学习笔记 | Nonlinear Schrödinger equation and split-step Fourier method solution - (1)

Zicong Jiang

Department of Electrical Engineering, Chalmers University of Technology, Sweden

zicongj@chalmers.se

Contents

1. Abstract

The fiber channel modeling is a crucial topic in optical fiber communication. An accurate fiber model allows the telecommunication enginners and researchers to design the optimal transmitter based on current fiber settings. In this blog. I try to record the derivation process of the nonlinear Schrödinger equation (NLSE) and how we can get the numerical method, split-step Fourier method (SSFM), to get the closed-form solution.

2. Numerical method - Split-Step Fourier Method

The NLSE is a partial differential equation (PDE) that denotes the input–output relation for optical signals that propagate through the entire optical fiber. The NLSE can be written as [1]

$$\frac{\partial v(t,z)}{\partial z} = -\frac{\alpha}{2}v(t,z) - i\frac{\beta_2}{2}\frac{\partial^2 v(t,z)}{\partial t^2} + i\gamma v(t,z)|v(t,z)|^2. \tag{1}$$

Where:

- α is the attenuation coefficient,
- β_2 is the chromatic dispersion coefficient,
- γ is the nonlinear Kerr parameter.
- $v(t,z)$ is the tramsmitted signal as a function of time t and distance z.
- i is the imaginary mark.

If we can solve the this PDE, we can easily investigate how signal varies in space-time domain through the fiber channel. Unfortunally, we can get analytical solution only in some specific case, which I will introduce in the future blog. Therefore, in order to get the solution for current fiber, we can use the numerical method called split-step Fourier method (SSFM) to get a close-form solution for NLSE. In SSFM, same as the name, the fiber will be divided into small segments like in Fig. 1 below.

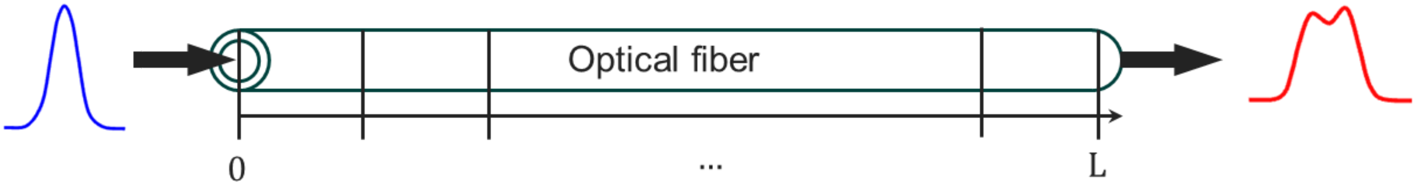

Fig. 1 Split-step method

In this case, we can isolate nonlinearity and dispersion effects individually, and solve them seperately to get the solution for a specific segment. If we do this repeatly for all segments, we can get solution at the output position of fiber.

2.1 Absence of Nonlinear effects

When we neglect the nonlinear term, the (1) can be writen into this format,

$$\frac{\partial v(t,z)}{\partial z} = -\frac{\alpha}{2}v(t,z) - i\frac{\beta_2}{2}\frac{\partial^2 v(t,z)}{\partial t^2}. \tag{2}$$

Which can be solved analytically by first transforming (2) into the Fourier domain using correspondence $\frac{\partial^n}{\partial t^n}v(t) \multimap \bullet (i2\pi f)^n V(f)$. We do the Fourier transform to (2), then we can obtain

$$\frac{\partial V(f,z)}{\partial z} = -\frac{\alpha}{2}V(f,z) + i\frac{\beta_2}{2}(2\pi f)^2 V(f,z), \tag{3}$$

We know the ordinary differential equation (ODE) in this format $\frac{\partial y}{\partial z} = ay$ can be solved and get solution $y = e^{az} y_0$. Where y_0 is the initial condition of y. The $(-\frac{\alpha}{2} + i\frac{\beta_2}{2}(2\pi f)^2)$ is the constant, so we can write (3) as

$$\begin{aligned} V(f,z) &= e^{(-\frac{\alpha}{2}+i\frac{\beta_2}{2}(2\pi f)^2)z} V(f,0) \\ &= H(f,z)V(f,0), \end{aligned}. \tag{4}$$

Where $H(f,z) = e^{(-\frac{\alpha}{2}+i\frac{\beta_2}{2}(2\pi f)^2)z}$ is the frequency response of a dispersion filter.

2.2 Absence of Dispersion

Similarly, when we neglect the linear term, the (1) can be expressed as

$$\frac{\partial v(t,z)}{\partial z} = -i\gamma v(t,z)|v(t,z)|^2. \tag{5}$$

Then, we can use the same ODE solution format to get the solution (4),

$$v(t,z) = e^{-i\gamma|v(t,z)|^2 z} v(t,z). \tag{6}$$

2.3 SSFM implementation

Now we already get the solution for nonlinear and linear term in (4) and (6). Then we can get the numerical solution for fiber segment by calculating the solution iterativly.

$$v(t,\Delta) = F^{-1}[H(f,z)V(f,0)]e^{-i\gamma|v(t,0)|^2\Delta}, \tag{7}$$

where Δ is the segment length, $F^{-1}[\cdot]$ is the inverse Fourier transform.

> The python code for SSFM is shown below:

```
def ssfm(gamma, x, dt, dz, num_steps, beta2, alpha):
    f = np.fft.fftfreq(x.shape[1], dt)
    omega = (2 * np.pi * f)
    dispersion = onp.exp(-alpha / 2.0 * dz + 1j * 0.5 * beta2 * (omega **
    for _ in range(num_steps):
        x = np.fft.ifft(np.fft.fft(x) * dispersion)
        x = x * np.exp(1j * gamma * dz * (np.abs(x) ** 2))
    return x
```

3. References

[1] Häger, Christian. On signal constellations and coding for long-haul fiber-optical systems. Chalmers Tekniska Hogskola (Sweden), 2014.