

Cahier de laboratoire

Travaux pratiques avancés (GPH-3000)

présenté à
Simon Rainville et Michel Piché

par
Maxence Larose (111 188 447 - maxence.larose.1@ulaval.ca)
Pierre-Olivier Janvier (111 187 987 - pierre-olivier.janvier.1@ulaval.ca)

Automne 2020

Expérience 4.2 : Laser à fibre femtoseconde (Simulation)

Préparation

Date de préparation : 15 octobre 2020

Date de l'expérimentation : 15 octobre 2020

But : Optimiser les réglages d'un système P-APM et obtenir des résultats en régime solitonique avec le code de simulation pour éventuellement comparer avec les résultats expérimentaux du laser à fibre dopée erbium du laboratoire.

Objectifs spécifiques:

- Optimiser les réglages du système P-APM en maximisant la pente à l'origine de la fonction de transmission du système.
- Utiliser les conditions optimales trouvées pour obtenir une impulsion solitonique avec le code de simulation sur MatLab
- Comparer avec les résultats expérimentaux.
- Examinez les résultats obtenus avec des réglages non optimisés des paramètres du système P-APM.

Matériel nécessaire:

- Code de simulation de laser femtoseconde sur MatLab
- Thèse de Michel Olivier

Partie 1 : Optimisation du réglage du système P-APM.

Afin d'optimiser les réglages du système, la pente à l'origine de la fonction de transmission du système P-APM doit être maximisée. Ainsi, on utilise l'expression analytique de la fonction de transmission suivante

$$\frac{P_{\text{out}}}{P_{\text{in}}} = \frac{|A_x^{\text{out}}|^2}{|A_x^{\text{in}}|^2} = \frac{1}{2}(1 + \cos(2\alpha) \cos[2(\alpha - 2\theta) + \Delta\phi]),$$

où on a que l'angle de rotation non linéaire est donné par

$$\Delta\phi = -\frac{\gamma L}{3} \sin(2\alpha) P.$$

La seconde équation est d'abord substituée dans l'équation de transmission. Ensuite, la dérivée de la première équation par rapport à P est réalisée et les valeurs des paramètres α et θ qui maximise cette dérivée lorsque $P = 0$ sont déterminées de manière analytique à l'aide des calculs suivants.

```
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
from math import cos, sin, pi
from sympy import *
```

```
gamma_SMF = 1.4e-3
L_SMF = 3
gamma_HI = 3.16e-3
L_HI = 0.5
```

```

gamma_DRAKA = 4.6e-3
L_DRAKA = 1

gammaL = gamma_SMF*L_SMF + gamma_HI*L_HI + gamma_DRAKA*L_DRAKA
alpha, theta, P = symbols('alpha theta P', real=True)

T = 1/2*(1+cos(2*alpha)*cos(2*(alpha-2*theta)-((gammaL)/3)*sin(2*alpha)*P))
dT_symbol = diff(T, P)
dT = lambdify([alpha, theta, P], dT_symbol)

alpha = np.linspace(-2*np.pi, 2*np.pi, 1000)
theta = np.linspace(-2*np.pi, 2*np.pi, 1000)

A, T = np.meshgrid(alpha, theta)
Z = dT(A, T, P=0)
i, j = np.where(Z >= (np.max(Z) - 0.0000001))

max_pairs = [pairs for pairs in zip(alpha[j], theta[i])]

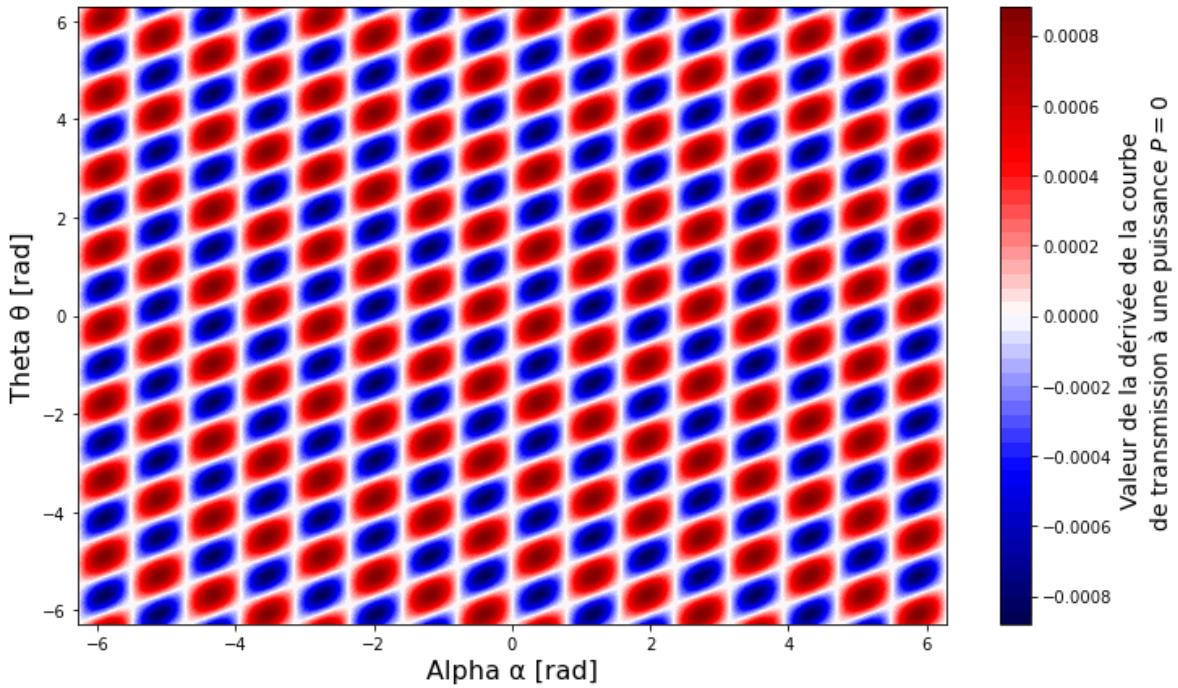
# print('Max at \n', np.asarray(max_pairs))

print('Max used for simulation is  $\alpha = {:.5f}$  rad,  $\theta = {:.5f}$ 
rad.'.format(0.39623691, -0.19497372))

plt.contourf(A, T, Z, cmap='seismic', levels=50)
plt.xlabel('Alpha  $\alpha$  [rad]', fontsize=16)
plt.ylabel('Theta  $\theta$  [rad]', fontsize=16)
# plt.xticks([-2*np.pi, -np.pi, 0, np.pi, 2*np.pi], ['- $\pi$ ', '- $\pi$ ', '0',
'$\pi$', '2$\pi$'], fontsize=16)
# plt.yticks([-2*np.pi, -np.pi, 0, np.pi, 2*np.pi], ['- $\pi$ ', '- $\pi$ ', '0',
'$\pi$', '2$\pi$'], fontsize=16)
cbar = plt.colorbar()
cbar.set_label(label='Valeur de la dérivée de la courbe \n de transmission à une
puissance  $P=0$$ ', fontsize=14)
fig = plt.gcf()
fig.set_size_inches(12, 7)
fig.savefig('alpha_theta.pdf', bbox_inches='tight', dpi=600)
plt.show()

```

Max used for simulation is $\alpha = 0.39624$ rad, $\theta = -0.19497$ rad.



Les paramètres utilisés pour la simulation sont donc $\alpha = 0.39624$ rad et $\theta = -0.19497$ rad. D'autres maxima de la figure précédente auraient pu être utilisés. Par ailleurs, la section suivante présente les calculs reliés aux paramètres de la fibre SMF-28 utilisé lors des expérimentations au laboratoire.

```

lambda0 = 1559.5e-9
c = 3e8
Dmode = 10.4e-6
Aeff = np.pi*(Dmode/2)**2
v0 = c/lambda0
gamma = (2*np.pi*2.6e-20*v0)/(c*Aeff)

print('gamma = ', gamma)

D = 18e-06 # s/m^2

beta = -(D*lambda0**2)/(2*np.pi*c)

print('beta = ', beta)

```

```

gamma =  0.0012331368535280043
beta =  -2.322427365515694e-26

```

Partie 2 : Convergence vers un soliton.

Ensuite, afin de pouvoir plus tard comparer les résultats de simulation avec ceux expérimentaux, le profil d'un soliton a été obtenu avec le simulateur matlab fourni et à l'aide des paramètres α et θ optimaux trouvés dans la partie 1. Ainsi, avec un $\alpha = 0.39624$ rad et $\theta = -0.19497$ rad on obtient la figure suivante

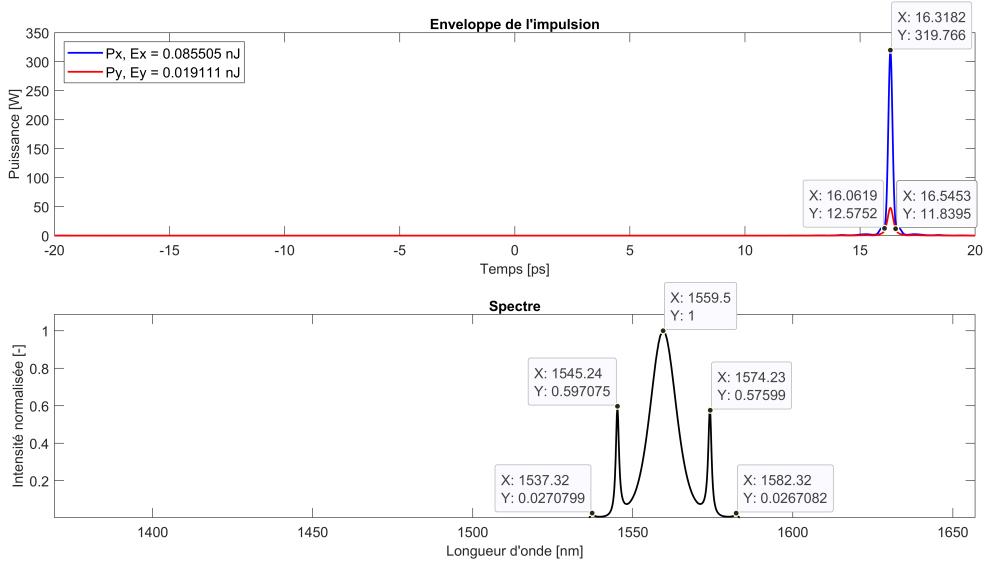


FIGURE 1 : Profil d'un soliton obtenu avec les conditions optimales pour α et θ .

Les différentes données obtenues lors de la simulation ont aussi été enregistrées et sont disponibles dans un fichier nommé `alpha_0.3962rad_theta_-0.195rad_lambda_1.5595e-06nm.mat` qui correspondent aux paramètres de simulation utilisés. Pour ce qui est de l'analyse de cette figure et des différents résultats découlant de celle-ci, ceux-ci seront présentés lors de la rédaction du rapport.

Partie 3 : Convergence vers un soliton.

Finalement, afin de tester le comportement du système dans des conditions non optimales, différents profils ont été obtenus en changeant les paramètres α et θ de manière à les rendre non optimaux. Les différentes figures obtenues sont présentées ci-dessous accompagnées des paramètres α et θ utilisés.

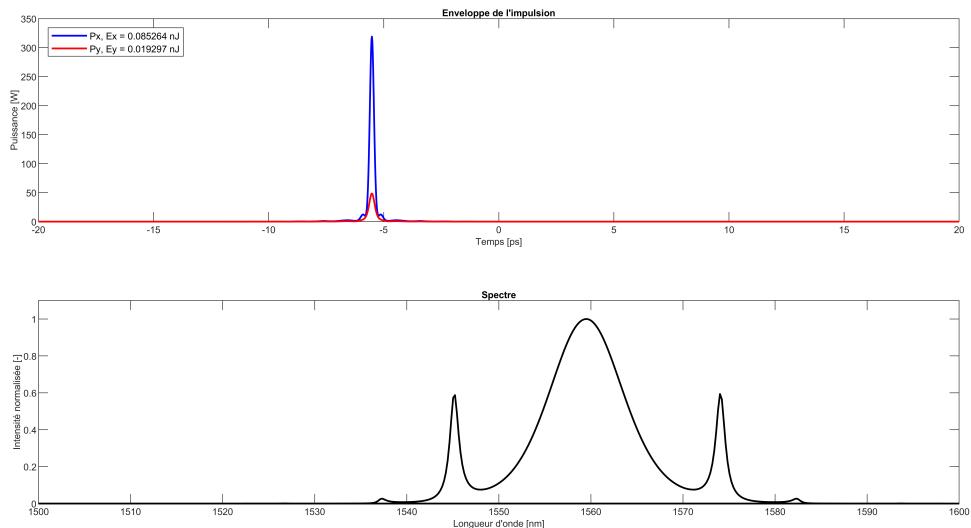


FIGURE 2 : Profil d'un soliton obtenu avec les conditions non optimales pour un α de 0.3987 et un θ de -0.1925.

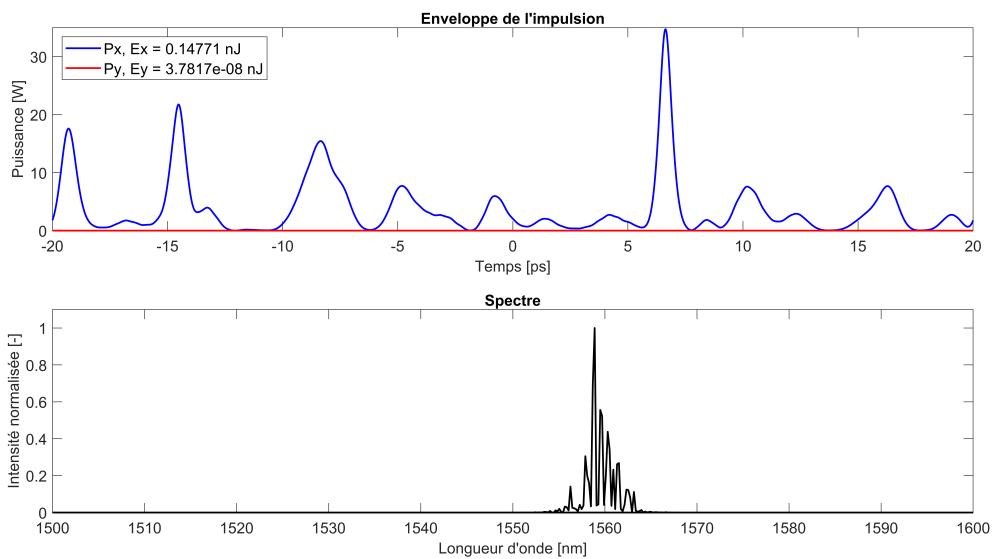


FIGURE 3 : Profil d'un soliton obtenu avec les conditions non optimales pour un α de 0 et un θ de 0.

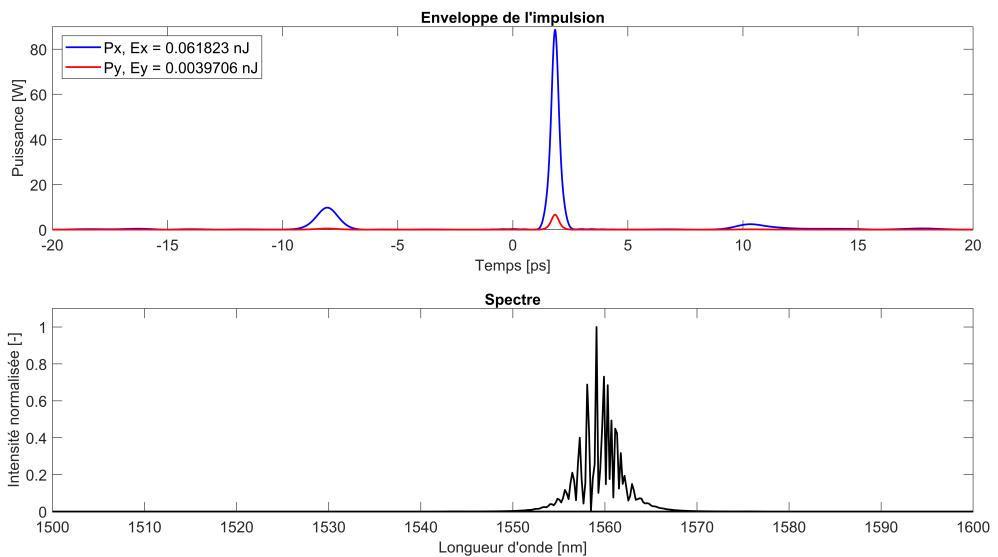


FIGURE 4 : Profil d'un soliton obtenu avec les conditions non optimales pour un α de 1.42 et un θ de 2.597.

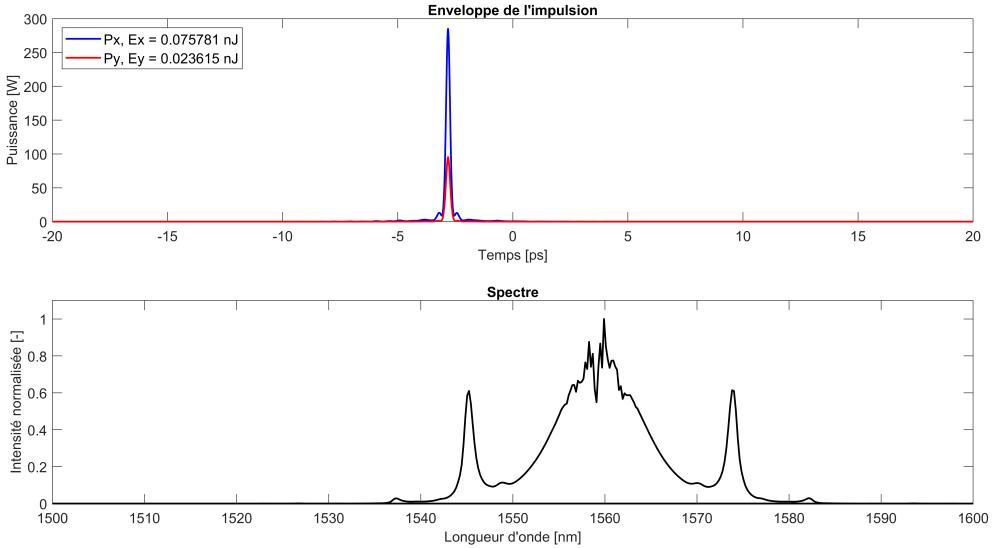


FIGURE 5 : Profil d'un soliton obtenu avec les conditions non optimales pour un α de 1.358 et un θ de 1.07.

Comme il est possible de remarquer dans les figures 2 à 5, les profils changent radicalement selon les paramètres α et θ . En effet, certain d'entre eux ne semblent pas présenter de comportement associé à un régime solitonique. Cela peut être causé par plusieurs choses. Une raison évidente serait que la simulation demanderait plus d'itérations avant de converger et finirai ainsi par démontrer les caractéristiques d'un tel régime. Par contre, en comparant, par exemple, la figure 2 avec la figure 1 on peut remarquer qu'il est tout de même possible d'obtenir un soliton même en utilisant des paramètres non optimaux. De plus, il est aussi possible de remarquer que peu importe la forme de l'impulsion résultante, la puissance pic de celle-ci varie grandement et proportionnellement selon la qualité du soliton obtenu.