Introduction aux méthodes de résolution numérique des équations différentielles

E. Montseny

Table des matières

1 Introduction

Déterminer la solution analytique d'une équation différentielle est à tout point de vue ce qu'il y a de plus intéressant ; il ne faut pas pour autant se voiler la face : en dehors des équations différentielles linéaires à coefficients constants ou d'ordre peu élevé, on est en général bien incapable de résoudre analytiquement une équation différentielle, *a foriori* lorsqu'elle est **non linéaire**.

On doit alors faire appel à des méthodes de résolution numérique, pour calculer une solution approchée au moyen d'ordinateurs. Cette méthode de résolution fait appel à des **schémas numériques**, qui sont des algorithmes de calcul permettant de résoudre numériquement une équation différentielle (i.e. en calculer une solution approchée). On présente dans ce petit court les bases de la construction de schémas numériques pour la résolution d'équations différentielles d'ordre 1.

2 Schémas numériques

2.1 Equations différentielles non linéaires

Si l'on voulait être vraiment cohérent, on ne devrait pas préciser le terme "non linéaires". Une équation différentielle, dans son écriture générale, est non linéaire, le préciser est redondant. Les équations différentielles linéaires sont un type particulier d'équations différentielles.

Une équation différentielle d'ordre 1 s'écrit :

$$\left\{ \begin{array}{l} y'(t) = f(t, y(t)) \quad \forall t > t_0 \\ y(t_0) = y_0 \end{array} \right. \tag{1}$$

où f est une fonction régulière. On se propose de résoudre numériquement des équations de ce type, c'est-à-dire approcher la solution $y(t)$ sur un ensemble dénombrable d'instants t_n, au moyen de calculs simples effectués par un ordinateur. Autrement dit : on discrétise le temps en instants successifs $t_0 < t_1 < t_2 < ... < t_n < ...$, tels que $t_{n+1} = t_n + \Delta t$ (Δt étant appelé le **pas de discrétisation**). On cherche alors à approcher $y(t_1), y(t_2), ..., y(t_n), ...$ par une suite de nombres $y_1, y_2, ..., y_n, ...$

2.2 Schéma d'Euler

C'est sans conteste le plus simple et le plus intuitif des schémas numériques. L'idée de la méthode est simple : approcher le terme dérivé $y'(t)$ de (1) par une simple différence divisée :

$$y'(t_n) \simeq \frac{y(t_n + \Delta t) - y(t_n)}{\Delta t},$$

Dit autrement, cette approximation consiste à prendre la tengeante à la courbe à t_n et considérer cette valeur pour dérivée jusqu'en t_{n+1}. En injectant cette approximation dans (1), on obtient :

$$\frac{y(t_n + \Delta t) - y(t_n)}{\Delta t} \simeq f(t_n, y(t_n)),$$

d'où le schéma de résolution numérique :

$$y_{n+1} = y_n + \Delta t\, f(t_n, y_n). \tag{2}$$

Ainsi, pour résoudre (1) numériquement, il suffit d'utiliser cette relation de proche en proche :
- Initialiser y_0
- Calculer y_{n+1} à partir de la valeur de y_n avec (2)
- $n = n + 1$

Remarque : Ce schéma est dit **explicite**, car la quantité y_{n+1} est exprimée explicitement en fonction de y_n, et qualifié de **schéma à un pas**, ce qui signifie que le calcul de y_{n+1} n'utilise que la valeur y_n à l'instant précédent (une méthode à deux pas utiliserait y_n et y_{n-1} etc.).

Remarque : Ce schéma est intimement lié à la méthode de quadrature des rectangles. En fait, la plupart des schémas numériques de résolution d'EDO sont liés à une méthode de quadrature (ou à une combinaison de plusieurs méthodes).

Il existe de nombreux autres schémas numériques, plus précis que le schéma d'Euler, qui ne seront pas présentés faute de temps. Citons par exemple le schéma du point milieu, le schéma de Heun, ou encore la classique famille des méthodes de Runge-Kutta.

2.3 Quelques notions importantes

1. **Erreur de consistance** : Il s'agit de l'erreur que génére l'utilisation du schéma numérique sur un pas. Cette erreur doit bien évidemment rester raisonnable si on veut espérer obtenir une solution approchée y_n proche de la solution exacte (inconnue) $y(t_n)$. Un schéma **consistant** est un schéma numérique qui est tel que cette erreur tend vers 0 lorsque l'on discrétise le temps de plus en plus fin (i.e. $\Delta t \to 0$). En effet, il faut noter un point important : plus le pas de temps Δt choisi est grand, moins le schéma est précis (on approche une dérivée par la tangente sur une "longue" période). Le schéma d'Euler est consistant.

2. Pour un pas de temps Δt fixé, plus un schéma est précis (plus l'erreur est petite), mieux c'est : ceci est traduit par **l'ordre du schéma numérique** (on ne s'attardera pas dessus). Plus un schéma est d'ordre faible, plus il faudra prendre un pas de temps petit pour avoir une solution approchée précise (et donc plus il faudra faire de calculs...). Le schéma d'Euler est d'ordre 1 et est donc peu précis.
3. **Stabilité**. Toute personne ayant programmé un schéma numérique a été confronté au problème de la stabilité, qui se traduit par une solution numérique qui tend vers l'infini alors qu'elle devrait pas (on dit qu'elle explose)[1]. On dit que le schéma est instable. Ceci peut-être dû soit à une erreur de programmation, soit au schéma lui-même, qui accumule les erreurs (de consistance) de manière incontrôlée. Le schéma d'Euler est stable. Si un schéma est stable et consistant, la solution numérique converge vers la solution exacte de l'EDO lorsque le pas Δt tend vers 0.
4. **Pas de discrétisation**. Il peut cependant arriver qu'un schéma stable semble... instable ! Ceci est dû au pas de temps Δt choisi qui, s'il est trop grand, génère des erreurs trop importantes pour la plage de temps considérée ; il en résulte une solution approchée qui s'écarte complètement de la solution exacte et tend vers l'infini ; cela ne veut pas nécessairement dire que le schéma est instable (cf. démo Matlab : on utilise un schéma d'Euler, stable, et pourtant la solution "explose"), mais simplement qu'il faut diminuer le pas de temps pour avoir une accumulation d'erreur moins importante.

2.4 Applications et démo Matlab

Soit l'équation différentielle :

$$\begin{cases} y'(t) = y(t) + \cos(t) \\ y(0) = -\frac{1}{2} \end{cases}$$

dont on connait la solution analytique :

$$y(t) = \frac{\sin t - \cos t}{2}.$$

On va résoudre cette EDO avec un schéma d'Euler et illustrer les notions introduites. Ecrire l'algorithme de résolution de cette équation par un schéma d'Euler et le mettre en oeuvre sous Matlab.

Faire de même pour l'équation suivante (de Bernoulli) :

$$\begin{cases} y'(t) = y(t) + t\,y^2(t) \\ y(0) = \frac{1}{2} \end{cases}$$

dont la solution est :

$$y(t) = \frac{1}{1 - t + e^{-t}}.$$

[1] bien sûr, certaines équations différentielles ont une solution exacte qui tend vers l'infini, mais ce n'est pas un problème d'instabilité dans ce cas.