t_{n+1}$  en partant de l'approximation  $u_n$  en  $t_n$  :

$$u_{n+1} = u_n + \Delta u,$$

où  $\Delta u$  n'est autre que la "pente" choisie multipliée par le pas  $h$ .

### 7.3.1 Méthode d'Euler progressive

Dans la **méthode d'Euler progressive**, on choisit d'approcher numériquement l'intégrale définie

$$\int_{t_n}^{t_{n+1}} f(t, y(t)) dt$$

à l'aide de la formule de quadrature non composite du **point de gauche** :

$$J_n^{\text{PG}}(f) = \underbrace{(t_{n+1} - t_n)}_{=h} f(t_n, y(t_n)).$$

En remplaçant  $y(t_n)$ , dont la valeur est inconnue, par l'approximation  $u_n$ , on obtient alors le **schéma numérique** suivant :

$$\begin{cases} u_{n+1} = u_n + h f_n, \\ u_0 = y_0. \end{cases}$$

où  $f_n = f(t_n, u_n)$ .

Dans cette méthode, la "pente"  $\Delta u/h$  choisie est

$$f_n \equiv f(t_n, u_n)$$

et

$$u_{n+1} = u_n + \Delta u = u_n + h f_n = u_n + h f(t_n, u_n).$$

### 7.3.2 Méthode d'Euler rétrograde

Dans la **méthode d'Euler rétrograde**, on approche numériquement l'intégrale définie

$$\int_{t_n}^{t_{n+1}} f(t, y(t)) dt$$

à l'aide de la formule de quadrature non composite du **point de droite** :

$$J_n^{\text{PD}}(f) = \underbrace{(t_{n+1} - t_n)}_{=h} f(t_{n+1}, y(t_{n+1})).$$

On obtient alors le **schéma numérique** suivant :

$$\begin{cases} u_{n+1} = u_n + h f_{n+1}, \\ u_0 = y_0. \end{cases}$$

Cette méthode est **implicite** car  $f_{n+1} = f(t_{n+1}, u_{n+1})$ . Ainsi, on est finalement amené à rechercher le point fixe de l'équation

$$u_{n+1} = u_n + h f_{n+1} = g(u_{n+1}).$$

Dans cette méthode, la "pente"  $\Delta u/h$  choisie est

$$f_{n+1} \equiv f(t_{n+1}, u_{n+1})$$

et

$$u_{n+1} = u_n + \Delta u = u_n + h f_{n+1} = u_n + h f(t_{n+1}, u_{n+1}).$$

Ici, pour obtenir l'approximation  $u_{n+1}$ , on est donc naturellement amené à résoudre une équation par la méthode de point fixe :

$$u_{n+1} = u_n + h f_{n+1} = g(u_{n+1}).$$

C'est ce que l'on fait par exemple à l'exercice 2 de la série 23. Il est également possible de résoudre cette équation par une autre méthode de recherche de zéros (voir par exemple l'exercice 3 de la série 23 dans lequel on utilise la fonction `newton` de SciPy).

### 7.3.3 Méthode de Crank-Nicolson

Dans la **méthode de Crank-Nicolson**, on approche numériquement l'intégrale définie à l'aide de la formule de quadrature non composite du **trapèze** :

$$J_n^{\text{TR}}(f) = \underbrace{(t_{n+1} - t_n)}_{=h} \frac{f(t_n, y(t_n)) + f(t_{n+1}, y(t_{n+1}))}{2}.$$

On obtient alors le **schéma numérique implicite** suivant :

$$\begin{cases} u_{n+1} = u_n + \frac{h}{2}(f_n + f_{n+1}), \\ u_0 = y_0. \end{cases}$$

La nature implicite de cette méthode rend cette dernière difficile à mettre en place. Souvent, on utilise plutôt un schéma légèrement modifié grâce à une approche de type “**prédicteur-correcteur**” : on commence par faire une **prédition**, c'est-à-dire un premier calcul, par exemple à l'aide du schéma d'Euler (progressif), pour obtenir une approximation

$$u_{n+1}^* = u_n + h f_n .$$

La valeur inconnue  $f_{n+1}$  est alors remplacée par

$$f_{n+1}^* = f(t_{n+1}, u_{n+1}^*) .$$

En procédant ainsi, on obtient la méthode de Heun décrite ci-dessous.

### 7.3.4 Méthode de Heun

Directement inspirée de la méthode de Crank-Nicolson, la **méthode de Heun** est une méthode **explicite** correspondant au schéma suivant :

$$\begin{cases} u_{n+1} &= u_n + \frac{h}{2}(f_n + f_{n+1}^*) , \\ u_0 &= y_0 . \end{cases}$$

Il s'agit d'une **méthode (de Runge-Kutta) explicite en deux étapes** :

$$\begin{aligned} K_1 &= f(t_n, u_n) , \\ K_2 &= f(t_{n+1}, u_n + h K_1) , \end{aligned}$$

où  $K_1$  et  $K_2$  sont les deux pentes importantes dans la méthode.

Ainsi, le schéma peut être réécrit de la façon suivante :

$$\begin{cases} u_{n+1} &= u_n + \frac{h}{2}(K_1 + K_2) , \\ u_0 &= y_0 . \end{cases}$$

### 7.3.5 Méthode d'Euler modifiée (améliorée)

On peut également envisager d'approcher numériquement l'intégrale du problème de Cauchy à l'aide de la formule de quadrature non composite du **point milieu** :

$$J_n^{\text{PM}}(f) = h f(t_n + \frac{h}{2}, y(t_n + \frac{h}{2})) .$$

Comme on ne connaît pas  $u_{n+1/2}$ , on exploite souvent la méthode d'Euler (progressive) pour écrire :

$$u_{n+1/2} = u_n + \frac{h}{2} f_n .$$

On obtient alors un schéma appelé **méthode d'Euler modifiée (améliorée)** :

$$\begin{cases} u_{n+1} &= u_n + h f_{n+1/2} , \\ u_0 &= y_0 , \end{cases}$$

## 7.4. Champ de directions

---

où  $f_{n+1/2} = f(t_n + \frac{h}{2}, u_{n+1/2})$ . On appelle également cette méthode **la méthode de Runge-Kutta "classique" à deux étapes** :

$$\begin{aligned} K_1 &= f(t_n, u_n), \\ K_2 &= f(t_n + \frac{h}{2}, u_n + \frac{h}{2} K_1). \end{aligned}$$

Ainsi, le schéma peut être réécrit de la façon suivante :

$$\begin{cases} u_{n+1} = u_n + h K_2, \\ u_0 = y_0. \end{cases}$$

### 7.3.6 Méthode classique de Runge-Kutta

Notre étude de l'intégration numérique suggère que l'on peut également envisager utiliser la formule de quadrature non composite de **Simpson** :

$$J_n^S(f) = \frac{h}{6} \left[ f(t_n, y(t_n)) + 4f(t_n + \frac{h}{2}, y(t_n + \frac{h}{2})) + f(t_{n+1}, y(t_{n+1})) \right].$$

La **méthode de Runge-Kutta classique (à 4 étapes) RK4** consiste à prendre (il s'agit d'un choix) :

$$\begin{aligned} K_1 &= f(t_n, u_n) \leftarrow (\text{point de gauche}) \\ K_2 &= f(t_n + \frac{h}{2}, u_n + \frac{h}{2} K_1) \leftarrow (\text{prédiction à l'aide d'Euler progressive}) \\ K_3 &= f(t_n + \frac{h}{2}, u_n + \frac{h}{2} K_2) \leftarrow (\text{prédiction à l'aide de } K_2) \\ K_4 &= f(t_{n+1}, u_n + h K_3) \leftarrow (\text{prédiction à l'aide de } K_3) \end{aligned}$$

Ainsi, le schéma RK4 peut être écrit de la façon suivante :

$$\begin{cases} u_{n+1} = u_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4), \\ u_0 = y_0. \end{cases}$$

## 7.4 Champ de directions

Soient  $I \subset \mathbb{R}$  un intervalle réel et une EDO du premier ordre à résoudre :

$$y' = f(t, y),$$

où

- $t$  est la variable indépendante,  $t \in I$ ;
- $y = y(t)$  est une fonction solution **cherchée** et supposée continûment différentiable sur  $I$ ,  $y : I \rightarrow \mathbb{R}$ ;
- $f(t, y)$  est une fonction **donnée** de  $t$  et  $y$ ,  $f : I \times \mathbb{R} \rightarrow \mathbb{R}$ .

En dessinant en tout point  $(t, y)$  du domaine de définition  $I \times \mathbb{R}$  de  $f$  un segment de pente  $f(t, y)$ , on obtient ce que l'on appelle le **champ de directions** de l'EDO  $y' = f(t, y)$ .

Une **solution particulière**  $y = y(t)$  de l'EDO est partout tangente à ce champ de directions. Ainsi, de manière plus concise, on peut donner la définition suivante :

**Définition 7.13.** Un champ de directions est une représentation graphique de la fonction  $y' = f(t, y)$  avec  $t$  en abscisse et  $y$  en ordonnée.

**Exemple 7.14.** Considérons la fonction

$$f(t, y) = 2t$$

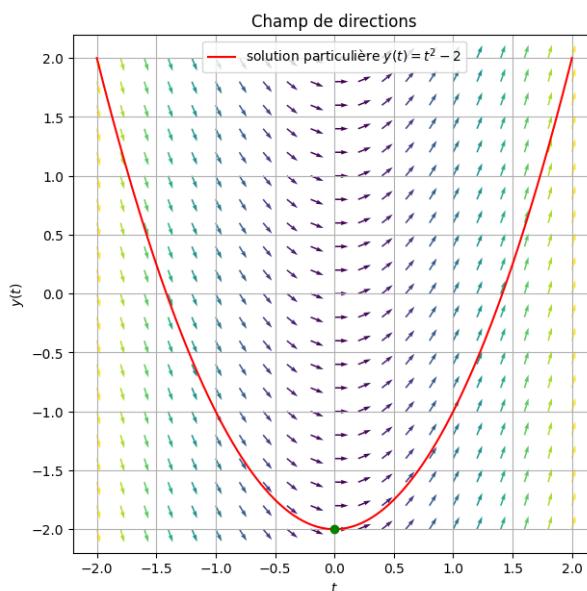
et le problème de Cauchy associé (sans préciser la condition initiale) :

$$y' = 2t.$$

La solution à ce problème est la famille de fonctions

$$y(t) = t^2 + A, \text{ où } A \text{ est une constante.}$$

Le champ de directions de  $y' = 2t$  permet de visualiser le comportement des fonctions solutions avant même d'effectuer le moindre calcul (analytique ou numérique) :



On remarque bien que, pour une valeur donnée de  $t$ , la pente de la fonction solution est la même pour toute valeur de  $y$  car  $y'$  ne dépend pas explicitement de  $y$ .

D'autre part, on vérifie que  $y(t) = t^2 - 2$  est ici une solution du problème de Cauchy considéré avec la condition initiale  $y(0) = -2$ . Cette solution particulière est en tout point de son graphe (dessiné ici en rouge) tangente au champ de directions.

◇

Dans le problème de Cauchy, le système évolue à partir de  $y_0$ , selon la dérivée  $y'$ . Le champ de directions est un outil qui permet de visualiser cette évolution. Il permet en quelque sorte de "visualiser" l'EDO.

## 7.5 Tableau de Butcher

Dans le cas d'**une partition régulière** de pas  $h$ , une méthode de Runge-Kutta (à un pas, explicite ou implicite) correspond à un **schéma général** de la forme :

$$u_{n+1} = u_n + h \sum_{i=1}^s b_i K_i,$$

où

- $s$  est le **nombre d'étapes** de la méthode,
- les  $s$  "pentes"  $K_i, i = 1, 2, \dots, s$ , sont données par

$$K_i = f(t_n + c_i h, u_n + h \sum_{j=1}^s a_{ij} K_j).$$

Les coefficients  $b_i, a_{ij}$  et  $c_i$  sont souvent donnés dans un **tableau de Butcher** de la forme :

$c_i$	$a_{ij}$
	$b_i$

Ainsi, ce tableau s'écrit pour le schéma RK4 :

$i=1$	0	0	0	0
$i=2$	$1/2$	$1/2$	0	0
$i=3$	$1/2$	0	$1/2$	0
$i=4$	1	0	0	1
	$1/6$	$1/3$	$1/3$	$1/6$
	$i=1$	$i=2$	$i=3$	$i=4$

Pour rappel, dans le schéma RK4, les quatre pentes suivantes interviennent :

$$\begin{aligned}
K_1 &= f(t_n, u_n) \leftarrow (\text{point de gauche}) \\
K_2 &= f\left(t_n + \frac{h}{2}, u_n + \frac{h}{2}K_1\right) \leftarrow (\text{prédition à l'aide d'Euler progressive}) \\
K_3 &= f\left(t_n + \frac{h}{2}, u_n + \frac{h}{2}K_2\right) \leftarrow (\text{prédition à l'aide de } K_2) \\
K_4 &= f(t_{n+1}, u_n + hK_3) \leftarrow (\text{prédition à l'aide de } K_3)
\end{aligned}$$

et la résolution numérique fait intervenir

$$\begin{cases} u_{n+1} = u_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4), \\ u_0 = y_0. \end{cases}$$

## 7.6 Stabilité et estimation de l'erreur

### 7.6.1 Stabilité

Nous allons étudier la **stabilité d'un schéma numérique** à partir d'un problème de Cauchy particulier :

$$\begin{cases} y'(t) = -\beta y, \text{ où } t > 0 \text{ et } \beta \in \mathbb{R}^+, \\ y(t_0) = y_0. \end{cases}$$

La solution à ce problème est la fonction

$$y(t) = y_0 \exp(-\beta t).$$

Numériquement, dans le cadre d'une partition régulière de pas  $h$ , le **schéma d'Euler progressif** s'écrit :

$$u_{n+1} = u_n + h(-\beta u_n) = (1 - \beta h)u_n, \text{ où } n = 0, 1, 2, \dots.$$

Ainsi,

$$u_{n+1} = (1 - \beta h)^{n+1}u_0, \text{ où } n = 0, 1, 2, \dots.$$

On remarque que, même si la solution du problème de Cauchy tend vers zéro lorsque  $t$  tend vers l'infini, la solution approchée lorsque  $n$  tend vers l'infini tend, en alternance, vers plus ou moins l'infini si  $u_0 \neq 0$  et  $1 - \beta h < -1$  :

$$(1 - \beta h)^n \xrightarrow{n \rightarrow \infty} \pm \infty.$$

Dans ce cas particulier, le schéma d'Euler progressif n'est **pas stable**.

Pour éviter cette **instabilité**, il est nécessaire de respecter la **condition de stabilité** suivante :

$$-1 < 1 - \beta h \Leftrightarrow h \leq \frac{2}{\beta}.$$

En appliquant le **schéma d'Euler rétrograde** au même problème de Cauchy particulier, il vient

$$\begin{aligned} u_{n+1} &= u_n + h(-\beta u_{n+1}) \Leftrightarrow (1 + \beta h)u_{n+1} = u_n \\ \Leftrightarrow u_{n+1} &= \frac{1}{1 + \beta h}u_n, \text{ où } n = 0, 1, 2, \dots \end{aligned}$$

Ainsi,

$$u_n = \frac{1}{(1 + \beta h)^n} u_0, \text{ où } n = 0, 1, 2, \dots,$$

et on observe que, pour tout  $h > 0$ , on a

$$\lim_{n \rightarrow \infty} u_n = 0.$$

Le schéma numérique d'Euler rétrograde est donc **stable** quelle que soit la valeur de  $h$ .

### 7.6.2 Erreur absolue commise

**Définition 7.15.** L'**erreur absolue commise** correspond à la valeur absolue de la **distance** (différence) au moment (point)  $t_{n+1}$  entre la valeur exacte  $y_{n+1}$  et la valeur approchée  $u_{n+1}$  est donnée par

$$d_{n+1} = y_{n+1} - u_{n+1}.$$

Deux types d'erreurs contribuent à l'erreur commise  $d_{n+1}$  :

- les **erreurs d'arrondi** qui correspondent à des représentations inexactes des nombres dans l'ordinateur;
- les **erreurs de troncature** qui sont la somme d'une **erreur locale** et d'une **erreur transportée** :

$$d_{n+1} = \underbrace{(y_{n+1} - u_{n+1}^*)}_{\text{a)}} + \underbrace{(u_{n+1}^* - u_{n+1})}_{\text{b)}},$$

où

- l'erreur de troncature **locale** correspond à l'erreur commise *sur une seule itération*, à partir de la valeur exacte au pas précédent;
- l'erreur de troncature **transportée** correspond aux erreurs accumulées depuis le temps initial.

La figure suivante illustre ces deux erreurs dans le cas de la méthode d'Euler progressive :

