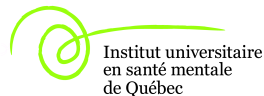




## Chapitre 7 : Résolution numérique des équations différentielles

Ibrahima Dione (Université Laval)

13 décembre 2016



1. André Fortin: *Analyse numérique pour ingénieurs*, Presses Internationales Polytechnique, Montréal, Québec (2016).

## ① Introduction

## ② Méthode d'Euler

## ③ Méthodes de Taylor

## ④ Méthodes de Runge-Kutta

### 4.2 Méthodes de Runge-Kutta d'ordre 2

Méthode d'Euler modifiée

Méthode du point milieu

### 4.3 Méthode de Runge-Kutta d'ordre 4

## ⑤ Systèmes d'équations différentielles

### 5.2 La méthode d'Euler explicite

### 5.3 La méthode de Runge-Kutta d'ordre 4

## ⑥ Équations d'ordre supérieur

# Introduction

La tâche dans ce chapitre consiste à déterminer une approximation de la fonction  $y(t)$ , solution du système

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(t_0) = y_0 \end{cases} \quad (1.1)$$

- ❶ La variable indépendante  $t$  représente très souvent (mais pas toujours) **le temps**.
- ❷ La variable dépendante est notée  $y$  et dépend bien sûr de  $t$ .
- ❸ La fonction  $f$  est pour le moment une fonction quelconque de deux variables et suffisamment différentiable.
- ❹ La condition  $y(t_0) = y_0$  est **la condition initiale**, c'est à dire l'état de la solution au moment où l'on commence à s'y intéresser.

## Définition 1.1

L'équation (1.1) est appelée **équation différentielle d'ordre 1**, car seule la dérivée d'ordre 1 de la variable dépendante  $y(t)$  est présente. Si des dérivées de  $y(t)$  d'ordre 2 apparaissaient dans l'équation (1.1), on aurait une équation différentielle d'ordre 2, et ainsi de suite.

## Exemple 1

Soit l'équation différentielle du premier ordre

$$\begin{cases} y'(t) = t \\ y(0) = 1 \end{cases} \quad (1.2)$$

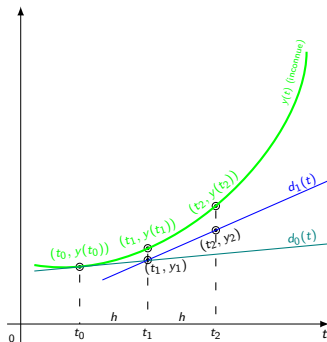
La solution, obtenue en intégrant la première équation du système (1.2), est  $y(t) = \frac{t^2}{2} + 1$ .

Par contre dans ce chapitre, on utilisera des méthodes numériques dans la résolution de cette équation différentielle. Ces outils numériques de résolution ne permettent pas d'obtenir une solution pour toutes les valeurs de la variable indépendante  $t$ , mais plutôt une approximation de la solution analytique seulement à **certaines valeurs de  $t$  notées  $t_i$**  et distancées d'une valeur  $h_i = t_{i+1} - t_i$ . Dans la plupart des méthodes présentées, cette distance est constante pour tout  $i$  et est notée  $h$ . On appelle  $h$ , **le pas de temps**.

### Remarque(s) :

- 1 On note  $y(t_i)$  **la solution analytique** de l'équation différentielle (1.1) en  $t = t_i$ .
- 2 Alors que  $y_i$  représente **la solution approximative** en  $t = t_i$  obtenue à l'aide d'une méthode numérique.

# Méthode d'Euler



Méthode d'Euler explicite

**Question:** À partir de l'équation différentielle (1.1) (avec la condition initiale  $y(t_0) = y_0$ ), comment obtenir une approximation de  $y(t_1)$  en  $t = t_1 = t_0 + h$  ?

- 1 On détermine, à partir du point  $(t_0, y_0)$ , la direction (ou pente) à la quelle on va avancer en choisissant celle-ci

$$y'(t_0) = f(t_0, y(t_0)) = f(t_0, y_0)$$

- 2 On suit donc la droite  $d_0(t)$ , passant par  $(t_0, y_0)$  et ayant comme pente  $f(t_0, y_0)$ , définie analytiquement par

$$d_0(t) = y_0 + f(t_0, y_0)(t - t_0)$$

- 3 On prend ainsi  $d_0(t_1)$  comme étant une approximation de la solution analytique  $y(t_1)$  :

$$\begin{aligned} y(t_1) &\simeq y_1 = d_0(t_1) = y_0 + f(t_0, y_0)(t_1 - t_0) \\ &= y_0 + hf(t_0, y_0) \end{aligned}$$

- 4 On refait l'analyse précédente à partir du point  $(t_1, y_1)$  pour approximer  $y(t_2)$ , et ainsi de suite

$$y(t_2) \simeq y_2 = d_1(t_2) = y_1 + hf(t_1, y_1)$$

## Remarque(s) :

**Note:** On constate que l'erreur commise à la première itération est réintroduite dans les calculs de la deuxième.

Ce-ci met en évidence une propriété importante des méthodes numériques de résolution des équations différentielles. En effet, **l'erreur introduite à la première itération a des répercussions sur les calculs de la deuxième itération**, ce qui signifie que les erreurs se propagent d'une itération à l'autre. Il en résulte que l'erreur  $|y(t_n) - y_n|$  augmente légèrement avec  $n$ .

### Algorithme 2.1: Méthode d'Euler explicite :

- ① Pour un pas de temps  $h$ , une condition initiale  $(t_0, y_0)$  et un nombre d'itérations  $N$
- ② Pour  $0 \leq n \leq N$  :
  - \*  $y_{n+1} = y_n + hf(t_n, y_n)$
  - \*  $t_{n+1} = t_n + h$
  - \* Écrire  $t_{n+1}$  et  $y_{n+1}$
- ③ Arrêt

## Exemple 2

À traiter en classe !

## Définition 2.1

- ① Une méthode de résolution d'équations différentielles est dite à un pas si elle est de la forme :

$$y_{n+1} = y_n + h\phi(t_n, y_n) \quad (2.1)$$

où  $\phi$  est une fonction quelconque. Une telle relation est appelée **équation aux différences**. La méthode est à un pas si, pour obtenir la solution en  $t = t_{n+1}$ , on doit utiliser la solution numérique au temps  $t_n$  seulement.

La méthode d'Euler explicite est bien sûr une méthode à un pas où  $\phi(t, y) = f(t, y)$ .

**Note:** On désigne **méthodes à pas multiples** les méthodes qui exigent également la solution numérique aux temps  $t_{n-1}, t_{n-2}, t_{n-3}, \dots$ .

- ② On dira qu'un schéma à un pas converge à l'ordre  $p$  si

$$\max_{1 \leq n \leq N} |y(t_n) - y_n| = \mathcal{O}(h^p) \quad (2.2)$$

où  $N$  est le nombre total de pas de temps.

- ③ L'**erreur de troncature locale** au point  $t = t_n$  est définie par

$$\tau_{n+1}(h) = \frac{y(t_{n+1}) - y(t_n)}{h} - \phi(t_n, y(t_n)) \quad (2.3)$$

**Note:** L'erreur de troncature locale mesure la précision avec laquelle la solution analytique vérifie l'équation aux différences (2.1).

## Méthodes de Taylor

La formule suivante est la base de la méthode de Taylor

$$y(t_{n+1}) = y(t_n) + f(t_n, y(t_n))h + \frac{h^2}{2} \left( \frac{\partial f(t_n, y(t_n))}{\partial t} + \frac{\partial f(t_n, y(t_n))}{\partial y} f(t_n, y(t_n)) \right) + \mathcal{O}(h^3) \quad (3.1)$$

En négligeant les termes d'ordre supérieur ou égal à 3, on en arrive à poser

$$y(t_{n+1}) \simeq y(t_n) + f(t_n, y(t_n))h + \frac{h^2}{2} \left( \frac{\partial f(t_n, y(t_n))}{\partial t} + \frac{\partial f(t_n, y(t_n))}{\partial y} f(t_n, y(t_n)) \right) \quad (3.2)$$

### Algorithme 3.1: Méthode de Taylor d'ordre 2 :

- ① Pour un pas de temps  $h$ , une condition initiale  $(t_0, y_0)$  et un nombre d'itérations  $N$
- ② Pour  $0 \leq n \leq N$  :
  - \*  $y_{n+1} = y_n + hf(t_n, y_n) + \frac{h^2}{2} \left( \frac{\partial f(t_n, y_n)}{\partial t} + \frac{\partial f(t_n, y_n)}{\partial y} f(t_n, y_n) \right)$
  - \*  $t_{n+1} = t_n + h$
  - \* Écrire  $t_{n+1}$  et  $y_{n+1}$
- ③ Arrêt

**Remarque(s) :** Suivant la notation (2.1), on a ici  $\phi(t, y(t)) = f(t, y(t)) + \frac{h}{2} \left( \frac{\partial f(t, y(t))}{\partial t} + \frac{\partial f(t, y(t))}{\partial y} f(t, y(t)) \right)$ . En vertu de la relation (3.1) et de la définition de l'erreur de troncature locale (2.3), on montre que  $\tau_{n+1}(h) = \mathcal{O}(h^2)$ . L'erreur de troncature locale de la méthode de Taylor est d'ordre 2 et la méthode converge à l'ordre 2 ( $p = 2$  dans la relation (2.2)).



## Méthodes de Runge-Kutta d'ordre 2

Bien que d'ordre 2, la méthode de Taylor exige d'évaluer les dérivées de la fonction  $f(t, y(t))$ , nécessitant ainsi de calculs supplémentaires :

$$y(t_{n+1}) = y(t_n) + f(t_n, y(t_n))h + \frac{h^2}{2} \left( \frac{\partial f(t_n, y(t_n))}{\partial t} + \frac{\partial f(t_n, y(t_n))}{\partial y} f(t_n, y(t_n)) \right) + \mathcal{O}(h^3) \quad (4.1)$$

Il existe cependant un moyen de contourner cette difficulté en développant les méthodes de Runge-Kutta en remplaçant cette dernière relation par une expression équivalente possédant le même ordre de précision  $\mathcal{O}(h^3)$  de la forme

$$y(t_{n+1}) = y(t_n) + a_1 h f(t_n, y(t_n)) + a_2 h f(t_n + a_3 h, y(t_n) + a_4 h) \quad (4.2)$$

La détermination des paramètres  $a_1, a_2, a_3$  et  $a_4$  de telle sorte que les expressions (4.1) et (4.2) aient toutes deux une erreur en  $\mathcal{O}(h^3)$ , procurera deux variantes de la méthode de Runge-Kutta d'ordre 2.

**Remarque(s) :** On ne trouve par ailleurs aucune dérivée partielle dans l'expression (4.2).

# Méthode d'Euler modifiée

En choisissant les valeurs  $a_1 = a_2 = \frac{1}{2}$ ,  $a_3 = 1$  et  $a_4 = f(t_n, y(t_n))$  dans l'équation (4.2), on obtient alors l'algorithme suivant dit de Méthode d'Euler modifiée

## Algorithme 4.1: Méthode d'Euler modifiée :

- ① Étant donné un pas de temps  $h$ , une condition initiale  $(t_0, y_0)$  et un nombre maximal d'itérations  $N$
- ② Pour  $0 \leq n \leq N$  :
  - \*  $\hat{y} = y_n + hf(t_n, y_n)$
  - \*  $y_{n+1} = y_n + \frac{h}{2} (f(t_n, y_n) + f(t_n + h, \hat{y}))$
  - \*  $t_{n+1} = t_n + h$
  - \* Écrire  $t_{n+1}$  et  $y_{n+1}$
- ③ Arrêt

**Remarque(s) :** Pour faciliter les calculs, l'évaluation de  $y_{n+1}$  a été scindée en deux étapes. La variable temporaire  $\hat{y}$  correspond tout simplement à une itération de la méthode d'Euler explicite. On fait ainsi une prédiction  $\hat{y}$  de la solution en  $t_{n+1}$  qui est corrigée (et améliorée) à la deuxième étape de l'algorithme. On parle alors d'une **méthode de prédiction-correction**.

## Méthode du point milieu

En remplaçant les valeurs des coefficients  $a_1 = 0$ ,  $a_2 = 1$ ,  $a_3 = \frac{1}{2}$  et  $a_4 = \frac{f(t_n, y(t_n))}{2}$  dans l'équation (4.2), on obtient l'algorithme de la Méthode du point milieu suivant.

### Algorithme 4.2: Méthode du point milieu :

- ① Pour un pas de temps  $h$ , une condition initiale  $(t_0, y_0)$  et un nombre d'itérations  $N$
- ② Pour  $0 \leq n \leq N$  :
  - \*  $k_1 = hf(t_n, y_n)$
  - \*  $y_{n+1} = y_n + h \left( f \left( t_n + \frac{h}{2}, y_n + \frac{k_1}{2} \right) \right)$
  - \*  $t_{n+1} = t_n + h$
  - \* Écrire  $t_{n+1}$  et  $y_{n+1}$
- ③ Arrêt

### Remarque(s) :

- ① L'algorithme précédent illustre bien pourquoi cette méthode est dite du point milieu. En effet la fonction  $f(t, y)$  est évaluée au point milieu de l'intervalle  $[t_n, t_{n+1}]$ .
- ② Les méthodes d'Euler modifiée et du point milieu étant du même ordre de troncature locale, leur précision est semblable. D'autres choix sont possibles pour les coefficients  $a_i$ , mais nous nous limitons aux deux précédents.

## Méthode de Runge-Kutta d'ordre 4

En reprenant le développement de Taylor de la fonction  $f$ , mais cette fois jusqu'à l'ordre 5, un raisonnement similaire à celui qui a mené aux méthodes de Runge-Kutta d'ordre 2 aboutit à la méthode de Runge-Kutta d'ordre 4, qui représente un outil d'une grande utilité.

### Algorithme 4.3: Méthode de Runge-Kutta d'ordre 4 :

- ① Étant donné un pas de temps  $h$ , une condition initiale  $(t_0, y_0)$  et un nombre maximal d'itérations  $N$
- ② Pour  $0 \leq n \leq N$  :
  - \*  $k_1 = hf(t_n, y_n)$
  - \*  $k_2 = hf(t_n + \frac{h}{2}, y_n + \frac{k_1}{2})$
  - \*  $k_3 = hf(t_n + \frac{h}{2}, y_n + \frac{k_2}{2})$
  - \*  $k_4 = hf(t_n + h, y_n + k_3)$
  - \*  $y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$
  - \*  $t_{n+1} = t_n + h$
  - \* Écrire  $t_{n+1}$  et  $y_{n+1}$
- ③ Arrêt

**Remarque(s) :** La méthode de Runge-Kutta d'ordre 4 est très fréquemment utilisée en raison de sa grande précision qui est mise en évidence dans l'exemple suivant

### Remarque(s) :

- 1 On a constaté que plus l'ordre d'une méthode est élevé, plus la méthode est précise.
- 2 Par contre, plus l'ordre de la méthode est élevé, plus elle est coûteuse en temps de calcul. Par exemple, la méthode d'Euler explicite (d'ordre 1) ne nécessite qu'une seule évaluation de la fonction  $f(t, y)$  à chaque pas de temps, alors que la méthode d'Euler modifiée (d'ordre 2) en demande 2 et que la méthode de Runge-Kutta d'ordre 4 exige 4 évaluations de la même fonction.

En d'autres termes, la méthode de Runge-Kutta d'ordre 4 demande à peu près deux fois plus de calculs que la méthode d'Euler modifiée et quatre fois plus que la méthode d'Euler explicite.

**Question:** N'est-t-il pas préférable d'utiliser la méthode d'Euler explicite avec un pas de temps 4 fois plus petit ou la méthode d'Euler modifiée d'ordre 2 avec un pas de temps 2 fois plus petit, plutôt que de se servir de la méthode de Runge-Kutta d'ordre 4 ?

### Exemple 3

## Systèmes d'équations différentielles

Considérons la forme générale d'un système de  $m$  équations différentielles avec conditions initiales

$$\left\{ \begin{array}{ll} y_1'(t) = f_1(t, y_1(t), y_2(t), \dots, y_m(t)) & (y_1(t_0) = y_{1,0}) \\ y_2'(t) = f_2(t, y_1(t), y_2(t), \dots, y_m(t)) & (y_2(t_0) = y_{2,0}) \\ y_3'(t) = f_3(t, y_1(t), y_2(t), \dots, y_m(t)) & (y_3(t_0) = y_{3,0}) \\ \vdots & \\ y_m'(t) = f_m(t, y_1(t), y_2(t), \dots, y_m(t)) & (y_m(t_0) = y_{m,0}) \end{array} \right. \quad (5.1)$$

Vectoriellement, le système (5.1) peut être écrit sous la forme

$$\left\{ \begin{array}{ll} \vec{y}'(t) &= \vec{f}(t, \vec{y}(t)) \\ \vec{y}(t_0) &= \vec{y}_0 \end{array} \right. \quad (5.2)$$

où les vecteurs  $\vec{y}'(t)$ ,  $\vec{f}(t, \vec{y}(t))$  et  $\vec{y}_0$ , tous de longueur  $m$ , sont donnés par

$$\vec{y}'(t) = \begin{pmatrix} y_1'(t) \\ y_2'(t) \\ y_3'(t) \\ \vdots \\ y_m'(t) \end{pmatrix} \quad \vec{f}(t, \vec{y}(t)) = \begin{pmatrix} f_1(t, y_1(t), y_2(t), \dots, y_m(t)) \\ f_2(t, y_1(t), y_2(t), \dots, y_m(t)) \\ f_3(t, y_1(t), y_2(t), \dots, y_m(t)) \\ \vdots \\ f_m(t, y_1(t), y_2(t), \dots, y_m(t)) \end{pmatrix} \quad \vec{y}_0 = \begin{pmatrix} y_{1,0} \\ y_{2,0} \\ y_{3,0} \\ \vdots \\ y_{m,0} \end{pmatrix}$$

### Remarque(s) :

- 1 On note  $y_i(t_n)$ , la valeur exacte de la  $i^{\text{e}}$  variable dépendante en  $t = t_n$  et  $y_{i,n}$ , son approximation numérique.
- 2 Les  $m$  équations du système (5.1) sont **couplées** en ce sens que l'équation différentielle régissant la variable dépendante  $y_i(t)$  peut dépendre de toutes les autres variables dépendantes.
- 3 Les  $m$  conditions initiales du système (5.1) assurent l'unicité de la solution.

## La méthode d'Euler explicite

La forme vectorielle (5.2) permet aux systèmes d'équations de généraliser toutes les méthodes numériques précédentes. En nous inspirant de la méthode d'Euler explicite, on part de  $(t_0, \vec{y}_0)$  pour obtenir le schéma vectoriel d'Euler explicite suivant

$$\begin{cases} \text{Partant de } (t_0, \vec{y}_0), \\ \vec{y}_{n+1} = \vec{y}_n + h\vec{f}(t_n, \vec{y}_n) \end{cases}$$

ou en développant avec  $\vec{y}_{n+1} = (y_{1,n+1}, y_{2,n+1}, \dots, y_{m,n+1})$ , on a

$$\begin{cases} \text{Partant de } (t_0, \vec{y}_0), \text{ on détermine } \vec{y}_{n+1} \text{ tel que} \\ y_{1,n+1} = y_{1,n} + hf_1(t_n, y_{1,n}, y_{2,n}, \dots, y_{m,n}) \\ y_{2,n+1} = y_{2,n} + hf_2(t_n, y_{1,n}, y_{2,n}, \dots, y_{m,n}) \\ \vdots \\ y_{m,n+1} = y_{m,n} + hf_m(t_n, y_{1,n}, y_{2,n}, \dots, y_{m,n}) \end{cases} \quad (5.3)$$

### Exemple 4

## La méthode de Runge-Kutta d'ordre 4

$$\left\{ \begin{array}{l}
 \text{Partant de } (t_0, \vec{y}_0), \\
 \text{On détermine } \vec{k}_1 = (k_{i,1})_{1 \leq i \leq m}, \\
 \quad k_{i,1} = hf_i(t_n, y_{1,n}, y_{2,n}, \dots, y_{m,n}) \\
 \text{On détermine } \vec{k}_2 = (k_{i,2})_{1 \leq i \leq m}, \\
 \quad k_{i,2} = hf_i\left(t_n + \frac{h}{2}, y_{1,n} + \frac{k_{1,1}}{2}, y_{2,n} + \frac{k_{2,1}}{2}, \dots, y_{m,n} + \frac{k_{m,1}}{2}\right) \\
 \text{On détermine } \vec{k}_3 = (k_{i,3})_{1 \leq i \leq m}, \\
 \quad k_{i,3} = hf_i\left(t_n + \frac{h}{2}, y_{1,n} + \frac{k_{1,2}}{2}, y_{2,n} + \frac{k_{2,2}}{2}, \dots, y_{m,n} + \frac{k_{m,2}}{2}\right) \\
 \text{On détermine } \vec{k}_4 = (k_{i,4})_{1 \leq i \leq m}, \\
 \quad k_{i,4} = hf_i(t_n + h, y_{1,n} + k_{1,3}, y_{2,n} + k_{2,3}, \dots, y_{m,n} + k_{m,3}) \\
 \text{Et finalement, on calcule} \\
 \quad y_{i,n+1} = y_{i,n} + \frac{1}{6} (k_{i,1} + 2k_{i,2} + 2k_{i,3} + k_{i,4})
 \end{array} \right.$$

Il faut calculer les  $m$  constantes  $k_{i,1}$  avant de passer au calcul des constantes  $k_{i,2}$  et ainsi de suite.



# Équations d'ordre supérieur

## Définition 6.1

La forme générale d'une équation différentielle d'ordre  $m$  avec conditions initiales est définie par le système suivant

$$\left\{ \begin{array}{l} y^{(m)}(t) = g(t, y(t), y^{(1)}(t), y^{(2)}(t), \dots, y^{(m-1)}(t)) \\ y(t_0) = c_1 \\ y^{(1)}(t_0) = c_2 \\ y^{(2)}(t_0) = c_3 \\ \vdots \\ y^{(m-2)}(t_0) = c_{m-1} \\ y^{(m-1)}(t_0) = c_m \end{array} \right. \quad (6.1)$$

**Note:** Ces conditions portent sur la fonction  $y(t)$  et ses  $(m - 1)$  premières dérivées à  $t = t_0$ .

## Exemple 5

- Exemple d'équation différentielle d'ordre 2

$$\begin{cases} y^{(2)}(t) = -y^{(1)}(t) + (y(t))^2 + t^2 - 5 \\ y(0) = 1 \\ y^{(1)}(0) = 2 \end{cases} \quad (6.2)$$

où  $g(t, y(t), y^{(1)}(t)) = -y^{(1)}(t) + (y(t))^2 + t^2 - 5$  et  $t_0 = 0$ .

- Exemple d'équation différentielle d'ordre 3

$$\begin{cases} y^{(3)}(t) = (y^{(2)}(t))^2 + 2y^{(1)}(t) + (y(t))^3 + t^4 + 1 \\ y(1) = 1 \\ y^{(1)}(1) = 0 \\ y^{(2)}(1) = 3 \end{cases} \quad (6.3)$$

où  $g(t, y(t), y^{(1)}(t), y^{(2)}(t)) = (y^{(2)}(t))^2 + 2y^{(1)}(t) + (y(t))^3 + t^4 + 1$  et  $t_0 = 1$ .

**Remarque(s) :**

- ❶ La première dérivée de la variable  $y(t)$  est notée  $y'(t)$  ou  $y^{(1)}(t)$  selon la situation.
- ❷ On se doit également de distinguer la dérivée seconde  $y^{(2)}(t)$  du carré de la fonction  $y(t)$ , qui est noté  $(y(t))^2$ .

**Théorème 6.1**

L'équation différentielle (6.1) d'ordre  $m$  avec les  $m$  conditions initiales est équivalente au système de  $m$  équations d'ordre 1 suivant

$$\left\{ \begin{array}{ll} y_1'(t) = y_2(t) & y_1(t_0) = c_1 \\ y_2'(t) = y_3(t) & y_2(t_0) = c_2 \\ \vdots & \\ y_{m-1}'(t) = y_m(t) & y_{m-1}(t_0) = c_{m-1} \\ y_m'(t) = g(t, y_1(t), y_2(t), \dots, y_m(t)) & y_m(t_0) = c_m \end{array} \right. \quad (6.4)$$

**Exemple 6**

## À Retenir !

Je dois pouvoir répondre aux questions <sup>a</sup> suivantes :

- ① Je comprends le concept de schéma en temps.
- ② Je comprends la construction d'un schéma basé sur le dev. de Taylor
- ③ Je sais distinguer entre les schémas de type Runge-Kutta et ceux basés sur Taylor.
- ④ Je comprends l'origine du concept de famille de schéma pour Runge-Kutta
- ⑤ Je sais appliquer les différents schémas pour une EDO d'ordre 1.
- ⑥ Je sais appliquer les schémas sur un système d'EDO d'ordre 1.
- ⑦ Je sais transformer une EDO d'ordre supérieur en système d'EDO.

---

a. André Fortin : *Analyse numérique pour ingénieurs*, Presses Internationales Polytechnique 2016.