UPMC Master P&A/SDUEE

UE MU4PY209

Méthodes Numériques et Calcul Scientifique

Résolution numérique des équations différentielles ordinaires (EDO)

2020–2021

albert.hertzog@upmc.fr

Table des matières

1 Introduction

1.1 Problème différentiel

— équation différentielle scalaire d'ordre $\boldsymbol{n}$

$$\boxed{\frac{\mathrm{d}^n y}{\mathrm{d}t^n} = f\left(t,\ y,\ \frac{\mathrm{d}y}{\mathrm{d}t},\ \ldots,\ \frac{\mathrm{d}^{n-1} y}{\mathrm{d}t^{n-1}}\right)}$$

où f est la fonction second membre donnée

$\Rightarrow$ **famille** de solutions $y(t)$ à $\boldsymbol{n}$ paramètres

— ensemble de $\boldsymbol{n}$ conditions imposées

$\Rightarrow$ choix d'**une** solution dans la famille

1.2 Deux types de problèmes différentiels à résoudre

— Conditions initiales données pour une seule valeur t_0 de t, par exemple

$$y(t_0) = y_0, \quad y'(t_0) = y'_0, \ \ldots \quad , \quad y^{(n-1)}(t_0) = y_0^{(n-1)}$$

Problème de **conditions initiales** ou de **Cauchy**

— Conditions données pour des valeurs distinctes de la variable indépendante t, par exemple :

$$y(t_0) = y_0, \quad y(t_1) = y_1, \ \ldots \quad , \quad y(t_{n-1}) = y_{n-1}$$

Problème de **conditions aux limites** (non traité, sauf problème de tir).

1.3 Équations différentielles scalaires du 1er ordre

Étudier d'abord les équations différentielles scalaires **du premier ordre**.

$\Rightarrow$ **famille de solutions** $y(t)$ à **un** paramètre (y_0)

$$\boxed{\frac{\mathrm{d}y}{\mathrm{d}t} = f(t,\ y(t)) \quad \text{avec} \quad y(t_0) = y_0} \quad \text{condition initiale}$$

Les EDO d'ordre supérieur se ramènent à des systèmes différentiels couplés du premier ordre (EDO vectorielles du premier ordre).

1.4 Unicité et problème bien posé : conditions suffisantes

La **condition de Lipschitz**

$$|f(t, y_2) - f(t, y_1)| \leqslant K \, |y_2 - y_1|$$

assure l'**unicité** de la solution.

$\left| \dfrac{\partial f}{\partial y}(t, y) \right| \leqslant K$ dans un domaine convexe $\Rightarrow$ condition de Lipschitz vérifiée.

Les erreurs d'arrondi amènent à **toujours résoudre un problème perturbé**.

Problème bien posé si : le problème faiblement perturbé (second membre ou condition initiale) possède une solution proche de celle du problème original.

La condition de Lipschitz assure que le problème est bien posé.

1.5 Méthodes de résolution numérique et notations

Résolution numérique approchée sur l'intervalle $[t_0, t_0 + L]$ de longueur L

Discrétisation par découpage de l'intervalle de longueur L selon un pas constant h

Échantillonnage de la solution aux instants $t_i = t_0 + ih$ pour $1 \leqslant i \leqslant n$.
Solution numérique : $u_i =$ approximation de $y(t_i)$

À partir de la condition initiale $u_0 = y(t_0)$ imposée,

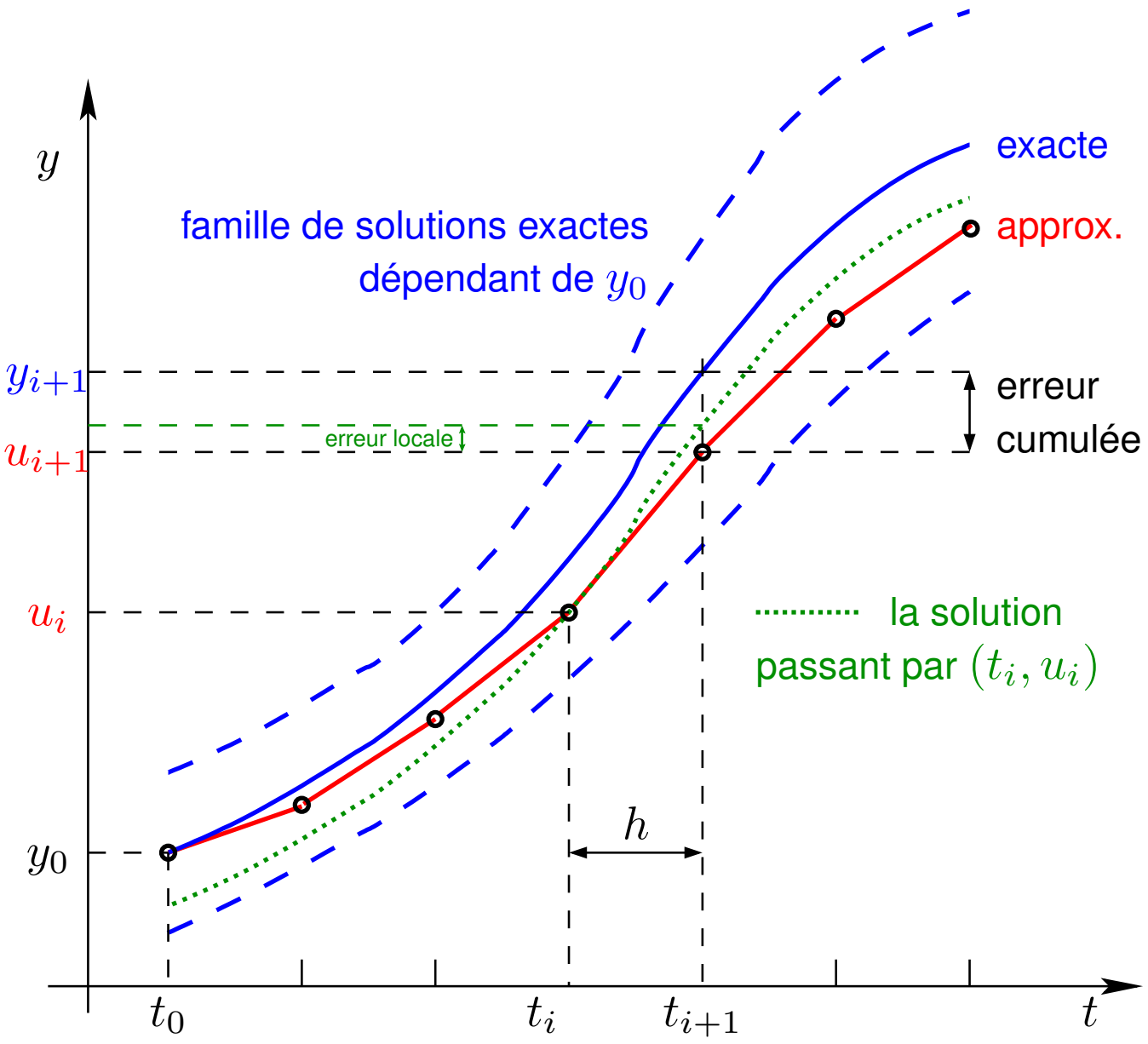
faire une **boucle** sur les abscisses t_i pour calculer l'approximation u_{i+1} à t_{i+1}
$\rightarrow$ approximer ainsi de proche en proche la solution sur l'intervalle L.

$\Rightarrow$ accumulation des erreurs dans la boucle

À chaque pas de la boucle, pour calculer u_{i+1}, on peut s'appuyer :

— sur **la dernière valeur** calculée u_i : **méthodes à un pas**

— sur **plusieurs valeurs** $u_{i-k} (k \geqslant 0)$ antérieurement calculées :
méthodes à plusieurs pas (initialisation nécessaire par méthode à un pas)



Méthode à pas constant

Découpage de l'intervalle de longueur L selon un pas fixe $h = L/n$.

$u_i =$ approximat. de $y(t_i)$

Un pas :

$t_i \to t_{i+1}$

$u_i \to u_{i+1}$

2 Méthodes à un pas

Constituent l'algorithme de base qui permet d'estimer la valeur de la solution à l'instant $t_{i+1} = t_i + h$, connaissant seulement u_i, celle à t_i.

La valeur à estimer peut être approchée par un développement limité de Taylor :

$$y(t_i + h) = y(t_i) + h\frac{\mathrm{d}y}{\mathrm{d}t}(t_i) + \frac{h^2}{2}\frac{\mathrm{d}^2y}{\mathrm{d}t^2}(t_i) + \cdots \tag{1}$$

Ordre n de la méthode $=$ plus grande puissance de h prise en compte dans l'approximation.

— Somme des termes négligés $=$ **erreur de troncature locale** $\propto h^{n+1}$ déterministe, augmente si le pas h augmente et si l'ordre de la méthode diminue

— Précision finie des opérations sur les réels $\Rightarrow$ **erreur d'arrondi** aléatoire augmente lorsque les calculs se compliquent, en particulier si le pas h diminue.

Indépendamment du coût (en temps de calcul) des opérations, et des cas où la fonction est tabulée, **ne pas croire que diminuer le pas améliore toujours** la qualité du résultat : un **compromis** doit être trouvé entre ces deux types d'erreurs.

2.1 Méthodes du premier ordre

2.1.1 Méthode d'Euler progressive (explicite)

Méthode du premier ordre d'intérêt pédagogique, à éviter en pratique

$$\boxed{u_{i+1} = u_i + hf(t_i, u_i)} \tag{2}$$

Exemple : stabilité

$$\frac{\mathrm{d}y}{\mathrm{d}t} = -\frac{y}{\tau} \quad \Rightarrow \quad \text{solution analytique} \quad y = y_0 e^{-t/\tau} \Rightarrow y_n = y_0 (e^{-h/\tau})^n$$

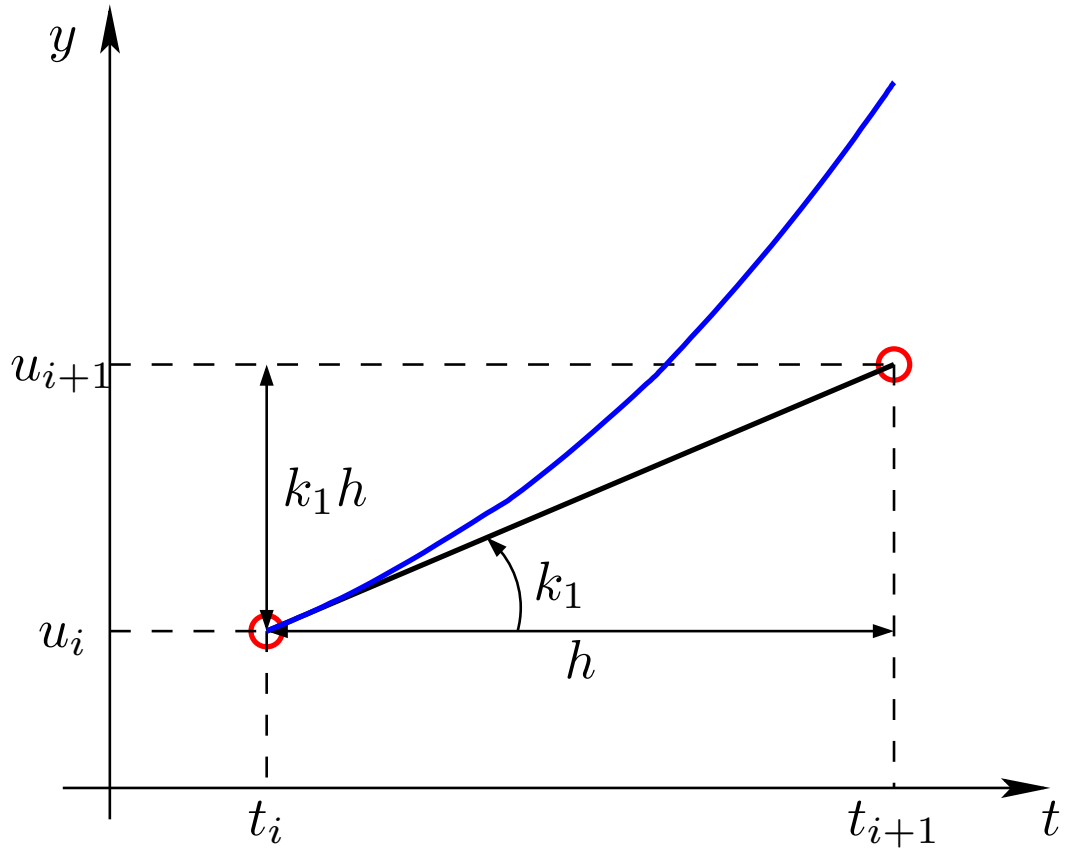
$$u_{i+1} = u_i - \frac{h}{\tau} u_i \quad \Rightarrow \quad \text{solution numérique} \quad u_n = y_0 \left(1 - h/\tau\right)^n$$

Si $\tau > 0$, la solution exacte vérifie $y(\infty) = 0$,

Mais pour l'approximation, $u_n \to 0 \Longleftrightarrow |1 - h/\tau| < 1 \Longleftrightarrow 0 < h < 2\tau$.

Condition de **stabilité** : $h < 2\tau$ (pas h petit)

Mais, si $h > \tau$, alors $(1 - h/\tau) < 0$: alternance de signe de la solution u_n.



Méthode d'Euler

Méthode **explicite** qui ne nécessite qu'une seule évaluation de la fonction second membre f par pas :

$k_1 = f(t_i, u_i)$

facilement **instable**

$$\frac{u_{i+1} - u_i}{h} = f(t_i, u_i)$$

voir dérivée avant

2.1.2 Méthode d'Euler rétrograde (implicite)

$$\boxed{u_{i+1} = u_i + hf(t_{i+1}, \boldsymbol{u_{i+1}})} \tag{3}$$

Méthode **implicite** : résolution itérative , plus difficile à mettre en œuvre, sauf si la forme de $f(t,u)$ permet le calcul analytique de u_{i+1} à partir de l'équation (3).

Avantage : meilleure **stabilité** que la méthode progressive explicite.

Exemple : stabilité

$$\frac{\mathrm{d}y}{\mathrm{d}t} = -\frac{y}{\tau} \quad \Rightarrow \quad \text{solution analytique} \quad y = y_0 e^{-t/\tau} \Rightarrow y_n = y_0 (e^{-h/\tau})^n$$

$$u_{i+1} = u_i - \frac{h}{\tau} u_{i+1} \quad \Rightarrow \quad \text{solution numérique} \quad u_{i+1} = \frac{u_i}{1 + h/\tau}$$

$$u_n = \frac{y_0}{(1 + h/\tau)^n}$$

Si $\tau > 0,\ y(\infty) = 0$, et aussi $u_n \to 0 \quad \forall \tau > 0, \forall h > 0$ solution **stable**

Mise en œuvre de la méthode d'Euler rétrograde : résolution de l'équation implicite par itération

$$\boxed{u_{i+1} = u_i + hf(t_{i+1}, \boldsymbol{u_{i+1}})}$$

Itérer l'application g pour rechercher son **point fixe**

$$v_2' = g(v_2) = u_i + hf(t_2, v_2)$$

Ce point fixe est la solution de l'équation implicite.

— Utilise plusieurs évaluations du second membre, sans calcul de ses dérivées.

— Très peu d'itérations nécessaires

Initialisation par le prédicteur avec Euler progressif

$$t_2 = t_i + h$$
$$k_1 = f(t_i, u_i)$$
$$v_2 = u_i + hk_1$$

Boucle pour recherche du point fixe de $g(v_2) = v_2' = u_i + hf(t_2, v_2)$

$$
\begin{aligned}
k_2 &= f(t_2, v_2) \\
v_2' &= u_i + hk_2 \\
\delta v_2 &= v_2' - v_2 \\
&\text{arrêt si} \quad |\delta v_2|^2 \leqslant \alpha^2 |v_2|^2 \\
v_2 &= v_2'
\end{aligned}
$$

La fonction g est contractante si $g'(v_2) = h \left| \frac{\partial f}{\partial v_2} \right| \leqslant 1$

ce qui est vérifié si le pas est assez faible.

Rappel : la condition de Lipschitz est $\left| \frac{\partial f}{\partial v_2} \right| \leqslant K$

Critère d'arrêt : choisir α faible, mais $\alpha > \varepsilon$.

2.2 Méthodes du deuxième ordre

Première idée : augmenter le nombre de termes du développement de Taylor :

rarement utilisé, car nécessite l'évaluation des dérivées partielles de f.

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t,\ y(t)) \qquad \Rightarrow \qquad \frac{\mathrm{d}^2y}{\mathrm{d}t^2} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t} = \frac{\partial f}{\partial t} + f\frac{\partial f}{\partial y} \tag{6}$$

Préférer utiliser **plusieurs évaluations du second membre** f en des points adaptés.

Centrer l'évaluation de la dérivée au point milieu $t_m = (t_i + t_{i+1})/2$.

$$y(t_i + h) = y(t_m) + \frac{h}{2}\frac{\mathrm{d}y}{\mathrm{d}t}(t_m) + \frac{1}{2}\frac{h^2}{4}\frac{\mathrm{d}^2y}{\mathrm{d}t^2}(t_m) + O(h^3) \tag{7a}$$

$$y(t_i) = y(t_m) - \frac{h}{2}\frac{\mathrm{d}y}{\mathrm{d}t}(t_m) + \frac{1}{2}\frac{h^2}{4}\frac{\mathrm{d}^2y}{\mathrm{d}t^2}(t_m) + O(h^3) \tag{7b}$$

Par différence, (approximation locale parabolique, voir aussi dérivée centrée à 2 termes)

$$y(t_i + h) - y(t_i) = h\frac{\mathrm{d}y}{\mathrm{d}t}(t_m) + O(h^3)$$

2.2.1 Méthode du point milieu

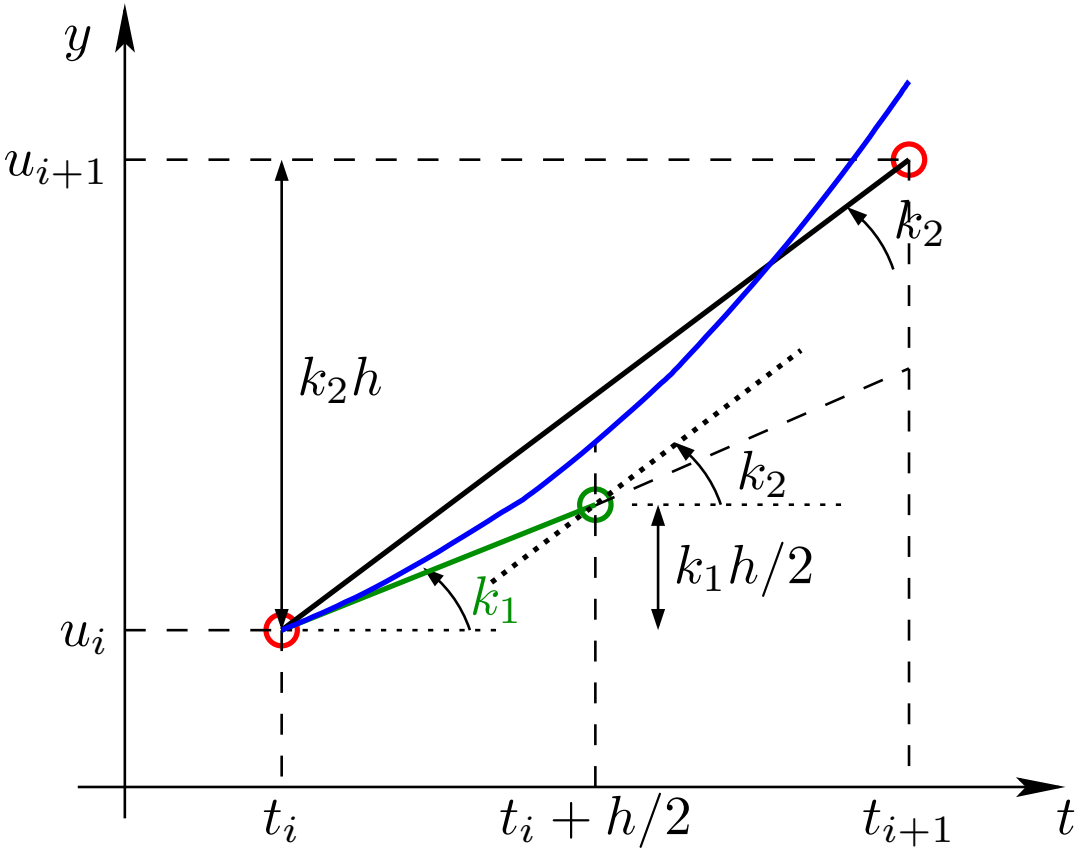
Nécessite l'évaluation du second membre f en 2 points :
en (t_i, u_i) et **au milieu** $(t_{i+1/2} = t_i + h/2,\ u_{i+1/2})$ d'un pas (hors grille).

$$u_{i+1} = u_i + hf\left(t_i + \frac{h}{2},\ u_i + \frac{h}{2} f(t_i, u_i)\right)$$

$$k_1 = f(t_i,\ u_i) \tag{8a}$$

($u_{i+1/2}$ calculé via Euler)
$$k_2 = f(t_i + \frac{h}{2},\ u_i + k_1 \frac{h}{2}) \tag{8b}$$

$$u_{i+1} = u_i + hk_2 \tag{8c}$$



Méthode du

point milieu

Méthode **explicite** qui nécessite deux évaluations du second membre par pas dont une hors grille.

2.2.2 Méthode d'Euler modifiée

En appliquant 7a et 7b à la dérivée et en faisant la somme, on peut remplacer la dérivée au milieu par la moyenne des dérivées aux extrémités de l'intervalle (voir méthode de quadrature dite des trapèzes) :

$$\frac{\mathrm{d}y}{\mathrm{d}t}(t_i) + \frac{\mathrm{d}y}{\mathrm{d}t}(t_{i+1}) = 2\frac{\mathrm{d}y}{\mathrm{d}t}(t_m) + O(h^2)$$

D'où une approximation n'utilisant pas la valeur de f au point milieu t_m :

$$u_{i+1} = u_i + \frac{h}{2}\left[f(t_i, u_i) + f(t_{i+1}, \boldsymbol{u_{i+1}})\right]$$

De nouveau, méthode a priori **implicite**, plus stable, mais plus lourde.

$\Rightarrow$ Contournement du problème en utilisant l'approximation d'Euler explicite (voir 2) pour évaluer u_{i+1} intervenant dans f.

$$\boxed{u_{i+1} = u_i + \frac{h}{2}\left[f(t_i, u_i) + f(t_{i+1}, u_i + hf(t_i, u_i))\right]}$$

Bilan : méthode de type **prédicteur-correcteur** équivalent à

— un demi-pas avec la pente initiale k_1

— et un demi-pas avec la pente k_2 du point prédit par Euler progressif.

$$k_1 = f(t_i,\ u_i) \tag{9a}$$

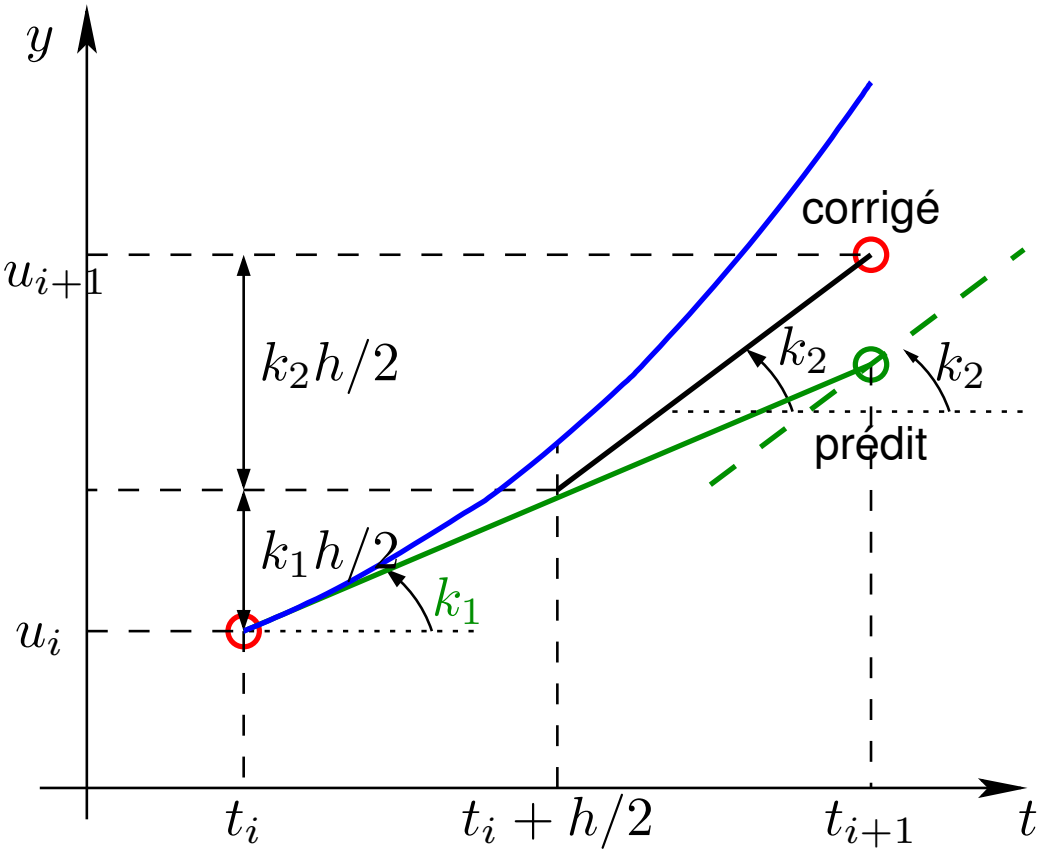
$$k_2 = f(t_{i+1},\ u_i + k_1 h) \tag{9b}$$

$$u_{i+1} = u_i + \frac{h}{2}\left[k_1 + k_2\right] \tag{9c}$$

Remarques

— deuxième ordre comme point milieu mais sans évaluation hors grille

— la résolution de l'équation implicite peut se faire en itérant la correction jusqu'à ce qu'elle devienne négligeable.



Méthode d'Euler modifiée

Méthode **explicite** qui nécessite deux évaluations de la fonction par pas en des points de la grille.

2.2.3 Méthode de Heun

$$k_1 = f(t_i,\ u_i) \tag{10a}$$

$$k_2 = f(t_i + \frac{2}{3}h,\ u_i + \frac{2}{3}k_1 h) \tag{10b}$$

$$u_{i+1} = u_i + \frac{h}{4}\left[k_1 + 3k_2\right] \tag{10c}$$

2.3 Méthodes de Runge Kutta

Plus généralement, avec r évaluations de f, on peut atteindre une méthode d'ordre r si $r \leqslant 4$. Pour atteindre l'ordre 5, six évaluations sont nécessaires.

$\Longrightarrow$ la méthode de Runge Kutta d'ordre 4 est très utilisée.

Les méthodes de Runge-Kutta sont **stables**.

2.3.1 Méthode de Runge Kutta d'ordre 3

$$k_1 = f(t_i, u_i) \tag{11a}$$

$$k_2 = f(t_i + \frac{h}{2},\ u_i + k_1 \frac{h}{2}) \tag{11b}$$

$$k_3 = f(t_i + h,\ u_i + (2k_2 - k_1)h) \tag{11c}$$

$$u_{i+1} = u_i + (k_1 + 4k_2 + k_3)\frac{h}{6} \tag{11d}$$

2.3.2 Méthode de Runge Kutta d'ordre 4

$$k_1 = f(t_i,\ u_i) \tag{12a}$$

$$k_2 = f(t_i + \frac{h}{2},\ u_i + k_1\frac{h}{2}) \tag{12b}$$

$$k_3 = f(t_i + \frac{h}{2},\ u_i + k_2\frac{h}{2}) \tag{12c}$$

$$k_4 = f(t_i + h,\ u_i + k_3 h) \tag{12d}$$

$$u_{i+1} = u_i + (k_1 + 2k_2 + 2k_3 + k_4)\frac{h}{6} \tag{12e}$$

2.4 Erreur absolue en fonction du pas et de l'ordre

nombre de pas $= L/h \quad \Longrightarrow \quad$ erreur globale $\sim$ erreur locale $\times\, L/h$

TABLE 1 – Erreur de **troncature seule**

Méthode	ordre	erreur locale	erreur globale
Euler explicite	1	$\propto h^2$	$\propto h$
Point milieu – Euler modifiée	2	$\propto h^3$	$\propto h^2$
Runge-Kutta 3	3	$\propto h^4$	$\propto h^3$
Runge-Kutta 4	4	$\propto h^5$	$\propto h^4$

Erreur d'**arrondi** locale indépendante de $h \Rightarrow$ erreur d'arrondi globale $\propto 1/h$

2.5 Exemple de l'équation logistique $\frac{\mathrm{d}y}{\mathrm{d}t} = y(1 - y/2)$

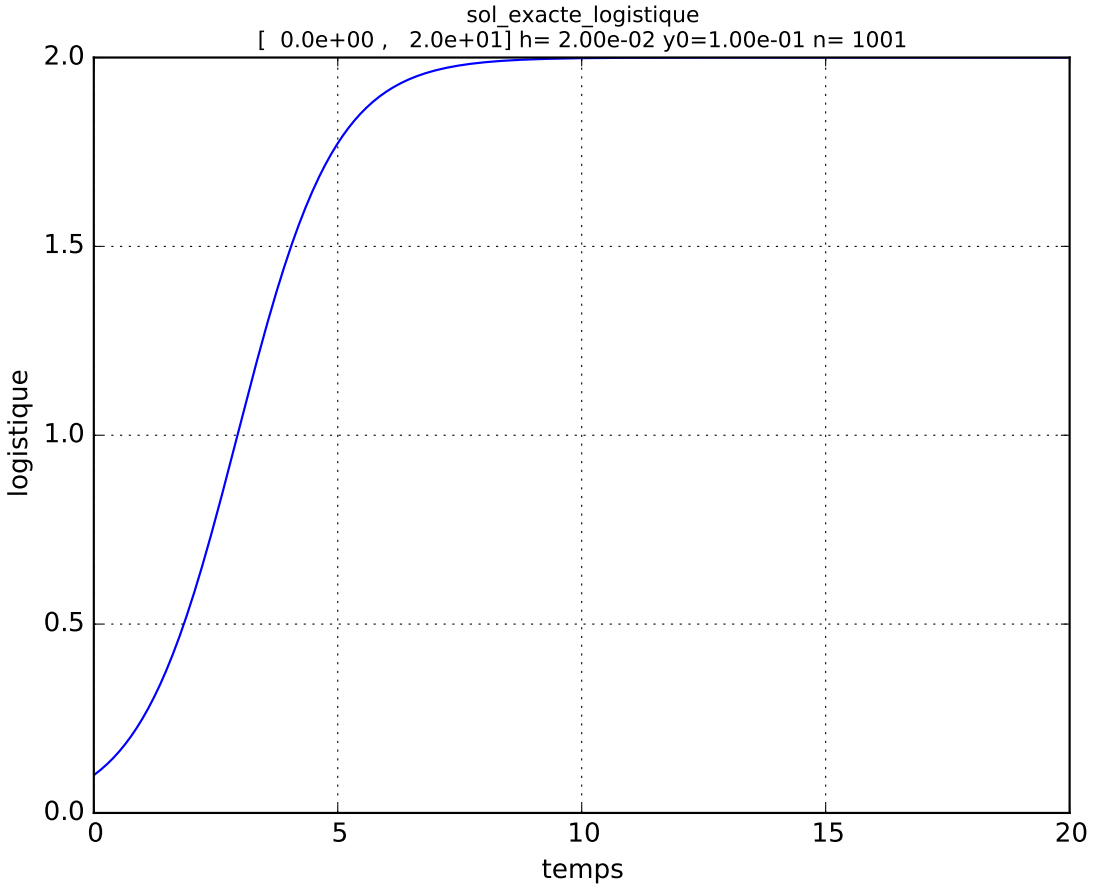


FIGURE 1 – Solution analytique de l'équation logistique pour $t_0 = 0$, $y(t_0) = 0.1$, $a = 1$ et $k = 2$.

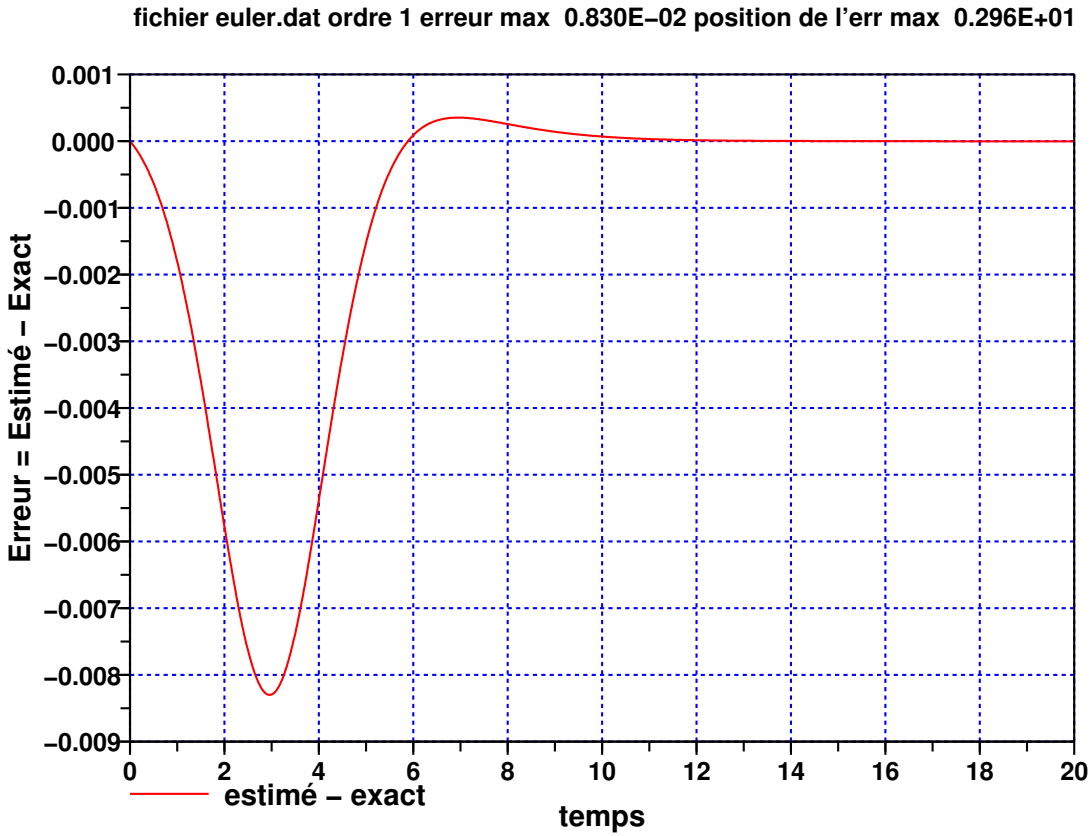


FIGURE 2 – Erreur dans l'intégration de l'équation logistique avec la méthode d'Euler pour $h = 0,02$. L'allure régulière montre que l'**erreur de troncature** domine.

Erreur de troncature locale liée à la courbure de la solution

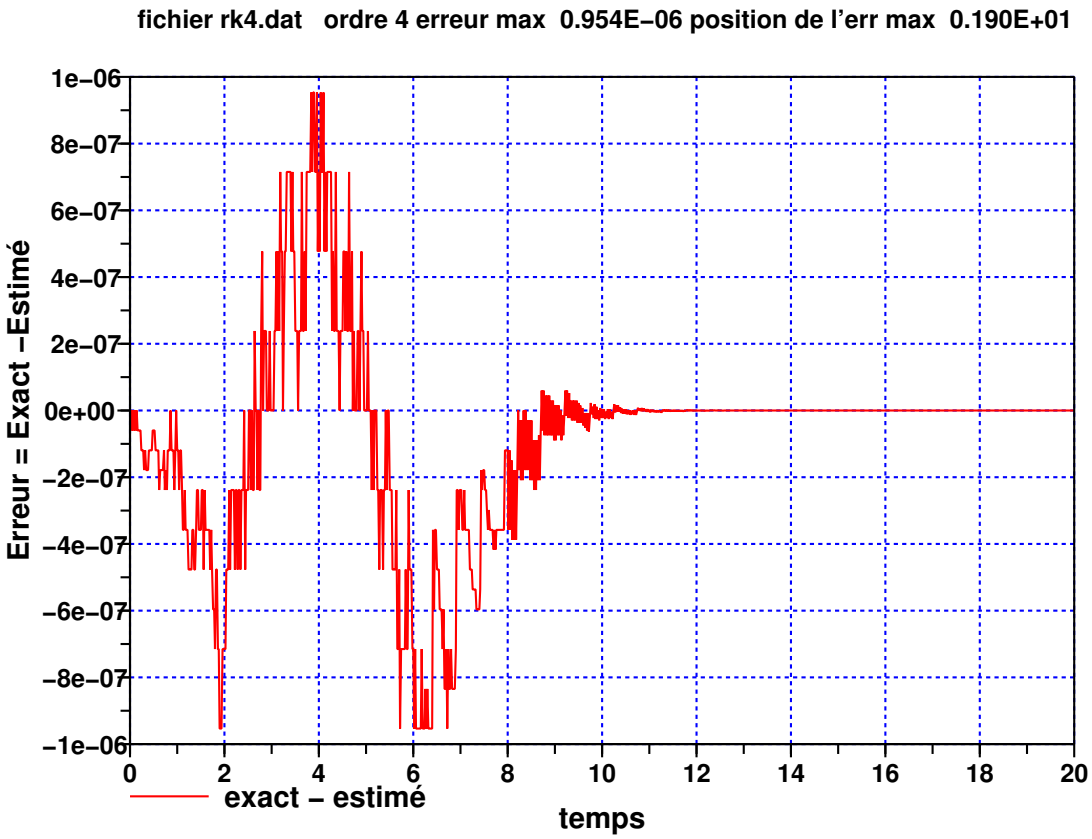
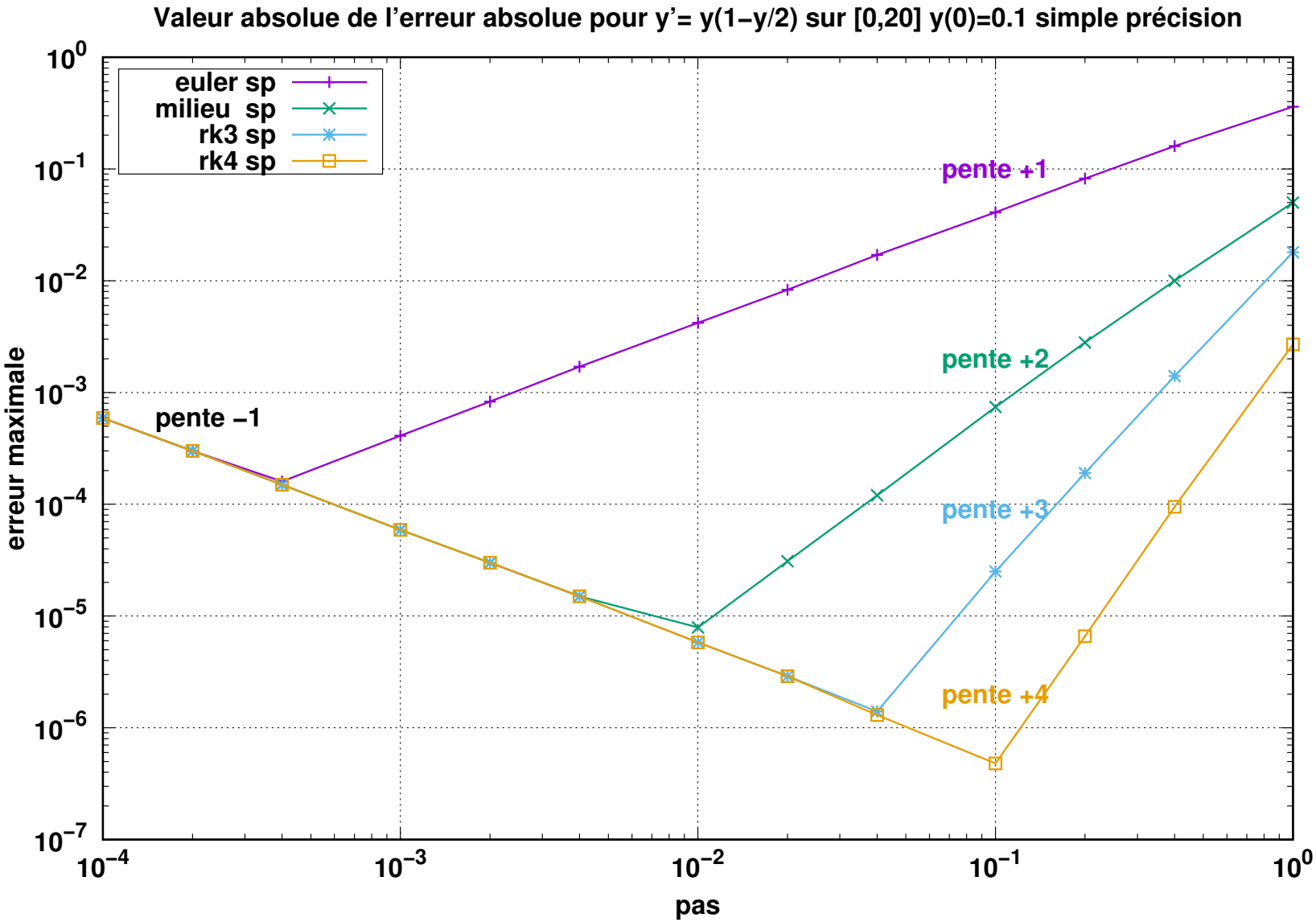
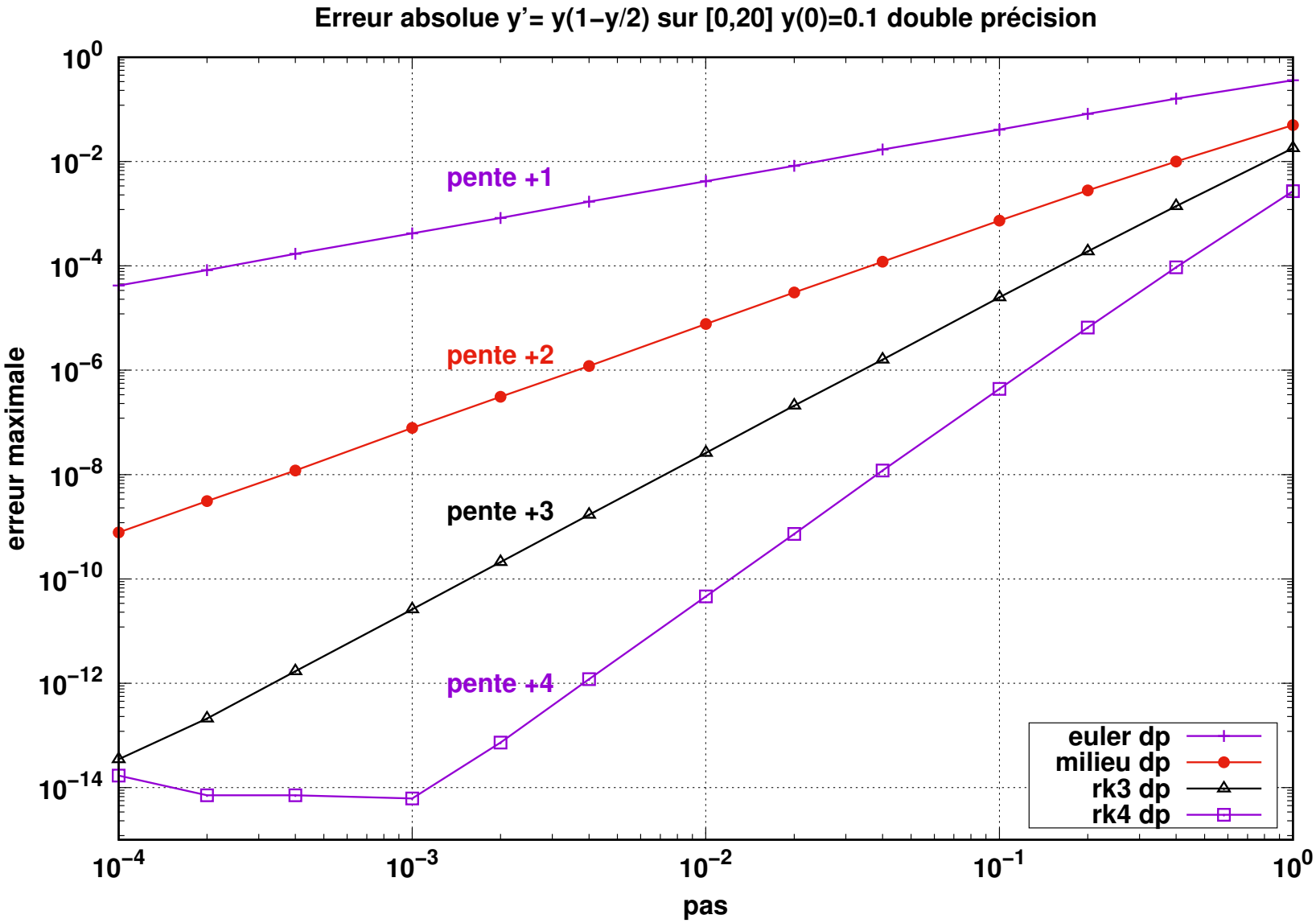


FIGURE 3 – Erreur dans l'intégration de l'équation logistique avec Runge Kutta d'ordre 4 pour $h = 0,02$. L'allure **bruitée** est caractéristique de l'**erreur d'arrondi** et on retrouve les niveaux de quantification des réels sur 32 bits.

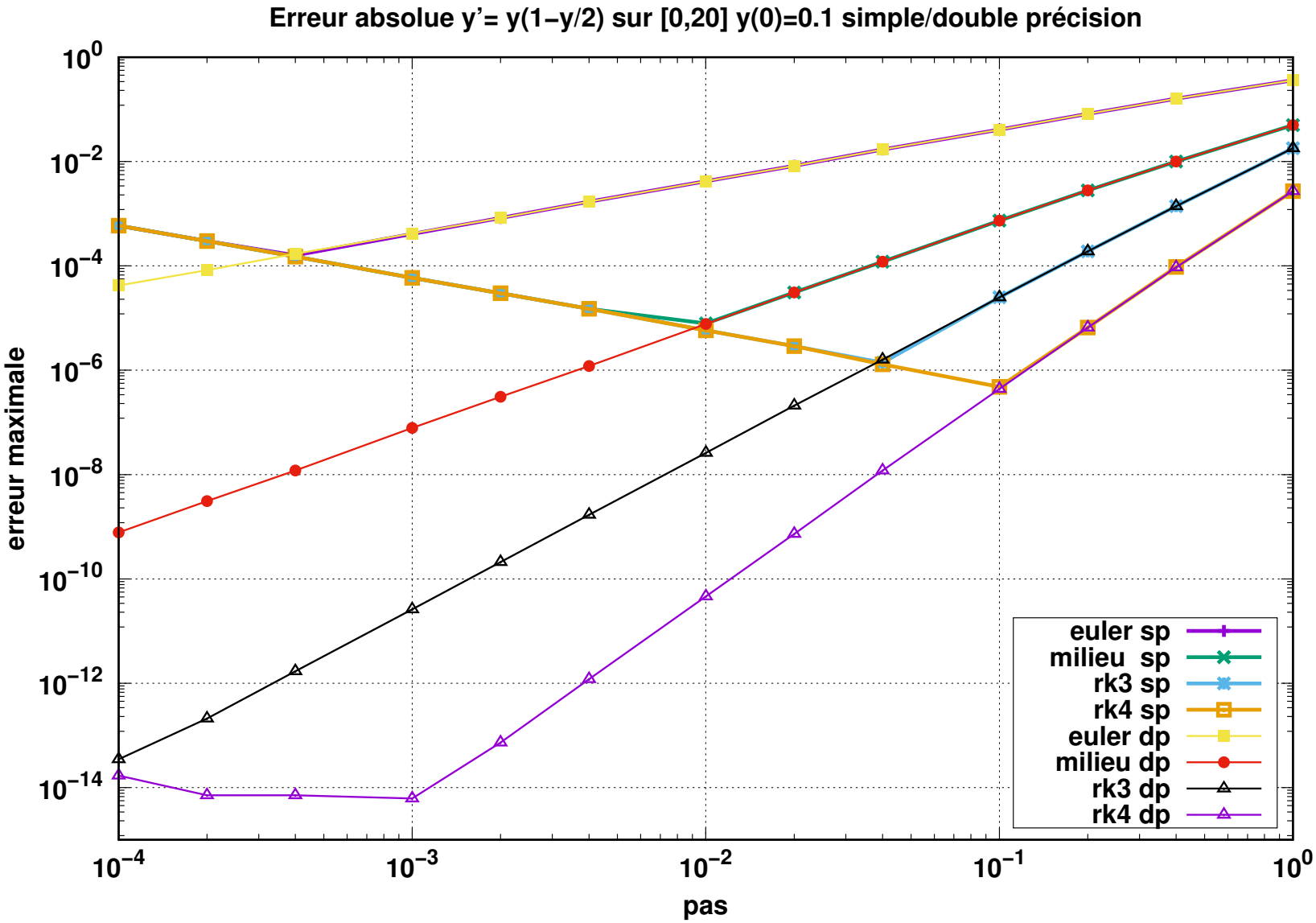
2.5.1 Exemple d'erreur totale maximale en simple précision (32 bits)



2.5.2 Exemple d'erreur totale maximale en double précision (64 bits)



2.5.3 Comparaison des erreurs maximales simple/double précision



3 Les EDO du premier ordre en pratique

3.1 Échelles de temps et problèmes raides

Ne pas oublier que chaque problème différentiel possède une ou plusieurs **échelles de temps propres** (périodes ou pseudo-périodes, constantes de temps).
La solution ne peut être représentée correctement qu'avec un pas assez inférieur au plus petit de ces temps propres.
Cette analyse impose donc une valeur maximale pour le pas.

Certains problèmes différentiels qualifiés de **raides** comportent des échelles de temps très différentes : leur intégration numérique s'avère délicate et coûteuse (pas faible pour respecter le temps court, mais nombreux pour accéder au temps long).
Il existe des méthodes spécifiques des EDO raides qui ne sont pas présentées ici.

3.2 Validation des résultats

Validation via une solution analytique d'un problème simplifié

Lorsqu'une solution analytique est disponible (par exemple pour certaines valeurs de paramètres qui permettent de simplifier l'EDO), sa comparaison avec la solution numérique permet de tester la méthode. Le calcul de l'erreur dans le domaine où la troncature domine permet d'extrapoler l'effet d'un changement de pas connaissant l'ordre de la méthode.

Validation sans solution analytique

Dans le cas où aucune solution analytique de référence n'est disponible, la validation s'appuie sur les mêmes outils que les méthodes adaptatives :

— diminution du pas (division par 2)
— augmentation de l'ordre de la méthode
— calcul d'invariants (énergie par exemple)

3.3 Structure des programmes de résolution d'EDO du 1er ordre

1. un fichier comportant **les fonctions seconds membres** de l'équation différentielle et les éventuelles solutions analytiques exactes ou approchées
2. un algorithme de base (appliquant une méthode d'ordre 1, 2, 3 ou 4 à la fonction second membre f passée en argument) permettant d'**avancer d'un pas** dans l'intégration de l'équation différentielle
3. un programme principal d'**interface avec l'utilisateur** qui choisit la méthode, le second membre, lit les paramètres (conditions initiales par ex.), déclenche et arrête **la boucle d'intégration** et stocke les résultats.
4. un fichier d'**utilitaires** notamment pour écrire les résultats dans un fichier pour visualisation ultérieure.

4 Systèmes d'équations différentielles du 1er ordre

4.1 Extension des méthodes scalaires explicites aux vecteurs

Système de n équations différentielles couplées du premier ordre associées à n conditions initiales

considérer les **vecteurs** $\vec{\mathbf{y}}$ et $\vec{\mathbf{f}}$.

$$
\begin{aligned}
\frac{\mathrm{d}y_1}{\mathrm{d}t} &= f_1(t,\ y_1, y_2, \ldots, y_n) \\
\frac{\mathrm{d}y_2}{\mathrm{d}t} &= f_2(t,\ y_1, y_2, \ldots, y_n) \\
\ldots &= \ldots \\
\frac{\mathrm{d}y_n}{\mathrm{d}t} &= f_n(t,\ y_1, y_2, \ldots, y_n)
\end{aligned}
$$

$$
\begin{pmatrix} y_1 \\ y_2 \\ \ldots \\ y_n \end{pmatrix} \quad \text{et} \quad \begin{pmatrix} f_1 \\ f_2 \\ \ldots \\ f_n \end{pmatrix}
$$

Les **méthodes explicites** de résolution des équations différentielles scalaires du premier ordre **s'appliquent aux systèmes**.

$$\frac{\mathrm{d}\vec{\mathbf{y}}}{\mathrm{d}t} = \vec{\mathbf{f}}(t,\ \vec{\mathbf{y}})$$

À chaque étape, effectuer les calculs **sur chaque composante** avant de passer à l'étape suivante : exemple avec point milieu

Étape 1 : vecteur des pentes au bord gauche de l'intervalle

$$\vec{\mathbf{k}_1} = \vec{\mathbf{f}}(t_1,\ \vec{\mathbf{y}_1})$$

$$\begin{aligned}
k_{1,1} &= f_1(t_1,\ y_{1,1},\ y_{1,2},\ \ldots,\ y_{1,n}) \\
k_{1,2} &= f_2(t_1,\ y_{1,1},\ y_{1,2},\ \ldots,\ y_{1,n}) \\
\ldots &= \ldots \\
k_{1,n} &= f_n(t_1,\ y_{1,1},\ y_{1,2},\ \ldots,\ y_{1,n})
\end{aligned}$$

avant de calculer...

Étape 2 : vecteur des pentes au point milieu prédit

$$\overrightarrow{\mathbf{k}_2} = \overrightarrow{\mathbf{f}}(t_1 + h/2,\ \overrightarrow{\mathbf{y}_1} + \overrightarrow{\mathbf{k}_1} h/2)$$

$$\begin{aligned}
k_{2,1} &= f_1(t_1 + h/2,\ y_{1,1} + k_{1,1}h/2,\ y_{1,2} + k_{1,2}h/2,\ \ldots,\ y_{1,n} + k_{1,n}h/2) \\
k_{2,2} &= f_2(t_1 + h/2,\ y_{1,1} + k_{1,1}h/2,\ y_{1,2} + k_{1,2}h/2,\ \ldots,\ y_{1,n} + k_{1,n}h/2) \\
\ldots &= \ldots \\
k_{2,n} &= f_n(t_1 + h/2,\ y_{1,1} + k_{1,1}h/2,\ y_{1,2} + k_{1,2}h/2,\ \ldots,\ y_{1,n} + k_{1,n}h/2)
\end{aligned}$$

Étape 3 : vecteur résultat au bord droit de l'intervalle

$$\overrightarrow{\mathbf{u}}_{i+1} = \overrightarrow{\mathbf{u}_i} + h\ \overrightarrow{\mathbf{k}_2}$$

5 Équations différentielles d'ordre supérieur

$$\frac{\mathrm{d}^n y}{\mathrm{d}t^n} = f\left(t,\ y,\ \frac{\mathrm{d}y}{\mathrm{d}t},\ \ldots,\ \frac{\mathrm{d}^{n-1} y}{\mathrm{d}t^{n-1}}\right)$$

Une EDO scalaire d'ordre n se ramène à un système de n équations différentielles du premier ordre couplées en posant :

$$\begin{pmatrix} y_1 \\ y_2 \\ \ldots \\ y_n \end{pmatrix} = \begin{pmatrix} y \\ y' \\ \ldots \\ y^{(n-1)} \end{pmatrix} \quad \Longrightarrow \quad \begin{pmatrix} y_1' \\ y_2' \\ \ldots \\ y_n' \end{pmatrix} = \begin{pmatrix} y_2 \\ y_3 \\ \ldots \\ f\left(t, y_1, y_2, \ldots, y_n\right) \end{pmatrix}$$

5.1 Exemple

Système linéaire du second ordre avec excitation $h(t)$

$$\frac{\mathrm{d}^2 y}{\mathrm{d}t^2} = a\frac{\mathrm{d}y}{\mathrm{d}t} + by + h(t)$$

Poser

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} y \\ y' \end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix} y_1' \\ y_2' \end{pmatrix} = \begin{pmatrix} y_2 \\ ay_2 + by_1 + h(t) \end{pmatrix}$$

Condition initiale vectorielle : position $y(t_0)$ et vitesse $y'(t_0)$

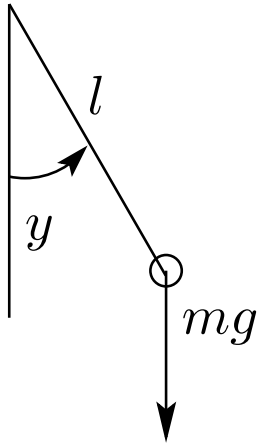
Remarque Système différentiel d'ordre p de dimension n

$\Rightarrow$ système différentiel couplé du premier ordre à np dimensions.

5.2 Exemple d'EDO d'ordre 2 : le pendule

Pendule non linéaire (y = position angulaire)

$$\boxed{\frac{\mathrm{d}^2 y}{\mathrm{d}t^2} = -k^2 \sin(y)} \quad \text{où} \quad k^2 = g/l \tag{13}$$



Pendule linéarisé (cas des petites amplitudes) : $\sin(y) \approx y$

$$\frac{\mathrm{d}^2 y}{\mathrm{d}t^2} = -k^2 y \tag{14}$$

l'équation linéarisée admet une solution analytique en $A\cos(kt) + B\sin(kt)$.

Exprimer cette EDO non linéaire du second ordre sous la forme d'un système différentiel couplé de dimension 2 mais du premier ordre.

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} y \\ y' \end{pmatrix} \quad \Longrightarrow \quad \begin{pmatrix} y_1' \\ y_2' \end{pmatrix} = \begin{pmatrix} y_2 \\ -k^2 \sin(y_1) \end{pmatrix}$$

Résolution système non-linéaire, avec le vecteur des valeurs initiales :

$$\begin{pmatrix} y_1(0) \\ y_2(0) \end{pmatrix} = \begin{pmatrix} y(0) \\ \dfrac{\mathrm{d}y}{\mathrm{d}t}(0) = a \end{pmatrix} = \begin{pmatrix} \text{position angulaire} \\ \text{vitesse angulaire} \end{pmatrix}$$

Énergie mécanique conservée (après division par ml^2) :

$$\frac{1}{2}\left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^2 + k^2(1-\cos y) = \text{constante}$$

Cas où $y(0) = 0$ (départ en position d'équilibre stable)

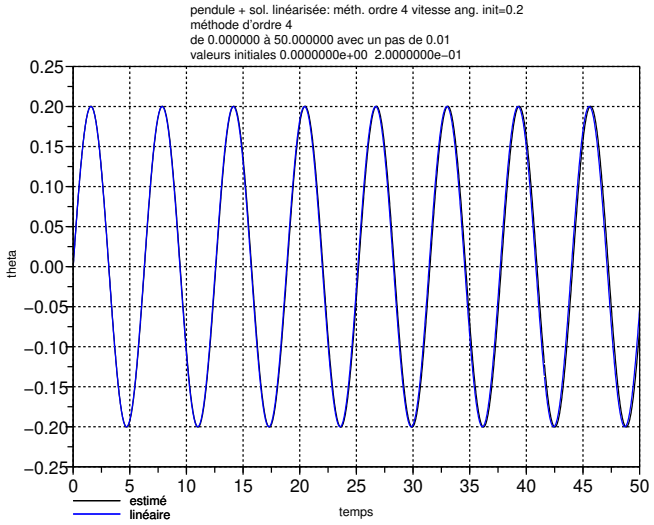
$$\frac{1}{2}\left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^2 + k^2(1-\cos y) = \frac{1}{2}\left(\frac{\mathrm{d}y}{\mathrm{d}t}(0)\right)^2$$

Vitesse angulaire minimale pour $y = \pi$ (position d'équilibre instable si atteinte).

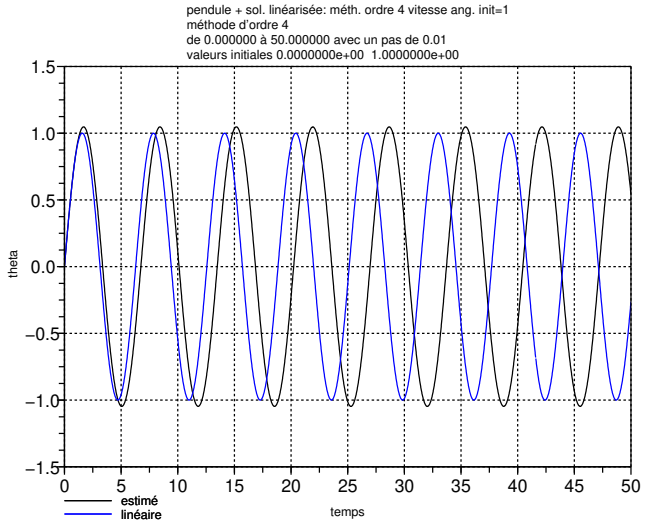
Si $a = \dfrac{\mathrm{d}y}{\mathrm{d}t}(0) > 2k$ (seuil) $\Rightarrow$ la vitesse angulaire ne s'annule pas (apériodique).

Étude de la transition périodique–apériodique selon a dans le cas où $k = 1$

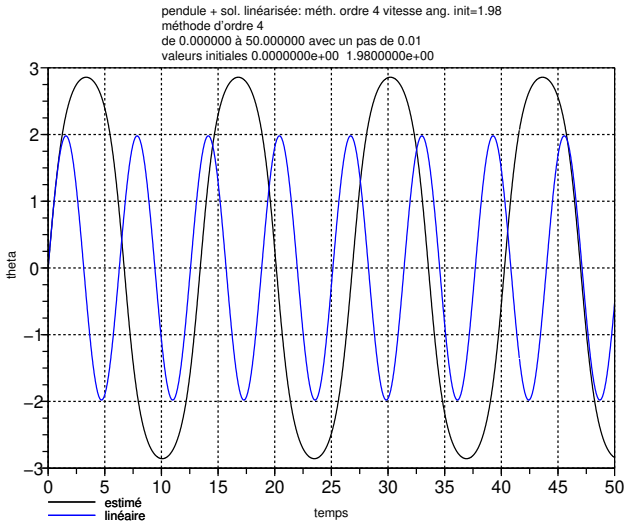
Comparaisons non-linéaire (Runge-Kutta 4)–analytique linéarisé : $y(t)$



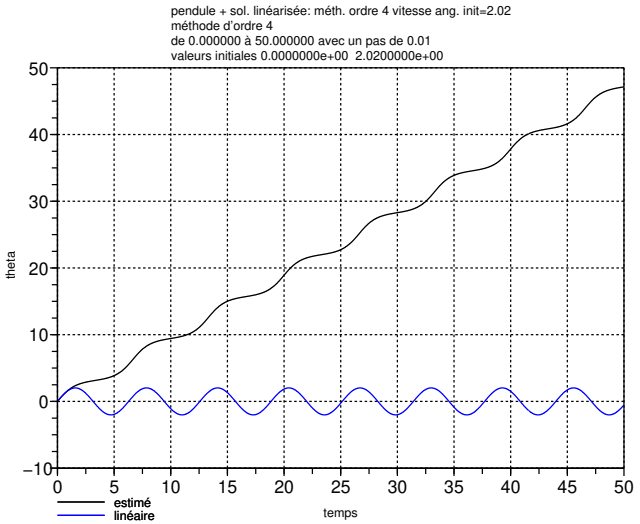
$a = 0.2 \ll 1$ linéarisable



$a = 1$ périodique non sinusoïdal

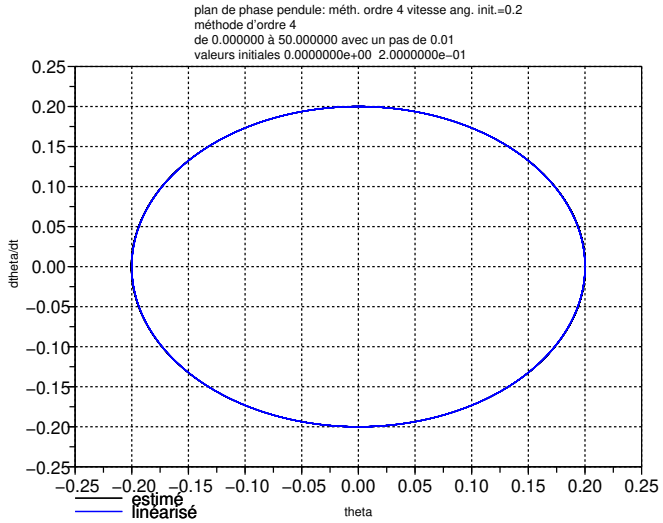


$a = 1.98$ périodique non sinusoïdal

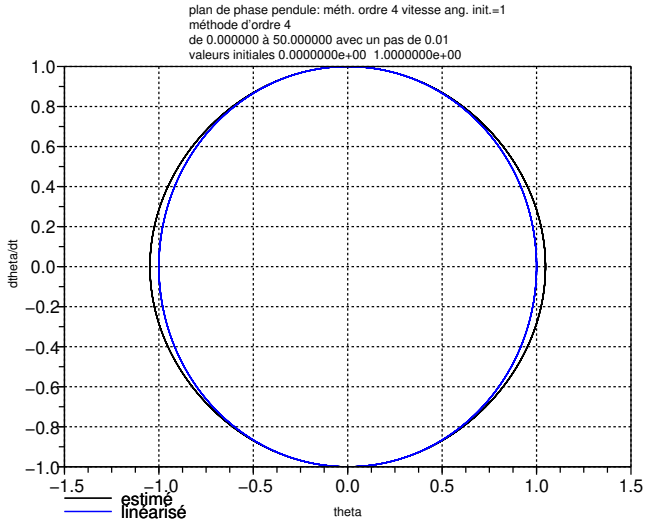


$a = 2.02$ apériodique

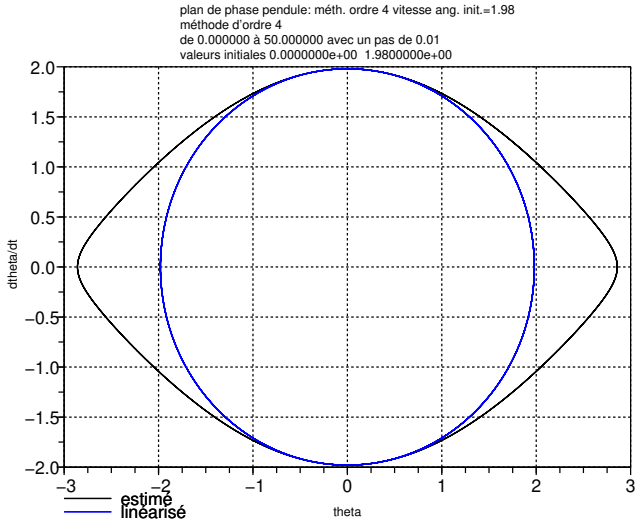
Comparaisons non-linéaire (RK 4)–analytique linéarisé : plan de phase $y'(y)$



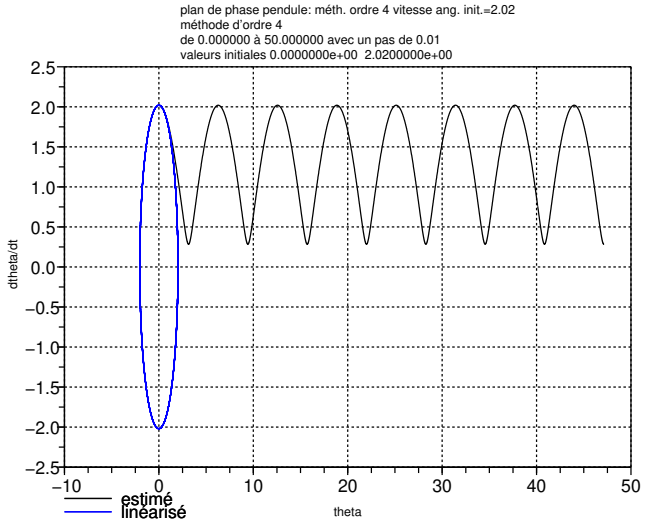
$a = 0.2 \ll 1$ linéarisable



$a = 1$ périodique non sinusoïdal

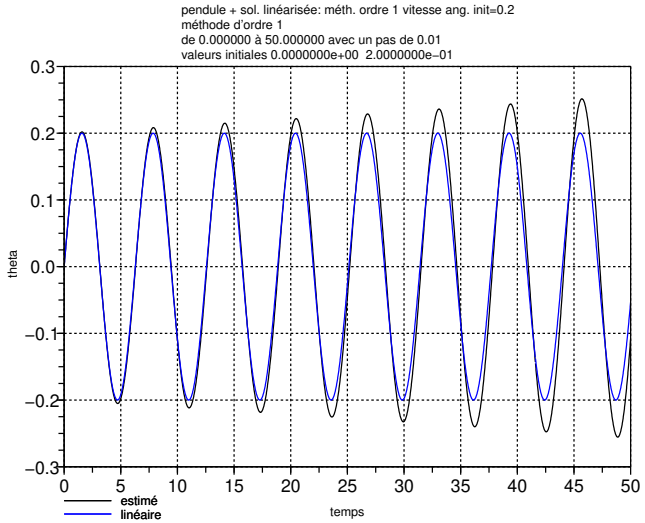


$a = 1.98$ périodique non sinusoïdal

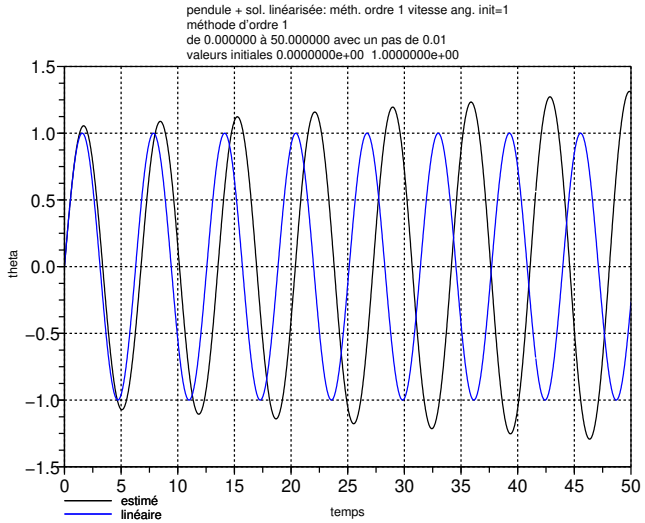


$a = 2.02$ apériodique

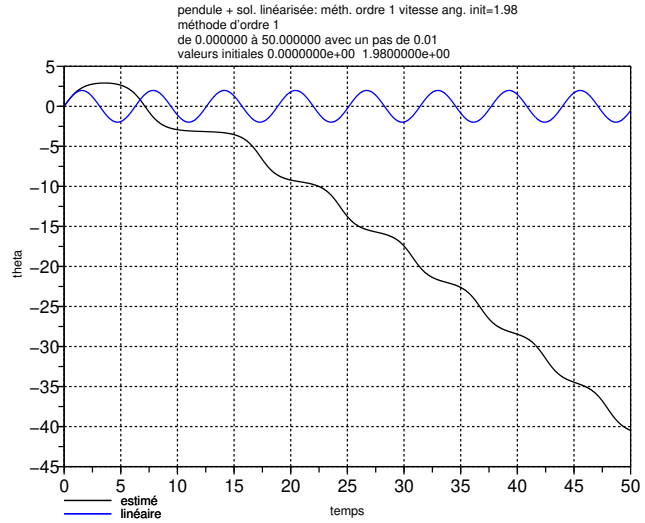
Comparaisons non-linéaire (Euler)–analytique linéarisé $y(t)$



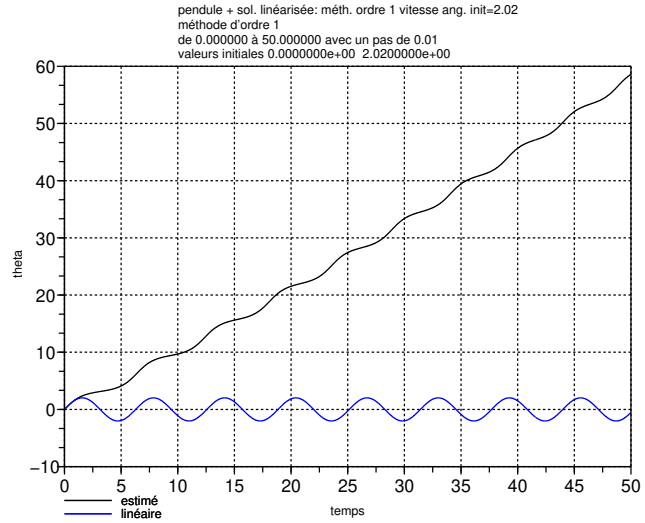
$a = 0.2$



$a = 1$

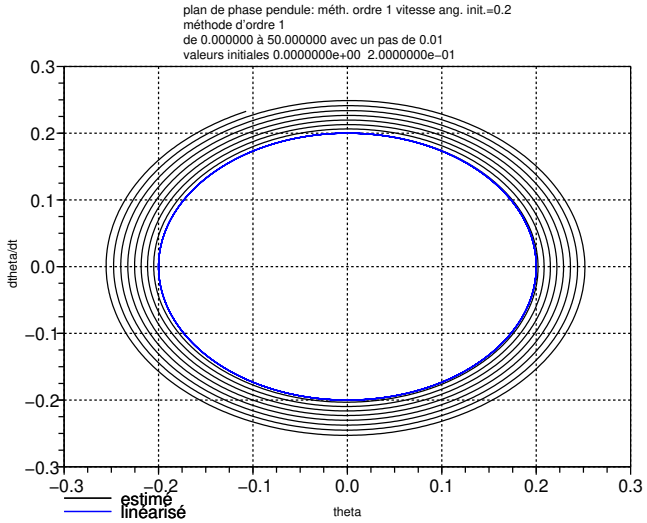


$a = 1.98$ apériodique selon Euler !

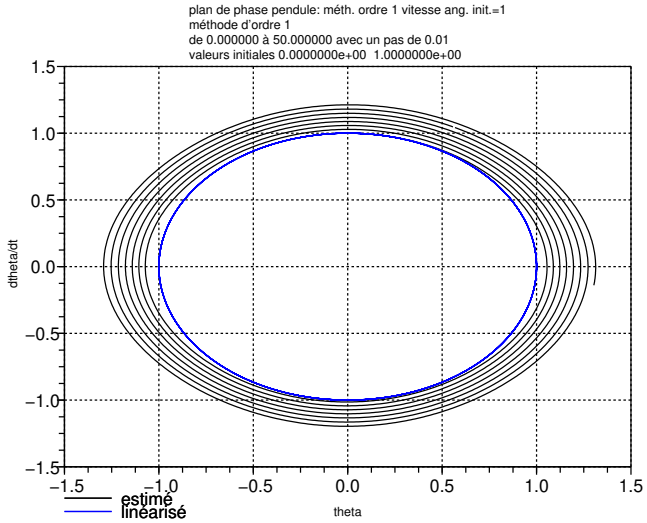


$a = 2.02$

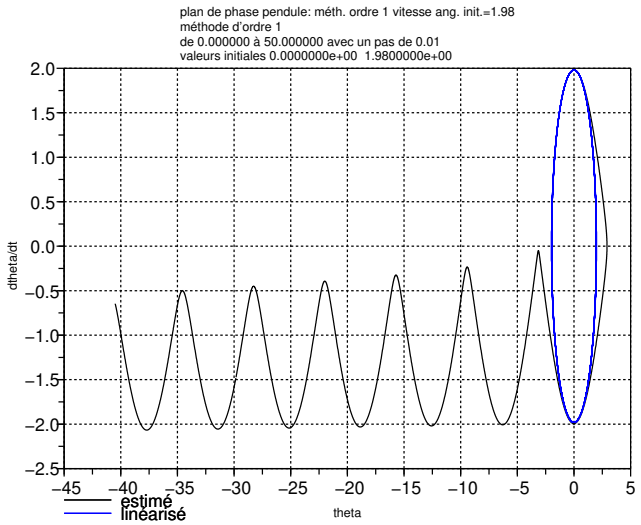
Comparaisons non-linéaire (Euler)–analytique linéarisé : plan de phase $y'(y)$



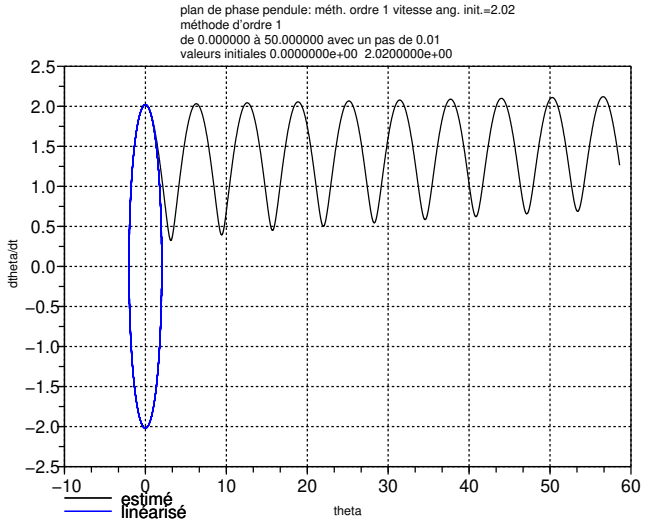
$a = 0.2$



$a = 1$



$a = 1.98$ apériodique selon Euler !



$a = 2.02$

Stabilité à long terme avec Euler progressive et rétrograde

Pendule linéarisé sans frottement représenté dans l'espace des phases : comportement à long terme d'un système non dissipatif

Même méthode sur les 2 composantes (position et vitesse) $h = 0.025$

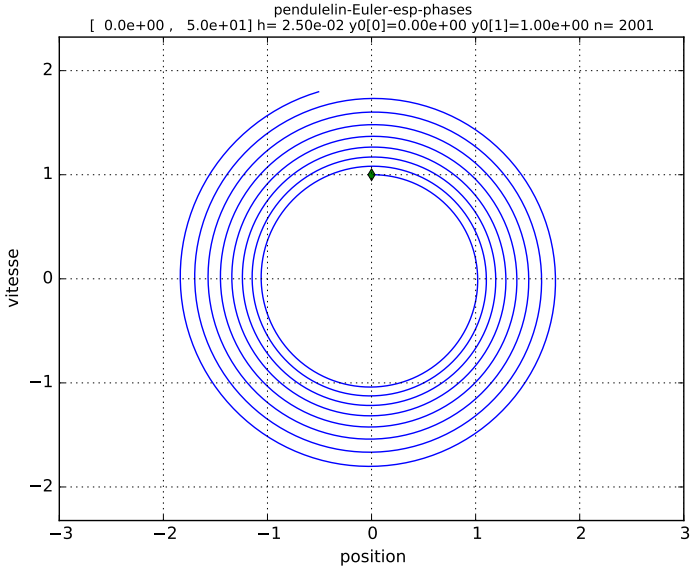


FIGURE 4 – Euler progressive : instable, amplification

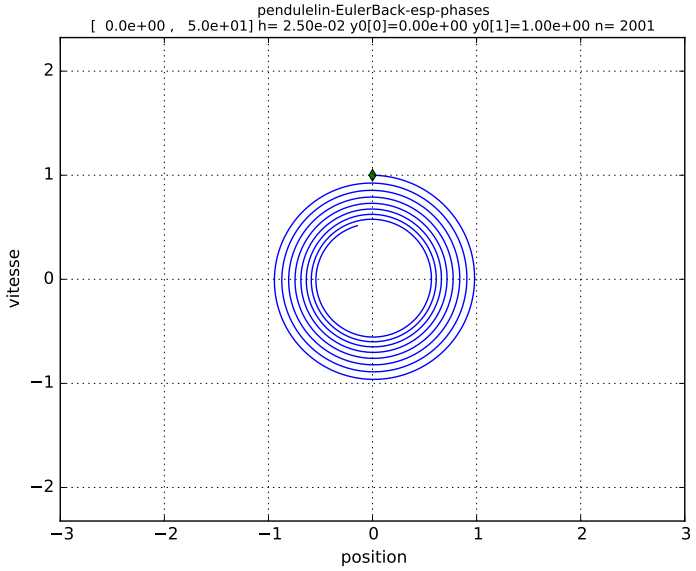


FIGURE 5 – Euler rétrograde : stable, contraction

Méthode d'Euler symplectique Méthodes progressives et méthodes rétrogrades présentent chacune des inconvénients antagonistes : dans le cas d'un système à deux composantes, on peut espérer les compenser en appliquant une méthode progressive sur z_1 et rétrograde sur z_2.

$$z_1(t+h) = z_1(t) + h\frac{\mathrm{d}z_1}{\mathrm{d}t}(t,\ z_1(t),\ z_2(t)) \tag{15}$$

$$z_2(t+h) = z_2(t) + h\frac{\mathrm{d}z_2}{\mathrm{d}t}(t+h,\ z_1(t+h),\ \boxed{z_2(t+h)}\ ) \tag{16}$$

Dans le cas d'un système séparable, la méthode rétrograde ne nécessite plus d'itération car $\frac{\mathrm{d}z_2}{\mathrm{d}t}$ ne dépend pas de z_2 et $z_1(t+h)$ a été calculé avant.

$$z_1(t+h) = z_1(t) + h\frac{\mathrm{d}z_1}{\mathrm{d}t}\ (t,\ \ z_2(t)) \tag{17}$$

$$z_2(t+h) = z_2(t) + h\frac{\mathrm{d}z_2}{\mathrm{d}t}(t+h,\ z_1(t+h)) \tag{18}$$

Exemple de méthode symplectique : alternance des méthodes progressive et rétrograde entre position et vitesse

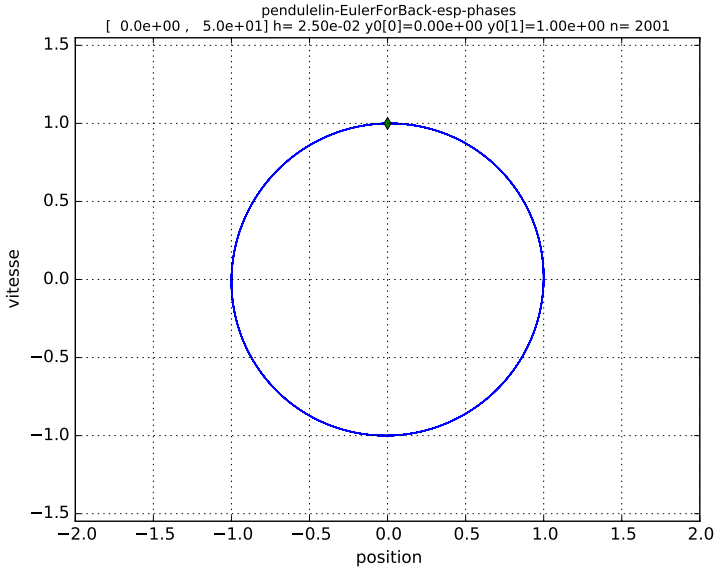


FIGURE 6 – Euler mixte

— progressive sur position,

— rétrograde sur vitesse.

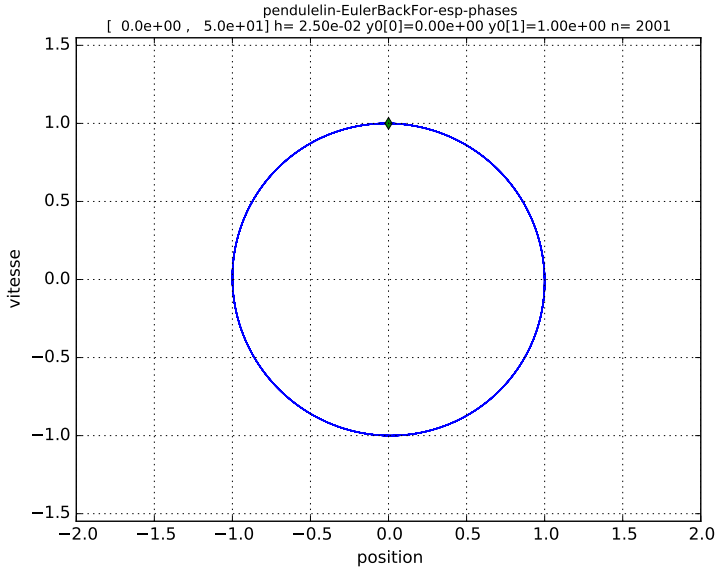


FIGURE 7 – Euler mixte

— rétrograde sur position,

— progressive sur vitesse.

6 Mise en œuvre vectorielle des méthodes à un pas

6.1 Introduction

— Les méthodes d'intégration doivent fonctionner **quelle que soit la taille p des vecteurs** qui représentent la solution $\vec{\mathbf{y}}$ et le second membre $\vec{\mathbf{f}}$ de l'EDO.

— C'est **le programme principal qui fixera cette taille**.

— Dans la fonction second membre, les tailles des tableaux des seconds membres effectifs seront héritées du programme principal et **non déclarées explicitement**.

Cependant seules les p composantes effectives de $\vec{\mathbf{f}}$ (2 pour le pendule : dérivée et dérivée seconde) seront calculées à partir des p composantes de $\vec{\mathbf{y}}$.

6.2 En C++ avec Eigen

Fonctions de type **`Eigen::VectorXf`** de taille p déterminée à l'exécution.

Déclaration tardive des `VectorXf`, une fois p connu dans le programme principal.

1. **la fonction second membre de l'équation différentielle** remplit le tableau **`second_mb`** de taille p représentant $\vec{\mathbf{f}}(\vec{\mathbf{y}}, t)$

```
// version C++ avec librairie Eigen
Eigen::VectorXf dydt_pendule(float t, Eigen::VectorXf y) {
  Eigen::VectorXf dydt(y.size()); // p = dimension de y(t)
  dydt(0) = y(1);
  dydt(1) = -sin(y(0));
  return dydt;
}
```

2. **pour chacune des méthodes à un pas** (Euler, point milieu et Runge Kutta) les pentes $\vec{k_i}$ seront des variables `Eigen::VectorXf`, locales à la fonction-méthode ; le résultat $\vec{u}_{i+1}$, qui est aussi vectoriel, sera retournée par la méthode (ce sera donc son type).

```
// C++ avec la librairie Eigen
VectorXf milieu(VectorXf u_i, float t_i, float h,
                function<VectorXf(float, VectorXf)> dydt) {
   // pentes, tailles non précisées ici
   VectorXf k1, k2;
   k1 = dydt(t_i, u_i); // dimensionnement implicite par affectation
   k2 = ...; // calcul de k2 à partir de k1
   return u_i + h*k2;
}
```

3. **Dans le programme principal** (et dans la procédure d'écriture sur fichier), les solutions vectorielles (analytique et numérique) sont représentées par des **`Eigen::MatrixXf`**. Mais la dimension temporelle n'est pas « vue » par les méthodes : elles travaillent sur des vecteurs de taille p dans un intervalle $[t_i, t_{i+1}]$. Pour respecter l'ordre par défaut de rangement des éléments utilisé par Eigen : **le premier indice (ligne) sera celui des composantes, le second (colonne) celui du temps**.

```
// MatrixXf déclarées dans le main
// p choisi selon la dimension du second membre
 MatrixXf u(p,n);  //  p equations et n instants
// dans la boucle sur les instants i :
// appel de la méthode du point milieu par ex.
 u.col(i+1) = milieu(u.col(i), t(i), pas, dydt);
// u.col(i+1) et u.col(i) : vecteurs à  p composantes
// t(i) : scalaire
```

Références

AKAI, TERRENCE J., *Applied Numerical Methods for Engineers*, 410 pages (Wiley, 1994), ISBN 0-471-57523-2.

BURDEN, RICHARD L. et J. DOUGLAS FAIRES, *Numerical Analysis*, 847 pages (Thompson, Brooks/Cole, 2005), huitième édition, ISBN 0-534-40499-5.

DEMAILLY, J.-P., *Analyse numérique et équations différentielles*, 350 pages (EDP Sciences, 2006), troisième édition, ISBN 978-2-86883-891-9.

GUILPIN, CH., *Manuel de calcul numérique appliqué*, 577 pages (EDP Sciences, 1999), ISBN 2-86883-406-X.

RAPPAZ, JACQUES et MARCO PICASSO, *Introduction à l'analyse numérique*, 268 pages (Presses polytechniques et universitaires romandes, 2010), ISBN 978-2-88074-851-7.