

temps discret
temps continu

(biol 367)

YOU KNOW, I DON'T
THINK MATH IS A SCIENCE.
I THINK IT'S A RELIGION.
ALL THESE EQUATIONS
ARE LIKE MIRACLES. YOU
TAKE TWO NUMBERS AND WHEN
YOU ADD THEM, THEY MAGICALLY
BECOME ONE *NEW* NUMBER!
NO ONE CAN SAY HOW IT
HAPPENS. YOU EITHER BELIEVE
IT OR YOU DON'T.



© Bill Watterson, Calvin & Hobbes.

1. Modèles d'évolution en temps discret

Notre exemple principal sera l'évolution de l'effectif d'une population, de génération en génération (ou d'année en année, ou même de mois en mois).

1.1. Modèle de Malthus

On considère des effectifs E_n pour $n = 1, 2, \dots$, E_1 étant l'effectif à la génération 1 (ou à l'année 1), etc. Le modèle de Malthus considère que la natalité et la mortalité sont constantes : elles ne varient pas avec le temps, elles ne dépendent pas de l'effectif de la population.

Si on a par exemple une natalité d'une naissance pour 1000 individus (sur la période de temps considérée), qu'on notera $\alpha = \frac{1}{1000}$, et une mortalité d'un décès pour 1500 individus, qu'on notera $\beta = \frac{1}{1500}$, à la génération n on a αE_n naissances et βE_n décès; la population à la génération suivante est donc

$$\begin{aligned} E_{n+1} &= E_n + \alpha E_n - \beta E_n \\ &= (1 + \alpha - \beta) E_n \\ &= (1 + r) E_n \end{aligned}$$

où on a noté $r = \alpha - \beta$ le taux d'accroissement (relatif) de la population. Dans notre exemple $r = \frac{1}{3000}$: on parle d'un taux d'accroissement de 3 pour 1000. Notons que

$$r = \frac{E_{n+1} - E_n}{E_n}.$$

Si $r > 0$, le taux de natalité est plus grand que le taux de mortalité, et la population augmente; si $r < 0$, la mortalité l'emporte et la population diminue.

On peut simuler l'évolution de l'effectif, génération après génération; dans la fonction suivante, le paramètre N est le nombre de générations à simuler, E_1 est l'effectif initial, et r est le taux d'accroissement.

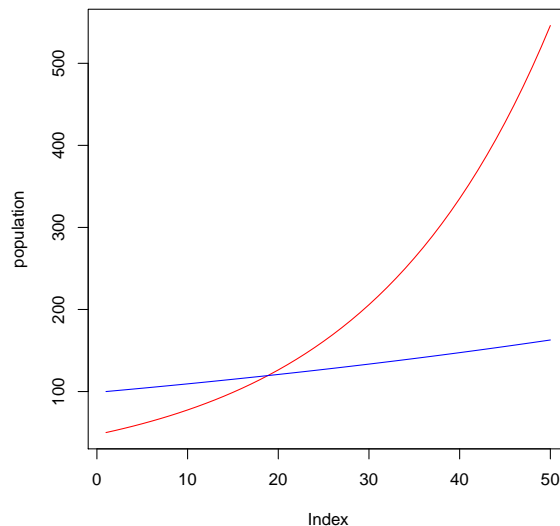
```
Malthus <- fonction(N, E1, r)
{
  E <- numeric(N);
  E[1] <- E1;
  for(i in 2:N)
  {
    E[i] <- (1+r)*E[i-1];
  }
  return(E);
}
```

On peut utiliser cette fonction pour explorer le comportement du modèle :

```
> Malthus(N=30, E1=100, r=0.003)
```

1. Modèles d'évolution en temps discret

```
[1] 100.0000 100.3000 100.6009 100.9027 101.2054 101.5090 101.8136 102.1190 102.4254
[10] 102.7326 103.0408 103.3499 103.6600 103.9710 104.2829 104.5957 104.9095 105.2243
[19] 105.5399 105.8565 106.1741 106.4926 106.8121 107.1326 107.4540 107.7763 108.0996
[28] 108.4239 108.7492 109.0755
> plot( Malthus(N=50, E1=50, r=0.05), ylab="population", type="l", col="red")
> lines( 1:50, Malthus(N=50, E1=100, r=0.01), col="blue")
```



On a ici superposé l'évolution pendant 50 générations de deux populations, une avec effectif initial $E_1 = 50$ et un taux de croissance de 5% (en rouge), l'autre avec effectif initial $E_1 = 100$ et un taux de croissance de 1% (en bleu).

Le graphe évoque une croissance exponentielle. En effet, on peut facilement trouver une expression de E_n en fonction de n et de l'effectif initial E_1 :

$$E_n = (1 + r)^{n-1} E_1. \quad (1.1)$$

Si $r > 0$, l'effectif de la population tend vers l'infini, et double toutes les $N = \frac{\log 2}{\log(1+r)}$ années (ou mois, générations, etc) (pourquoi?). On appelle N la demi-période; si r est petit, on a $N \approx \frac{0.7}{r}$, et dans notre exemple $N \approx 233$.

Si $r < 0$, l'effectif tend vers 0, et est divisé par 2 toutes les $N = -\frac{\log 2}{\log(1+r)}$ années (mois, etc); on appelle N la demi-vie de la population.

La formule 1.1 peut être utilisée pour calculer l'évolution beaucoup plus simplement.

```
> n <- 1:30
> E <- 100*(1.003)**n
> E
[1] 100.3000 100.6009 100.9027 101.2054 101.5090 101.8136 102.1190 102.4254 102.7326
[10] 103.0408 103.3499 103.6600 103.9710 104.2829 104.5957 104.9095 105.2243 105.5399
[19] 105.8565 106.1741 106.4926 106.8121 107.1326 107.4540 107.7763 108.0996 108.4239
[28] 108.7492 109.0755 109.4027
```

1.2. Modèle de Fibonacci

Le mathématicien italien Fibonacci (XIII^e siècle) a posé dans son *Liber Abaci* le problème suivant :

« Un homme met un couple de lapins dans un lieu isolé de tous les côtés par un mur. Combien de couples obtient-on en un an si chaque couple engendre tous les mois un nouveau couple à compter du troisième mois de son existence? »

En voici brièvement la solution : en notant u_n le nombre de couples au n^e mois, on a $u_n = u_{n-1} + u_{n-2}$, c'est-à-dire le nombre de couples au mois précédent, plus le nombre de couples qui étaient déjà présents deux mois plus tôt et qui commencent à procréer.

La suite définie par $u_1 = 1$, $u_2 = 1$, et $u_n = u_{n-1} + u_{n-2}$ est appelée *suite de Fibonacci*.

La fonction suivante renvoie un vecteur contenant les n premiers termes de la suite de Fibonacci.

```
Leonardo <- function(n)
{
  u <- numeric(n);
  u[1] <- 1;
  u[2] <- 1;
  for(i in 3:n)
  {
    u[i] <- u[i-1] + u[i-2];
  }
  return(u);
}
```

Exercice Répondre à la question posée par Fibonacci. Avec la fonction `plot()`, déterminer le comportement de la suite quand n augmente; il est intéressant d'utiliser une échelle logarithmique, c'est-à-dire de considérer $\log(u_n)$ plutôt que u_n .

1.3. Modèle de Fibonacci avec mortalité

Pour insister sur le fait que le comportement exponentiel du modèle de Fibonacci n'est pas dû à l'absence de mortalité, introduisons une modification : le cuniculiculteur retire les lapins de l'enclos à leur sixième mois (juste après qu'ils aient eu des petits). On a maintenant 5 classes d'âges de lapins; l'effectif de la classe d'âge k au mois n sont $E_n^{(k)}$, et leur évolution de mois en mois en donnée par

$$\begin{aligned} E_n^{(1)} &= E_{n-1}^{(2)} + E_{n-1}^{(3)} + E_{n-1}^{(4)} + E_{n-1}^{(5)} \\ E_n^{(2)} &= E_{n-1}^{(1)} \\ E_n^{(3)} &= E_{n-1}^{(2)} \\ E_n^{(4)} &= E_{n-1}^{(3)} \\ E_n^{(5)} &= E_{n-1}^{(4)} \end{aligned}$$

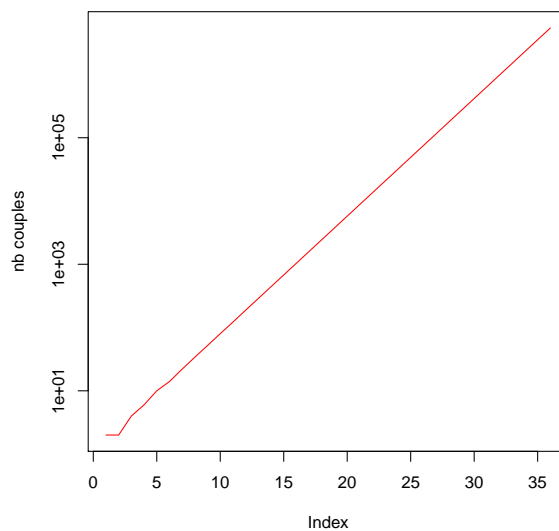
La première équation correspond au nombre de naissances (autant que de lapins dans les classes 2 à 5), et les équations suivantes au vieillissement : à chaque mois, le nombre de lapins dans la classe d'âge k est le nombre de lapins dans la classe d'âge $k-1$ au mois précédent. Il n'y a pas de mortalité avant le 5^e mois, mais au début du 6^e mois tous les lapins sont prélevés. Le code R correspondant est le suivant :

1. Modèles d'évolution en temps discret

```
Leo <- function(n)
{
  E1 <- numeric(n);
  E2 <- numeric(n);
  E3 <- numeric(n);
  E4 <- numeric(n);
  E5 <- numeric(n);
  E1[1] <- 2
  for(i in 2:n)
  {
    # naître
    E1[i] <- E2[i-1]+E3[i-1]+E4[i-1]+E5[i-1];
    # vieillir
    E2[i] <- E1[i-1];
    E3[i] <- E2[i-1];
    E4[i] <- E3[i-1];
    E5[i] <- E4[i-1];
  }
  return( E1+E2+E3+E4+E5 );
}
```

On vérifie que le comportement de la population, une fois écoulés quelques mois, reste la croissance exponentielle.

```
> plot(Leo(36), type="l", col="red", ylog=TRUE, log="y", ylab="nb couples")
```



Exercice : On peut modifier le modèle pour faire intervenir une mortalité de 10% entre les classe d'âge 3 et 4, et 20% entre les classes d'âge 4 et 5. On peu en outre avoir un taux de natalité de 0,8 par lapin et par mois

dans la classe 2, de 1 dans les classes 3, 4, et 5. Les équations d'évolution deviennent

$$\begin{aligned}E_n^{(1)} &= 0,8 \times E_{n-1}^{(2)} + E_{n-1}^{(3)} + E_{n-1}^{(4)} + E_{n-1}^{(5)} \\E_n^{(2)} &= E_{n-1}^{(1)} \\E_n^{(3)} &= E_{n-1}^{(2)} \\E_n^{(4)} &= 0,9 \times E_{n-1}^{(3)} \\E_n^{(5)} &= 0,8 \times E_{n-1}^{(4)}\end{aligned}$$

Modifiez le code en conséquence et observez le comportement du modèle.

1.4. Modèle logistique

Dans le modèle logistique, on fait varier le taux de croissance avec la taille de la population; l'idée sous-jacente est que le milieu peut accueillir de façon optimale une population d'effectif K ; on appelle K la *capacité biotique*. Si l'effectif est inférieur, la population augmente : on a $r > 0$; si l'effectif est supérieur à K , il y a sur-population, et la population diminue : on a $r < 0$.

La façon la plus simple de faire dépendre r de E est de supposer une relation linéaire entre r et E :

$$r = r_0 \left(1 - \frac{E}{K} \right).$$

La relation $E_{n+1} = (1 + r)E_n$ devient donc

$$E_{n+1} = \left(1 + r_0 \left(1 - \frac{E_n}{K} \right) \right) E_n.$$

Exercice Tracer (à main levée) le graphe de r en fonction de E (placer en particulier la valeur quand $E = 0$ et quand $E = K$).

Réaliser une fonction R qui simule l'évolution d'une population suivant ce modèle. On pourra prendre dans un premier temps $r_0 = 0,03$ et $K = 1000$. Faire plusieurs simulations pour comprendre le comportement du modèle.

Observez ensuite ce qui se passe quand on prend des valeurs plus importantes de r_0 : par exemple, $r_0 = 1,8$, $r_0 = 2,2$, $r_0 = 2,5$, $r_0 = 2,8$.

1.5. Proies et prédateurs

On va tenter une modélisation d'une population de lapins et de renard qui partagent le même environnement.

Si on a au temps t , L_t lapins dans la forêt, on suppose que chaque renard parvient à attraper

$$n = \frac{n_{\max} L_t}{L_t + c} \quad (1.2)$$

lapins. On prendra à titre d'exemple les valeurs $n_{\max} = 40$ et $c = 10000$.

1. Modèles d'évolution en temps discret

On suppose que la population de lapins, en l'absence de renard, suivrait une évolution logistique avec une capacité biotique K et un taux de croissance initial r_0 ; il faut soustraire à chaque génération le nombre de lapins qui ont été attrapés par les renards, soit $R_t \times n$:

$$r = r_0 \left(1 - \frac{L_t}{K}\right) \quad (1.3)$$
$$L_{t+1} = (1 + r) \times (L_t - R_t \times n)$$

ou plutôt, pour éviter d'avoir des valeurs négatives,

$$L_{t+1} = \max(0, (1 + r) \times (L_t - R_t \times n)) \quad (1.4)$$

On prendra à titre d'exemple $r_0 = 0,5$ et $K = 20000$.

La population de renards a un taux de croissance ρ qui dépend linéairement du nombre n de lapins attrapés par chaque renard :

$$\rho = \rho_0 \times \left(\frac{n}{a} - 1\right) \quad (1.5)$$

On a alors (en prenant à nouveau un max pour éviter les valeurs négatives)

$$R_{t+1} = \max(0, (1 + \rho) \times R_t). \quad (1.6)$$

On prendra à titre d'exemple $\rho_0 = 0.5$ et $a = 20$.

Écrire une fonction pour simuler l'évolution de ces deux populations : il faut créer un vecteur L et un vecteur R , et les « remplir » en faisant une boucle dans laquelle on calculera (dans le bon ordre) les différentes valeurs données par les équations 1.2, 1.3, 1.4, 1.5, 1.6.

Faire des graphes pour visualiser le comportement du modèle. Que se passe-t-il si des super-prédateurs (les humains !) prélèvent 20% de la population des renards à chaque génération ?

1.6. Simulation d'un processus non déterministe

On s'intéresse ici à un analogue stochastique du modèle de Malthus. On fixe un réel r avec $1 + r > 0$.

Dans le modèle de Malthus, on considère une population à générations séparées dont les effectifs sont donnés par $E_{n+1} = (1 + r)E_n$.

Ici, E_n étant donné, E_{n+1} est tiré selon une loi de Poisson de paramètre $\lambda = (1 + r)E_n$ (l'espérance de E_{n+1} est donc $(1 + r)E_n$), ce qu'on écrit

$$E_{n+1} \sim \mathcal{P}((1 + r)E_n).$$

On peut tirer facilement un entier suivant une loi de Poisson avec `rpois()` ; par exemple, pour tirer un entier selon une loi de Poisson de paramètre $\lambda = 10$:

```
> rpois(n=1, lambda=10)
[1] 8
```

(Lire l'aide de `rpois` pour des détails!)

Écrire un programme qui simule l'évolution d'une telle population pour un nombre de générations donné, et étudier le comportement de ce modèle. (On s'intéressera en particulier aux cas $r < 0$, $r = 0$ et $r > 0$).

1.7. Dérive génétique

Le **modèle de Hardy-Weinberg** considère une population panmictique (c'est-à-dire que les unions se font au hasard) d'individus diploïdes, sans migrations, *de taille infinie* (ou, en pratique, de très grande taille), sans mutations, sans sélection gamétique, à générations séparées.

Dans ces conditions, considérons un locus di-allélique A/a , et notons p_n la fréquence de l'allèle a à la génération n . Le génome d'un individu de la génération $n+1$ est composé de deux gamètes issus de la génération n . Chacun de ces gamètes porte l'allèle a avec probabilité p_n : la fréquence de l'allèle a à la génération $n+1$ est donc $p_{n+1} = p_n$.

Dans ce modèle, les fréquences alléliques restent constantes au fil des générations.

Le **modèle de Wright-Fisher** est presque identique au modèle de Hardy-Weinberg : on y abandonne simplement l'hypothèse de la taille infinie. On considère donc une population de taille N , constante au fil des générations.

Pour comprendre l'évolution des fréquences alléliques, on va s'intéresser au nombre d'exemplaires E_n de l'allèle a dans le génome des individus de la génération n . Les individus étant diploïdes, on a $0 \leq E_n \leq 2N$; la fréquence de l'allèle a à la génération n est $p_n = E_n/2N$.

Quelle est la valeur de E_{n+1} ? On ne peut la donner de façon déterministe : il y a $2N$ locus A/a dans la génération $n+1$, chacun provenant d'un gamète émis par la génération n , et donc porteur de a avec probabilité $p_n = E_n/2N$; E_{n+1} est donc une variable aléatoire qui suit une loi binomiale de paramètres $2N$, $E_n/2N$, soit

$$E_{n+1} \sim \text{Bin}\left(2N, \frac{E_n}{2N}\right).$$

Ainsi, la fréquence de l'allèle a va changer au fil des générations ; il est possible qu'au fil du temps elle atteigne la valeur 0 ou la valeur 1, auquel cas elle va ensuite rester constante : un des deux allèles a disparu ; on dit que l'allèle restant s'est *fixé*.

On peut simuler facilement une variable aléatoire qui suit une loi binomiale avec `rbinom()`, ici avec paramètres $2N$ et $E/2N$:

```
> N <- 100
> E <- 38
> rbinom(1, 2*N, E/(2*N))
[1] 41
```

Réaliser un programme qui simule l'évolution du nombre d'exemplaires E_n de l'allèle a (suivant Wright-Fisher) au cours du temps. Faire plusieurs simulations, en essayant différentes valeurs de départ pour N et E_1 . Faire des graphiques. Qu'observez-vous ?

Pistes pour plus de travail Pour des conditions initiales données, estimer la probabilité de fixation après $N = 100$ ou $N = 1000$ générations. Estimer la distribution du temps de fixation.

Modifier le modèle en faisant varier N au fil du temps, en introduisant une possibilité de mutation, etc.

1.8. Sélection

Dans le modèle de Hardy-Weinberg on considère une population idéale, sous des hypothèses simplificatrices (panmixie, pangamie, absence de sélection, de mutation, de migration, population infinie, générations séparées...). En considérant un locus di-allélique A/a , ce modèle permet de montrer que les fréquences p, q de

1. Modèles d'évolution en temps discret

ces allèles sont constantes au fil des générations; on montre également qu'un individu tiré au hasard dans la population est de génotype AA, Aa, et aa avec probabilité respective p^2 , $2pq$, q^2 .

On va s'intéresser aux conséquences de l'abandon d'une des hypothèses du modèle de Hardy-Weinberg : l'absence de sélection. On conserve les autres hypothèses, notamment la panmixie, l'absence de sélection gamétique, les générations séparées, et la population de taille infinie.

Dans ce modèle, les individus qui composent une génération donnée ont des chances inégales de transmettre leurs gamètes à la génération suivante; ceci peut être dû à des différences de mortalité avant l'âge reproductif, ou encore à une fécondité réduite.

Valeur sélective

On considère un locus di-allélique d'allèles A et a, « soumis à sélection », c'est-à-dire que la fécondité des individus dépend de leur génotype.

On notera p_n la fréquence de l'allèle A **pour la génération n** (disons, pour pallier toute ambiguïté due à la possibilité d'une mortalité dépendant du génotype, à la naissance). On notera par commodité $q_n = 1 - p_n$ la fréquence de l'allèle a.

Notons r, s et t les valeurs sélectives des trois génotypes :

Génotype :	AA	Aa	aa
Valeur sélective :	r	s	t

On pourrait définir les valeurs sélectives comme la probabilité qu'un gamète émis par un individu de génotype donné soit présent à la génération suivante; en fait, il suffit que r, s et t soient proportionnels à ce nombre; seules les valeurs sélectives relatives importent, c'est-à-dire les proportions $r : s : t$. En pratique, r, s et t peuvent prendre n'importe quelle valeur supérieure ou égale à 0.

En l'absence de sélection, on a $r = s = t$.

Évolution au fil des générations

En utilisant le modèle de l'urne gamétique, on montre que la fréquence de l'allèle A à la génération $n + 1$ est

$$p_{n+1} = \frac{p_n^2 r + p_n q_n s}{p_n^2 r + 2p_n q_n s + q_n^2 t}.$$

ou encore $p_{n+1} = f(p_n)$, où $f(x) = \frac{x^2 r + x(1-x)s}{x^2 r + 2x(1-x)s + (1-x)^2 t}$.

Créer une fonction `evolution(n, p1, r, s, t)` qui calcule les fréquences p_1, \dots, p_n , les paramètres étant la valeur initiale p_1 , et les valeurs sélectives r, s , et t . Créer un graphe donnant l'évolution de p pour $n = 40$ générations, avec comme valeur de départ $p_1 = 0.4$, et $r = 1, s = 1, t = 2$.

Quel est le comportement de la suite des fréquences (p_n) quand $r = s = t$?

Et quand $r = s = 1$ et $t = 0$? À quel cas concret cela vous semble-t-il correspondre?

Déterminer le comportement de la suite des fréquences (p_n) quand $r > s > t > 0$ (par exemple $r = 3, s = 2, t = 1$); quand $r > s > 0$ et $s < t$; et quand $r < s$ et $s > t > 0$.

2. Modèles d'évolution en temps continu

Introduction

Les modèles en temps continu sont à distinguer des modèles d'évolution en temps discret, qui sont adaptés à des populations se reproduisant à intervalles de temps réguliers (reproduction annuelle, ou bisannuelle)...

Dans les modèles en temps discret, on donne E_{n+1} , l'effectif de la population au temps $n + 1$, en fonction de E_n ; ou, de façon équivalente, on donne le taux d'accroissement (relatif) entre n et $n + 1$,

$$r = \frac{E_{n+1} - E_n}{E_n}$$

qui peut dépendre de n et de E_n . Le modèle le plus simple est le modèle de Malthus qui prend r constant.

L'analogue continu du modèle de Malthus postule que le taux d'accroissement instantané (relatif à la taille de la population) de la population,

$$r = \frac{E'(t)}{E(t)},$$

est constant.

L'accroissement $E_{n+1} - E_n$ a été remplacé par la dérivée $E'(t)$. On parle d'une équation différentielle.

2.1. Préliminaire : la notion de dérivée

Soit f une fonction définie sur un intervalle $I \subset \mathbb{R}$ (une fonction du temps ou de l'espace...). Soit t un point de I .

On dit que f est dérivable en t si

$$\lim_{h \rightarrow 0} \frac{f(t+h) - f(t)}{h} \text{ existe (et est finie)}$$

On note cette limite $f'(t)$.

C'est la pente de la tangente au graphe de f en t ; on peut également l'interpréter comme un *taux d'accroissement instantané* en t .

Si $f(t)$ est une grandeur dimensionnée, par exemple une biomasse exprimée en gramme, et que t est un temps exprimé en heure, alors $f'(t)$ est en grammes par seconde...

Notons que $\frac{f(t+h) - f(t)}{h}$ est la pente de la corde qui joint $(t, f(t))$ et $(t+h, f(t+h))$; c'est le taux d'accroissement de f entre t et $t+h$.

Lorsque f est dérivable en tous les points de l'intervalle I , on note f' la fonction qui à $t \in I$ associe $f'(t)$.

On peut ensuite définir les dérivées successives de f , sa dérivée seconde f'' étant la dérivée de f' , etc.

2. Modèles d'évolution en temps continu

2.1.1. Approximations de la dérivée

Pour calculer la dérivée des fonctions usuelles il y a des formules de dérivation. Si on n'est pas fort en dérivation, on peut essayer un logiciel de calcul formel (par exemple `maple`; comme logiciel libre il y a `yacas`).

Cependant il y a des cas où on ne peut pas faire ça, par exemple parce qu'on n'a accès qu'aux valeurs de la fonction, à travers un calcul trop complexe pour qu'on sache écrire une formule exploitable, ou à travers un processus de mesure par exemple. C'est une des motivations possibles pour s'intéresser aux *approximations numériques* d'une dérivée.

Différences finies

On fixe dans toute cette section $h \in \mathbb{R}$ (typiquement, h sera « petit », par exemple $h = 0,1$ ou $h = 0,001$). On définit la *différence finie en avant* de f , $\Delta f(x)$, par

$$\Delta f(x) = \frac{f(x+h) - f(x)}{h}.$$

C'est la pente de la corde qui joint $(x, f(x))$ et $(x+h, f(x+h))$.

La *différence finie en arrière* est

$$\nabla f(x) = \frac{f(x) - f(x-h)}{h}.$$

C'est la pente de la corde qui joint $(x-h, f(x-h))$ et $(x, f(x))$.

Si h est petit, $\Delta f(x)$ et $\nabla f(x)$ sont des approximations de $f'(x)$.

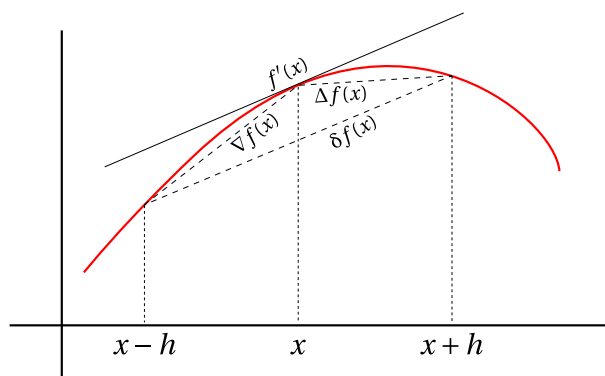


FIGURE 2.1. – Différence centrée

Sur la figure 2.1 on voit que dans les cas typiques l'une des deux valeurs $\Delta f(x)$ et $\nabla f(x)$ est une approximation par excès (supérieure à la vraie valeur), et l'autre est une approximation par défaut (inférieure à la vraie valeur).

L'idée de prendre la moyenne des deux paraît alors naturelle.

On pose

$$\begin{aligned} \delta f(x) &= \frac{\Delta f(x) + \nabla f(x)}{2} \\ &= \frac{f(x+h) - f(x-h)}{2h} \end{aligned}$$

On appelle $\delta f(x)$ la *différence centrée* en x .

On voit que c'est également la pente de la corde qui joint $(x-h, f(x-h))$ et $(x+h, f(x+h))$. En s'appuyant sur le dessin, on a à nouveau l'intuition que cette approximation est de meilleure qualité que $\Delta f(x)$ et $\nabla f(x)$.

Peut-on confirmer cette intuition en estimant la qualité de ces approximations ?

Exercice Écrire des fonctions R pour calculer l'approximation de la dérivée de $f(x) = \exp(x)$ pour chacune des méthodes ; plus précisément, une fonction `dexp(a, h)` calculera une approximation de $f'(a)$ en un point a , pour une valeur de h donnée. En calculant la valeur de $f'(0)$ pour diverses valeurs de h , tracer le graphe de l'erreur commise en fonction de h , pour chaque méthode.

Exercice Écrire une fonction `Delta(F, x, h)` qui calcule une approximation de la dérivée de la fonction F selon la méthode des différences finies en avant.

2.1.2. Qualité des approximations de la dérivée

On peut montrer le résultat suivant :

Théorème 1 Soit f une fonction d'une variable réelle.

— Si f est deux fois dérivable et si $|f''(c)| \leq M_2$ pour tout $c \in]x_0, x_0 + h[$, alors

$$|\Delta f(x_0) - f'(x_0)| \leq \frac{1}{2} M_2 h.$$

— Si f est trois fois dérivable et si $|f'''(c)| \leq M_3$ pour tout $c \in]x_0 - h, x_0 + h[$, alors

$$|\delta f(x_0) - f'(x_0)| \leq \frac{1}{6} M_3 h^2.$$

On peut résumer les résultats ainsi :

Si la fonction est trois fois dérivable, l'ordre de grandeur de l'erreur commise par Δf est h ; si on essaie d'améliorer l'approximation en choisissant un « pas de temps deux fois plus petit » $h' = \frac{1}{2}h$, on s'attend à ce que l'erreur soit divisée par deux. On dit également qu'on a une convergence linéaire en h .

Par contre, l'ordre de grandeur de l'erreur commise par δf est h^2 ; si on essaie d'améliorer l'approximation en choisissant $h' = \frac{1}{2}h$, on s'attend cette fois-ci à ce que l'erreur soit divisée par quatre. On dit également qu'on a une convergence quadratique (ou d'ordre 2) en h .

Ainsi même si on ne connaît pas les valeurs des constantes M_2 et M_3 , on sait que dès que h sera assez petit l'approximation par la différence centrée δf est meilleure que l'approximation par la différence en avant Δf : si $h \ll 1$ alors $h^2 \ll h$.

2.2. Le modèle de Malthus en temps continu

La modélisation d'une population bactérienne se divisant de façon asynchrone conduit, en première approche, à l'équation différentielle suivante :

$$E'(t) = r \cdot E(t). \quad (2.1)$$

Ici l'effectif $E(t)$ est une fonction du temps. À chaque instant t , l'effectif $E(t)$ de la population bactérienne croît de façon proportionnelle à $E(t)$; comme dans le cas discret, le coefficient r tient compte à la fois de la mortalité et de la division ; ainsi, un choix de $r < 0$ signifie qu'il y a plus de cellules qui meurent que de cellules qui se divisent.

2. Modèles d'évolution en temps continu

La solution de cette équation différentielle est

$$E(t) = E_0 \exp(rt) \quad (2.2)$$

c'est-à-dire une croissance exponentielle. Ce modèle ne peut être valide que pour un temps limité, le milieu de croissance n'ayant qu'une capacité limitée.

Notons qu'on a

$$\begin{aligned} \frac{E(t+1)}{E(t)} &= \frac{\exp(r(t+1))}{\exp(rt)} \\ &= \exp(r) \\ &\simeq 1+r \end{aligned}$$

Cette dernière approximation étant valable quand r est petit; ceci souligne l'analogie avec le cas discret.

2.2.1. Résolution approchée (schéma d'Euler)

La méthode qui suit est très générale, et s'appliquera à toutes les équations différentielles.

On suppose l'effectif $E(0) = E_0$ au temps $t = 0$ connu. Pour obtenir une approximation de $E(t)$ entre 0 et T , on va se contenter de calculer $u_k = E(k \cdot h)$ pour $k = 0, 1, 2, \dots$. Pour ce faire, on va transformer l'équation différentielle 2.1 en équation discrète, on remplaçant la dérivée $E'(t)$ par son approximation par la différence finie $\frac{1}{h}(E(t+h) - E(t))$.

On obtient alors

$$\begin{aligned} \frac{E(t+h) - E(t)}{h} &\simeq rE(t) \\ E(t+h) &\simeq E(t) + hrE(t) \\ E(t+h) &\simeq (1 + hr)E(t) \end{aligned}$$

On appliquant cette dernière équation à $t = kh$, on en déduit

$$u_{k+1} = (1 + hr)u_k$$

Exercice Écrire une fonction qui calcule la suite u_k des valeurs de $u_0 = E_0$ et de h données. Tracer le graphe de la solution approchée et comparer avec la solution exacte 2.2.

2.3. Modèle logistique continu

On fait dépendre r de $E(t)$, la taille de la population. Comme dans le modèle discret, on pose $r = r_0 \left(1 - \frac{E(t)}{K}\right)$. L'équation différentielle devient

$$E'(t) = r_0 \cdot y \cdot \left(1 - \frac{E(t)}{K}\right) E(t)$$

Résoudre numériquement cette équation avec le schéma d'Euler.

2.4. Fonctions implémentant le schéma d'Euler

On vous fournira en cours un fichier contenant trois fonctions `Euler.pc`, `Euler.pm` et `Euler.pv`. Ces trois fonctions prennent les arguments suivants :

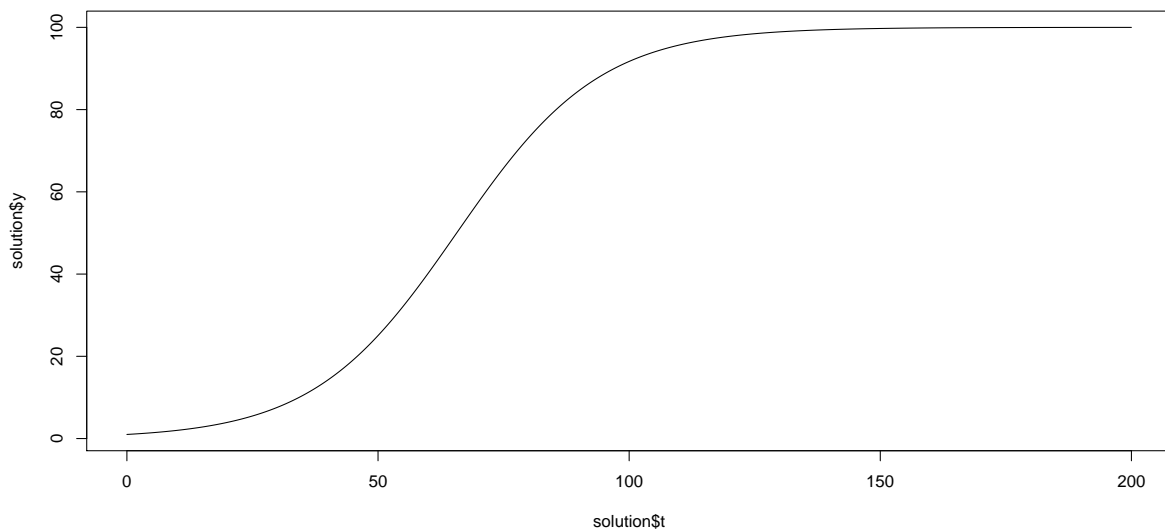
- `F` : une fonction $F(t, y, \dots)$ où t est le temps, $y(t)$ est un vecteur de longueur n correspondant à $y(t)$; la fonction renvoie la valeur de $y'(t)$
- `y0` : un vecteur de longueur n correspondant à $y(t_0)$
- `t0` : le temps t_0 où commence l'intégration (valeur par défaut $t_0 = 0$)
- `T` : le temps T durant lequel on désire intégrer l'équation différentielle
- `h` : le pas d'intégration ((valeur par défaut $h = 0,01$))
- `...` : paramètres additionnels à passer à la fonction `F`

La fonction `Euler.pv` admet également un argument `Nmax`, qui est le nombre maximal d'itérations à réaliser (valeur par défaut 50000).

Le résultat de cette fonction est une liste à deux composantes, une composante `$t` qui donne les temps t_0, t_1, \dots où la solution a été calculée; et une composante `$y` qui donne les valeurs de $y(t_0), y(t_1), \dots$. Selon que la dimension n de `y0` est égale à 1 ou plus grande, `$y` est un vecteur ou une matrice à n lignes.

On pourra par exemple s'en servir pour résoudre l'exercice précédent ainsi :

```
F.logit <- function(t, y, r0, K)
{
  return( r0*(1-y/K)*y )
}
solution <- Euler.pc(F.logit, 1, T = 200, r0 = 0.07, K = 100)
plot(solution$t, solution$y, type="l")
```



2.5. Systèmes d'équations différentielles

Prenons comme exemple ce système d'équations.

$$\begin{cases} x'(t) = y(t) \\ y'(t) = -x(t) \end{cases}$$

Le remplacement des dérivées par leur approximation conduit à

$$\begin{cases} x(t+h) = x(t) + h \cdot y(t) \\ y(t+h) = y(t) - h \cdot x(t) \end{cases}$$

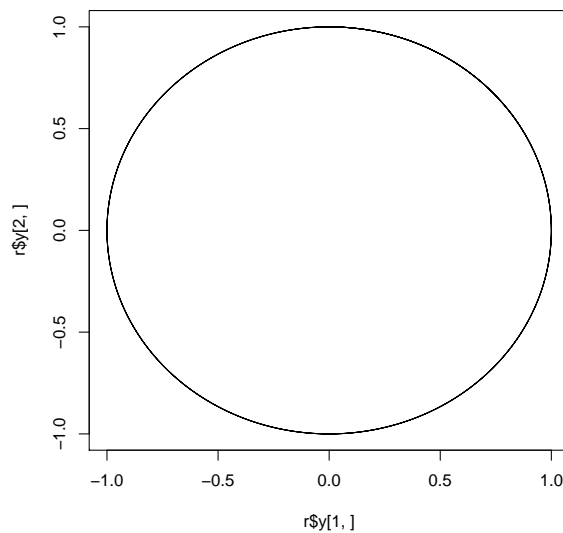
et en posant $u_k = x(kh)$, $v_k = y(kh)$, on a

$$\begin{cases} u_{k+1} = u_k + h \cdot v_k \\ v_{k+1} = v_k - h \cdot u_k \end{cases}$$

Exercice : Réaliser une fonction R pour résoudre numériquement ce système, et faire le graphe des points $x(t), y(t)$ pour t entre 0 et 20 (tenter de prendre $h = 0,1$ puis $h = 0,001$).

Alternativement, avec la fonction `Euler.pv()` fournie en cours, on pourra procéder ainsi :

```
f <- function(t, y)
{
  return( c(y[2], -y[1]));
}
r <- Euler.pv(f, c(0,1), T=20)
plot( r$y[1,], r$y[2,], type="l" )
```



2.6. Pendule oscillant

L'angle θ que fait un pendule avec la verticale suit l'équation différentielle

$$\theta'' + k \sin(\theta) = 0.$$

On va poser $y_1(t) = \theta(t)$ et $y_2(t) = \theta'(t)$. L'équation différentielle ci-dessus se ré-écrit alors (n'écrivons plus la variable t à chaque fois)

$$\begin{cases} y_1' = y_2 \\ y_2' = -k \sin(y_1) \end{cases}$$

Pour résoudre ce système avec `Euler.pm()`, et faire le grahe de la solution, on procédera (par exemple) comme suit :

```
f.pendule <- function(t, y, k)
{
  y1 <- y[1];
  y2 <- y[2];
  dy1 <- y2;
  dy2 <- -k*sin(y1);
  return( c(dy1,dy2) );
}
r <- Euler.pm(f.pendule, c(pi/4,0), T = 20, k = 2)
plot(r$t, r$y[1,], type="l" )
```

La fonction `f.pendule` a été écrite de façon à bien faire apparaître le système d'équation différentielle. Une écriture plus condensée est bien sûr possible.

2.7. Réaction chimique « classique »

On considère une réaction chimique $A + B \xrightarrow{k} C$ La petite constante k sur la flèche donne la vitesse de la réaction : si on note $a(t)$, $b(t)$ et $c(t)$ la concentration de A, B et C à l'instant t , on a à chaque instant une quantité $k \cdot a(t) \cdot b(t)$ de réactifs A et B qui réagissent, ce qui se traduit par

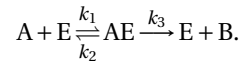
$$\begin{cases} a' &= -k \cdot a \cdot b \\ b' &= -k \cdot a \cdot b \\ c' &= k \cdot a \cdot b \end{cases}$$

Il s'agit du modèle le plus simple, on peut être amené à remplacer le produit $a(t)b(t)$ par $a(t)^\alpha b(t)^\beta$, par exemple.

La résolution est laissée en exercice (conditions initiales, $a(0) = 1, b(0) = 2, c(0) = 0$; on prendra $k = 0.01$ par exemple).

2.8. Réaction chimique enzymatique : cinétique de Michaelis-Menten

On considère une réaction $A \longrightarrow B$ avec catalyse enzymatique, c'est-à-dire



En notant $a(t)$, $e(t)$, $c(t)$, et $b(t)$ les concentrations de A, E, du complexe AE et de B, on a les équations d'évolution suivantes :

$$\begin{cases} a' &= -k_1 \cdot a \cdot e + k_2 \cdot c \\ e' &= -k_1 \cdot a \cdot e + k_2 \cdot c + k_3 \cdot c \\ c' &= k_1 \cdot a \cdot e - k_2 \cdot c - k_3 \cdot c \\ b' &= k_3 \cdot c \end{cases}$$

2.8.1. Traitement classique

Ce système étant complexe, fait des simplifications pour relier directement $b'(t)$ à $a(t)$, pour « court-circuiter » l'enzyme.

En pratique, la concentration en enzyme (E et AE) est très petite comparée à celle du substrat A. Dans ces conditions, après une courte phase de démarrage de la réaction, tous les sites enzymatiques ou presque seront occupés; on va supposer que $c(t)$ est constante, c'est-à-dire $c' = 0$.

On vérifie avec les équations ci-dessus que $e' + c' = 0$, ce qui traduit le fait que la quantité d'enzyme lié à A ou non est constante. On prend des conditions initiales $e(0) = e_0$ et $c(0) = 0$, et on a $e(t) + c(t) = e_0$, d'où $e = e_0 - c$.

On a donc

$$\begin{aligned} k_1 \cdot a \cdot e - k_2 \cdot c - k_3 \cdot c &= 0 \\ k_1 \cdot a \cdot (e_0 - c) &= (k_2 + k_3) \cdot c \\ a \cdot e_0 &= \left(a + \frac{k_2 + k_3}{k_1}\right) c \end{aligned}$$

En posant $K_M = \frac{k_2 + k_3}{k_1}$, on a $c = \frac{a \cdot e_0}{K_M + a}$, et $b' = k_3 \cdot c = (k_3 \cdot e_0) \frac{a}{K_M + a}$.

On voit que la réaction a une vitesse maximale $k_3 e_0$, dont on se rapproche quand la concentration en substrat A est importante.

2.8.2. Traitement numérique

Il faudra ici utiliser la fonction `Euler.pv()` qui se comportera mieux que `Euler.pm()` sur ce problème particulier.

```
f.mm <- function(t, y, k = c(1,0.08,10))
{
  a <- y[1];
  e <- y[2];
  c <- y[3];
  b <- y[4];
  da <- -k[1]*a*e + k[2]*c;
```

```

de <- -k[1]*a*e + k[2]*c + k[3]*c;
dc <- k[1]*a*e - k[2]*c - k[3]*c;
db <- k[3]*c;
return(c(da,de,dc,db));
}
r <- Euler.pv(f.mm, c(100,10,0,0), T = 2)

```

Le tracé d'une solution et son interprétation seront faits en cours.

2.9. La loi du minimum de Liebig

C'est la « brique de base » de nombreux modèles en écologie (à l'origine, en agronomie). On va considérer le problème de l'évolution d'une population d'algues unicellulaires.

2.9.1. Modèle

On note y la concentration des algues. N et P sont les concentrations en nutriments (azote et phosphore). Le taux de reproduction des algues est α , et le taux de mortalité est β , de sorte qu'on a

$$\frac{dy}{dt} = \alpha y - \beta y = (\alpha - \beta)y. \quad (2.3)$$

Le taux de mortalité sera ici supposé constant. Le taux de natalité dépend de la concentration en nutriments, selon le principe que c'est l'élément le plus rare qui limite la croissance.

$$\alpha = \min\left(\frac{\alpha_N N}{k_N + N}, \frac{\alpha_P P}{k_P + P}, \alpha_{\max}\right) \quad (2.4)$$

La concentration en nutriments varie selon les équations

$$\begin{cases} \frac{dN}{dt} = -(\alpha - \beta)c_N y \\ \frac{dP}{dt} = -(\alpha - \beta)c_P y \end{cases} \quad (2.5)$$

Interprétation du modèle

L'équation 2.4 traduit l'idée que la vitesse de croissance est celle donnée par le facteur en quantité minimale (par rapport à ses besoins). Le terme

$$\frac{\alpha_P P}{k_P + P}$$

traduit la consommation du phosphore et on a un terme analogue pour l'azote. On reconnaît des termes analogues aux vitesses de réactions enzymatiques dans le modèle de Michaelis-Menten; ces termes sont appropriés pour modéliser la façon dont les algues assimilent les éléments.

Les équations 2.5 traduisent le fait que le processus de naissance des algues consomme des nutriments, tandis que le processus de mort en libère. Les constantes c_N et c_P sont reliées à la quantité d'azote (N) et de phosphore (P) contenue dans les algues.

2. Modèles d'évolution en temps continu

Milieu	N	P
Pauvre	5	1
Riche	1000	100

TABLE 2.1. – Concentrations en azote et phosphore dans des milieux riches ou pauvres (en $\mu\text{g/l}$)

À saturation en phosphore (terme P), on a

$$\frac{\alpha_P P}{k_P + P} \simeq \alpha_P;$$

quand au contraire le phosphore est rare ($P \ll k_P$), on a

$$\frac{\alpha_P P}{k_P + P} \simeq \frac{\alpha_P}{k_P} P.$$

Le terme k_P est la demi-saturation : quand $P = k_P$, on a $\frac{\alpha_P P}{k_P + P} = \frac{1}{2} \alpha_P$.

On suppose que ces deux nutriments sont les seuls facteurs limitants, les autres étant supposés disponibles en abondance. Si par exemple P est très abondant également, et qu'on peut supposer que les variations de sa concentration ont un effet négligeable, de sorte qu'on aura toujours

$$\frac{\alpha_P P}{k_P + P} \simeq \alpha_P,$$

on peut simplifier cette équation en remplaçant α_{\max} par

$$\alpha'_{\max} = \min(\alpha_P, \alpha_{\max}).$$

On a alors

$$\alpha = \min\left(\frac{\alpha_N N}{k_N + N}, \alpha'_{\max}\right)$$

On suppose souvent que le phosphore et l'azote sont les éléments limitants, les autres éléments nécessaires à la croissance des algues étant présents en abondance.

On peut au contraire si c'est nécessaire ajouter d'autres termes pour prendre en compte d'autres éléments (autres nutriments, lumière).

2.9.2. Ordre de grandeur des constantes

On écrira les concentrations en $\mu\text{g/l}$.

Les proportions d'azote et le phosphore dans les algues sont typiquement $c_N = 0,7$ et $c_P = 0,08$.

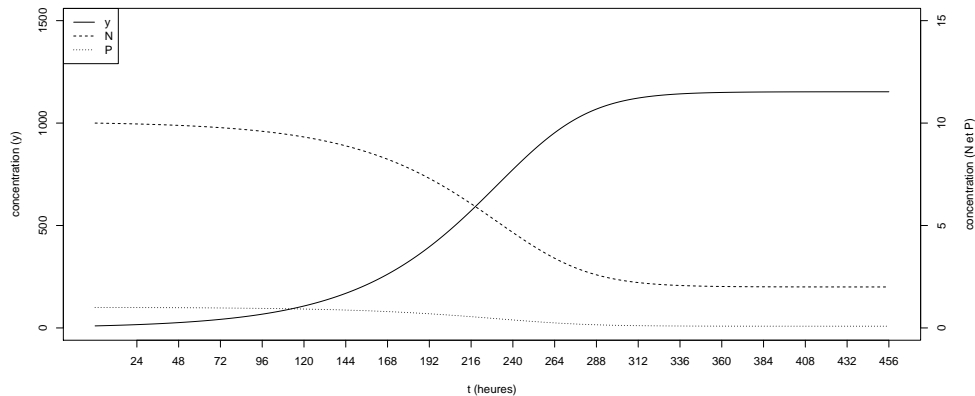
Certaines algues peuvent avoir une croissance très rapide, α pouvant atteindre des valeurs aussi haute que $0,05 \text{ h}^{-1}$, ce qui correspond, si il n'y a pas de facteur limitant et en négligeant la mortalité, à une croissance exponentielle $y(t) = \exp(0,05t)y(0)$, et donc à plus que tripler la biomasse en 24 h. On pourra essayer pour α_{\max} , et également α_P et α_N , des valeurs entre 0,01 et 0,05, tout en gardant une mortalité faible, par exemple $\beta = 0,005$.

Les demi-saturations pourront varier en 1 et 100 $\mu\text{g/l}$ selon les espèces. Par exemple, pour l'algue *Microcystis*, on a $k_P = 6 \mu\text{g/l}$ et $k_N = 60 \mu\text{g/l}$.

2.9.3. Un exemple

On prend les conditions initiales $y(0) = 10$, $N(0) = 100$, $P(0) = 10$; les constantes valent $\alpha_{\max} = 0,04$, $\beta = 0,005$, $\alpha_N = 0,045$, $\alpha_P = 0,04$, $k_N = 60$, $k_P = 6$, $c_N = 0,07$ et $c_P = 0,008$.

Après intégration numérique on doit obtenir quelque chose de proche de la figure suivante.



2.10. Le retour du chaos : les équations de Lorenz

Lorenz a découvert en étudiant de façon numérique des systèmes d'équations différentielles destinées à modéliser les phénomènes météo, que certaines équations étaient instables. Comme dans le cas de l'équation logistique discrète (avec r_0 proche de 3), une petite différence sur les conditions initiales est vite amplifiée, et le système en devient imprévisible.

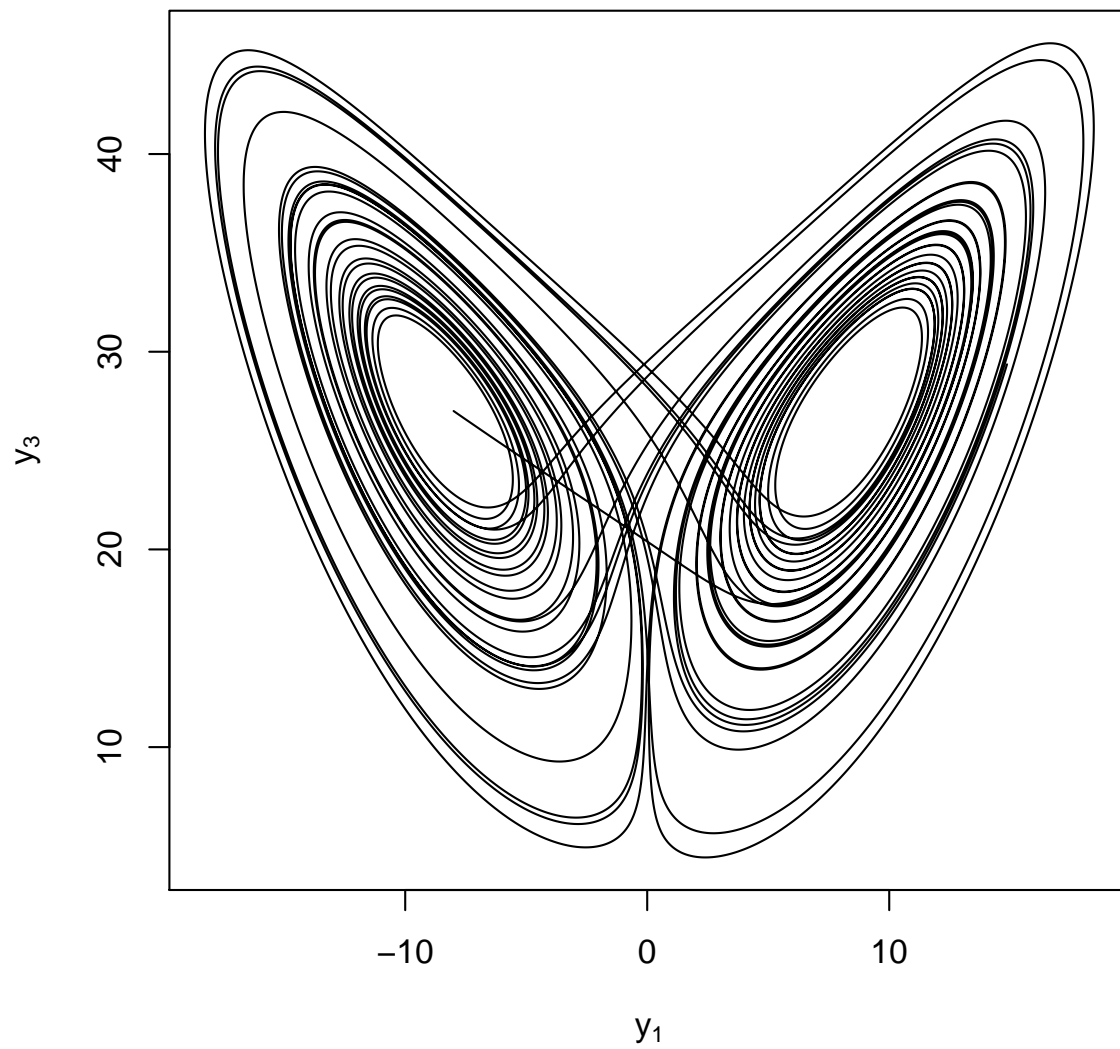
Le système d'équation suivant est issu de ses travaux :

$$\begin{cases} y_1' = 10(y_2 - y_1) \\ y_2' = -y_1 y_3 + 28 y_1 - y_2 \\ y_3' = y_1 y_2 - \frac{8}{3} y_3 \end{cases}$$

Sans commentaires, la résolution numérique :

```
f.lorenz <- function(t,y)
{
  dy1 <- 10*(y[2]-y[1])
  dy2 <- -y[1]*y[3]+28*y[1]-y[2]
  dy3 <- y[1]*y[2]-8*y[3]/3
  return( c(dy1,dy2,dy3) );
}
r <- Euler.pm(f.lorenz, c(-8,8,27), T = 40, h=1e-3)
plot(r$y[1,], r$y[3,], type="l", xlab = expression(y[1]), ylab = expression(y[3])) )
```

2. Modèles d'évolution en temps continu



A. Schémas de résolution numérique d'équations différentielles

On va considérer deux méthodes, le schéma d'Euler et le schéma du point milieu. Elles seront présentées dans le cas de la dimension 1, mais la généralisation au cas des systèmes d'équations différentielles est facile.

On suppose donc donnée une équation différentielle,

$$y' = f(t, y),$$

et une condition initiale $y(t_0) = c_0$.

On va chercher une fonction $y(t)$ qui soit une approximation de la solution de cette équation différentielle, définie pour $t \leq t_0$. On se contentera des valeurs y_0, y_1, y_2 , etc, aux points $t_0, t_1 = t_0 + h, t_2 = t_1 + h = t_0 + 2h$, etc, où h est un petit *pas de temps* de valeur fixée à l'avance.

Il suffit de savoir passer de la valeur de $y(t)$ à celle de $y(t + h)$; on connaît $y_0 = y(t_0)$, on commencera donc par calculer $y_1 = y(t_1) = y(t_0 + h)$, puis $y_2 = y(t_1 + h)$, etc.

Pour assurer que les méthodes présentées fonctionnent bien, il faut faire une petite supposition technique sur la fonction f : on suppose qu'il existe une constante L telle qu'on ait toujours $|f(t, a) - f(t, b)| < L|a - b|$: la vitesse à laquelle la fonction varie quand on fait varier la seconde variable est bornée. En pratique on ne se souciera pas de cette condition.

A.1. Schéma d'Euler

Dans l'équation $y'(t) = f(t, y(t))$, on remplace $y'(t)$ par son approximation $\Delta y(t) = \frac{y(t+h) - y(t)}{h}$. On a donc l'approximation

$$y(t+h) \simeq y(t) + h \cdot f(t, y(t)).$$

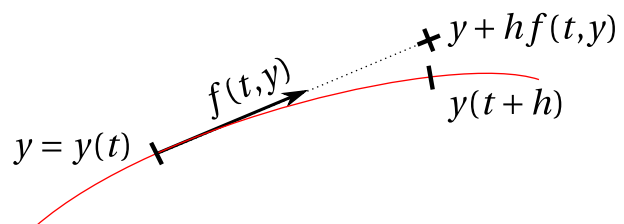


FIGURE 1.1. – Schéma d'Euler

On a donc ramené l'équation différentielle au calcul des suites :

$$\begin{cases} t_{k+1} &= t_k + h \\ y_{k+1} &= y_k + hf(t_k, y_k) \end{cases}$$

Quelle est l'erreur commise par la méthode?

Théorème 2 *L'erreur commise par le schéma d'Euler au temps $t_0 + T$ est plus petite que $k \cdot \exp(\alpha \cdot T) \cdot h$ (où k, α sont des constantes qu'on n'explicite pas).*

A. Schémas de résolution numérique d'équations différentielles

Les erreurs s'accumulent de pas en pas : c'est ce que traduit le terme en $\exp(\alpha T)$. Cependant, plus le pas est petit, meilleure est l'approximation ; la convergence est linéaire en h , c'est-à-dire que si on divise h par 10 on divisera l'erreur par 10.

Exercice Réaliser un programme R qui applique cette méthode à l'équation différentielle $y' = \mu_0 \cdot y \cdot (1 - \frac{y}{K})$. Explorer le comportement du modèle ; comparer avec le cas discret.

A.2. Schéma du point milieu

Le schéma du point milieu donne de meilleures approximations que le schéma d'Euler. L'idée de base est la même que celle qui a été mise en œuvre pour les dérivées : la différence centrée $\delta f(x)$ donne de meilleures approximations de $f'(x)$ que la différence finie en avant, parce que la corde qui passe par les points d'abscisses $x-h$ et $x+h$ est « presque parallèle » à la tangente.

Ici on connaît bien la dérivée en $y(t)$, c'est $f(t, y(t))$; ce qu'on cherche, c'est la pente de la corde qui joint les points d'abscisses t et $t+h$. On va l'approcher par la tangente au point d'abscisse $t + \frac{h}{2}$.

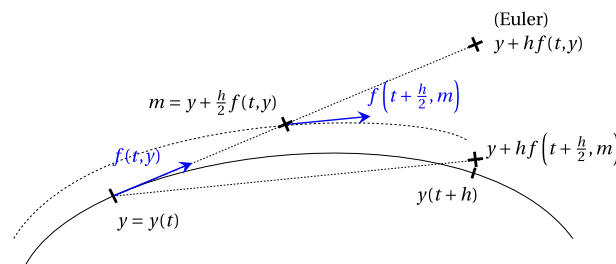


FIGURE 1.2. – Schéma du point milieu

On commence par « un demi-pas d'Euler » : on calcule

$$m = y(t) + \frac{h}{2} f(t, y(t)).$$

C'est une approximation de $y(t + \frac{h}{2})$. La dérivée en ce point est donc donnée par $f(t + \frac{h}{2}, m)$. On prend l'approximation

$$y(t+h) \simeq y(t) + hf(t + \frac{h}{2}, m).$$

On a donc la relation de récurrence suivante pour les valeurs des t_k et des y_k :

$$\begin{cases} t_{k+1} &= t_k + h \\ y_{k+1} &= y_k + hf(t_k + \frac{h}{2}, y_k + \frac{h}{2}f(t_k, y_k)) \end{cases}$$

Quelle est l'erreur commise par cette nouvelle méthode ?

Théorème 3 L'erreur commise par le schéma du point milieu au temps $t_0 + T$ est plus petite que $k \cdot \exp(\alpha \cdot T) \cdot h^2$ (où k, α sont des constantes).

Cette fois ci, la convergence est *quadratique* en h : si on divise h par 10, on divise l'erreur par 100. On s'attend donc à ce que cette méthode fonctionne mieux en général.

A.3. Méthode d'Euler à pas variable

Quand on intègre une équation différentielle (ou un système) « raide », c'est-à-dire où la valeur de $f(t, y)$ est très grande en certains points, la méthode d'Euler peut rencontrer des problèmes.

Exemple d'équation différentielle raide Considérons l'équation suivante.

$$y' = \frac{1}{y^2 + 0.001}, \quad y(0) = 0.$$

Au point initial $t = 0$, $y = 0$, la valeur de $f(y, t) = \frac{1}{y^2 + 0.001}$ est égale à 1000!

Si on utilise la méthode d'Euler avec un pas de temps raisonnable comme $h = 0,01$, on commence à $t_0 = 0$ et $y_0 = 0$; dès la première itération on calcule $y_1 = y_0 + h * f(t_0, y_0) = 0 + 0,01 * 1000 = 10$ ce qui n'est clairement pas raisonnable.

Une solution est de prendre h beaucoup plus petit, par exemple $h = 10^{-5}$, mais quand la valeur de $y(t)$ s'est éloignée de 0, par exemple pour $y(t)$ voisin de 1 (on a alors également $f(t, y) \simeq 1$) ce pas très petit ralentit inutilement le calcul.

Une solution simple à mettre en œuvre est de faire varier le pas de temps avec la valeur de $f(t, y)$. Par exemple, si on fixe une « petite » valeur de ε (par exemple $\varepsilon = 0,1$ ou $0,01$), on peut prendre un pas de temps

$$h = \frac{\varepsilon}{\sqrt{1 + \|f(t, y)\|^2}},$$

où $\|f(t, y)\|^2 = f_1(t, y)^2 + \dots + f_n(t, y)^2$ (norme euclidienne). Ainsi quand la pente $f(t, y)$ est petite, h sera proche de ε , et quand cette pente est grande, h sera proche de $\frac{\varepsilon}{\|f(t, y)\|}$.

On a la relation de récurrence suivante pour les valeurs des t_k et des y_k :

$$\begin{cases} h_k &= \frac{\varepsilon}{\sqrt{1 + \|f(t_k, y_k)\|^2}} \\ t_{k+1} &= t_k + h_k \\ y_{k+1} &= y_k + h_k f(t_k, y_k). \end{cases}$$

Cette solution simple a, comme la méthode d'Euler, une convergence linéaire en ε , le pas de temps « de base ». Elle permet la plupart du temps d'intégrer des systèmes raides, cependant il existe d'autres méthodes à pas variable beaucoup plus performantes (en particulier, dont la convergence est quadratique ou plus), notamment le « schéma d'Euler adaptatif ».

Exercice Montrer que le système d'équations de Michaelis-Menten est raide; on pourra prendre comme valeurs pour k_1 , k_2 et k_3 , respectivement 1, 0,08 et 10, et comme valeurs initiales pour $a(t)$, $e(t)$, $c(t)$ et $b(t)$, respectivement 100, 10, 0 et 0. Intégrer numériquement ce système, et comparer le résultat avec l'approximation faite précédemment.

A.4. Une solution toute faite : la bibliothèque deSolve

Il existe bien sûr des solutions toutes faites en R. Nous en présentons une très brièvement.

Il faut d'abords installer la bibliothèque : `install.packages("deSolve")`.

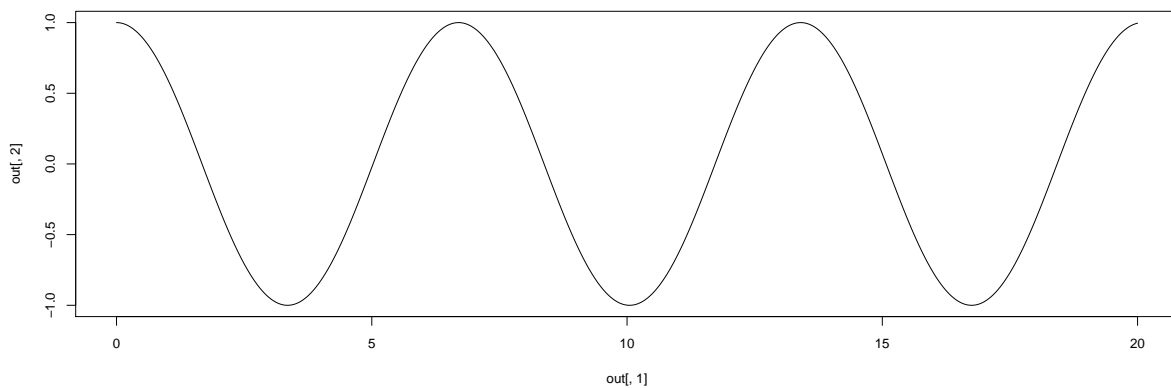
Cette bibliothèque propose la fonction `ode(y, times, func, parms)`, à laquelle on passera

A. Schémas de résolution numérique d'équations différentielles

- l'état initial y (la valeur de $y(0)$),
 - un vecteur de temps $times$ (les instants où on veut calculer $y(t)$),
 - une fonction $func$ à trois paramètres exactement, du type $f(t, y, parms)$, qui renvoie le vecteur $y'(t)$ **encapsulé dans une liste**,
 - un vecteur de paramètres $parms$ qui sera passé à $func$ à chaque appel.
- par exemple pour le système oscillant présenté ci-dessus :

```
f_osc <- function(t,y,a)
{
  y1 <- y[1];
  y2 <- y[2];
  d1 <- y2;
  d2 <- -a*sin(y1);
  return(list(c(d1,d2)));
}

library("deSolve")
out <- ode(y = c(1,0), times = seq(0,20,0.01), func = f_osc, parms = 1)
plot(out[,1], out[,2], type="l")
```



B. Les trois variations sur le schéma d'Euler réalisées en cours

B.1. Schéma d'Euler à pas constant

```
Euler.pc <- function( F, y0, t0 = 0, T, h = 1e-2, ... )
{
  T <- seq(t0, T, by = h);
  N <- length(T);
  Y <- matrix(nrow=length(y0), ncol=N);
  Y[,1] <- y0;
  for(i in 2:N)
  {
    Y[,i] <- Y[,i-1] + h*F(T[i-1],Y[,i-1], ...);
  }
  return(list(t=T, y=Y[,1:N]));
}
```

B.2. Schéma du point milieu

```
Euler.pm <- function( F, y0, t0 = 0, T, h = 1e-2, ... )
{
  T <- seq(t0, T, by = h);
  N <- length(T);
  Y <- matrix(nrow=length(y0), ncol=N);
  Y[,1] <- y0;
  for(i in 2:N)
  {
    m <- Y[,i-1] + h*F(T[i-1],Y[,i-1], ...)/2;
    Y[,i] <- Y[,i-1] + h*F(T[i-1]+h/2, m, ...);
  }
  return(list(t=T, y=Y[,1:N]));
}
```

B.3. Schéma du point milieu à pas variable

```
Euler.pv <- function(F, y0, t0 = 0, T, h = 1e-2, Nmax = 50000, ...)  
{  
  Y <- matrix(nrow = length(y0), ncol=Nmax);  
  T1 <- numeric(Nmax);  
  Y[,1] <- y0;  
  T1[1] <- t0;  
  y <- y0;  
  t <- t0;  
  for(i in 2:Nmax)  
  {  
    Fty <- F(t,y, ...);  
    # pour avoir  $h1^2 + h1^2 * \text{sum}(Fty^2) = h^2$   
    h1 <- h/sqrt(1+sum(Fty**2));  
    m <- y + h1*Fty/2;  
    y <- y + h1*F(t+h1/2, m, ...)  
    t <- t + h1;  
    Y[,i] <- y;  
    T1[i] <- t;  
    if(t > T)  
      break;  
  }  
  return(list(t=T1[1:i], y=Y[,1:i]));  
}
```