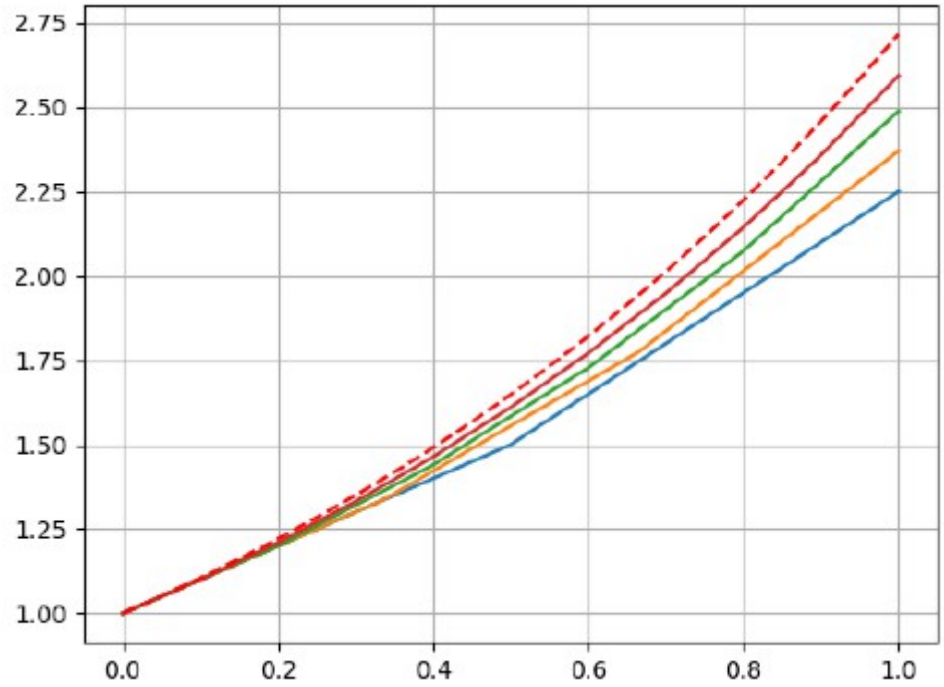
Tout ce que vous avez toujours voulu savoir sur la méthode d'Euler

Contexte : dans de nombreuses situations issues de la Physique ou de la S2i, interviennent des équations différentielles. En général, on ne sait pas résoudre exactement – au sens mathématique du terme – ces équations différentielles, càd que l'on ne dispose pas de formule donnant l'expression exacte des solutions.

Motivation : à l'aide des informations contenues dans l'équation différentielle, construire une « solution approchée » de cette équation.



Explicitons les différents termes du problème

Equation différentielle (ED) : c'est une équation dans laquelle l'inconnue est une fonction, et donnant un lien entre cette fonction et sa dérivée (ou ses dérivées successives).

Exemple 1 : y'(x) = y(x) est une ED dont les solutions sont les fonctions dérivables f telles que pour tout réel x on a : f '(x) = f(x).

Exemple 2 : dy/dt = y est la même ED que plus haut, écrit « à la Physicienne ».

Exemple 3 : y"(x) = sin(y(x)) est une ED dont les solutions sont les fonctions deux fois dérivables f (càd que f ' et f " existent) telles que f "(x) = sin(f(x)) pour tout réel x.

Condition initiale : très souvent en Physique, on ne recherche pas toutes les solutions d'une ED donnée, mais seulement la solution vérifiant une (ou plusieurs) condition(s) initiale(s). Ces conditions initiales sont des valeurs que l'on impose de prendre à la solution f.

Exemple : [y'(x) = y(x) et y(0) = 1] est une ED avec condition initiale. Il existe cette fois une unique solution, la fonction f telle que f ' = f, et f(0) = 1.

Construire une « solution approchée » d'une ED (avec condition initiale) : lorsque l'on ne sait pas résoudre exactement une ED (nous en parlerons dans le cours de Maths de cette année), on peut construire une fonction approximant la solution de cette ED, grâce à la *méthode d'Euler*.
Graphiquement, cela signifie que l'on construit la courbe représentative d'une fonction qui donne une estimation « fiable » de la solution exacte.

Un exemple pour présenter la méthode d'Euler

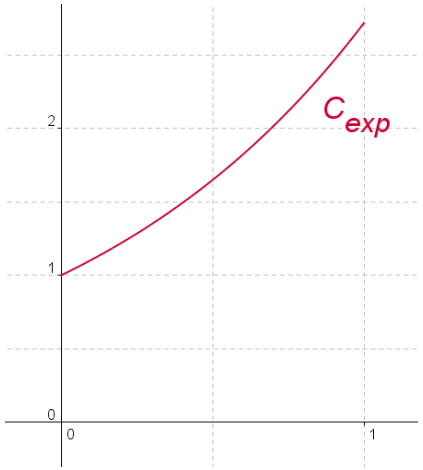
Dans la suite de ce document, on considère l'ED avec condition initiale :

(***) $[\ y'(x) = y(x) \text{ et } y(0) = 1\ ]$

Observez que sur cet exemple, la recherche d'une solution approchée est « inutile », puisque l'on connaît la solution exacte de ce problème...

En effet, il existe une unique fonction f telle que pour tout réel x on a f '(x) = f(x), et qui satisfait la condition f(0) = 1 : la fonction exponentielle.

L'objectif des pages suivantes est d'expliquer le principe de la méthode d'Euler pour obtenir une solution approchée de **(***)** sur l'intervalle [0, 1].



La clef de la méthode d'Euler

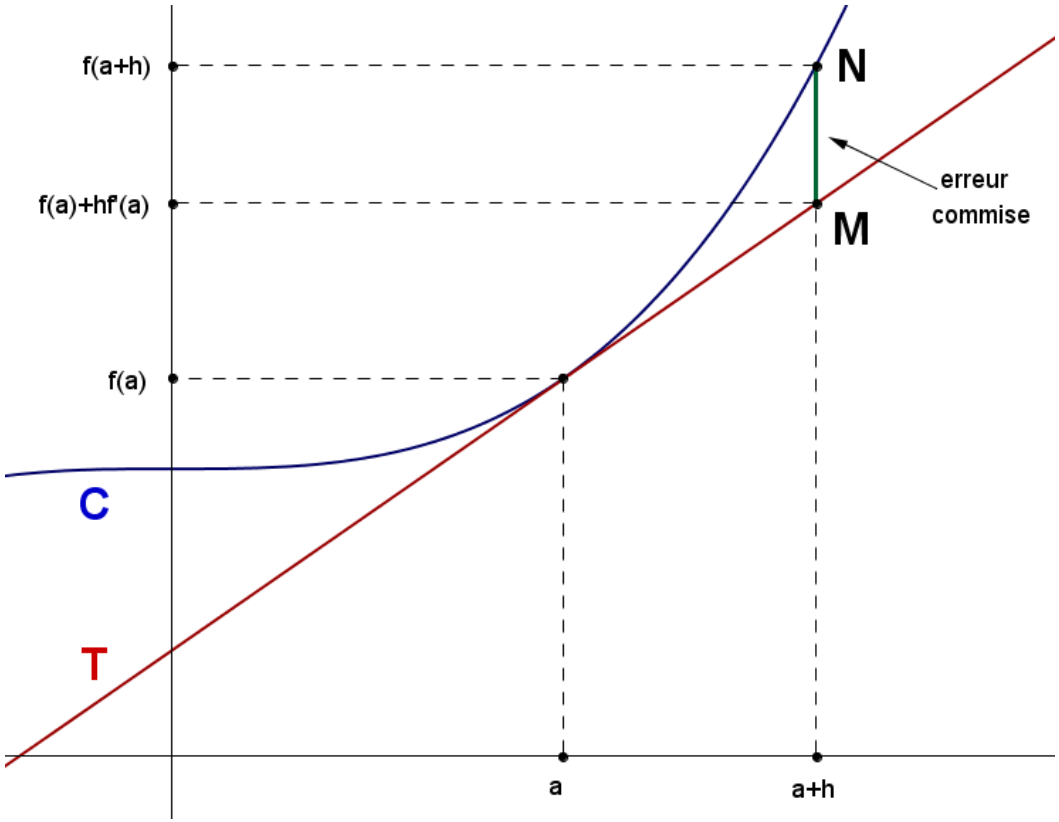
Soit f une fonction dérivable en a (avec a réel).

La courbe représentative C de f admet au point d'abscisse admet une tangente T, d'équation :

$y = f'(a)(x - a) + f(a)$ que l'on peut réécrire sans peine : $y = f(a) + f'(a)(x - a)$ **(###)**

Soit h un réel.

- Le point N d'abscisse a+h de C a pour coordonnées (a+h, f(a+h)) ;
- Le point M d'abscisse a+h de T a pour coordonnées (a+h, f(a)+h f '(a)) (pour obtenir l'ordonnée, prendre x = a+h dans l'équation **(###)**).



Principe de la méthode d'Euler

Avec les notations de la page précédente, le principe de la méthode d'Euler consiste à approcher la fonction f par la fonction affine représentée par T (graphiquement, cela signifie que l'on approche la courbe C par sa tangente).

Dans ce contexte, la valeur approchée de f(a+h) fournie par la méthode d'Euler sera donc l'ordonnée de M, càd que l'on se satisfait de l'approximation :

$$f(a+h) \approx f(a) + h\, f'(a) \qquad \textbf{(@@@)}$$

En effectuant cette approximation, on commet une erreur *E*:

$$E = | f(a+h) - f(a) - h\, f'(a) |$$

Cette erreur est matérialisée par la longueur du segment [MN] sur le graphe de la page précédente.

Remarque : intuitivement, il apparaît assez clairement que plus « h » sera petit, plus l'erreur commise sera petite.

En d'autres termes, plus « h » est petit, plus l'approximation **(@@@)** est bonne.

****

Ces observations faites, on explique dans ce qui suit comment utiliser cette approximation pour construire une solution approchée de l'ED .

Exemple d'application

Considérons donc l'ED avec condition initiale :

$$\textbf{(***)} \quad [\ y'(x) = y(x) \ \text{ et } \ y(0) = 1\ ]$$

On veut construire une courbe approchant celle de l'unique solution de ce problème sur l'intervalle [0, 1] (*rappelons que cette unique solution est la fonction exponentielle*).

Première approximation

Prenons : a = 0 et h = 1. On a alors :

$$f(a) = f(0) = 1 \text{ (condition initiale) ; } f'(0) = f(0) = 1 \text{ (ED) ; et } \ a+h = 1$$

Dans ce cas, l'approximation **(@@@)** donne :

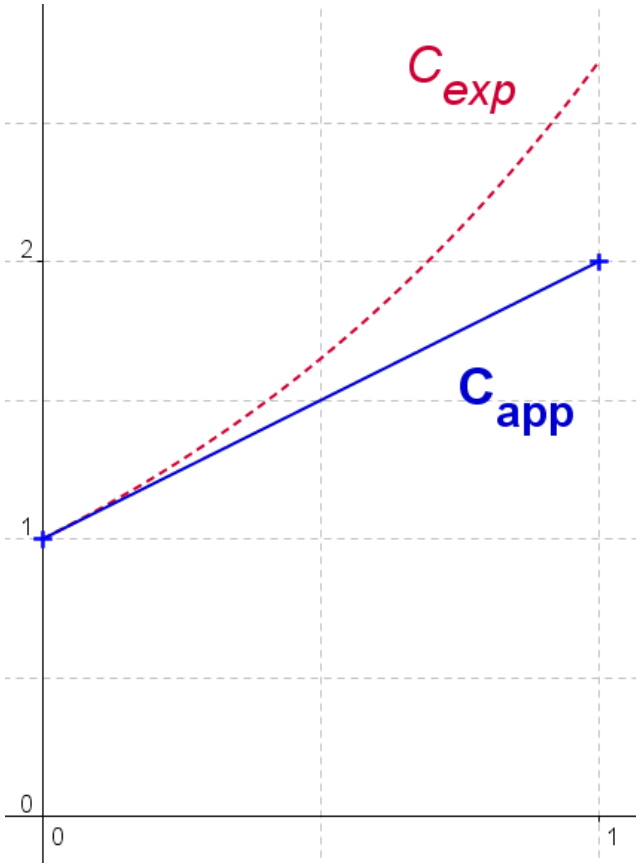
$$f(1) \approx f(0) + 1*f'(0) \quad \text{soit :} \quad f(1) \approx 2$$

La courbe de la solution approchée est dans ce cas un segment, joignant les points de coordonnées (0, 1) et (1,2).

Au passage, l'approximation ci-dessus signifie (puisque dans ce cas on connaît la solution exacte) que :

$$e^1 \approx 2$$

Illustration graphique de la première approximation



La courbe de la solution approchée (notée C_{app}) est un segment approchant très grossièrement la courbe représentative de la solution exacte sur l'intervalle [0, 1].

Pour améliorer cette approximation, l'idée est d'utiliser un « h » plus petit, afin de réduire l'erreur commise.

A cette fin, on partage l'intervalle [0, 1] en deux morceaux de longueur égale. On détaille ceci dans la seconde approximation.

Deuxième approximation

On partage l'intervalle [0, 1] en deux morceaux : [0, 1/2] et [1/2, 1].

Prenons dans un premier temps : a = 0 et h = 1/2. On a alors :

$$f(a) = f(0) = 1 \text{ (condition initiale) ; } f'(0) = f(0) = 1 \text{ (ED) ; et } a+h = 1/2$$

Dans ce cas, l'approximation **(@@@)** donne :

$$f(1/2) \approx f(0) + (1/2)*f'(0) \quad \text{soit :} \quad f(1/2) \approx 3/2$$

Dans un second temps, prenons donc : a = 1/2 et h = 1/2. On a alors :

$$f(a) = f(1/2) = 3/2 \text{ (approx précédente) ; } f'(1/2) = f(1/2) = 3/2 \text{ (ED) ; et } a+h = 1$$

Dans ce cas, l'approximation **(@@@)** donne :

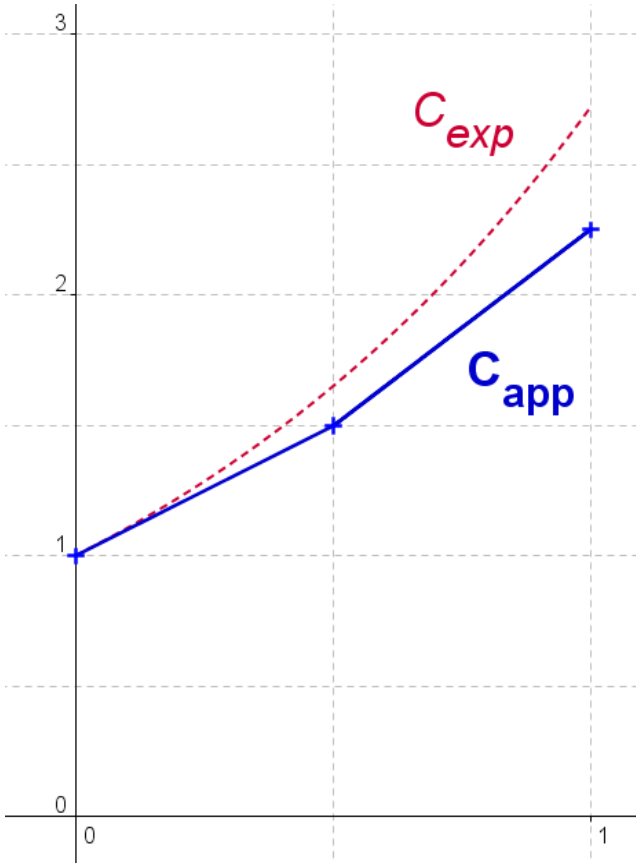
$$f(1) \approx f(1/2) + (1/2)*f'(1/2) \quad \text{soit :} \quad f(1) \approx 9/4$$

La courbe de la solution approchée est alors une ligne polygonale (constituée de deux segments), joignant les points de coordonnées (0, 1), (1/2, 3/2) et (1, 9/4).

Au passage, l'approximation ci-dessus signifie (puisque dans ce cas on connaît la solution exacte) que :

$$e^1 \approx 2.25$$

Illustration graphique de la seconde approximation



La courbe de la solution approchée (notée C_{app}) est un une ligne polygonale approchant légèrement mieux (que la première approx) la courbe représentative de la solution exacte sur l'intervalle [0, 1].

Pour améliorer encore cette approximation, l'idée est, vous l'avez compris, d'utiliser un « h » encore plus petit, afin de réduire encore l'erreur commise.

A cette fin, on partage l'intervalle [0, 1] en trois morceaux de longueur égale. On détaille ceci dans la troisième approximation.

Troisième approximation

On partage l'intervalle [0, 1] en trois morceaux : [0, 1/3], [1/3, 2/3] et [2/3, 1].

Prenons dans un premier temps : a = 0 et h = 1/3. On a alors :

$$f(a) = f(0) = 1 \text{ (condition initiale) ; } f'(0) = f(0) = 1 \text{ (ED) ; et } a+h = 1/3$$

Dans ce cas, l'approximation **(@@@)** donne :

$$f(1/3) \approx f(0) + (1/3)*f'(0) \quad \text{soit :} \quad f(1/3) \approx 4/3$$

Dans un second temps, prenons donc : a = 1/3 et h = 1/3. On a alors :

$$f(a) = f(1/3) = 4/3 \text{ (approx précédente) ; } f'(1/3) = f(1/3) = 4/3 \text{ (ED) ; et } a+h = 2/3$$

Dans ce cas, l'approximation **(@@@)** donne :

$$f(2/3) \approx f(1/3) + (1/3)*f'(1/3) \quad \text{soit :} \quad f(2/3) \approx 16/9$$

Enfin, prenons: a = 2/3 et h = 1/3. On a alors :

$$f(a) = f(2/3) = 16/9 \text{ (approx précédente) ; } f'(2/3) = f(2/3) = 16/9 \text{ (ED) ; et } a+h = 1$$

Dans ce cas, l'approximation **(@@@)** donne :

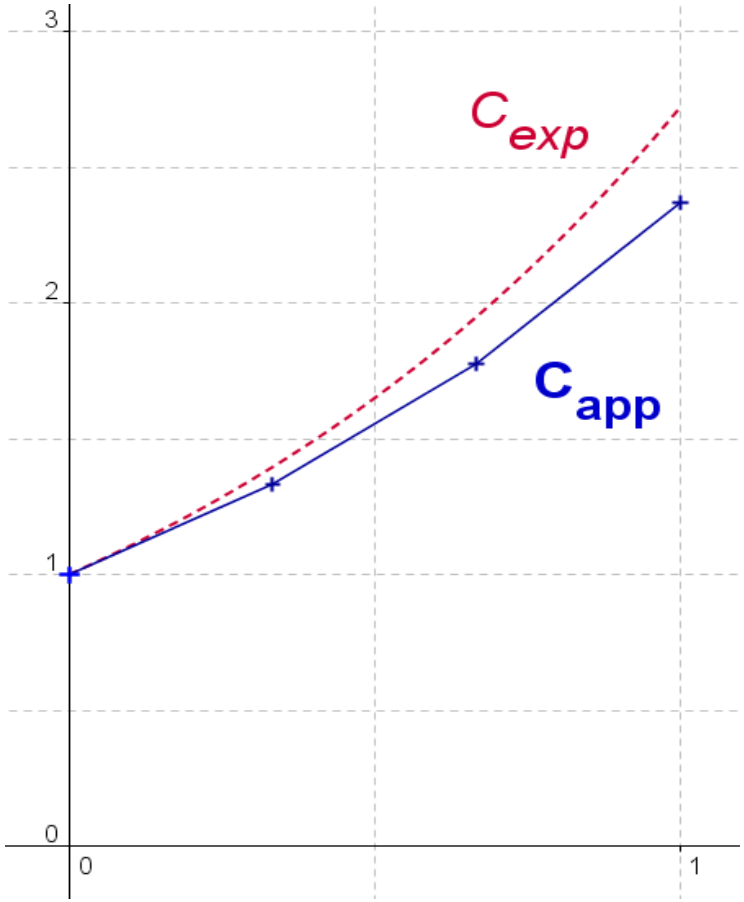
$$f(1) \approx f(2/3) + (1/3)*f'(2/3) \quad \text{soit :} \quad f(2/3) \approx 64/27$$

La courbe de la solution approchée est alors une ligne polygonale (constituée de trois segments), joignant les points de coordonnées (0, 1), (1/3, 4/3), (2/3, 16/9) et (1, 64/27).

Au passage, l'approximation ci-dessus signifie que :

$$e^1 \approx 2.37$$

Illustration graphique de la troisième approximation



La courbe de la solution approchée (notée C_{app}) est une ligne polygonale approchant encore un peu mieux la courbe représentative de la solution exacte sur l'intervalle [0, 1].

Pour améliorer encore cette approximation...

Généralisation

Il résulte des exemples précédents que la qualité de l'approximation sera d'autant plus grande que le nombre de « petits segments » pris dans l'intervalle [0, 1] sera grand.

Il résulte également des calculs précédents... que l'on va assez vite confier les lourds calculs à l'ordinateur !

Dans ce contexte, on se propose ci-dessous de généraliser les approximations précédentes (à un nombre de points quelconque), de décrire l'algorithme général de la méthode d'Euler, et de le coder en Python.

Les ingrédients

- Un entier N : le nombre de « petits segments »
- Un réel h : h = 1/N, la longueur d'un petit intervalle (appelé *pas* de la méthode d'Euler)
- N+1 nombre réels : $x_0 = 0$, $x_1 = 1/N$, $x_2 = 2/N$,…, $x_N = N/N = 1$: ce sont les abscisses des points de la ligne polygonale

L'objectif à atteindre

- Construire une liste de N+1 nombre réels : y_0, y_1, y_2,…, y_N correspondant aux ordonnées de la ligne polygonale. Ces réels sont les valeurs approchées des images des réels x_i par la solution de l'ED.

  C'est cette construction que l'on détaille page suivante.

Construction de la liste des N+1 nombre réels : $y_0, y_1, y_2, \ldots, y_N$

- $y_0 = 1$ (Condition initiale **(¥)**)
- Puis : $y_1 = y_0 + h * f'(x_0)$ (approximation **(@@@)**) d'où : $y_1 = y_0 + h * f(x_0)$ (équation différentielle)
  Finalement :

$$y_1 = y_0 + h * y_0$$

- On recommence avec y_2: $y_2 = y_1 + h * f'(x_1)$ d'où : $y_2 = y_1 + h * f(x_1)$ (équation différentielle)
  Finalement :

$$y_2 = y_1 + h * y_1$$

- Et ainsi de suite...
- Plus généralement, pour tout entier k entre 0 et N-1, on a la relation

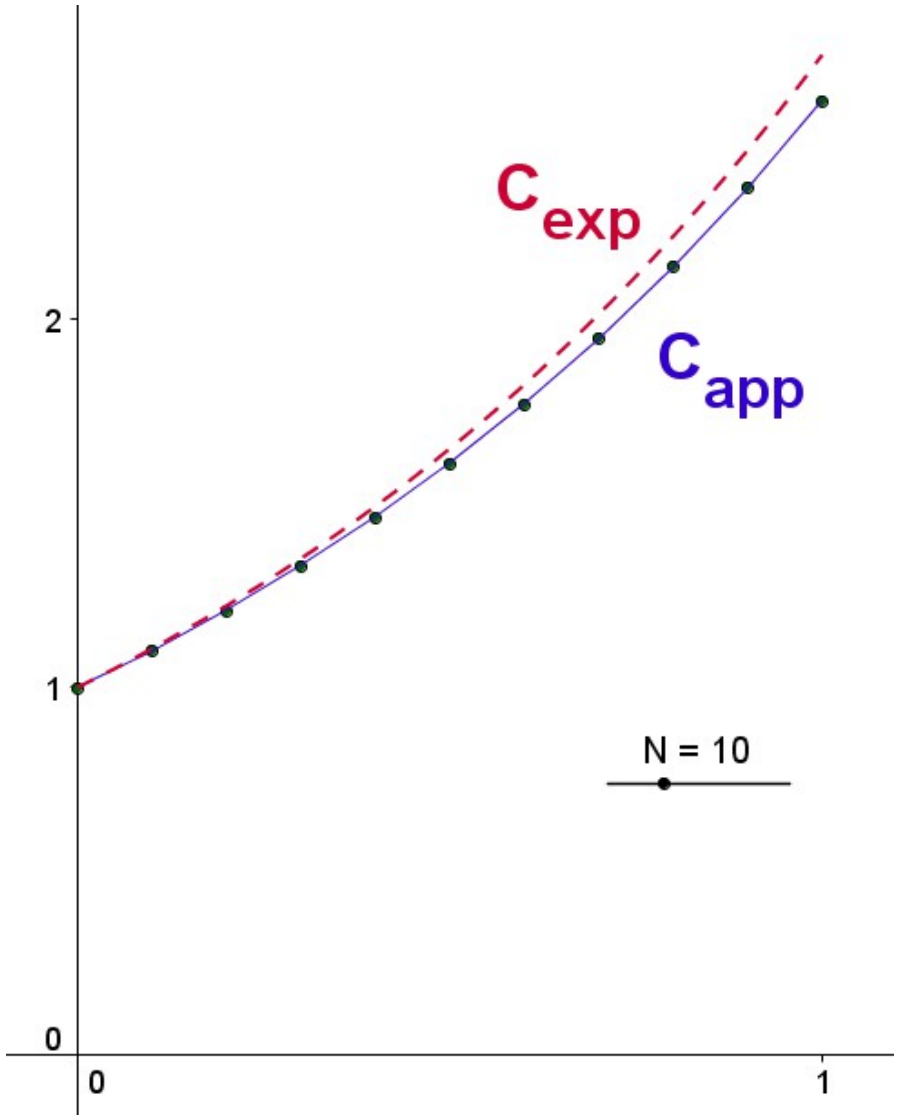
$$y_{k+1} = y_k + h * y_k \qquad (®)$$

En résumé, la suite des valeurs des y_k est définie par une condition initiale **(¥)**, et une relation de récurrence **(®)**.

Remarque : au passage, la suite des x_k est définie par une condition initiale ($x_0 = 0$), et une relation de récurrence :

$$x_{k+1} = x_k + h \qquad \text{(pour tout entier k entre 0 et N-1)}$$

Illustration : représentation de la solution approchée pour N = 10



Codage en Python de la méthode d'Euler

L'objectif est à présent de rédiger un programme demandant à la machine de créer deux listes (celle des « x_k » et celle des « y_k ») intervenant dans la méthode d'Euler.

Pour y parvenir, on présente ci-dessous le strict minimum à connaître sur la définition et la syntaxe des listes en langage Python.

- **Un outil : les listes en Python**

Une liste L est définie en Python avec des crochets : L1 = [1, 2, 3] et L2 = [4] sont des exemples de listes (de longueurs respectives 3 et 1).

On peut faire « la somme » de deux listes : en langage informatique, on parle plutôt de *concaténation* de listes. Par exemple, la concaténation des listes L1 et L2 précédentes est :

L1 + L2 = [1, 2, 3, 4]

Code Python de la méthode d'Euler (pour la résolution de l'ED (***))

```python
N = 10 # nombre de "petits segments"

h = 1/N # définition du pas = "longueur d'un petit segment" (le dt des Physiciens)

x = 0 # initialisation de x ("x0 = 0"); valeur de l'abscisse initiale

y = 1 # initialisation de y ("y0 = 1"); valeur de l'ordonnée initiale

LX = [x] # initialisation de la liste des abscisses

LY = [y] # initialisation de la liste des ordonnées

for k in range(N):

    y = y + h * y # relation de récurrence entre "y(k+1) et y(k)"
    LY = LY + [y] # on rajoute dans la liste des ordonnées la valeur "y(k+1)"

    x = x + h # relation de récurrence entre "x(k+1) et x(k)"
    LX = LX + [x] # on rajoute dans la liste des abscisses la valeur "x(k+1)"
```