

# Lecture 5

## Multi-Element Propagation

### Part 1:

Short Pulse Fiber Laser

### Part 2:

Mamyshev Oscillator

### Part 3:

Micro Comb Generator



# Multi-Element Propagation: Short Pulse Fiber Lasers

The layout of a typical short pulse oscillator (not only fiber laser) is shown in the image containing several elements in a ring cavity forming a

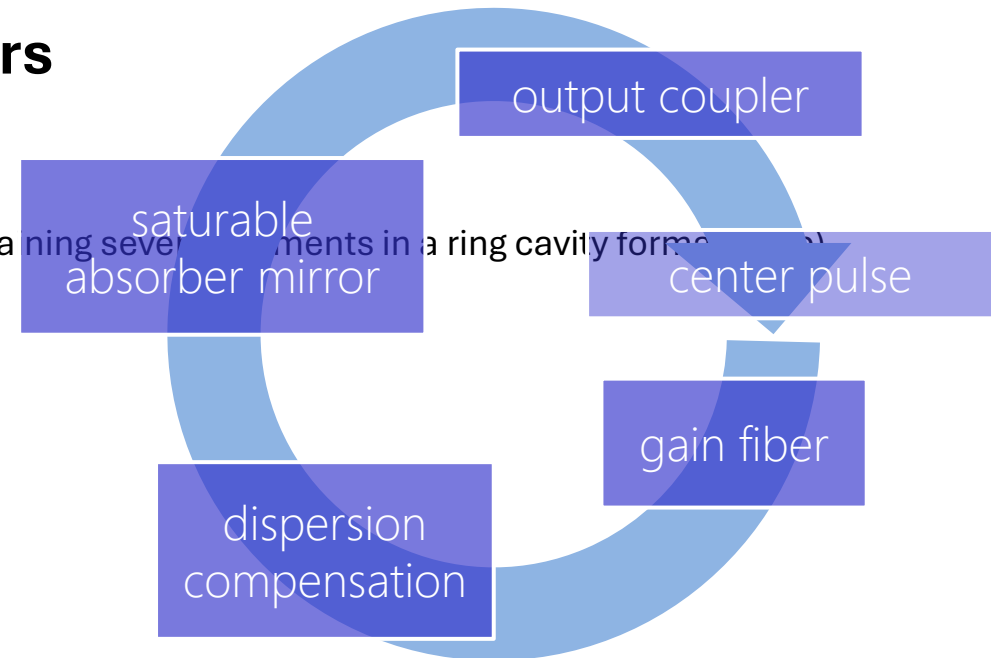
Each Element is modelled by an element typically based on the NLSE:

$$iA_z + \frac{g}{2} A + i\beta_2 A_{tt} = i\gamma |A|^2 A$$

Other elements might be simulated by other means, for example, the saturable absorber mirror is modelled by a fast saturable reflectivity/transmission according to:

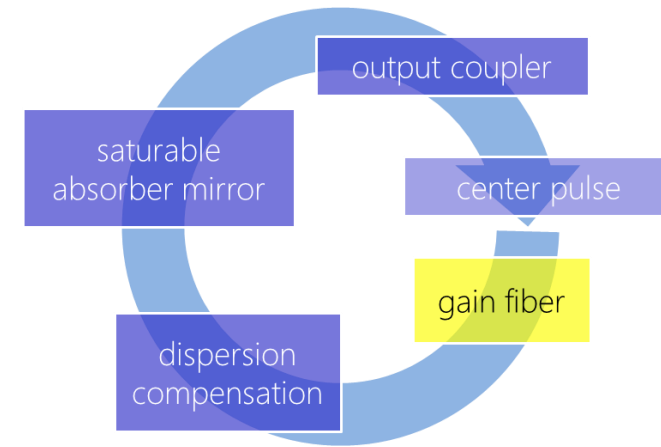
$$R = R_{unsat} + R_{sat} \cdot \left( 1 - \frac{1}{1 + P / P_{sat}} \right)$$

**Each element setup used for this tutorial example is explained in the following >>**



# Multi-Element Propagation: Short Pulse Fiber Lasers

Set up the **gain fiber** as a *standard propagation* with saturable gain. We also use 10 um mode field diameter and a Gaussian gain spectrum of 40 nm width around 1060 nm.



Propagation parameter

standard propagation Setup >

general

define free parameters

measure and parse 100 >

write file

waveguide

loss 0.0 1/m

gain 30 dB/m

mode diameter 10.0 μm

gamma 0.002415094 1/(W m)

effects

dispersion

SPM / TPA

Raman

self-steepening

LLE

numerics

steps 100

stepsize 0.01 m

distance 1.0 m

adaptive local error 1e-5 presets: >

Gain

steady state gain (long pulses to cw)

saturate gain with  $E_{sat} = 1e-11$  J  $g = g_0 / (1 + \frac{E}{E_{sat}})$

profile

gain profile

Center 1060 nm Width 40 nm

shape Gauss

add second peak

Center 1060 nm Width 40 nm

shape const

ratio of second to first peak (set to zero for only one peak): 0

user defined gain file

use ASCII file for gain profile given in  $g(1/m)$  vs. wavelength (separator TAB)

file ...

temporal gain saturation (pulses shorter than population relaxation time)

use temporal gain saturation instead of steady state gain and profiles, with

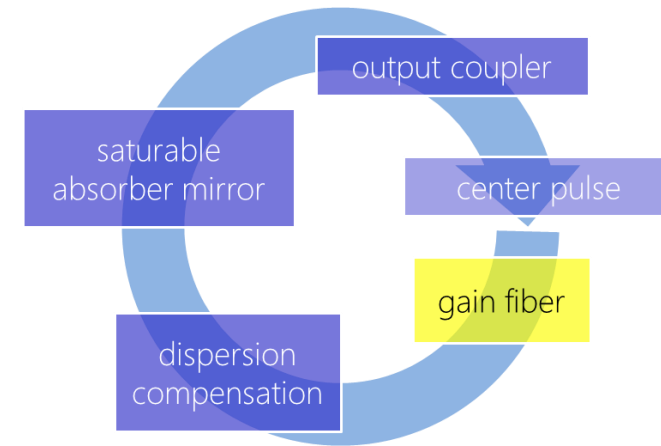
saturation fluence F 30 J/cm<sup>2</sup>  $g(T) = g_0 \exp\left(-\frac{1}{E_{sat}} \int_{-\infty}^T |A(t)|^2 dt\right)$  with  $E_{sat} = F \cdot A_{eff}$

saturation energy with current MFD: J

Back

# Multi-Element Propagation: Short Pulse Fiber Lasers

Now set the dispersion of that gain fiber to the dispersion of fused silica (e.g. predefined with its Taylor series at 1060 nm)



Propagation parameter

Load Save

standard propagation Setup >

general

define free parameters

measure and parse 100 >

write file

waveguide

loss 0.0 1/m

gain 30 dB/m

mode diameter 10.0 μm

gamma 0.002415094 1/(W m)

effects

dispersion

SPM / TPA

Raman

self-steepening

LLE

dispersion term

$$\frac{\partial A}{\partial z} = \dots + \sum_{n \geq 1} \beta_n \frac{i^{n+1}}{n!} \frac{\partial^n}{\partial T^n} A$$

beta 0  
 beta 1  
 beta 2  
 D

dispersion model

Taylor expansion series

Sellmeier coefficients

photonic crystal fiber

gas-filled silica-hollow core fiber

force retarded time frame (beta0=beta1=0) @ data array center wavelength

Use dispersion

do not use dispersion

auto y min -100 max 100 auto x min 350 max 2400

D (ps/nm/km)

Wavelength (nm)

Setup >>

Dispersion Setup

Taylor Series @ 1060 nm predefined more ...

more ...

fused silica @1060nm

Beta1 0 ps/m compensate at:

Beta2 0.01640341019153872 ps<sup>2</sup>/m D -27.49938529

Beta3 4.427598189728069e-5 ps<sup>3</sup>/m S 0.176321

Beta4 -6.116862670739135e-8

Beta5 2.00994101399889e-10

Beta6 -6.78474442990305e-13

Beta7 2.11068254026679e-15

Beta8 -1.1571317270575e-17

Beta9 1.21431865715793e-19

Beta10 -9.78136736258149e-22

Beta11 5.06311226016692e-24

Beta12 -1.65466283477635e-26

Beta13 3.16818781280129e-30

Beta14 -2.7448256015098e-33

Trust region from 400

force retarded time frame (beta0=beta1=0) @ data array center wavelength

grating compressor >>

Save

Load

OK Cancel

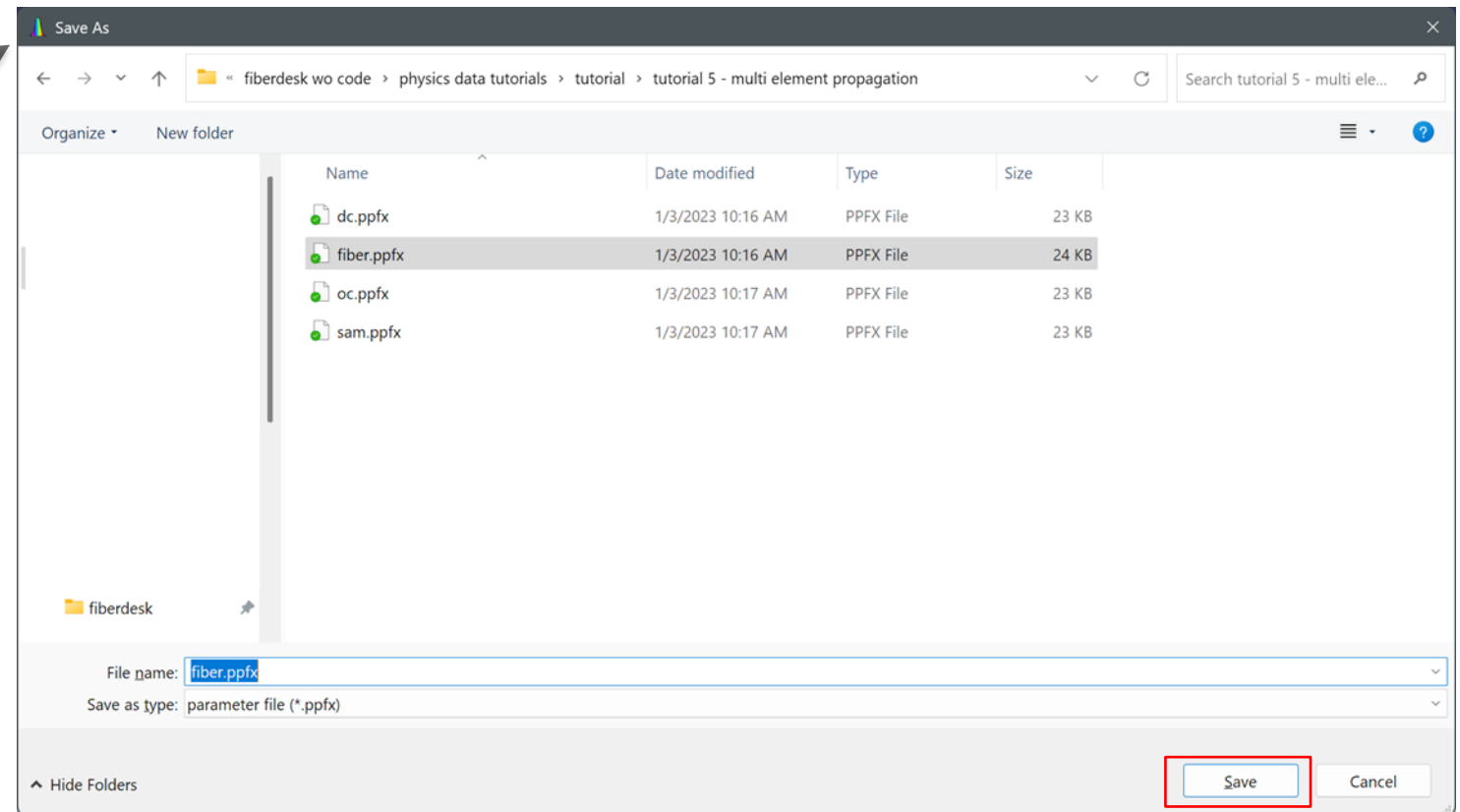
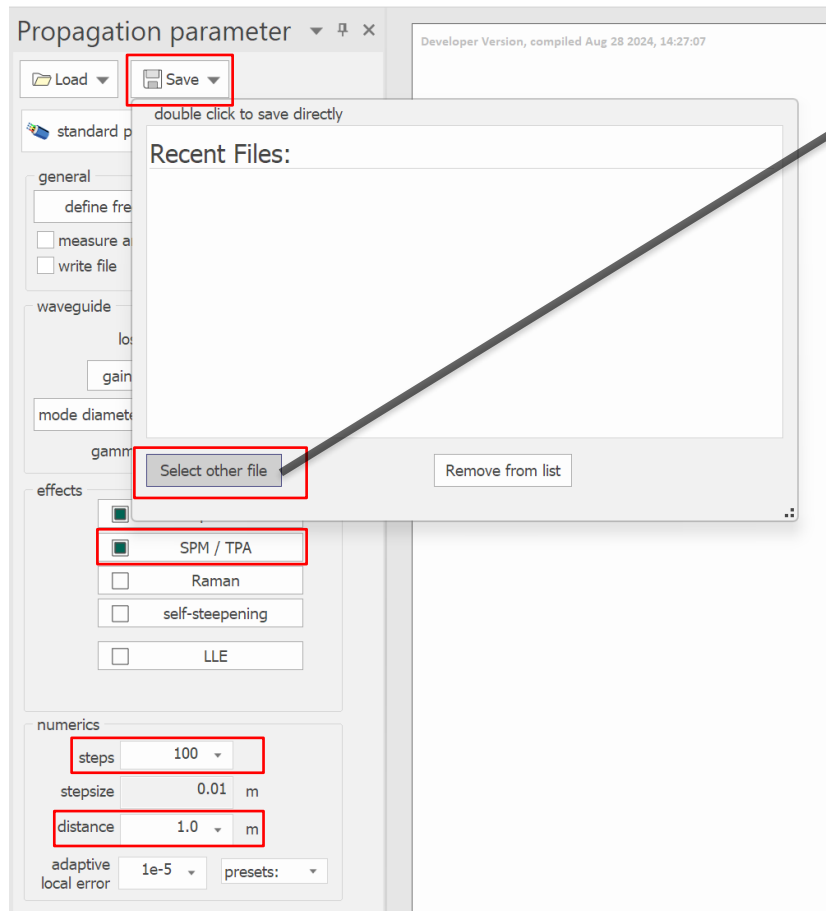
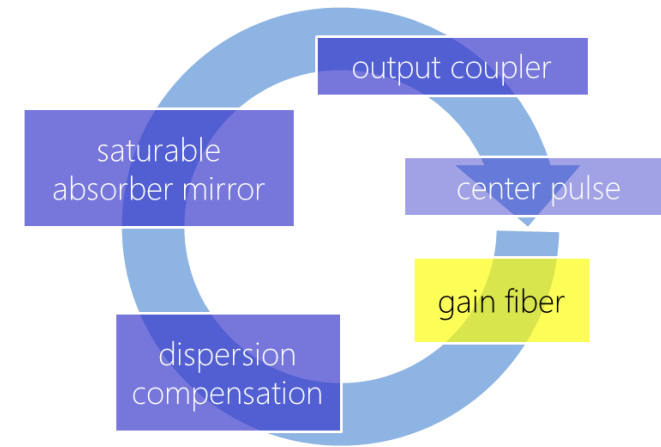
copy dispersion ((nm),D[ps/nm/km],b2[ps<sup>2</sup>/m])

copy beta2 + group delay [nm],b2 [ps<sup>2</sup>/m], GD[ps/m]

# Multi-Element Propagation: Short Pulse Fiber Lasers

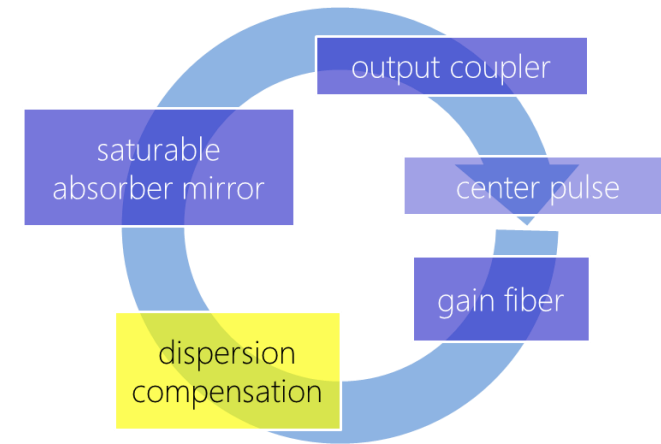
Switch on SPM, set the standard values here and make the propagation 1 m long with 100 steps.

Now save the amplifier to the file „**fiber.ppfx**“ for later use as an element.



# Multi-Element Propagation: Short Pulse Fiber Lasers

For the dispersion compensation, we only set second order dispersion in the dispersion dialog  
Furthermore, set the gain to zero, switch off SPM etc. Only dispersion need to be set.  
As it is a linear step, a single step is enough.



gamma 0.002415094 1/(W m)

effects

- dispersion
- SPM / TPA
- Raman
- self-steepening
- LLE

numerics

- steps 1
- stepsize 1 m
- distance 1 m
- adaptive local error 0
- presets:

dispersion term

$$\frac{\partial A}{\partial z} = \dots + \sum_{n \geq 1} \beta_n \frac{i^{n+1}}{n!} \frac{\partial^n}{\partial T^n} A$$

dispersion model

- Taylor expansion series
- Sellmeier coefficients
- photonic crystal fiber
- gas-filled silica-hollow core fiber
- force retarded time frame (beta0=beta1=0) @ data array center wavelength

Use dispersion  do not use dispersion

Dispersion Setup

Taylor Series @ 1060 nm

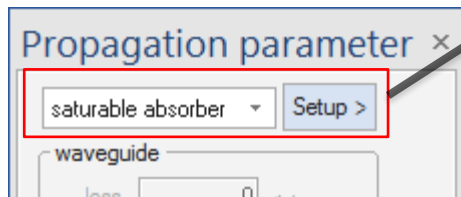
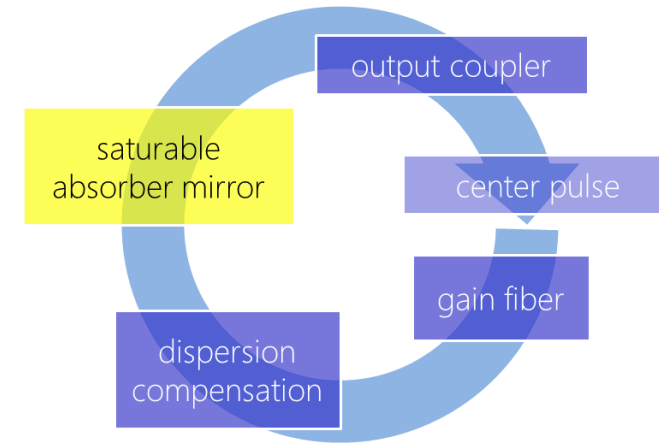
Beta1	0	ps/m
Beta2	-0.02	ps <sup>2</sup> /m
Beta3	0.0	ps <sup>3</sup> /m
Beta4	0.0	
Beta5	0	
Beta6	0	
Beta7	0	
Beta8	0	
Beta9	0	
Beta10	0	
Beta11	0	
Beta12	0	

Save the element as dispersion compensation „dc.ppfx“, as done before, for later use as an element.

# Multi-Element Propagation: Short Pulse Fiber Lasers

The saturable absorber is a different model, select „saturable absorber“ on top of the propagation parameter dialog.

Then, set it up with the parameters on the right.



**Saturable Loss**

Fast saturable loss

R0 70 %  
dR 30 %  
PA 100 W

$$R = R_0 + \Delta R - \frac{\Delta R}{1 + \frac{|A(T)|^2}{P_A}}$$

OK  
Cancel

saturable absorber mirror with time constants

unsaturated reflectivity 60 %      temporal response A 0.2 ps  
saturable reflectivity 30 %  
saturation fluence 80  $\mu\text{J}/\text{cm}^2$   
focal spot diameter 10  $\mu\text{m}$   
saturation energy 0.062831852 nJ

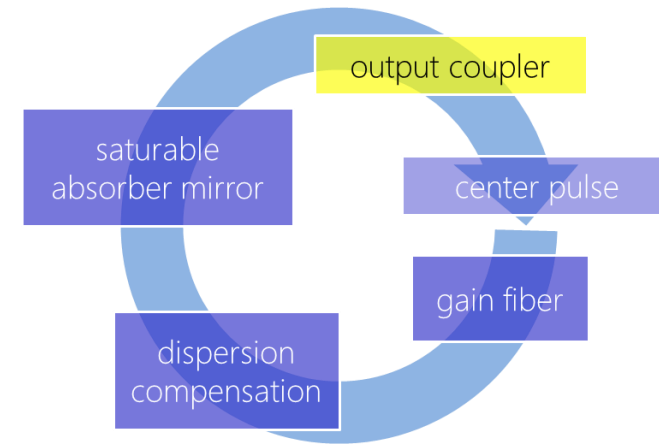
use  $R=R_0+dR*\sin^2((\text{Pi}/2)*(P/\text{PA})+\text{phi}_0)$

R0 70      dR 30      PA 1      phi\_0 0

Save the element as „SA.ppfx“, as done before, for later use as an element.

# Multi-Element Propagation: Short Pulse Fiber Lasers

The output coupler can be modelled by a simple **field manipulation**, where the field is multiplied by the outcoupling ration.

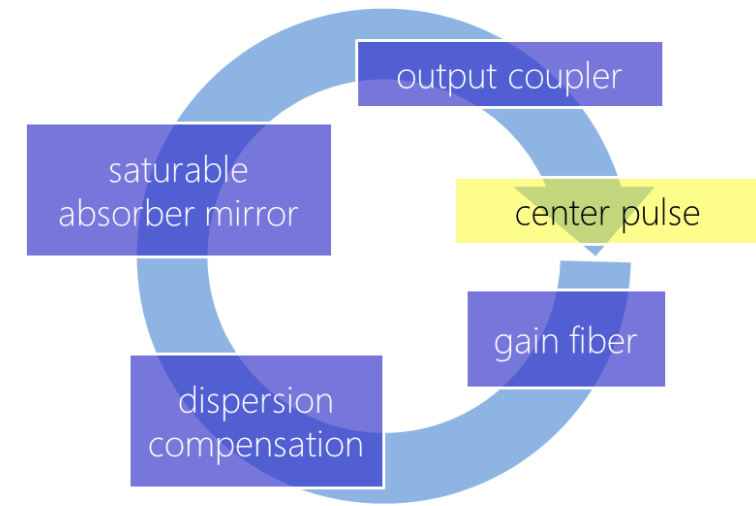
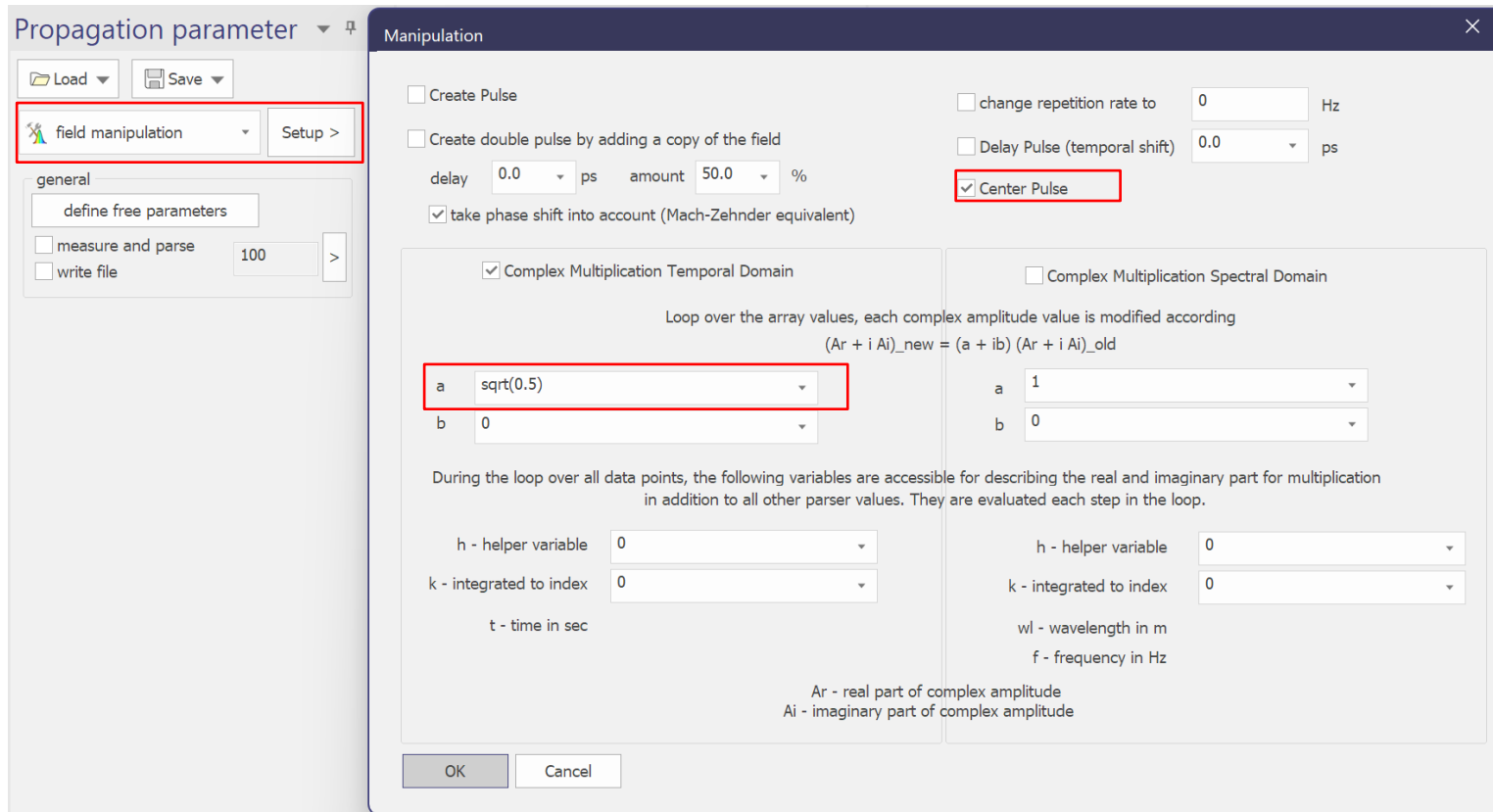


The screenshot shows the 'Manipulation' dialog box in a software interface. The 'field manipulation' option is selected in the left sidebar. A text box in the center of the dialog reads: **Outcoupling: 50% means complex multiplication with  $\sqrt{0.5}$** . The 'a' parameter in the 'Complex Multiplication Temporal' section is set to  $\sqrt{0.5}$ . The dialog also includes options for 'Create Pulse', 'Create double pulse by adding a copy of t', 'delay', 'amount', 'take phase shift into account (Mach-Zehnder)', and 'Complex Multiplication Temporal'. The 'a' parameter is highlighted with a red box.

# Multi-Element Propagation: Short Pulse Fiber Lasers

Centering the pulse in the time domain helps to converge the pulse, as changes are measured in the time domain. This is done in the field manipulation dialog.

It also means that it can be combined with the field manipulation of the output coupler before in a single element. The full manipulation dialog is shown below.

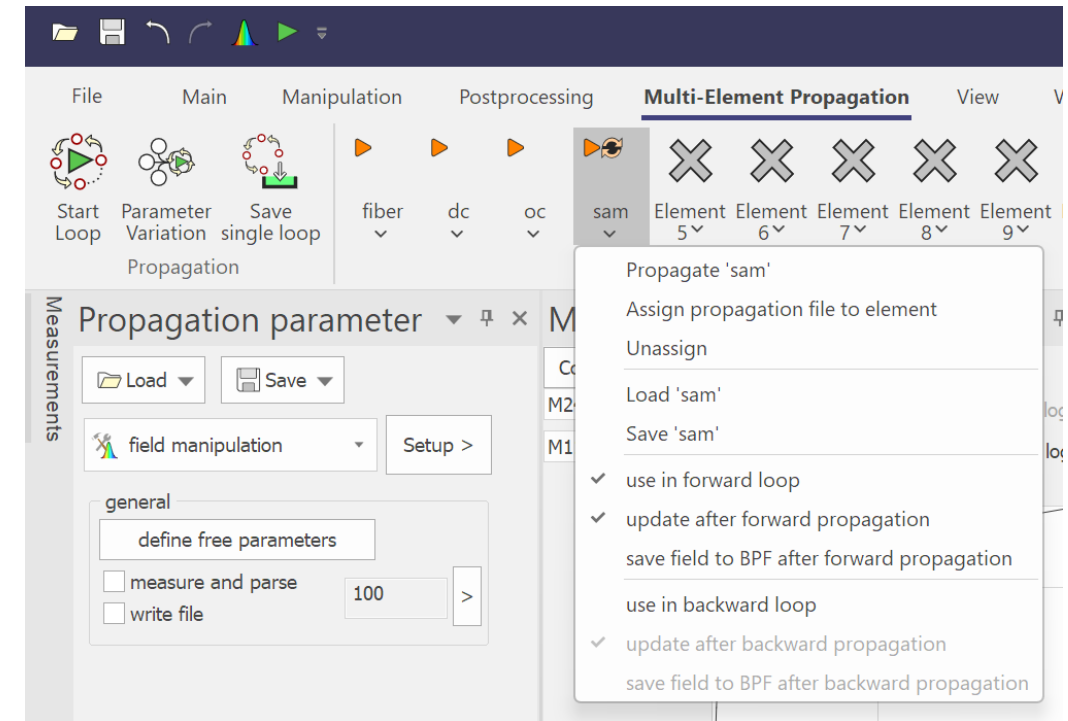
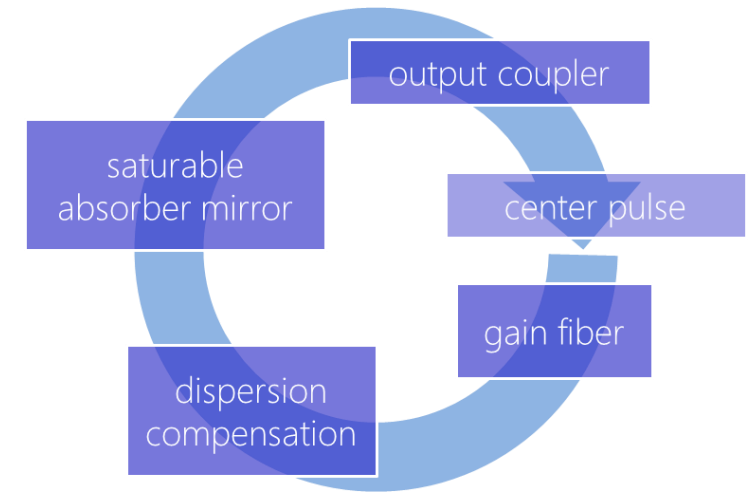


Save the element „OC.ppfx“, as done before, for later use as an element.

# Multi-Element Propagation: Short Pulse Fiber Lasers

With all elements defined, let put together the multi-element propagation:

- assign all files to elements in the right order of the cavity
- Select „use in forward loop“ for all elements
- Select the last one to be updated after each loop to see convergence live during simulation
- Icons on top change according to selected status



# Multi-Element Propagation: Short Pulse Fiber Lasers

Before the start of the multi-element propagation, we need to define the initial conditions, that is, the field where the propagation should start.

So, create the initial field with the right parameters for the final field and usually, for such short pulse oscillators, the initial condition is nothing else than quantum noise.

Pulse Profile and Data Array

data array setup

Size: 1k (2<sup>10</sup>)

array center wavelength: 1060 nm

half interval: 5 ps

field profile definition

Type: Gauss

FWHM: 1 ps

TempShift: 0 ps

phase: 0 rad

wavelength: 1060 nm

2nd order spectral phase: 0 fs<sup>2</sup>

3rd order spectral phase: 0 fs<sup>3</sup>

energy:  0 J

average power:  0 W

repetition rate: 1e+11 Hz  cw

scramble spectral phase (random phase)

phase diffusion model with given linewidth

add quantum noise (one photon per spectral node)

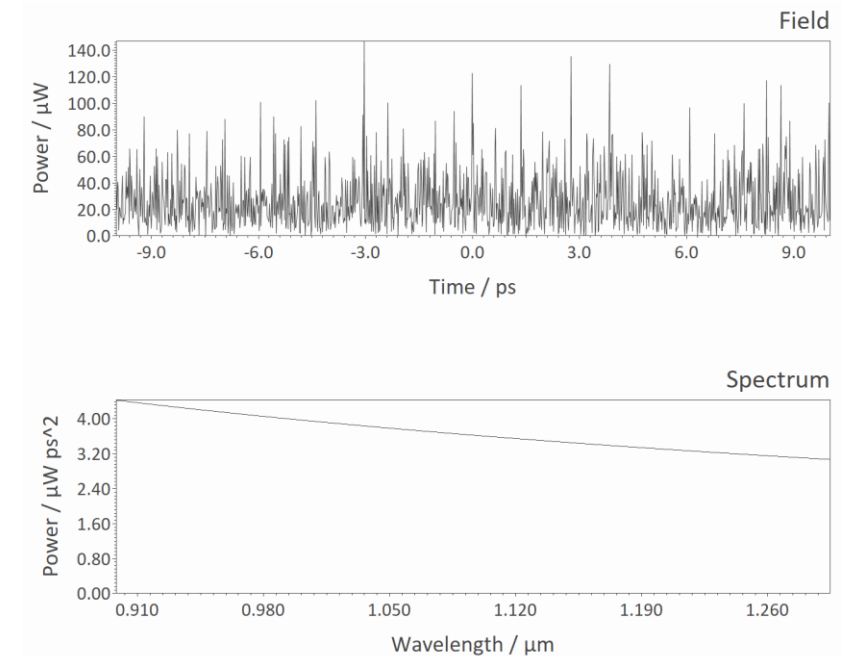
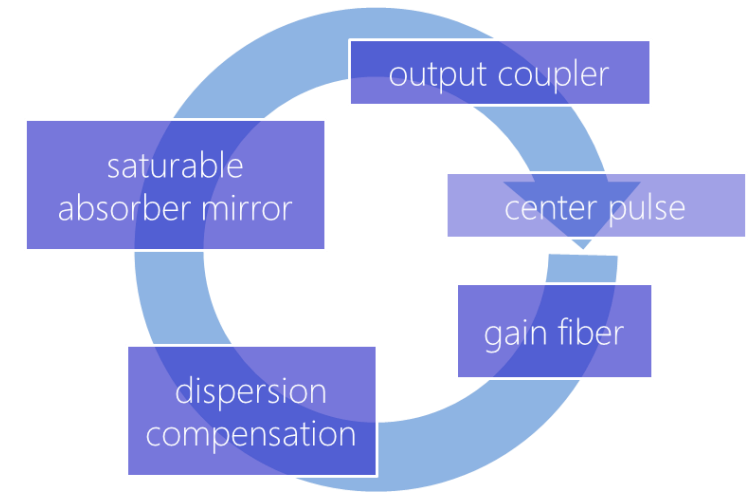
double pulsing

separation: 0 ps relative magnitude: 0

create field in data array 1

add field to data array 1

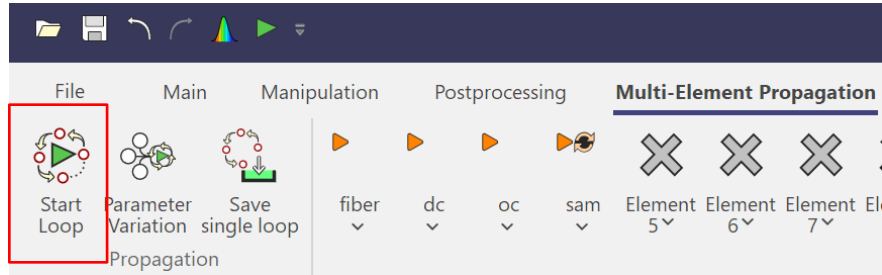
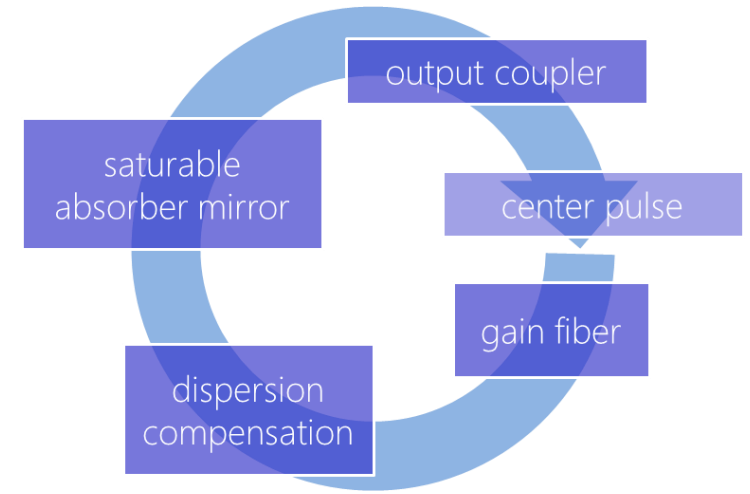
OK Apply Cancel reset



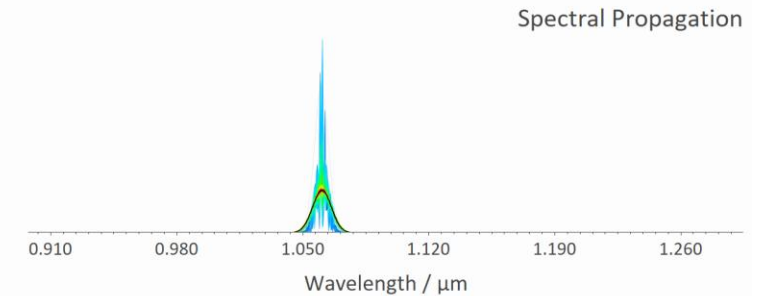
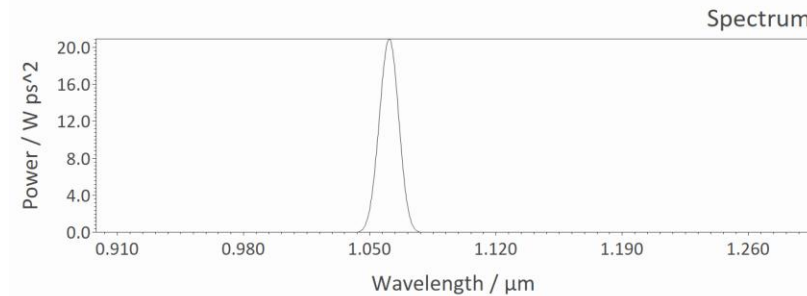
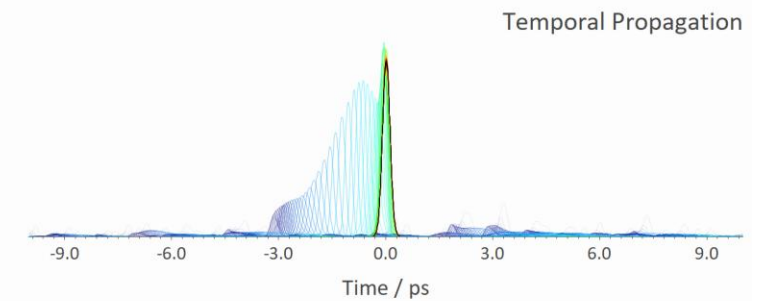
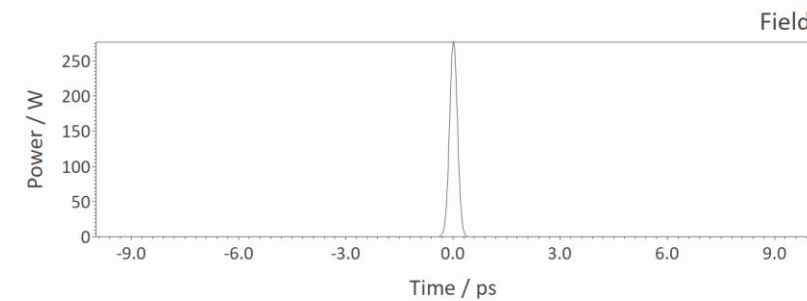
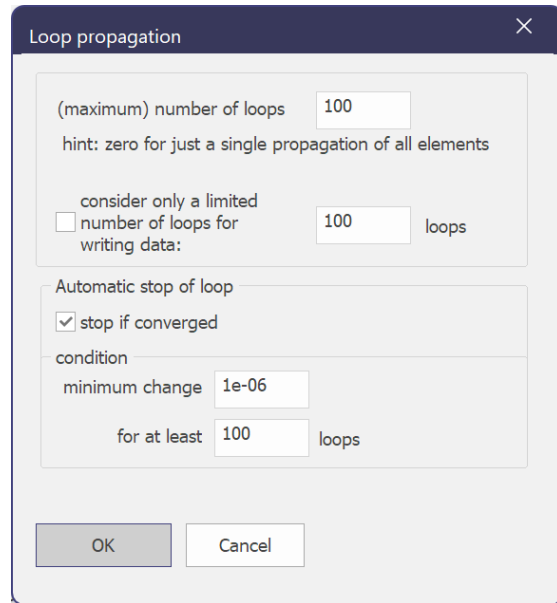
Initial field in temporal and spectrum domain.

# Multi-Element Propagation: Short Pulse Fiber Lasers

Now press „Start Loop“ with the setup dialog and press „Ok“ to let the simulation run.





You should see, that the field and spectrum is converging to a solution that is stable:

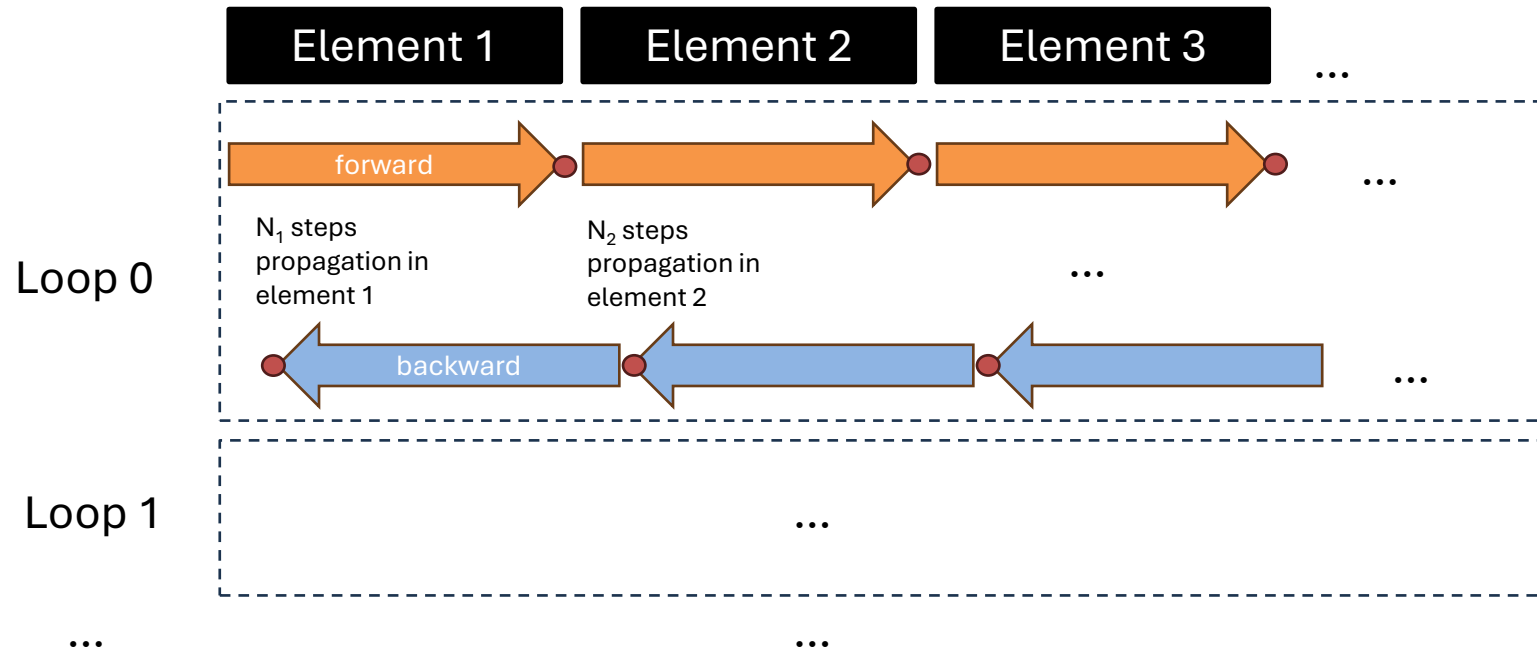
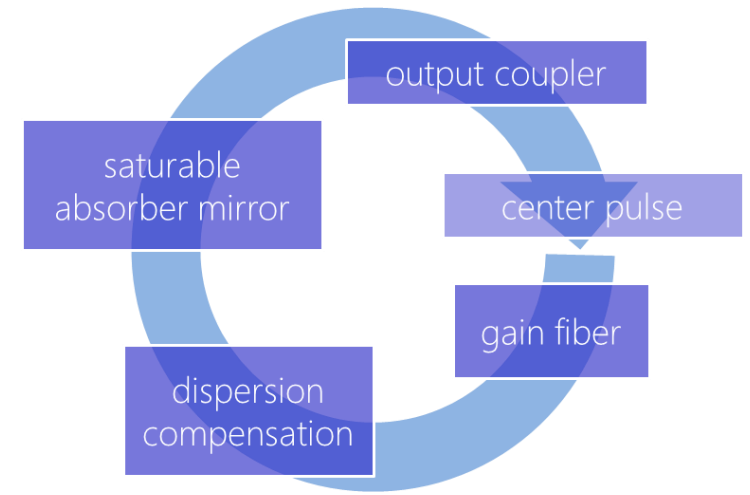


# Multi-Element Propagation: Short Pulse Fiber Lasers

How do I get information out of the multi element propagation?

Let us first take a look at the possibilities of data along the propagation:




- Changes along each elements propagation 
- Only the result at the end of each element 
- Both depend on the loop count

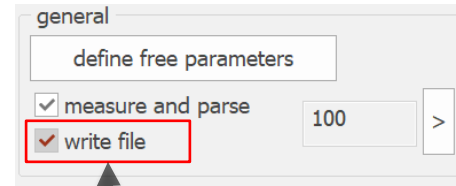


# Multi-Element Propagation: Short Pulse Fiber Lasers

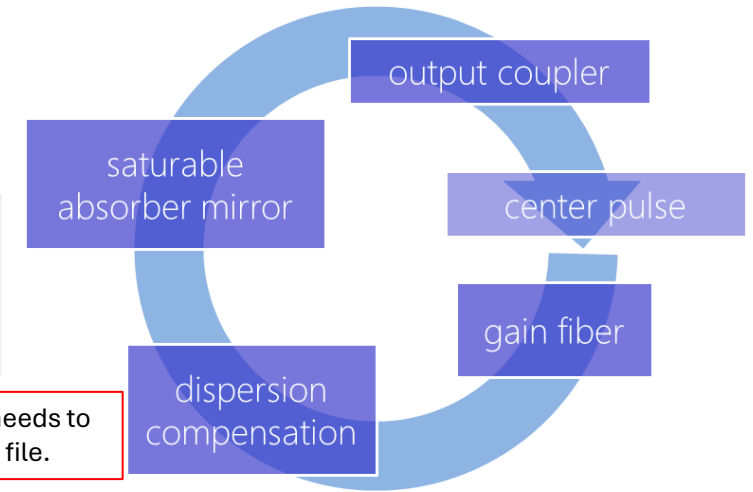
How do I get information out of the multi element propagation?

Let us first take a look at the possibilities of data along the propagation:

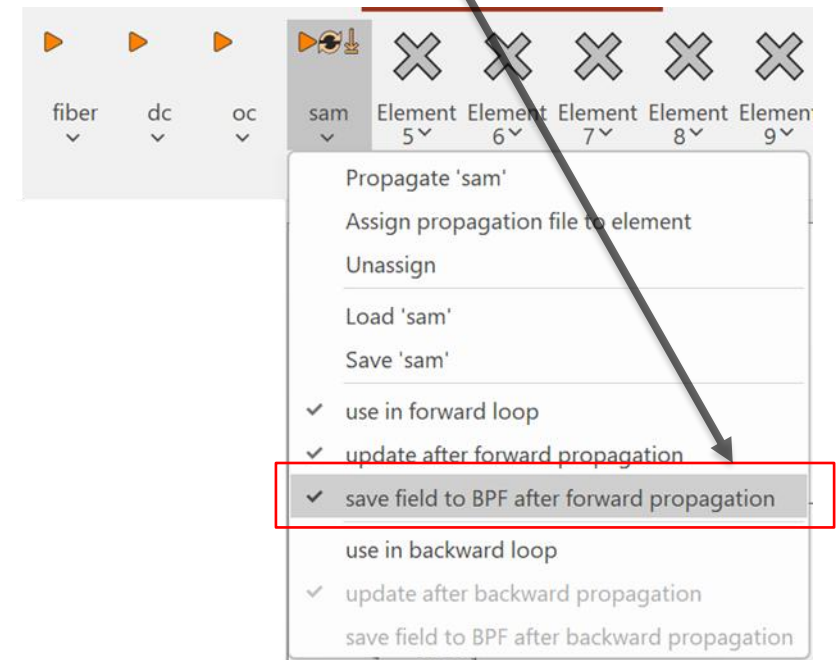
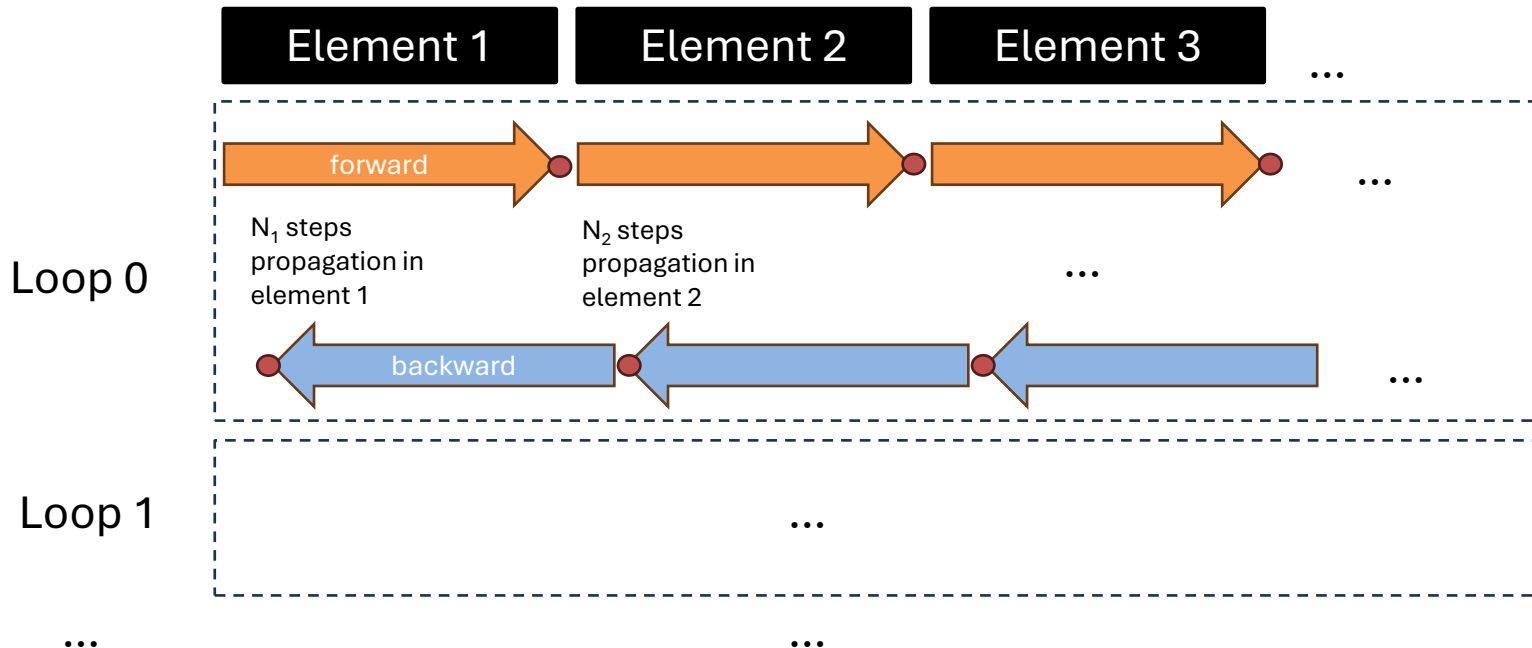
- Changes along each elements propagation  
- Only the result at the end of each element 
- Both depend on the loop count



To save the full propagation results in *each element*, „write file“ needs to be switched on and saved to the particular elements propagation file.



To save the results after the elements, select it in the sub-menu of the element:



# Multi-Element Propagation: Short Pulse Fiber Lasers

How do I get information out of the multi element propagation?

## Example:

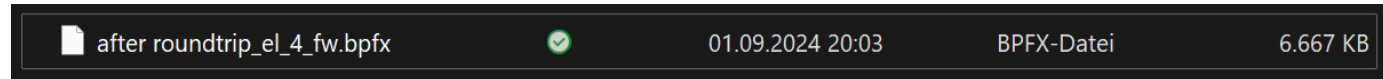
- Only the result at the end of each element ●



On start you will be asked about a filename, if not given in the dialog. Please use a filename, that is a base filename, as the element number and forward/backward signature will be attached to this base filename.

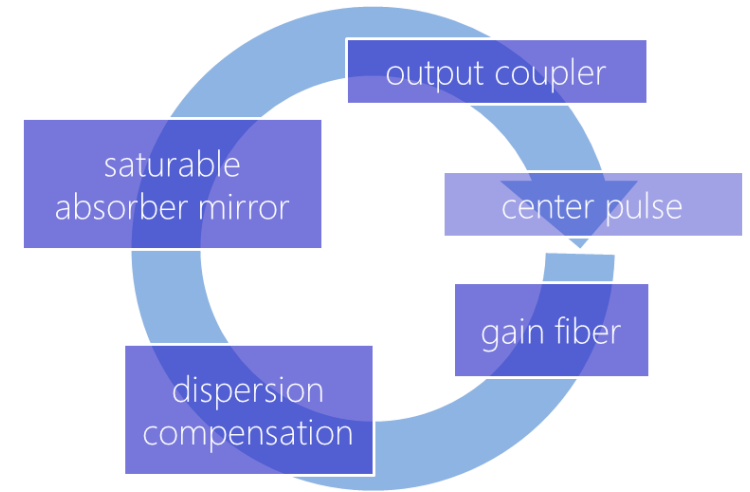
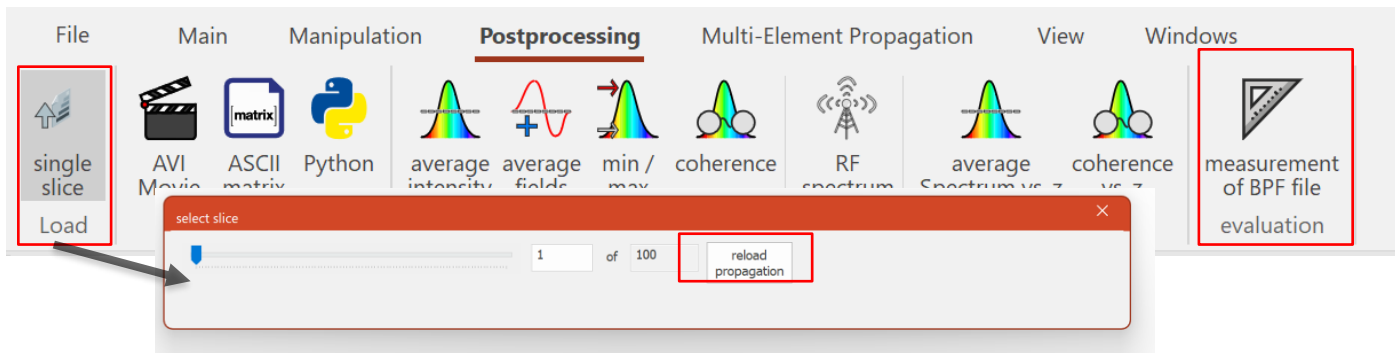
In my case, I used „after\_propagation“ as the filename

After the multi-element propagation is finished, the following file is created:

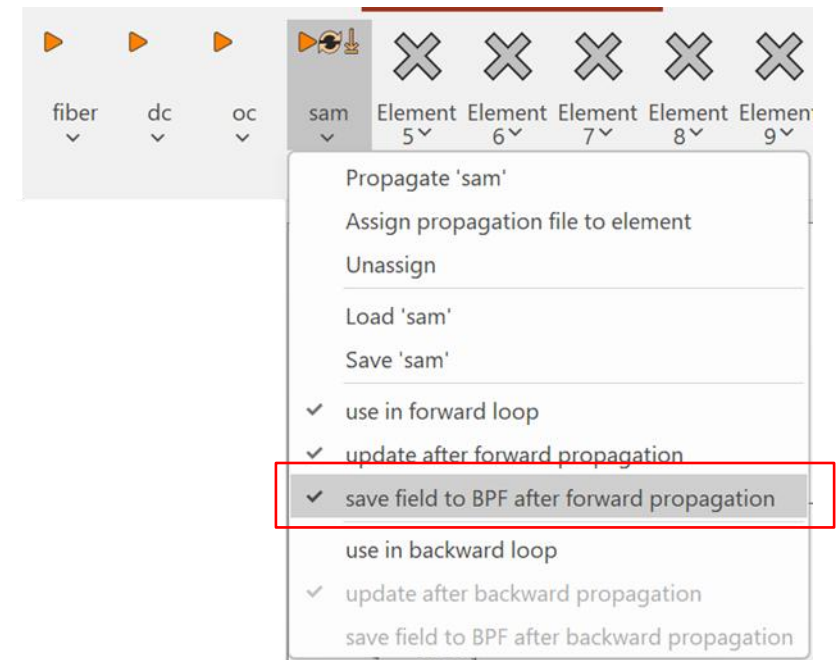


You can reload the field depending on the loop count, and also reload it into fiberdesk's 2D plot.

Also consider measurements with this propagation file.

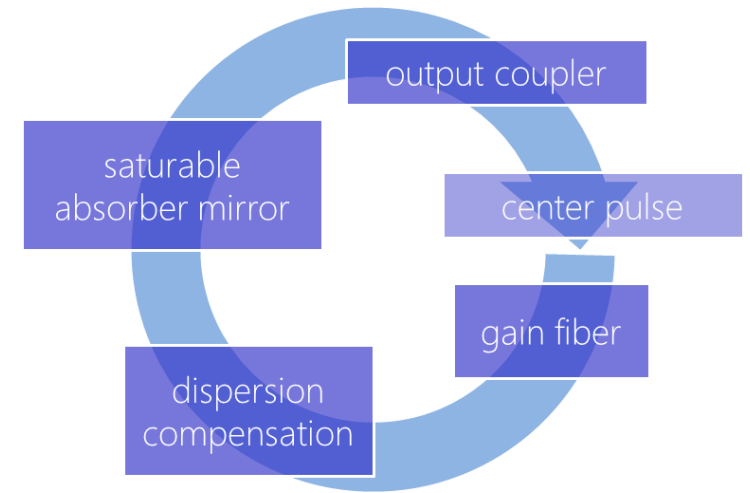


To save the results after the elements, select it in the sub-menu of the element:



# Multi-Element Propagation: Short Pulse Fiber Lasers

You can also load, display and manipulate the saved data in PYTHON. Download the python script (jupyter notebook) from the homepage to process BPFx files.



```
Preview Code Blame 282 lines (282 loc) · 239 KB Raw Copy Download Edit
```

### Example Reading propagation files - bpfx

Import the necessary libs, after installing them in your environment. Please also ensure that lxml is installed by conda install lxml

```
In [5]: import matplotlib
import struct
import base64
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

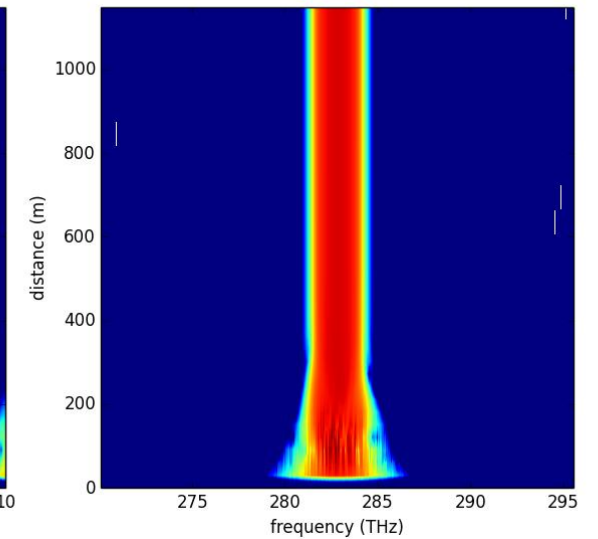
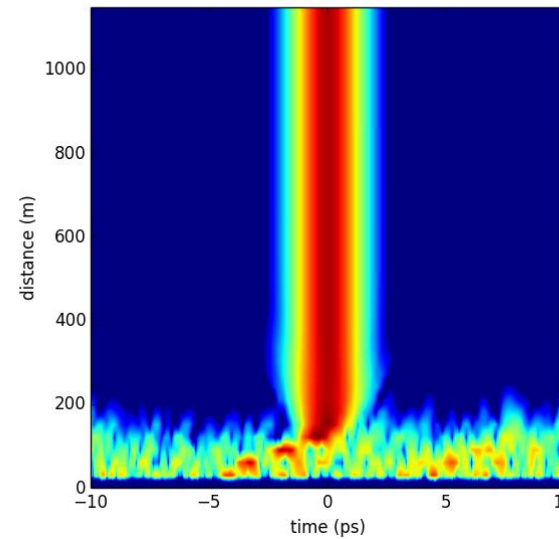
Now define the following class that load the fiberdesk propagation file (bpfx) that is actually an xml file, so pandas read\_xml is used.

```
In [6]: class fiberdesk_bpfx:
def __init__(self, name):
self.filename = name
self.frames = 0
self.bSecondFieldSaved = 0
self.datapoints = 0
self.time_intervall = 0.0
self.freq_intervall = 0.0
self.frequency_0 = 0.0
self.distance = 0.0

self.df = pd.read_xml(name)

last_index = self.df.index.size-1 # minus header
if "number_of_frames" in self.df.columns:
self.frames = int(self.df.at[last_index, "number_of_frames"])
#print(self.frames)
entries = last_index / self.frames
if "datapoints" in self.df.columns:
self.datapoints = int(self.df.at[last_index, "datapoints"])
#print(self.datapoints)

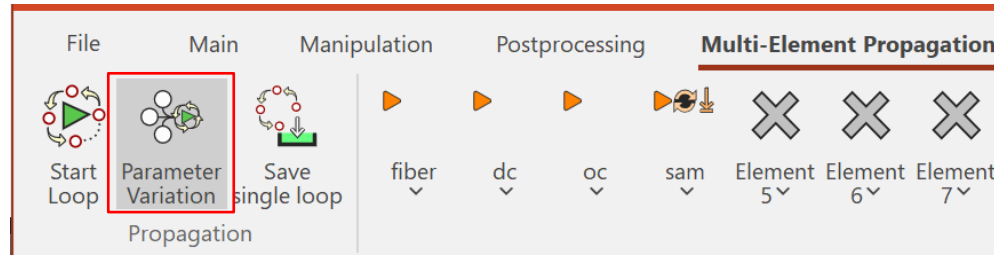
# read axis from first entry (index 0), assume that it does no change along propagation
if "freq_intervall" in self.df.columns:
self.freq_intervall = self.df.at[0, "freq_intervall"]
if "time_intervall" in self.df.columns:
self.time_intervall = self.df.at[0, "time_intervall"]
if "FrequencyNull" in self.df.columns:
self.frequency_0 = self.df.at[0, "FrequencyNull"]
if "accumulated_distance" in self.df.columns:
pos = self.df["accumulated_distance"].dropna().to_numpy()
self.distance = np.max(pos)-np.min(pos)
#print(self.distance)
```



# lecture 5

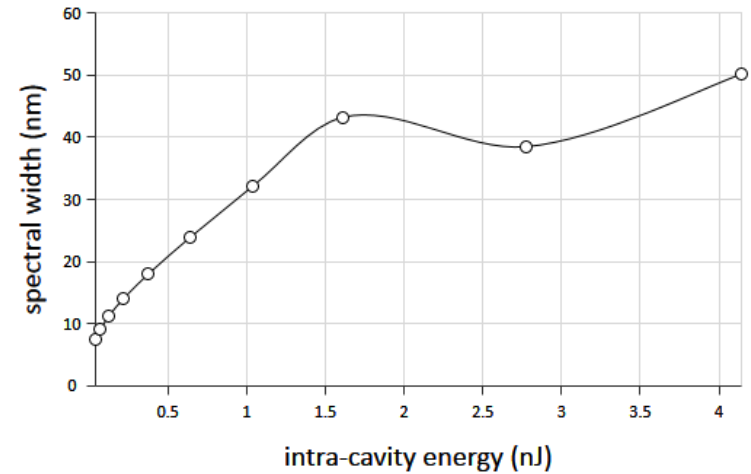
## Multi-Element Propagation:

Example: Short Pulse Fiber Lasers



Multi-element > Parameter variation

we change the gain saturation to increase the energy (remark: intracavity energy!)



2D  use final field as input for next loop (otherwise use start/create field)

Px6 - NLSE.gain\_saturz = [ ] x J, with x 1e-011 to 1e-009, datapoints 10, log steps

in Element 0 - fibe x-axis value = 1e9\*M2

save to file C:\Users\admin\Desktop\tutorial\oscillator simulations\energy scaling.pvf select base file

Result = 1e9\*M20 update or use M20 - spec.width.m

auto axis x intra-cavity energy (nJ)

auto axis y spectral width (nm)

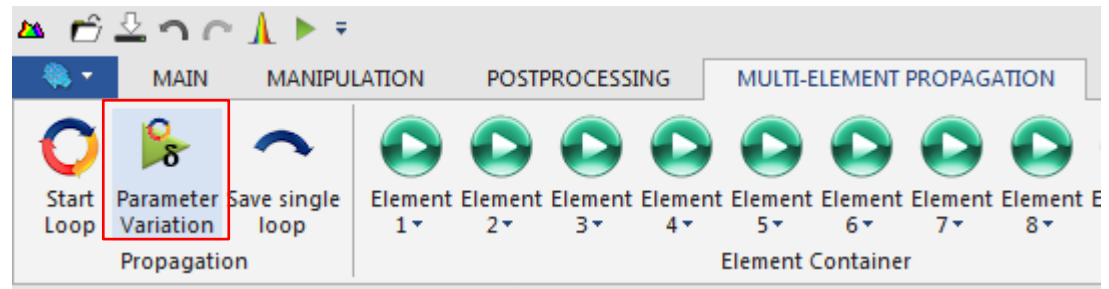
start multiple elements

setup >>

# lecture 5

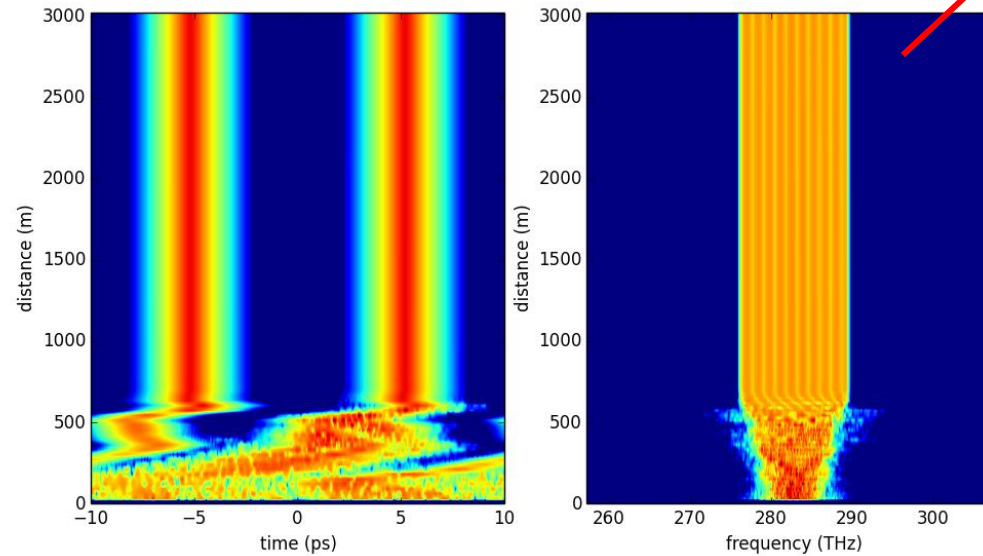
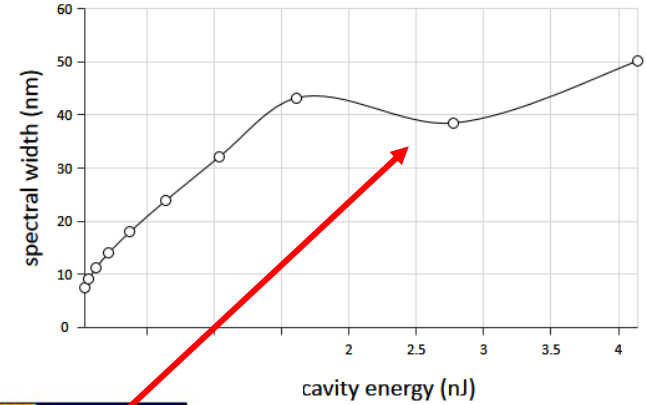
## Multi-Element Propagation:

Example: Short Pulse Fiber Lasers



Multi-element > Parameter variation

we change the gain saturation to increase the energy (remark: intracav



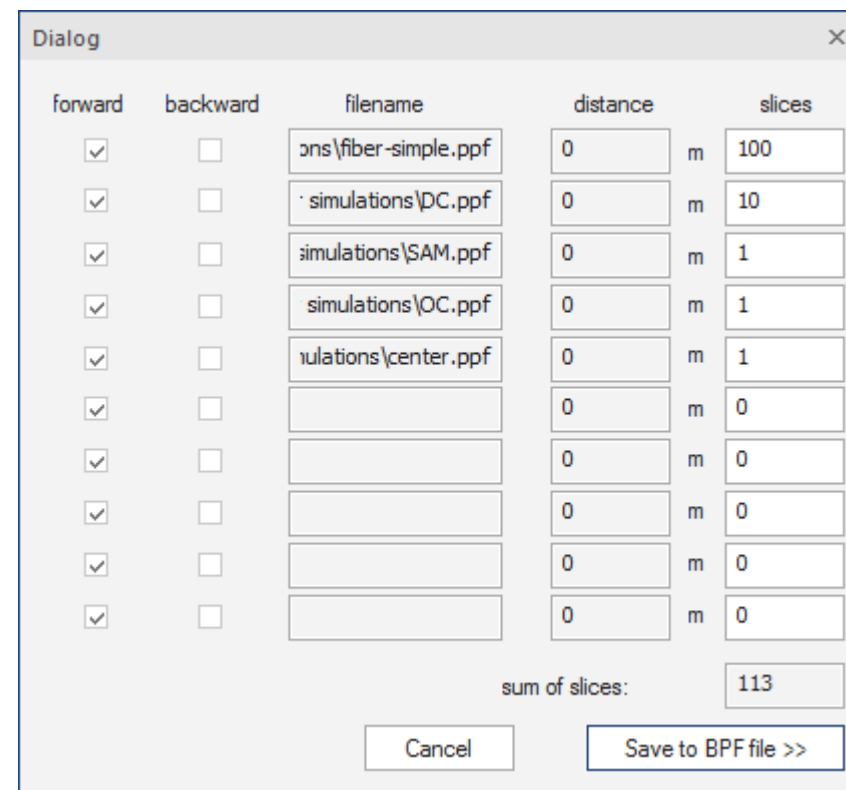
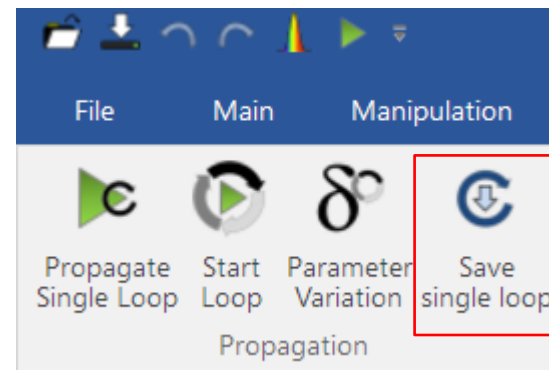
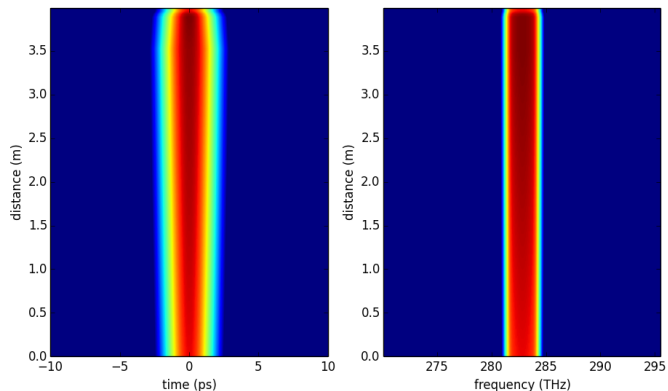
# lecture 5

## Multi-Element Propagation:

Example: Short Pulse Fiber Lasers

Intracavity evolution

- (1) select stable solution from saved file
- (2) specify slices to be saved
- (3) post-process

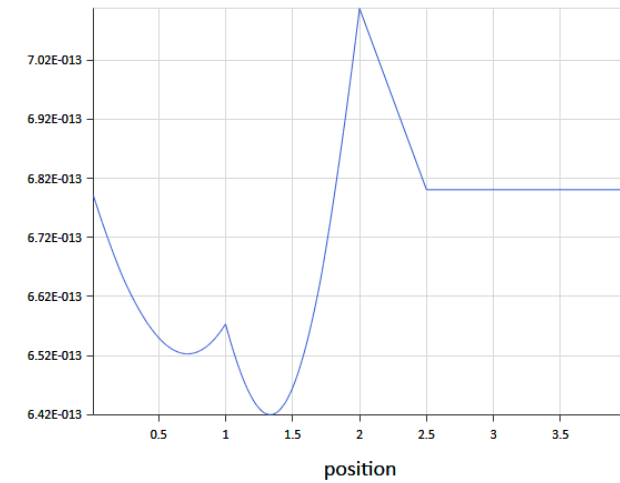
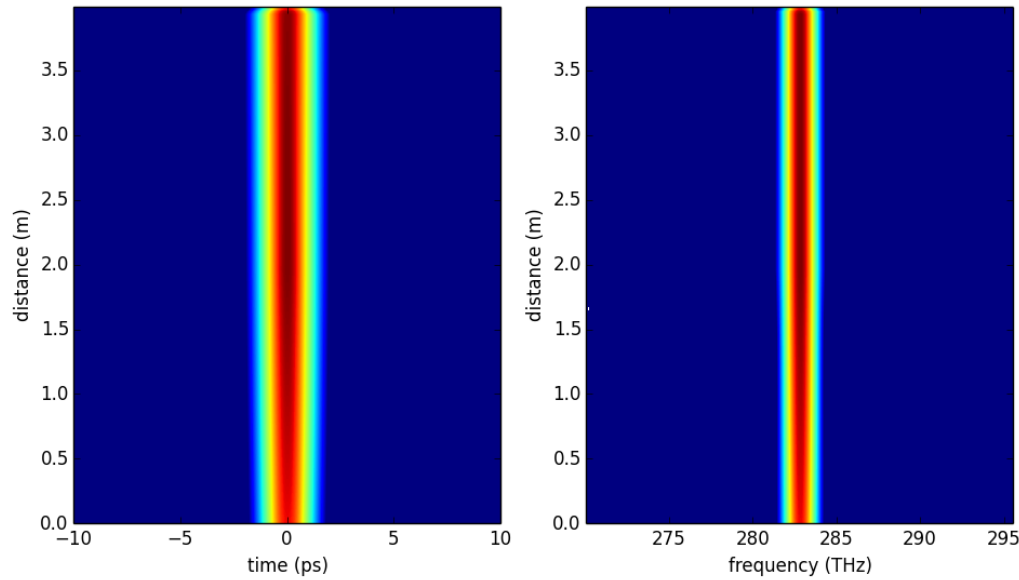
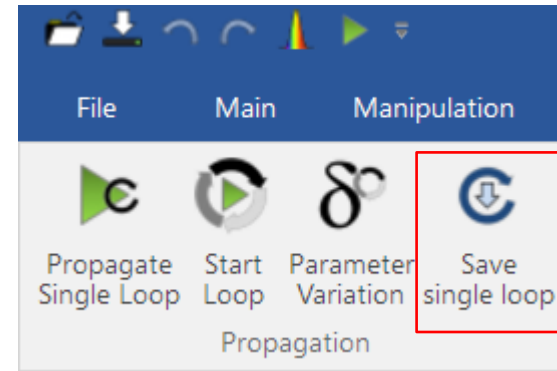


# lecture 5

## Multi-Element Propagation:

Example: Short Pulse Fiber Lasers

soliton solution:  $\beta_{2@DC} = -0.06 \text{ ps}^2$

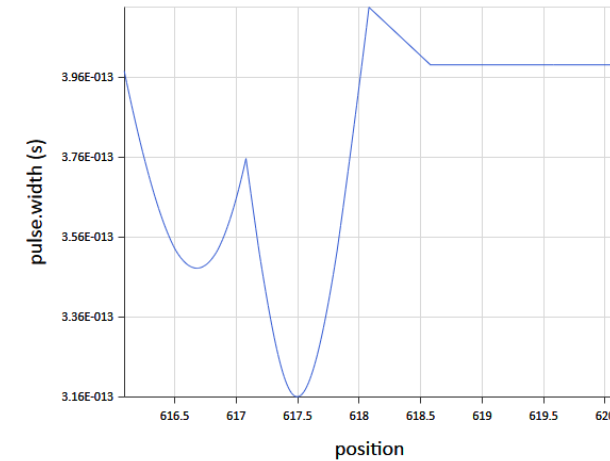
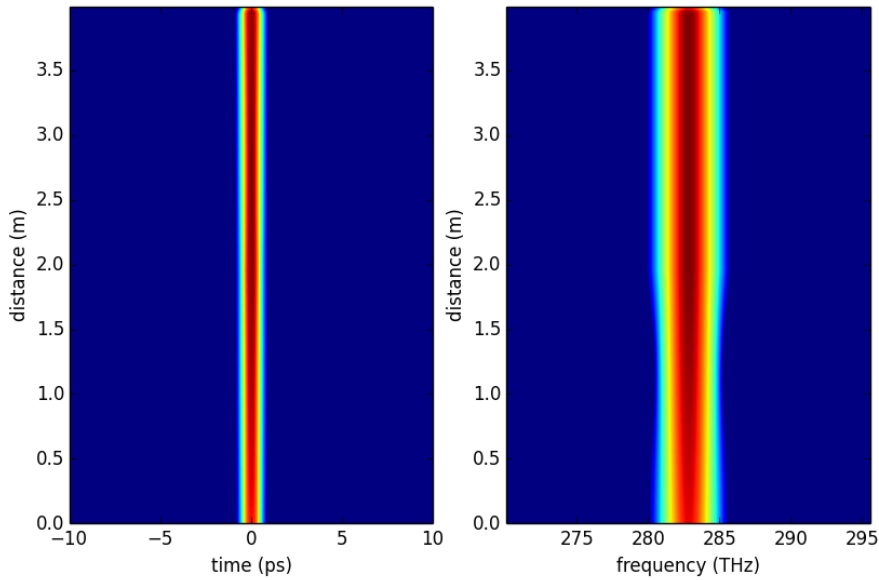
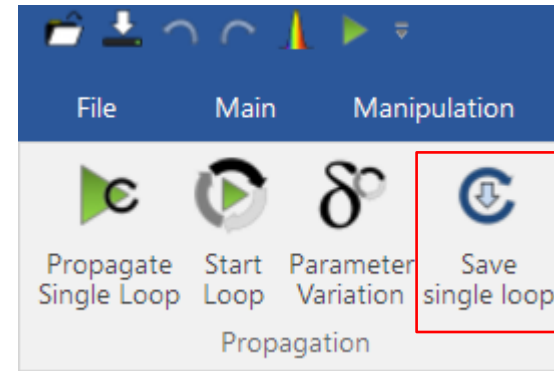


# lecture 5

## Multi-Element Propagation:

Example: Short Pulse Fiber Lasers

soliton solution:  $\beta_{2@DC} = -0.04 \text{ ps}^2$

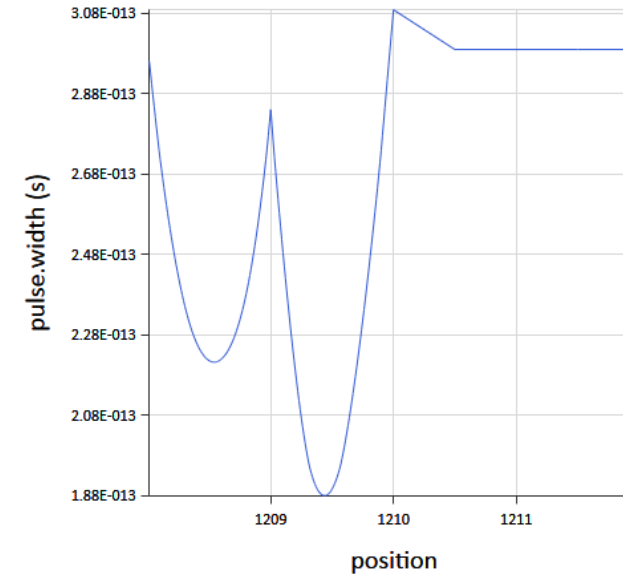
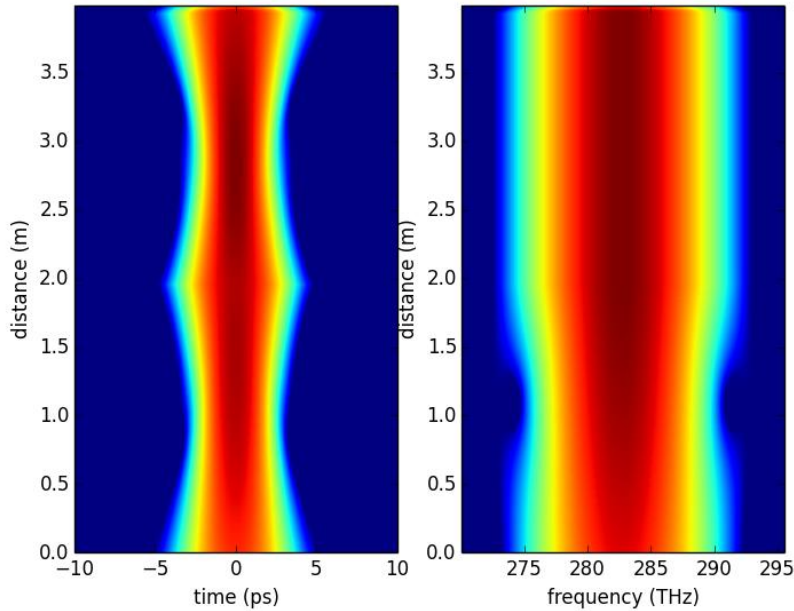
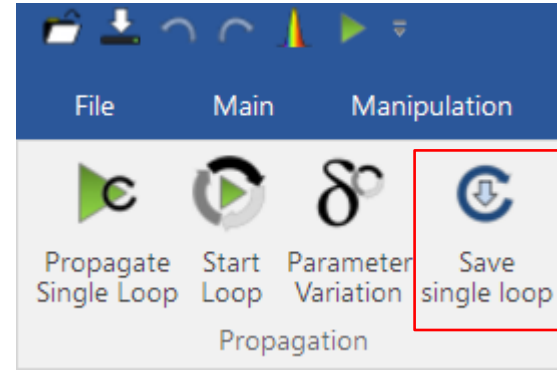


# lecture 5

## Multi-Element Propagation:

Example: Short Pulse Fiber Lasers

toward stretched pulse:  $\beta_{2@DC} = -0.03 \text{ ps}^2$

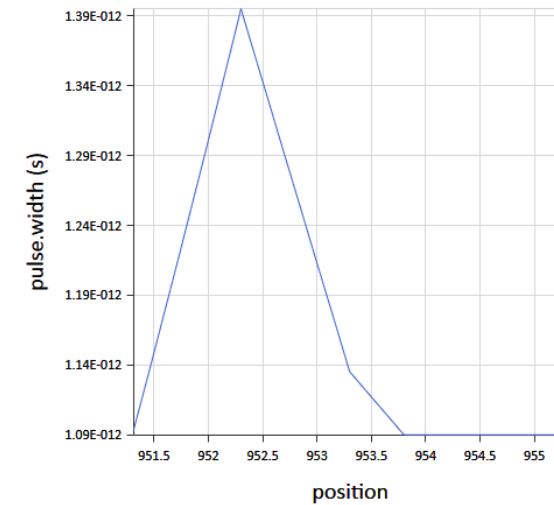
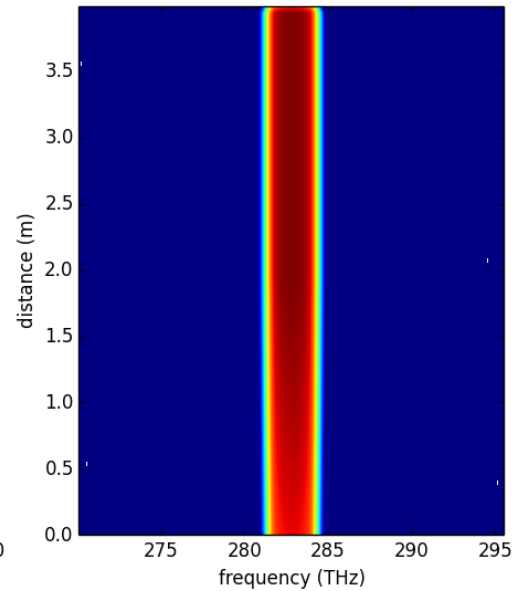
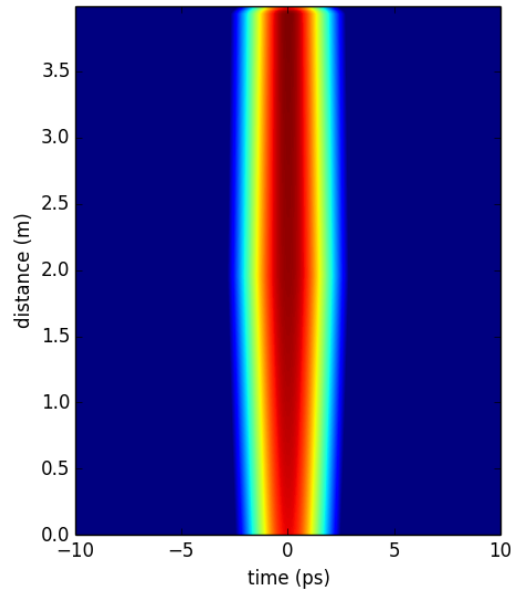
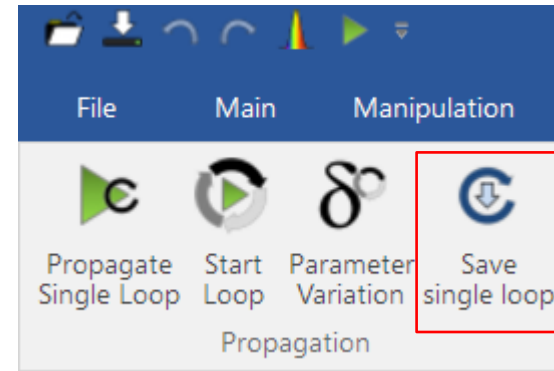


# lecture 5

## Multi-Element Propagation:

Example: Short Pulse Fiber Lasers

similariton:  $\beta_{2@DC} = -0.02 \text{ ps}^2$

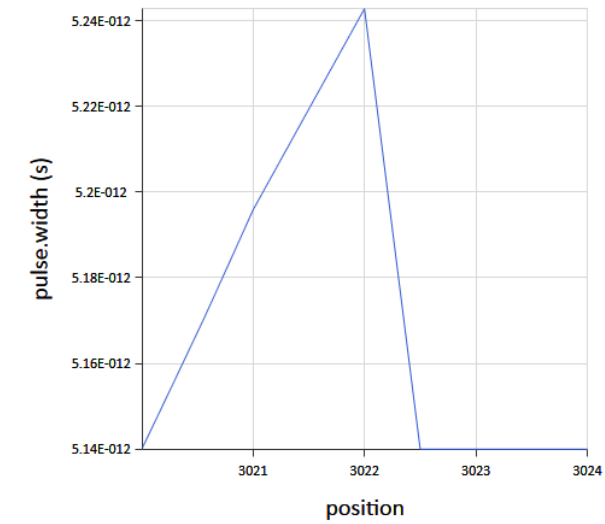
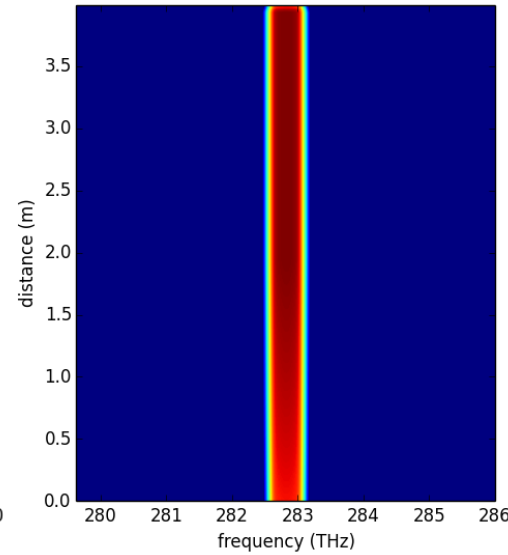
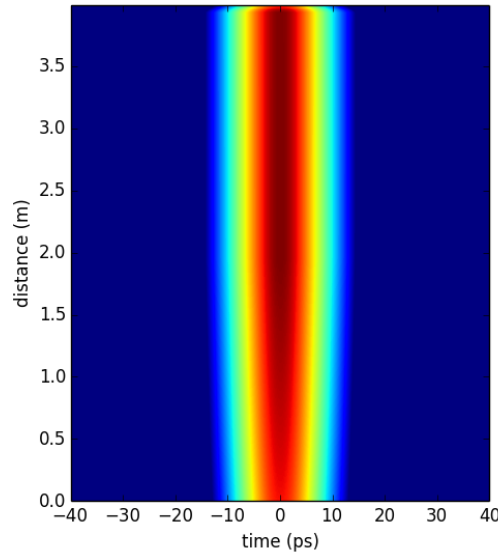
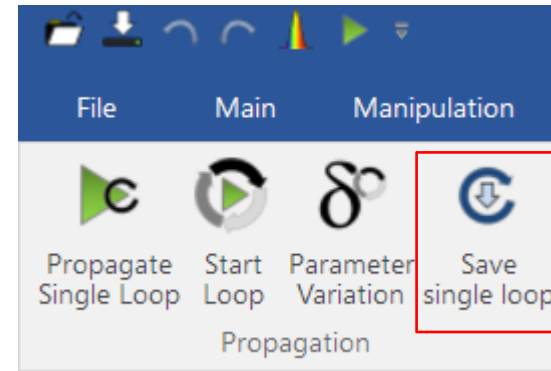


# lecture 5

## Multi-Element Propagation:

Example: Short Pulse Fiber Lasers

chirped pulse oscillator:  $\beta_{2@DC} = +0.02 \text{ ps}^2$



# Lecture 5

## Multi-Element Propagation

### Part 1:

Short Pulse Fiber Laser

### Part 2:

Mamyshev Oscillator

### Part 3:

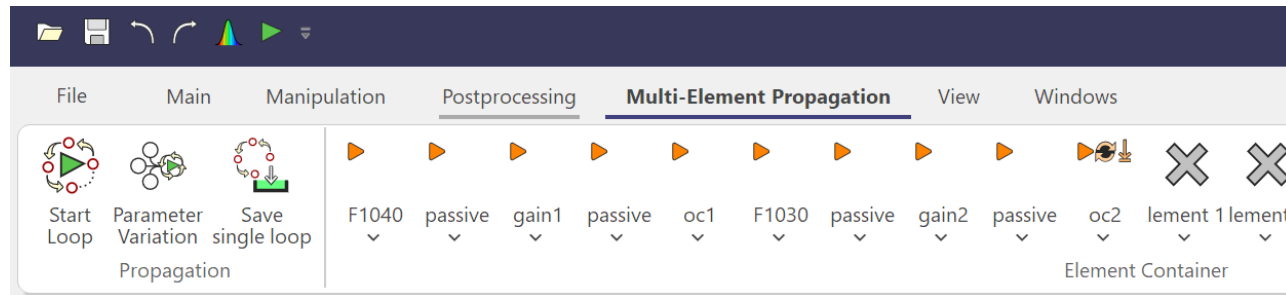
Micro Comb Generator



# Multi-Element Propagation: Mamyshev Oscillator

See for example: Z. Liu, Z. Ziegler, L. G. Wright, and F. W. Wise. "Megawatt peak power from a Mamyshev oscillator." *Optica* **4**, 649 (2017).

The model of the Mamyshev oscillator is using the following elements:



It uses two separated filters at 1030 nm and 1040 nm as well as passive and active fibers in between. The gain fibers are different after the filter. An output coupler is used at the end.

# Multi-Element Propagation: Mamyshev Oscillator

The filters are defined as follows:

Pulse manipulation propagation

Saved as: **F1040.ppfx**

Manipulation

Create Pulse

Create double pulse by adding a copy of the field

delay 0.0 ps amount 50.0 %

take phase shift into account (Mach-Zehnder equivalent)

change repetition rate to 0 Hz

Delay Pulse (temporal shift) 0.0 ps

Center Pulse

Complex Multiplication Temporal Domain

Complex Multiplication Spectral Domain

Loop over the array values, each complex amplitude value is modified according

$$(Ar + i Ai)_{new} = (a + ib) (Ar + i Ai)_{old}$$

a sqrt(0.75)

b 0

During the loop over all data points, the following variables are accessible for describing the real and imaginary part for multiplication in addition to all other parser values. They are evaluated each step in the loop.

h - helper variable 0

k - integrated to index 0

t - time in sec

wl - wavelength in m

f - frequency in Hz

Ar - real part of complex amplitude

Ai - imaginary part of complex amplitude

OK Cancel

Contains a general loss of 25 % and a filter transmission of 70%. The width ( $1/e^2$ ) is 4 nm centered at 1040 nm.

Pulse manipulation propagation

Saved as: **F1030.ppfx**

Manipulation

Create Pulse

Create double pulse by adding a copy of the field

delay 0.0 ps amount 50.0 %

take phase shift into account (Mach-Zehnder equivalent)

change repetition rate to 0 Hz

Delay Pulse (temporal shift) 0.0 ps

Center Pulse

Complex Multiplication Temporal Domain

Complex Multiplication Spectral Domain

Loop over the array values, each complex amplitude value is modified according

$$(Ar + i Ai)_{new} = (a + ib) (Ar + i Ai)_{old}$$

a sqrt(0.75)

b 0

During the loop over all data points, the following variables are accessible for describing the real and imaginary part for multiplication in addition to all other parser values. They are evaluated each step in the loop.

h - helper variable 0

k - integrated to index 0

t - time in sec

wl - wavelength in m

f - frequency in Hz

Ar - real part of complex amplitude

Ai - imaginary part of complex amplitude

OK Cancel

Similar, but centered at 1030 nm.

# Multi-Element Propagation: Mamyshev Oscillator

The passive fiber is defined as a standard propagation. The dispersion is only second order at 1035 nm.

standard propagation  
Saved as: **passive.ppfx**

The 'Propagation parameter' dialog box is shown with the following settings:

- general:** 'define free parameters' button; 'measure and parse' set to 100; 'write file' button with 'more >'.
- waveguide:** 'loss' set to 0.0 (1/m); 'gain' set to 0 (1/m); 'mode diameter' set to 10.0 (μm); 'gamma' set to 0.0024734295 (1/(W m)).
- effects:** 'dispersion', 'SPM / TPA', 'Raman', and 'self-steepening' are checked; 'LLE' is unchecked.
- numerics:** 'steps' set to 100; 'stepsize' set to 0.008 m; 'distance' set to 0.8 m; 'adaptive local error' set to 6e-07; 'presets' dropdown.
- performance:** 'no background propagation file' and 'no live screen update' are unchecked.

The 'dispersion term' dialog box displays the following equation and options:

$$\frac{\partial A}{\partial z} = \dots + \sum_{n \geq 1} \beta_n \frac{i^{n+1}}{n!} \frac{\partial^n}{\partial T^n} A$$

Options for the dispersion term:

- beta 0
- beta 1
- beta 2
- D

The 'dispersion model' section includes:

- Taylor expansion series (with 'Setup >>' button)
- Sellmeier coefficients
- photonic crystal fiber
- gas-filled silica-hollow core fiber
- force retarded time frame (beta0=beta1=0) @ data array center wavelength

At the bottom, there are two main options:  Use dispersion and  do not use dispersion (with an 'auto' checkbox).

The 'Dispersion Setup' dialog box shows a Taylor Series expansion at 1035 nm with the following coefficients:

Order	Value	Unit
Beta1	0	ps/m
Beta2	0.02	ps <sup>2</sup> /m
Beta3	0.0	ps <sup>3</sup> /m
Beta4	0	
Beta5	0	
Beta6	0	
Beta7	0	
Beta8	0	
Beta9	0	

# Multi-Element Propagation: Mamyshev Oscillator

The output coupler is again defined as a pulse manipulation:

Pulse manipulation propagation

Saved as: OC1.ppfx

Manipulation

Create Pulse

change repetition rate to 0 Hz

Create double pulse by adding a copy of the field

Delay Pulse (temporal shift) 0.0 ps

delay 0.0 ps amount 50.0 %

Center Pulse

take phase shift into account (Mach-Zehnder equivalent)

Complex Multiplication Temporal Domain

Complex Multiplication Spectral Domain

Loop over the array values, each complex amplitude value is modified according

$$(Ar + i Ai)_{new} = (a + ib) (Ar + i Ai)_{old}$$

a sqrt(0.5)

b 0

During the loop over all data points, the following variables are accessible for describing the real and imaginary part for multiplication in addition to all other parser values. They are evaluated each step in the loop.

h - helper variable 0

k - integrated to index 0

t - time in sec

wl - wavelength in m

f - frequency in Hz

Ar - real part of complex amplitude  
Ai - imaginary part of complex amplitude

OK Cancel

50% Transmission.

Pulse manipulation propagation

Saved as: OC2.ppfx

Manipulation

Create Pulse

change repetition rate to 0 Hz

Create double pulse by adding a copy of the field

Delay Pulse (temporal shift) 0.0 ps

delay 0.0 ps amount 50.0 %

Center Pulse

take phase shift into account (Mach-Zehnder equivalent)

Complex Multiplication Temporal Domain

Complex Multiplication Spectral Domain

Loop over the array values, each complex amplitude value is modified according

$$(Ar + i Ai)_{new} = (a + ib) (Ar + i Ai)_{old}$$

a sqrt(0.5)

b 0

During the loop over all data points, the following variables are accessible for describing the real and imaginary part for multiplication in addition to all other parser values. They are evaluated each step in the loop.

h - helper variable 0

k - integrated to index 0

t - time in sec

wl - wavelength in m

f - frequency in Hz

Ar - real part of complex amplitude  
Ai - imaginary part of complex amplitude

OK Cancel

12% Transmission. (88% out-coupling)

# Multi-Element Propagation: Mamyshev Oscillator

The first gain fiber is defined as a standard propagation, same dispersion as the passive fiber:

Standard propagation  
Saved as: gain1.ppx

Propagation parameter

Load Save

standard propagation Setup >

general

define free parameters

measure and parse 100

write file more >

waveguide

loss 0 1/m

gain 10 dB/m

mode diameter 10.0 μm

gamma 0.0024734295 1/(W m)

effects

dispersion

SPM / TPA

Raman

self-steepening

LLE

numerics

steps 100

stepsize 0.025 m

distance 2.5 m

adaptive local error 6e-07 presets: >

performance

no background propagation file

no live screen update

Gain

steady state gain (long pulses to cw)

saturate gain with  $E_{sat} = 9.15e-9$  J

$$g = g_0 / (1 + \frac{E}{E_{sat}})$$

profile

gain profile

Center 1035 nm Width 40 nm

shape Gauss

add second peak

Center 1060 nm Width 40 nm

shape const

ratio of second to first peak (set to zero for only one peak): 0

user defined gain file

use ASCII file for gain profile given in g(1/m) vs. wavelength (separator TAB)

file ...

temporal gain saturation (pulses shorter than population relaxation time)

use temporal gain saturation instead of steady state gain and profiles, with

saturation fluence F 30 J/cm<sup>2</sup>

saturation energy with current MFD: J

$$g(T) = g_0 \exp\left(-\frac{1}{E_{sat}} \int_{-\infty}^T |A(t)|^2 dt\right) \text{ with } E_{sat} = F \cdot A_{eff}$$

Back

# Multi-Element Propagation: Mamyshev Oscillator

The second gain fiber is defined similarly, different gain saturation

Standard propagation  
Saved as: gain2.ppx

Propagation parameter

Load Save

standard propagation Setup >

general

define free parameters

measure and parse 100

write file more >

waveguide

loss 0 1/m

gain 10 dB/m

mode diameter 10.0 μm

gamma 0.002473429 1/(W m)

effects

dispersion

SPM / TPA

Raman

self-steepening

LLE

numerics

steps 200

stepsize 0.0125 m

distance 2.5 m

adaptive local error 6e-07 presets: >

performance

no background propagation file

no live screen update

Gain

steady state gain (long pulses to cw)

saturate gain with  $E_{sat} = 36.6e-9$  J

$$g = g_0 / \left(1 + \frac{E}{E_{sat}}\right)$$

profile

gain profile

Center 1035 nm Width 40 nm

shape Gauss

add second peak

Center 1060 nm Width 40 nm

shape const

ratio of second to first peak (set to zero for only one peak): 0

user defined gain file

use ASCII file for gain profile given in g(1/m) vs. wavelength (separator TAB)

file ...

temporal gain saturation (pulses shorter than population relaxation time)

use temporal gain saturation instead of steady state gain and profiles, with

saturation fluence F 30 J/cm²

saturation energy with current MFD: J

$$g(T) = g_0 \exp\left(-\frac{1}{E_{sat}} \int_{-\infty}^T |A(t)|^2 dt\right) \text{ with } E_{sat} = F \cdot A_{eff}$$

Back

# Multi-Element Propagation: Mamyshev Oscillator

As most Mamyshev oscillators are not self starting, we seed with a short pulse:

## Create Pulse:

Pulse Profile and Data Array

data array setup

Size: 1k ( $2^{10}$ )

array center wavelength: 1035 nm

half interval: 10 ps

vacuum length (full interval): 0.00599584916 m

fit all parameters to the closest value of the array:

Type: Gauss

duration (FWHM): 0.1 ps (sets spectral width)

TempShift: 0 ps

phase: 0 rad

wavelength: 1035 nm  same as array

2nd order spectral phase: 0 fs<sup>2</sup>

3rd order: 0 fs<sup>3</sup>

energy:  1e-09 J

average power:  0.1 W

repetition rate: 100e6 Hz  cw

scramble spectral phase (random phase)

phase diffusion modell with given linewidth

add quantum noise: 1 photons per spectral node

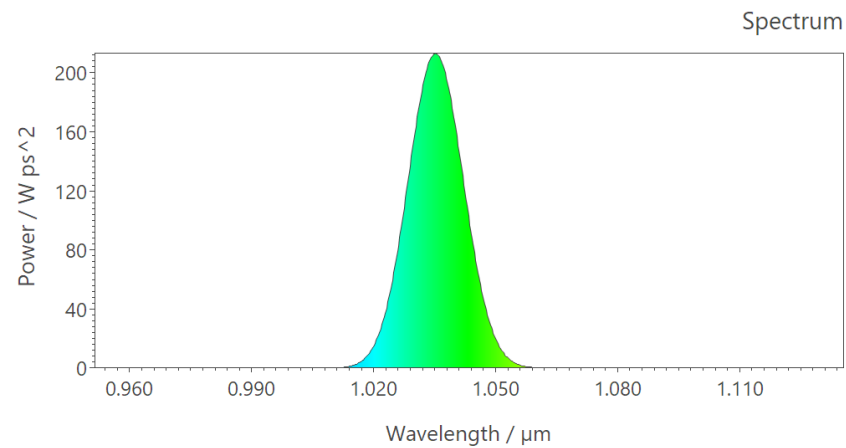
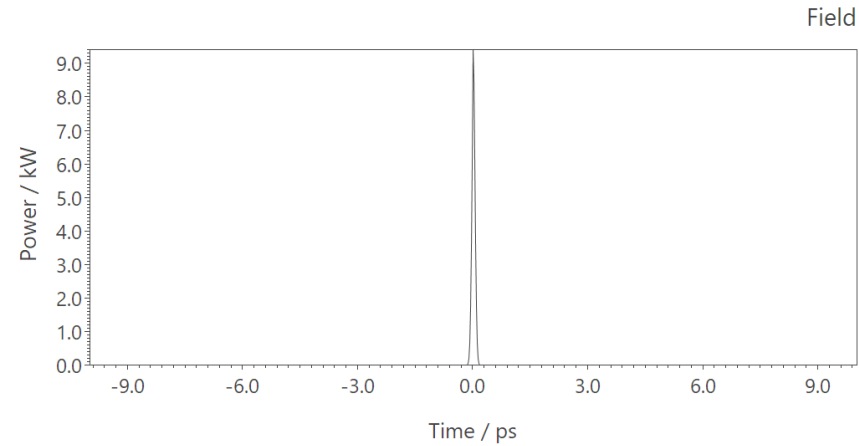
double pulsing

separation: 0 ps relative magnitude: 0

create field in data array 1

add field to data array 1

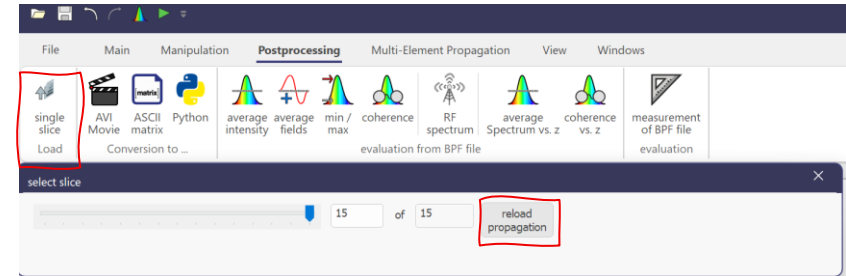
OK Apply Cancel reset



# Multi-Element Propagation: Mamyshev Oscillator



Then Start the Loop with the following parameter. After some roundtrips, we see the pulse converging. As at the end of the last element, the field is written to a propagation file. With “Postprocessing” -> “Load single slice” -> “Reload propagation” the full evolution at each roundtrip can be visualized.



### Multi Element Propagation Setup

(maximum) number of loops  hint: zero for just a single propagation of all elements

Base filename: (element and loop information will be added)

Show in Explorer

Show in Explorer

consider only a limited number of loops for writing data:  loops

Automatic stop of loop

stop if converged

condition

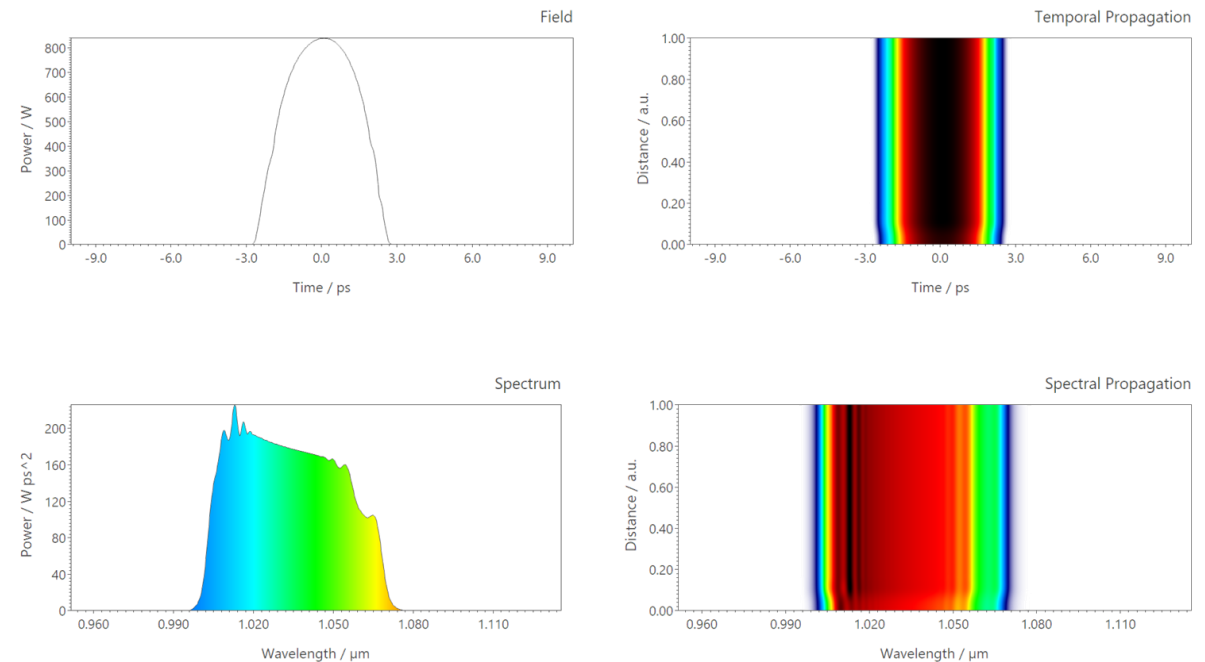
minimum change

for at least  loops

OK Cancel

## Mamyshev Oscillator

multi-element example

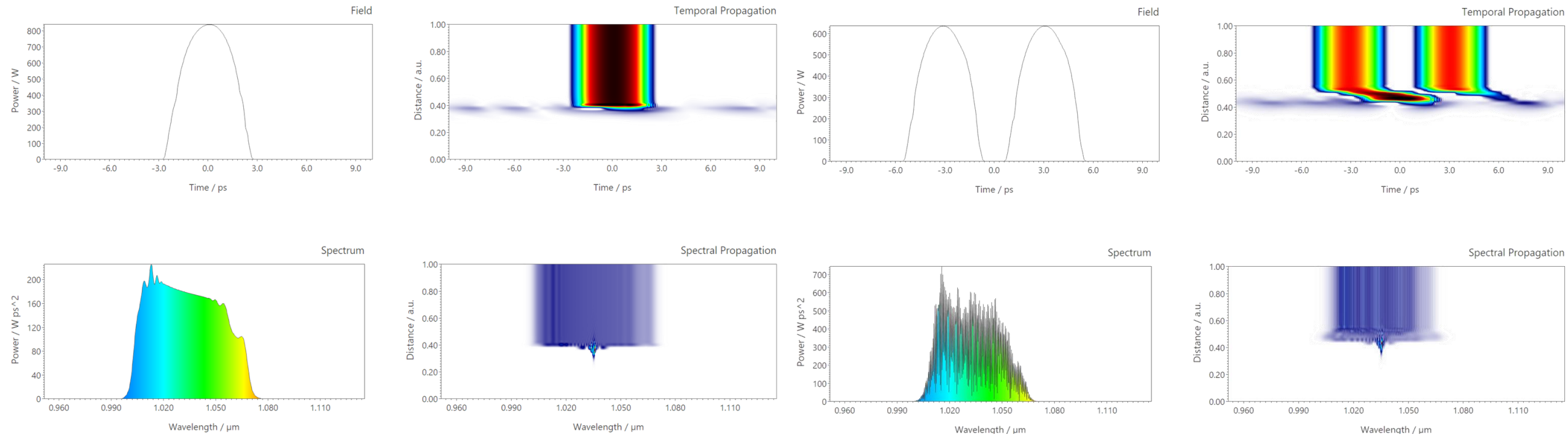


distance: 0.000 m position: 0.000 m energy: 3.094 nJ average power: 0.309 W roundtrip: 15

# Multi-Element Propagation: Mamyshev Oscillator



Restart for instance with a random field (average power = 0, one photon per spectral node noise switched on in the create field dialog) will give a result similar result as shown here (but not the same as it is random). Mamyshev Oscillators are known to be not easily self starting.



# Lecture 5

## Multi-Element Propagation

### Part 1:

Short Pulse Fiber Laser

### Part 2:

Mamyshev Oscillator

### Part 3:

Micro Comb Generator



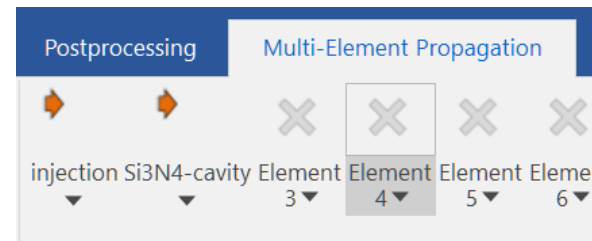
# Multi-Element Propagation: Micro Comb Generator

Alessia Pasquazi, Marco Peccianti, Luca Razzari, David J. Moss, Stéphane Coen, Miro Erkintalo, Yanne K. Chembo, Tobias Hansson, Stefan Wabnitz, Pascal Del'Haye, Xiaoxiao Xue, Andrew M. Weiner, Roberto Morandotti, Micro-combs: A novel generation of optical sources, Physics Reports, Volume 729, 2018, Pages 1-81.

Fiberdesk's multi element propagation can be used to simulate roundtrips in the micro-resonator. Please see section 5.1 of the reference for details. In principle, the first element simulates the in/outcoupling (boundary conditions) and the second element is the nonlinear propagation within the resonator.

$$E^{(m+1)}(0, \tau) = \sqrt{\theta} E_{\text{in}} + \sqrt{1 - \theta} E^{(m)}(L, \tau) e^{i\phi_0},$$
$$\frac{\partial E(z, \tau)}{\partial z} = -\frac{\alpha_i}{2} E + i \sum_{k \geq 2} \frac{\beta_k}{k!} \left( i \frac{\partial}{\partial \tau} \right)^k E + i\gamma |E|^2 E$$

So, only two elements need to be defined:



Please also note that the average propagation equation (LLE) can be used for simulating Micro Comb Generation but are intended for another tutorial.

# Multi-Element Propagation: Micro Comb Generator

The injection element is done using the “pulse injection” propagation.  
As you can see, in the setup a low transmission (High Q Cavity) and long pulse duration (to simulate a cw injection) is given.

Save as injection.ppf

The image shows two overlapping software windows. The background window is titled "Propagation param..." and has a dropdown menu set to "pulse injection" with a "Setup >" button. It contains several input fields and checkboxes:

- waveguide**: loss (0.0 1/m), gain (0 1/m), MFD (3 μm), gamma (0.026834381551362 1/(W m)), Esat (2.1205 μJ)
- simulation**:  dispersion,  spm / TPA,  Raman,  self-steepening,  temporal gain saturation
- steps: 10, stepsize: 0.000599584916 m

The foreground window is titled "pulse injection (by adding to current array)" and contains a diagram and several parameters:

The diagram shows an input field  $E_{in}$  entering a vertical rectangular element. Two output fields emerge:  $E_{out}$  and  $E_0$ . The equation  $E_{out} = \sqrt{T} \cdot E_{in} + \sqrt{1-T} \cdot E_0$  is displayed next to the diagram.

Parameters in the foreground window:

- $E_0$  - existing field in data array
- $E_{in}$  - defined below
- $E_{out}$  - interfering fields
- transmission T: 0.1 %
- phase phi\_0: 0 rad
- Type: Gauss
- FWHH: 1e6 ps +/- 0 ps
- TempShif: 0.0 ps +/- 0 ps
- wavelength: 1550 nm +/- 0 nm
- Chirp: 0.0 fs<sup>2</sup> +/- 0 fs<sup>2</sup>
- energy: 0.001e-9 J +/- 0 J

Buttons at the bottom include "OK", "Cancel", and "zero deviations".



# Multi-Element Propagation: Micro Comb Generator

The ring itself is defined as a standard propagation with the dispersion of the material (including the waveguide dispersion), the nonlinearity and length.

The dispersion is a simple second order approx. only.

Save as Si3N4-cavity.ppf

dispersion term

$$\frac{\partial A}{\partial z} = \dots + \sum_{n \geq 1} \beta_n \frac{i^{n+1}}{n!} \frac{\partial^n}{\partial T^n} A$$

Dispersion Setup

Taylor Series @ 1550 nm

Beta1	0	ps/m
Beta2	-0.02	ps <sup>2</sup> /m
Beta3	0.0	ps <sup>2</sup> /m
Beta4	0	
Beta5	0	
Beta6	0	
Beta7	0	
Beta8	0	
Beta9	0	

dispersion model

- Taylor expansion series
- Sellmeier coefficients
- photonic crystal fiber
- gas-filled silica-hollow core fiber

force retarded time frame (beta0=beta1=0)

Use dispersion     do not use dispersion

self phase modulation / two photon absorption term

$$\frac{\partial A}{\partial z} = \dots + i\gamma(1 - f_R)A(T)$$
$$\gamma = \frac{\omega_0 n_2}{c A_{eff}} \text{ and } A_{eff} = \frac{\pi}{4} MFD^2$$

n2 2.3e-19 m<sup>2</sup>/V

fR 0.15

TPA 0 m/W

TPA is experimental so far ...

saturate SPM    saturation power density 1.0 GW/cm<sup>2</sup>

use SPM and TPA     exclude SPM

Propagation param... Setup >

standard propagation

waveguide

loss 0.0 1/m

gain 0 1/m

MFD 5 μm

gamma 0.069433962264150 1/(W m)

Esat 5.8904 μJ

simulation

dispersion     Raman

spm / TPA     self-steepening

parameter

temporal gain saturation

steps 10

stepsize 5.99584916e-05 m

distance 0.000599584916 m

measure and parse

write file 100

adaptive local error 1e-07

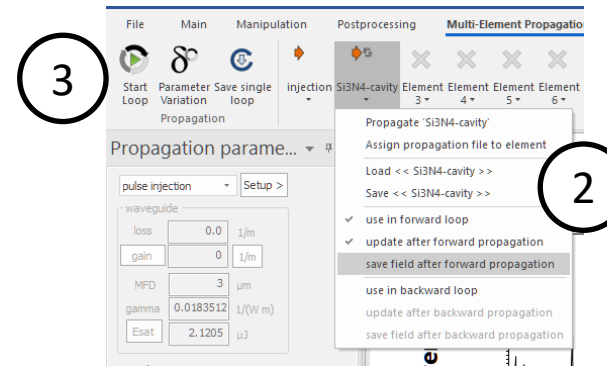
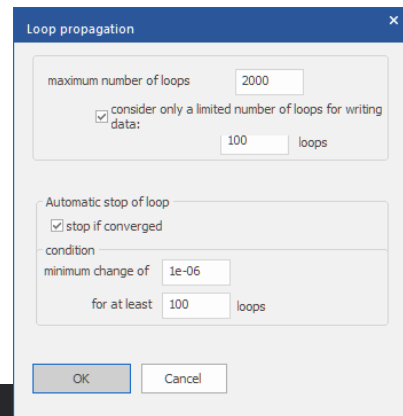
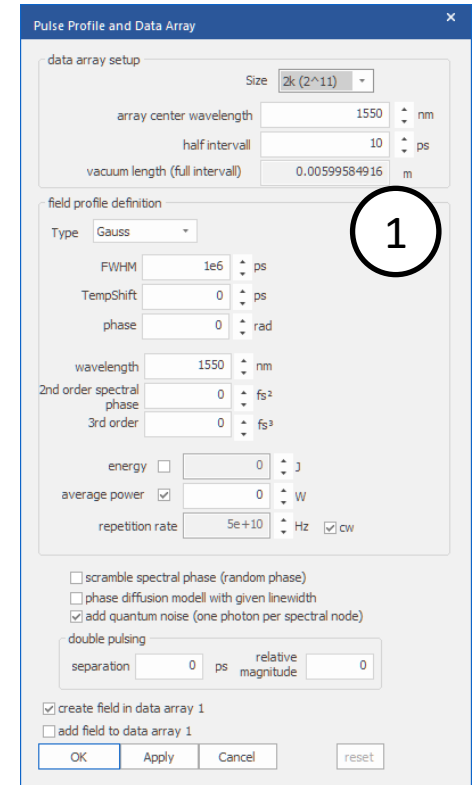
presets:

# Multi-Element Propagation: Micro Comb Generator

Starting the multi-element propagation. (Please make sure to have done the first steps of this tutorial to learn multi-element propagation.)

Quick reminder:

- (1) Setup a field with a temporal window fixed to the length of the cavity (here  $\sim 60\mu\text{m}$ ). This allows to draw a roundtrip map later on. We also only use quantum noise as the input come from the injected field.
- (2) Assign the two propagations files to elements and use both in forward loop direction. Also, to see the chance after the propagation, switch on “update after forward propagation” of the second element. You might also save the field after forward propagation, in order to postprocess it later on, e.g. plot the graph on the next slide.
- (3) Setup and start the loop.



# Multi-Element Propagation: Micro Comb Generator

Using the python script, you can draw the save file and get this roundtrip map. It shows the built-up of structures from the cw input. The parameters now need to be refined to enable the desired output.

