

# Résolution numérique d'équations différentielles

Le but de ce chapitre est de répondre au problème suivant : étant donnée une équation différentielle écrite sous la forme

$$\forall x \in I, \quad u'(x) = f(u(x), x),$$

et une initialisation  $u(x_0) = y_0$ , calculer une approximation de la fonction  $u$  sur l'intervalle  $I$ .

On admet le théorème de Cauchy-Lipschitz donnant l'unicité de la solution à ce problème sous certaines conditions, qu'on supposera réunies.

La question initiale qui peut se poser est alors la façon de représenter la fonction solution  $u$ . Il serait vain de penser pouvoir en donner une expression par des fonctions usuelles, et une représentation purement numérique ne pourra être complète (puisqu'on ne peut renvoyer qu'un nombre fini de valeurs). On pourra distinguer deux situations :

- On est intéressé par une description globale de toute la fonction. Dans ce cas, on peut choisir un pas  $p$  suffisamment petit, et calculer les valeurs de  $u$  à intervalles réguliers de pas  $p$ . On peut compléter ensuite par interpolation linéaire (ou représenter par un graphe, qui lui-même sera tracé par interpolation linéaire, par exemple sous Python).
- On n'est intéressé que par la valeur en  $x$ . Dans ce cas, on fait de même pour l'intervalle  $[x_0, x]$ , au lieu de l'intervalle global. On obtient alors une valeur approchée de  $u(x)$ , sans avoir besoin de faire d'interpolation linéaire.

Dans les deux cas, l'idée est la même : elle consiste à progresser par petits pas, et à calculer à chaque étape une valeur approchée au pas suivant à partir des valeurs déjà obtenues.

Ainsi, partant d'un intervalle  $[a, b]$ , où  $a = x_0$ , on en considère une subdivision  $a = x_0 < x_1 < \dots < x_n = b$ , et on cherche ensuite des approximations des  $u(x_k)$ .

Assez souvent, mais pas systématiquement, on considérera pour  $(x_k)$  la subdivision régulière définie pour tout  $k \in \llbracket 0, n \rrbracket$  par  $x_k = a + kh$ , où  $h = \frac{b-a}{n}$  est le pas constant de la subdivision.

Notre but est de donner plusieurs méthodes d'approximation des valeurs  $u(x_k)$ , et d'étudier, pour la plus simple des méthodes, la convergence des valeurs approchées vers les valeurs exactes lorsque  $n$  tend vers l'infini. Le problème de la convergence se pose essentiellement pour les valeurs lointaines de  $x_0$ . En effet les approximations aux points successifs étant calculées à l'aide des approximations précédentes, les erreurs s'ajoutent de pas en pas : plus on s'éloigne de  $x_0$ , plus l'approximation obtenue va être mauvaise.

Comme toujours en analyse numérique, les questions qui se posent sont les suivantes :

- Rechercher des algorithmes donnant une erreur acceptable, en un temps raisonnable
- Contrôler précisément l'erreur
- Trouver un compromis entre temps de calcul et précision.

Nous ne prétendons pas répondre exhaustivement à ces questions dans ce cours, dont le point de vue se veut essentiellement descriptif et intuitif. Par exemple, le contrôle précis de l'erreur est un problème dépassant les techniques dont nous disposons.

Enfin, remarquons que même si on expose les méthodes pour  $a < b$  (donc pour des valeurs de  $x$  supérieures à la valeur d'initialisation), tout ce qu'on dit est valable aussi lorsque  $b < a$ , en considérant l'intervalle  $[b, a]$ . Dans ce cas, le pas est négatif, et on progresse vers l'arrière.

## I Méthode d'Euler

Dans ce qui suit, on se donne une subdivision  $(x_k)_{0 \leq k \leq n}$  de l'intervalle  $[a, b]$ , et on note  $h$  son pas :

$$h = \max_{k \in \llbracket 0, n-1 \rrbracket} (x_{k+1} - x_k).$$

Dans le cas d'une subdivision régulière, les valeurs  $x_{k+1} - x_k$  sont toutes égales à  $\frac{b-a}{n}$ , donc  $h = \frac{b-a}{n}$ .

La méthode d'Euler est basée sur une idée très simple :

- L'équation différentielle  $u' = f(u, x)$  et la connaissance de  $u(x_0)$  nous fournissent la connaissance de  $u'(x_0)$ , donc de la tangente à la courbe en  $x_0$ .
- On approche alors localement la courbe de  $u$  par sa tangente. Pour un pas suffisamment petit, on peut considérer que cette approximation est valable sur tout l'intervalle  $[x_0, x_1]$ .
- On en déduit alors une valeur approchée  $\tilde{u}(x_1)$  de  $u(x_1)$ .
- On fait comme si cette valeur approchée était la valeur exacte : en utilisant l'équation différentielle, on en déduit une valeur approchée  $\tilde{u}'(x_1)$  de  $u'(x_1)$ .
- On approche à nouveau la courbe de  $u$  par sa tangente, elle-même approchée par la droite calculée à l'aide des valeurs approchées calculées de  $u(x_1)$  et  $u'(x_1)$ . C'est cette droite qui approche la tangente qu'on prend comme nouvelle approximation pour le calcul de  $u(x_2)$ .
- On continue de la sorte.

La construction est illustrée par la figure 9.1. On se rend vite compte qu'on accumule les erreurs et que pour des valeurs éloignées de  $x_0$ , l'approximation risque de ne pas être très bonne. Vous illustrerez ces problèmes de divergence en TP.

Remarkons aussi que si  $u$  est définie sur un intervalle semi-infini (du type  $[0, +\infty[$  par exemple), on ne peut pas définir le pas en fonction de la longueur de l'intervalle total : on considère dans ce cas une subdivision suffisamment petite, et on progresse jusqu'aux valeurs de la variable qui nous intéresse. Il faut faire attention au fait qu'en général, quelle que soit la valeur  $h > 0$  du pas qu'on se donne, lorsque le nombre de pas effectués tend vers l'infini, la solution approchée obtenue s'éloigne fortement de la solution théorique.

### Méthode 9.1.1 (Méthode d'Euler)

Conformément aux explications précédentes, la méthode d'Euler consiste à approcher  $u(x_k)$  par la valeur  $u_k = \tilde{u}(x_k)$ , calculée par la récurrence suivante :

$$u_0 = y_0 \quad \text{et} \quad \forall k \in \llbracket 0, n-1 \rrbracket, \quad u_{k+1} = f(u_k, x_k)(x_{k+1} - x_k) + u_k.$$

Si la subdivision est régulière, de pas constant  $h = \frac{b-a}{n}$ , on peut réécrire la relation de récurrence :

$$\forall k \in \llbracket 0, n-1 \rrbracket, \quad u_{k+1} = hf(u_k, x_k) + u_k.$$

### Proposition 9.1.2 (Majoration de l'erreur pour le calcul de $u_1$ )

Soit  $u$  une fonction de classe  $\mathcal{C}^2$  sur  $[a, b]$ , et  $M_2$  un majorant de  $|u''|$ . Alors, si  $h$  désigne le pas de la subdivision (non nécessairement régulière), c'est-à-dire  $h = \max(x_{k+1} - x_k)$ , alors

$$|u_1 - u(x_1)| \leq \frac{M_2}{2} h^2.$$

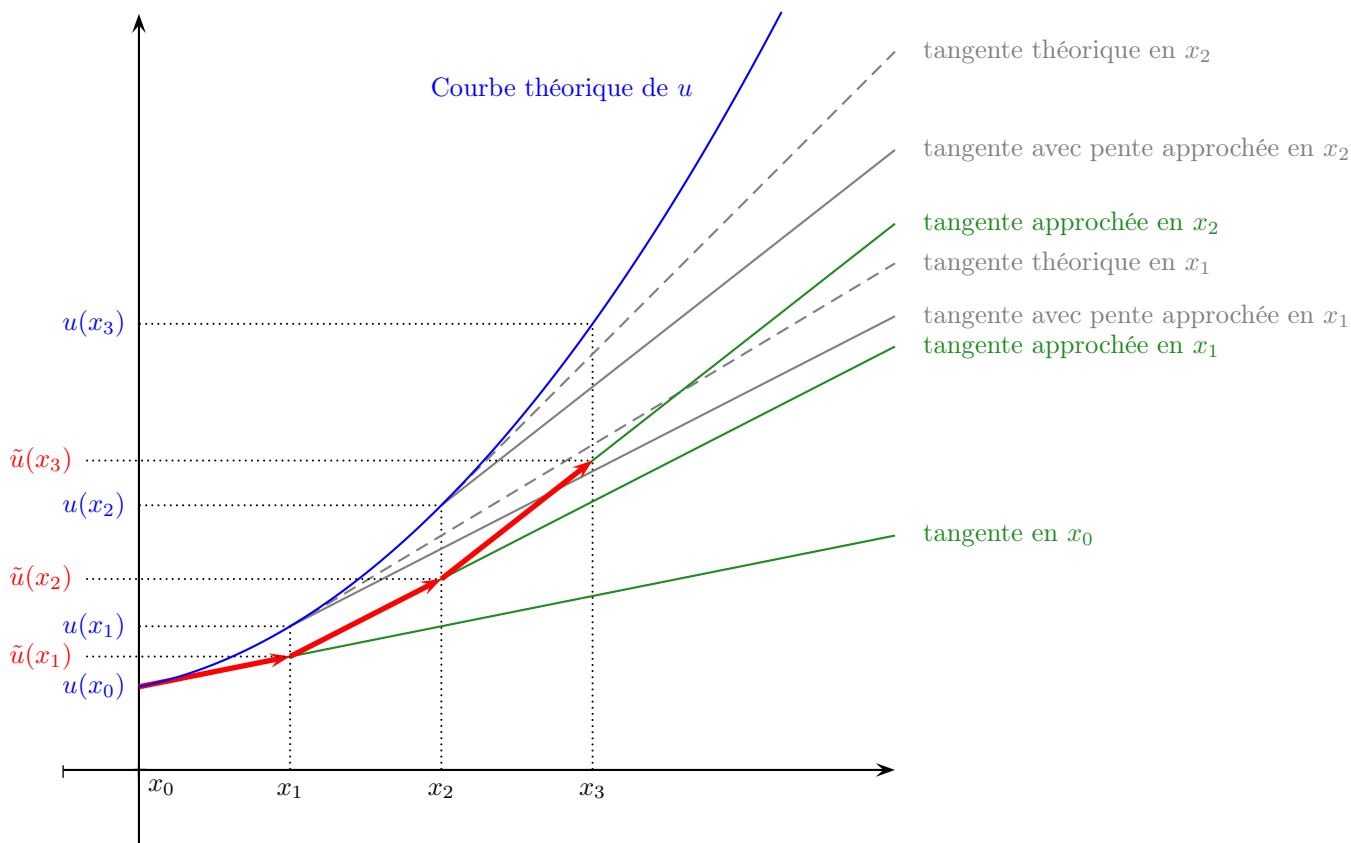


FIGURE 9.1 – Méthode d'Euler

Dans le cas d'une subdivision régulière en  $n$  intervalles, cela donne :

$$|u_1 - u(x_1)| \leq \frac{M_2}{2} \frac{(b-a)^2}{n^2}.$$

◁ Éléments de preuve.

On a  $u_1 = u(x_0) + hu'(x_0)$ . L'inégalité de Taylor-Lagrange donne alors le résultat voulu. ▷

**Proposition 9.1.3 (Inégalité de récurrence pour l'erreur)**

Soit  $f$  de classe  $C^2$  sur  $[a, b]$  et  $M_2$  comme ci-dessus. On suppose la subdivision régulière, de pas  $h$ . Supposons que  $f$  est  $L$ -lipschitzienne par rapport à sa première variable. En notant, pour tout  $k \in \llbracket 0, n \rrbracket$ ,  $e_k = |u(x_k) - u_k|$ , on a :

$$\forall k \in \llbracket 0, n \rrbracket, \quad e_{k+1} \leq (hL + 1)e_k + \frac{M_2}{2}h^2.$$

◁ Éléments de preuve.

Il s'agit encore de l'inégalité de Taylor-Lagrange, mais pour pouvoir l'utiliser, il faut passer des valeurs calculées aux valeurs théoriques. L'erreur faite en approchant  $u(x_k)$  par  $u_k$  est l'erreur précédente  $e_k$ . L'erreur faite en approchant la pente de la tangente par  $f(u_k, x_k)$  se contrôle par le caractère lipschitzien de  $f$ , cela fournit le facteur  $hLe_k$ . ▷

**Corollaire 9.1.4 (Majoration de l'erreur)**

Sous les mêmes hypothèses sur  $f$ , en négligeant les erreurs d'arrondi dans les calculs, l'erreur faite sur le calcul de  $u(x_k)$  vérifie :

$$\forall k \in \mathbb{N}, \quad e_k \leq \frac{M_2 h}{2L} \left( (1 + hL)^k - 1 \right) \leq e^{khL} \frac{M_2 h}{2L}.$$

Cette majoration reste valable même pour des valeurs de  $k$  dépassant  $n$ , si on poursuit la méthode d'Euler au-delà de la valeur  $b$ , à condition que  $M_2$  soit un majorant de  $|u''|$  sur l'intervalle adéquat, mais dans ce cas, on a un contrôle exponentiel divergent, donc très mauvais (ce qui explique la possibilité d'une divergence forte par rapport à la solution théorique).

Si on reste dans l'intervalle compact  $[a, b]$ , on obtient une majoration uniforme de l'erreur :

$$\forall k \in \llbracket 0, n \rrbracket, \quad e_k \leq \frac{Mh}{2L} e^{(b-a)L}.$$

◁ **Éléments de preuve.**

La première inégalité provient de ce qui précède. On peut d'abord résoudre la suite arithmético-géométrique obtenue en remplaçant l'inégalité par une égalité dans la proposition précédente. Ou raisonner directement par récurrence.

La seconde inégalité provient de l'inégalité de convexité  $1 + hL \leq e^{hL}$ . ▷

Nous étudions maintenant l'exemple particulier de l'équation différentielle  $u' = au$ , avec  $u(0) = 1$ . Évidemment, la solution théorique est bien connue, il s'agit de l'exponentielle :  $x \mapsto e^{ax}$ . La connaissance de la solution théorique va nous permettre de déterminer de façon plus précise l'erreur faite par la résolution par la méthode d'Euler, dans ce cas particulier. Notre but par l'étude de cet exemple est de montrer qu'en général, on ne peut pas avoir un meilleur contrôle en  $+\infty$  (lorsqu'on continue la méthode au-delà du point  $b$ ) que ce contrôle exponentiel obtenu dans le corollaire qui précède.

On considère ici le pas  $h = \frac{1}{n}$ . Implicitement, cela revient à considérer l'intervalle initial d'étude  $[0, 1]$ , puis à prolonger le calcul à l'infini.

On montre qu'alors, on a, pour tout  $k \in \mathbb{N}$ ,

$$u_k = \left( 1 + \frac{a}{n} \right)^k$$

On vérifie d'ailleurs la cohérence de ce résultat au point 1 correspondant à  $k = n$ , l'expression trouvée étant de limite  $e^a$  (calcul classique et facile).

L'erreur s'exprime alors de façon exacte :

$$e_k = \left| u\left(\frac{k}{n}\right) - u_k \right| = \left| e^{a\frac{k}{n}} - \left( 1 + \frac{a}{n} \right)^k \right| \underset{k \rightarrow +\infty}{\sim} e^{a\frac{k}{n}}.$$

Ainsi, pour tout pas  $\frac{1}{n}$  fixé, la méthode d'Euler prolongée à l'infini donne une divergence exponentielle sur cet exemple.

**Remarque 9.1.5 (Méthode d'Euler pour  $f$  indépendante de  $u$ )**

Si  $f$  ne dépend que de  $x$ , l'équation à résoudre est  $y' = f(x)$ . Il s'agit d'une primitivation, donc d'un calcul d'intégrale. La relation de récurrence sur les termes  $u_k$  est alors simplement

$$y_{k+1} = y_k + f(x_k)h.$$

On se rend compte que la méthode d'Euler appliquée à cette situation correspond très précisément à la méthode des rectangles pour le calcul des intégrales, dont on connaît bien la lenteur et le manque d'efficacité...

## II Notion d'ordre d'une méthode

De façon générale, lorsqu'on dispose d'une méthode de résolution approchée d'équations différentielles, on veut pouvoir estimer son efficacité, et en particulier l'erreur d'approximation faite par cette méthode. Les calculs exacts sont parfois durs à mener : majorer de façon exacte l'erreur dans le cas de la méthode particulièrement simple d'Euler a déjà demandé beaucoup de travail. Pour des méthodes plus sophistiquées, cela relève quasiment de l'exploit !

On opère alors une hypothèse simplificatrice pour le calcul de l'erreur, en négligeant l'erreur faite sur la pente lors du calcul de la tangente approchée. Ainsi, notant  $f'_k$  l'erreur induite par l'étape  $k$ , l'erreur totale au rang  $k$  est

$$e'_k = \sum_{i=1}^k f'_i.$$

Pour tout  $k \in \llbracket 1, n-1 \rrbracket$ , on a donc  $e'_{k+1} = e'_k + f'_{k+1}$ .

Par ailleurs, on fait la supposition, pour le calcul de l'erreur à la  $k$ -ième étape que la valeur calculée  $u(x_k)$  est la valeur exacte. Ainsi, l'erreur approchée  $f'_k$  faite à la  $k$ -ième étape se calcule de la même manière que celle de la première étape, lorsqu'on dispose des valeurs exactes (figure 9.2). L'erreur totale ainsi obtenue est alors

$$e_k = \sum_{i=1}^n f_k.$$

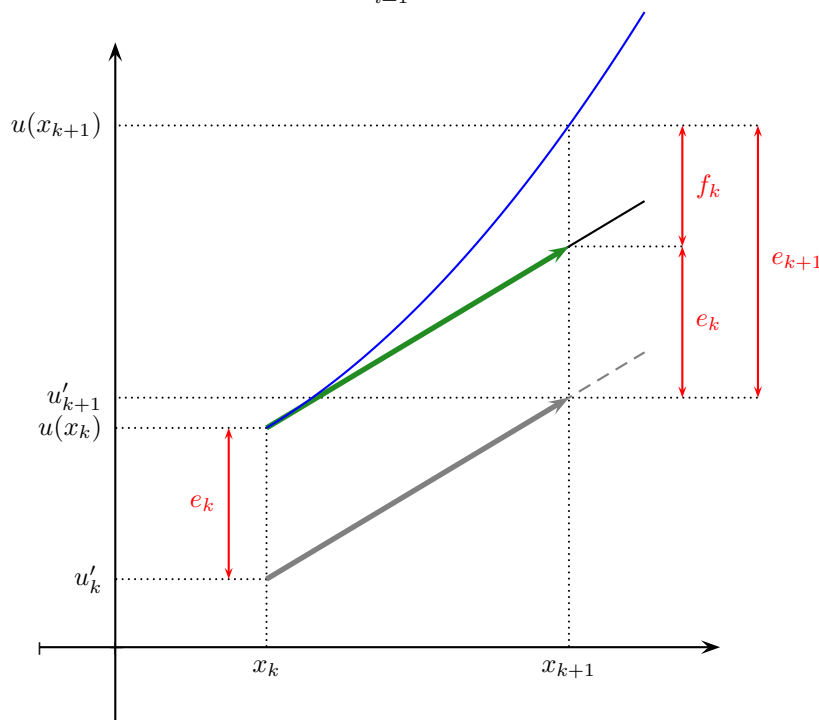


FIGURE 9.2 – Erreurs pour la méthode d'Euler

On définit alors l'erreur de consistance par :

### Définition 9.2.1 (Erreur de consistance)

L'erreur de consistance d'une méthode de résolution d'ED est la quantité :

$$e(h) = \sum_{k=1}^n |u'_k - u(x_k)|,$$

où  $h = \frac{b-a}{n}$ , et  $u'_k$  est la valeur approchée qu'on obtiendrait par la méthode utilisée, en prenant comme point de départ  $(x_{k-1}, y(x_{k-1}))$  (donc la valeur théorique).

**Définition 9.2.2 (Ordre d'une méthode)**

L'ordre d'une méthode de résolution numérique d'ED est l'ordre de la puissance de  $h$  contrôlant l'erreur de consistance.

**Exemple 9.2.3 (Ordre de la méthode d'Euler)**

Le calcul de  $|u_1 - u(t_1)|$  effectué plus haut pour la méthode d'Euler est valable plus généralement pour le calcul de  $|u'_k - u(x_k)|$ . On obtient alors l'erreur de consistance :

$$e(h) \leq \sum_{k=1}^n \frac{M_2}{2} h^2 = \frac{M_2(b-a)}{2} h.$$

Ainsi, la méthode d'Euler est d'ordre 1.

**Remarques 9.2.4**

1. La majoration de l'exemple précédent ne donne *stricto sensu* pas l'ordre de la méthode, mais seulement une minoration de l'ordre. Pour justifier que l'ordre est exactement 1, il faut montrer qu'on ne peut pas obtenir une meilleure convergence en exhibant un cas particulier dans lequel on sait calculer l'erreur de consistance de façon exacte. On pourrait reprendre l'exemple de l'exponentielle et calculer l'erreur de consistance dans ce cas particulier afin de conclure.
2. Très souvent, on peut obtenir une majoration uniforme  $|u'_k - u(x_k)| \leq Mh^a$ , valable sur tout l'intervalle  $[a, b]$  (c'est-à-dire pour tout  $k \in \llbracket 1, n \rrbracket$ ). L'ordre de la méthode est dans ce cas (au moins)  $a - 1$ .

### III Équations différentielles d'ordre supérieur

On admet que la méthode d'Euler reste valable pour des fonctions vectorielles  $U : \mathbb{R} \rightarrow \mathbb{R}^p$ , et  $f : \mathbb{R}^p \times \mathbb{R} \rightarrow \mathbb{R}^p$  vérifiant  $U'(t) = f(U(t), t)$ . On pourra alors ramener une équation différentielle linéaire d'ordre  $p$  à cette situation, après mise sous forme normale :

$$u^{(p)}(t) = F(u(t), u'(t), \dots, u^{(p-1)}(t), t).$$

Il suffit pour cela de poser la fonction vectorielle

$$U : t \mapsto \begin{pmatrix} u(t) \\ u'(t) \\ \vdots \\ u^{(p-1)}(t) \end{pmatrix}.$$

En notant  $U = \begin{pmatrix} u_0 \\ \vdots \\ u_{p-1} \end{pmatrix}$ , cette fonction vectorielle est alors solution de l'équation différentielle

$$U'(t) = \begin{pmatrix} u_1(t) \\ \vdots \\ u_{p-1}(t) \\ F(u_0(t), \dots, u_{p-1}(t), t) \end{pmatrix}.$$

Il suffit donc de définir  $F$  par

$$f(x_0, \dots, x_{p-1}, t) = (x_1, \dots, x_{p-1}, F(x_0, \dots, x_{p-1}, t)).$$

Cette manipulation peut aussi être faite en vue d'utiliser la fonction `odeint` (ODE pour Ordinary Differential Equation) du module `scipy.integrate`, prenant en argument une fonction  $f$ , une valeur initiale  $y_0$  et un tableau  $T$  de temps (la subdivision de l'intervalle en laquelle calculer les valeurs de  $u$ ), et revoyant sous forme d'un tableau les valeurs de  $u$  aux temps de  $T$ . La fonction  $f$  peut être vectorielle; dans ce cas, les données vectorielles de  $\mathbb{R}^p$  doivent être rentrées sous forme d'une liste de coordonnées (ou d'un tableau `numpy`).

La fonction `odeint` ci-dessus est basée sur la méthode de Runge-Kutta d'ordre 4. Même si les méthodes de Runge et Kutta sont hors programme, nous les évoquons ci-dessous (d'un point de vue purement descriptif), afin de comprendre comment est définie cette fonction `odeint`. Il s'agit de méthodes basées sur le même principe que la méthode d'Euler (qui en est d'ailleurs un cas particulier).

## IV Méthode de Runge-Kutta, HP

La méthode de Runge-Kutta a été mise au point au tout début du 20<sup>e</sup> siècle par Carl Runge et Martin Wilhelm Kutta, deux mathématiciens allemands.

L'idée est grossièrement la même que pour la méthode d'Euler : on subdivise l'intervalle d'étude, et on approche à chaque étape la courbe par une certaine droite, dont la pente est choisie judicieusement. Dans la méthode d'Euler, la pente choisie est la pente de la tangente au point initial de l'intervalle. La méthode de Runge-Kutta consiste à prendre comme pente une meilleure valeur approchée de la pente moyenne entre  $x_k$  et  $x_{k+1}$  : cette pente moyenne nous donnerait alors la valeur exacte de  $u(x_{k+1})$  (si on part de la valeur exacte de  $u(x_k)$ ). Plus on s'approche de cette pente, plus l'erreur va être petite.

L'idée est alors de faire une moyenne des pentes en différents points répartis de façon déterminée sur l'intervalle  $[x_k, x_{k+1}]$  (figure 9.3). Ces pentes en des points intermédiaires seront elles même calculées de façon approchée par une méthode de type Euler. Les différentes méthodes de Runge-Kutta correspondent alors au choix du nombre de points intermédiaires pour le calcul de la pente moyenne approchée, et au choix de leur répartition dans l'intervalle  $[x_k, x_{k+1}]$ .

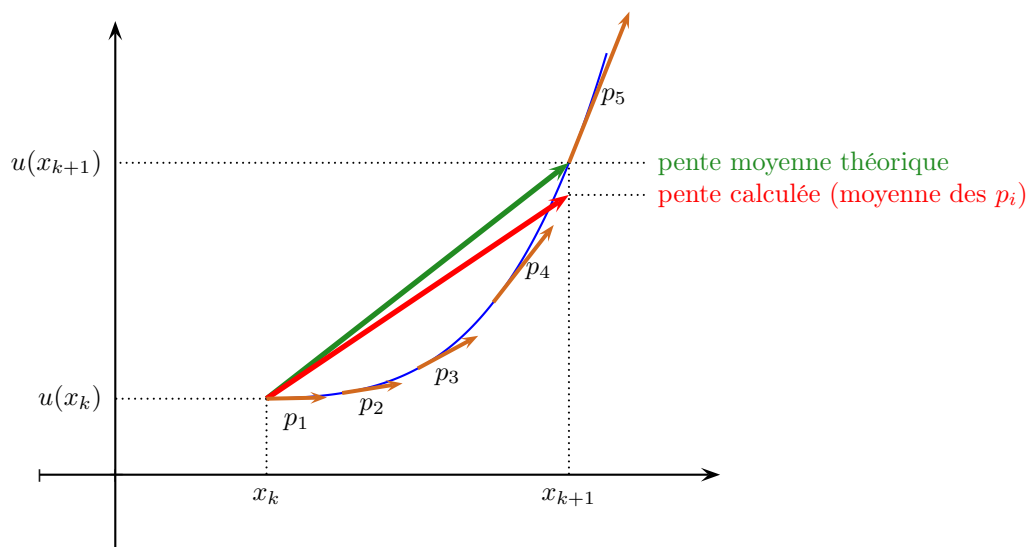


FIGURE 9.3 – Pente moyenne sur un intervalle  $[x_k, x_{k+1}]$

Soit  $s$  un entier (correspondant au nombre de points intermédiaires pour le calcul de la moyenne des pentes). Soit  $\beta_1 \leq \dots \leq \beta_s$  des réels de  $[0, 1]$  donnant la répartition des points  $q_1, \dots, q_s$  dans  $[x_k, x_{k+1}]$ , (les  $\beta_i$  correspondent à une paramétrisation linéaire par  $[0, 1]$  de l'intervalle  $[x_k, x_{k+1}]$ ). Ainsi :

$$q_i = x_k + \beta_i h,$$

où  $h = x_{k+1} - x_k$  est le pas de la subdivision, supposée régulière. Soit ensuite  $(\alpha_1, \dots, \alpha_s)$  un  $s$ -uplet de réels de  $[0, 1]$  tels que

$$\alpha_1 + \dots + \alpha_s = 1.$$

On approche la pente moyenne  $p = \frac{u(x_{k+1}) - u(x_k)}{h}$  sur l'intervalle  $[x_{k+1}, x_k]$  par la moyenne pondérée par les  $\alpha_i$  des pentes théoriques aux points  $q_i$ , à savoir :

$$p \simeq \sum_{i=1}^s \alpha_i f(u(q_i), q_i).$$

Comme on ne connaît pas précisément  $u(q_i)$ , on utilise une approximation des  $u(q_i)$ , calculée suivant le même procédé que dans la méthode d'Euler, par approximations successives.

#### Méthode 9.4.1 (calcul d'une valeur approchée de la pente moyenne sur $[x_0, x_1]$ )

Pour expliquer la construction, plaçons-nous sur l'intervalle  $[x_0, x_1]$ . Ainsi,  $u(x_0) = y_0$  est connu. En notant  $p_i$  la pente calculée au point  $q_i$ , on procède de la sorte :

- On calcule  $p_1$  par la méthode d'Euler en partant de  $x_k$ . On commence par déterminer une valeur approchée  $v_1$  de  $u(q_1)$  :

$$v_1 = u(x_0) + (q_1 - x_0)f(u(x_0), x_0).$$

On définit alors  $p_1 = f(v_1, q_1)$

- On calcule  $p_2$  en repartant du début de l'intervalle (en  $x_0$ ), mais en approchant cette fois la courbe par la droite de pente  $p_1$  calculée dans l'étape précédente, et issue du point  $(x_0, u(x_0))$ . Une approximation de  $u(q_2)$  est alors :

$$v_2 = u(x_0) + (q_2 - x_0)p_1 \quad \text{et} \quad p_2 = f(v_2, q_2).$$

- De façon plus générale,  $p_k$  étant déterminé,  $p_{k+1}$  est calculé en repartant de  $x_0$  et en approchant la courbe par une droite de pente  $p_k$  issue de  $(x_0, u(x_0))$  :

$$v_{k+1} = u(x_0) + (q_{k+1} - x_0)p_k \quad \text{et} \quad p_{k+1} = f(v_{k+1}, q_{k+1}).$$

- Les  $p_i$  étant tous déterminés ainsi, on approche  $p$  la pente moyenne sur  $[x_0, x_1]$  (c'est-à-dire le taux d'accroissement) par

$$p \simeq \sum_{i=1}^s \alpha_i p_i.$$

#### Méthode 9.4.2 (Méthode de Runge-Kutta)

Les  $\beta_i$  (donc la répartition des  $q_i$  dans un intervalle  $[x_k, x_{k+1}]$ ) et les poids  $\alpha_i$  étant donnés, la méthode de Runge Kutta s'opère de la façon suivante : on calcule successivement des valeurs approchées  $u_k$  de  $u(x_k)$ . Supposant  $u_k$  déterminé, on détermine  $u_{k+1}$  ainsi :

- on calcule une approximation  $p$  de la pente sur  $[x_k, x_{k+1}]$  comme expliqué ci-dessus, en considérant qu'on peut remplacer dans les calculs la valeur exacte  $u(x_k)$  par la valeur approchée  $u_k$ .
- On calcule  $u_{k+1}$  en approchant  $u$  sur  $[x_k, x_{k+1}]$  par la droite issue de  $(x_k, u_k)$ , de pente  $p$  :

$$u_{k+1} = u_k + hp.$$

Nous explicitons ci-dessous les trois cas les plus fréquents (méthodes Runge-Kutta d'ordre 1, 2 et 4), obtenues pour certains choix de  $s$ , des répartitions  $\beta_i$  et des pondérations  $\alpha_i$ .



**Définition 9.4.3 (Méthode de Runge-Kutta d'ordre 1, RK1)**

La méthode de Runge-Kutta d'ordre 1 consiste en le choix de  $s = 1$  (un seul point pour obtenir l'approximation),  $\alpha_1 = 1$  et  $\beta_1 = 0$  (pour faire la moyenne, on considère la pente au bord gauche de l'intervalle)

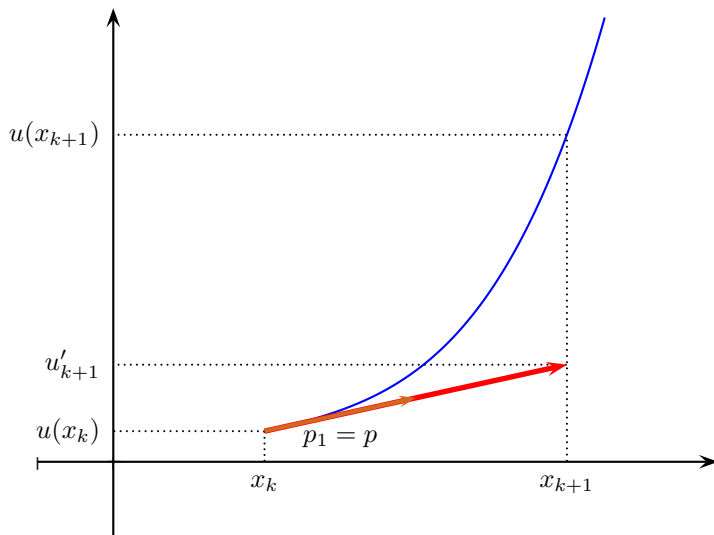


FIGURE 9.4 – Runge-Kutta d'ordre 1

On reconnaît évidemment une méthode déjà étudiée !

**Proposition 9.4.4 (RK1 = Euler)**

La méthode de Runge-Kutta d'ordre 1 n'est autre que la méthode d'Euler. Cela justifie son ordre 1.

**Définition 9.4.5 (Méthode de Runge-Kutta d'ordre 2, RK2)**

La méthode de Runge-Kutta d'ordre 2 consiste en le choix de  $s = 1$ ,  $\alpha_1 = 1$  et  $\beta_1 = \frac{1}{2}$ .

Ainsi, dans cette méthode, la pente moyenne est approchée par la pente au point médian. Graphiquement, ce choix semble bien meilleur (figure 9.5). D'ailleurs, on a déjà rencontré une méthode d'analyse numérique dans laquelle le choix du point médian était judicieux. Les deux situations ne sont d'ailleurs pas sans rapport l'une avec l'autre :

**Remarque 9.4.6 (RK2 et méthode du point milieu)**

La méthode RK2 pour la résolution de  $y' = f(x)$  n'est autre que le calcul intégral par la méthode du point milieu.

**Méthode 9.4.7 (Calcul des approximations successives par RK2)**

Suivant la description générale,  $u_k$  étant connu, on obtient  $u_{k+1}$  par la méthode RK2 de la façon suivante :

$$v = u_k + \frac{h}{2} f(u_k, x_k) \quad p = f(v, x_k + \frac{h}{2}) \quad u_{k+1} = u_k + hp.$$

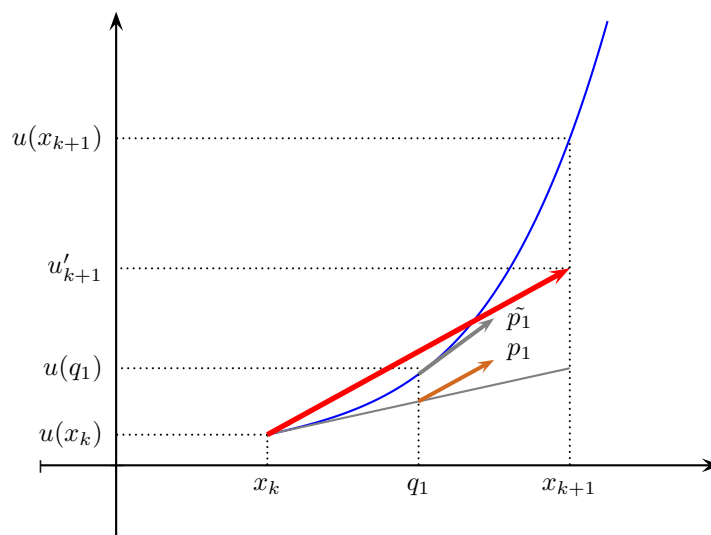


FIGURE 9.5 – Runge-Kutta d'ordre 2

**Théorème 9.4.8 (ordre de RK2)**

Comme son nom l'indique, la méthode RK2 est une méthode d'ordre 2 (si  $f$  est lipschitzienne).

◁ **Éléments de preuve.**

Estimer  $u(x_1) - x_1$  en introduisant  $u(x_{\frac{1}{2}})$ , où  $x_{\frac{1}{2}} = x + \frac{h}{2}$ . Remplacer  $x_1$  par son expression, sommer 2 formules de Taylor (les termes d'ordre 1 s'ajoutent, ceux d'ordre 2 se compensent, et il reste un  $O(h^3)$ ). Regrouper les termes en  $h \times f$  en utilisant le caractère lipschitzien, puis encore Taylor. Pour conclure, il faudrait trouver un exemple pour lequel on obtient effectivement un terme en  $h^3$ . Le cas de l'exponentielle (équation  $u' = u$ ) convient. ▷

On présente maintenant l'algorithme de Runge-Kutta classique, appelé algorithme de Runge-Kutta d'ordre 4, pour une raison que je vous laisse deviner. C'est l'algorithme de résolution numérique d'ED le plus fréquemment employé, du fait qu'il allie simplicité de programmation (une fois qu'on a compris la méthode) et rapidité de convergence. Cette rapidité de convergence est aussi ce qui assure la fiabilité du calcul, puisqu'elle permet d'éviter une trop grande accumulation d'erreurs d'arrondi.

**Définition 9.4.9 (Méthode de Runge-Kutta d'ordre 4, RK4)**

La méthode de Runge-Kutta d'ordre 4 consiste en le choix de  $s = 4$ , de la répartition  $(\beta_1, \beta_2, \beta_3, \beta_4) = (0, \frac{1}{2}, \frac{1}{2}, 1)$ , et des pondérations  $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (\frac{1}{6}, \frac{2}{6}, \frac{2}{6}, \frac{1}{6})$ .

Le fait de choisir deux fois le même point n'est pas redondant : en notant  $q$  ce point milieu, on commence par trouver une première approximation de la valeur de  $u(q)$ , donc une première approximation de la pente  $p$  en  $q$ , puis, on en recalcule une nouvelle approximation (qu'on espère meilleure, ou au moins compensant l'erreur précédente) en utilisant cette fois pour le calcul la nouvelle pente choisie (voir figure 9.6).

**Méthode 9.4.10 (Calcul des approximations successives par RK4)**

Suivant la description générale,  $u_k$  étant connu, on obtient  $u_{k+1}$  par la méthode RK4 par l'enchaînement suivant de calculs :

- $v_1 = u_k$

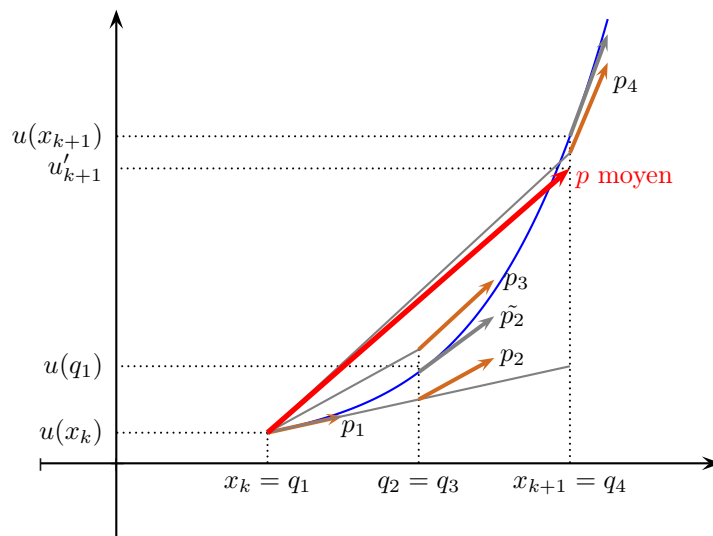


FIGURE 9.6 – Runge-Kutta d'ordre 4

- $p_1 = f(v_1, x_k)$
- $v_2 = u_k + \frac{h}{2}p_1$
- $p_2 = f(v_2, x_k + \frac{h}{2})$
- $v_3 = u_k + \frac{h}{2}p_2$
- $p_3 = f(v_3, x_k + \frac{h}{2})$
- $v_4 = u_k + hp_3$
- $p_4 = f(v_4, x_k + h)$
- $p = \frac{1}{6}(p_1 + 2p_2 + 2p_3 + p_4)$
- $u_{k+1} = u_k + ph$ .

Le choix des pondérations 1, 2, 2, 1, les deux points du milieu étant confondus, nous rappelle la pondération 1, 4, 1 de la méthode de Simpson. Ici aussi, la méthode de Simpson peut être vue comme un cas particulier de la méthode RK4 :

**Remarque 9.4.11 (RK4 et méthode de Simpson)**

La méthode RK4 appliquée à la résolution de  $y' = f(x)$  n'est autre que la méthode de Simpson.

Cette constatation peut nous faire espérer une très bonne convergence et efficacité de la méthode RK4! En effet, on obtient, sans surprise vu le nom donné à la méthode :

**Théorème 9.4.12 (Ordre de la méthode RK4, admis)**

La méthode RK4 est d'ordre 4.