

Eqdf3

December 18, 2020

1 Une équation différentielle avec conditions aux limites

Marc Lorenzi

18 décembre 2020

```
[1]: import matplotlib.pyplot as plt
import math
```

Dans tout le notebook, $L > 0$ est un réel fixé.

1.1 1. Une équation différentielle

1.1.1 1.1 Le problème

Pour tous réels λ, μ , considérons le problème suivant :

$(\mathcal{P}_{\lambda, \mu})$ *Trouver toutes les fonctions $f : [0, L] \rightarrow \mathbb{R}$ qui sont solutions de l'équation différentielle*

$$(\mathcal{E}_{\lambda, \mu}) \quad y'' + \lambda y' + \mu y = 0$$

et qui vérifient $f(0) = f(L) = 0$.

Nous avons là ce que l'on appelle une équation différentielle avec des *conditions aux limites*. Contrairement au problème de Cauchy où on impose la valeur de f et de f' **en un même point**, on impose ici la valeur de f **en deux points distincts**.

1.1.2 1.2 L'équation caractéristique

L'équation caractéristique de l'équation différentielle $(\mathcal{E}_{\lambda, \mu})$ est

$$(E_{\lambda, \mu}) \quad X^2 + \lambda X + \mu = 0$$

Clairement, une discussion s'impose sur la nullité et le signe de $\Delta = \lambda^2 - 4\mu$. Il y a donc trois cas à considérer.

1.1.3 1.3 Le cas $\Delta = 0$

Supposons $\Delta = 0$. L'équation caractéristique a une unique racine réelle α . Les solutions f de $(\mathcal{E}_{\lambda,\mu})$ vérifient pour tout $x \in [0, L]$

$$f(x) = (ax + b)e^{\alpha x}$$

où $a, b \in \mathbb{R}$. La condition $f(0) = 0$ nous donne $b = 0$. Puis, la condition $f(L) = 0$ impose $aLe^{\alpha L} = 0$, d'où $a = 0$. Ainsi, seule la fonction nulle est solution.

1.1.4 1.4 Le cas $\Delta > 0$

Supposons $\Delta > 0$. L'équation caractéristique a deux racines réelles distinctes α et β . Les solutions f de $(\mathcal{E}_{\lambda,\mu})$ vérifient pour tout $x \in [0, L]$

$$f(x) = ae^{\alpha x} + be^{\beta x}$$

où $a, b \in \mathbb{R}$. La condition $f(0) = 0$ nous donne $a + b = 0$, d'où $b = -a$. La condition $f(L) = 0$ impose

$$a(e^{\alpha L} - e^{\beta L}) = 0$$

d'où $a = 0$ car $\alpha \neq \beta$. Ainsi, seule la fonction nulle est solution.

1.1.5 1.5 Le cas $\Delta < 0$

Supposons $\Delta < 0$. L'équation caractéristique a deux racines non réelles conjuguées $p \pm i\omega$ où $p \in \mathbb{R}$ et $\omega > 0$. Les solutions f de $(\mathcal{E}_{\lambda,\mu})$ vérifient pour tout $x \in [0, L]$

$$f(x) = ae^{px} \cos \omega x + be^{px} \sin \omega x$$

La condition $f(0) = 0$ nous donne $a = 0$. La condition $f(L) = 0$ impose alors $b \sin \omega L = 0$. Si on prend $b = 0$ on retrouve la fonction nulle.

Si, en revanche, $b \neq 0$, le réel ω vérifie

$$\sin \omega L = 0$$

Cette égalité est vérifiée si et seulement si il existe $k \in \mathbb{N}^*$ tel que

$$\omega = \frac{k\pi}{L}$$

Les solutions non nulles sont ainsi les fonctions définies pour $x \in [0, L]$ par

$$f(x) = be^{px} \sin \frac{k\pi x}{L}$$

où $p \in \mathbb{R}$, $b \in \mathbb{R}^*$ et $k \in \mathbb{N}^*$.

1.2 2. Courbes

1.2.1 2.1 Fonctions utilitaires

La fonction `subdi` renvoie une subdivision régulière à $n + 1$ points du segment $[a, b]$.

```
[2]: def subdi(a, b, n):
      h = (b - a) / n
      return [a + k * h for k in range(n + 1)]
```

La fonction $f : x \mapsto be^{px} \sin \frac{k\pi x}{L}$.

```
[3]: def f(L, b, p, k, x):
      return b * math.exp(p * x) * math.sin(k * math.pi * x / L)
```

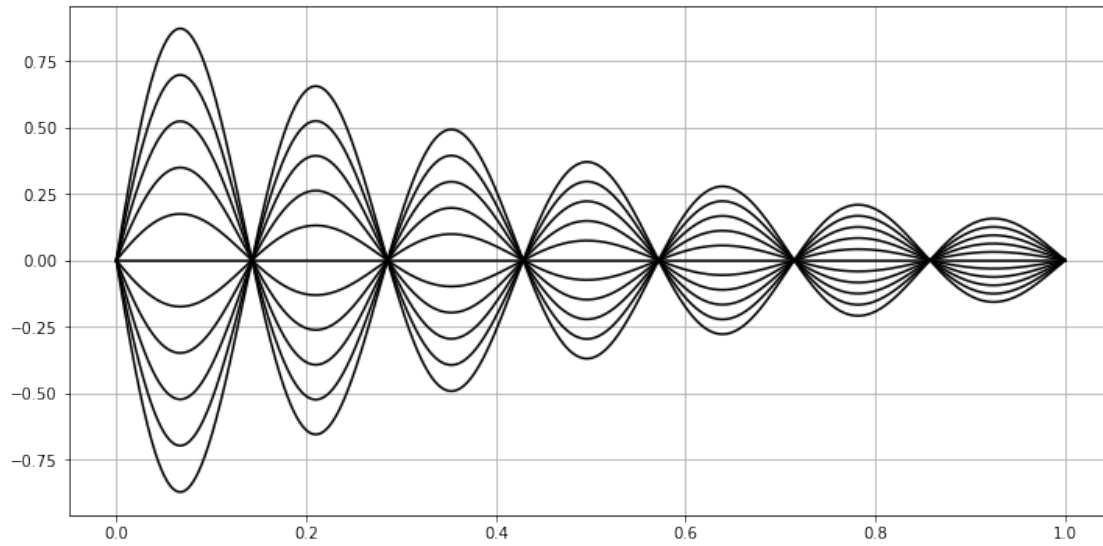
1.2.2 2.2 b variable, k, p fixés

On trace $f(L, b, p, k, x)$ pour k et p fixés et b variant dans la liste de réels `bs`.

```
[4]: def tracer1(L, p, k, bs):
      xs = subdi(0, L, 1000)
      for b in bs:
          ys = [f(L, b, p, k, x) for x in xs]
          plt.plot(xs, ys, 'k')
      plt.grid()
```

```
[5]: plt.rcParams['figure.figsize'] = (12, 6)
```

```
[6]: bs = subdi(-1, 1, 10)
      tracer1(1, -2, 7, bs)
```



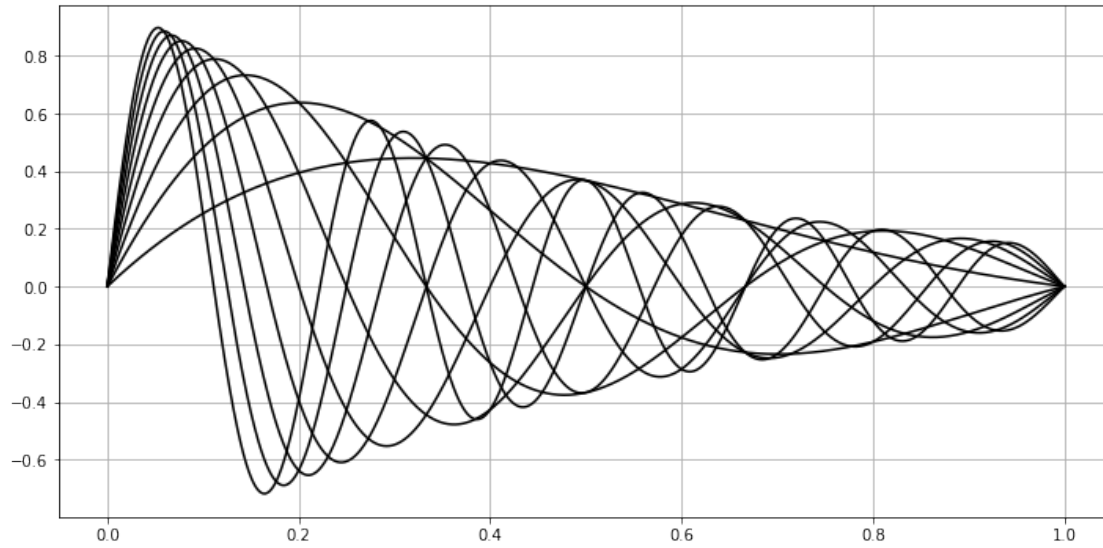
1.2.3 2.3 k variable, b, p fixés

On trace $f(L, b, p, k, x)$ pour b et p fixés et k variant dans la liste de réels ks .

```
[7]: def tracer2(L, b, p, ks):
      xs = subdi(0, L, 1000)
      for k in ks:
          ys = [f(L, b, p, k, x) for x in xs]
          plt.plot(xs, ys, 'k')
      plt.grid()
```

```
[8]: plt.rcParams['figure.figsize'] = (12, 6)
```

```
[9]: ks = list(range(1, 10))
      tracer2(1, 1, -2, ks)
```



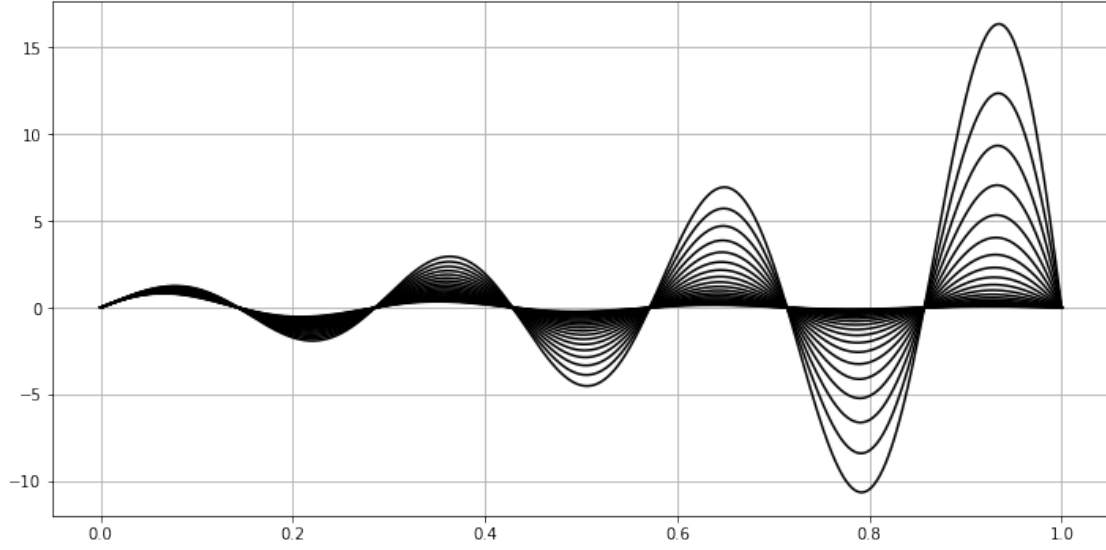
1.2.4 2.4 p variable, k, b fixés

On trace $f(L, b, p, k, x)$ pour b et k fixés et p variant dans la liste de réels ps .

```
[10]: def tracer3(L, b, k, ps):
      xs = subdi(0, L, 1000)
      for p in ps:
          ys = [f(L, b, p, k, x) for x in xs]
          plt.plot(xs, ys, 'k')
      plt.grid()
```

```
[11]: plt.rcParams['figure.figsize'] = (12, 6)
```

```
[12]: ps = subdi(-3, 3, 20)
      tracer3(1, 1, 7, ps)
```



1.2.5 2.5 Les couples (λ, μ) admissibles

Quels sont précisément les couples $(\lambda, \mu) \in \mathbb{R}^2$ pour lesquels le problème $(\mathcal{P}_{\lambda, \mu})$ a une solution non identiquement nulle ? Une première condition est $\Delta = \lambda^2 - 4\mu < 0$. Posons $\Delta = -4\omega^2$, où

$$\omega = \sqrt{\mu - \frac{1}{4}\lambda^2}$$

Les racines de l'équation caractéristique sont ainsi

$$-\frac{1}{2}\lambda \pm i\omega$$

Il y a une solution non nulle au problème si et seulement si il existe $k \in \mathbb{N}^*$ tel que $\omega = \frac{k\pi}{L}$, d'où

$$\Delta = -\frac{4k^2\pi^2}{L^2}$$

En remplaçant Δ par sa valeur, on obtient ainsi la condition nécessaire et suffisante

$$L^2(\lambda^2 - 4\mu) + 4k^2\pi^2 = 0$$

ou encore

$$(\mathcal{C}) \quad \mu = \frac{\lambda^2}{4} + \frac{k^2\pi^2}{L^2}$$

Pour tout $\lambda \in \mathbb{R}$ et tout $k \in \mathbb{N}^*$ il existe un unique $\mu \in \mathbb{R}$ qui convienne. Remarquons enfin que, k étant fixé, les couples (λ, μ) qui nous intéressent sont sur une parabole.

La fonction `tracer_admissibles` trace les couples (λ, μ) qui conduisent à une solution non triviale du problème $(\mathcal{P}_{\lambda, \mu})$. Par la remarque ci-dessus, on obtient donc un ensemble de paraboles.

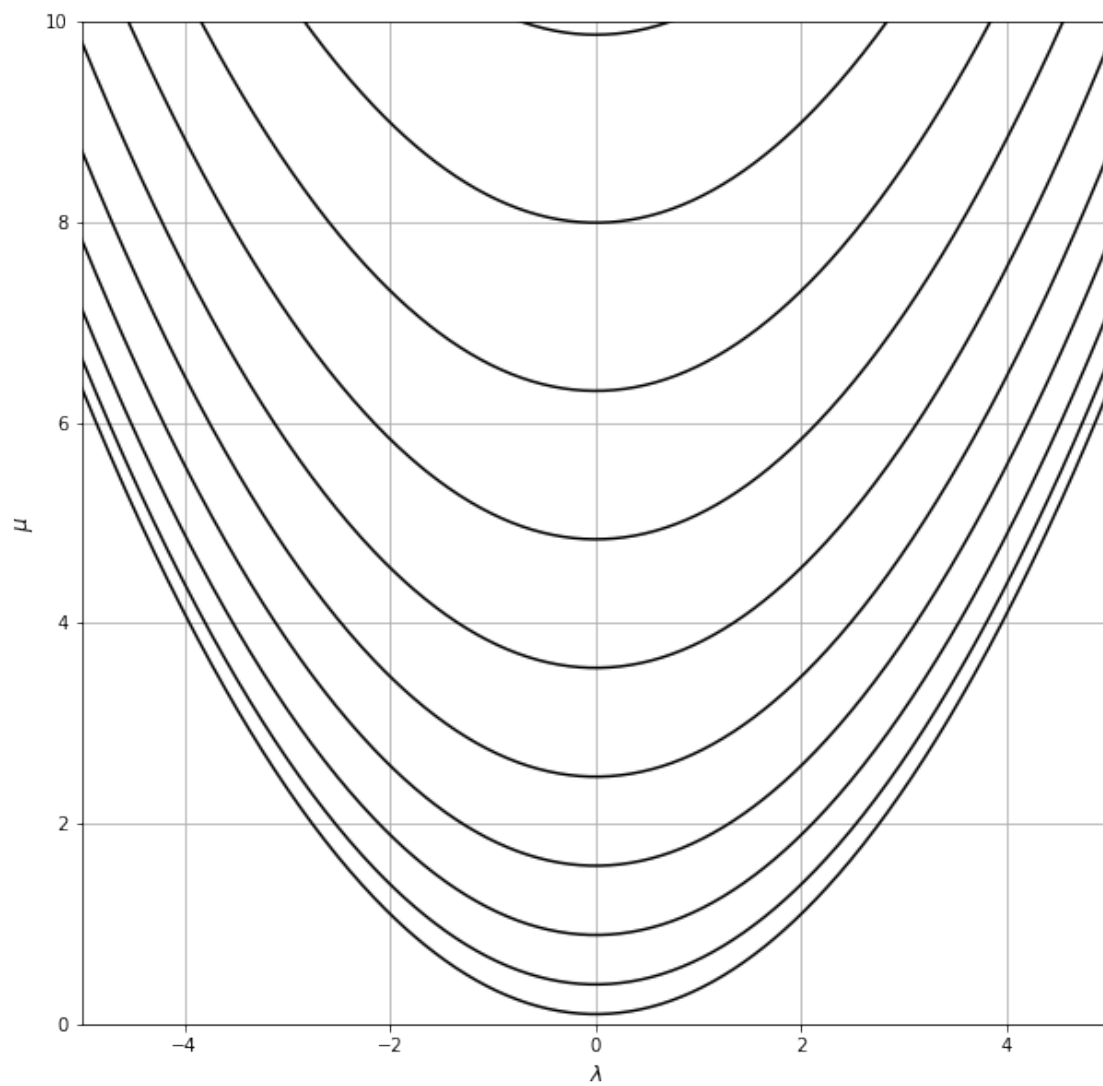
Pour des raisons esthétiques on a pris $L = 10$, cela permet d'avoir des paraboles dont la courbure est plus prononcée.

```
[13]: def phi(l, k, L):  
      return l ** 2 / 4 + k ** 2 * math.pi ** 2 / L ** 2
```

```
[14]: def tracer_admissibles():  
      L = 10  
      lambdas = subdi(-5, 5, 500)  
      ks = list(range(1, 11))  
      for k in ks:  
          mus = [phi(l, k, L) for l in lambdas]  
          plt.plot(lambdas, mus, 'k')  
      plt.xlim(-5, 5)  
      plt.ylim(0, 10)  
      plt.xlabel(r'$\lambda$', fontsize=12)  
      plt.ylabel(r'$\mu$', fontsize=12)  
      plt.grid()
```

```
[15]: plt.rcParams['figure.figsize'] = (10, 10)
```

```
[16]: tracer_admissibles()
```



[]: