In [1]:
```
from IPython.display import display, Latex
from IPython.core.display import HTML
css_file = './custom.css'
HTML(open(css_file, "r").read())
```

Out[1]:

In [2]:
```
from math import *
```

M62_CM8 : Schémas à un pas de type Runge-Kutta

Table of Contents

Schémas de Runge-Kutta

Les schémas de Runge-Kutta sont des méthodes à un pas qui approchent l'intégrale $\int_{t_n}^{t_{n+1}} \varphi(t, y(t))\mathrm{d}t$ par une formule de quadrature en des points donnés toujours à l'intérieur de l'intervalle $[t_n; t_{n+1}]$:

$$t_{n,i} \overset{\text{déf}}{=} t_n + c_i h \qquad \text{avec } c_i \in [0; 1]$$

$$\int_{t_n}^{t_{n+1}} \varphi(t, y(t))\mathrm{d}t \approx h \sum_{i=1}^{s} b_i \varphi(t_{n,i}, y(t_{n,i})).$$

Le problème est que, si $t_{n,i}$ n'est pas un point de la discrétisation, l'on ne connait pas $y(t_{n,i})$. L'évaluation des $\varphi(t_{n,i}, y(t_{n,i}))$ en les points intérieurs et alors approchée par une autre formule de quadrature:

$$\varphi(t_{n,i}, y(t_{n,i})) \approx \varphi\left(t_{n,i}, y_n + h \sum_{j=1}^{s} a_{ij} \varphi(t_{n,j}, y(t_{n,j}))\right).$$

Cela donne

$$y_{n+1} \approx y_n + h \sum_{i=1}^{s} \left[b_i \varphi\left(t_{n,i}, y_n + h \sum_{j=1}^{s} a_{ij} \varphi(t_{n,j}, y_{n,j})\right)\right]$$

Une méthode de Runge-Kutta à $s \geq 1$ étages s'écrit:

$$\begin{cases} u_0 = y(t_0) = y_0, \\ u_{n+1} = u_n + h \sum_{i=1}^{s} b_i K_i & n = 0, 1, \dots N-1 \\ K_i = \varphi \left(t_n + h c_i, u_n + h \sum_{j=1}^{s} a_{ij} K_j \right) & i = 1, \dots s. \end{cases}$$

Matrice de Butcher

Les coefficients sont généralement organisés en deux vecteurs $\mathbf{b} = (b_1, \dots, b_s)^T$, $\mathbf{c} = (c_1, \dots, c_s)^T$ et une matrice $\mathbb{A} = (a_{ij})_{1 \leq i,j \leq s}$. Le tableau

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

est appelée *matrice de Butcher* de la méthode Runge-Kutta considérée.

- Si $a_{ij} = 0$ pour $j \geq i$ (i.e. la matrice $\mathbb{A}$ est triangulaire inférieure stricte) alors la méthode est *explicite* car chaque K_i peut être explicitement calculé en fonction des $i-1$ coefficients $K_1, \dots, K_{i-1}$ déjà connus;
- dans les autres cas la méthode est *implicite* et il faut résoudre un système non linéaire de dimension s pour calculer les K_i. L'augmentation des calculs pour les schémas implicites rend leur utilisation coûteuse; un compromis acceptable est le suivant:
  - si $a_{ij} = 0$ pour $j > i$ (i.e. la matrice $\mathbb{A}$ est triangulaire inférieure) alors la méthode est *semi-implicite*, i.e. chaque K_i est solution de l'équation non linéaire

    $$\boxed{K_i} = \varphi \left(t_n + c_i h, u_n + h a_{ii} \boxed{K_i} + h \sum_{j=1}^{i-1} a_{ij} K_j \right)$$

    Un schéma semi-implicite implique donc la résolution de s équations non linéaires indépendantes.
    Remarque: la méthode est implicite non pas parce que u_{n+1} dépend de lui même, mais parce qu'au moins un K_i dépend de lui même.

Ordre de convergence

Théorème [Ordre 1]
La méthode de Runge-Kutta est consistante (i.e. d'ordre 1) ssi

- $\sum_{j=1}^{s} b_i = 1$,
- $c_i = \sum_{j=1}^{s} a_{ij}$.

Remarque En particulier, pour une méthode *explicite* on aura $c_1 = a_{1,j} = 0$ ainsi $K_1 = \varphi(t_n, u_n)$ et $c_2 = a_{21}$ ainsi $K_2 = \varphi(t_n + c_2 h, u_n + h c_2 K_1)$.

Théorème [Ordre 2, 3 et 4]
La méthode de Runge-Kutta est

- d'ordre ≥ 2 si, de plus, $\sum_j b_j c_j = \frac{1}{2}$
- d'ordre ≥ 3 si, de plus, $\sum_j b_j c_j^2 = \frac{1}{3}$ et $\sum_{i,j} b_i a_{ij} c_j = \frac{1}{6}$
- d'ordre ≥ 4 si, de plus, $\sum_j b_j c_j^3 = \frac{1}{4}$, $\sum_{i,j} b_i c_i a_{ij} c_j = \frac{1}{8}$, $\sum_{i,j} b_i a_{ij} c_j^2 = \frac{1}{12}$ et $\sum_{i,j,k} b_i a_{ij} a_{jk} c_k = \frac{1}{24}$.

Théorème
L'ordre d'une méthode RK **explicite** à s étapes ne peut pas être plus grand que s.
De plus, il n'existe pas de méthode à s étapes d'ordre s si $s \geq 5$.
L'ordre d'une méthode RK **implicite** à s étapes ne peut pas être plus grand que $2s$.

Remarque (le cas des systèmes) Une méthode de Runge-Kutta peut être aisément étendue aux systèmes d'équations différentielles ordinaires. Néanmoins, l'ordre d'une méthode RK dans le cas scalaire ne coïncide pas nécessairement avec celui de la méthode analogue dans le cas vectoriel. En particulier, pour $p \geq 4$, une méthode d'ordre p dans le cas du système autonome $\mathbf{y}'(t) = \boldsymbol{\varphi}(t, \mathbf{y})$, avec $\boldsymbol{\varphi}: \mathbb{R}^m \to \mathbb{R}^n$ est encore d'ordre p quand on l'applique à l'équation scalaire autonome $\mathbf{y}'(t) = \boldsymbol{\varphi}(\mathbf{y})$, mais la réciproque n'est pas vraie.

A-stabilité

Rappelons la définition de schéma A-stable:

Soit $\beta > 0$ un nombre réel positif et considérons le problème de Cauchy

$$\begin{cases} y'(t) = -\beta y(t), \quad \text{pour } t > 0, \\ y(0) = y_0 \end{cases}$$

où $y_0 \neq 0$ est une valeur donnée. Sa solution est $y(t) = y_0 e^{-\beta t}$ donc

$$\lim_{t \to +\infty} y(t) = 0.$$

Soit $h > 0$ un pas de temps donné, $t_n = nh$ pour $n \in \mathbb{N}$ et notons $u_n \approx y(t_n)$ une approximation de la solution y au temps t_n. Si, sous d'éventuelles conditions sur h, on a

$$\lim_{n \to +\infty} u_n = 0,$$

alors on dit que **le schéma est A-stable.**

Une méthode de Runge-Kutta à $s \geq 1$ étages pour $y'(t) = -\beta y(t)$ s'écrit:

$$\begin{cases} u_0 = y_0, \\ u_{n+1} = u_n + h \displaystyle\sum_{i=1}^{s} b_i K_i & n = 0, 1, \dots N-1 \\ \\ K_i = -\beta \left(u_n + h \displaystyle\sum_{j=1}^{s} a_{ij} K_j \right) & i = 1, \dots s. \end{cases}$$

On peut monter que, contrairement aux méthodes multi-pas, la taille des régions de stabilité absolue des méthodes RK augmente quand l'ordre augmente.

Exercice : schémas RK à un étage

Étudier les méthodes RK avec $s = 1$:

1. écrire le générique schéma implicite RK à un étage
2. étudier son ordre de convergence
3. écrire le générique schéma explicite RK à un étage
4. étudier son ordre de convergence
5. étudier la A-stabilité

Correction : écriture d'un schéma RK à un étage quelconque

Ces méthodes ont matrice de Butcher

$$\begin{array}{c|c} c_1 & a_{11} \\ \hline & b_1 \end{array} \text{ avec } \begin{cases} c_1 = a_{11} \\ b_1 = 1 \end{cases} \quad \Longrightarrow \quad \begin{array}{c|c} c_1 & c_1 \\ \hline & 1 \end{array} \quad \Longrightarrow \quad \begin{cases} u_0 = y_0 \\ K_1 = \varphi\left(t_n + hc_1, u_n + hc_1 K_1\right) \\ u_{n+1} = u_n + hK_1 \qquad\qquad n = 0, 1, \dots N-1 \end{cases}$$

Par exemple, si on choisi $c_1 = 1$ on trouve le schéma implicite

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array} \quad \Longrightarrow \begin{cases} u_0 = y_0 \\ K_1 = \varphi\left(t_{n+1}, u_n + hK_1\right) \\ u_{n+1} = u_n + hK_1 \qquad\qquad n = 0, 1, \dots N-1 \end{cases}$$

Notons que ce schéma n'est rien d'autre que le schéma d'Euler implicite car $u_{n+1} = u_n + hK_1$, donc si on remplace le $u_n + hK_1$ dans la définition de K_1 par u_{n+1} on obtient

$$K_1 = \varphi\left(t_{n+1}, u_n + hK_1\right) = \varphi\left(t_{n+1}, u_{n+1}\right) \quad \Longrightarrow \quad u_{n+1} = u_n + hK_1 = u_n + h\varphi\left(t_{n+1}, u_{n+1}\right)$$

Correction : étude de l'ordre d'un schéma RK à un étage quelconque

Si on cherche une méthode d'ordre 2 alors il faut que $1 \times c_1 = \frac{1}{2}$: il existe une seule méthode d'ordre 2, elle est implicite et sa matrice de Butcher est

$$\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array} \quad \Longrightarrow \quad \begin{cases} u_0 = y_0 \\ K_1 = \varphi\left(t_n + \frac{h}{2}, u_n + \frac{h}{2} K_1\right) \\ u_{n+1} = u_n + h K_1 \qquad\qquad n = 0, 1, \dots N-1 \end{cases}$$

Notons que $u_{n+1} = u_n + hK_1$, donc si on remplace le $u_n + \frac{h}{2} K_1 = \frac{u_n + (u_n + hK_1)}{2}$ dans la définition de K_1 par u_{n+1} on obtient

$$K_1 = \varphi\left(t_n + \frac{h}{2}, u_n + \frac{h}{2} K_1\right) = \varphi\left(t_n + \frac{h}{2}, \frac{u_n + u_{n+1}}{2}\right) \quad \Longrightarrow \quad u_{n+1} = u_n + hK_1 = u_n + h\varphi\left(t_n + \frac{h}{2}, \frac{u_n + u_{n+1}}{2}\right)$$

Correction : écriture d'un schéma RK à un étage explicite

Si le schéma est explicite alors

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array} \quad \Longrightarrow \quad \begin{cases} u_0 = y_0 \\ K_1 = \varphi\left(t_n, u_n\right) \\ u_{n+1} = u_n + h K_1 \quad n = 0, 1, \dots N-1 \end{cases}$$

Il existe un seul schéma explicite RK à un étage, il s'agit du schéma d'Euler explicite.

Correction : étude de l'ordre d'un schéma RK à un étage explicite

Si le schéma est explicite alors il est d'ordre 1.

Correction : étude de la A-stabilité

Une méthode de Runge-Kutta à 1 étage pour $y'(t) = -\beta y(t)$ s'écrit:

$$\begin{cases} u_0 = y_0, \\ u_{n+1} = u_n + hK_1 & n = 0, 1, \dots N-1 \\ K_1 = -\beta \left(u_n + hc_1 K_1 \right). \end{cases}$$

On trouve donc $(1 + \beta h c_1) K_1 = -\beta u_n$ ainsi

$$\begin{cases} u_0 = y_0 \\ u_{n+1} = \left(1 - \frac{(\beta h)}{1 + c_1 (\beta h)} \right) u_n \quad n = 0, 1, \dots N-1 \end{cases}$$

Par induction on obtient

$$u_n = \left(1 - \frac{\beta h}{1 + c_1 \beta h} \right)^n u_0.$$

Par conséquent, $\lim\limits_{n \to +\infty} u_n = 0$ si et seulement si

$$\left| 1 - \frac{\beta h}{1 + c_1 \beta h} \right| < 1.$$

Notons x le produit $\beta h > 0$ (donc $x > 0$) et q la fonction $q(x) = 1 - \frac{x}{1 + c_1 x}$ (avec $c_1 \in [0; 1]$).

$$|q(x)| < 1 \quad \iff \quad -1 < 1 - \frac{x}{1 + c_1 x} < 1 \quad \iff \quad \begin{cases} \frac{x}{1 + c_1 x} < 2 \\ \frac{x}{1 + c_1 x} > 0 \end{cases} \overset{\substack{x>0 \\ c_1 \geq 0}}{\iff} \quad \frac{x}{1 + c_1 x} < 2 \quad \overset{c_1 x \geq 0}{\iff} \quad x < 2(1 + c_1 x) \quad \iff$$

$$(1 - 2c_1) x < 2 \quad \iff \quad \begin{cases} x < \frac{2}{1 - 2c_1} & \text{si } 1 - 2c_1 > 0 \\ \forall x & \text{sinon.} \end{cases}$$

Conclusion:

- si $c_1 \geq \frac{1}{2}$ le schéma est inconditionnellement A-stable,
- si $c_1 < \frac{1}{2}$ le schéma est A-stable ssi $\beta h < \frac{2}{1 - 2c_1}$.

Pour nos exemples, on obtient:

- si $c_1 = 1$ (Euler Implicite) alors le schéma est inconditionnellement A-stable et d'ordre 1,
- si $c_1 = \frac{1}{2}$ alors le schéma est inconditionnellement A-stable et d'ordre 2,
- si $c_1 = 0$ (Euler Explicite) alors le schéma est A-stable ssi $h < \frac{2}{\beta}$ et il est d'ordre 1.

Exercice : schémas RK à deux étages

Étudier les méthodes RK avec $s=2$:

1. écrire le générique schéma implicite RK à deux étages
2. écrire le générique schéma semi-implicite RK à deux étages
3. écrire le générique schéma explicite RK à deux étages
4. étudier l'ordre de convergences des schémas explicites
5. étudier la A-stabilité des schémas explicites

Correction : écriture d'un schéma RK à deux étages quelconque

Les méthodes avec $s=2$ ont matrice de Butcher

$$\begin{array}{c|cc} c_1 & a_{11} & a_{12} \\ c_2 & a_{21} & a_{22} \\ \hline & b_1 & b_2 \end{array} \text{ avec } \begin{cases} c_1 = a_{11} + a_{12} \\ c_2 = a_{21} + a_{22} \\ b_1 + b_2 = 1 \end{cases} \implies \begin{array}{c|cc} c_1 & a_{11} & c_1 - a_{11} \\ c_2 & a_{21} & c_2 - a_{21} \\ \hline & 1-b_2 & b_2 \end{array}$$

$$\implies \begin{cases} u_0 = y_0 \\ K_1 = \varphi\left(t_n + hc_1, u_n + h(a_{11}K_1 + (c_1 - a_{11})K_2)\right) \\ K_2 = \varphi\left(t_n + hc_2, u_n + h(a_{21}K_1 + (c_2 - a_{21})K_2)\right) \\ u_{n+1} = u_n + h\left((1-b_2)K_1 + b_2K_2\right) \end{cases} \qquad n = 0, 1, \dots N-1$$

Voici un exemple, appelée *méthode de Gauss*:

$$\begin{array}{c|cc} \frac{1}{2} - \gamma & \frac{1}{4} & \frac{1}{4} - \gamma \\ \frac{1}{2} + \gamma & \frac{1}{4} + \gamma & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \qquad \text{avec } \gamma = \frac{\sqrt{3}}{6}$$

Cette méthode est d'ordre 4 et on peut monter que c'est la seule méthode d'ordre 4 à deux étages.

In [3]:
```
from sympy import *
gamma=sqrt(3)/6
c=[1/2-gamma,1/2+gamma]
b=[1/2,1/2]
A=[[1/4,1/4-gamma],[1/4+gamma,1/4]]
s=len(c)
print("Consistance")
print(sum(b)==1)
for i in range(s):
    print(sum(A[i]).simplify()==c[i])
print("\nOrdre 2")
print(sum([b[i]*c[i] for i in range(s)]).simplify()==1/2)
print("\nOrdre 3")
print(sum([b[i]*c[i]**2 for i in range(s)]).simplify()==1/3)
print(sum([b[i]*A[i][j]*c[j] for i in range(s) for j in range(s)]).simplify()==1/6)
print("\nOrdre 4")
print(sum([b[i]*c[i]**3 for i in range(s)]).simplify()==1/4)
print(sum([b[i]*c[i]*A[i][j]*c[j] for i in range(s) for j in range(s)]).simplify()==1/8)
print(sum([b[i]*A[i][j]*c[j]**2 for i in range(s) for j in range(s)]).simplify()==1/12)
print(sum([b[i]*A[i][j]*A[j][k]*c[k] for i in range(s) for j in range(s) for k in range(s)]).simplify()==1/24)
```

```
Consistance
True
True
True

Ordre 2
True

Ordre 3
True
True

Ordre 4
True
True
True
True
```

Correction : écriture d'un schéma RK à deux étages semi-implicite

Les méthodes semi-implicites vérifient, de plus, $c_1 = a_{11}$ ainsi la matrice de Butcher est

$$\begin{array}{c|cc} c_1 & c_1 & 0 \\ c_2 & a_{21} & c_2 - a_{21} \\ \hline & 1-b_2 & b_2 \end{array} \quad \Longrightarrow \quad \begin{cases} u_0 = y_0 \\ K_1 = \varphi\left(t_n + hc_1, u_n + hc_1 K_1\right) \\ K_2 = \varphi\left(t_n + hc_2, u_n + h(a_{21}K_1 + (c_2 - a_{21})K_2\right)) \\ u_{n+1} = u_n + h\left((1-b_2)K_1 + b_2 K_2\right) \end{cases} \qquad n = 0, 1, \dots N-1$$

Par exemple, si on choisi $c_1 = 0$, $c_1 = 1$ et $a_{21} = a_{22} = b_1 = b_2 = \frac{1}{2}$on trouve le schéma semi-implicite

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \quad \Longrightarrow \quad \begin{cases} u_0 = y_0 \\ K_1 = \varphi\left(t_n, u_n\right) \\ K_2 = \varphi\left(t_{n+1}, u_n + \frac{h}{2}(K_1 + K_2)\right) \\ u_{n+1} = u_n + \frac{h}{2}(K_1 + K_2) \end{cases} \qquad n = 0, 1, \dots N-1$$

Notons que ce schéma n'est rien d'autre que le schéma de Crank-Nicolson car $u_{n+1} = u_n + \frac{h}{2}(K_1 + K_2)$, donc si on remplace le $u_n + \frac{h}{2}(K_1 + K_2)$ dans la définition de K_2 par u_{n+1} on obtient

$$K_2 = \varphi\left(t_{n+1}, u_n + \frac{h}{2}(K_1 + K_2)\right) = \varphi\left(t_{n+1}, u_{n+1}\right)$$

ainsi

$$u_{n+1} = u_n + \frac{h}{2}(K_1 + K_2) = u_n + \frac{h}{2}(\varphi\left(t_n, u_n\right) + \varphi\left(t_{n+1}, u_{n+1}\right))$$

Correction : écriture d'un schéma RK à deux étages explicite

Les méthodes avec $s = 2$ explicites ont matrice de Butcher

$$\begin{array}{c|cc} 0 & 0 & 0 \\ c_2 & c_2 & 0 \\ \hline & 1-b_2 & b_2 \end{array} \quad \Longrightarrow \quad \begin{cases} u_0 = y_0 \\ K_1 = \varphi\left(t_n, u_n\right) \\ K_2 = \varphi\left(t_n + hc_2, u_n + hc_2 K_1\right) \\ u_{n+1} = u_n + h\left((1-b_2)K_1 + b_2 K_2\right) \end{cases} \quad n = 0, 1, \dots N-1$$

Correction : étude de l'ordre d'un schéma RK à deux étages explicite

- Pour avoir l'ordre 2 il faut que $b_2 c_2 = \frac{1}{2}$.
- Pour avoir l'ordre 3 il faudrait que $b_2 c_2^2 = \frac{1}{3}$ et $b_2 a_{21} c_1 = \frac{1}{6}$ ce qui est impossible car $c_1 = 0$.

Conclusion: un schéma RK à deux étages explicite d'ordre 2 s'écrit

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \frac{1}{2\alpha} & \frac{1}{2\alpha} & 0 \\ \hline & 1-\alpha & \alpha \end{array} \quad \Longrightarrow \quad \begin{cases} u_0 = y_0 \\ K_1 = \varphi\left(t_n, u_n\right) \\ K_2 = \varphi\left(t_n + \frac{h}{2}\alpha, u_n + \frac{h}{2}\alpha K_1\right) \\ u_{n+1} = u_n + h\left((1-\alpha)K_1 + \alpha K_2\right) \quad n = 0, 1, \dots N-1 \end{cases}$$

Voici deux cas particuliers:

- $\alpha = \frac{1}{2}$: schéma de Heun

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \quad \Longrightarrow \quad \begin{cases} u_0 = y_0 \\ K_1 = \varphi(t_n, u_n) \\ K_2 = \varphi(t_{n+1}, u_n + hK_1) \\ u_{n+1} = u_n + h\left(\frac{1}{2}K_1 + \frac{1}{2}K_2\right) \quad n = 0, 1, \dots N-1 \end{cases}$$

- $\alpha = 1$: schéma d'Euler Modifié $$ \begin{array}{c|cc}

```
0 & 0             & 0\\
```

\frac{1}{2} & \frac{1}{2} & 0\ \hline

```
  & 0             & 1
```

\end{array} \quad \implies

$$\begin{cases} u_0 = y_0 \\ K_1 = \varphi(t_n, u_n) \\ K_2 = \varphi(t_n + \frac{1}{2}h, u_n + \frac{1}{2}hK_1) \\ u_{n+1} = u_n + hK_2 \qquad\qquad n = 0, 1, \dots N-1 \end{cases}$$

$$

Correction : étude de la A-stabilité des schémas explicites

Le schéma donné appliqué à cette équation devient

$$\begin{cases} u_0 = y_0 \\ u_{n+1} = \left(1 - (\beta h) + c_2 b_2 (\beta h)^2\right) u_n \quad n = 0, 1, \dots N-1 \end{cases}$$

Par induction on obtient

$$u_n = \left(1 - (\beta h) + c_2 b_2 (\beta h)^2\right)^n u_0.$$

Par conséquent, $\lim_{n \to +\infty} u_n = 0$ si et seulement si

$$\left|1 - (\beta h) + c_2 b_2 (\beta h)^2\right| < 1.$$

Notons x le produit $\beta h > 0$ (donc $x > 0$) et q le polynôme $q(x) = 1 - x + c_2 b_2 x^2$.

- Si $c_2 b_2 = 0$, le schéma se réduit au schéma d'Euler explicite.
- Si $c_2 b_2 > 0$, c'est une parabole convexe et le sommet est situé en $\left(\frac{1}{2c_2b_2} > 0, 1 - \frac{1}{4c_2b_2}\right)$.

  $$|q(x)| < 1 \quad \iff \quad x < \frac{1}{c_2 b_2}$$

  En particulier, lorsque le schéma est d'ordre 2 on a $b_2 c_2 = \frac{1}{2}$ et le schéma est A-stable ssi $h < \frac{2}{\beta}$.
- Si $c_2 b_2 < 0$, c'est une parabole concave et le sommet est situé en $\left(\frac{1}{2c_2b_2} < 0, 1 - \frac{1}{4c_2b_2}\right)$.

  $$|q(x)| < 1 \quad \iff \quad x < \frac{1 - \sqrt{1 - 8 b_2 c_2}}{2 b_2 c_2}$$

Exemples à 3 étages explicites

Un schéma RK à 3 étages explicite a pour matrice de Butcher

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ c_2 & a_{21} & 0 & 0 \\ c_3 & a_{31} & a_{32} & 0 \\ \hline & b_1 & b_2 & b_3 \end{array} \quad \text{avec} \begin{cases} c_2 = a_{21} \\ c_3 = a_{31} + a_{32} \\ b_1 + b_2 + b_3 = 1 \end{cases} \quad \Longrightarrow \begin{cases} u_0 = y_0 \\ K_1 = \varphi\left(t_n, u_n\right) \\ K_2 = \varphi\left(t_n + hc_2, u_n + hc_2 K_1\right) \\ K_3 = \varphi\left(t_n + hc_3, u_n + h(a_{31} K_1 + a_{32} K_2)\right) \\ u_{n+1} = u_n + h\left(b_1 K_1 + b_2 K_2 + b_3 K_3\right) \qquad n = 0, 1, \dots N-1 \end{cases}$$

Voici deux exemples:

- schéma de Heun à trois étages

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ \frac{2}{3} & 0 & \frac{2}{3} & 0 \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} \end{array} \quad \Longrightarrow \begin{cases} u_0 = y_0 \\ K_1 = \varphi\left(t_n, u_n\right) \\ K_2 = \varphi\left(t_n + \frac{h}{3}, u_n + \frac{h}{3} K_1\right) \\ K_3 = \varphi\left(t_n + \frac{2}{3}h, u_n + \frac{2}{3}hK_2\right) \\ u_{n+1} = u_n + \frac{h}{4}(K_1 + 3K_3) \qquad n = 0, 1, \dots N-1 \end{cases}$$

Étant une méthode explicite à 3 étapes, elle a au mieux ordre 3. Vérifions qu'elle est effectivement d'ordre 3:

In [4]:
```
c=[0,1/3,2/3]
b=[1/4,0,3/4]
A=[[0,0,0],[1/3,0,0],[0,2/3,0]]
s=len(c)
print("Consistance")
print(sum(b)==1)
for i in range(s):
    print(sum(A[i])==c[i])
print("\nOrdre 2")
print(sum([b[i]*c[i] for i in range(s)])==1/2)
print("\nOrdre 3")
print(sum([b[i]*c[i]**2 for i in range(s)])==1/3)
print(sum([b[i]*A[i][j]*c[j] for i in range(s) for j in range(s)])==1/6)
```

```
Consistance
True
True
True
True
True

Ordre 2
True

Ordre 3
True
True
```

- schéma

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 1 & -1 & 2 & 0 \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array} \quad \Longrightarrow \quad \begin{cases} u_0 = y_0 \\ K_1 = \varphi\left(t_n, u_n\right) \\ K_2 = \varphi\left(t_n + \frac{h}{2}, u_n + \frac{h}{2} K_1\right) \\ K_3 = \varphi\left(t_{n+1}, u_n + h(-K_1 + 2K_2)\right) \\ u_{n+1} = u_n + \frac{h}{6}(K_1 + 4K_2 + K_3) \qquad n = 0, 1, \ldots N-1 \end{cases}$$

Étant une méthode explicite à 3 étapes, elle a au mieux ordre 3. On vérifie ci-dessous qu'elle est d'ordre 3:

In [5]:
```
from fractions import Fraction
c=[0,Fraction(1,2),1]
b=[Fraction(1,6),Fraction(2,3),Fraction(1,6)]
A=[[0,0,0],[Fraction(1,2),0,0],[-1,2,0]]
s=len(c)
print("Consistance")
print(sum(b)==1)
for i in range(s):
    print(sum(A[i])==c[i])
print("\nOrdre 2")
print(sum([b[i]*c[i] for i in range(s)])==Fraction(1,2))
print("\nOrdre 3")
print(sum([b[i]*c[i]**2 for i in range(s)])==Fraction(1,3))
print(sum([b[i]*A[i][j]*c[j] for i in range(s) for j in range(s)])==Fraction(1,6))
```

```
Consistance
True
True
True
True

Ordre 2
True

Ordre 3
True
True
```

- schéma

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 1 & -1 & 2 & 0 \\ \hline & -\frac{1}{6} & \frac{4}{3} & -\frac{1}{6} \end{array} \quad \Longrightarrow \quad \begin{cases} u_0 = y_0 \\ K_1 = \varphi\left(t_n, u_n\right) \\ K_2 = \varphi\left(t_n + \frac{h}{2}, u_n + \frac{h}{2} K_1\right) \\ K_3 = \varphi\left(t_{n+1}, u_n + h(-K_1 + 2K_2)\right) \\ u_{n+1} = u_n + \frac{h}{6}(-K_1 + 8K_2 - K_3) \qquad n = 0, 1, \ldots N-1 \end{cases}$$

Étant une méthode explicite à 3 étapes, elle a au mieux ordre 3. On vérifie ci-dessous qu'elle n'est que d'ordre 2:

In [6]:
```
c=[0,Fraction(1,2),1]
b=[-Fraction(1,6),Fraction(4,3),-Fraction(1,6)]
A=[[0,0,0],[Fraction(1,2),0,0],[-1,2,0]]
s=len(c)
print("Consistance")
print(sum(b)==1)
for i in range(s):
    print(sum(A[i])==c[i])
print("\nOrdre 2")
print(sum([b[i]*c[i] for i in range(s)])==Fraction(1,2))
print("\nOrdre 3")
print(sum([b[i]*c[i]**2 for i in range(s)])==Fraction(1,3))
print(sum([b[i]*A[i][j]*c[j] for i in range(s) for j in range(s)])==Fraction(1,6))
```

```
Consistance
True
True
True
True
True

Ordre 2
True

Ordre 3
False
False
```

Exemples à 4 étages explicites

Un schéma RK à 4 étages explicite a pour matrice de Butcher

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ c_2 & a_{21} & 0 & 0 & 0 \\ c_3 & a_{31} & a_{32} & 0 & 0 \\ c_4 & a_{41} & a_{42} & a_{43} & 0 \\ \hline & b_1 & b_2 & b_3 & b_4 \end{array} \quad \text{avec} \quad \begin{cases} c_2 = a_{21} \\ c_3 = a_{31} + a_{32} \\ c_4 = a_{41} + a_{42} + a_{43} \\ b_1 + b_2 + b_3 + b_4 = 1 \end{cases} \quad \Longrightarrow \quad \begin{cases} u_0 = y_0 \\ K_1 = \varphi\left(t_n, u_n\right) \\ K_2 = \varphi\left(t_n + hc_2, u_n + h(a_{21}K_1)\right) \\ K_3 = \varphi\left(t_n + hc_3, u_n + h(a_{31}K_1 + a_{32}K_2)\right) \\ K_4 = \varphi\left(t_n + hc_4, u_n + h(a_{41}K_1 + a_{42}K_2 + a_{43}K_3)\right) \\ u_{n+1} = u_n + h\left(b_1K_1 + b_2K_2 + b_3K_3 + b_4K_4\right) \qquad n = 0, 1, \dots N-1 \end{cases}$$

Voici deux exemples (le premier est le plus connus):

- schéma RK4-1 ("La" méthode de Runge-Kutta)

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array} \quad \Longrightarrow \quad \begin{cases} u_0 = y_0 \\ K_1 = \varphi\left(t_n, u_n\right) \\ K_2 = \varphi\left(t_n + \frac{h}{2}, u_n + \frac{h}{2}K_1\right)\big) \\ K_3 = \varphi\left(t_n + \frac{h}{2}, u_n + \frac{h}{2}K_2\right) \\ K_4 = \varphi\left(t_{n+1}, u_n + hK_3\right) \\ u_{n+1} = u_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \quad n = 0, 1, \dots N-1 \end{cases}$$

  Il est bien d'ordre 4:

In [7]:
```
c=[0,Fraction(1,2),Fraction(1,2),1]
b=[Fraction(1,6),Fraction(1,3),Fraction(1,3),Fraction(1,6)]
A=[[0,0,0,0],[Fraction(1,2),0,0,0],[0,Fraction(1,2),0,0],[0,0,1,0]]
s=len(c)
print("Consistance")
print(sum(b)==1)
for i in range(s):
    print(sum(A[i])==c[i])
print("\nOrdre 2")
print(sum([b[i]*c[i] for i in range(s)])==Fraction(1,2))
print("\nOrdre 3")
print(sum([b[i]*c[i]**2 for i in range(s)])==Fraction(1,3))
print(sum([b[i]*A[i][j]*c[j] for i in range(s) for j in range(s)])==Fraction(1,6))
print("\nOrdre 4")
print(sum([b[i]*c[i]**3 for i in range(s)])==Fraction(1,4))
print(sum([b[i]*c[i]*A[i][j]*c[j] for i in range(s) for j in range(s)])==Fraction(1,8))
print(sum([b[i]*A[i][j]*c[j]**2 for i in range(s) for j in range(s)])==Fraction(1,12))
print(sum([b[i]*A[i][j]*A[j][k]*c[k] for i in range(s) for j in range(s) for k in range(s)])==Fraction(1,24))
```

```
Consistance
True
True
True
True
True

Ordre 2
True

Ordre 3
True
True

Ordre 4
True
True
True
True
```

- schéma RK4-2

$$
\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
\frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 \\
\frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\
1 & 1 & -2 & 2 & 0 \\
\hline
 & \frac{1}{6} & 0 & \frac{2}{3} & \frac{1}{6}
\end{array}
\quad \Longrightarrow \quad
\begin{cases}
u_0 = y_0 \\
K_1 = \varphi\left(t_n, u_n\right) \\
K_2 = \varphi\left(t_n + \frac{h}{4}, u_n + \frac{h}{4} K_1\right)\right) \\
K_3 = \varphi\left(t_n + \frac{h}{2}, u_n + \frac{h}{2} K_2\right) \\
K_4 = \varphi\left(t_{n+1}, u_n + h(K_1 - 2K_2 + 2K_3)\right) \\
u_{n+1} = u_n + \frac{h}{6}(K_1 + 4K_3 + K_4) \qquad\qquad n = 0, 1, \ldots N-1
\end{cases}
$$

Il est bien d'ordre 4:

In [8]:
```
c=[0,Fraction(1,4),Fraction(1,2),1]
b=[Fraction(1,6),0,Fraction(2,3),Fraction(1,6)]
A=[[0,0,0,0],[Fraction(1,4),0,0,0],[0,Fraction(1,2),0,0],[1,-2,2,0]]
s=len(c)
print("Consistance")
print(sum(b)==1)
for i in range(s):
    print(sum(A[i])==c[i])
print("\nOrdre 2")
print(sum([b[i]*c[i] for i in range(s)])==Fraction(1,2))
print("\nOrdre 3")
print(sum([b[i]*c[i]**2 for i in range(s)])==Fraction(1,3))
print(sum([b[i]*A[i][j]*c[j] for i in range(s) for j in range(s)])==Fraction(1,6))
print("\nOrdre 4")
print(sum([b[i]*c[i]**3 for i in range(s)])==Fraction(1,4))
print(sum([b[i]*c[i]*A[i][j]*c[j] for i in range(s) for j in range(s)])==Fraction(1,8))
print(sum([b[i]*A[i][j]*c[j]**2 for i in range(s) for j in range(s)])==Fraction(1,12))
print(sum([b[i]*A[i][j]*A[j][k]*c[k] for i in range(s) for j in range(s) for k in range(s)])==Fraction(1,24))
```

```
Consistance
True
True
True
True
True

Ordre 2
True

Ordre 3
True
True

Ordre 4
True
True
True
True
```

- schéma RK4-3 (règle 3/8)

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ \frac{2}{3} & -\frac{1}{3} & 1 & 0 & 0 \\ 1 & 1 & -1 & 1 & 0 \\ \hline & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \end{array} \quad \Longrightarrow \quad \begin{cases} u_0 = y_0 \\ K_1 = \varphi\left(t_n, u_n\right) \\ K_2 = \varphi\left(t_n + \frac{h}{3}, u_n + \frac{h}{3} K_1\right) \\ K_3 = \varphi\left(t_n + \frac{2}{3}h, u_n + \frac{h}{3}(-K_1 + 3K_2)\right) \\ K_4 = \varphi\left(t_{n+1}, u_n + h(K_1 - K_2 + K_3)\right) \\ u_{n+1} = u_n + \frac{h}{8}(K_1 + 3K_2 + 3K_3 + K_4) \qquad n = 0, 1, \ldots N-1 \end{cases}$$

Il est bien d'ordre 4:

In [9]:
```
from fractions import Fraction
c=[0,Fraction(1,3),Fraction(2,3),1]
b=[Fraction(1,8),Fraction(3,8),Fraction(3,8),Fraction(1,8)]
A=[[0,0,0,0],[Fraction(1,3),0,0,0],[-Fraction(1,3),1,0,0],[1,-1,1,0]]
s=len(c)
print("Consistance")
print(sum(b)==1)
for i in range(s):
    print(sum(A[i])==c[i])
print("\nOrdre 2")
print(sum([b[i]*c[i] for i in range(s)])==Fraction(1,2))
print("\nOrdre 3")
print(sum([b[i]*c[i]**2 for i in range(s)])==Fraction(1,3))
print(sum([b[i]*A[i][j]*c[j] for i in range(s) for j in range(s)])==Fraction(1,6))
print("\nOrdre 4")
print(sum([b[i]*c[i]**3 for i in range(s)])==Fraction(1,4))
print(sum([b[i]*c[i]*A[i][j]*c[j] for i in range(s) for j in range(s)])==Fraction(1,8))
print(sum([b[i]*A[i][j]*c[j]**2 for i in range(s) for j in range(s)])==Fraction(1,12))
print(sum([b[i]*A[i][j]*A[j][k]*c[k] for i in range(s) for j in range(s) for k in range(s)])==Fraction(1,24))
```

```
Consistance
True
True
True
True
True
True

Ordre 2
True

Ordre 3
True
True

Ordre 4
True
True
True
True
```

Exemple à 5 étages explicite

Un schéma RK à 5 étages explicite a pour matrice de Butcher

$$\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ c_2 & a_{21} & 0 & 0 & 0 & 0 \\ c_3 & a_{31} & a_{32} & 0 & 0 & 0 \\ c_4 & a_{41} & a_{42} & a_{43} & 0 & 0 \\ c_5 & a_{51} & a_{52} & a_{53} & a_{54} & 0 \\ \hline & b_1 & b_2 & b_3 & b_4 & b_5 \end{array} \quad \Longrightarrow \quad \begin{cases} u_0 = y_0 \\ K_1 = \varphi\left(t_n, u_n\right) \\ K_2 = \varphi\left(t_n + hc_2, u_n + h(a_{21}K_1)\right) \\ K_3 = \varphi\left(t_n + hc_3, u_n + h(a_{31}K_1 + a_{32}K_2\right) \\ K_4 = \varphi\left(t_n + hc_4, u_n + h(a_{41}K_1 + a_{42}K_2 + a_{43}K_3)\right) \\ K_5 = \varphi\left(t_n + hc_5, u_n + h(a_{51}K_1 + a_{52}K_2 + a_{53}K_3 + a_{54}K_4)\right) \\ u_{n+1} = u_n + h\left(b_1K_1 + b_2K_2 + b_3K_3 + b_4K_4 + b_5K_5\right) \qquad n = 0, 1, \dots N-1 \end{cases}$$

avec $\begin{cases} c_2 = a_{21} \\ c_3 = a_{31} + a_{32} \\ c_4 = a_{41} + a_{42} + a_{43} \\ c_5 = a_{51} + a_{52} + a_{53} + a_{54} \\ b_1 + b_2 + b_3 + b_4 + b_5 = 1 \end{cases}$

Voici un exemple:

- schéma de Merson

$$\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{8} & 0 & \frac{3}{8} & 0 & 0 \\ 1 & \frac{1}{2} & 0 & -\frac{3}{2} & 2 & 0 \\ \hline & \frac{1}{6} & 0 & 0 & \frac{2}{3} & \frac{1}{6} \end{array} \quad \Longrightarrow \quad \begin{cases} u_0 = y_0 \\ K_1 = \varphi\left(t_n, u_n\right) \\ K_2 = \varphi\left(t_n + \frac{h}{3}, u_n + \frac{h}{3}K_1\right) \\ K_3 = \varphi\left(t_n + \frac{h}{3}, u_n + \frac{h}{6}(K_1 + K_2)\right) \\ K_4 = \varphi\left(t_n + \frac{h}{2}, u_n + \frac{h}{8}(K_1 + 3K_3)\right) \\ K_5 = \varphi\left(t_n + h, u_n + \frac{h}{2}(K_1 - 3K_3 + 4K_4)\right) \\ u_{n+1} = u_n + \frac{h}{6}(K_1 + 4K_4 + K_5) \qquad n = 0, 1, \dots N-1 \end{cases}$$

In [10]:
```
from fractions import Fraction
c=[0,Fraction(1,3),Fraction(1,3),Fraction(1,2),1]
b=[Fraction(1,6),0,0,Fraction(2,3),Fraction(1,6)]
A=[[0,0,0,0,0],[Fraction(1,3),0,0,0,0],[Fraction(1,6),Fraction(1,6),0,0,0],[Fraction(1,8),0,Fraction(3,8),0,0],[Fraction(1,2),0,
-Fraction(3,2),2,0]]
s=len(c)
print("Consistance")
print(sum(b)==1)
for i in range(s):
    print(sum(A[i])==c[i])
print("\nOrdre 2")
print(sum([b[i]*c[i] for i in range(s)])==Fraction(1,2))
print("\nOrdre 3")
print(sum([b[i]*c[i]**2 for i in range(s)])==Fraction(1,3))
print(sum([b[i]*A[i][j]*c[j] for i in range(s) for j in range(s)])==Fraction(1,6))
print("\nOrdre 4")
print(sum([b[i]*c[i]**3 for i in range(s)])==Fraction(1,4))
print(sum([b[i]*c[i]*A[i][j]*c[j] for i in range(s) for j in range(s)])==Fraction(1,8))
print(sum([b[i]*A[i][j]*c[j]**2 for i in range(s) for j in range(s)])==Fraction(1,12))
print(sum([b[i]*A[i][j]*A[j][k]*c[k] for i in range(s) for j in range(s) for k in range(s)])==Fraction(1,24))
```

```
Consistance
True
True
True
True
True
True

Ordre 2
True

Ordre 3
True
True

Ordre 4
True
True
True
True
```