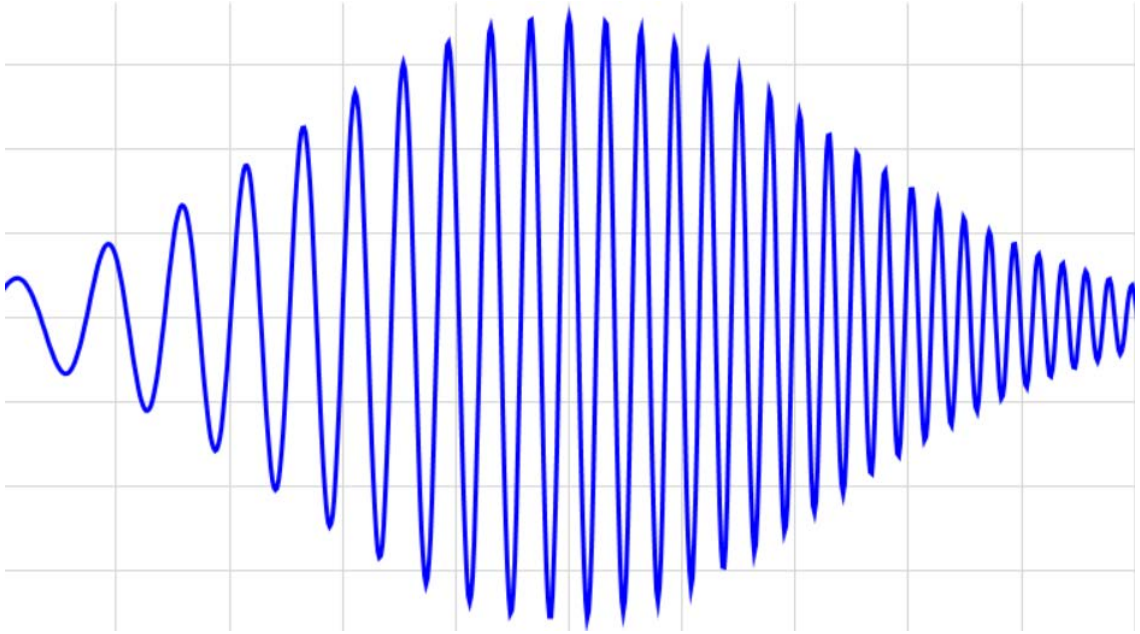
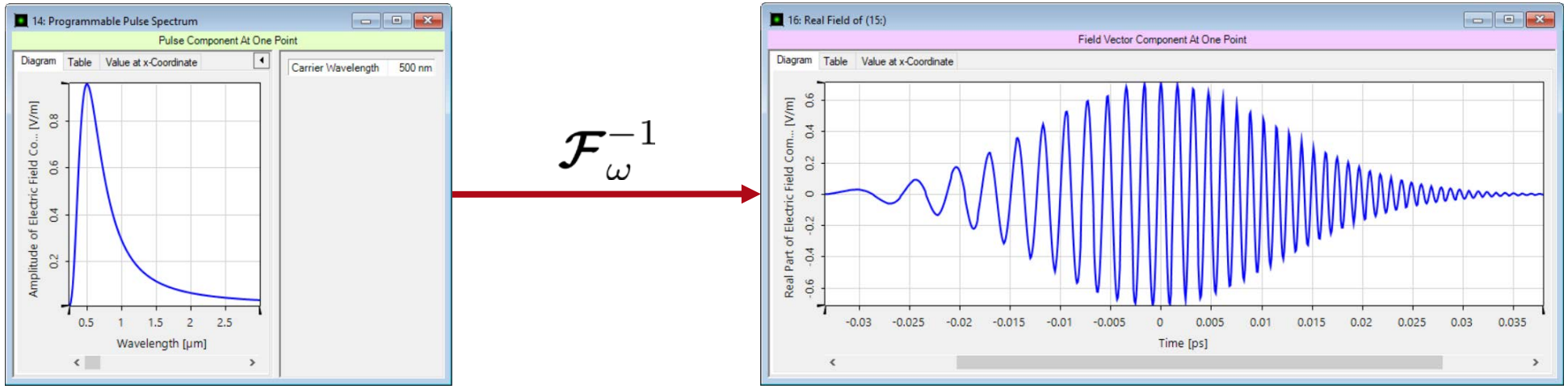
Programming a Chirped Gaussian Pulse Spectrum

Abstract



VirtualLab Fusion provides the freedom to define the pulse spectrum, using the Programmable Pulse Spectrum. The generated spectrum can be used in combination with existing spatial source models. Users can not only specify a spectrum directly in frequency domain, but also possible in time domain, and VirtualLab will automatically calculate the corresponding spectrum. This example shows how to generate a chirped Gaussian pulse, with the specification given in time domain.

Task Description



- Specify a chirped Gaussian pulse in time domain by programming

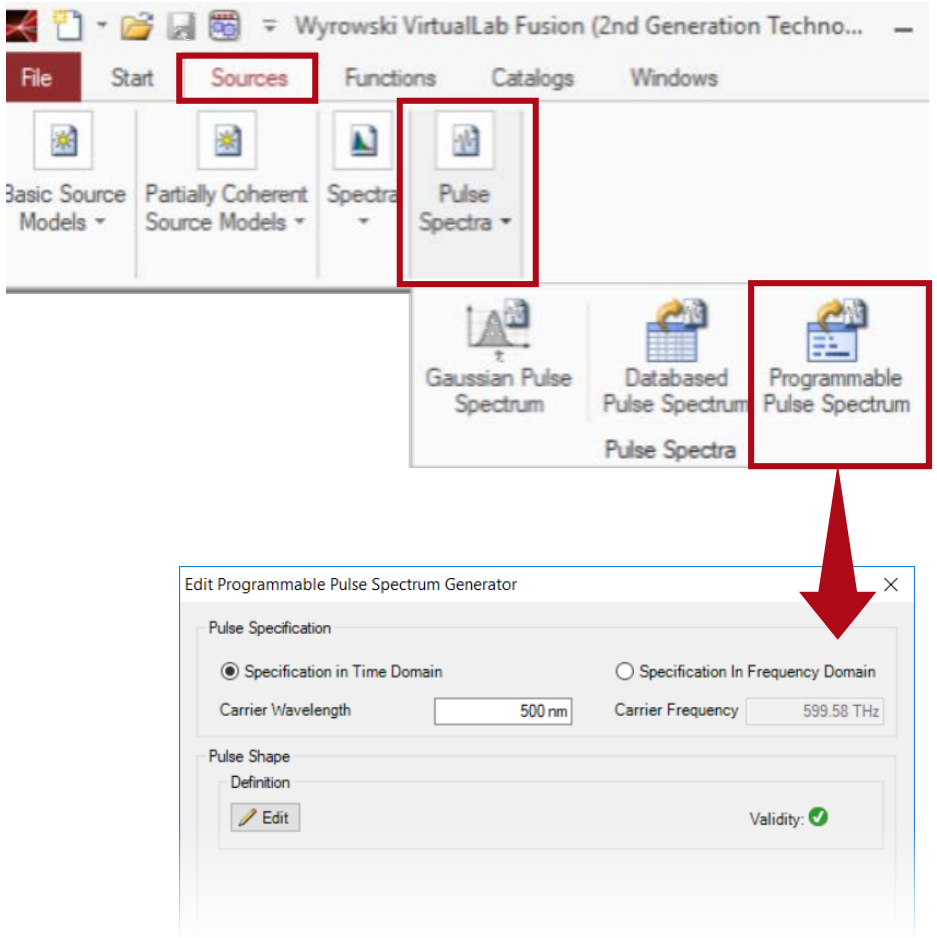
$$E(t) = \exp(-at^2)\exp[i(\omega_0 t + bt^2)] \tag{1}$$

with $a = \frac{2\ln 2}{\tau_p^2}$, where τ_p denoting the pulse duration, ω_0 is carrier frequency, and b denoting the chirp coefficient.

- The Programmable Pulse Spectrum returns the relation spectrum of the specified pulse spectrum.

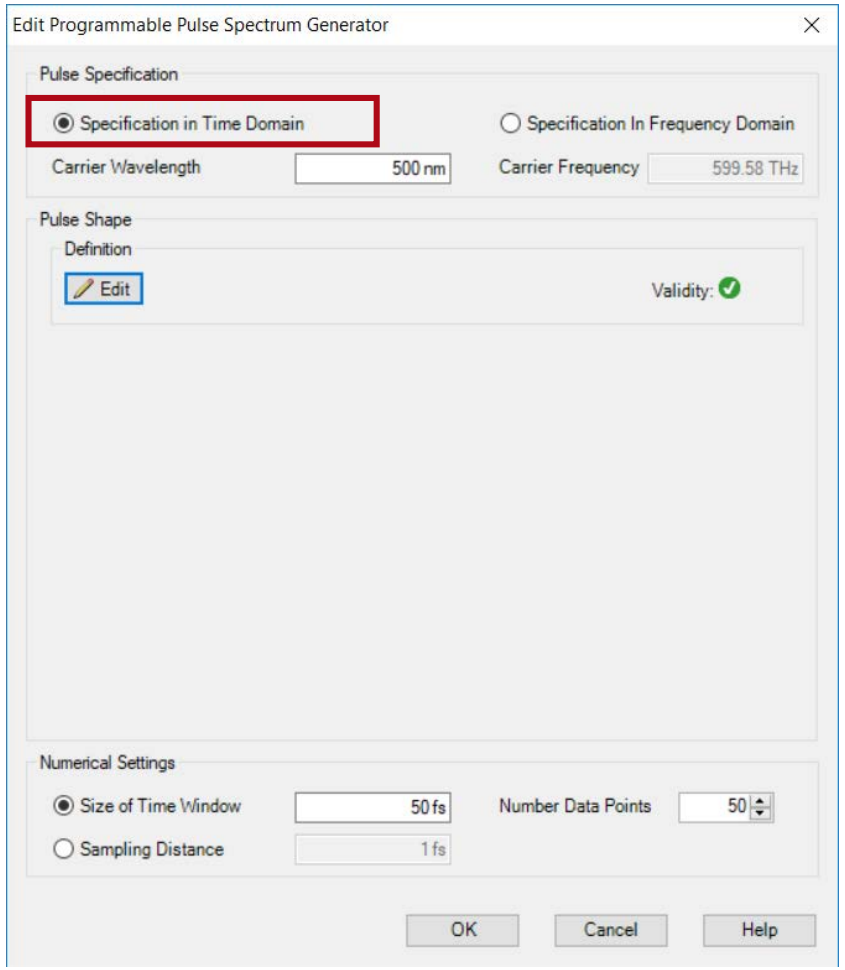
Programmable Pulse Spectrum: Open Edit Window

- Initialization
  - Open the edit window of programmable pulse spectrum



Specification of Domain

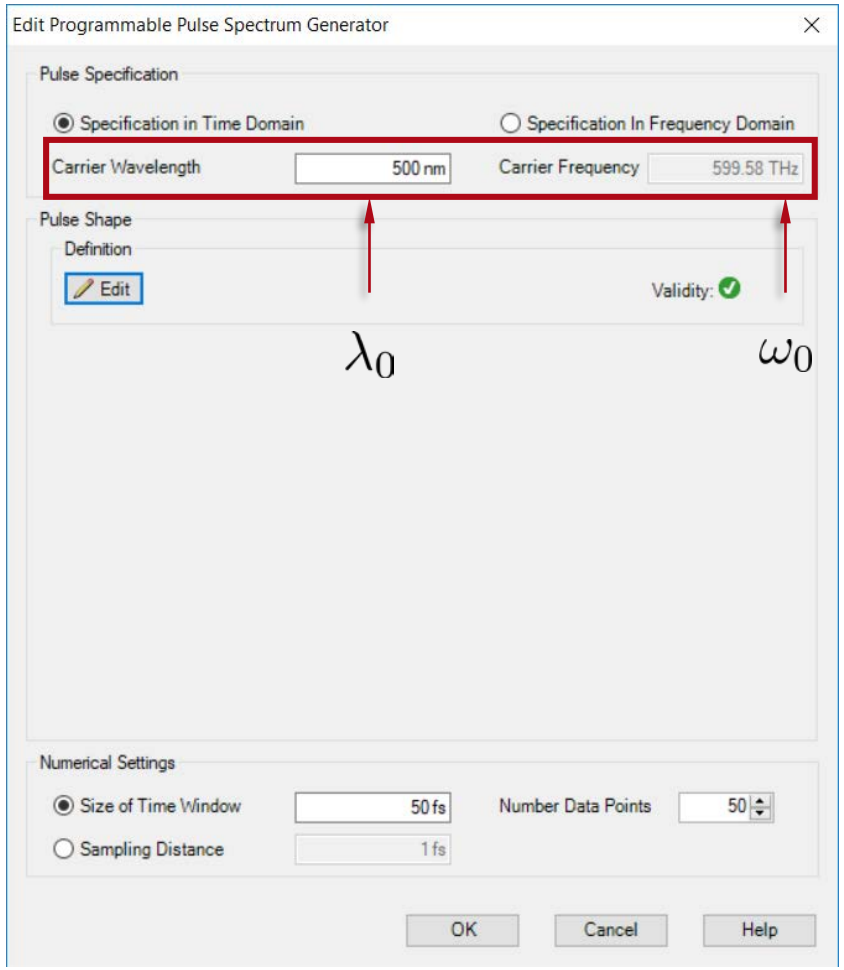
- Initialization
  - Open the edit window of programmable pulse spectrum
  - Select Specification in Time Domain, because the equation of chirped Gaussian pulse in time domain is used to specify the pulse / pulse spectrum.



Define the Carrier Wavelength

- Initialization
  - Open the edit window of programmable pulse spectrum
  - Select Specification in Time Domain, because the equation of chirped Gaussian pulse in time domain is used to specify the pulse / pulse spectrum.
  - Specify Carrier Wavelength in Eq. (1)

$$\lambda_0 = \frac{2\pi c}{\omega_0}$$

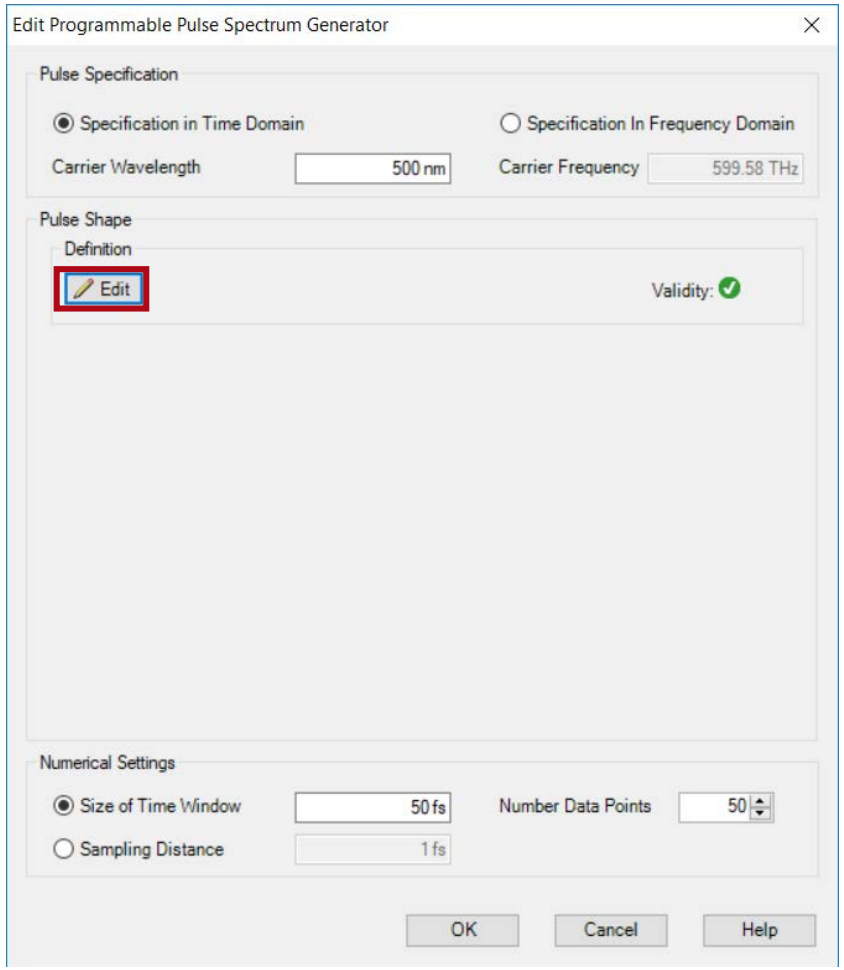


Define the Carrier Wavelength

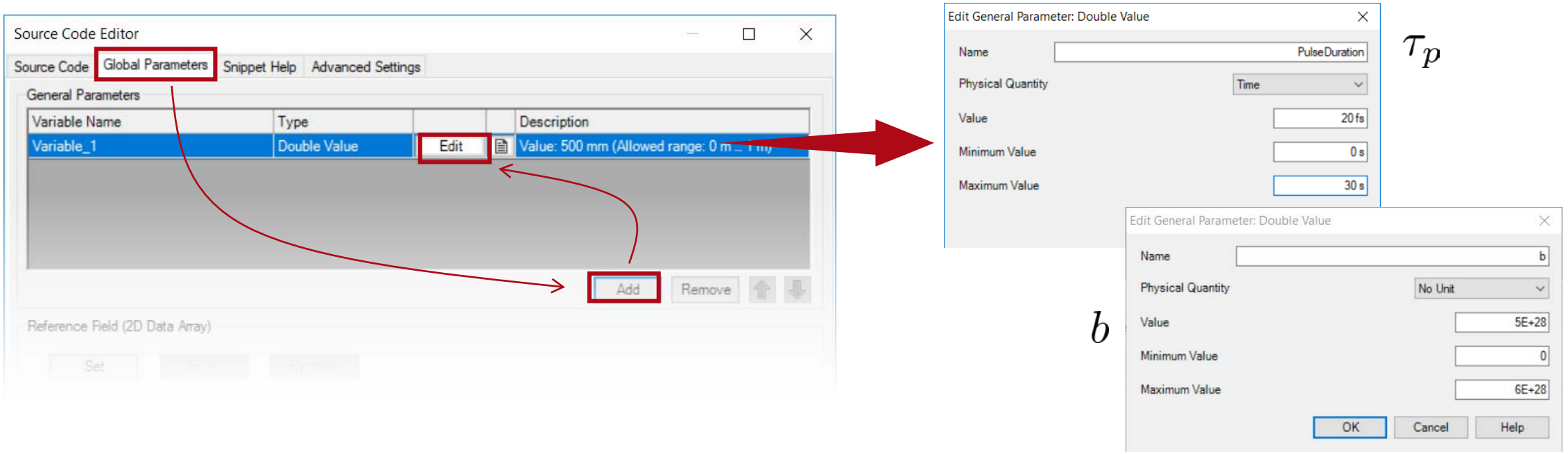
- Initialization
  - Open the edit window of programmable pulse spectrum
  - Select Specification in Time Domain, because the equation of chirped Gaussian pulse in time domain is used to specify the pulse / pulse spectrum.
  - Specify Carrier Wavelength in Eq. (1)

$$\lambda_0 = \frac{2\pi c}{\omega_0}$$

- Programming of pulse



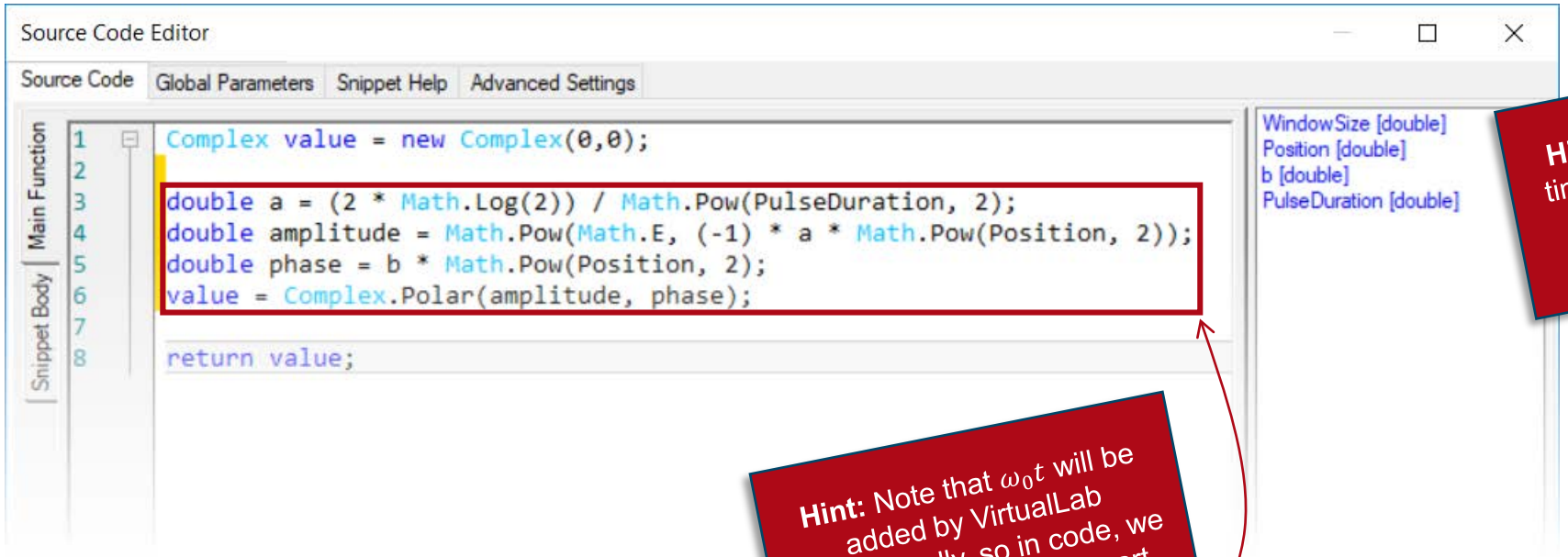
Programming of Pulse: Global Parameters



$$E(t) = \exp(-at^2)\exp[i(\omega_0 t + bt^2)] \tag{1}$$

with $a = \frac{2\ln 2}{\tau_p^2}$, where τ_p denoting the pulse duration, ω_0 is carrier frequency, and b denoting the chirp coefficient.

Programming of Pulse: Source Code



Hint: Note specification in time domain is selected, so here parameter *Position* denotes t

Hint: Note that $\omega_0 t$ will be added by VirtualLab automatically, so in code, we don't need to add this part.

$$E(t) = \exp(-at^2)\exp[i(\omega_0 t + bt^2)] \tag{1}$$

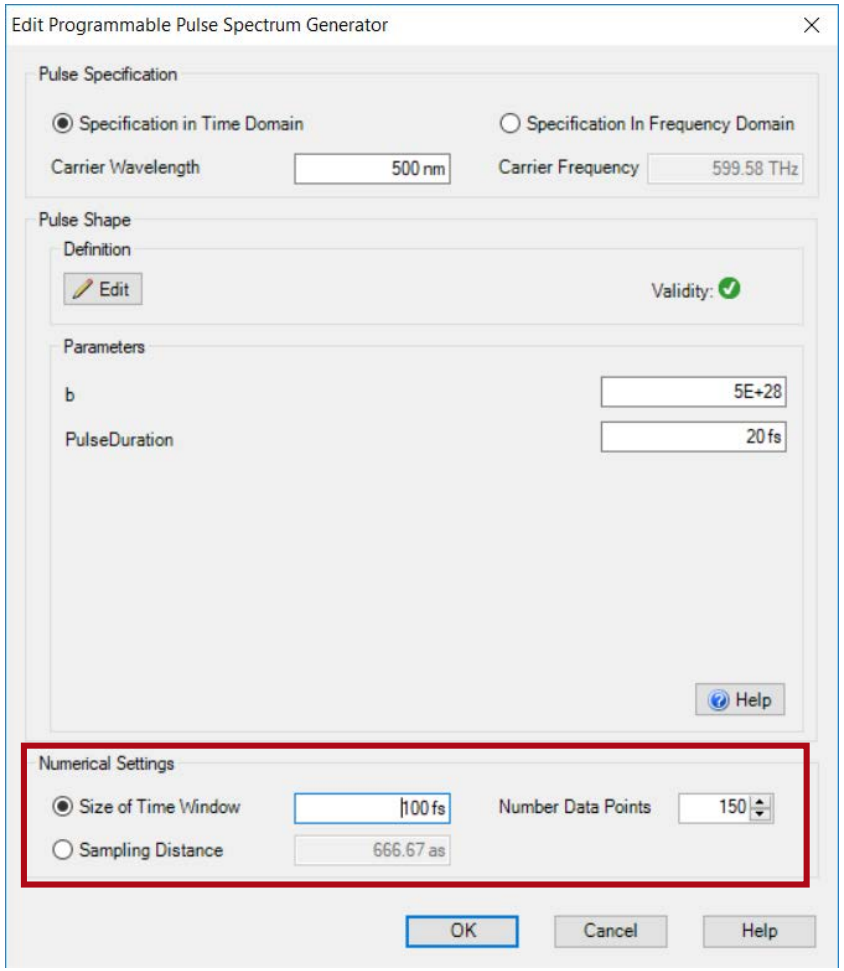
with $a = \frac{2\ln 2}{\tau_p^2}$, where τ_p denoting the pulse duration, ω_0 is carrier frequency, and b denoting the chirp coefficient.

Numerical Settings

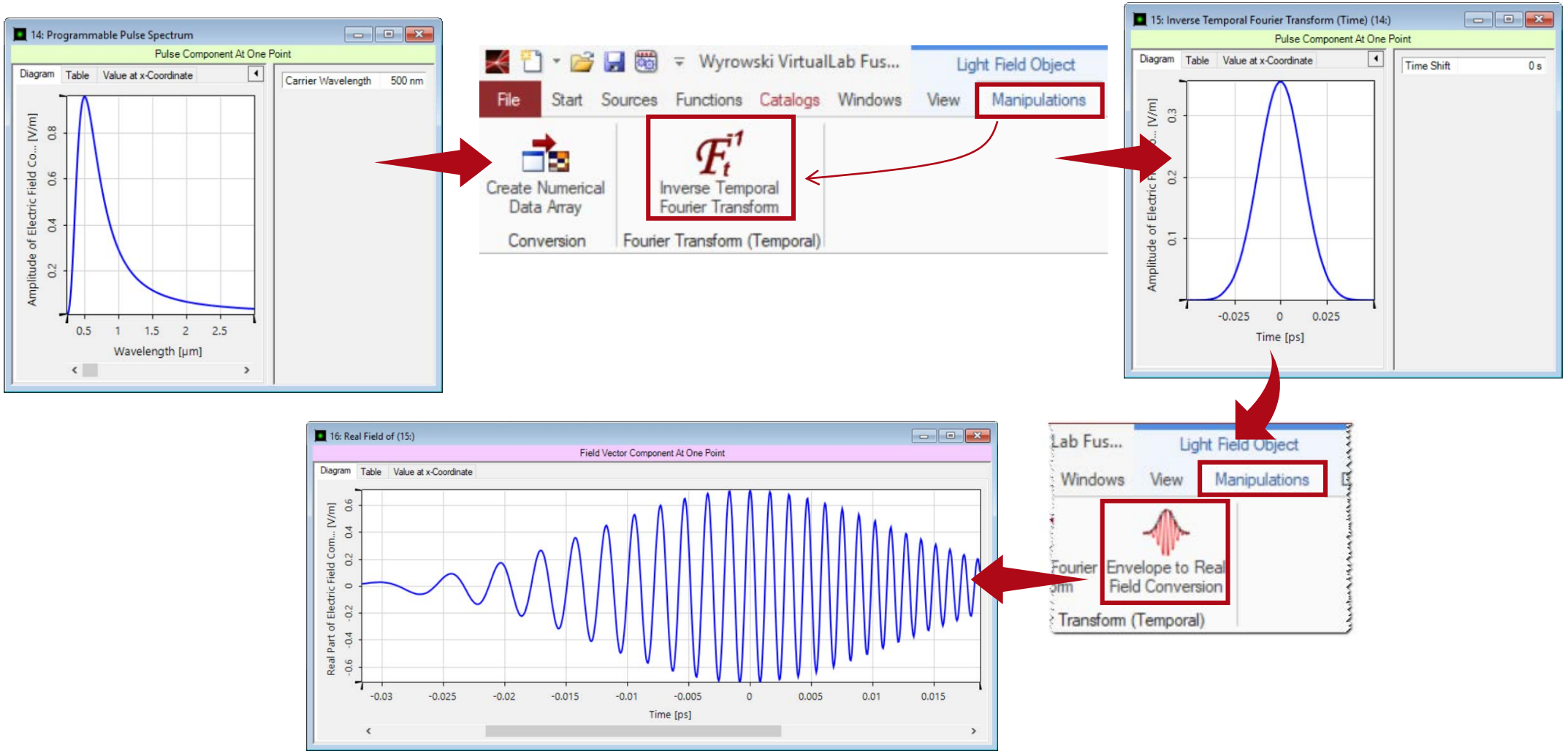
- Initialization
  - Open the edit window of programmable pulse spectrum
  - Select Specification in Time Domain, because the equation of chirped Gaussian pulse in time domain is used to specify the pulse / pulse spectrum.
  - Specify Carrier Wavelength in Eq. (1)

$$\lambda_0 = \frac{2\pi c}{\omega_0}$$

- Programming of pulse
- Numerical Settings
  - Define proper sampling parameters



Results


14: Programmable Pulse Spectrum
Pulse Component At One Point
Diagram
Table
Value at x-Coordinate
Carrier Wavelength
500 nm
Amplitude of Electric Field Co... [V/m]
Wavelength [µm]
Wyrowski VirtualLab Fus...
Light Field Object
File
Start
Sources
Functions
Catalogs
Windows
View
Manipulations
Create Numerical Data Array
Conversion
Inverse Temporal Fourier Transform
Fourier Transform (Temporal)
15: Inverse Temporal Fourier Transform (Time) (14:)
Pulse Component At One Point
Time Shift
0 s
Time [ps]
16: Real Field of (15:)
Field Vector Component At One Point
Real Part of Electric Field Com... [V/m]
Envelope to Real Field Conversion
Transform (Temporal)

Document Information

title	Programming a Chirped Gaussian Pulse Spectrum
document code	CZT.0039
version	1.0
toolbox(es)	Starter Toolbox
VL version used for simulations	7.4.0.49
category	Application use case
further reading	- Programmable Light Source, Function, Interface and Medium