

# Équations différentielles ordinaires

Ahmed Ammar ([ahmed.ammar@fst.utm.tn](mailto:ahmed.ammar@fst.utm.tn))

Institut Préparatoire aux Études Scientifiques et Techniques, Université de Carthage.

Feb 12, 2020

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Convergence et de stabilité de la méthode d'Euler: Cas des systèmes linéaires</b>	<b>2</b>
2.1	La méthode d'Euler explicite (progressive) . . . . .	2
2.2	La méthode d'Euler implicite (rétrograde) . . . . .	3
2.3	Exemple: Oscillateur libre amorti [masse, ressort, amortisseur] . . . . .	4
2.4	Conclusion . . . . .	10

## 1 Introduction

Dans les domaines scientifiques et industriels, il est courant aujourd'hui d'étudier la nature ou les dispositifs technologiques au moyen de modèles sur ordinateur. Avec de tels modèles, l'ordinateur agit comme un laboratoire virtuel où les expériences peuvent être effectuées de manière rapide, fiable, sûre et économique.

Les équations différentielles constituent l'un des outils mathématiques les plus puissants pour comprendre et prédire le comportement des systèmes dynamiques de la nature, de l'ingénierie et de la société. Un système dynamique est un système avec un état, généralement exprimé par un ensemble de variables, évoluant dans le temps. Par exemple, un pendule oscillant, la propagation d'une maladie et les conditions météorologiques sont des exemples de systèmes dynamiques. Nous pouvons utiliser les lois fondamentales de la physique, ou l'intuition simple, pour exprimer des règles mathématiques qui régissent l'évolution du système dans le temps. Ces règles prennent la forme d'équations différentielles.

## 2 Convergence et de stabilité de la méthode d'Euler: Cas des systèmes linéaires

En mécanique classique, les équations du mouvement d'un système mécanique (systèmes de points matériels, système de solides) sont des équations différentielles du second ordre par rapport au temps. La connaissance des positions et des vitesses des points à l'instant  $t = 0$  suffit à déterminer le mouvement pour  $t > 0$ .

Ces équations sont souvent non linéaires car les forces elles-mêmes le sont (par exemple la force de gravitation) et car l'accélération est souvent une fonction non linéaire des degrés de liberté. Dans ce cas, il est fréquent que l'on ne connaisse pas de solution analytique exacte. On est alors amené à rechercher une solution approchée par une méthode numérique.

Cette partie du cours explique le principe de ce type d'intégration numérique. On prendra l'exemple de l'oscillateur harmonique (dont la solution exacte est connue) auquel on appliquera la méthode numérique d'Euler. On abordera les notions importantes de *convergence* et de *stabilité*.

On verra aussi des variantes de la méthode d'Euler, qui peuvent être utilisées pour résoudre des systèmes conservatifs à N corps, par exemple en dynamique moléculaire.

De manière générale soit le système d'équations différentielles suivant:

$$\dot{\mathbf{u}} = f(\mathbf{u}) \quad (1)$$

Où  $\mathbf{u}$  peut être un vecteur d'état et  $f(\mathbf{u})$  peut être linéaire ou non linéaire.

Soit  $f(\mathbf{u}) = \mathbf{A} \cdot \mathbf{u}$  avec  $\mathbf{A}$  une matrice. Donc on peut écrire l'équation (2) comme suit:

$$\dot{\mathbf{u}} = \mathbf{A} \cdot \mathbf{u} \quad \text{avec } \mathbf{u}(t = 0) = \mathbf{u}_0 \quad (2)$$

La solution analytique exacte d'un tel système est de la forme:

$$\mathbf{u}(t) = e^{\mathbf{A}t} \cdot \mathbf{u}_0 \quad (3)$$

On se propose d'appliquer différentes méthodes d'Euler au système (2).

### 2.1 La méthode d'Euler explicite (progressive)

$$\frac{\mathbf{u}_{k+1} - \mathbf{u}_k}{\Delta t} \approx \dot{\mathbf{u}}_k = f(\mathbf{u}_k) \quad (4)$$

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \Delta t f(\mathbf{u}_k) \quad (5)$$

Si  $\dot{\mathbf{u}} = \mathbf{A} \cdot \mathbf{u}$  alors;

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \Delta t \mathbf{A} \cdot \mathbf{u}_k = (\mathbf{I} + \Delta t \mathbf{A}) \cdot \mathbf{u}_k \quad (6)$$

Où  $\mathbf{I}$  est la matrice identité.

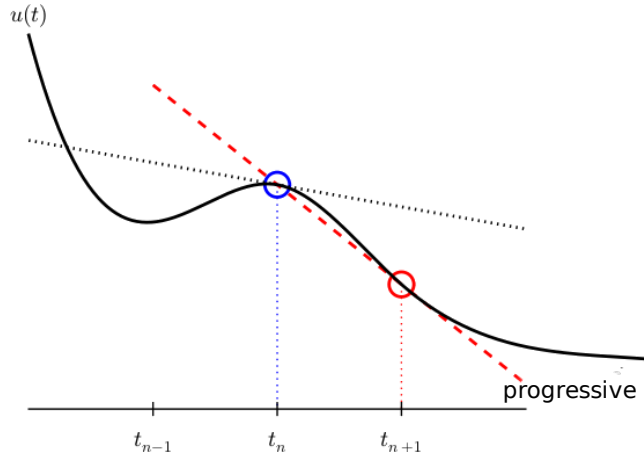


Figure 1: Illustration d'une approximation par différence progressive de la dérivée.

## 2.2 La méthode d'Euler implicite (rétrograde)

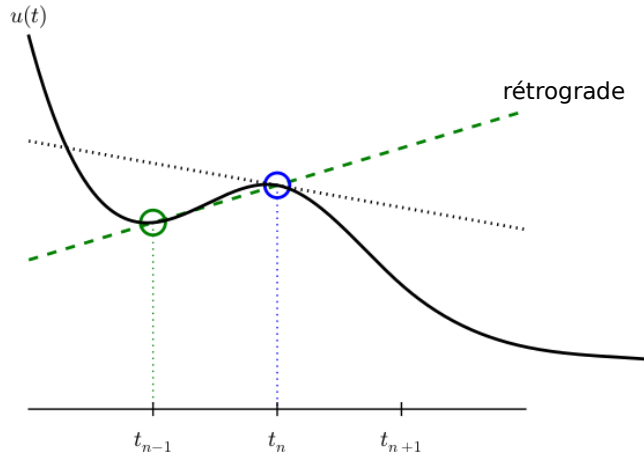


Figure 2: Illustration d'une approximation par différence rétrograde de la dérivée.

$$\frac{\mathbf{u}_{k+1} - \mathbf{u}_k}{\Delta t} \approx \dot{\mathbf{u}}_{k+1} = f(\mathbf{u}_{k+1}) \quad (7)$$

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \Delta t f(\mathbf{u}_{k+1}) \quad (8)$$

Si  $\dot{\mathbf{u}} = \mathbf{A} \cdot \mathbf{u}$  alors;

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \Delta t \mathbf{A} \cdot \mathbf{u}_{k+1} \quad (9)$$

$$(\mathbf{I} - \Delta t \mathbf{A}) \cdot \mathbf{u}_{k+1} = \mathbf{u}_k \quad (10)$$

$$\mathbf{u}_{k+1} = (\mathbf{I} - \Delta t \mathbf{A})^{-1} \cdot \mathbf{u}_k \quad (11)$$

Où  $\mathbf{I}$  est la matrice identité.

### 2.3 Exemple: Oscillateur libre amorti [masse, ressort, amortisseur]

Un bloc de masse  $m$  est lié à l'extrémité libre d'un ressort de raideur  $k$ , de longueur au repos  $l$ , de masse négligeable et d'élasticité parfaite, l'autre extrémité du ressort étant fixe. Le système est supposé dans l'espace (on néglige la force de pesanteur). Le seul mouvement possible pour le bloc est une translation suivant  $x$ ; on assimilera le bloc à un point matériel  $M$ .

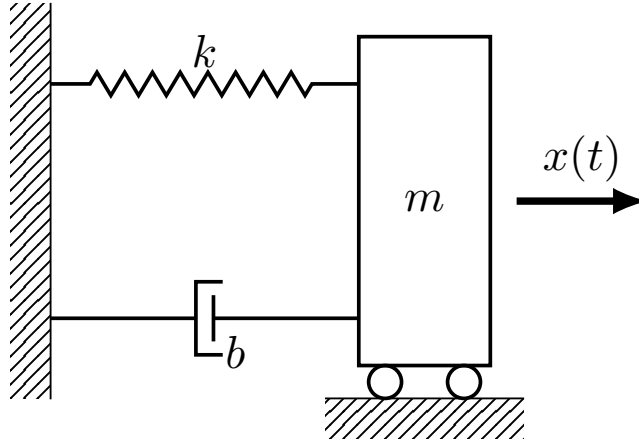


Figure 3: Schéma d'un système dynamique oscillant amorti unidimensionnel.

#### Bilan des forces.

- La force de rappel  $\vec{F}_r = -k \vec{x}$  où  $k$  est un coefficient positif et  $\vec{x}$  le vecteur position de  $M$ .
- Le système est amorti. L'amortissement de type visqueux est représenté par un amortisseur qui exerce la force dissipative (ou force d'amortissement visqueux)  $\vec{F}_a = -b \vec{v}$  où  $b$  est un coefficient positif et  $\vec{v}$  le vecteur vitesse de  $M$ .

**Équation de mouvement.** La deuxième loi de Newton pour le système peut être écrite avec l'accélération multipliée par la masse du côté gauche et la somme des forces du côté droit:

$$m \vec{a} = \vec{F}_a + \vec{F}_r \quad (12)$$

$$m\ddot{x} = -b\dot{x} - kx \quad (13)$$

$$m\ddot{x} + b\dot{x} + kx = 0 \quad (14)$$

On réécrit cette équation sous la forme canonique suivante :

$$\ddot{x} + 2\zeta\omega_0\dot{x} + \omega_0^2x = 0 \quad (15)$$

avec  $\omega_0 = \sqrt{\frac{k}{m}}$  désigne une *pulsation caractéristique* et  $\zeta = \frac{b}{2\sqrt{km}}$  est une quantité positive sans dimension, appelée *taux d'amortissement*.

C'est une équation différentielle linéaire d'ordre 2 à coefficients constants.

On peut trouver numériquement la solution de l'équation (15) à l'aide des méthodes d'Euler à partir du système d'équations différentielles ordinaires suivant:

$$\dot{x} = v \quad (16)$$

$$\dot{v} = -2\zeta\omega_0v - \omega_0^2x \quad (17)$$

$$(18)$$

$$\frac{d}{dt} \begin{pmatrix} x \\ v \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\omega_0^2 & -2\zeta\omega_0 \end{pmatrix} \cdot \begin{pmatrix} x \\ v \end{pmatrix} \quad (19)$$

L'équation (2.3) est de la forme:  $\dot{\mathbf{u}} = \mathbf{A} \cdot \mathbf{u}$  avec:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ -\omega_0^2 & -2\zeta\omega_0 \end{pmatrix}$$

et

$$\mathbf{u} = \begin{pmatrix} x \\ v \end{pmatrix}$$

Supposons que nous voulions résoudre le problème avec:  $\omega_0 = 2\pi$ ,  $\zeta = 0.25$ ,  $\mathbf{u}_0 = \begin{pmatrix} x(t=0) \\ v(t=0) \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$ ,  $\Delta t = 0.01$  pour  $t \in [0, 10]$ . Ce sera une solution sinusoïdale amortie.

**Solution avec la méthode d'Euler explicite.** Nous implémentons l'expression explicite d'Euler montrée dans (6) dans le code python suivant:

```

## NOM DU PROGRAMME: OscillateurEulerExp.py
## IMPORTATION
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d.axes3d import Axes3D
# SYSTÈME: OSCILLATEUR LIBRE AMORTI
w = 2*np.pi # fréquence propre
a = 0.25 # rapport d'amortissement
A = np.array([[0, 1], [-w**2, -2*a*w]])
dt = 0.01 # pas du temps
Tf = 10 # temps finale de la simulation
nsteps = int(Tf/dt)
# CONDITION INITIAL: à t = 0; x = 2, v = 0
u0 = np.array([2,0])
%% ITÉRATION: EULER EXPLICITE
Texp = np.zeros(nsteps)
Uexp = np.zeros((2, nsteps))
Texp[0] = 0.0
Uexp[:,0] = u0
for k in range(nsteps-1):
    Texp[k+1] = Texp[k] + dt
    Uexp[:,k+1] = np.dot((np.eye(2) + dt * A), Uexp[:,k])

plt.figure(figsize=(10,5))
# PLOT POSITION vs TEMPS
plt.suptitle("Simulation d'un oscillateur libre amorti avec un pas d'intégration "+ r"$ \Delta t = $" + str(dt) + "s", fontweight = "bold")
plt.subplot(1,2,1)
plt.plot(Texp,Uexp[0,:], linewidth=2, color = 'k')
plt.xlabel("Temps")
plt.ylabel("Position")
plt.title("Trajectoire de la mass M (Euler explicite)")
# DIAGRAMME DE PHASE 2D
plt.subplot(1,2,2)
plt.plot(Uexp[0,:],Uexp[1,:], linewidth=2, color = 'k')
plt.xlabel("Position")
plt.ylabel("Vitesse")
plt.title("Trajectoire de phase (Euler explicite)")
plt.savefig("EulerExp1D.png"); plt.savefig("EulerExp1D.pdf")
# DIAGRAMME DE PHASE 3D
plt.figure()
ax = plt.axes(projection="3d")
ax.plot(Texp, Uexp[0,:],Uexp[1,:], linewidth=2, color = 'k')
ax.set_xlabel("Temps")
ax.set_ylabel("Position")
ax.set_zlabel("Vitesse")
ax.set_title("Trajectoire de phase (Euler explicite)")
plt.savefig("EulerExp3D.png"); plt.savefig("EulerExp3D.pdf")
plt.show()

```

La figure 4 est générée par le code `OscillateurEulerExp.py`, montrant la divergence et l'instabilité de la méthode Euler explicite. En effet, le pas d'intégration  $\Delta t$  agit considérablement sur la qualité de la simulation et donne un résultat inacceptable physiquement.

Dans le cas d'intégration avec la méthode d'Euler explicite, la figure 5 montre que nous avons un problème d'augmentation d'amplitude dans le cas d'un oscillateur non amorti (courbe bleue pour  $\zeta = 0$ ). Plus le temps de simulation

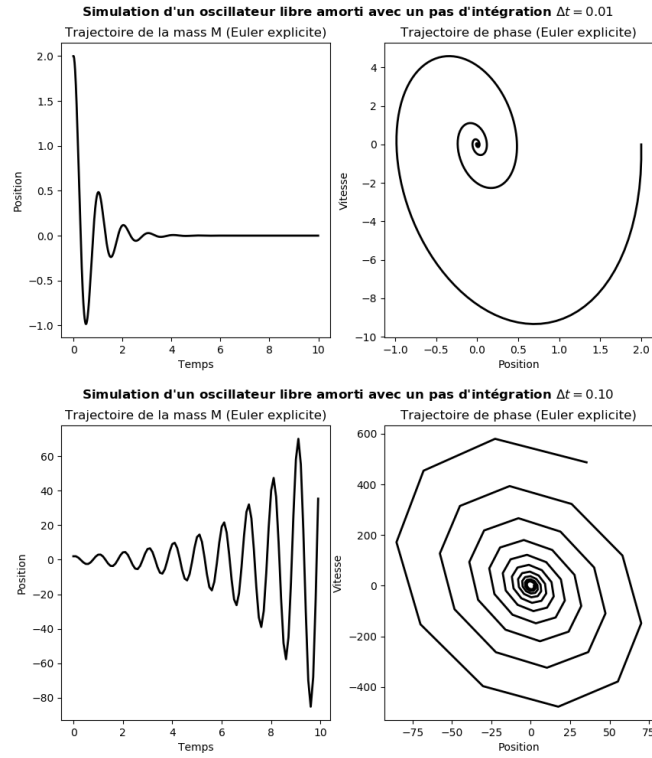


Figure 4: Simulation d'un système oscillant avec différents pas de temps;  $\Delta t = 0.01$  et  $\Delta t = 0.1$  et pour  $\zeta = 0.25$ .

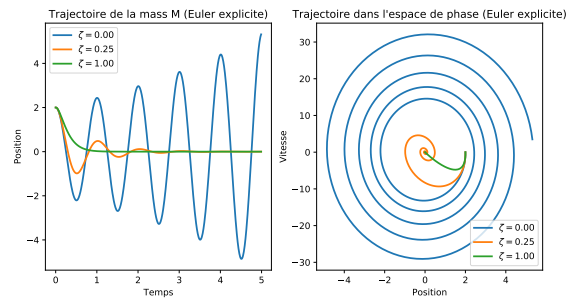


Figure 5: Simulation d'un système oscillant avec différentes valeurs de  $\zeta$  et pour  $\Delta t = 0.01$ .

est long, plus l'amplitude augmente, ce qui n'est pas ce que nous attendons de l'évolution du système dans le temps. En d'autres termes, l'amplitude devrait être constante dans le temps pour un système oscillant non amorti.

**Solution avec la méthode d'Euler implicite.** Nous implémentons l'expression implicite d'Euler montrée dans (11) dans le code python suivant:

```
## NOM DU PROGRAMME: OscillateurEulerImp.py
## IMPORTATION
import numpy as np
from numpy.linalg import inv
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d.axes3d import Axes3D
# SYSTÈME: OSCILLATEUR LIBRE AMORTI
w = 2*np.pi # fréquence propre
a = 0.25 # rapport d'amortissement
A = np.array([[0, 1], [-w**2, -2*a*w]])
dt = 0.1 # pas du temps
Tf = 10 # temps finale de la simulation
nsteps = int(Tf/dt)
# CONDITION INITIAL: à t = 0; x = 2, v = 0
u0 = np.array([2,0])
### ITÉRATION: EULER IMPLICITE
Timp = np.zeros(nsteps)
Uimp = np.zeros((2, nsteps))
Timp[0] = 0.0
Uimp[:,0] = u0
for k in range(nsteps-1):
    Timp[k+1] = Timp[k] + dt
    Uimp[:,k+1] = np.dot(inv(np.eye(2) - dt * A), Uimp[:,k])

plt.figure(figsize=(10,5))
# PLOT POSITION vs TEMPS
plt.suptitle("Simulation d'un oscillateur libre amorti avec un pas d'intégration "+ r"$ \Delta t = %$")
fontweight = "bold")
plt.subplot(1,2,1)
plt.plot(Timp,Uimp[0,:], linewidth=2, color='k')
plt.xlabel("Temps")
plt.ylabel("Position")
plt.title("Trajectoire de la mass M (Euler implicite)")
# DIAGRAMME DE PHASE 2D
plt.subplot(1,2,2)
plt.plot(Uimp[0,:],Uimp[1,:], linewidth=2, color='k')
plt.xlabel("Position")
plt.ylabel("Vitesse")
plt.title("Trajectoire de phase (Euler implicite)")
plt.savefig("Eulerimp1D_2.png"); plt.savefig("Eulerimp1D_2.pdf")
# DIAGRAMME DE PHASE 3D
plt.figure()
ax = plt.axes(projection="3d")
ax.plot(Timp, Uimp[0,:],Uimp[1,:], linewidth=2, color='k')
ax.set_xlabel("Temps")
ax.set_ylabel("Position")
ax.set_zlabel("Vitesse")
ax.set_title("Trajectoire de phase (Euler implicite)")
plt.savefig("Eulerimp3D_2.png"); plt.savefig("Eulerimp3D_2.pdf")
plt.show()
```

La figure 6 est générée par le code `OscillateurEulerImp.py`, montrant que la méthode d'Euler implicite est plus stable que la méthode Euler explicite. Nous remarquons toujours qu'il y a un effet du changement du pas d'intégration  $\Delta t$



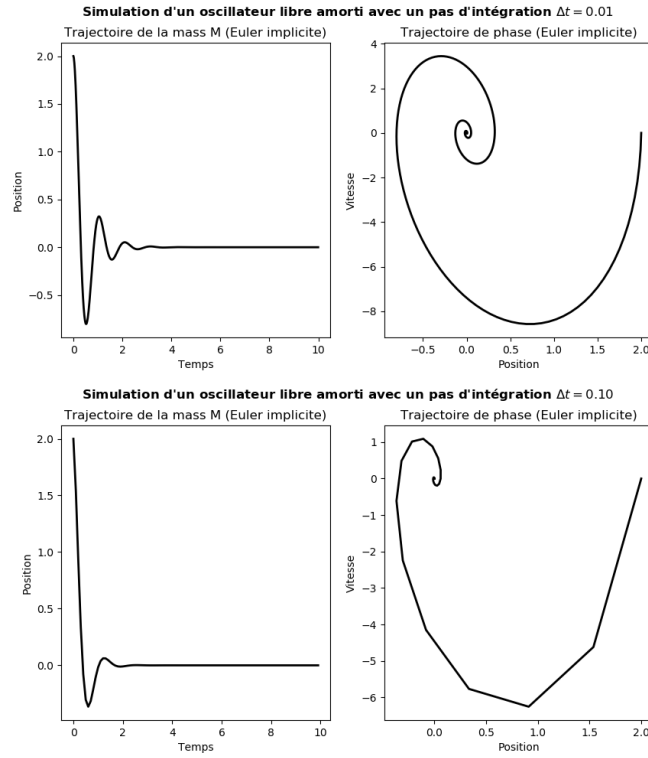


Figure 6: Simulation d'un système oscillant avec différents pas de temps;  $\Delta t = 0.01$  et  $\Delta t = 0.1$  et pour  $\zeta = 0.25$ .

sur la qualité de la simulation mais le résultat du calcul est désormais acceptable physiquement.

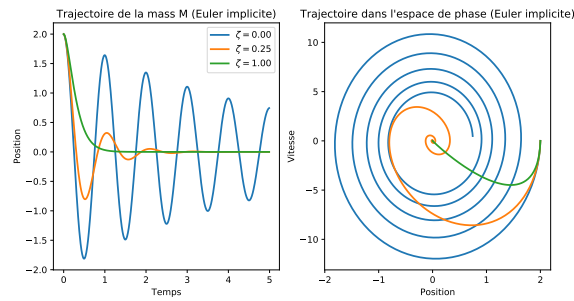


Figure 7: Simulation d'un système oscillant avec différentes valeurs de  $\zeta$  et pour  $\Delta t = 0.01$ .

Même problème avec l'amplitude pour le cas d'intégration avec la méthode implicite d'Euler, la figure 7 montre que nous avons un problème de diminution d'amplitude dans le cas d'un oscillateur non amorti (courbe bleue pour  $\zeta = 0$ ). Comme indiqué ci-dessus, l'amplitude devrait être constante dans le temps pour un système oscillant non amorti.

## 2.4 Conclusion

La conclusion ici est que la méthode Euler implicite est plus stable que celle explicite. Les deux méthodes posent un problème fondamental avec ses amplitudes croissantes et décroissantes, pour le cas d'oscillateur libre non amorti, et qu'un très petit  $\Delta t$  est nécessaire pour obtenir des résultats satisfaisants. Plus la simulation est longue, plus  $\Delta t$  doit être petit. Il est certainement temps de rechercher des méthodes numériques plus stables et plus efficaces tels que les méthodes de Runge-Kutta.