

Résolution numérique des équations différentielles

Introduction

La résolution numérique des équations différentielles est le domaine de l'analyse numérique où les applications sont les plus nombreuses. Que ce soit en mécanique des fluides, en transfert de chaleur..., on aboutit souvent à la résolution d'équations différentielles, de systèmes d'équations différentielles ou plus généralement d'équations aux dérivées partielles.

Parmi les avantages des méthodes numériques, permettent d'étudier des problèmes complexes pour lesquels on ne connaît pas de solution analytique, mais qui sont d'un grand intérêt pratique.

Dans ce chapitre, **les diverses méthodes de résolution proposées sont d'autant plus précises qu'elles sont d'ordre élevé.** Nous commençons par des méthodes relativement simples ayant une interprétation géométrique. Elles nous conduiront progressivement à des méthodes plus complexes telles les méthodes de Runge-Kutta d'ordre 4, qui permettent d'obtenir des résultats d'une grande précision. Nous considérons principalement les équations différentielles avec conditions initiales. Nous prenons comme point de départ la formulation générale d'une équation différentielle d'ordre 1 avec condition initiale. Notre but consiste à déterminer une fonction $y(t)$ solution de :

$$\begin{cases} y'(t) &= f(t, y(t)) \\ y(t_0) &= y_0 \end{cases} \quad (1)$$

La variable indépendante est t , la variable dépendante est notée y qui dépend de t . La fonction f est une fonction de deux variables que nous supposons suffisamment différentiable. Il s'agit d'obtenir $y(t)$ pour $t \geq t_0$

Remarque 1 On note $y(t_i)$ la solution analytique de l'équation différentielle (1) en $t = t_i$. On note y_i la solution approximative en $t = t_i$ obtenue à l'aide d'une méthode numérique.

1.1 Méthodes numériques de résolution

Nous considérons la résolution numérique du problème de Cauchy (1) sur l'intervalle $[t_0, t_0 + T]$, on subdivise $[t_0, t_0 + T]$ en N sous-intervalles $[t_n, t_{n+1}]$, $n = 0, 1, \dots, N-1$, on note par $h = t_{n+1} - t_n$ le pas de discrétisation c-à-d $h = \frac{T}{N}$. L'idée des méthodes de discrétisation est de construire une suite de valeurs $(y)_{0 \leq n \leq N}$, approchant aux points $(t_i)_{i=0}^N$ la solution y du problème considéré, c'est-à-dire : $y_n \approx y(t_n), 0 \leq n \leq N$

On distingue deux classes de méthodes de discrétisation pour la résolution des équations différentielles

ordinaires. Celle des méthodes à un pas (où à pas séparés) qui sont caractérisées par le fait que, pour tout $n \geq 0$, la valeur approchée y_{n+1} de la solution au point t_{n+1} fournie par la méthode ne dépend que de la valeur y_n calculée à l'étape précédente. Au contraire, les méthodes à pas multiples (également dites à pas liés) font appel aux approximations de la solution en un certain nombre t_i antérieurs avec $0 \leq i \leq n$ pour déterminer la valeur approchée y_{i+1} .

1.1.1 La méthode d'Euler

La méthode d'Euler (the forward Euler method) est la plus ancienne et la plus simple des méthodes numériques d'approximation des équations différentielles ordinaires du premier ordre. Elle est définie par la relation de récurrence, ou schéma (scheme)

$$x_{n+1} = x_n + hf(t_n, x(t_n)). \quad n = 0, 1, \dots, N-1 \quad (2)$$

Interprétation géométrique

On a $y(t_0) = y_0$. Le but est d'obtenir une approximation de la solution en $t = t_1 = t_0 + h$. pour obtenir le point (t_1, y_1) , qui est une approximation du point $(t_1, y(t_1))$. Nous avons la pente $y'(t)$ en $t = t_0$.

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

L'équation de la droite passant par (t_0, y_0) et de pente $f(t_0, y_0)$ est donnée par :

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

En $t = t_1$, on a :

$$d_0(t_1) = y_0 + f(t_0, y_0)(t_1 - t_0) = y_0 + hf(t_0, y_0) = y_1$$

c.à.d, $d(t_1)$ est proche de la solution analytique $y(t_1)$, on a :

$$y(t_1) \simeq y_1 = d_0(t_1) = y_0 + hf(t_0, y_0)$$

Pour faire la deuxième itération et obtenir une approximation de $y(t_2)$, on refait l'analyse précédente à partir du point (t_1, y_1) . La pente de la solution analytique en $t = t_1$ est :

$$y'(t_1) = f(t_1, y(t_1)) \simeq f(t_1, y_1)$$

On construit la droite

$$d_1(t) = y_1 + f(t_1, y_1)(t - t_1)$$

En $t = t_2$, on a :

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

La même analyse se fait pour les points $t_3, t_4, \dots$ On abouti à l'algorithme suivant

Algorithme : Méthode d'Euler explicite

- 1 Etant donné le pas h , (t_0, y_0) et un nombre maximal N
- 2 Pour $0 \leq n \leq N$:
- 3 $y_{n+1} = y_n + hf(t_n, y_n)$
- 4 $t_{n+1} = t_n + h$
- 5 Ecrire t_{n+1} et y_{n+1}

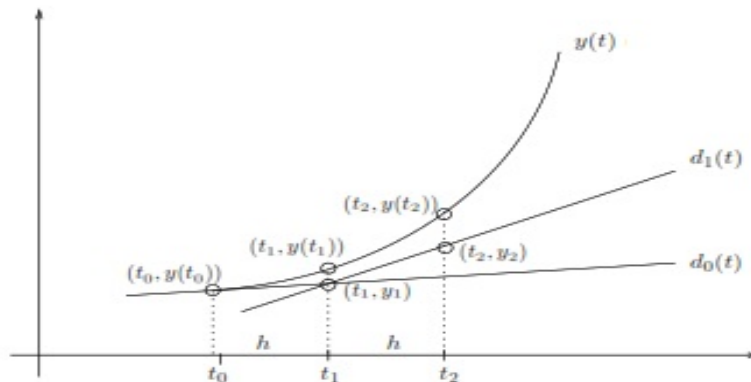


FIGURE 1.1 – Méthode d'Euler explicite

Remarque 2 La méthode d'Euler explicite découle également de la méthode d'intégration de Reimann à gauche en intégrant les deux membres de (1) entre t_n et t_{n+1} , il vient

$$y_{n+1} - y_n = \int_{t_n}^{t_{n+1}} f(s, y(s)) \, ds \simeq h f(t_n, y(t_n))$$

Où l'intégrale de Reimann à gauche est donnée par

$$\int_a^b f(x) \, dx \simeq (b - a) f(a)$$

Exemple 1 Résoudre de manière approchée à l'aide du schéma d'Euler explicite le problème suivant

$$\begin{cases} y'(t) = -y + t + 1 = f(t, y(t)) \\ y(0) = 1 \quad t \in [0, 1] \end{cases}$$

Le tableau suivant rassemble les résultats des dix premiers pas. La solution analytique de cette équation différentielle est $y(t) = e^{-t} + t$, ce qui permet de comparer les solutions numériques et analytiques.

t_i	$y(t_i)$	y_i	$ y(t_i) - y_i $
0, 0	1, 000000	1, 000000	0.000000e + 00
0, 1	1, 004837	1, 000000	4.837418e - 03
0, 2	1, 018731	1, 010000	8.730753e - 03
0, 3	1, 040818	1, 029000	1.181822e - 02
0, 4	1, 070302	1, 056100	1.422005e - 02
0, 5	1, 106531	1, 090490	1.604066e - 02
0, 6	1, 148812	1, 131441	1.737064e - 02
0, 7	1, 196585	1, 178297	1.828840e - 02
0, 8	1, 249329	1, 230467	1.886175e - 02
0, 9	1, 306570	1, 287420	1.914917e - 02
1, 0	1, 367879	1, 348678	1.920100e - 02

script matlab

```

1 clear
2 clc
3 h = 0.1;
4 n = (1/h) + 1;
5 t = 0 : h : 1
6 y(1) = 1;
7 f = @(t,y)t - y + 1;
8 yex = t + exp(-t)
9 for i = 2 : n
10 y(i) = y(i-1) + h * f(t(i-1), y(i-1));
11 end
12 fid=fopen('euler.txt','w')
13 fprintf(fid, '\n t(i)          exa          app          err');
14 for k = 1 : n
15 err(k) = abs(yex(k) - y(k));
16 fprintf(fid, '\n %f      %f      %f      %e \n', t(k), yex(k), y(k), err(k));
17 end

```

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

$$y_{n+1} = y_n + h\varphi(t_n, y_n) \quad (3)$$

où φ est une fonction quelconque. La méthode est à un pas si, pour obtenir la solution en $t = t_{n+1}$, on doit utiliser la solution numérique en t_n seulement. On désigne par méthodes à pas multiples, les méthodes qui exigent la solution numérique en $t_{n-1}, t_{n-2}, t_{n-3} \dots$.

Définition 2 On dit qu'un schéma à un pas converge à l'ordre p si :

$$\max_{1 \leq n \leq N} |y(t_n) - y_n| = O(h^p) \quad (4)$$

où N est le nombre total de pas.

on introduit l'erreur de troncature locale de la méthode au point $t = t_n$

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

$$\tau_{n+1}(h) = \frac{y(t_{n+1}) - y(t_n)}{h} - \varphi(t_n, y(t)) \quad (5)$$

Remarque 3 Dans le cas de la méthode d'Euler explicite ($\varphi(t, y) = f(t, y)$). En effectuant un développement de Taylor au voisinage du point $t = t_n$, on trouve :

$$\begin{aligned} y(t_{n+1}) &= y(t_n + h) = y(t_n) + hy'(t_n) + \frac{h^2}{2}y''(t_n) + O(h^3) \\ &= y(t_n) + hf(t_n, y(t_n)) + \frac{h^2}{2}y''(t_n) + O(h^3) \end{aligned}$$

puisque $y'(t_n) = f(t_n, y(t_n))$. L'erreur de troncature locale (5) devient donc :

$$\tau_{n+1}(h) = \frac{y(t_{n+1}) - y(t_n)}{h} - f(t_n, y(t_n)) = \frac{y''(t_n)h}{2} + O(h^2)$$

ou plus simplement $\tau_{n+1}(h) = O(h)$ et la méthode d'Euler explicite converge donc à l'ordre 1 ($p = 1$ dans la relation (4)).

1.1.2 Méthodes de Taylor

Le développement est une généralisation de la méthode d'Euler, qui permet d'obtenir des algorithmes dont l'erreur de troncature locale est d'ordre plus élevé. En effectuant le développement de Taylor au second ordre.

On cherche, en $t = t_n$, une approximation de la solution en $t = t_{n+1}$. On a

$$y(t_{n+1}) = y(t_n + h) = y(t_n) + hy'(t_n) + \frac{h^2}{2}y''(t_n) + O(h^3)$$

D'après l'équation (1), on trouve :

$$y(t_{n+1}) = y(t_n) + hf(t_n, y(t_n)) + \frac{h^2}{2}f'(t_n, y(t_n)) + O(h^3) \quad (6)$$

Alors que

$$f'(t, y(t)) = \frac{\partial f(t, y(t))}{\partial t} + \frac{\partial f(t, y(t))}{\partial y}y'(t)$$

c'est-à-dire :

$$f'(t, y(t)) = \frac{\partial f(t, y(t))}{\partial t} + \frac{\partial f(t, y(t))}{\partial y}f(t, y(t))$$

On obtient :

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

c-à-d

$$y(t_{n+1}) \simeq y(t_n) + hf(t_n, y(t_n)) + \frac{h^2}{2}\left(\frac{\partial f(t_n, y(t_n))}{\partial t} + \frac{\partial f(t_n, y(t_n))}{\partial y}f(t_n, y(t_n))\right) \quad (7)$$

Algorithme : Méthode de Taylor d'ordre 2

```

1 Etant donné le pas  $h$ ,  $(t_0, y_0)$  et un nombre maximal  $N$ 
2 Pour  $0 \leq n \leq N$  :
3    $y_{n+1} = y_n + hf(t_n, y_n) + \frac{h^2}{2}\left(\frac{\partial f(t_n, y_n)}{\partial t} + \frac{\partial f(t_n, y_n)}{\partial y}f(t_n, y_n)\right)$ 
4    $t_{n+1} = t_n + h$ 
5 écrire  $t_{n+1}$  et  $y_{n+1}$ 
```

Exemple 2 Résoudre de manière approchée à l'aide du schéma de Taylor le problème suivant :

$$\begin{cases} y'(t) = -y + t + 1 = f(t, y(t)) \\ y(0) = 1 \quad t \in [0, 1] \end{cases}$$

On a $\frac{\partial f(t, y(t))}{\partial t} = 1$, $\frac{\partial f(t, y(t))}{\partial y} = -1$, par suite on a

$$\begin{aligned} y_{n+1} &= y_n + hf(t_n, y_n) + \frac{h^2}{2}\left(\frac{\partial f(t_n, y_n)}{\partial t} + \frac{\partial f(t_n, y_n)}{\partial y}f(t_n, y_n)\right) \\ &= y_n + h(1 + t_n + y_n) + \frac{h^2}{2}(t_n - y_n) \end{aligned}$$

Les résultats sont données dans le tableau suivant

t_i	$y(t_i)$	y_i	$ y(t_i) - y_i $
0,0	1,000000	1,000000	0,000000
0,1	1,004837	1,005000	0,000163
0,2	1,018731	1,019025	0,000294
0,3	1,040818	1,041218	0,000400
0,4	1,070302	1,070802	0,000482
0,5	1,106531	1,107075	0,000544
0,6	1,148812	1,149404	0,000592
0,7	1,196585	1,197210	0,000625
0,8	1,249329	1,249975	0,000646
0,9	1,306570	1,307228	0,000658
1,0	1,367879	1,368541	0,000662

Remarque 4 1. On remarque que l'erreur dans la méthode Taylor 2 est plus petite que l'erreur dans la méthode explicite d'Eulerr.

2. En utilisant le développement de Taylor à l'ordre 3, 4, $\dots$, on obtient des méthodes encore plus précises, dans ce cas On doit évaluer les dérivées de la fonction $f(t, y(t))$ à l'ordre 3, 4, $\dots$ c-à-d faire le calcul de

$$\frac{\partial^2 f}{\partial t^2}, \frac{\partial^2 f}{\partial y^2}, \frac{\partial^2 f}{\partial t \partial y} \dots,$$

Exemple 3 Résoudre de manière approchée à l'aide du schéma de Taylor le problème suivant

$$\begin{cases} y'(t) = ty = f(t, y(t)) \\ y(1) = 2 \quad t \in [1, 3] \end{cases}$$

La solution exacte de ce problème est donnée par $2e^{(t^2-1)/2}$ et $\frac{\partial f(t, y(t))}{\partial t} = y, \frac{\partial f(t, y(t))}{\partial y} = t$.

Dans ce cas l'algorithme de Taylor devient

$$\begin{aligned} y_{n+1} &= y_n + hf(t_n, y_n) + \frac{h^2}{2} \left(\frac{\partial f(t_n, y_n)}{\partial t} + \frac{\partial f(t_n, y_n)}{\partial y} f(t_n, y_n) \right) \\ &= y_n + h(t_n y_n) + \frac{h^2}{2} (y_n + t_n(t_n y_n)) \end{aligned}$$

1.2 Méthodes de Runge-Kutta

Les méthodes de Taylor nécessitent le calcul des dérivées partielles de la fonction $f(t, y)$. Il serait avantageux de disposer de méthodes d'ordre plus élevé tout en gardant le même ordre des méthodes de Taylor ces méthodes sont appelées méthodes de Runge-Kutta.

1.2.1 Méthodes de Runge-Kutta d'ordre 2

Le développement de Taylor est donnée par :

$$y(t_{n+1}) = y(t_n) + hf(t_n, y(t_n)) + \frac{h^2}{2} \left(\frac{\partial f(t_n, y(t_n))}{\partial t} + \frac{\partial f(t_n, y(t_n))}{\partial y} f(t_n, y(t_n)) \right) + O(h^3) \quad (8)$$

Le but est de remplacer l'équation (8) par une expression équivalente possédant le même ordre de précision ($O(h^3)$). Soit :

$$y(t_{n+1}) = y(t_n) + a_1 hf(t_n, y(t_n)) + a_2 hf(t_n + a_3 h, y(t_n) + a_4 h) \quad (9)$$

où les paramètres a_1, a_2, a_3 et a_4 sont à déterminer.

Rappel 1

$$f(a+h, b+k) = f(a, b) + \frac{\partial f}{\partial x}(a, b)h + \frac{\partial f}{\partial y}(a, b)k + \frac{1}{2} \frac{\partial^2 f}{\partial x^2}(a, b)h^2 + \frac{1}{2} \frac{\partial^2 f}{\partial y^2}(a, b)k^2 + \frac{\partial^2 f}{\partial x \partial y}(a, b)hk$$

Le développement de Taylor de la $f(t_n + a_3h, y(t_n) + a_4h)$ autour du point $(t_n, y(t_n))$ est donnée par :

$$f(t_n + a_3h, y(t_n) + a_4h) = f(t_n, y(t_n)) + a_3h \frac{\partial f(t_n, y(t_n))}{\partial t} + a_4h \frac{\partial f(t_n, y(t_n))}{\partial y} + O(h^2)$$

En substituant cette expression dans (9) on trouve :

$$y(t_{n+1}) = y(t_n) + (a_1 + a_2)hf(t_n, y(t_n)) + a_2a_3h^2 \frac{\partial f(t_n, y(t_n))}{\partial t} + a_2a_4h^2 \frac{\partial f(t_n, y(t_n))}{\partial y} + O(h^3) \quad (10)$$

Pour déterminer les coefficients a_1, a_2, a_3 et a_4 , il suffit de comparer terme à terme (8) et (10) :

- coefficients respectifs $f(t_n, y(t)) : h = (a_1 + a_2)h$
- coefficients respectifs de $\frac{\partial f(t_n, y(t_n))}{\partial t} : \frac{h^2}{2} = a_2a_3h^2$
- coefficients respectifs de $\frac{\partial f(t_n, y(t_n))}{\partial y} : \frac{h^2}{2}f(t_n, y(t_n)) = a_2a_4h^2$

On obtient un système de 3 équations à 4 inconnues :

$$\begin{cases} a_1 + a_2 &= 1 \\ a_2a_3 &= \frac{1}{2} \\ a_2a_4 &= \frac{f(t_n, y(t_n))}{2} \end{cases} \quad (11)$$

Le système (11) est sous-déterminé, c-à-d il possède 4 inconnues et 3 équations donc n'a pas de solution unique. Cela offre plusieurs choix et variantes de la méthode de Runge-Kutta. Voici les choix les plus utilisés.

1.2.2 Méthode d'Euler modifiée**cas 1**

$$a_1 = a_2 = \frac{1}{2}, a_3 = 1 \quad \text{et} \quad a_4 = f(t_n, y(t))$$

En remplaçant ces valeurs des coefficients a_i dans l'équation 9, on obtient l'algorithme suivant.

Algorithme : d'Euler modifiée

```

1 Etant donné le pas  $h$ ,  $(t_0, y_0)$  et un nombre maximal  $N$ 
2 Pour  $0 \leq n \leq N$ 
3    $K = y_n + hf(t_n, y_n)$ 
4    $y_{n+1} = y_n + \frac{h}{2}(f(t_n, y_n) + f(t_n + h, K))$ 
5    $t_{n+1} = t_n + h$ 
6 écrire  $t_{n+1}$  et  $y_{n+1}$ 
```

script matlab

```

1 clear all
2 clc
3 h=0.1; n=(1/h)+1; t=0:h:1 y(1)=1;
4 f=@(t,y)t-y+1;
5 yex = t+exp(-t)
6 for i=2:n
7 K = y(i-1) +h*f (t(i-1),y(i-1));
8 y(i)=y(i-1)+h/2*(f(t(i-1),y(i-1))+f(t(i),K));
9 end
10 fid=fopen('EulerModifie.txt','w')
11 fprintf(fid,'\n t(i) \,\,\,\, exa \,\,\,\, app \,\,\,\, err');
12 for k=1:n
13 err(k) = abs(yex(k) - y(k));
14 fprintf(fid,'\n %f %f %f %e \n',t(k),yex(k),y(k),err(k));
15 end

```

Exemple 4 Résoudre de manière approchée à l'aide du schéma d'Euler modifiée le problème suivant

$$\begin{cases} y'(t) = -y + t + 1 = f(t, y(t)) \\ y(0) = 1 \quad t \in [0, 1] \end{cases}$$

Les résultats obtenus sont les mêmes que celle obtenus par la méthode de Taylor d'ordre 2.

1.2.3 Méthode du point milieu

cas 2

$$a_1 = 0, \quad a_2 = 1, \quad a_3 = \frac{1}{2} \quad \text{et} \quad a_4 = \frac{f(t_n, y(t_n))}{2}$$

En remplaçant les valeurs des coefficients a_i dans l'équation 9, on obtient l'algorithme suivant. **Algorithme** : du point milieu

```

1 Etant donné le pas h, (t0, y0) et un nombre maximal N
2 Pour 0 ≤ n ≤ N
3 k1 = hf(tn, yn)
4 yn+1 = yn + h(f(tn +  $\frac{h}{2}$ , yn +  $\frac{k_1}{2}$ ))
5 tn+1 = tn + h
6 écrire tn+1 et yn+1

```

Remarque 5

La méthode de point milieu découle également de la méthode d'intégration de Reimann au point milieu de l'intervalle t_n, t_{n+1}

$$y_{n+1} - y_n = \int_{t_n}^{t_{n+1}} f(s, y(s)) \, ds \simeq h f\left(t_n + \frac{h}{2}, y\left(t_n + \frac{h}{2}\right)\right)$$

ici, on connaît seulement la valeur de y_n , pour donner une approximation de la solution au point $t_n + \frac{h}{2}$ on utilise le schéma d'Euler explicite

$$y\left(t_n + \frac{h}{2}\right) \simeq y(t_n) + \frac{h}{2} f(t_n, y(t_n))$$

ce qui donne le schéma suivant

$$y_{n+1} = y_n + h\left(f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2} f(t_n, y_n)\right)\right)$$

A savoir l'intégrale de Reimann au point milieu est donnée par

$$\int_a^b f(x) \, dx \simeq (b-a) f\left(\frac{a+b}{2}\right)$$

1.3 Construction d'une méthode de Runge–Kutta explicite

En toute généralité, une méthode de Runge-Kutta à s étage pour la résolution du problème de Cauchy (1) est définie par

$$y_{n+1} = y_n + h \sum_{i=1}^s b_i k_i, \quad k_i = f\left(t_n + c_i h, y_n + h \sum_{j=1}^s a_{ij} k_j\right), \quad i = 1, \dots, s, \quad n = 0, \dots, N-1$$

Une telle méthode est caractérisée par la donnée des coefficients $\{a_{ij}\}_{1 \leq i, j \leq s}$, $\{b_i\}_{1 \leq i \leq s}$ et $\{c_i\}_{1 \leq i \leq s}$, que l'on représente dans le tableau de Butcher suivant :

(M_1)	c_1	a_{11}	a_{12}	$\cdots$	a_{1s}
(M_2)	c_2	a_{21}	a_{22}	$\cdots$	a_{2s}
$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$
(M_s)	c_s	a_{s1}	a_{s2}	$\cdots$	a_{ss}
(M)		b_1	b_2	$\cdots$	b_s

où les méthodes d'intégration approchées correspondent aux lignes que l'on peut encore écrire, en introduisant la matrice carrée $\mathbf{A}$ d'ordre s et les vecteurs $\mathbf{b}$ et $\mathbf{c}$ sous la forme

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}^T \end{array}$$

Remarque 6

- Si $a_{ij} = 0, 1 \leq i \leq j \leq s$, chacune des quantités k_i s'exprime uniquement en fonction de valeurs $k_j, 1 \leq j < i \leq s$, déjà connues et la méthode de Runge-Kutta est dite *explicite*. Sinon, la méthode est dite *implicite*

- Dans le cas où la méthode de RK est explicite, le tableau de Butcher se réduit à :

$$\begin{array}{c|cccc}
 (M_1) & c_1 & 0 & 0 & \cdots & 0 \\
 (M_2) & c_2 & a_{21} & 0 & \cdots & 0 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots \\
 (M_s) & c_s & a_{s1} & a_{s2} & \cdots & 0 \\
 \hline
 (M) & & b_1 & b_2 & \cdots & b_s
 \end{array}$$

- En particulier, on aura toujours

$$c_1 = 0, \quad c_i = \sum_{j=1}^{i-1} a_{ij}, \quad \sum_{j=1}^s b_j = 1, \quad k_1 = f(t_n, y_n) .$$

- Dans un tableau de Butcher, les composantes du vecteur $\mathbf{c}$ sont les incréments de x_n et les entrées de la matrice $\mathbf{A}$ sont les multiplicateurs des pentes approchées, qui, après multiplication par le pas h , deviennent les incréments de y_n . Les composantes du vecteur $\mathbf{b}$ sont les poids qui multiplient les k_j .

1.3.1 Méthodes de RungeKutta explicites d'ordre 1,2, 3 et 4

Les méthodes de Runge-Kutta sont à un pas mais elles sont à plusieurs étages.

Ordre 1 : (un seul étage) i.e $s = 1$, qui le seul choix possible

$$\begin{array}{c|c}
 0 & 0 \\
 \hline
 & 1
 \end{array}$$

i.e

$$y_{n+1} = y_n + hb_1k_1, \quad k_1 = f(t_n + c_1h, y_n + ha_{1,1}k_1) = f(t_n, y_n)$$

Ordre 2 : Les méthodes de Runge- Kutta explicites à 2 étages sont données par la formule (à gauche) et le tableau de Butcher (à droite) :

$$\left\{ \begin{array}{l} k_1 = f(t_n, y_n) \\ k_2 = f(t_n + c_2h, y_n + ha_{2,1}k_1) \\ y_{n+1} = y_n + h(b_1k_1 + b_2k_2) \end{array} \right. \quad (12)$$

$$\begin{array}{c|cc}
 & \mathbf{c} & \mathbf{A} \\
 \hline
 k_1 & 0 & 0 \\
 k_2 & c_2 & a_{2,1} \quad 0 \\
 y_{n+1} & \mathbf{b}^T & b_1 \quad b_2
 \end{array}$$

Première étape : Développement de $y(t)$ en t_n :

$$y(t_{n+1}) = y(t_n + h) = y(t_n) + hy'(t_n) + \frac{h^2}{2}y''(t_n) + O(h^3) \quad (13)$$

Si l'on dérive l'équation différentielle par rapport à t , en $t=t_n$

on obtient des expressions pour $y'(t)$ et $y''(t)$:

$$y'(t_n) = f(t_n, y(t_n)) ,$$

$$y''(t_n) = \frac{d}{dt}f(t, y(t))|_{t=x_n} = f'_t(t_n, y(t_n)) + f'_y(t_n, y(t_n))f(t_n, y(t_n)) ,$$

qu'on substitue dans (13) :

$$y(t_{n+1}) = y(t_n) + hf(t_n, y(t_n)) + \frac{h^2}{2} \overbrace{[f'_t(t_n, y(t_n)) + f'_y(t_n, y(t_n))f(t_n, y(t_n))]}^{y''(t_n)} + O(h^3) \quad (14)$$

Deuxième étape : Développement des termes de la formule de RK (12) en $(t_n, y(t_n))$ on a :

$$\begin{aligned} k_2 &= f(t_n + c_2h, y_n + ha_{2,1}k_1) \\ &= f(t_n, y(t_n)) + c_2hf'_t(t_n, y(t_n)) + ha_{2,1}k_1f'_y(t_n, y(t_n)) \\ &= f(t_n, y(t_n)) + c_2hf'_t(t_n, y(t_n)) + ha_{2,1}f(t_n, y(t_n))f'_y(t_n, y(t_n)) \end{aligned}$$

En substituant dans la formule de RK (12) on trouve :

$$\begin{aligned} y_{n+1} &= y_n + h(b_1k_1 + b_2k_2) \\ &= y_n + h(b_1 + b_2)f(t_n, y(t_n)) + b_2c_2h^2f'_t(t_n, y(t_n)) + h^2b_2a_{2,1}f(t_n, y(t_n))f'_y(t_n, y(t_n)) \end{aligned}$$

Troisième étape : Pour que cette expressions et l'expressions (14) soient égales à l'ordre h^3 , il faut que

$$\begin{cases} b_1 + b_2 &= 1 \\ b_2c_2 &= \frac{1}{2} \\ b_2a_{2,1} &= \frac{1}{2} \end{cases}$$

On a donc 3 équations à 4 inconnues, ce qui donne lieu à une famille de solutions à 1 paramètre.

Exemples de méthodes de Runge-Kutta du 2^{ième} ordre à 2 étages

La méthode d'Euler améliorée : On écrit la méthode d'Euler améliorée comme une méthode de RungeKutta du 2^{ième} ordre à 2 étages (gauche) et son tableau de Butcher (droite)

$$\begin{cases} k_1 &= f(t_n, y_n) \\ k_2 &= f(t_n + h, y_n + hk_1) \\ y_{n+1} &= y_n + \frac{h}{2}(k_1 + k_2) \end{cases} \quad \begin{array}{c|cc} & c & A \\ \hline k_1 & 0 & 0 \\ k_2 & 1 & 1 \quad 0 \\ \hline y_{n+1} & b^T & \frac{1}{2} \quad \frac{1}{2} \end{array}$$

La méthode du point-milieu : La méthode du point-milieu sous forme de Runge-Kutta s'écrit :

$$\begin{cases} k_1 &= f(t_n, y_n) \\ k_2 &= f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1) \\ y_{n+1} &= y_n + hk_2 \end{cases} \quad \begin{array}{c|cc} & c & A \\ \hline k_1 & 0 & 0 \\ k_2 & \frac{1}{2} & \frac{1}{2} \quad 0 \\ \hline y_{n+1} & b^T & 0 \quad 1 \end{array}$$

La méthode de Heun : La méthode de Heun sous forme de Runge-Kutta s'écrit :

$$\begin{cases} k_1 &= f(t_n, y_n) \\ k_2 &= f(t_n + \frac{2}{3}h, y_n + \frac{2h}{3}k_1) \\ y_{n+1} &= y_n + h(\frac{1}{4}k_1 + \frac{3}{4}k_2) \end{cases} \quad \begin{array}{c|cc} & c & A \\ \hline k_1 & 0 & 0 \\ k_2 & \frac{2}{3} & \frac{2}{3} \quad 0 \\ \hline y_{n+1} & b^T & \frac{1}{4} \quad \frac{3}{4} \end{array}$$

Ordre 3 : Les méthodes de Runge- Kutta explicites à 3 étages sont données par la formule (à gauche) et le tableau de Butcher (à droite) :

$$\begin{cases} k_1 = f(t_n, y_n) \\ k_2 = f(t_n + c_2 h, y_n + h a_{2,1} k_1) \\ k_3 = f(t_n + c_3 h, y_n + h(c_3 - a_{3,2})k_1 + h a_{3,2} k_2) \\ y_{n+1} = y_n + h(b_1 k_1 + b_2 k_2 + b_3 k_3) \end{cases} \quad (15)$$

	c	A		
k_1	0	0		
k_2	c_2	$a_{2,1}$	0	
k_3	c_3	$a_{3,1}$	$a_{3,2}$	0
y_{n+1}	b^T	b_1	b_2	b_3

En effectuant le calcul de la même manière que la section précédente i.e

Première étape : Développement de $y(t)$ en t_n :

$$y(t_{n+1}) = y(t_n + h) = y(t_n) + h y'(t_n) + \frac{h^2}{2} y''(t_n) + \frac{h^3}{6} y'''(t_n) + O(h^4) \quad (16)$$

Si l'on dérive l'équation différentielle par rapport à t , en $t=t_n$, on obtient des expressions pour $y'(t)$, $y''(t)$ et $y'''(t)$

$$\begin{cases} y'(t_n) = f(t_n, y(t_n)) \\ y''(t_n) = f'_t(t_n, y(t_n)) + f'_y(t_n, y(t_n)) f(t_n, y(t_n)) \\ y'''(t_n) = \left(\frac{\partial^2 f}{\partial t^2}(t_n, y(t_n)) + 2f(t_n, y(t_n)) \frac{\partial^2 f}{\partial t \partial y}(t_n, y(t_n)) + (f(t_n, y(t_n)))^2 \frac{\partial^2 f}{\partial y^2}(t_n, y(t_n)) \right. \\ \left. + \left(\frac{\partial f}{\partial t}(t_n, y(t_n)) + f(t_n, y(t_n)) \frac{\partial f}{\partial y}(t_n, y(t_n)) \right) \frac{\partial f}{\partial y}(t_n, y(t_n)) \right) + O(h^4). \end{cases}$$

En substituant les valeurs de $y'(t_n)$, $y''(t_n)$ et $y'''(t_n)$ dans (16) on trouve :

$$\begin{aligned} y_{n+1} = & y_n + h f(t_n, y(t_n)) + \frac{h^2}{2} \left(\frac{\partial f}{\partial t}(t_n, y(t_n)) + f(t_n, y(t_n)) \frac{\partial f}{\partial y}(t_n, y(t_n)) \right) \\ & + \frac{h^3}{6} \left(\frac{\partial^2 f}{\partial t^2}(t_n, y(t_n)) + 2f(t_n, y(t_n)) \frac{\partial^2 f}{\partial t \partial y}(t_n, y(t_n)) + (f(t_n, y(t_n)))^2 \frac{\partial^2 f}{\partial y^2}(t_n, y(t_n)) \right. \\ & \left. + \left(\frac{\partial f}{\partial t}(t_n, y(t_n)) + f(t_n, y(t_n)) \frac{\partial f}{\partial y}(t_n, y(t_n)) \right) \frac{\partial f}{\partial y}(t_n, y(t_n)) \right) + O(h^4) \end{aligned} \quad (17)$$

Deuxième étape : Développement des termes de la formule de RK dans le schéma (15) i.e k_2 et k_3 on trouve :

$$\begin{aligned} k_2 = & f(t_n, y(t_n)) + h c_2 \left(\frac{\partial f}{\partial t}(t_n, y(t_n)) + k_1 \frac{\partial f}{\partial y}(t_n, y(t_n)) \right) \\ & + \frac{h^2}{2} c_2^2 \left(\frac{\partial^2 f}{\partial t^2}(t_n, y(t_n)) + 2k_1 \frac{\partial^2 f}{\partial t \partial y}(t_n, y(t_n)) + k_1^2 \frac{\partial^2 f}{\partial y^2}(t_n, y(t_n)) \right) + O(h^3) \end{aligned} \quad (18)$$

et

$$\begin{aligned} k_3 = & f(t_n, y(t_n)) + h \left[c_3 \frac{\partial f}{\partial t}(t_n, y(t_n)) + ((c_3 - a_{3,2})k_1 + a_{3,2}k_2) \frac{\partial f}{\partial y}(t_n, y(t_n)) \right] \\ & + \frac{h^2}{2} \left(c_3^2 \frac{\partial^2 f}{\partial t^2}(t_n, y(t_n)) + 2c_3((c_3 - a_{3,2})k_1 + a_{3,2}k_2) \frac{\partial^2 f}{\partial t \partial y}(t_n, y(t_n)) \right. \\ & \left. + ((c_3 - a_{3,2})k_1 + a_{3,2}k_2)^2 \frac{\partial^2 f}{\partial y^2}(t_n, y(t_n)) \right) + O(h^3) \end{aligned} \quad (19)$$

En substituant (18) et (19) dans (15), on obtient :

$$\begin{aligned}
 y_{n+1} = & y_n + h(b_1 + b_2 + b_3)f(t_n, y(t_n)) + h^2(b_2c_2 + b_3c_3) \left[\frac{\partial f}{\partial t}(t_n, y(t_n)) + f(t_n, y(t_n)) \frac{\partial f}{\partial y}(t_n, y(t_n)) \right] \\
 & + \frac{h^3}{2}(b_2c_2^2 + b_3c_3^2) \left[\frac{\partial^2 f}{\partial t^2}(t_n, y(t_n)) + 2f(t_n, y(t_n)) \frac{\partial^2 f}{\partial t \partial y}(t_n, y(t_n)) + (f(t_n, y(t_n)))^2 \frac{\partial^2 f}{\partial y^2}(t_n, y(t_n)) \right] \\
 & + \frac{h^3}{2}2b_3c_2a_{32} \left[\frac{\partial f}{\partial t}(t_n, y(t_n)) + f(t_n, y(t_n)) \frac{\partial f}{\partial y}(t_n, y(t_n)) \right] \frac{\partial f}{\partial y}(t_n, y(t_n)) + O(h^4)
 \end{aligned} \tag{20}$$

Troisième étape : Il faut à présent faire coïncider les termes du développement (20) avec ceux de (17) on trouve :

$$\begin{cases} b_1 + b_2 + b_3 &= 1 \\ b_2c_2 + b_3c_3 &= \frac{1}{2} \\ b_2c_2^2 + b_3c_3^2 &= \frac{1}{3} \\ b_3c_2a_{32} &= \frac{1}{6} \end{cases}$$

On a donc 4 équations à 6 inconnues, ce qui donne lieu à une famille de solutions à 2 paramètre.

Exemples de méthodes de Runge-Kutta du 3^{ème} ordre à 3 étages

Parmi les solutions de ce système il ya deux méthodes qui sont connues sous le nom de :

1) Méthode de Heun : c'est une méthode d'ordre trois à trois niveaux, donnée par le système suivant, dont de tableau de Butcher associé à droite

$$\begin{cases} k_1 &= f(t_n, y_n) \\ k_2 &= f(t_n + \frac{h}{3}, y_n + \frac{h}{3}k_1) \\ k_3 &= f(t_n + \frac{2}{3}h, y_n + \frac{2}{3}hk_2) \\ y_{n+1} &= y_n + \frac{h}{4}(k_1 + 3k_3) \end{cases}$$

	c	A		
k_1	0	0	0	0
k_2	$\frac{1}{3}$	$\frac{1}{3}$	0	0
k_3	$\frac{2}{3}$	0	$\frac{2}{3}$	0
y_{n+1}	b^T	$\frac{1}{4}$	0	$\frac{3}{4}$

2) Méthode de Kutta : c'est une méthode d'ordre trois à trois niveaux, donnée par le système suivant, dont de tableau de Butcher associé à droite

$$\begin{cases} k_1 &= f(t_n, y_n) \\ k_2 &= f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1) \\ k_3 &= f(t_n + h, y_n - h(k_1 - 2k_2)) \\ y_{n+1} &= y_n + \frac{h}{6}(k_1 + 4k_2 + k_3) \end{cases}$$

	c	A		
k_1	0	0	0	0
k_2	$\frac{1}{2}$	$\frac{1}{2}$	0	0
k_3	1	-1	2	0
y_{n+1}	b^T	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

L'obtention de méthodes d'ordre supérieur est un travail difficile, le nombre de conditions augmentant avec l'ordre : il faut en satisfaire 8 pour arriver à l'ordre 4.

Exemples de méthodes de Runge-Kutta du 4^{ème} ordre à 4 étages

Une méthode de Runge-Kutta explicite d'ordre quatre, due à Kutta, très populaire est donnée par le système ci-dessous, dont de tableau de Butcher associé à droite

$$\left\{ \begin{array}{l} k_1 = f(t_n, y_n) \\ k_2 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1) \\ k_3 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2) \\ k_4 = f(t_n + h, y_n + hk_3) \\ y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{array} \right.$$

	c	A				
k_1	0	0	0	0	0	
k_2	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	
k_3	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	
k_4	1	0	0	1	0	
y_{n+1}	b^T	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$	

Une autre méthode d'ordre quatre, également découverte par Kutta, est donnée par le système ci-dessous, dont de tableau de Butcher associé à droite

$$\left\{ \begin{array}{l} k_1 = f(t_n, y_n) \\ k_2 = f(t_n + \frac{h}{3}, y_n + \frac{h}{3}k_1) \\ k_3 = f(t_n + \frac{2}{3}h, y_n - \frac{h}{3}k_2) + hk_3 \\ k_4 = f(t_n + h, y_n + h(k_1 + k_2 - k_3)) \\ y_{n+1} = y_n + \frac{h}{8}(k_1 + 3k_2 + 3k_3 + k_4) \end{array} \right.$$

	c	A				
k_1	0	0	0	0	0	
k_2	$\frac{1}{3}$	$\frac{1}{3}$	0	0	0	
k_3	$\frac{2}{3}$	0	$-\frac{1}{3}$	1	0	
k_4	1	1	1	-1	0	
y_{n+1}	b^T	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$	

Rappel 2

Notons par $F_1, F_2, F_3, \dots$ les dérivée successives de f alors :

- $F_1 = f_t + f f_y$, $F_2 = f_{tt} + 2f f_{ty} + f^2 f_{yy}$, $F_3 = f_{ttt} + 3f f_{tty} + 3f^2 f_{tyy} + f^3 f_{yyy}$
- En différenciant l'équation, (1) nous trouvons

$$\left\{ \begin{array}{l} y^{(2)} = f_t + f_y y' = f_t + f_y f = F_1 \\ y^{(3)} = f_{tt} + 2f f_{ty} + f^2 f_{yy} + f_y (f_x + f f_y) = F_2 + f_y F_1 \end{array} \right.$$

Exemple 5 Soit le problème de Cauchy :

$$\left\{ \begin{array}{l} y'(t) = y - \frac{2t}{y} = f(t, y(t)) \\ y(0) = 1 \quad t \in [0, 1] \end{array} \right.$$

En prenant un pas constant $h = 0.2$, calculer la solution au point $x = 1$ en utilisant :

1. La méthode d'Euler (ordre 1)
2. La méthode de Taylor (ordre 2)
3. La méthode de Runge-Kutta (ordre 2)
4. La méthode de Heun (ordre 3)
5. La méthode de Runge-Kutta (ordre 4)

Comparer les résultats numériques avec ceux de la solution analytique donnés par :

$$y = \sqrt{2t + 1}$$

script matlab

```
1 clear all
2 clc
```

```

3 h=0.2;
4 n=(1/h)+1;
5 t=0:h:1
6 ye(1)=1;yT(1)=1;yRK2(1)=1;yh(1)=1;yRK4(1)=1;
7 f=@(t,y)-((2*t)/y)+y
8 yex = sqrt(2*t+1)
9 %Euler (ordre 1)
10 for i=2:n
11 ye(i)=ye(i-1)+h*f(t(i-1),ye(i-1));
12 end
13 %Taylor (ordre 2)
14 for i=1:n-1
15 yT(i+1)=yT(i)+h*f(t(i),yT(i))
16 +(h^2/2)*((-2/yT(i))+(f(t(i),yT(i)))*(1+(2*t(i)/(yT(i))^2)))
17 end
18 %RK2 (Euler améliorée ordre 2)
19 for i=2:n
20 RK2= f(t(i-1),yRK2(i-1));
21 RK22= f(t(i-1)+h,yRK2(i-1)+h*RK2);
22 yRK2(i)=yRK2(i-1) +(h/2)*(RK2+RK22);
23 end
24 %Heun3 (ordre 3)
25 for i=2:n
26 H1= f(t(i-1),yh(i-1));
27 H2= f(t(i-1)+h/3,yh(i-1)+(h/3)*H1);
28 H3= f(t(i-1)+2*h/3,yh(i-1)+(2*h/3)*H1);
29 yh(i)=yh(i-1) +(h/4)*(H1+3*H2);
30 end
31 %RK4 (ordre 4)
32 for i=2:n
33 RK41= f(t(i-1),yRK4(i-1));
34 RK42= f(t(i-1)+h/2,yRK4(i-1)+h/2*RK41);
35 RK43= f(t(i-1)+h/2,yRK4(i-1)+h/2*RK42);
36 RK44= f(t(i-1)+h,yRK4(i-1)+h*RK43);
37 yRK4(i)=yRK4(i-1) +(h/6)*(RK41+2*RK42+2*RK43+RK44);
38 end
39 fid=fopen('Resultat_exemple.txt','w')
40 fprintf(fid,'\n t(i)      Euler      Taylor      RK2      Heun      ...
      RK4      exacte\n');
41 for k=1:n
42 fprintf(fid,'\n %3.2f      %6.6f      %6.6f      %6.6f      %6.6f      %8.6f      %8.6f ...
      \n',t(k),ye(k),yT(k),yRK2(k),yh(k),yRK4(k),yex(k));
43 end
44 plot(t, ye, 'g--*', t, yT, 'k--', t, yRK2, 'r--', t, yh, 'm--', t, yRK4, 'b--', t, yex, 'y--')
45 grid
46 legend('Euler', 'Taylor', 'RK2', 'Heun', 'RK4', 'Exacte', 'Location', 'northwest')

```

Les résultats sont données dans le tableau suivant

t_i	Euler	Taylor	RK2	Heun	RK4	Exacte
0.0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
0.2	1.200000	1.180000	1.186667	1.191250	1.183229	1.183216
0.4	1.373333	1.335957	1.348312	1.357177	1.341667	1.341641
0.6	1.531495	1.474795	1.493704	1.507411	1.483281	1.483240
0.8	1.681085	1.600414	1.627861	1.647572	1.612514	1.612452
1.0	1.826948	1.715073	1.754205	1.781691	1.732142	1.732051