

Chapitre IV. Résolution numérique des équations différentielles ordinaires

IV.1 Introduction

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

Dans ce chapitre nous allons étudier des méthodes approximatives notamment : la *méthode d'Euler* et les *méthodes de Range-Kutta*, qui sont utilisées pour calculer des solutions approchées des équations différentielles de premier ordre.

Equations différentielles d'ordre 1

La forme générale des équations différentielles d'ordre 1 est :

$$y'(t) = f(t, y(t)) \quad \text{avec} \quad y(t_0) = y_0 \quad (\text{IV-1})$$

où t est une variable indépendante. $y(t)$ est une variable dépendante de t . $f(t, y(t))$ est une fonction de deux variables suffisamment différentiable. $y'(t)$ est la dérivée première de $y(t)$.

$y(t_0) = y_0$ est une condition initiale donnée.

La résolution numérique de (IV-1) consiste à obtenir approximativement $y(t)$ pour $t \geq t_0$ en utilisant différentes méthodes numériques.

Si des dérivées de $y(t)$ d'ordre 2 apparaissent dans une l'équation différentielle, on aurait une équation d'ordre 2, et ainsi de suite.

Remarques importantes

- Avec les outils numériques de résolution d'équations différentielles, il n'est plus possible d'obtenir une solution pour toutes les valeurs de la variable indépendante t . On obtient plutôt une approximation de la solution analytique seulement à certaines valeurs de t notée t_i et distancées d'une valeur $h_i = t_{i+1} - t_i$.
- Dans les méthodes présentées dans ce chapitre, la distance h_i est constante pour tout i et est notée h , elle est appelée le pas de t .
- On note $y(t_i)$ la solution analytique de l'équation différentielle (IV-1) en $t = t_i$.
- On note y_i la solution approximative en $t = t_i$ obtenue à l'aide d'une méthode numérique.

IV.2 Méthode d'Euler

Soit un intervalle $[a, b]$ des valeurs de t . On considère $(n + 1)$ valeurs équidistantes dans $[a, b]$ notés $t_0, t_1, t_2, \dots, t_i, t_{i+1}, \dots, t_n$ (ou n sous intervalles de même longueur h), tel que :

$$t_0 = a, \quad t_n = b, \quad t_{i+1} = t_i + h, \quad h = \frac{b - a}{n}, \quad i = 0, 1, 2, \dots, n - 1$$

Pour $t = t_0$ on a :

$$y'(t_0) = f(t_0, y(t_0)) = f(t_0, y_0) \quad (\text{IV-2})$$

qui est la pente de la tangente à la courbe de $y(t)$ au point (t_0, y_0) (voir Figure IV.1). L'équation de cette tangente est donc :

$$d_0(t) = y_0 + f(t_0, y_0)(t - t_0) \quad (\text{IV-3})$$

Pour $t = t_1$ on a :

$$d_0(t_1) = y_0 + f(t_0, y_0)(t_1 - t_0) = y_0 + hf(t_0, y_0) \quad (\text{IV-4})$$

Dans la méthode d'Euler on considère que la solution approximative en $t = t_1$ est :

$$y_1 = d_0(t_1) = y_0 + hf(t_0, y_0) \quad (\text{IV-5})$$

Chapitre IV. Résolution numérique des équations différentielles ordinaires

Sur la Figure IV.1, on remarque bien qu'il y a une différence entre la solution approximative y_1 et la solution exacte $y(t_1)$ pour l'abscisse t_1 .

La pente de la tangente à la courbe de $y(t)$ au point $(t_1, y(t_1))$ (voir Figure IV.1) est :

$$y'(t_1) = f(t_1, y(t_1)) \quad (\text{IV-6})$$

Puisque la valeur exacte de $y(t_1)$ est inconnue, on la remplace approximativement par la valeur approchée y_1 , soit alors :

$$y'(t_1) = f(t_1, y(t_1)) \simeq f(t_1, y_1) \quad (\text{IV-7})$$

Puis on définit une droite passant par le point (t_1, y_1) (voir la Figure IV.1) par :

$$d_1(t) = y_1 + f(t_1, y_1)(t - t_1) \quad (\text{IV-8})$$

qui permettra d'estimer $y(t_2)$ par le même raisonnement précédent. Il vient que :

$$y(t_2) \simeq y_2 = d_1(t_2) = y_1 + hf(t_1, y_1) \quad (\text{IV-9})$$

On constate que l'erreur commise à la première itération est réintroduite dans le calcul de la deuxième itération.

Si on applique la même procédure pour estimer respectivement les valeurs $y(t_3), y(t_4), \dots, y(t_n)$ on obtient la formule itérative de la méthode d'Euler suivante :

$$y_{i+1} = y_i + hf(t_i, y_i) \quad (\text{IV-10})$$

avec : $t_0 = a, t_n = b, t_{i+1} = t_i + h, h = \frac{b-a}{n}, i = 0, 1, 2, \dots, n-1$

Cette formule permet donc d'obtenir le point (t_{i+1}, y_{i+1}) à partir du point (t_i, y_i) en commençant par un point initial donné (t_0, y_0) .

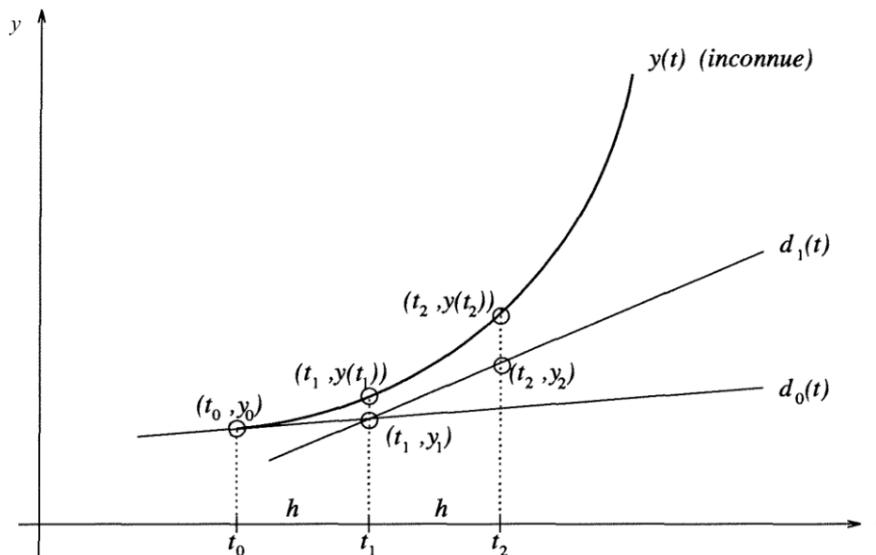


Figure IV.1 : Illustration graphique de la méthode approximative d'Euler

D'un autre côté, on remarque que les erreurs se propagent d'une itération à l'autre. Il en résulte de façon générale que l'erreur à chaque itération i est $|y(t_i) - y_i|$, qui augmente légèrement avec i .

IV.2.1 Algorithme de la méthode d'Euler

Les étapes de calcul par la méthode d'Euler suivent l'algorithme suivant :

1. Donner la fonction $f(t, y)$, un intervalle $[a, b]$, une condition initiale (t_0, y_0) , un nombre de sous intervalles n .
2. Calculer le pas h .

Chapitre IV. Résolution numérique des équations différentielles ordinaires

3. Calculer les valeurs (t_{i+1}, y_{i+1}) à partir des valeurs (t_i, y_i) en utilisant la formule de la méthode d'Euler.

4. Afficher les résultats et arrêter.

IV.2.2 L'erreur de troncature locale

L'erreur de troncature locale au point $t = t_i$ est définie par :

$$\tau_{i+1}(h) = \frac{y(t_{i+1}) - y(t_i)}{h} - f(t_i, y(t_i)) \quad (\text{IV-11})$$

L'erreur de troncature locale mesure la précision avec laquelle la solution analytique vérifie la formule de la méthode d'Euler (IV-10).

Il est très important de noter que l'on utilise la solution exacte $y(t_i)$ (et non y_i) dans la définition de l'erreur de troncature locale. Cela s'explique par le fait que l'on cherche à mesurer l'erreur introduite par la formule (IV-10), en supposant que la méthode était exacte jusque-là.

Le développement de Taylor de la fonction $y(t)$ autour du point $t = t_i$ à l'ordre 2 est :

$$y(t_{i+1}) = y(t_i + h) = y(t_i) + y'(t_i)h + \frac{y''(t_i)h^2}{2} + O(h^3) \quad (\text{IV-12})$$

ou encore :

$$y(t_{i+1}) = y(t_i) + f(t_i, y(t_i))h + \frac{y''(t_i)h^2}{2} + O(h^3) \quad (\text{IV-13})$$

Donc l'erreur de troncature locale devient :

$$\tau_{i+1}(h) = \frac{y(t_i + h) - y(t_i)}{h} - f(t_i, y(t_i)) \approx \frac{y''(t_i)h}{2} \quad (\text{IV-14})$$

IV.2.3 Exercice IV.1

On veut résoudre numériquement l'équation différentielle suivante :

$$y'(t) - \frac{3}{t}y(t) = t, \quad y_0 = y(1) = 3, \quad t \in [1, 2] \quad (\text{IV-15})$$

1) Calculer la solution approchée par la méthode d'Euler pour $n = 4$ sous-intervalles.

2) La solution analytique de l'équation différentielle (IV-15) est :

$$y(t) = -t^2 + 4t^3 \quad (\text{IV-16})$$

Calculer $y(1.25), y(1.5), y(1.75), y(2)$ puis comparer entre les solutions approchée et exacte.

IV.2.4 Corrigé d'exercice IV.1

1) D'abord écrivons l'équation (IV-15) sous la forme : $y'(t) = f(t, y(t)) = t + \frac{3}{t}y(t)$

Puis on utilise la formule d'Euler (IV-10). On a :

$h = (2 - 1)/4 = 0.25, t_0 = a = 1, t_4 = b = 2$. Pour $n = 4$ on fait uniquement 4 itérations :

Itération 1 :

$$y_1 = y_0 + h \times f(t_0, y_0) = 3 + 0.25 \times f(1, 3) = 3 + 0.25 \times \left(1 + \frac{3}{1} \times 3\right) = 5.5$$

$$t_1 = t_0 + h = 1 + 0.25 = 1.25$$

Itération 2 :

$$y_2 = y_1 + h \times f(t_1, y_1) = 5.5 + 0.25 \times f(1.25, 5.5) = 5.5 + 0.25 \times \left(1.25 + \frac{3}{1.25} \times 5.5\right) = 9.1125$$

$$t_2 = t_1 + h = 1.25 + 0.25 = 1.5$$

Itération 3 :

$$y_3 = y_2 + h \times f(t_2, y_2) = 9.1125 + 0.25 \times f(1.5, 9.1125) = 14.0438$$

$$t_3 = t_2 + h = 1.5 + 0.25 = 1.75$$

Itération 4 :

$$y_4 = y_3 + h \times f(t_3, y_3) = 20.5000$$

$$t_4 = t_3 + h = 1.75 + 0.25 = 2$$

2) Comparaison entre solutions approchée et exacte :

Itération i	t_i	$(y_i)_{Euler}$	$(y_i)_{exacte}$
1	1.25	5.5	6.2500
2	1.5	9.1125	11.2500
3	1.75	14.0438	18.3750
4	2	20.5000	28

Tableau IV-1 : Comparaison entre solutions approchée et exacte (corrigé d'exercice IV.1)

On constate que l'erreur commise à la méthode de trapèze est assez considérable pour $n = 4$, et elle se propage d'une itération à l'autre.

IV.2.5 Travail pratique IV.1 : Implémentation MATLAB de la méthode d'Euler

1) En appliquant l'algorithme de la méthode d'Euler, écrire un programme Matlab qui permet de calculer la solution approchée de l'équation (IV-15) par la méthode d'Euler pour $n = 20$ sous-intervalles, et la solution exacte (IV-16) en même temps.

Note : Représenter les solutions approchées et exactes sur le même graphe afin de comparer.

2) Exécuter ce programme pour des nombres de sous-intervalles $n = 50, 100$ et 200 . Conclure.

IV.2.6 Corrigé de travail pratique IV.1

1) Le programme MATLAB de la méthode d'Euler :

```
clc;clear all;close all;
a=1;
b=2;
n=20
h=(b-a)./n;
t=a:h:b;
y=zeros(size(t));
y(1)=3;
f=@(t,y)t+(3./t)*y;
for i=1:n
t(i)=t(1)+(i-1)*h;
k1=h*f(t(i),y(i));
y(i+1)=y(i)+k1;
end
ys=-(t.^2)+4.*(t.^3);
plot(t,y,'*r',t,ys)
xlabel('t')
ylabel('y')
title('courbes des solutions approchée (pour n=20) et exacte.')
legend('approximée','exacte')
```

Après exécution on obtient :

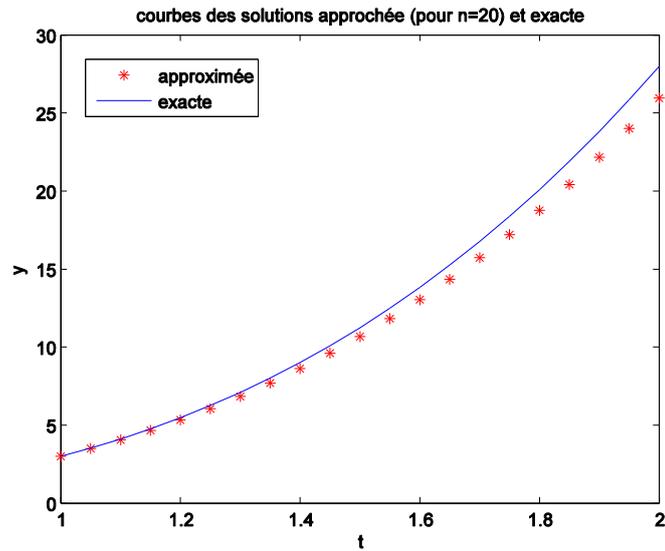


Figure IV.2 : Comparaison entre solution exacte et approchée obtenue par méthode d'Euler pour $n = 20$
(Corrigé de TP IV.1)

3) En augmentant le nombre de sous-intervalles n on obtient :

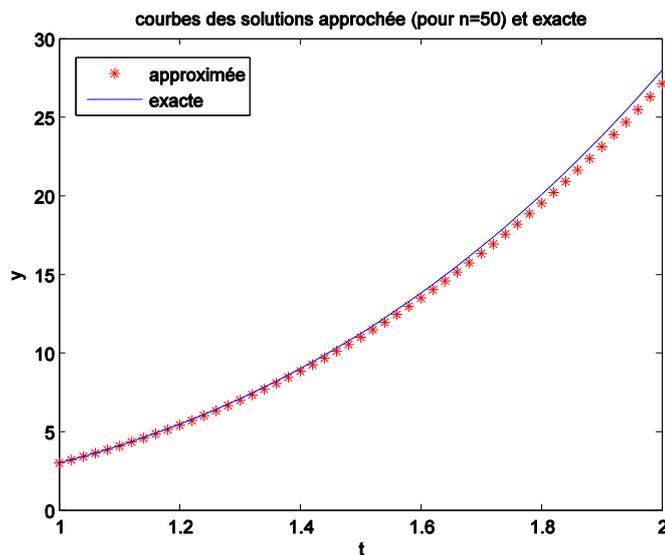


Figure IV.3 : Comparaison entre solution exacte et approchée obtenue par méthode d'Euler pour $n = 50$
(Corrigé de TP IV.1)

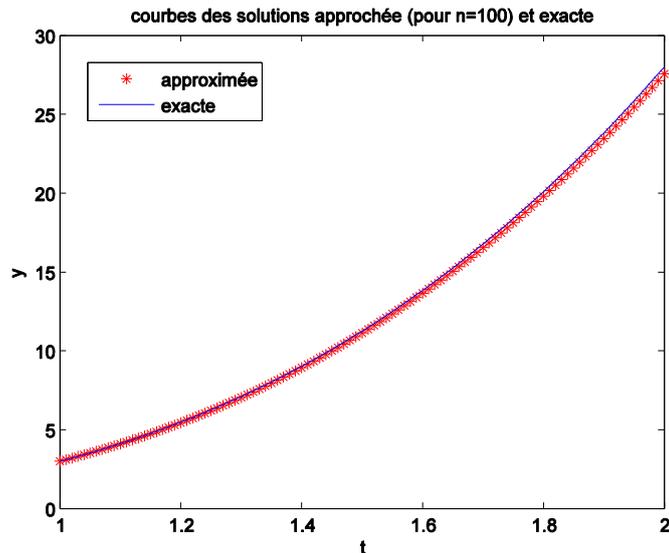


Figure IV.4 : Comparaison entre solution exacte et approchée obtenue par méthode d'Euler pour $n=100$ (Corrigé de TP IV.1)

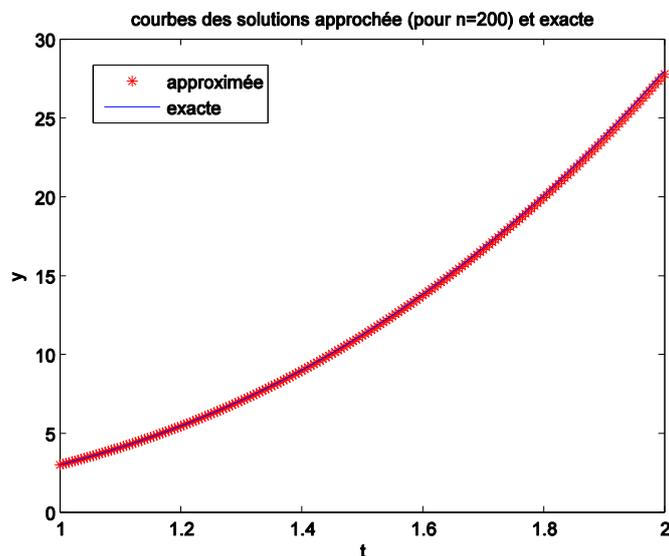


Figure IV.5 : Comparaison entre solution exacte et approchée obtenue par méthode d'Euler pour $n=200$ (Corrigé de TP IV.1)

On conclut que lorsque le nombre de sous-intervalles augmente, la solution approchée par la méthode d'Euler se rapproche plus en plus à la solution exacte.

IV.3 Méthode de Taylor d'ordre 2

Le développement de Taylor autorise une généralisation immédiate de la méthode d'Euler, qui permet d'obtenir des algorithmes dont l'erreur de troncature locale est d'ordre plus élevé. Nous nous limitons cependant à la méthode de Taylor du second ordre.

On cherche, au temps $t = t_i$, une approximation de la solution en $t = t_{i+1}$. On a immédiatement :

$$y(t_{i+1}) = y(t_i + h) = y(t_i) + y'(t_i)h + \frac{y''(t_i)h^2}{2} + O(h^3) \quad (\text{IV-17})$$

En se servant de l'équation différentielle (IV-1), on trouve :

$$y(t_{i+1}) = y(t_i) + f(t_i, y(t_i))h + \frac{f'(t_i, y(t_i))h^2}{2} + O(h^3) \quad (\text{IV-18})$$

Chapitre IV. Résolution numérique des équations différentielles ordinaires

La règle de dérivation en chaîne assure que :

$$f'(t, y(t)) = \frac{\partial f(t, y(t))}{\partial t} + \frac{\partial f(t, y(t))}{\partial y} y'(t) \quad (\text{IV-19})$$

c'est-à-dire :

$$f'(t, y(t)) = \frac{\partial f(t, y(t))}{\partial t} + \frac{\partial f(t, y(t))}{\partial y} f(t, y(t)) \quad (\text{IV-20})$$

on obtient donc :

$$y(t_{i+1}) = y(t_i) + hf(t_i, y(t_i)) + \frac{h^2}{2} \left(\frac{\partial f(t_i, y(t_i))}{\partial t} + \frac{\partial f(t_i, y(t_i))}{\partial y} f(t_i, y(t_i)) \right) + O(h^3) \quad (\text{IV-21})$$

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

$$y(t_{i+1}) \approx y(t_i) + hf(t_i, y(t_i)) + \frac{h^2}{2} \left(\frac{\partial f(t_i, y(t_i))}{\partial t} + \frac{\partial f(t_i, y(t_i))}{\partial y} f(t_i, y(t_i)) \right) \quad (\text{IV-22})$$

On remplace la solution analytique $y(t_i)$ par son approximation y_i on trouve la formule de la méthode de Taylor d'ordre 2 suivante :

$$y_{i+1} = y_i + hf(t_i, y_i) + \frac{h^2}{2} \left(\frac{\partial f(t_i, y_i)}{\partial t} + \frac{\partial f(t_i, y_i)}{\partial y} f(t_i, y_i) \right) \quad (\text{IV-23})$$

Avec :

$$t_{i+1} = t_i + h$$

On en conclut que les erreurs se propagent d'une itération à une autre.

IV.3.1 Algorithme de la méthode de Taylor d'ordre 2

Les étapes de calcul par la méthode de Taylor d'ordre 2 suivent l'algorithme suivant :

1. Donner la fonction $f(t, y)$, un intervalle $[a, b]$, une condition initiale (t_0, y_0) , un nombre de sous intervalles n .
2. Calculer le pas h .
3. Calculer les valeurs (t_{i+1}, y_{i+1}) à partir des valeurs (t_i, y_i) en utilisant la formule de la méthode de Taylor d'ordre 2.
4. Afficher les résultats et arrêter.

IV.4 Méthodes de Runge-Kutta

Il est possible d'obtenir des méthodes de Taylor encore plus précises en poursuivant le développement de Taylor (IV-14) jusqu'à des termes d'ordre élevé. On doit alors évaluer les dérivées de la fonction $f(t, y(t))$ d'ordre de plus en plus élevé, ce qui nécessite le calcul supplémentaire de :

$$\frac{\partial^2 f}{\partial t^2}, \frac{\partial^2 f}{\partial y^2}, \frac{\partial^2 f}{\partial t \partial y}, \dots, \frac{\partial^{i+j} f}{\partial t^i \partial y^j} \quad (\text{IV-24})$$

Pour cette raison, les méthodes obtenues sont difficiles à utilisées. Il existe cependant un moyen de contourner cette difficulté en développant les méthodes de Runge-Kutta.

Il serait avantageux de disposer de méthodes d'ordre de plus en plus élevé tout en évitant les désavantages des méthodes de Taylor, qui nécessitent l'évaluation des dérivées partielles de la fonction $f(t, y(t))$. Une voie est tracée par les méthodes de Runge-Kutta, qui sont calculées sur les méthodes de Taylor du même ordre.

IV.4.1 Méthodes de Runge-Kutta d'ordre 2

On a vu que le développement de la méthode de Taylor passe par la relation (IV-21). Le but est de remplacer cette dernière relation par une expression équivalente possédant le même ordre de précision ($O(h^3)$). On propose la forme :

$$y(t_{i+1}) = y(t_i) + a_1 hf(t_i, y(t_i)) + a_2 hf(t_i + a_3 h, y(t_i) + a_4 h) \quad (IV-25)$$

où on doit déterminer les paramètres a_1, a_2, a_3 et a_4 de telle sorte que les expressions (IV-21) et (IV-25) aient toutes deux une erreur en $O(h^3)$. On ne trouve par ailleurs aucune dérivée partielle dans cette expression. Pour y arriver, on doit recourir au développement de Taylor en deux variables autour du point $(t_i, y(t_i))$. On a ainsi :

$$\begin{aligned} f(t_i + a_3 h, y(t_i) + a_4 h) \\ = f(t_i, y(t_i)) + a_3 h \frac{\partial f(t_i, y(t_i))}{\partial t} + a_4 h \frac{\partial f(t_i, y(t_i))}{\partial y} + O(h^2) \end{aligned} \quad (IV-26)$$

La relation (IV-25) devient alors :

$$\begin{aligned} y(t_{i+1}) = y(t_i) + (a_1 + a_2)hf(t_i, y(t_i)) + a_2 a_3 h^2 \frac{\partial f(t_i, y(t_i))}{\partial t} \\ + a_2 a_4 h^2 \frac{\partial f(t_i, y(t_i))}{\partial y} + O(h^3) \end{aligned} \quad (IV-27)$$

On voit immédiatement que les expressions (II-17) et (IV-27) sont du même ordre. Pour déterminer les coefficients a_i , il suffit de comparer ces deux expressions terme à terme :

- Coefficients respectifs de $f(t_i, y(t_i))$: $h = (a_1 + a_2)h$
- Coefficients respectifs de $\frac{\partial f(t_i, y(t_i))}{\partial t}$: $\frac{h^2}{2} = a_2 a_3 h^2$
- Coefficients respectifs de $\frac{\partial f(t_i, y(t_i))}{\partial y}$: $\frac{h^2}{2} f(t_i, y(t_i)) = a_2 a_4 h^2$

On obtient ainsi un système non linéaire de 3 équations comprenant 4 inconnues :

$$\begin{cases} 1 & = & (a_1 + a_2) \\ \frac{1}{2} & = & a_2 a_3 \\ \frac{f(t_i, y(t_i))}{2} & = & a_2 a_4 \end{cases} \quad (IV-28)$$

Ce système est sous déterminé en ce sens qu'il y a moins d'équations que d'inconnues et qu'il n'a donc pas de solution unique. Cela offre une marge de manœuvre qui favorise la mise au point de plusieurs variantes de la méthode de Range-Kutta. Voici le plus couramment utilisé.

IV.4.2 Méthode d'Euler modifiée

Les différents paramètres sont donnés par :

$$a_1 = a_2 = \frac{1}{2}, \quad a_3 = 1 \quad \text{et} \quad a_4 = f(t_i, y(t_i)) \quad (IV-29)$$

On établit sans peine que ces coefficients satisfont aux trois équations du système non linéaire. Il suffit ensuite de remplacer ces valeurs dans l'équation (IV-27). Pour ce faire, on doit négliger le terme $O(h^3)$ et remplacer la valeur exacte $y(t_i)$ par approximation y_i . On obtient alors la formule suivante :

$$y_{i+1} = y_i + \frac{h}{2} \times (f(t_i, y_i) + f(t_{i+1}, \hat{y})) \quad \text{avec} \quad \hat{y} = y_i + h \times f(t_i, y_i) \quad \text{et} \quad t_{i+1} = t_i + h \quad (IV-30)$$

IV.4.3 Algorithme de la méthode d'Euler modifiée

Les étapes de calcul par la méthode d'Euler modifiée suivent l'algorithme suivant :

Chapitre IV. Résolution numérique des équations différentielles ordinaires

1. Donner la fonction $f(t, y)$, un intervalle $[a, b]$, une condition initiale (t_0, y_0) , un nombre de sous intervalles n .
2. Calculer le pas h .
3. Calculer les valeurs (t_{i+1}, y_{i+1}) à partir des valeurs (t_i, y_i) en utilisant la formule de la méthode d'Euler modifiée.
4. Afficher les résultats et arrêter.

Une autre méthode de Runge-Kutta d'ordre 2 qui est très utilisée est la méthode du point milieu, qui correspond au choix suivant des coefficients a_i .

IV.4.4 Méthode du point milieu

Les différents paramètres sont donnés par :

$$a_1 = 0, \quad a_2 = 1, \quad a_3 = \frac{1}{2} \quad \text{et} \quad a_4 = \frac{f(t_i, y(t_i))}{2} \quad (\text{IV-31})$$

En remplaçant ces valeurs des coefficients a_i dans l'équation (IV-27), on obtient la formule suivante :

$$y_{i+1} = y_i + h \times f\left(t_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right) \quad \text{avec} \quad k_1 = h \times f(t_i, y_i) \quad \text{et} \quad t_{i+1} = t_i + h \quad (\text{IV-32})$$

IV.4.5 Algorithme de la méthode du point milieu

Les étapes de calcul par la méthode du point milieu suivent l'algorithme suivant :

1. Donner la fonction $f(t, y)$, un intervalle $[a, b]$, une condition initiale (t_0, y_0) , un nombre de sous intervalles n .
2. Calculer le pas h .
3. Calculer les valeurs (t_{i+1}, y_{i+1}) à partir des valeurs (t_i, y_i) en utilisant la formule de la méthode du point milieu.
4. Afficher les résultats et arrêter.

Remarque

Les méthodes d'Euler modifiée et du point milieu étant du même ordre de troncature locale, leur précision est semblable. D'autres choix sont possibles pour les coefficients a_i , mais nous nous limitons aux deux précédents.

IV.4.6 Méthode de Runge-Kutta d'ordre 4

En reprenant le développement de Taylor de la fonction f , mais cette fois jusqu'à l'ordre 5, un raisonnement similaire à celui qui a mené aux méthodes de Runge-Kutta d'ordre 2 aboutit à un système de 8 équations non linéaires comprenant 10 inconnues. Le résultat finale est la méthode de Runge-Kutta d'ordre 4 (RK4), qui représente un outil d'une grande utilité. Sa formule est donnée par :

$$y_{i+1} = y_i + \frac{1}{6} \times f(k_1 + 2k_2 + 2k_3 + k_4) \quad (\text{IV-33})$$

avec :

$$\begin{aligned} k_1 &= h \times f(t_i, y_i) \\ k_2 &= h \times f\left(t_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right) \\ k_3 &= h \times f\left(t_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right) \\ k_4 &= h \times f(t_i + h, y_i + k_3) \\ t_{i+1} &= t_i + h \end{aligned}$$

IV.4.7 Algorithme de la méthode de Runge-Kutta d'ordre 4

Les étapes de calcul par la méthode de Runge-Kutta d'ordre 4 suivent l'algorithme suivant :

Chapitre IV. Résolution numérique des équations différentielles ordinaires

1. Donner la fonction $f(t, y)$, un intervalle $[a, b]$, une condition initiale (t_0, y_0) , un nombre de sous intervalles n .
2. Calculer le pas h .
3. Calculer les valeurs (t_{i+1}, y_{i+1}) à partir des valeurs (t_i, y_i) en utilisant la formule de la méthode de Runge-Kutta d'ordre 4.
4. Afficher les résultats et arrêter.

Remarque

La méthode de Runge-Kutta d'ordre 4 est très fréquemment utilisée en raison de sa grande précision.

IV.4.8 Exercice IV.2

- 1) Calculer la solution approchée de l'équation différentielle (IV-15) par la méthode de Range-Kutta d'ordre 4 pour $n = 4$ sous-intervalles.
- 2) Comparer avec les résultats obtenus dans l'exercice IV.1. Conclure.

IV.4.9 Corrigé d'exercice IV.2

- 1) On utilise la formule de Range-Kutta d'ordre 4 (IV-33). On a :

$$y'(t) = f(t, y(t)) = t + \frac{3}{t}y(t), \quad h = \frac{2-1}{4} = 0.25, \quad (t_0, y_0) = (1, 3)$$

Pour $n = 4$ on fait uniquement 4 itérations. Les résultats obtenus sont dans le tableau suivant :

Itération	k_1	k_2	k_3	k_4	y_{i+1}	t_{i+1}
1	2.5000	3.1146	3.3194	4.1042	6.2454	1.25
2	4.0597	4.8575	5.0751	6.0352	11.2387	1.5
3	5.9944	6.9767	7.2034	8.3413	18.3547	1.75
4	8.3038	9.4714	9.7049	11.0223	27.9678	2

Tableau IV-2 : Solution approchée obtenues par la méthode de RK4 (Corrigé d'exercice IV.2)

Exemple de calcul (première itération) :

$t_0 = a = 1$ et $y_0 = 3$ (données). Calculons alors y_1 et t_1 :

$$k_1 = h \times f(t_0, y_0) = 0.25 \times f(1, 3) = 2.5$$

$$k_2 = h \times f\left(t_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}\right) = 0.25 \times f(1.1250, 4.2500) = 3.1146$$

$$k_3 = h \times f\left(t_0 + \frac{h}{2}, y_0 + \frac{k_2}{2}\right) = 0.25 \times f(1.1250, 4.5573) = 3.3194$$

$$k_4 = h \times f(t_0 + h, y_0 + k_3) = 0.25 \times f(1.25, 5.5573) = 4.1042$$

$$y_1 = y_0 + \frac{1}{6} \times (k_1 + 2k_2 + 2k_3 + k_4) = 3 + \frac{1}{6} \times (2.5 + 2 \times 3.1146 + 2 \times 3.3194 + 4.1042) = 6.2454$$

$$t_1 = t_0 + h = 1 + 0.25 = 1.25$$

Remarquons que ce calcul est très long en utilisant la calculatrice.

- 2) Comparaison avec les résultats de l'exercice IV.1 :

Itération	t_{i+1}	Solution approchée		Solution exacte
		y_{i+1} (Euler)	y_{i+1} (RK4)	y_{i+1} (exacte)
1	1.25	5.5	6.2454	6.2500
2	1.5	9.1125	11.2387	11.2500
3	1.75	14.0438	18.3547	18.3750
4	2	20.5000	27.9678	28

Chapitre IV. Résolution numérique des équations différentielles ordinaires

Tableau IV-3 : Comparaison entre solutions approchées (Euler et RK4) et exacte (Corrigé d'exercice IV.2)

On remarque que la solution obtenue par la méthode RK4 est très proche à la solution exacte par rapport la solution obtenue par la méthode d'Euler. La méthode RK4 est donc plus précise.

Cependant l'erreur se propage toujours d'une itération à l'autre.

IV.4.10 Travail pratique IV.2 : Implémentation MATLAB de la méthode de Range-Kutta d'ordre 4

1) En appliquant l'algorithme de la méthode de Range-Kutta d'ordre 4 (RK4), écrire un programme Matlab qui permet de calculer la solution approchée de l'équation (IV-15) par la méthode de Range-Kutta pour $n = 20$ sous-intervalles, et la solution exacte (IV-16) en même temps.

Note : Représenter les solutions approchées et exactes sur le même graphe afin de comparer.

2) Exécuter ce programme pour des nombres de sous-intervalles $n = 50, 100$ et 200 . Conclure.

3) Matlab dispose des commandes prédéfinies permettant de calculer la solution approchée des équations différentielles ordinaires. Les commandes ode23 et ode45 sont très utiles, leurs syntaxes sont :

$$[t, y] = \text{ode23}(f, [a, b], y_0) \quad \text{et} \quad [t, y] = \text{ode45}(f, [a, b], y_0)$$

En utilisant ces commandes calculer des solutions approchées (représenter les résultats graphiquement avec la solution exacte afin de comparer).

IV.4.11 Corrigé de travail pratique IV.2

1) Le programme Matlab de la méthode RK4 :

```
clc;clear all;close all;
a=1;
b=2;
n=20
h=(b-a)./n;
t=a:h:b;
y=zeros(size(t));
y(1)=3;
f=@(t,y)t+(3./t)*y;
for i=1:n
t(i)=t(1)+(i-1)*h;
k1=h*f(t(i),y(i));
k2=h*f(t(i)+(h./2),y(i)+(k1./2));
k3=h*f(t(i)+(h./2),y(i)+(k2./2));
k4=h*f(t(i)+h,y(i)+k3);
y(i+1)=y(i)+(1./6)*(k1+2.*k2+2.*k3+k4);
end
ys=-(t.^2)+4.*(t.^3);
plot(t,y,'*r',t,ys)
xlabel('t')
ylabel('y')
title('courbes des solutions approchée (pour n=20) et exacte.')
legend('approximée','exacte')
```

Après exécution on obtient :

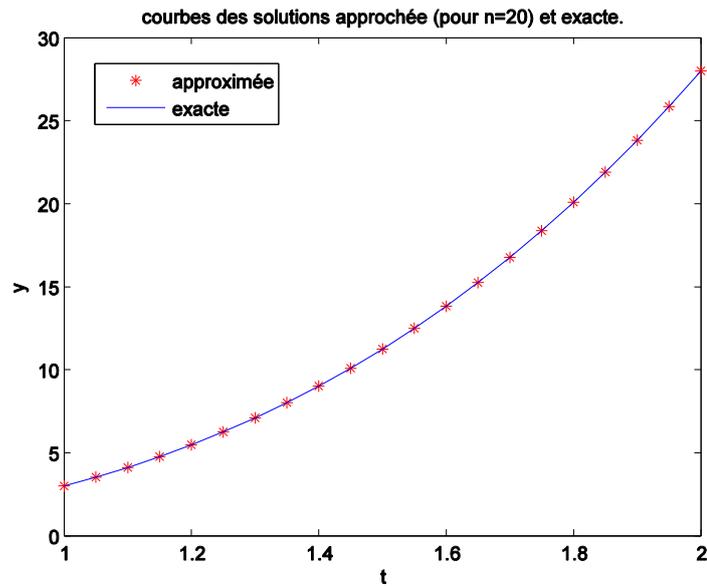


Figure IV.6 : Comparaison entre solution exacte et approchée obtenue par méthode de RK4 pour $n = 20$ (Corrigé de TP IV.2)

2) En augmentant le nombre de sous-intervalles n on obtient :

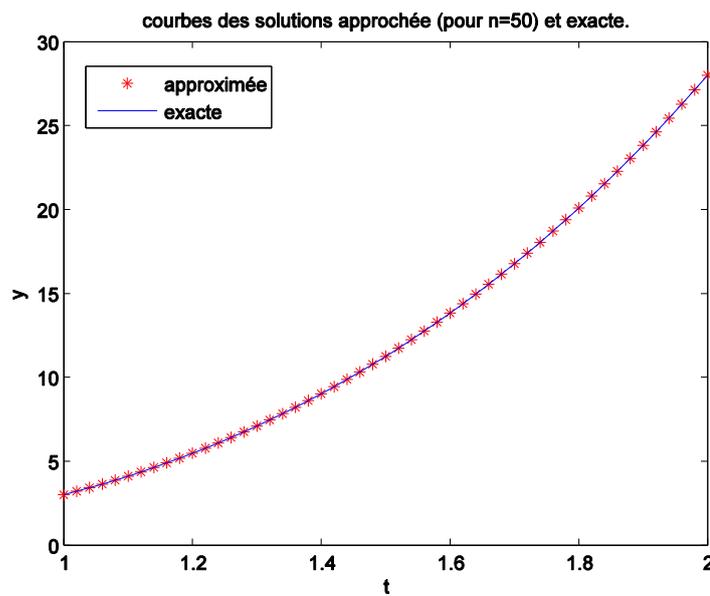


Figure IV.7 : Comparaison entre solution exacte et approchée obtenue par méthode de RK4 pour $n = 50$ (Corrigé de TP IV.2)

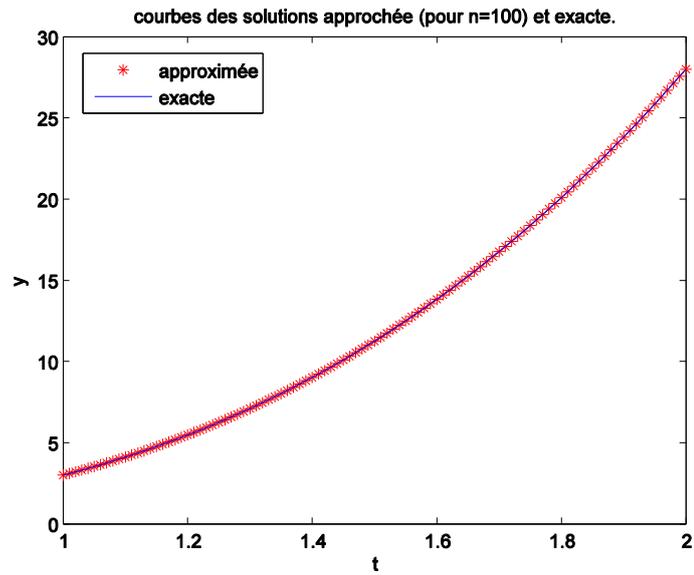


Figure IV.8 : Comparaison entre solution exacte et approchée obtenue par méthode de RK4 pour $n = 100$ (Corrigé de TP IV.2)

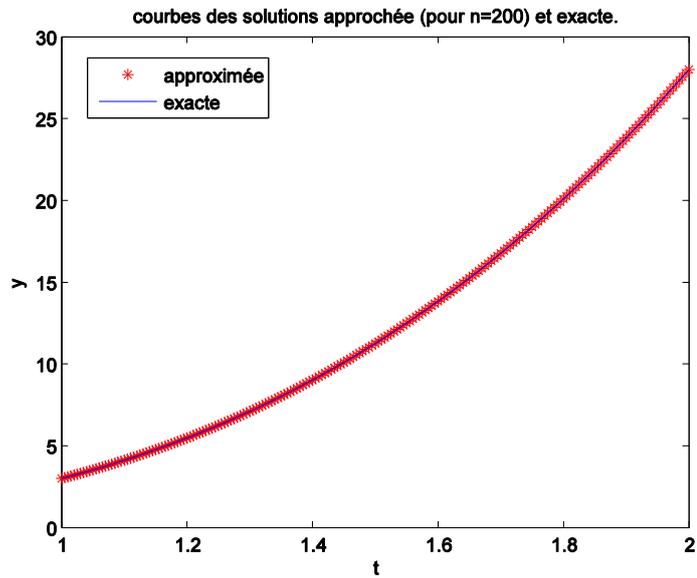


Figure IV.9 : Comparaison entre solution exacte et approchée obtenue par méthode de RK4 pour $n = 200$ (Corrigé de TP IV.2)

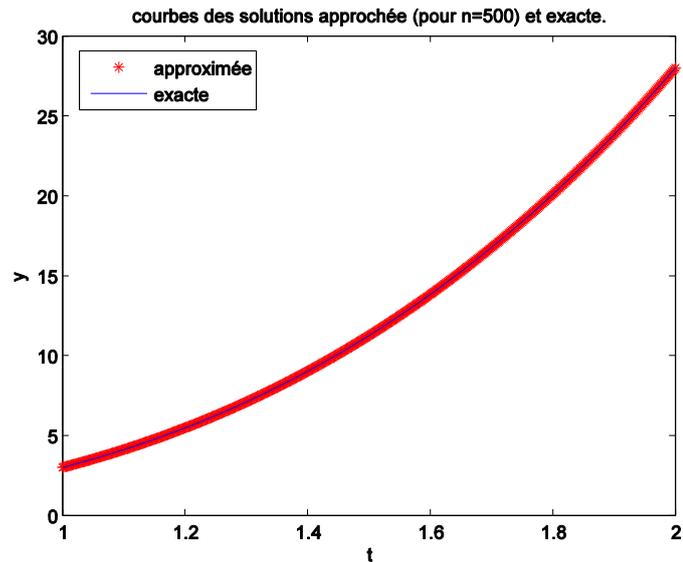


Figure IV.10 : Comparaison entre solution exacte et approchée obtenue par méthode de RK4 pour $n=500$ (Corrigé de TP IV.2)

On remarque que lorsque le nombre de sous-intervalles augmente, la solution approchée par la méthode de RK4 se rapproche plus en plus à la solution exacte. De plus, en comparant avec les résultats obtenus dans le TP IV.1 on remarque que la méthode de RK4 est plus précise que la méthode d'Euler.

3) Le calcul par des commandes prédéfinies :

On écrit un programme dans un fichier M :

```
a=1;
b=2;
y0=3;
f=@(t,y) t+(3./t)*y;
[t,y]=ode23(f,[a ,b],y0)
ys=-(t.^2)+4.*(t.^3);
plot(t,y,'*r',t,ys)
xlabel('t')
ylabel('y')
title('courbes des solutions approchée (commande ode23) et exacte.')
legend('approximée par commande ode23','exacte')
```

C'est le même programme pour la commande ode45. Après exécution on obtient :

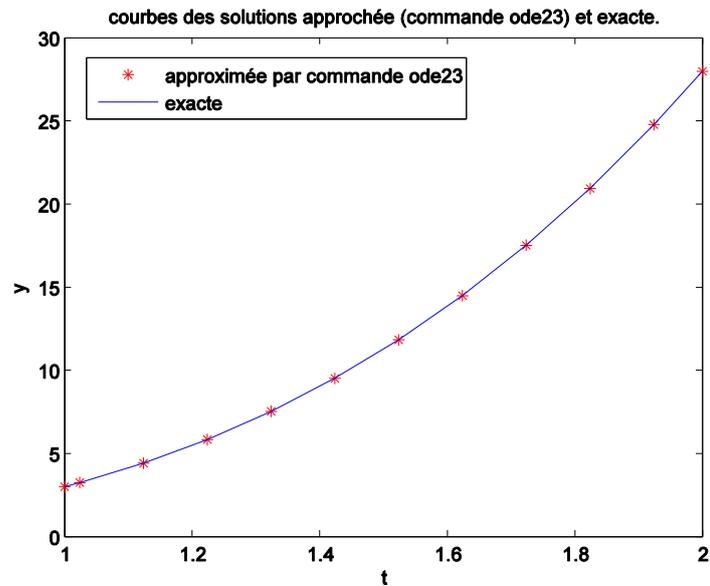


Figure IV.11 : Comparaison entre solution exacte et approchée obtenue par commande ode23 (Corrigé de TP IV.2)

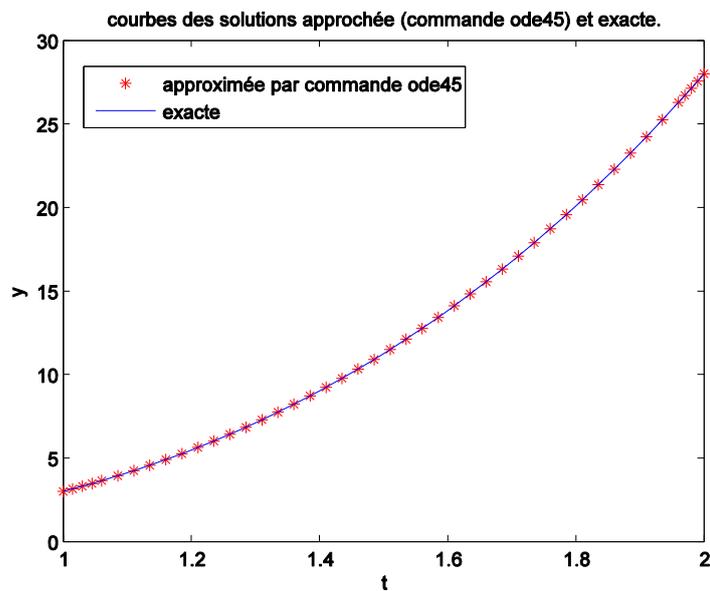


Figure IV.12 : Comparaison entre solution exacte et approchée obtenue par commande ode45 (Corrigé de TP IV.2)