Remerciements

Avant tout, tous mes remerciements à Dieu de m'avoir facilité les tâches en réalisant le mémoire de fin d'étude $2^{ème}$ cycle LMD en vue d'obtenir le diplôme de master en mathématiques, Comme je remercie infiniment mon père pour son soutien financier qui m'a apporté durant tout le parcourt de mes études et j'adresse également mes sincères remerciements à ma chère mère de m'avoir réservé un climat de préparation de travail favorable et de leur conseils qui m'ont aidé à concrétiser mon objectif et grâce à leur bonne orientation j'ai pu arriver à découvrir la filière qui répond à mes connaissances scientifiques, comme, je remercie chaleureusement l'encadreur **MERZOUGUI ABDELKRIM** d'avoir accepté de diriger et suivre mon travail dans le but de réaliser le mémoire de fin d'études $2^{ème}$ cycle LMD pour l'obtention du diplôme master en mathématiques ainsi pour son soutien, son sérieux, ses conseils et ses suggestions tout au long de ce travail et je tiens également à remercier les membres du jury qui m'ont fait l'honneur de juger ce modeste mémoire et venir proclamer le résultat de mon travail sans oublier de remercier tous ceux qui ont contribué de prés ou de loin à la réalisation de ce travail.

Table des matières

Introduction

Le présent document regroupe les differentes méthodes et techniques de calcul des équations différentielles qui permettent de résoudre de manière exacte ou le plus souvent de maniére approchée. Il peut s'agir de déterminer l'inconnue d'une équation de calcul la valeur d'une fonction en un point ou sur un intervalle d'intégrer une fonction. Les équations différentielles sont sollicitées dans plusieurs domaines à savoir, la physique, la mécanique et la biologie, etc.

L'étude de ce sujet a été répartie sous forme de trois chapitres qui présentent respectivement :

- généralités et définitions sur EDOs, Rappel sur la résolution classique des EDOs 1^{er} ordre et $2^{ème}$ ordre, le problème de cauchy, principe général des méthodes numériques, propriétés des méthodes numériques et les principales méthodes numériques.

- Méthode d'Euler explicite et implicite, méthode d'Euler améliorée, méthode d'Euler cauchy, méthode de runge kutta, méthode de burlish stoer, méthode de runge kutta fehlberg et comparaison entre toutes les méthodes, etc.

- Méthode Adams-Bashforth-moulton, méthode de Hamming, méthode de Milne Simpson, méthode de Gear, méthode de Tir et méthode des différences finies.

Chapitre 1

Notions préliminaires sur les EDOs

1.1 Introduction

Les équations différentielles ordinaires (**EDO**) apparaissent très souvent dans la modélisation de la physique et des sciences de l'ingénieur. Trouver la solution d'une **EDO** ou d'un système d'**EDO** est ainsi un problème courant, souvent difficile ou impossible à résoudre de façon analytique. Il est alors nécessaire de recourir à des méthodes numériques pour résoudre ces équations différentielles.

1.2 Généralités et définitions sur EDOs

Définition 1.2.1 *Une équation différentielle d'un (premier ordre) de la forme* $y' = f(x, y)$ *exprime le taux de changement de la variable dépendante* y *en ce qui concerne un changement de la variable indépendante* x *comme fonction* $f(x, y)$ *de la variable indépendante* x *et de la variable dépendante* y*.*

Théorème 1.2.1 *[4] (existence et unicité)*

Si la fonction f *est continue sur* $\mathbb{R}^+ \times \mathbb{R}$ *et verifie la condition de Lipschtiz il existe une constante* $L\rangle 0$ *telle que*

$|f(t, v) - f(t, w)| \leq L\,|v - w|\,, \forall v, w \in \mathbb{R}, \forall t\rangle 0$

alors le problème de cauchy admet une solution globale (c-à-d pour tout $t\rangle 0$*) et elle est unique.*

Remarque 1.2.1 *Notons que si la condition de Lipschitz n'est pas satisfaisante, alors il n'existe pas une solution ou il peut exister plus d'une solution. Si la fonction $f(x,y)$ est indépendante de la variable dépendante y, alors la solution est*

$$y(x) = y_0 + \int_{x_0}^{x} f(t)dt$$

Définition 1.2.2 *Soit une fonction $y(x)$ définie sur un intervalle de $\mathbb{R}$ et de classe C^p (continûment dérivable d'ordre p). On appelle équation différentielle d'ordre p une équation de la forme :*

$$F\left(x, y, y', y", \ldots, y^{(p)}\right) = 0 \tag{1.1}$$

*On appelle forme canonique d'une **EDO** une expression du type :*

$$y^{(p)} = f\left(x, y, y', y", \ldots, y^{(p-1)}\right)$$

Toute équation différentielle canonique peut être écrite comme un système d'équations différentielles du premier ordre en introduisant $p-1$ fonctions définies comme :

$$\begin{cases} y_1 &= y \\ y_2 &= y' \\ \cdots & \cdots \\ y_p &= y^{(p-1)} \end{cases}$$

*L'équation canonique se met sous la forme du système d'**EDO** d'ordre 1 suivant :*

$$\begin{cases} y_1^{'} &= y_2 \\ y_2^{'} &= y_3 \\ \cdots & \cdots \\ y_p^{'} &= f\left(x, y, y_1, y_2 \cdots, y_p\right) \end{cases}$$

Définition 1.2.3 *Résoudre une équation différentielle $F(x, y, y^{'}, y", ..., y^{(n)}) = 0$ sur un intervalle $I \subset \mathbb{R}$ ou $\mathbb{R}$ tout entier, c'est trouver toutes les fonctions f telles que :*

a)- f soit n fois dérivable sur I

b)-$\forall x \in I, F(x, f(x), f^{'}(x), f"(x), ..., f^{(n)}(x)) = 0$

Une fonction qui vérifie les conditions (a) et (b) est appelée solution de l'équation différentielle et sa courbe représentative est appelée courbe intégrale de l'équation différentielle.

On appelle courbes intégrales d'une équation différentielle l'ensemble des courbes représentatives de toutes les solutions de cette équation différentielle.

1.3 Résolution classique des EDOs 1^{er} ordre

Equations de Bernoulli $a(x)y^{'}+b(x)y+c(x)y^{\alpha}=0$	$t=y^{1-\alpha}$ on obtient $t^{'}=(1-\alpha)\frac{y^{'}}{y^{\alpha}}$. $a(x)y^{'}+b(x)y+c(x)y^{\alpha}=0$ $\Longleftrightarrow a(x)\frac{y^{'}}{y^{\alpha}}+b(x)\frac{y}{y^{\alpha}}+c(x)=0 \Longleftrightarrow \frac{a(x)}{1-a}t^{'}+b(x)t+c(x)=0$ on obtient donc une équation linéaire du premier ordre en t.
Equations de Ricatti $y^{'}=a(x)y^2+b(x)y+c(x)$	On pose $y=y_1+\frac{1}{t}$ avec $y_1^{'}=a(x)y_1^2+b(x)y_1+c(x)$ et t ne s'annulant pas sur l'intervalle de résolution on a $y^{'}=y_1^{'}+\frac{t^{'}}{t^2}$ $y^{'}=a(x)y^2+b(x)y+c(x) \Longleftrightarrow y_1^{'}-\frac{t^{'}}{t^2}=a(x)\left(y_1+\frac{1}{t}\right)^2+b(x)\left(y_1+\frac{1}{t}\right)+c(x) \Longleftrightarrow \frac{-t^{'}}{t^2}=2a(x)\frac{y_1}{t}+a(x)\frac{1}{t^2}+b(x)\frac{1}{t} \Longleftrightarrow$ $t^{'}+(2a(x)y_1+b(x))t=-a(x)$. on obtient donc une équation linéaire du premier ordre en t.
Equations de Lagrange $y=xf(y^{'})+g(y^{'})$	On pose $t=y^{'}=\frac{\partial y}{\partial x}$. L'équation devient $y=xf(t)+g(t)$ $=x(t)f(t)+g(t)$ On la dérive par rapport à t: $\frac{\partial y}{\partial t}=x^{'}(t)f(t)+x(t)f^{'}(t)+g^{'}(t)$Donc $tx^{'}(t)=x^{'}(t)f(t)+x(t)f(t)+g^{'}(t)$

1.4 Résolution classique des EDOs du second ordre à variable constante

Equations homogènes

Définition 1.4.1 *On appelle équation différentielle, homogène, linéaire du second ordre à coefficients constants toute équation différentielle qui peut se mettre sous la forme :*

$$ay^{"}+by^{'}+cy=0 \tag{1.4.1}$$

où a, b et $c\in\mathbb{R}$ $(a\neq 0)$.

Définition 1.4.2 *On appelle polynôme caractéristique de l'équation (1.4.1), le polynôme $P=a\lambda^2+b\lambda+c$. dont le discriminant et le réel $\triangle=b^2-4ac$.*

si r_1, $r_2$sont les racines de p $(\Delta\rangle 0)$	$y=c_1e^{r_1x}+c_2e^{r_2x}$ c_1, $c_2\in\mathbb{R}$
si r racine de p $(\Delta=0)$	$y=(c_1x+c_2)e^{rx}$
si $r_1=\alpha+bi$ et $r_2=\alpha-bi$ sont les racines de p $(\Delta\langle 0)$	$y=e^{\alpha x}(c_1\cos(bx)+c_2\sin(bx))$

Equations non homogènes

Définition 1.4.3 *On appelle équation différentielle,non homogène, linéaire du second ordre à coefficients constants toute équation différentielle qui peut se mettre sous la forme :*

$$ay^{"} + by^{'} + cy = f(x) \tag{1.4.2}$$

la solution de (1.4.2) est de la forme :$y = y_h + y_p$ telle que
y_h la solution de $ay^{"} + by^{'} + cy = 0$
y_p admet plusieurs formes basées sur la forme de $f(x)$.

La fonction $f(x)$	Les solutions particulières y_p
$f(x) = p_n(x) e^{\beta x}$	$y_p = p_n(x) e^{\beta x}$ $y_p = x^K p_n(x) e^{\beta x}$
$f(x) = p_n(x)\cos(wx) + q_m(x)\sin(wx)$	$y_p = p_n(x)\cos(wx) + q_m(x)\sin(wx)$ $y_p = x^K [p_n(x)\cos(wx) + q_m(x)\sin(wx)]$
$f(x) = [p_n(x)\cos(wx) + q_m(x)\sin(wx)] e^{\beta x}$	$y_p = e^{\beta x}[p_j(x)\cos(wx) + q_j(x)\sin(wx)]$ $y_p = x^K e^{\beta x}[p_n(x)\cos(wx) + q_m(x)\sin(wx)]$

1.5 Le problème de cauchy

Définition 1.5.1 *Le problème est bien posé virifie l'éxistence de la solution,l'unicité de la solution et la stabilité par rapport à la condition initiale*

Définition 1.5.2 *On appelle problème de Cauchy ou problème à la valeur initiale le problème qui consiste à trouver une fonction $y(x)$ définie sur l'intervalle $[a,b]$ telle que :*

$$\begin{cases} y^{'}(x) = f(x, y(x)) \quad ; \quad \forall x \in [a,b] \\ y(a) = y_0 \end{cases}$$

et vérifie le théorème d'éxistence et d'unicité, on dit que le problème est bien posé.

1.6 Principe géneral des méthodes numériques

Pour obtenir une approximation numérique de la solution $y(x)$ sur l'intervalle $[a, b]$, nous allons estimer la valeur de cette fonction en un nombre fini de points x_i, pour $i = 1, \ldots, n$, constituants les noeuds du maillage. La solution numérique discrète obtenue aux points x_i est notée $y_i = y(x_i)$. Le pas de discrétisation est noté h , ce pas est géneralement équidistant.

Les techniques de résolution des **EDOs** sont basées sur :

1. l'approximation géométrique de la fonction
2. les formules d'intégration numérique (rectangle, trapèze, Simpson...)
3. les développements de Taylor au voisinage de x_i

1.7 Propriétés des méthodes numériques

Plusieurs notions mathématiques sont introduites lors de la résolution d'**EDO** au moyen de leurs équivalents discrétisés. Les trois principales sont la **convergence**, la **stabilité** et la **consistance**, permettant de relier la solution exacte des équations continues à la solution exacte des équations discrétisées et à la solution numérique obtenue.

1.7.1 Consistance d'une méthode

La consistance est la propriété qui assure que la solution approchée de l'équation discrétisée tend vers la solution exacte de l'équation continue lorsque le pas de discrétisation h tend vers 0.

1.7.2 Stabilité d'une méthode

C'est la propriété qui assure que la différence entre la solution numérique obtenue et la solution exacte des équations discrétisées reste bornée. La stabilité indique si l'erreur augmente ou non au cours du calcul. Une méthode peut être stable sous condition (elle sera dite conditionnellement stable) ou toujours stable (elle sera dite inconditionnellement stable).

1.7.3 Ordre de précision d'une méthode

L'erreur de troncature ϵ est définie comme la différence entre la solution exacte $\tilde{y}$ et l'approximation numérique obtenue y_n, soit : $\epsilon_n = |\tilde{y}(x_n) - y_n| = o(h^p)$. L'ordre de précision de la méthode est donnée par l'entier p.

1.7.4 Convergence et taux de convergence d'une méthode

Une méthode est convergente si, lorsque le pas de discrétisation tend vers 0, la solution numérique tend vers la solution exacte de l'équation continue.

Une méthode est convergente à l'ordre l ssi :

$\lim_{n\to\infty} \max_i |\epsilon_i| = o\left(h^l\right)$

Résultat théorique : Une méthode stable et consistance est convergente.

1.8 Les principales méthodes numériques

Les principales méthodes de résolution numérique des **EDOs** sont séparées en deux grands types :

1.8.1 les méthodes à un pas

Pour ces méthodes, le calcul de la valeur discrète y_{n+1} au noeud x_{n+1} fait intervenir la valeur y_n obtenue à l'abscisse précédente.

Les principales méthodes sont :

- Méthodes d'**E**uler explicite et implicite
- Méthode d'**E**uler améliorée
- Méthode d'**E**uler-**C**auchy
- Méthode de **C**rank-**N**icholson
- Méthodes de **R**unge et **K**utta

1.8.2 les méthodes à pas multiples

Pour ces méthodes, le calcul de la valeur discrète y_{n+1} au noeud x_{n+1} fait intervenir plusieurs valeurs $y_n, y_{n-1}, y_{n-2}, \ldots$: obtenues aux abscisses précédentes.

Les principales méthodes sont :

- Méthodes d'**A**dams-**B**ashforth-Moulton
- Méthodes de **G**ear
- Méthode de **H**amming
- Méthodes de **T**ir

et nous allons étudier les méthodes sus indiquées en détail dans les chapitres deux et trois.

Chapitre 2

Méthode de résolution des EDOs à un pas

2.1 Introduction

Le but de ce chapitre est de montrer comment programmer facilement et efficacement les schémas classiques de résolution numérique des équations différentielles à l'aide de Matlab. On met en évidence leurs ordres de convergence respectifs sur un exemple simple.

2.2 Méthode à un pas

La formulation générale des méthodes à un pas explicite est :

$$\begin{cases} y_0 & \text{donné} \\ y_{n+1} & = & y_n + \phi\left(x_n, y_n, h\right) \end{cases}$$

Où la fonction ϕ définit la méthode utilisée.

La formulation générale des méthodes à un pas implicite est :

$$\begin{cases} y_0 & \text{donné} \\ y_{n+1} & = & y_n + \phi\left(x_n, y_n, y_{n+1}, h\right) \end{cases}$$

L'obtention de la solution à chaque abscisse nécessite la résolution d'une équation. Ces méthodes sont obtenues en intégrant l'équation différentielle et en utilisant des formules

d'intégration numérique pour le second membre. L'ordre du schéma est égal au degré du polynôme +1 pour lequel l'intégration est exacte.

Résultats théoriques :

1) Si la fonction ϕ est lipchitzienne par rapport à la deuxième variable alors les méthodes explicites sont stables.

2) Les méthodes implicites sont toujours stables.

3) Les méthodes sont consistances ssi

$$\forall x \in [a,b], \phi(x,y,0) = f(x,y)$$

2.2.1 Méthodes d'Euler explicite et implicite

En parlant des solutions numériques aux **EDOs**, chacune commence par la méthode d'Euler, puisqu'elle est facile à comprendre et simple pour programmer.

Explication de la méthode d'Euler :

On part d'un point M de coordonnées (X_M, Y_M) appartenant à la solution approchée. Le point suivant est le point E de coordonnées (X_E, Y_E), tel que $X_E = X_M + h$ et tel que le point E est sur la droite passant par M de pente $F(X_M, Y_M)$, On itère le procédé en partant du point (X_0, Y_0) et en reprenant à chaque étape le point E comme nouveau point M, autant de fois qu'il le faut pour que X décrive l'intervalle demandé.

Voir la figure:(2.2.1).

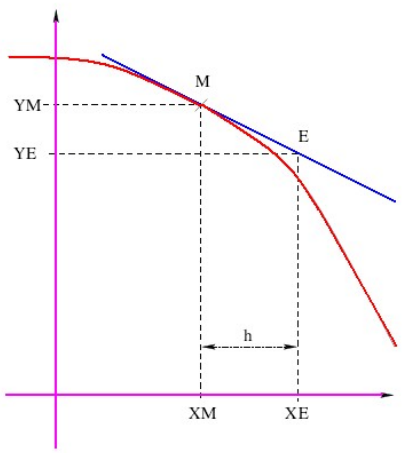


Figure 2.2.1 : Méthode d'Euler

Quoique sa basse exactitude la garde d'être employée couramment pour résoudre des odes, elle nous donne un indice au concept de base de la solution numérique pour une équation simplement et clairement. Considérons une équation de premier ordre :

$$y^{'}(t) + y(t) = 1 \qquad \text{avec} \qquad y(0) = 0 \tag{1.1}$$

Sa solution analytique est de la forme suivante :

$$y(t) = 1 - \exp(-t) \tag{1.2}$$

Cependant, une solution exacte n'existe pas pour chaque équation, même si elle existe, il n'est pas facile de trouver. C'est pourquoi nous devrions étudier les solutions numériques aux équations différentielles. Comment traduisons-nous l'équation en forme qui peut facilement être manipulée par ordinateur? Tout d'abord, nous devons remplacer la dérivée $y^{'}(t) = dy/dt$ dans l'équation par une dérivée numérique, où le pas «h» est déterminé sur les conditions d'exactitude et les contraintes de calcul.

$$\begin{cases} \frac{y(t+h)-y(t)}{h} + y(t) = 1 \\ y(t+h) = (1-h)\,y(t) + h \quad \text{avec } y(0) = 0 \end{cases} \tag{1.3}$$

Et résoudre cette équation différentielle point par point avec l'augmentation de t par h chaque fois, de $t = 0$.

$$\begin{cases} y(h) = (1-h)y(0) + h = (1-h)y_0 + h \\ y(2h) = (1-h)y(h) + h = (1-h)^2 y_0 + (1-h)h + h \\ y(3h) = (1-h)y(2h) + h = (1-h)^3 y_0 + \sum_{m=0}^{2} (1-h)^m h \end{cases} \tag{1.4}$$

(1.4) est une solution numérique de l'équation (1.1) avec la solution analytique (1.2).

la solution numérique (1.4) avec le pas $h = 0.5$ et $h = 0.25$ sont énumérées dans le tableau (2.2.1) et représentées dans la figure (2.2.2)

Tableau (2.2.1) une solution numérique de l'équation (1.1) obtenue par la méthode d'Euler

t	$h = 0.5$	$h = 0.25$
0.25		$y(0.25) = (1 - \ h)y_0 + h \ = 1/4 = 0.25$
0.50	$y(0.50) = (1 - \ h)y_0 + h \ = 1/2 = 0.5$	$y(0.50) = (3/4)y(0.25) + 1/4 = 0.4375$
0.75		$y(0.75) = (3/4)y(0.50) + 1/4 = 0.5781$
1.00	$y(1.00) = (1/2)y(0.5) + 1/2 = 3/4 = 0.75$	$y(1.00) = (3/4)y(0.75) + 1/4 = 0.6836$
1.25		$y(1.25) = (3/4)y(1.00) + 1/4 = 0.7627$
1.50	$y(1.50) = (1/2)y(1.0) + 1/2 = 7/8 = 0.875$	$y(1.50) = (3/4)y(1.25) + 1/4 = 0.8220$
...	. . . . .	. . . . .

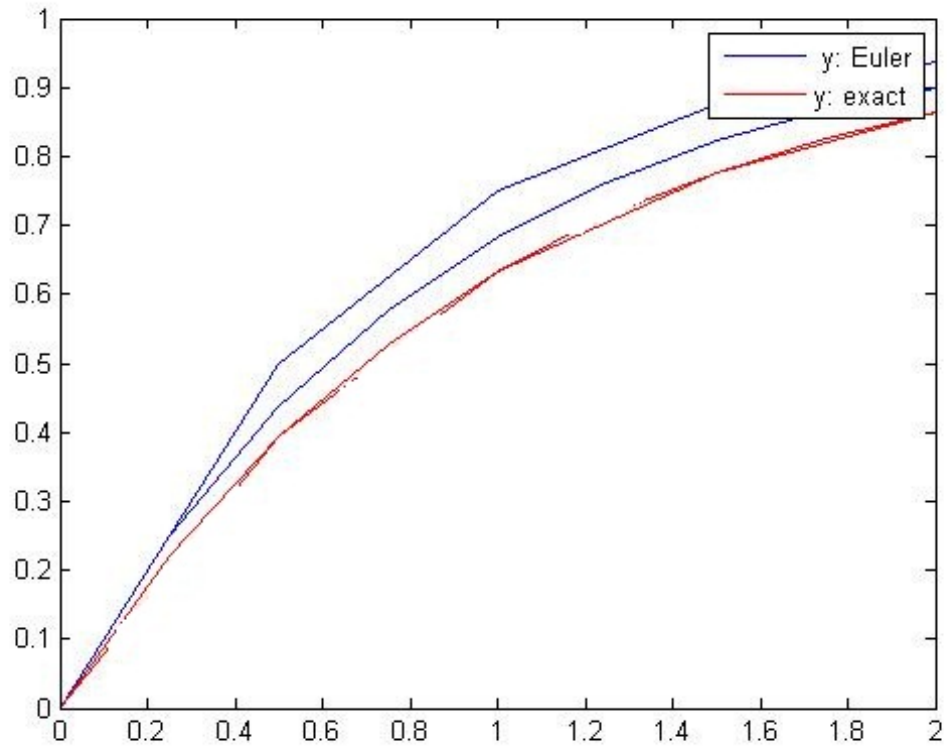


Figure 2.2.2 : Exemple de solution d'une équation différentielle par la méthode d'Euler.

La figure (2.2.2) nous indique que tellement le pas est petit la solution approchée est proche de la solution exacte ainsi que l'erreur diminue. L'algorithme peut être décrit comme :

$$y_{k+1} = y_k + hf(t_k, y_k) \qquad \text{avec } y(t_0) = y_0 \tag{1.6}$$

Méthode explicite d'ordre 1, dont l'algorithme est :

$$\begin{cases} y_0 \text{ donné } \in [a,b] \\ y_{n+1} = y_n + hf(x_n, y_n) \end{cases}$$

La méthode peut s'interpréter de plusieurs manières :

1) Via les formules d'intégration numérique :

la méthode est le résultat de l'application de la formule des rectangles basée au point x_n.

2) Géométriquement : la méthode revient à remplacer localement en chaque point x_n la courbe solution par sa tangente.

3) Via les développements de Taylor : la méthode provient du développement de Taylor d'ordre 1 de la fonction y au voisinage de x_n.

Méthode implicite d'ordre 1, dont l'algorithme est :

$$\begin{cases} y_0 \text{ donné } \in [a,b] \\ y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}) \end{cases}$$

2.2.2 Méthode d'Euler améliorée

Méthode explicite dont l'algorithme est :

$$\begin{cases} y_0 & \text{donné} & \in [a,b] \\ y^*_{n+1} & = & y_n + \frac{h}{2} f(x_n, y_n) \\ y_{n+1} & = & y_n + \frac{h}{2}\left[f(x_n, y_n) + f(x_{n+1}, y^*_{n+1})\right] \end{cases}$$

Géométriquement, la méthode consiste à remplacer dans la méthode d'Euler la pente de la tangente en (x_n, y_n) par la valeur corrigée au milieu de l'intervalle $[x_n, x_{n+1}]$.

2.2.3 Méthode d'Euler-Cauchy

Cette méthode est dite aussi la méthode de Newton-Cauchy modifiée ou méthode du point milieu. C'est une méthode explicite dont l'algorithme est :

$$\begin{cases} y_0 & \text{donné} & \in [a,b] \\ y^*_{n+1} & = & y_n + \frac{h}{2} f\left(x_n, y_n\right) \\ y_{n+1} & = & y_n + hf\left(x_n + \frac{h}{2}, y^*_{n+1}\right) \end{cases}$$

La méthode du point milieu consiste à approcher, au voisinage de t_0, la solution de l'équation différentielle $y^{'}(t) = f(t, y(t))$, $y(t_0) = y_0$ $t_0 \in [a;b]$, par une parallèle à la tangente au "point milieu de la solution obtenue par la méthode d'Euler" :Le "point milieu de la solution obtenue par la méthode d'Euler" est le point de coordonnées :

$$\left(t_0 + \frac{h}{2}, y_0 + \frac{h}{2} f\left(t_0, y_0\right)\right)$$

Et sa tangente a donc comme pente :

$$\alpha = f\left(t_0 + \frac{h}{2}, y_0 + \frac{h}{2} f\left(t_0, y_0\right)\right)$$

La méthode du point milieu fait passer du point (t_0, y_0) au point (t_1, y_1) avec :

$$\begin{cases} t_1 & = & t_0 + h \\ y_1 & = & y_0 + h * \alpha \quad = \quad y_0 + h * f\left(t_0 + \frac{h}{2}, y_0 + \frac{h}{2} * f\left(t_0, y_0\right)\right) \end{cases}$$

qui réalise une seule étape de cette méthode et qui permet de passer d'un point d'abscisse t_0 au point d'abscisse $t_0 + h$, ces points étant situés sur le graphe de la solution approchée par cette méthode.

Remarque 2.2.1 *La méthode d'Euler est une méthode numérique peu coûteuse numériquement, mais peu précise.*

Des améliorations sont possibles dés que l'on considère des points intermédiaires, ce que nous allons voir ci-dessous en considérant des méthodes dites de Runge-Kutta.

Comparaison entre Euler explicite Euler modifié et Euler Cauchy :

En prenant la même équation précédente on trouve les résultats suivants :

Dans la figure (2.2.3) on remarque que seule la méthode d'Euler Cauchy s'approche de la solution exacte avec un pas de $h = 0.2$.

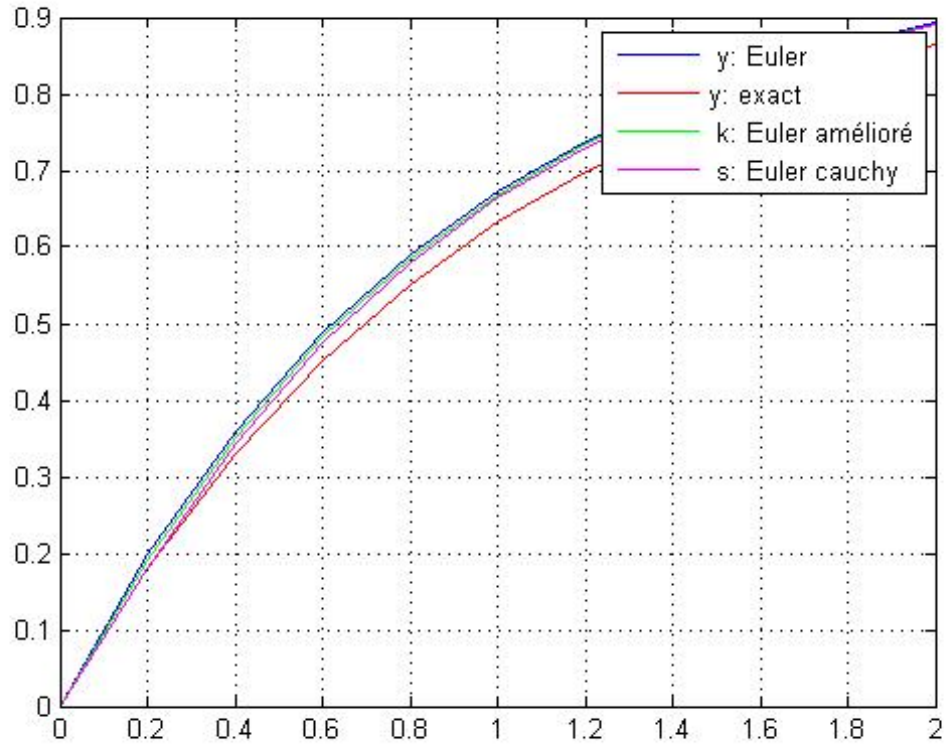


Figure 2.2.3 : Comparaison entre euler explicite, euler modifié et euler cauchy.

2.2.4 Méthode de Crank - Nicholson

Méthode implicite d'ordre 2 dont l'algorithme est donné par :

$$\begin{cases} y_0 & \text{donné} \\ y_{n+1} & = \quad y_n + \frac{h}{2}\left[f\left(x_n, y_n\right) + f\left(x_{n+1}, y_{n+1}\right)\right] \end{cases}$$

Elle est obtenue en utilisant la formule d'intégration numérique des trapèzes.

2.2.5 Méthode de Heun

Dans cette méthode (encore appelée méthode de Newton modifiée), on évalue l'incrément en deux étapes. La première étape consiste à écrire :

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} f\left(x_n, y_n\right) dx$$

en estimant l'intégrale par la méthode des trapèzes, soit :

$$y_{n+1} = y_n + \frac{h}{2}\left(f\left(x_n, y_n\right) + f\left(y_{n+1}, x_{n+1}\right)\right) \tag{1.8}$$

Cette formule est une formule implicite en y_{n+1}. De ce fait, on ne connaît pas encore y_{n+1} à ce stade. On trouve une valeur approchée de $f(y_{n+1}, x_{n+1})$ grâce à la méthode d'Euler. C'est la seconde étape. On a

$$\begin{cases} p_{n+1} & = & y_n + hf\left(y_n, x_n\right) \\ \approx & & y_{n+1} \end{cases}$$

La méthode de Heun met en jeu ce que l'on appelle un schéma de type prédicateur/correcteur. Le schéma prédicateur donne une estimation de y_{n+1} qui permet de prédire la valeur de la fonction en x_{n+1}. Le schéma correcteur associé est donné par la relation :

$$y_{n+1} = y_n + \frac{h}{2}\left[f\left(x_n, y_n\right) + f\left(y_n + hf\left(x_n, y_n\right), x_{n+1}\right)\right]$$

2.2.6 Méthode de Runge-Kutta

Méthode de Rungue Kutta 2

La méthode de Runge-Kutta d'ordre 2 , combine deux itérations successives de la méthode d'Euler explicite. Dans un premier temps la dérivée en (x_n, y_n) est évaluée pour faire une première estimation du point suivant (noté «A» voir la figure (2.2.6) ci dessous). L'estimation provisoire de y_{n+1} en ce point est ensuite utilisée pour affiner le calcul de la dérivée. Une nouvelle approximation de celle-ci est obtenue en prenant sa valeur à mi-parcours (prise au point B). C'est cette valeur de la dérivée qui sera ensuite utilisée pour estimer le prochain pas y_{n+1} (point C). La procédure d'intégration se résume à :

$$\begin{cases} \alpha & = & hf\left(x_n, y_n\right) \\ \beta & = & hf\left(x_n + \frac{h}{2}, y_n + \frac{\alpha}{2}\right) \\ y_{n+1} & = & y_n + \beta + o\left(h^3\right) \end{cases}$$

Même si cette méthode demande deux fois plus opérations de calcul que la méthode d'Euler explicite pour effectuer un seul pas, le résultat est plus précis et plus stable.

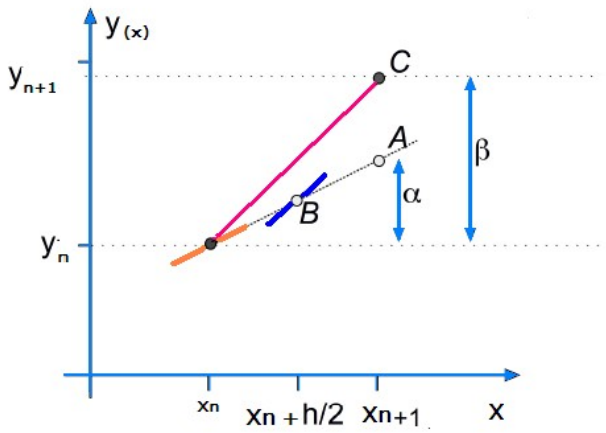


Figure 2.2.6 : Principe de fonctionnement de la méthode de Runge-Kutta du second ordre pour effectuer un seul pas

Une méthode explicite d'ordre 2, peut être obtenue par l'utilisation de la formule des trapèzes. L'algorithme, noté Rk_2, s'écrit :

$$\begin{cases} y_0 & \text{donné} \\ y^*_{n+1} & = & y_n + hf\left(x_n, y_n\right) \\ y_{n+1} & = & y_n + \frac{h}{2}\left[f\left(x_n, y_n\right) + f\left(x_{n+1}, y^*_{n+1}\right)\right] \end{cases}$$

Elle est obtenue en utilisant la formule d'intégration numérique des trapèzes.

Méthode de Runge et Kutta explicite à un point

$$\begin{cases} y_0 & \text{donné} \\ y^*_{n+1} & = & y_n + \frac{h}{2\alpha} f\left(x_n, y_n\right) \\ y_{n+1} & = & y_n + h\left(1-\alpha\right) f\left(x_n, y_n\right) + \alpha f\left(x_n + \frac{h}{2\alpha}, y^*_{n+1}\right) \end{cases}$$

Avec α un nombre réel compris entre 0 et 1. Les valeurs de α les plus utilisées sont $\alpha = 1$, $\alpha = 1/2$ et $\alpha = 3/4$. Ces méthodes sont d'ordre 2.

Méthodes RK implicites à un point

La formule de récurrence est définie par la relation :

$$y_{n+1} = y_n + h\left[\left(1-\theta\right) f\left(x_n, y_n\right) + \theta f\left(x_{n+1}, y_{n+1}\right)\right]$$

Où $\theta \in]0,1]$

1. Si $\theta = 0$, on retrouve la méthode d'Euler.
2. Si $\theta = 1/2$, la méthode est d'ordre 2 et s'appelle la méthode des trapèzes.
3. Si $\theta \neq 1/2$, la méthode est d'ordre 1.

Méthodes RK explicites à 2 points intermédiaires

Ces méthodes sont définies par les relations :

$$\begin{cases} y_{n,1} &= y_n + \frac{h}{3} f(x_n, y_n) \\ y_{n,2} &= y_n + \frac{2h}{3} f\left(x_n + \frac{h}{3}, y_{n,1}\right) \\ y_{n+1} &= y_n + \frac{h}{4}\left(f(x_n, y_n) + 3f\left(x_n + \frac{2h}{3}, y_{n,2}\right)\right) \end{cases}$$

ou par

$$\begin{cases} y_{n,1} &= y_n + \frac{h}{2} f(x_n, y_n) \\ y_{n,2} &= y_n + h\left(f(x_n, y_n) + 2f\left(x_n + \frac{h}{2}, y_{n,1}\right)\right) \\ y_{n+1} &= y_n + \frac{h}{6}\left(f(x_n, y_n) + 4f\left(x_n + \frac{h}{2}, y_{n,1}\right) + f(x_{n+1}, y_{n,2})\right) \end{cases}$$

Ces deux méthodes sont d'ordre 3. La première est parfois appelée méthode de Heun.

Méthodes RK explicites à 3 points intermédiaires

La méthode suivante est de loin la plus connue et utilisée. Les relations de récurrence sont les suivantes :

$$\begin{cases} y_{n,1} &= y_n + \frac{h}{2} f(x_n, y_n) \\ y_{n,2} &= y_n + \frac{h}{2} f\left(x_n + \frac{h}{2}, y_{n,1}\right) \\ y_{n,3} &= y_n + \frac{h}{2} f\left(x_n + \frac{h}{2}, y_{n,2}\right) \\ y_{n+1} &= y_n + \frac{h}{6}\left(f(x_n, y_n) + 2f\left(x_n + \frac{h}{2}, y_{n,1}\right) + 2f\left(x_n + \frac{h}{2}, y_{n,2}\right) + f(x_{n+1}, y_{n,3})\right) \end{cases}$$

Cette méthode est d'ordre 4.

Méthode RK d'ordre 4

La méthode de Runge-Kutta d'ordre 4 combine quatre itérations successives. Sa précision est généralement encore meilleure. La procédure d'intégration devient :

$$\begin{cases} y_0 & \text{donné} \\ k_1 & = & hf\left(x_n, y_n\right) \\ k_2 & = & hf\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right) \\ k_3 & = & hf\left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right) \\ k_4 & = & hf\left(x_n + h, y_n + k_3\right) \\ y_{n+1} & = & y_n + \frac{1}{6}\left[k_1 + 2k_2 + 2k_3 + k_4\right] \end{cases}$$

2.2.7 Méthode de Burlish - Stoer

L'idée de cette méthode repose sur les trois principes suivants :

1. la résolution de l'équation différentielle pour un accroissement de Δx est donnée par une fonction qui dépend de h, mais qui tend vers une limite finie (indépendante de h) , on cherche à estimer la valeur exacte du système différentiel à intégrer en calculant pour différents pas et en prenant la limite d'un pas tendant vers zéro. La figure (2.2.7) illustre graphiquement les différentes estimations de la valeur de la fonction solution en utilisant trois pas d'intégration différents. Ce point est très comparable à celui de la méthode de Romberg décrite pour l'intégration.

2. Le second principe de la méthode consiste à extrapoler cette limite non pas sur la base de développements polynomiaux mais de développements en fractions rationnelles.

3. Le troisième principe consiste à utiliser des fonctions d'erreur qui sont paires en pas d'intégration.

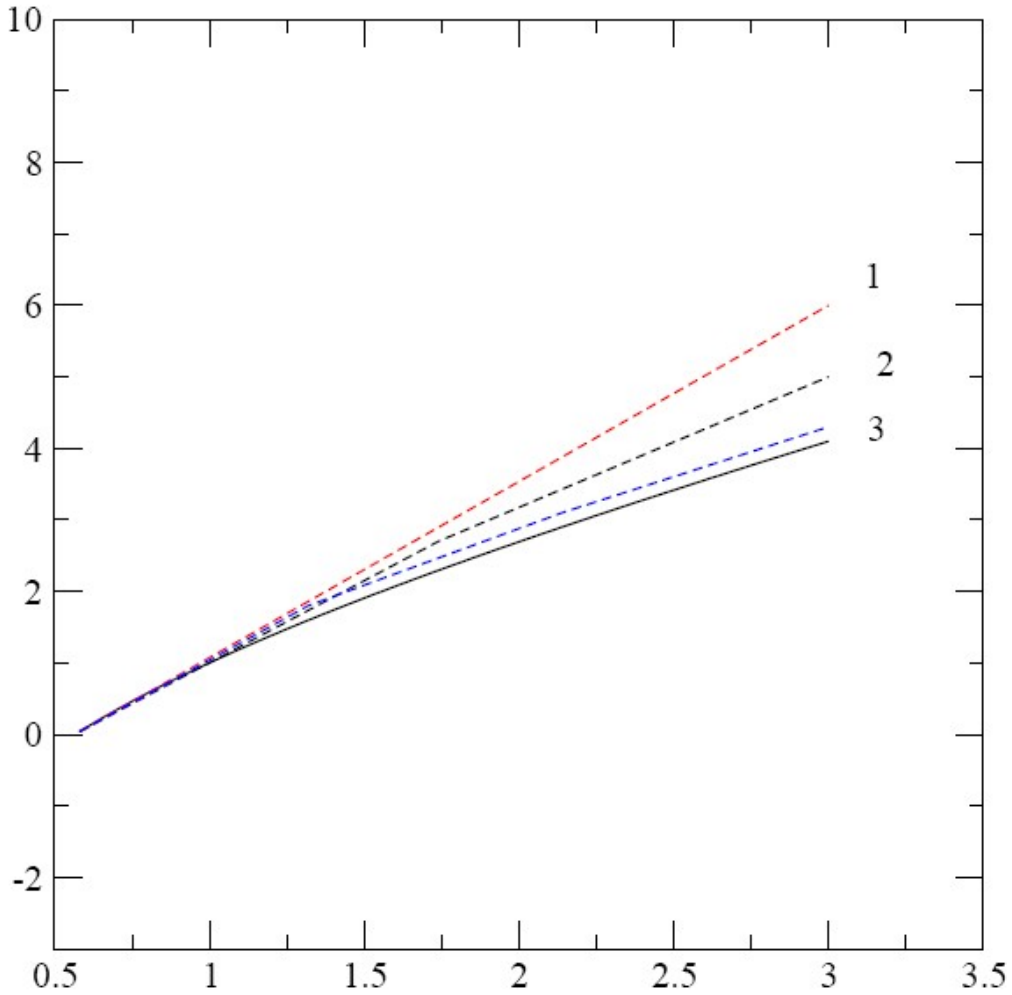


Figure 2.2.7 : Schéma illustrant le principe de la méthode de Burlish-Stoer. Les lignes en pointillés correspondent aux intégrations réalisées avec trois pas d'intégration différents. La courbe en trait plein correspond à la solution exacte.

2.3 Comparaison entre quelques méthodes

On va comparer l'allure des courbes pour quelques méthodes, que nous avons étudiées précédemment et voir la rapidité de chaque méthode Avec la fonction

$$y^{'}(t) = -y + 1 \qquad \text{avec} \qquad y(0) = 0$$

qui a pour solution exacte :

$$y(t) = 1 - exp(-t)$$

Pour l'allure des courbes on obtient la (2.3.1) :

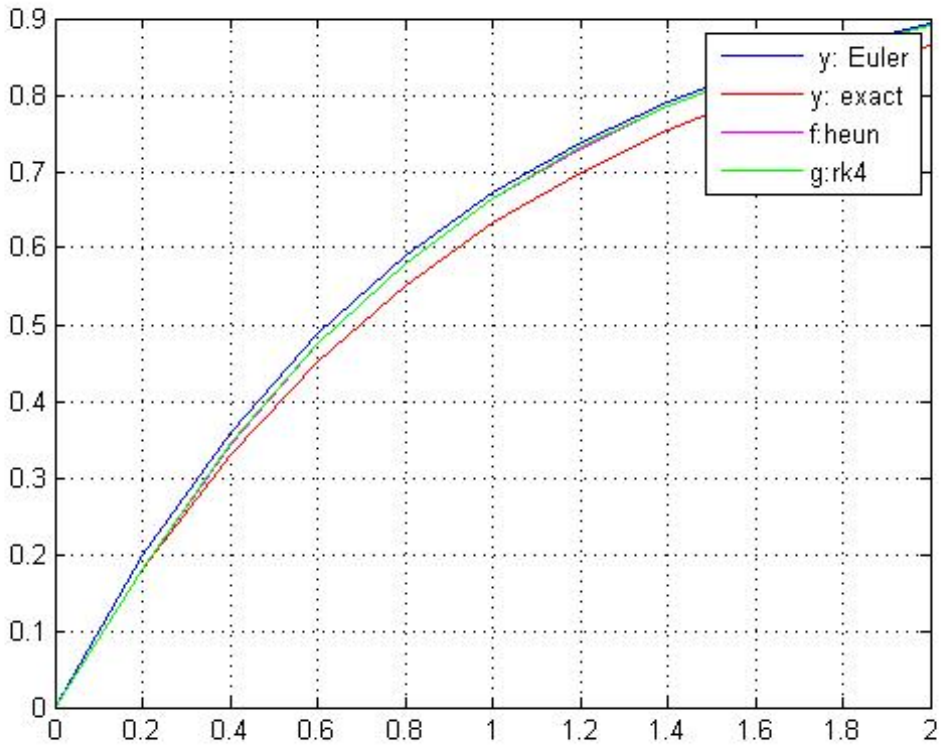


Figure 2.3.1 : Solution numérique d'une équation du 1^{er} ordre.

2.3.1 Comparaison entre toutes les méthodes

Sur la figure (2.3.2) on voit bien que Euler Explicite s'éloigne de la solution exacte par contre les méthodes de RK_4 et Heun sont les plus précises

erreur RK_4	=	$1.1979e-007$ c'est l'erreur la plus petite
erreur Euler explecite	=	0.0070
erreur Euleur Modifier	=	0.0034
erreur Heun	=	$2.3774e-004$
erreur euler cauchy	=	$2.3774e-004$

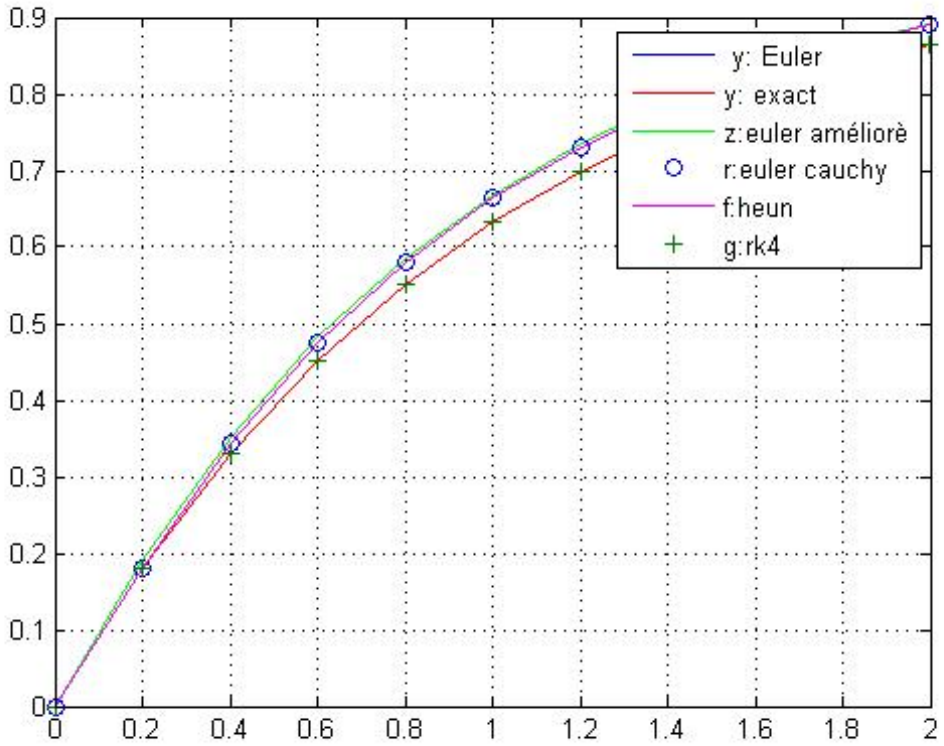


Figure 2.3.2 : La comparaison entre toutes les méthodes

2.4 La méthode de Runge-Kutta-Fehlberg

La façon pour garantir l'exactitude dans la solution d'un système d'équation différentielle est de résoudre le problème en employant deux fois les tailles du « pas » h et $h/2$ et de comparer les réponses aux points de maille correspondant à la taille du «pas» le plus grand. Mais ceci exige une quantité significative de calcul pour la taille du pas la plus petite et doit être répétée si on le détermine que l'accord n'est pas assez bon.

La méthode de Runge-Kutta-Fehlberg (dénoté $RKF45$) a un procédé pour déterminer si la taille du « pas » appropriée h est employée. À chaque étape, deux approximations différentes pour la solution sont faites et comparées, chaque étape exige l'utilisation des six valeurs suivantes :

$$\begin{cases} k_1 &= hf\left(x_n, y_n\right) \\ k_2 &= hf\left(x_n + \frac{1}{4}h, y_n + \frac{1}{4}k_1\right) \\ k_3 &= hf\left(x_n + \frac{3}{8}h, y_n + \frac{3}{32}k_1 + \frac{9}{32}k_2\right) \\ k_4 &= hf\left(x_n + \frac{12}{13}h, y_n + \frac{1932}{2197}k_1 - \frac{7200}{2197}k_2 + \frac{7296}{2197}k_3\right) \\ k_5 &= hf\left(x_n + h, y_n + \frac{439}{216}k_1 - 8k_2 + \frac{3680}{513}k_3 - \frac{845}{4104}k_4\right) \\ k_6 &= hf\left(x_n + \frac{1}{2}h, y_n - \frac{8}{27}k_1 + 2k_2 - \frac{3544}{2565}k_3 + \frac{1859}{4104}k_4 - \frac{11}{40}k_5\right) \end{cases} \tag{a}$$

Alors une approximation de la solution du système d'équation différentielle est faite pour suivre une méthode de RK_4 :

$$y_{n+1} = y_n + \frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4101}k_4 - \frac{1}{5}k_5 \tag{b}$$

La meilleure valeur de la solution est déterminée pour suivre une méthode de Runge-Kutta d'ordre 5 :

$$Z_{n+1} = y_n + \frac{16}{135}k_1 + \frac{6656}{12,825}k_3 + \frac{28,561}{56,430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6 \tag{c}$$

La taille du pas optimale sh peut être déterminée en multipliant les temps scalaires de s avec la taille du pas courant h. Le s scalaire est :

$$s = \left(\frac{tol\ h}{2\left|Z_{n+1} - y_{n+1}\right|}\right)^{\frac{1}{4}} \approx 0.84\left(\frac{tol\ h}{\left|Z_{n+1} - y_{n+1}\right|}\right)^{\frac{1}{4}} \tag{d}$$

Exemple 2.4.1 *résolution d'une équation avec la méthode $RKF45$ et $RK4$*

$$y^{'} = 1 + y^2 \qquad avec \qquad y(0) = 0 \qquad [0, 1.4]$$

avec la solution exacte :

$$y\left(t\right) = \tan\left(t\right)$$

Tableau $RKF45$

k	t_k	$RKF45\,(y_k)$	*La solution exacte* $y(t_k)=\tan(t_k)$	*Erreur* $\lvert y(t_k)-y_k\rvert$
0	0.0	0	0	0
1	0.2	0.2027100	0.2027100	0
2	0.4	0.4227933	0.4227931	0.0000002
3	0.6	0.6841376	0.6841368	0.0000008
4	0.8	1.0296434	1.0296386	0.0000048
5	1.0	1.5574398	1.5774077	0.0000321
6	1.1	1.9648085	1.9647597	0.0000488
7	1.2	2.5722408	2.5721516	0.0000892
8	1.3	3.6023295	3.6021024	0.0002271
9	1.35	4.4555714	4.4552218	0.0003496
10	1.4	5.7985045	5.7978837	0.0006208

Il est important d'apprendre qu'une taille fixe du « pas » n'est pas la meilleure stratégie quoiqu'elle donne une table gentille apparaissante des valeurs. Si les valeurs nécessaires ne sont pas dans la table, l'interpolation polynôme devrait être employée.

Tableau : RK_4

k	t_k	$RK4\,(y_k)$	*La solution exacte* $y(t_k)=\tan(t_k)$	*Erreur* $\lvert y(t_k)-y_k\rvert$
0	0.0	0	0	0
1	0.1	0.1003346	0.1003347	0.0000001
2	0.2	0.2027099	0.2027100	0.0000001
3	0.3	0.3093360	0.3093362	0.0000002
4	0.4	0.4227930	0.4227932	0.0000002
5	0.5	0.5463023	0.5463025	0.0000002
6	0.6	0.6841368	0.6841368	0.0000000
7	0.7	0.8422886	0.8422884	0.0000002
8	0.8	1.0296391	1.0296386	0.0000005
9	0.9	1.2601588	1.2601582	0.0000006
10	1.0	1.5574064	1.5574077	0.0000013
11	1.1	1.9647466	1.9647597	0.0000131
12	1.2	2.5720718	2.5721516	0.0000798
13	1.3	3.6015634	3.6021024	0.0005390
14	1.4	5.7919748	5.7978837	0.0059089

Comparaison entre RKF45 et RK4 pour la résolution d'une équation différentielle

On utilise $RKF45$ avec le tol de valeur $= 2 \times 10^{-5}$. Il a changé la taille du pas automatiquement et a produit les 10 approximations à la solution dans le tableau $RKF45$.

Et on utilise RK_4 avec la taille du pas a priori de $h = 0.1$, ce qui a exigé à l'ordinateur de produire 14 approximations aux points équidistants dans le tableau $RK4$. Les approximations au bon point final sont :

$$y\,(1.4) \approx y_{10} = 5.7985045 \text{ et } y\,(1.4) \approx y_{14} = 5.7919748$$

Et l'erreur est de :

$$E_{10} = -0.0006208 \text{ et } E_{14} = 0.0059089$$

Remarque 2.4.1 *Pour les méthodes $RKF45$ et $RK4$, respectivement La méthode $RKF45$ a une erreur très petite.*

2.5 Conclusion

Les intégrateurs implicites sont souvent moins précis, plus compliqués à utiliser. Mais ils sont plus stables que les intégrateurs explicites: ils divergent moins rapidement. Pour les méthodes $RKF45$ et RK_4 (respectivement). La méthode $RKF45$ a une erreur très petite Donc on peut dire que :

•Runge-Kutta d'ordre 4 :

-moins de défauts

-Précision est généralement encore meilleure

-Bon rapport qualité/prix

-Mais pas une réponse universelle !

•Euler :

-Beaucoup de défauts

-Déconseillée par presque tout

-Mais rapide à implémenter

•Heun :

-Fait la moyenne entre 2 pentes, les méthodes de Runge-Kutta font la moyenne de m pentes...

La précision dépend de l'ordre

-Euler :ordre 1

-Heun :ordre 2

-Runge-Kutta :ordre 4

-Cependant l'utilisation d'ordre supérieur n'est pas très intéressant...

Chapitre 3

Méthode de résolution des EDOs à pas multiples

3.1 Introduction

Un intégrateur "intelligent" possède une procédure de contrôle de la méthode de convergence, c'est à dire un moyen d'estimer l'erreur commise par le calcul sur un pas d´intégration et la possibilité de choisir en conséquence un nouveau pas si le système différentiel aborde une région ou la fonction prend des valeurs plus importantes. Le calcul de cette estimation entraîne un surcoût de calcul, qu'il convient de bien gérer afin de minimiser cet effort supplémentaire.

L'idée la plus simple pour estimer cette erreur consiste à calculer la solution donnée par un algorithme (de Runge-Kutta d'ordre 4 par exemple) pour deux pas d´intégration différents, h et $2h$. Soit $y(x+2h)$ la solution exacte à $x+2h$ et $y(x+h)$ la solution exacte à $x+h$, on a

$\cdot y(x+2h) = y_1 + (2h)^5 \phi + o(h^6)$

$\cdot y(x+h) = y_2 + 2(h^5)\phi + o(h^6)$

Où ϕ est une fonction qui reste constante sur l'intervalle x, $x+2h$ à l'ordre h^5.

La première équation correspond à une intégration avec un pas égal à $2h$ tandis que la seconde correspond à deux intégrations successives avec un pas de h. La différence $\Delta = y_2 - y_1$ fournit une estimation de l'erreur commise avec un pas d'intégration h.

3.2 Méthode Adams–Bashforth–Moulton

Méthodes basées sur des techniques d'intégration numérique, dont la formulation générale est :

$$y_{n+1} = y_n + h \sum_{j=-1}^{p} \beta_j f\left(x_{n-j}, y_{n-j}\right)$$

3.2.1 Méthode d'Adams-Bashforth à 2 pas explicite d'ordre 2

$$\begin{cases} y_0 & \text{donné} \\ y_1 & \text{calculé avec une méthode à un pas} \\ y_{n+1} & = y_n + \frac{h}{2}\left(3f\left(x_n, y_n\right) - f\left(x_{n-1}, y_{n-1}\right)\right) \end{cases}$$

3.2.2 Méthode d'Adams-Bashforth à 3 pas explicite d'ordre 3

$$\begin{cases} y_0 & \text{donné} \\ y_1, y_2 & \text{calculés avec une méthode à un pas} \\ y_{n+1} & = y_n + \frac{h}{12}\left(23f\left(x_n, y_n\right) - 16f\left(x_{n-1}, y_{n-1}\right) + 5f\left(x_{n-2}, y_{n-2}\right)\right) \end{cases}$$

3.2.3 Méthode d'Adams-Bashforth à 4 pas explicite d'ordre 4

$$\begin{cases} y_0 & \text{donné} \\ y_1, y_2, y_3 & \text{calculés avec une méthode à un pas} \\ y_{n+1} & = y_n + \frac{h}{24}\left(55f\left(x_n, y_n\right) - 59f\left(x_{n-1}, y_{n-1}\right) + 37f\left(x_{n-2}, y_{n-2}\right) - 9f\left(x_{n-3}, y_{n-3}\right)\right) \end{cases}$$

3.2.4 Méthode d'Adams-Moulton à 1 pas implicite d'ordre 2

$$\begin{cases} y_0 & \text{donné} \\ y_{n+1} & = y_n + \frac{h}{2}\left(f\left(x_n, y_n\right) + f\left(x_{n+1}, y_{n+1}\right)\right) \end{cases}$$

3.2.5 Méthode d'Adams-Moulton à 2 pas implicite d'ordre 3

$$\begin{cases} y_0 & \text{donné} \\ y_1 & \text{calculé avec une méthode à un pas} \\ y_{n+1} & = y_n + \frac{h}{12}\left(5f\left(x_{n+1}, y_{n+1}\right) + 8f\left(x_n, y_n\right) - f\left(x_{n-1}, y_{n-1}\right)\right) \end{cases}$$

3.2.6 Méthode d'Adams-Moulton à 3 pas implicite d'ordre 4

$$\begin{cases} y_0 & \text{donné} \\ y_1, y_2 & \text{calculés avec une méthode à un pas} \\ y_{n+1} & = y_n + \frac{h}{24}\left(9f\left(x_{n+1}, y_{n+1}\right) + 19f\left(x_n, y_n\right) - 5f\left(x_{n-1}, y_{n-1}\right) + f\left(x_{n-2}, y_{n-2}\right)\right) \end{cases}$$

Pour rendre les méthodes d'Adams-Moulton explicite, on remplace le y_{n+1} "génant" par son estimation par la méthode d'Adams-Bashforth. Sont construits ainsi des schémas explicites dits prédicteur-correcteur, par exemple un **schéma d'ordre** 2 **est :**

$$\begin{cases} y_0 & \text{donné} \\ y_1 & \text{calculé avec une méthode à un pas} \\ y^*_{n+1} & = y_n + \frac{h}{2}\left(3f\left(x_n, y_n\right) - f\left(x_{n-1}, y_{n-1}\right)\right) \\ y_{n+1} & = y_n + \frac{h}{2}\left(f\left(x_n, y_n\right) + f\left(x_{n+1}, y^*_{n+1}\right)\right) \end{cases}$$

Le schéma prédicteur-correcteur d'ordre 4 **est :**

$$\begin{cases} y_0 & \text{donné} \\ y_1, y_2, y_3 & \text{calculés avec une méthode à un pas} \\ y^*_{n+1} & = y_n + \frac{h}{24}\left(55f\left(x_n, y_n\right) - 59f\left(x_{n-1}, y_{n-1}\right) + 37f\left(x_{n-2}, y_{n-2}\right) - 9f\left(x_{n-3}, y_{n-3}\right)\right) \\ y_{n+1} & = y_n + \frac{h}{24}\left(9f\left(x_{n+1}, y^*_{n+1}\right) + 19f\left(x_n, y_n\right) - 5f\left(x_{n-1}, y_{n-1}\right) + f\left(x_{n-2}, y_{n-2}\right)\right) \end{cases}$$

Exemple 3.2.1 *On reprend une équation différentielle :*

$$y'\left(t\right) = -y\left(t\right) + t + 1 \qquad \textit{avec} \qquad y\left(0\right) = 1$$

avec la solution exacte

$$y\left(t\right) = \exp\left(-t\right) + t$$

On fait appel cette fois aux méthodes de prédiction-corréction d'ordre 2 *et* 4*. Les premiers pas de temps sont calculés par une méthode de* RK_4 *qui sont déjà servis à résoudre cette équation différentielle. La méthode de prédiction-correction d'ordre* 2 *exige de connaitre* y_0*, qui vaut* 1 *et* y_1 *qui a été calculé au préalable de la méthode de* RK_4 *et qui vaut* 1.0048375*. La principe itération donne d' abord une équation :*

$$\begin{aligned} y_2^* &= y_1 + \tfrac{h}{2}\left(3f\left(t_1, y_1\right) - f\left(t_0, y_0\right)\right) \\ &= 1.0048375 + \tfrac{0.1}{2}\left(3f\left(0.1, 1.0048375\right) - f\left(0, 1\right)\right) \\ &= 1.019111875 \end{aligned}$$

et ensuite une correction :

$$\begin{aligned} y_2 &= y_1 + \tfrac{h}{2}\left(f\left(t_2, y_2^*\right) + f\left(t_1, y_1\right)\right) \\ &= 1.0048375 + \tfrac{0.1}{2}\left(f\left(0.2, 1.019111875\right) + f\left(0.1, 1.0048375\right)\right) \\ &= 1.018640031 \end{aligned}$$

Les autres itérations sont résumées dans le tableau suivant :

t	y_n^*	y_n	e_n (*Erreur*)
0.0		1.000000000	0
0.1		1.004837500	$0.81964040 \times 10^{-7}$
0.2	1.019111875	1.018640031	$0.90721828 \times 10^{-4}$
0.3	1.041085901	1.040653734	$0.16448607 \times 10^{-3}$
0.4	1.070487675	1.070096664	$0.22338196 \times 10^{-3}$
0.5	1.106614851	1.106261088	$0.26957140 \times 10^{-3}$
0.6	1.148826758	1.148506695	$0.30494011 \times 10^{-3}$
0.7	1.196543746	1.196254173	$0.33112990 \times 10^{-3}$
0.8	1.249241382	1.248979396	$0.36149325 \times 10^{-3}$
0.9	1.306445195	1.306208166	$0.36797857 \times 10^{-3}$
1.0	1.367725911	1.367511462	$0.36797857 \times 10^{-3}$

L'erreur à $t = 0.1$ *et beaucoup plus faible qu'aux autres valeurs de* t *puisque la solution numérique à cet endroit à été calculée à l'aide d'une méthode d'ordre* 4*. Cela explique également l'absence de prédicateur pour cette valeur. De manière similaire, la méthode de prédiction-correction d'ordre* 4 *requiert le calcul de* y_1*,* y_2 *et* y_3 *à l'aide de la méthode de* RK_4*. Par la suite, il suffit d'utiliser le schéma de prédiction-correction d'ordre* 4*, pour* $n \succeq 3$*. La première itération s'effectue commme suit :*

$$\begin{aligned} y_4^* &= y_3 + \tfrac{0.1}{24}\left(55f\left(t_3, y_3\right) - 59f\left(t_2, y_2\right) + 37f\left(t_1, y_1\right) - 9f\left(t_0, y_0\right)\right) \\ &= 1.0408184 + \tfrac{0.1}{24}\begin{pmatrix} 55f\left(0.3, 1.04081884\right) - 59f\left(0.2, 1.0187309\right) \\ +37f\left(0.1, 1.0048375\right) - 9f\left(0, 1\right) \end{pmatrix} \\ &= 1,1070323 \end{aligned}$$

$$\begin{aligned} y_4 &= y_3 + \tfrac{0.1}{24}\left(9f\left(t_4, y_4^*\right) + 19f\left(t_3, y_3\right) - 5f\left(t_2, y_2\right) + f\left(t_1, y_1\right)\right) \\ &= 1.0408184 + \tfrac{0.1}{24}\begin{pmatrix} 9f\left(0.4, 1.1070323\right) - 19f\left(0.3, 1.04008184\right) \\ -5f\left(0.2, 1.0187309\right) + f\left(0.1, 1.0048375\right) \end{pmatrix} \\ &= 1,0703199 \end{aligned}$$

On obtient enfin les resultats suivants :

t	y_n^*	y_n	$e_n\,(Erreur)$
0.0		1.000000000	0
0.1		1.0048375	0.819640×10^{-7}
0.2		1.0187309	0.148328×10^{-6}
0.3		1.0408184	0.201319×10^{-6}
0.4	1.1070323	1.0703199	0.127791×10^{-6}
0.5	1.1065332	1.1065303	0.391302×10^{-6}
0.6	1.1488136	1.1488110	0.603539×10^{-6}
0.7	1.1965869	1.1965845	0.772415×10^{-6}
0.8	1.2493302	1.2493281	0.903669×10^{-6}
0.9	1.3065706	1.3065687	0.100294×10^{-5}
1.0	1.3678801	1.3678784	0.107514×10^{-5}

l'erreur est ici beaucoup plus petite qu'avec la méthode d'ordre 2.

3.3 Méthode de Hamming

Méthode de Hamming avec des formules de modification

$$\begin{aligned}
\text{prédicteur} \quad &: \quad p_{k+1} = y_{k-3} + \frac{4h}{3}\left(2f_{k-2} - f_{k-1} + 2f_k\right) \\
\text{modificateur} \quad &: \quad m_{k+1} = p_{k+1} + \frac{112}{121}\left(c_k - p_k\right) \\
\text{correcteur} \quad &: \quad c_{k+1} = \frac{1}{8}\left\{9y_k - y_{k-2} + 3h\left(-f_{k-1} + 2f_k + f\left(x_{k+1}, m_{k+1}\right)\right)\right\} \\
y_{k+1} \quad &= \quad c_{k+1} - \frac{9}{121}\left(c_{k+1} - p_{k+1}\right)
\end{aligned}$$

avec y_0 donné et y_1 y_2 et y_3 calculés avec une méthode à un pas.

3.4 Méthode de Milne Simpson

$$\begin{aligned}
\text{prédicteur} \quad &: \quad p_{k+1} = y_{k-3} + \frac{4h}{3}\left(2f_{k-2} - f_{k-1} + 2f_k\right) \\
\text{correcteur} \quad &: \quad y_{k+1} = y_{k-1} + \frac{h}{3}\left(f_{k-1} + 4f_1 + f_{k+1}\right)
\end{aligned}$$

avec y_0 donné et y_1, y_2 et y_3 calculés avec une méthode à un pas.

3.5 Méthodes de Gear

Les méthodes de Gear ne sont pas construites à partir de techniques d'intégration numérique mais directement à partir de polynôme d'interpolation passant par p points

$$(x_{i+1}, y_{i+1}), (x_i, y_i), ..., (x_{i-p+1}, y_{i+1})$$

3.5.1 Méthode de Gear à 2 pas implicite d'ordre 2

$$\begin{cases}
y_0 & \text{donné} \\
y_1 & \text{calculé avec une méthode à un pas} \\
y_{n+1} & = \frac{4}{3}y_n - \frac{1}{3}y_{n-1} + \frac{2h}{3}f\left(x_{n+1}, y_{n+1}\right)
\end{cases}$$

3.5.2 Méthode de Gear à 3 pas implicite d'ordre 3

$$\begin{cases} y_0 & \text{donné} \\ y_1, y_2 & \text{calculés avec une méthode à un pas} \\ y_{n+1} & = \frac{18}{11} y_n - \frac{9}{11} y_{n-1} + \frac{2}{11} y_{n-2} + \frac{2h}{11} f\left(x_{n+1}, y_{n+1}\right) \end{cases}$$

3.5.3 Méthode de Gear à 4 pas implicite d'ordre 4

$$\begin{cases} y_0 & \text{donné} \\ y_1, y_2, y_3 & \text{calculés avec une méthode à un pas} \\ y_{n+1} & = \frac{48}{25} y_n - \frac{36}{25} y_{n-1} + \frac{16}{25} y_{n-2} + \frac{3}{25} y_{n-3} + \frac{12h}{25} f\left(x_{n+1}, y_{n+1}\right) \end{cases}$$

Chapitre 4

Méthode de résolution des EDOs à pas multiples avec conditions aux limites

4.1 problème de la valeur aux limites

Un problème de la valeur marginale (BVP) est une équation de $N^{iéme}$-ordre avec certaines des valeurs de la variable dépendante $x(t)$ et leurs dérivées spécifiques au temps initial t_0 et d'autres spécifiques au temps final t_f.

$$[BVP]_N : x^{(N)}(t) = f\left(t, x(t), x^{'}(t), x^{(2)}(t), \ldots, x^{(N-1)}(t)\right)$$

Avec les valeurs limites

$$x(t_1) = x_{10}, x(t_2) = x_{21}, \ldots, x^{(N-1)}(t_N) = x_{N,N-1}$$

Dans certains cas, quelques relations entre les valeurs initiales et les valeurs finales peuvent être données comme état de frontière mélangé au lieu de valeurs initiales/finales spécifiques. Cette section couvre la méthode de tir et la méthode de différence finie cela peut être employée pour résoudre un BVP de second ordre :

$$[BVP]_2 : x^{"}(t) = f(t, x(t)) \qquad \text{avec} \qquad x(t_0) = x_0 \; x(t_f) = x_f$$

4.2 Méthode de Tir

Dans cette section, nous faisons l'étude des équations différentielles linéaires d'ordre 2 avec conditions aux limites de la forme :

$$\begin{cases} y^{"}(x) &= a_2(x)\, y^{'}(x) + a_1(x)\, y(x) + a_0(x) \\ & y(a) = y_a \qquad \text{et} \qquad y(b) = y_b \end{cases} \tag{4.2.1}$$

On suppose les fonctions $a_i(x)$ suffisamment régulières pour assurer l'existence et l'unicité des équations différentielles que nous rencontrerons. La différence entre les équations différentielles avec conditions initiales et celles avec conditions aux limites est illustrée à la figure (4.2.1). Dans le premier cas, à $t = t_0$, la fonction $y(t_0)$ ainsi que sa pente $y^{'}(t_0)$ sont connues. Dans le cas des équations avec conditions aux limites, on ne connait que les valeurs de la fonction $y(x)$ aux deux extrémités de l'intervalles soit $y(a)$ et $y(b)$. On remarque qu'il n'y a aucune condition initiale liée à la dérivée de la fonction $y(x)$ en $x = a$. La condition initiale sur la dérivée est remplacée par une condition sur la fonction $y(x)$ à l'autre extrémité de l'intervalle $[a, b]$. Il suffit en effet de remarquer que l'on peut remplacer l'équation différentielle (4.2.1) par deux équations différentielles avec conditions initiales.

Théorème 4.2.1 *[1] La solution de l'équation différentielle avec conditions aux limites (4.2.1) est donnée par :*

$$y(x) = y_1(x) + \left(\frac{y_b - y_1(b)}{y_2(b)}\right) y_2(x) \tag{4.2.2}$$

où $y_1(x)$ et $y_2(x)$ sont les conditions des équations différentielles avec conditions initiales suivantes :

$$\begin{cases} y_1^{"}(x) &= a_2(x)\, y_1^{'}(x) + a_1(x)\, y_1(x) + a_0(x) \\ & y_1(a) = y_a \qquad et \qquad y_1^{'}(a) = 0 \end{cases} \tag{4.2.3}$$

et

$$\begin{cases} y_2^{"}(x) &= a_2(x)\, y_2^{'}(x) + a_1(x)\, y_2(x) \\ & y_2(a) = 0 \qquad et \qquad y_2^{'}(a) = 1 \end{cases} \tag{4.2.4}$$

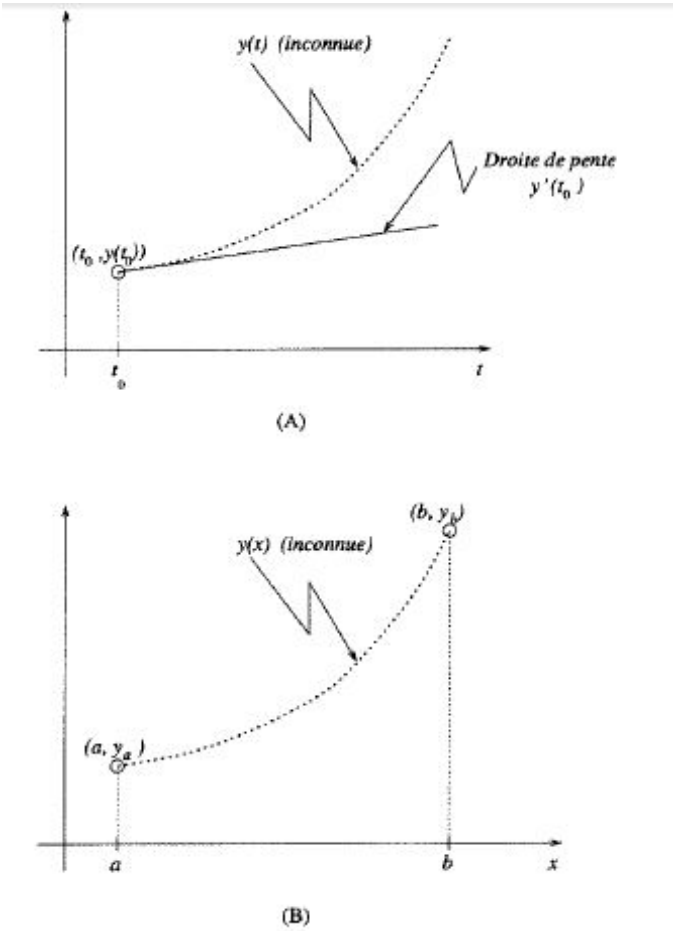


Figure 4.2.1 : Shéma représente les conditions initiales (A) et les conditions aux limites (B)

Preuve. voir [1]. ■

Remarque 4.2.1 *On note l'absence du terme* $a_0(x)$ *de l'équation différentielle relative à* $y_2(x)$.

La figure (4.2) illustre les fonctions $y_1(x)$ et $y_2(x)$ ainsi que la solution recherchée. Si on regarde les conditions initiales relatives à $y_1(x)$ et $y_2(x)$, on constate entre autres choses que ni $y_1(x)$ ni $y_2(x)$ ne verifie la bonne condition aux limites en $x = b$. par contre, en utilisant l'équation (4.2.2) on obtient la bonne solution.

On peut résoudre les équations différentielles avec conditions initiales (4.2.3) et (4.2.4) à l'aide de la méthode de RK_4. On doit d'abord transformer chacune d'elles en un système de 2 équations différentielles d'ordre 1. En posant :

$$\begin{aligned} u_1(x) &= y_1(x) \qquad \text{et} \qquad v_1(x) = y_2(x) \\ u_2(x) &= y_1'(x) \qquad \text{et} \qquad v_2(x) = y_2'(x) \end{aligned}$$

On obtient les deux systèmes suivants :

$$\begin{cases} u_1'(x) = v_2(x) & (u_1(a) = y_a) \\ u_2'(x) = a_2(x)\, u_2(x) + a_1(x)\, u_1(x) + a_0(x) & (u_2(a) = 0) \\ v_1'(x) = v_2(x) & (v_1(a) = 0) \\ v_2'(x) = a_2(x)\, v_2(x) + a_1(x)\, v_1(x) & (v_2(a) = 1) \end{cases} \tag{4.2.5}$$

la solution finale peut alors s'ecrire :

$$\begin{cases} y(x) &= y_1(x) + \left(\dfrac{y_b - y_1(b)}{y_2(b)} \right) y_2(x) \\ &= u_1(x) + \left(\dfrac{y_b - u_1(b)}{v_1(b)} \right) v_1(x) \end{cases}$$

selon les nouvelles variables $u_1(x)$ et $v_1(x)$.

Exemple 4.2.1 *Soit l'équation différentielle suivante :*

$$\begin{cases} y''(x) = -\frac{2}{x} y'(x) + \frac{1}{x^2} \\ y(1) = 0 \ \text{et} \ y(2) = 0.693147 \end{cases}$$

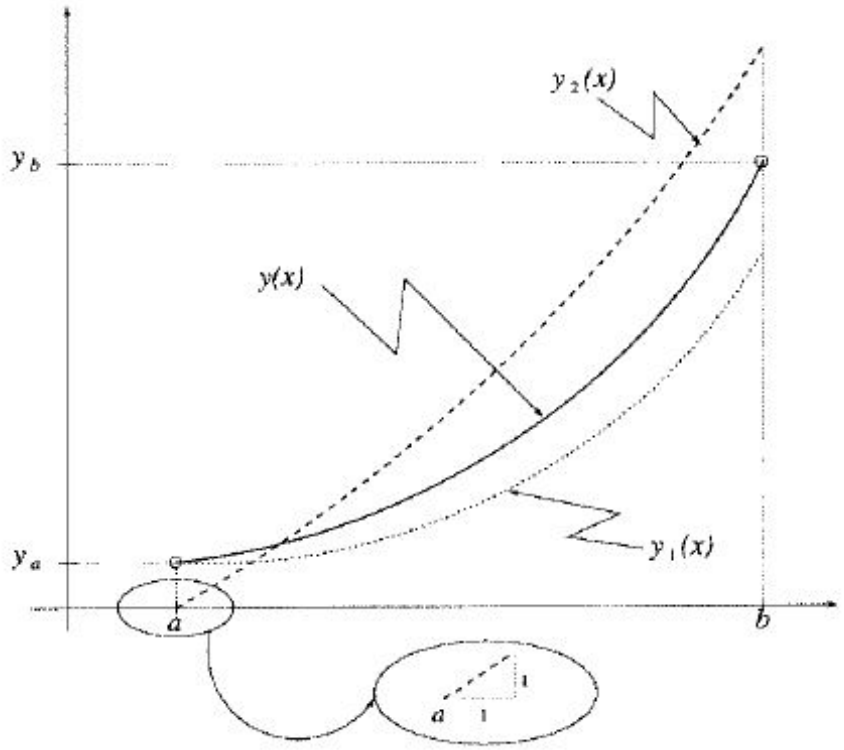
avec la solution exacte

$$y(x) = \ln(x)$$

On a dans ce cas :

$$a_2(x) = -\frac{2}{x}, \ a_1(x) = 0 \qquad et \qquad a_0(x) = \frac{1}{x^2}$$

Le tableau qui suit présente la solution de cette équation différentielle.



x	$y_1(x) = u_1(x)$	$y_2(x) = v_1(x)$	$y(x)$	*Erreur*
1.0	0	0	0	0
1.1	0.00440251	0.09090702	0.09530975	0.4299×10^{-6}
1.2	0.01565698	0.16666359	0.18232096	0.5997×10^{-6}
1.3	0.03159742	0.23076568	0.26236363	0.6325×10^{-6}
1.4	0.05076044	0.28571054	0.33647164	0.5924×10^{-6}
1.5	0.07213427	0.33332955	0.40546459	0.5143×10^{-6}
1.6	0.09500609	0.37499626	0.47000321	0.4173×10^{-6}
1.7	0.11886594	0.41176106	0.53062794	0.3123×10^{-6}
1.8	0.14334454	0.44444091	0.58778646	0.2057×10^{-6}
1.9	0.16817191	0.47368079	0.64185379	0.1008×10^{-6}
2.0	0.19314933	0.49999670	0.69314718	0.4400×10^{-9}

On a employé la méthode de RK_4 pour calculer $y_1(x)$ et $y_2(x)$. On note que :

$$\begin{cases} y_1(b) = y_1(2) = 0.19314933 \\ y_2(b) = y_2(2) = 0.49999670 \end{cases}$$

ce qui permet le calcul de $y(x)$ à l'aide de l'équation(4.2.2).

4.3 Méthode des différences finies

L'idée de cette méthode est de diviser l'intervalle entier $[t_0, t_f]$ en segments de N de largeur $h = (t_f - t_0)/N$, et de rapprocher les premieres et deuxièmes dérivées dans l'équations pour chaque point de grille par les formules de différence centrale. Ceci mène à un système de tri diagonal des équations en ce qui concerne $(n-1)$ variables

$$[x_i = x(t_0) + ih = 1 \ldots\ldots\ldots\ldots..N-1].$$

Cependant, afin que ce système d'équations soit résolu facilement, il devrait être linéaire, en impliquant que ses coefficients peuvent ne pas contenir aucune limite de x. Par exemple, on considère un BVP comprenant l'équation linéaire de second ordre

$$x''(t) + a_1(t)\, x'(t) + a_0(t)\, x(t) = u(t) \text{ avec } x(t_0) = x_0, x(t_f) = x_N$$

Selon la méthode de différence finie, nous divisons l'intervalle de solution $[t_0\ t_f]$ en segments de N et convertir l'équation pour chaque point de grille $t_i = t_0 + ih$, dans une équation à différences

$$\begin{cases} \dfrac{x_{i+1} - 2x_i + x_{i-1}}{h^2} + a_{1i}\dfrac{x_{i+1} - x_{i-1}}{2h} + a_{0i}x_i & = & u_i \\ (2 - ha_{1i})\, x_{i-1} + (-4 + 2h^2 a_{0i})\, x_i + (2 + ha_{1i})\, x_{i+1} & = & 2h^2 u_i \end{cases}$$

Puis, tenant compte de l'état de frontière cela $x_0 = x(t_0)$ et

$x_N = x(t_f)$, nous rassemblons le tout les $(N-1)$ équations pour construire un système tri diagonal avec des équations linéaires

$$\begin{bmatrix} -4+2h^2a_{01} & 2+ha_{11} & 0 & \cdots & 0 & 0 \\ 2-ha_{12} & -4+2h^2a_{02} & 2+ha_{12} & \cdots & 0 & 0 \\ 0 & 2-ha_{13} & -4+2h^2a_{03} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2+ha_{1,N-3} & \\ 0 & 0 & 0 & \cdots & -4+2h^2a_{0,N-2} & 2+ha_{1,N-2} \\ 0 & 0 & 0 & \cdots & 2-ha_{1,N-1} & -4+2h^2a_{0,N-1} \end{bmatrix}$$

$$\times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{N-3} \\ x_{N-2} \\ x_{N-1} \end{bmatrix} = \begin{bmatrix} 2h^2u_1 - (2-ha_{11})\,x_0 \\ 2h^2u_2 \\ 2h^2u_3 \\ \vdots \\ 2h^2u_{N-3} \\ 2h^2u_{N-2} \\ 2h^2u_{N-1} - (2-ha_{1,N-1})\,x_N \end{bmatrix}$$

Exemple 4.3.1 *résolution d'une équation avec la méthode des différences finies*

$$-x^{"}(t) + tx(t) = (t+1) + \sin(t) \qquad avec \qquad x(0) = 0 \ et \ x(2\pi) = 0$$

avec la solution exacte

$$x(t) = \sin(t)$$

on trouve :

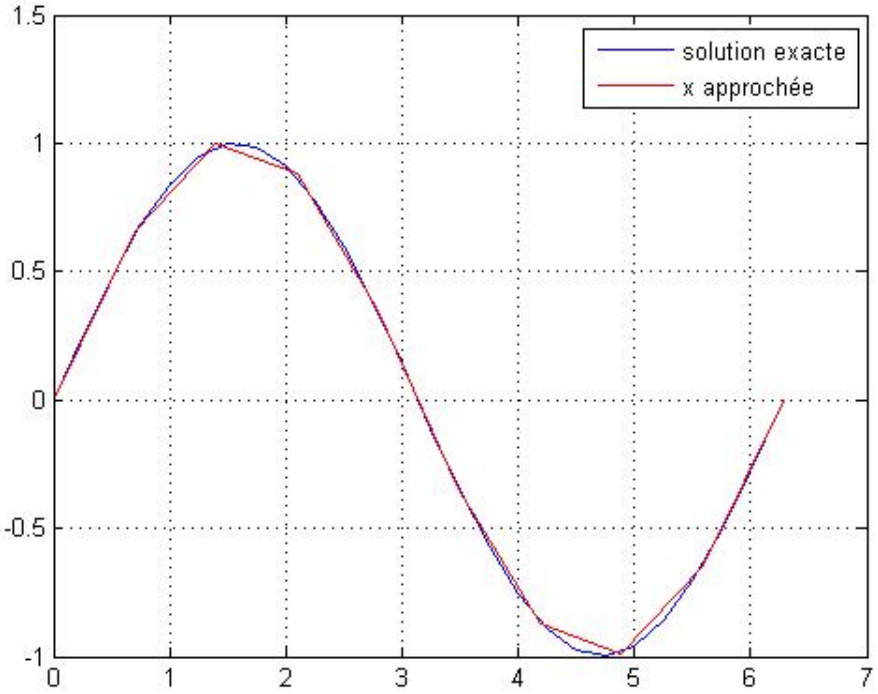


Figure 4.3.1 : Solution d'un BVP obtenir en employant la méthode de différence finie

Remarque 4.3.1 *On note à propos de la méthode de tir et de la méthode de différence finie :*

1- Tandisque la méthode de tir s'applique à BVP linéaire/non linéaire, la méthode de différence finie convient à BVP linéaire. Cependant, nous pouvons également appliquer la méthode de différence finie d'une façon itérative pour résoudre BVP non linéaire.

2- Les deux méthodes peuvent être modifiées pour résoudre BVP avec des états de frontière mélangés

Conclusion

Les méthodes numériques de la résolution des EDOs sont appliqées dans plusieurs domaines, dont les principales méthodes sont divisées en deux grands types :les méthodes à un pas et les méthodes à pas multiples.

La résolution numérique d'équation différentielle est trés souvent nécessaire, faute de l'éxistence de solutions analytiques.

Et finalement, nous constatons que la meilleure méthode et celle qui a la solution approchée proche de la solution exacte avec un minimun d'erreur.

Bibliographie

[1] **A.Fortin,** Analyse numérique pour ingénieurs.

[2] **C.Dossal**, Approximation de solutions d'équations différentielles, schémas numériques, 2013.

[3] **E.Goncalvés**, Résolution numérique, Discretisation des EDP et EDO, 2005.

[4] **E.Montseny**, Introduction aux méthodes de résolution numérique des equations différentielles.

[5] **F.Boyer**, Equations différentielles ordinaires, Equations aux dérivées partielles, analyse théorique et numérique, Université de provence et paul cézanne, 2010.

[6] **F.Gouaisbaut,** Résolution des équations différentielles ordinaires, univérsité de toulouse, ups, 2009

[7] **F.Wlazinki,** Equations différentielles.

[8] **H.Djelouah**, Résolution d'équation différentielle avec Matlab, Algérie, 2009.

[9] **J.Lonis merrien,** Analyse Numérique avec Matlab, DUNOD,paris,2007.

[10] **J.Pierre Croisille**, Rappels sur les équations différentielles ordinaires, 2008.

[11] **René**, Equations différentielles ordinaires et aux dérivées partielles, 2008.

[12] **R.Herbin,** Analyse numérique des équations aux dérivées partielles, université aix marseille1, 2012.

[13] **V.Guinot** & **B.Cappelare**, Méthodes Numériques Appliquées,2005- 2006.

ANNEXE

Exemple pour l'équation y'=1-y(t) avec y(0)=0

```
**********Programme de la fonction**********
function [dydt]=rhs(t,y)
dydt=1-y;
**********************************************************
**********Programme de la Méthode d'Euler **********
% Code pour la solution de l'equation differential :
%
% dy/dt = f(t,y)
%
% Le coté droit i.e. f(t,y) doit être dans la fonction rhs.m
%
clear
y(1)=input(' entrer la valeur du point initial de y: ');
tmax = input(' entrer la valeur maximale de t: ');
tmin = input(' entrer la valeur manimale de t: ');
h = input(' entrer la valeur du pas h: ');
nstep = (tmax-tmin)/h;
t = tmin:h:tmax;
%
% rk2 loop
fprintf('\n n t(n) y(n) f(t,y) y(n+1)\n\n'...
);
```

```
for n = 1:nstep
rhs(t(n),y(n));
y(n+1)=y(n)+h*rhs(t(n),y(n));
fprintf('%2i %4.2f %4.2f %6.4f %6.4f\n',...
n,t(n),y(n),rhs(t(n),y(n)),y(n+1));
end
% output
yexact=1-exp(-t);
plot(t,y,'b',t,yexact,'r');
grid
legend(' y: Euler', 'y: exact');
maxerror=max(abs(y-yexact))
```

********** La Méthode d'Euler améliorée **********

```
% Code pour la solution de l'equation differential :
%
% dy/dt = f(t,y)
%
% Le coté droit i.e. f(t,y) doit être dans la fonction rhs.m
%
clear
y(1)=input(' entrer la valeur du point initial de y: ');
tmax = input(' entrer la valeur maximale de t: ');
tmin = input(' entrer la valeur manimale de t: ');
h = input(' entrer la valeur du pas h: ');
nstep = (tmax-tmin)/h;
t = tmin:h:tmax;
%
% rk2 loop
fprintf('\n n t(n) y(n) f(t,y) ynew(n+1) y(n+1)\n\n'...
);
```

```
for n = 1:nstep
rhs(t(n),y(n));
ynew(n+1)=y(n)+h/2*(rhs(t(n),y(n)));
y(n+1)=y(n)+h/2*(rhs(t(n),y(n))+rhs(t(n+1),ynew(n+1)));
fprintf('%2i %4.2f %4.2f %6.4f %6.4f\n',...
n,t(n),y(n),rhs(t(n),y(n)),ynew(n+1),y(n+1));
end
% output
yexact=1-exp(-t);
plot(t,y,'b',t,yexact,'r');
grid
legend(' y: Euler amelioré', 'y: exact');
maxerror=max(abs(y-yexact))
****************** La Méthode d'Euler Cauchy *****************
% Code pour la solution de l'equation differential :
%
% dy/dt = f(t,y)
%
% Le coté droit i.e. f(t,y) doit être dans la fonction rhs.m
%
clear
y(1)=input(' entrer la valeur du point initial de y: ');
tmax = input(' entrer la valeur maximale de t: ');
tmin = input(' entrer la valeur manimale de t: ');
h = input(' entrer la valeur du pas h: ');
nstep = (tmax-tmin)/h;
t = tmin:h:tmax;
%
% rk2 loop
fprintf('\n n t(n) y(n) f(t,y) ynew(n+1) y(n+1)\n\n'...
```

```
);
for n = 1:nstep
rhs(t(n),y(n));
ynew(n+1)=y(n)+h/2*(rhs(t(n),y(n)));
y(n+1)=y(n)+h*(rhs(t(n)+h/2,ynew(n+1)));
fprintf('%2i %4.2f %4.2f %6.4f %6.4f\n',...
n,t(n),y(n),rhs(t(n),y(n)),ynew(n+1),y(n+1));
end
% output
yexact=1-exp(-t);
plot(t,y,'b',t,yexact,'r');
grid
legend(' y: Euler cauchy', 'y: exact');
maxerror=max(abs(y-yexact))
**************************************************************
**** La Méthde d'Euler pour le pas h=0.5 et h=0.25 ****
% Code pour la solution de l'equation differential :
%
% dy/dt = f(t,y)
%
% Le coté droit i.e. f(t,y) doit être dans la fonction rhs.m
%
clear
y(1)=input(' entrer la valeur du point initial de y: ');
tmax = input(' entrer la valeur maximale de t: ');
tmin = input(' entrer la valeur manimale de t: ');
h = input(' entrer la valeur du pas h: ');
nstep = (tmax-tmin)/h;
t = tmin:h:tmax;
%
```

```
% rk2 loop
fprintf('\n n t(n) y(n) f(t,y) y(n+1)\n\n'...
);
for n = 1:nstep
rhs(t(n),y(n));
y(n+1)=y(n)+h*rhs(t(n),y(n));
fprintf('%2i %4.2f %4.2f %6.4f %6.4f\n',...
n,t(n),y(n),rhs(t(n),y(n)),y(n+1));
end
% output
yexact=1-exp(-t);
plot(t,y,'b',t,yexact,'r');
hold on
grid
legend(' y: Euler', 'y: exact');
maxerror=max(abs(y-yexact))
```

****

Comparaison entre Euler, Euler cauchy et Euler amélioré

```
% Code pour la solution de l'equation differential :
%
% dy/dt = f(t,y)
%
% Le coté droit i.e. f(t,y) doit être dans la fonction rhs.m
%
clear
y(1)=input(' entrer la valeur du point initial de y: ');
tmax = input(' entrer la valeur maximale de t: ');
tmin = input(' entrer la valeur manimale de t: ');
h = input(' entrer la valeur du pas h: ');
nstep = (tmax-tmin)/h;
```

```
t = tmin:h:tmax;
%
% rk2 loop
fprintf('\n n t(n) y(n) f(t,y) y(n+1)\n\n'...
);
for n = 1:nstep
rhs(t(n),y(n));
y(n+1)=y(n)+h*rhs(t(n),y(n));
fprintf('%2i %4.2f %4.2f %6.4f %6.4f\n',...
n,t(n),y(n),rhs(t(n),y(n)),y(n+1));
end
for n = 1:nstep
rhs(t(n),y(n));
ynew(n+1)=y(n)+h/2*(rhs(t(n),y(n)));
z(n+1)=y(n)+h/2*(rhs(t(n),y(n))+rhs(t(n+1),ynew(n+1)));
fprintf('%2i %4.2f %4.2f %6.4f %6.4f\n',...
n,t(n),y(n),rhs(t(n),y(n)),ynew(n+1),y(n+1));
end
for n = 1:nstep
rhs(t(n),y(n));
ynew(n+1)=y(n)+h/2*(rhs(t(n),y(n)));
r(n+1)=y(n)+h*(rhs(t(n)+h/2,ynew(n+1)));
fprintf('%2i %4.2f %4.2f %6.4f %6.4f\n',...
n,t(n),y(n),rhs(t(n),y(n)),ynew(n+1),y(n+1));
end
% output
yexact=1-exp(-t);
plot(t,y,'b+',t,yexact,'r',t,z,'g',t,r,'m');
grid
legend(' y: Euler', 'y: exact','z:euler amélioré','r:euler cauchy');
```

```
a=max(abs(y-yexact))
b=max(abs(z-yexact))
c=max(abs(r-yexact))
**********************************************************************La
Méthode de Renge Kutta4 **********
% Code pour la solution de l'equation differential :
%
% dy/dt = f(t,y)
%
% Le coté droit i.e. f(t,y) doit être dans la fonction rhs.m
%
clear
y(1)=input(' entrer la valeur du point initial de y: ');
tmax = input(' entrer la valeur maximale de t: ');
tmin = input(' entrer la valeur manimale de t: ');
h = input(' entrer la valeur du pas h: ');
nstep = (tmax-tmin)/h;
t = tmin:h:tmax;
%
fprintf('\n n t(n) y(n) f(t,y) k1 k2 k3 k4 y(n+1)\n\n');
for n = 1:nstep
rhs(t(n),y(n));
k1=rhs(t(n),y(n));
k2=rhs(t(n)+h/2,y(n)+h*k1/2);
k3=rhs(t(n)+h/2,y(n)+h*k2/2);
k4=rhs(t(n)+h,y(n)+h*k3);
y(n+1)=y(n)+(h/6)*(k1+2*k2+2*k3+k4);
fprintf('%2i %4.2f %4.2f %6.4f %6.4f %6.4f %6.4f %6.4f %6.4f\n',...
n,t(n),y(n),rhs(t(n),y(n)),k1,k2,k3,k4,y(n+1));
end
```

```
% output
yexact=1-exp(-t);
plot(t,y,'b',t,yexact,'r');
grid
legend(' y: rk4', 'y: exact');
maxerror=max(abs(y-yexact))
```

**********La Méthode de Heun**********

```
% Code pour la solution de l'equation differential :
%
% dy/dt = f(t,y)
%
% Le coté droit i.e. f(t,y) doit être dans la fonction rhs.m
%
clear
y(1)=input(' entrer la valeur du point initial de y: ');
tmax = input(' entrer la valeur maximale de t: ');
tmin = input(' entrer la valeur manimale de t: ');
h = input(' entrer la valeur du pas h: ');
nstep = (tmax-tmin)/h;
t = tmin:h:tmax;
%
% rk2 loop
% output
fprintf('\n n t(n) y(n) f(t,y) y(n+1)\n\n'...
);
for n = 1:nstep
k1=rhs(t(n),y(n));
t(n+1)=t(n)+h;
k2=rhs(t(n+1),y(n)+(h*k1));
```

```
y(n+1)=y(n)+h*(k1+k2)/2;
fprintf('%2i %4.2f %4.2f %6.4f %6.4f\n',...
n,t(n),y(n),rhs(t(n),y(n)),y(n+1));
end
yexact=1-exp(-t);
plot(t,y,'b',t,yexact,'r');
grid
legend(' y: heun', 'y: exact');
maxerror=max(abs(y-yexact))
```

********** Comparaison entre tout les Méthodes **********

```
% Code pour la solution de l'equation differential :
%
% dy/dt = f(t,y)
%
% Le coté droit i.e. f(t,y) doit être dans la fonction rhs.m
%
clear
y(1)=input(' entrer la valeur du point initial de y: ');
tmax = input(' entrer la valeur maximale de t: ');
tmin = input(' entrer la valeur manimale de t: ');
h = input(' entrer la valeur du pas h: ');
nstep = (tmax-tmin)/h;
t = tmin:h:tmax;
%
% rk2 loop
fprintf('\n n t(n) y(n) f(t,y) y(n+1)\n\n'...
);
for n = 1:nstep
rhs(t(n),y(n));
```

```
y(n+1)=y(n)+h*rhs(t(n),y(n));
fprintf('%2i %4.2f %4.2f %6.4f %6.4f\n',...
n,t(n),y(n),rhs(t(n),y(n)),y(n+1));
end
for n = 1:nstep
rhs(t(n),y(n));
ynew(n+1)=y(n)+h/2*(rhs(t(n),y(n)));
z(n+1)=y(n)+h/2*(rhs(t(n),y(n))+rhs(t(n+1),ynew(n+1)));
fprintf('%2i %4.2f %4.2f %6.4f %6.4f\n',...
n,t(n),y(n),rhs(t(n),y(n)),ynew(n+1),y(n+1));
end
for n = 1:nstep
rhs(t(n),y(n));
ynew(n+1)=y(n)+h/2*(rhs(t(n),y(n)));
r(n+1)=y(n)+h*(rhs(t(n)+h/2,ynew(n+1)));
fprintf('%2i %4.2f %4.2f %6.4f %6.4f\n',...
n,t(n),y(n),rhs(t(n),y(n)),ynew(n+1),y(n+1));
end
for n = 1:nstep
k1=rhs(t(n),y(n));
t(n+1)=t(n)+h;
k2=rhs(t(n+1),y(n)+(h*k1));
f(n+1)=y(n)+h*(k1+k2)/2;
fprintf('%2i %4.2f %4.2f %6.4f %6.4f\n',...
n,t(n),y(n),rhs(t(n),y(n)),f(n+1));
end
for n = 1:nstep
rhs(t(n),y(n));
k1=rhs(t(n),y(n));
k2=rhs(t(n)+h/2,y(n)+h*k1/2);
```

```
k3=rhs(t(n)+h/2,y(n)+h*k2/2);
k4=rhs(t(n)+h,y(n)+h*k3);
g(n+1)=y(n)+(h/6)*(k1+2*k2+2*k3+k4);
fprintf('%2i %4.2f %4.2f %6.4f %6.4f %6.4f %6.4f %6.4f %6.4f\n',...
n,t(n),y(n),rhs(t(n),y(n)),k1,k2,k3,k4,y(n+1));
end
% output
yexact=1-exp(-t);
plot(t,y,'b',t,yexact,'r',t,z,'g',t,r,'o',t,f,'m',t,g,'+');
grid
legend(' y: Euler', 'y: exact','z:euler amélioré','r:euler cauchy','f:heun','g:rk4');
a=max(abs(y-yexact))
b=max(abs(z-yexact))
c=max(abs(r-yexact))
d=max(abs(f-yexact))
e=max(abs(g-yexact))
```

********** La Méthode de ABM d'ordre 4 **********

```
% Code pour la solution de l'equation differential :
%
% dy/dt = f(t,y)
%
% Le coté droit i.e. f(t,y) doit être dans la fonction rhs.m
%
clear
y(1)=input(' entrer la valeur du point initial de y: ');
tmax = input(' entrer la valeur maximale de t: ');
tmin = input(' entrer la valeur manimale de t: ');
h = input(' entrer la valeur du pas h: ');
nstep = (tmax-tmin)/h;
```

```
t = tmin:h:tmax;
%
% rk2 loop
fprintf('\n n t(n) y(n) f(t,y) y(n+1)\n\n'...
);
for n = 1:3
rhs(t(n),y(n));
k1=rhs(t(n),y(n));
k2=rhs(t(n)+h/2,y(n)+h*k1/2);
k3=rhs(t(n)+h/2,y(n)+h*k2/2);
k4=rhs(t(n)+h,y(n)+h*k3);
y(n+1)=y(n)+(h/6)*(k1+2*k2+2*k3+k4);
fprintf('%2i %4.2f %4.2f %6.4f %6.4f %6.4f %6.4f %6.4f %6.4f\n',...
n,t(n),y(n),rhs(t(n),y(n)),k1,k2,k3,k4,y(n+1));
end
for n = 4:nstep
rhs(t(n),y(n));
v(n+1)=y(n)+h/24*(55*rhs(t(n),y(n))-59*rhs(t(n-1),y(n-1)) +37*rhs(t(n-2),y(n-2))-9*rhs(t(n-
3),y(n-3)));
y(n+1)=y(n)+h/24*(9*rhs(t(n+1),v(n+1))+19*rhs(t(n),y(n))-5*rhs(t(n-1),y(n-1))+rhs(t(n-
2),y(n-2)));
fprintf('%2i %4.2f %4.2f %6.4f %6.4f %6.4f %6.4f %6.4f %6.4f\n',...
n,t(n),y(n),rhs(t(n),y(n)),k1,k2,k3,k4,y(n+1));
end
% output
yexact=t+exp(-t);
plot(t,y,'b',t,yexact,'r');
grid
a=max(abs(y-yexact))
************************************************************
```

```
********** La Méthode de différence finis **********
clc
%résolution de Pb des différences finies
%-u"(x)+c(x)u=f(x) de l'intervalle ]0,2pi[
%c(x)=x, f(x)=(x+1)*sin(x)
% la sol exact u = sin(x)
c=inline('x');
f=inline('(x+1).*sin(x)');
a=0;
b=2*pi;
alpha=0;
beta=0;
n=input('donnez n = ');
h=(b-a)/n;
x=a:h:b;
l=2/(h^2)+c(x(2:n));
v=ones(n-2,1);
A=diag(l)-(1/(h^2))*diag(v,1)-(1/(h^2))*diag(v,-1)
F=[f(x(2))+(1/(h^2))*alpha,f(x(3:n1)),f(x(n))+(1/(h^2))*beta]';
u=inv(A)*F;
uapp=[alpha;u;beta]
solext=inline('sin(x)');
uexact=[solext(x)]'
err=abs(uapp-uexact)
t=0:1/200:2*pi;
plot(t,solext(t),x,uapp,'r',x,err)
************************************************************
```