

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 center pulse.

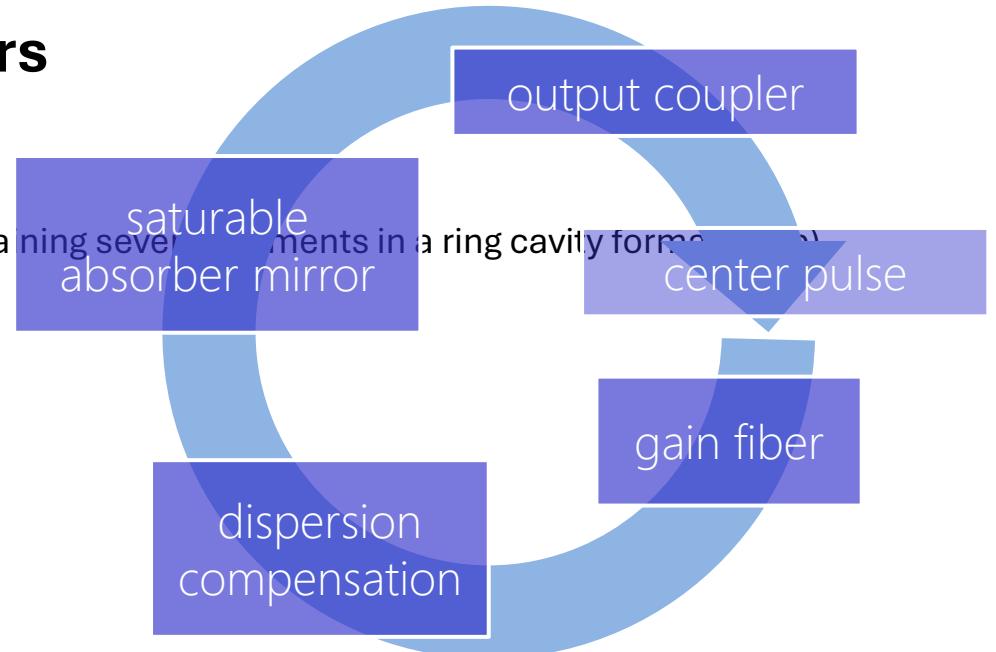
Each Element is modelled by an element typically based on the NLSE:

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

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

general

waveguide

loss 0.0 1/m

gain 30 dB/m

mode diameter 10.0 μm

effects

numerics

steps 100

stepsize 0.01 m

distance 1.0 m

adaptive local error 1e-5

Back

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 ...

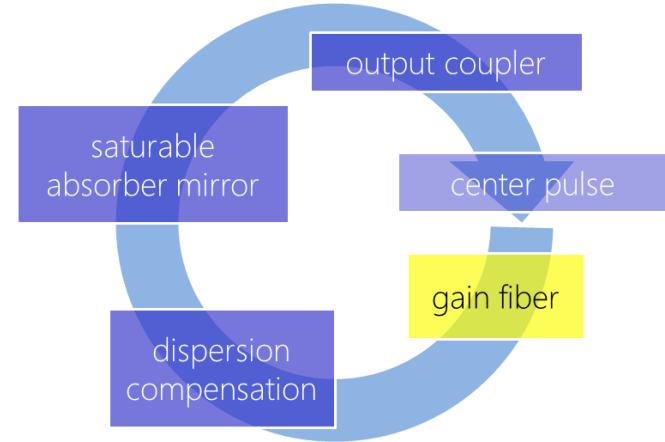
temporal gain saturation (pulses shorter than population relaxation time)

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

saturation fluence F 30 J/cm²

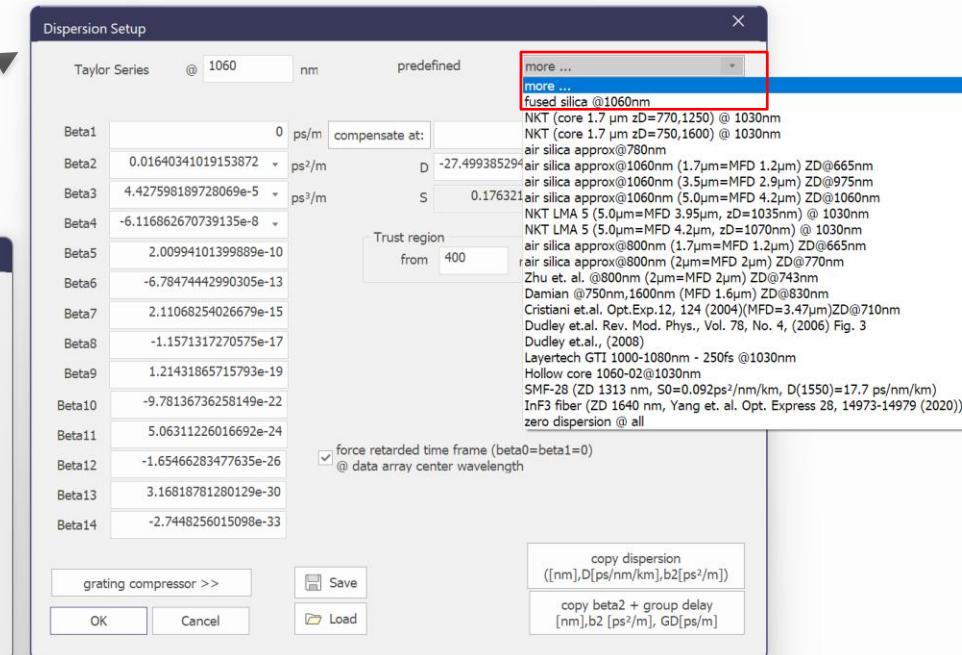
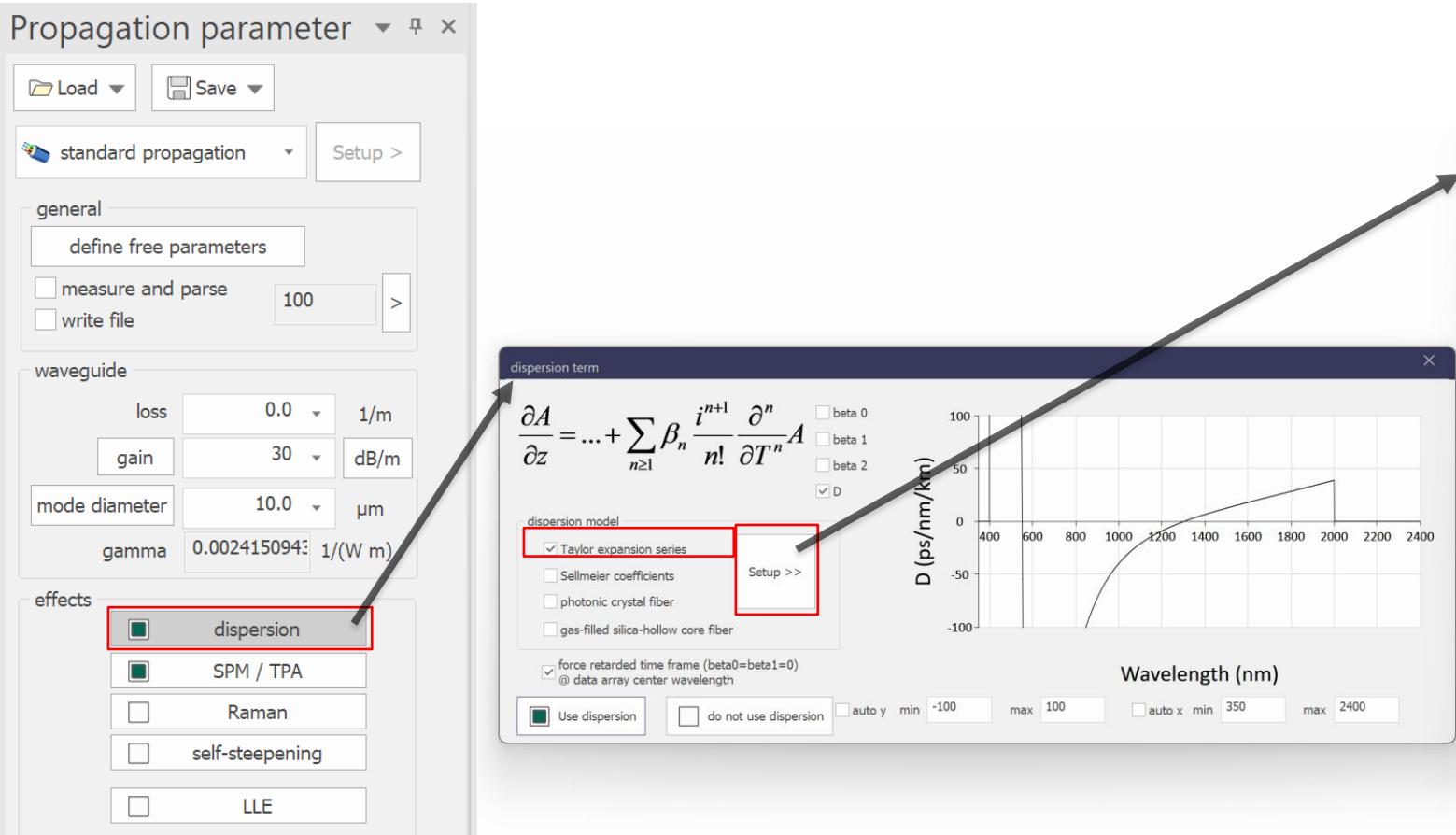
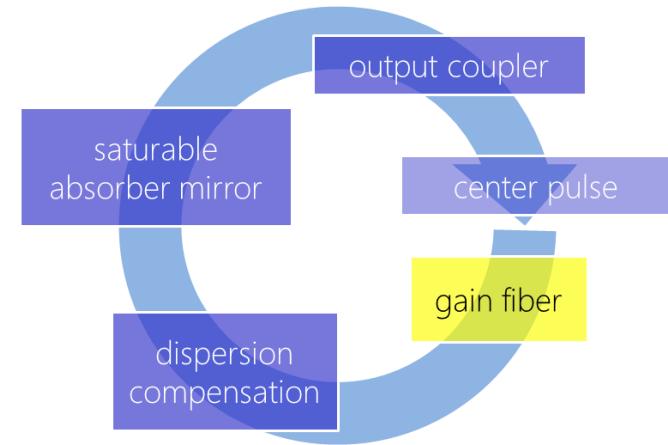
$$g(T) = g_0 \exp \left(-\frac{1}{E_{sat}} \int_{-\infty}^T |A(t)|^2 dt \right)$$
 with $E_{sat} = F \cdot A_{eff}$

saturation energy with current MFD: J



Multi-Element Propagation: Short Pulse Fiber Lasers

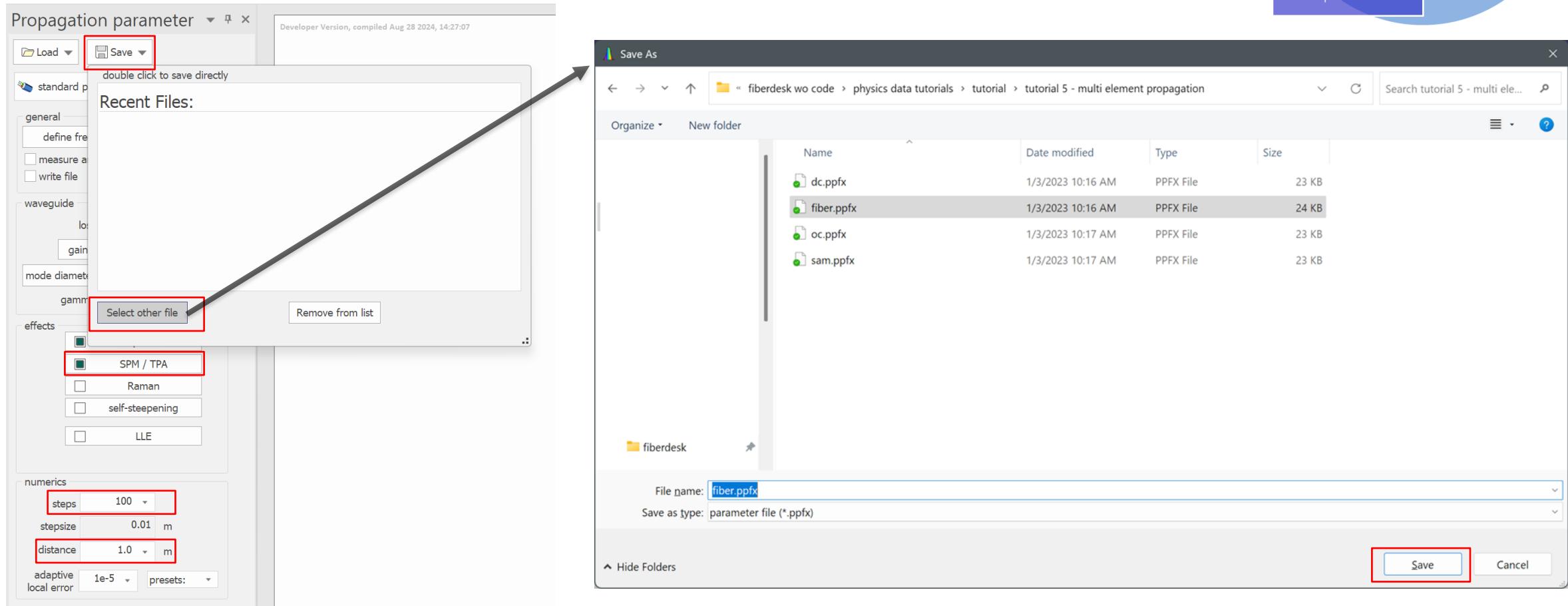
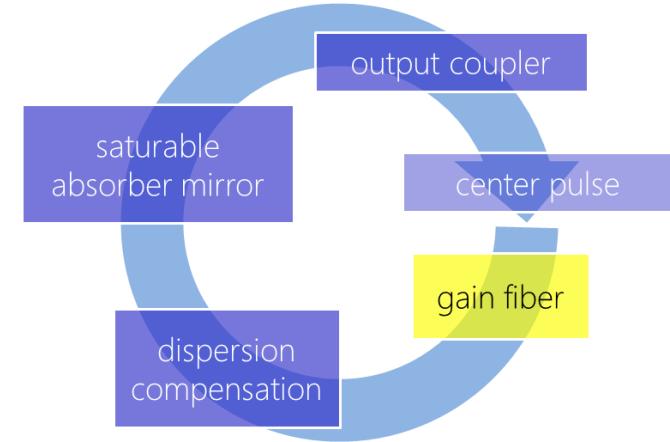
Now set the dispersion of that gain fiber to the dispersion of fused silica (e.g. predefined with its Taylor series at 1060 nm)



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.ppf“ for later use as an element.

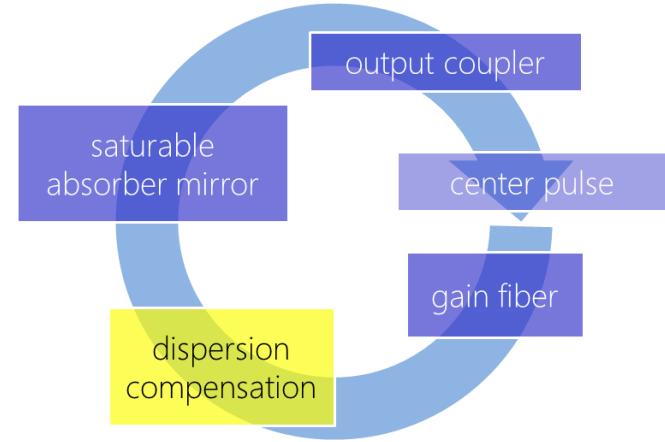


Multi-Element Propagation: Short Pulse Fiber Lasers

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 0.0024150943 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

Setup >>

Use dispersion do not use dispersion

Dispersion Setup

Taylor Series @ 1060 nm

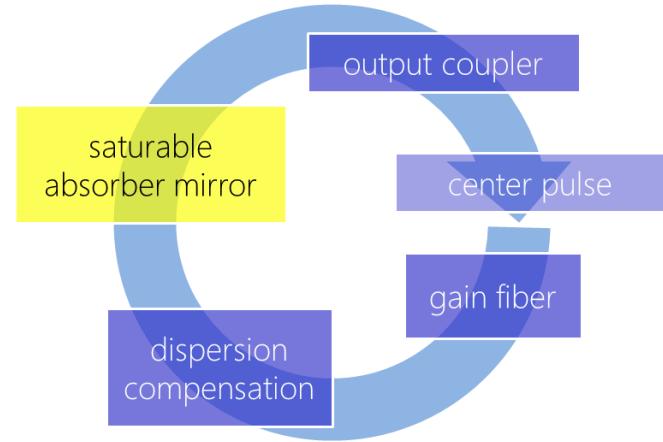
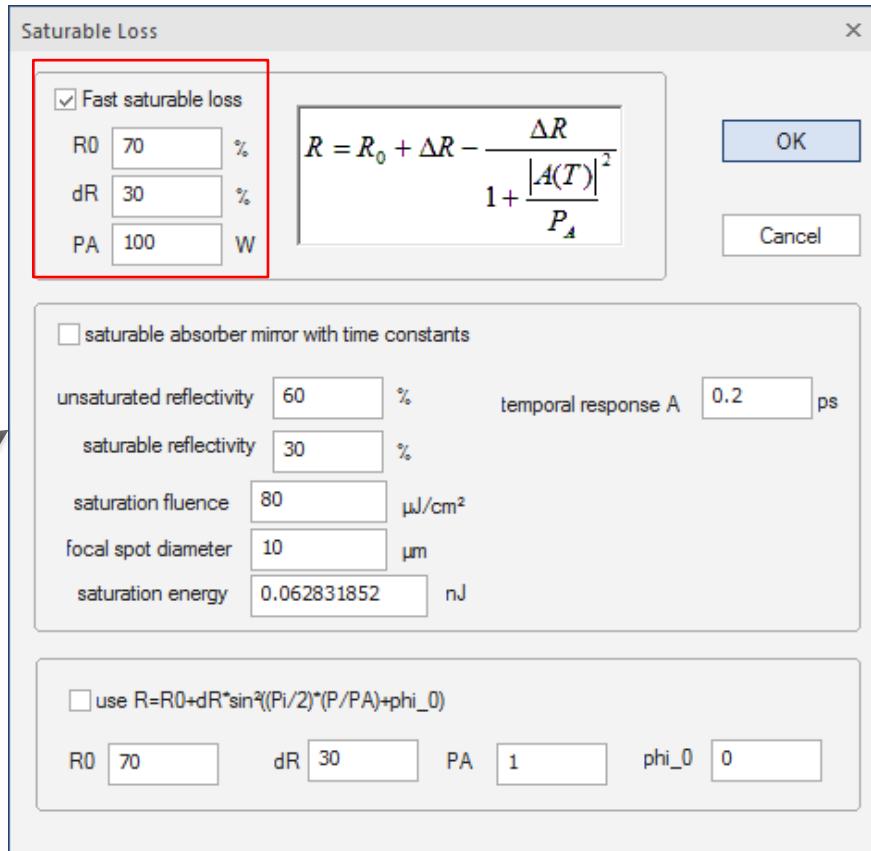
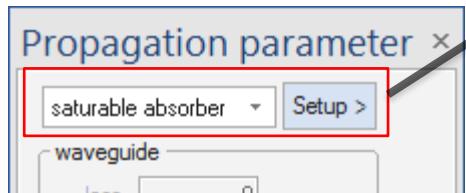
Beta	Value	Unit
Beta1	0	ps/m
Beta2	-0.02	ps ² /m
Beta3	0.0	ps ³ /m
Beta4	0.0	
Beta5	0	
Beta6	0	
Beta7	0	
Beta8	0	
Beta9	0	
Beta10	0	
Beta11	0	
Beta12	0	

Save the element as dispersion compensation „dc.ppf“, 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.

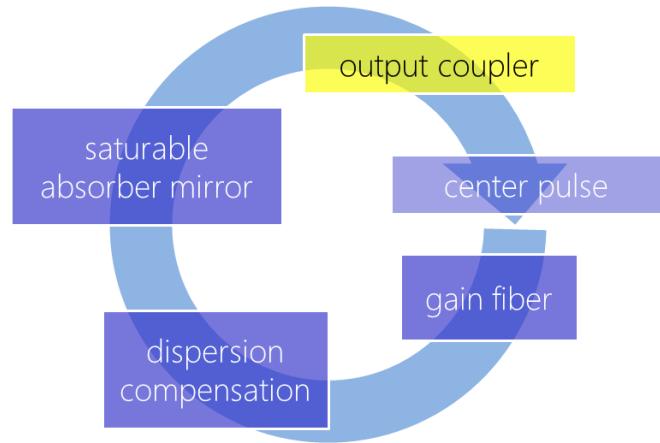
Then, set it up with the parameters on the right.



Save the element as „SA.ppf“, 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.



Propagation parameter

Manipulation

field manipulation

Create Pulse

Create double pulse by adding a copy of t

delay 0.0 ps amount 50.0

take phase shift into account (Mach-Zehnder)

Complex Multiplication Temporally

Loop over all data points

Outcoupling:
50% means complex multiplication with $\sqrt{0.5}$

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

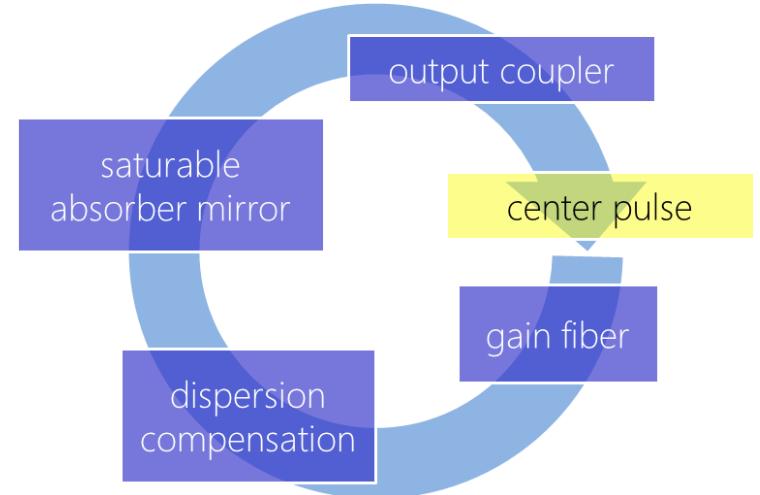
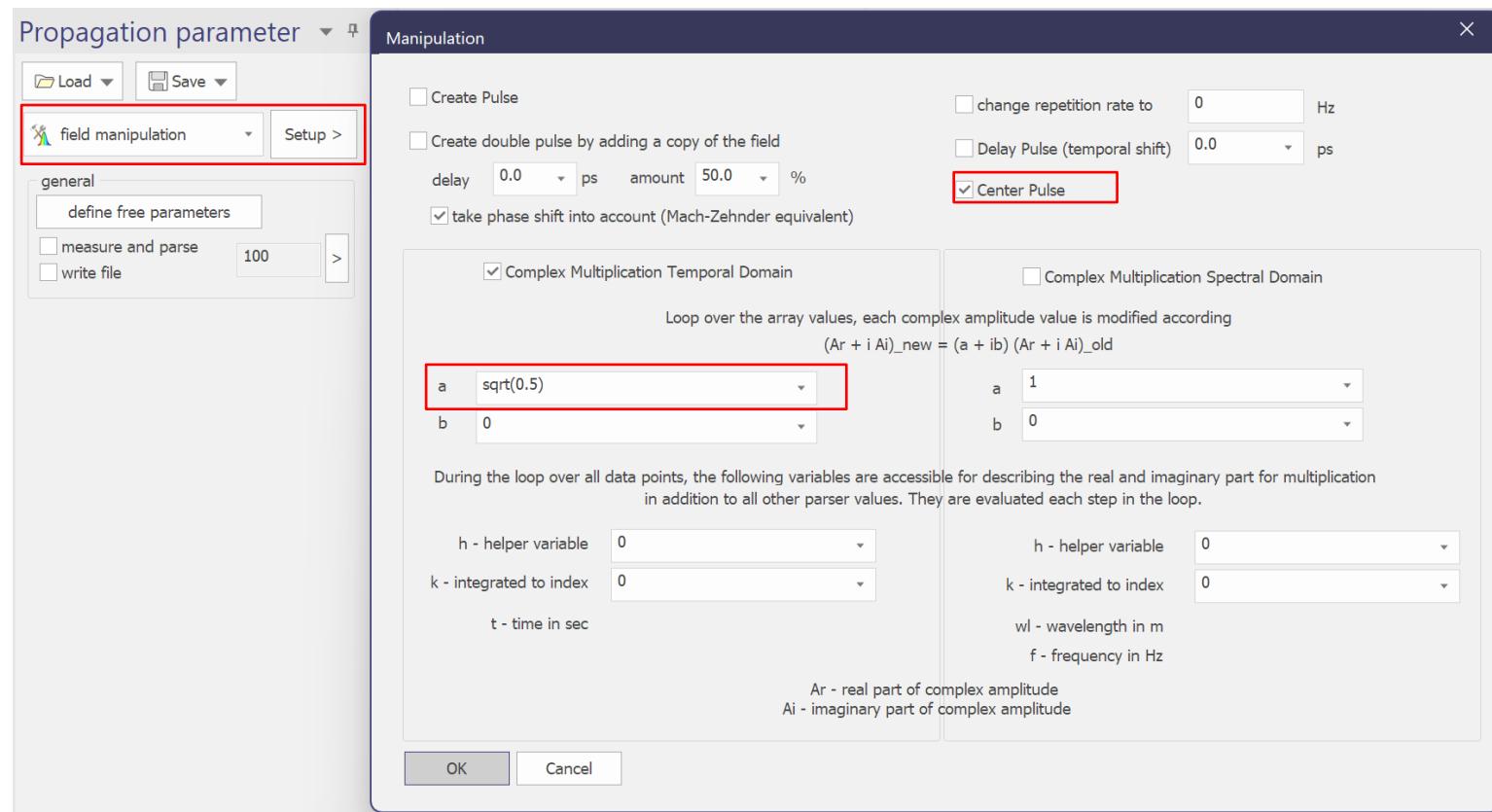
Ai - imaginary part of complex amplitude

OK Cancel

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.

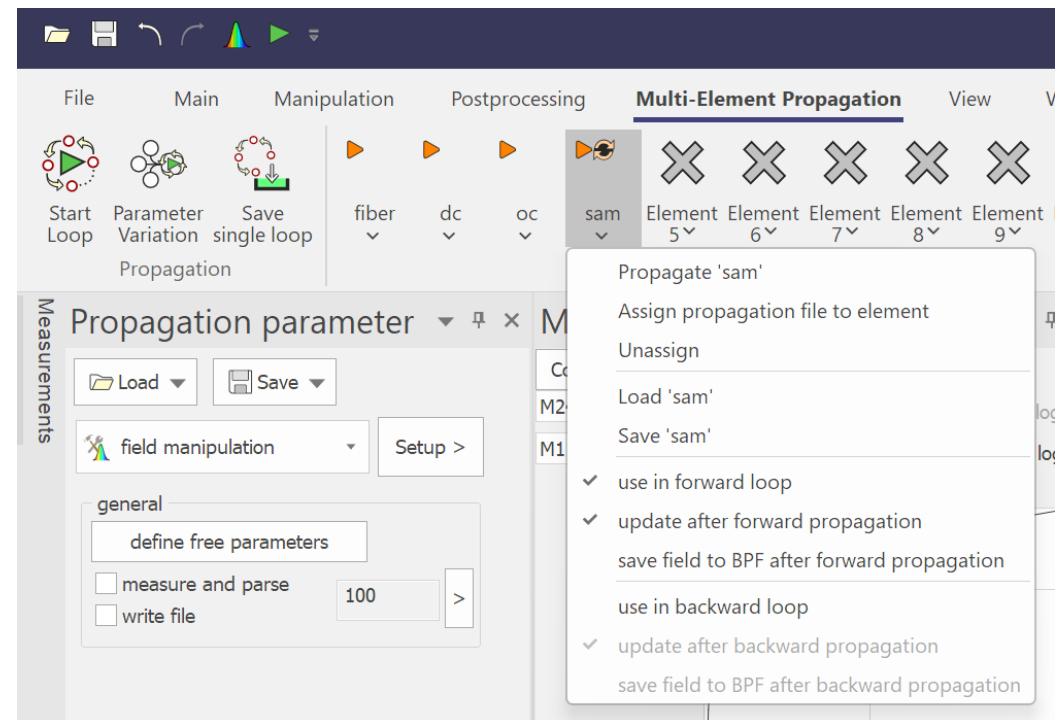
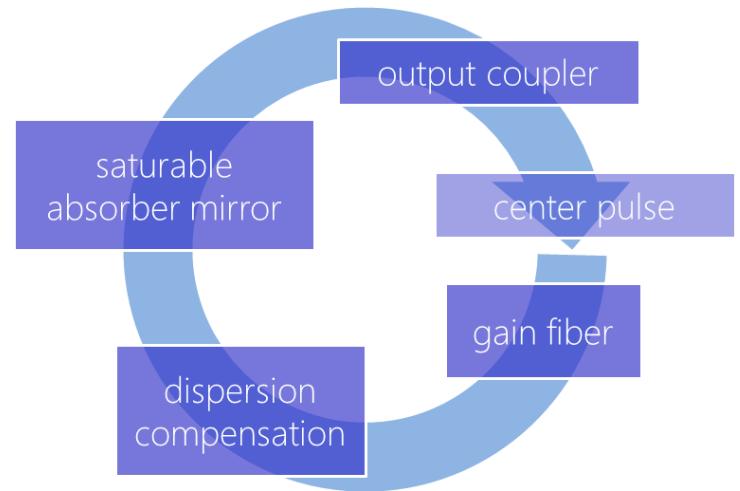


Save the element „OC.ppf“, as done before, for later use as an element.

Multi-Element Propagation: Short Pulse Fiber Lasers

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

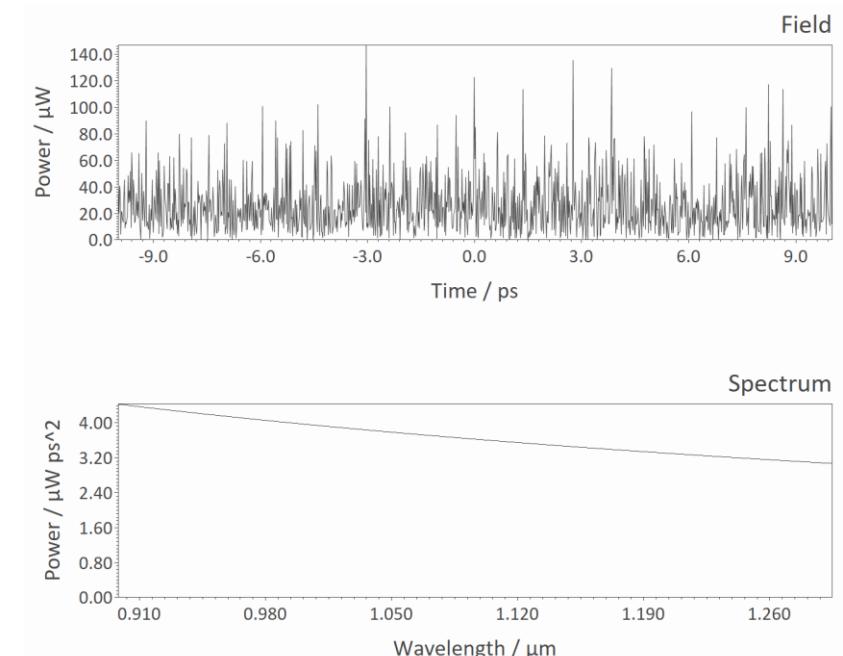
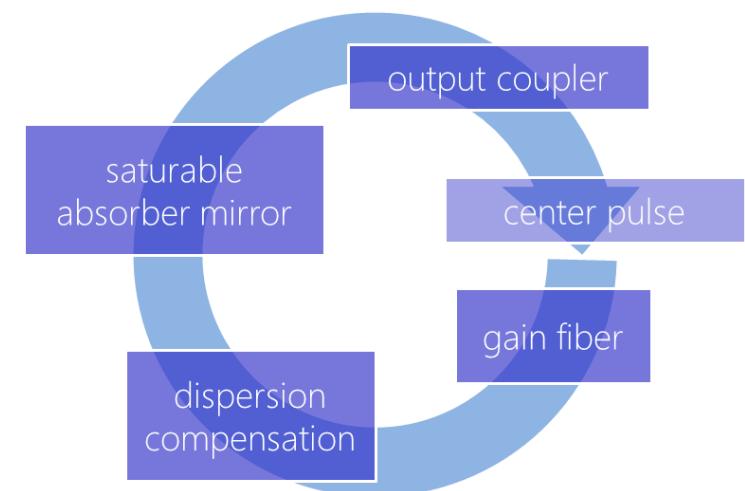
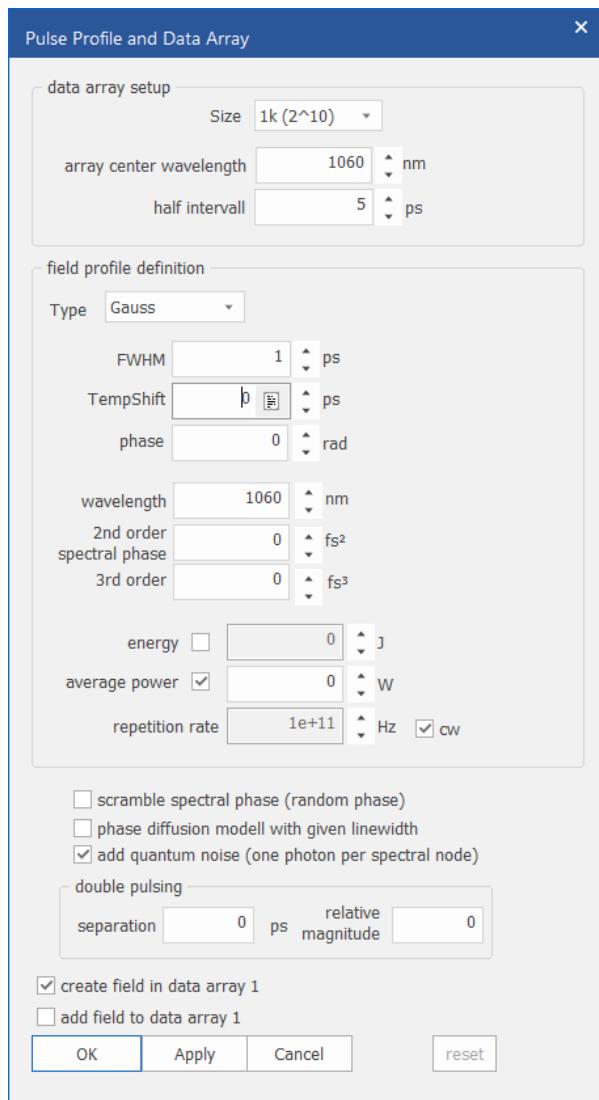
- assign all files to elements in the right order of the cavity
- Select „use in forward loop“ for all elements
- Select the last one to be updated after each loop to see convergence live during simulation
- Icons on top change according to selected status



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.

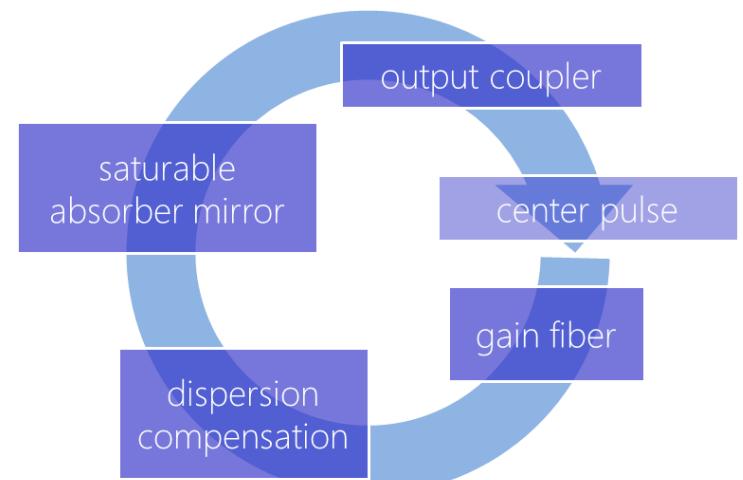
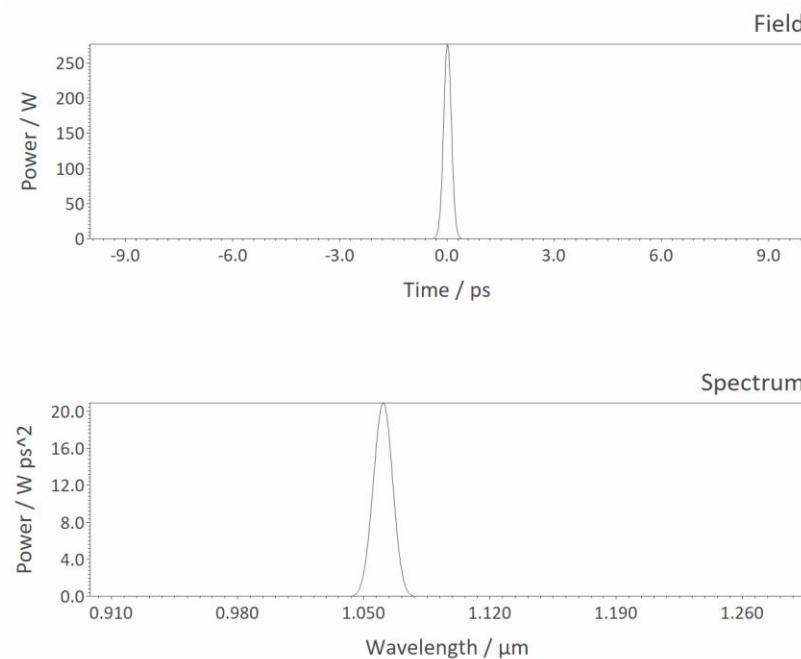
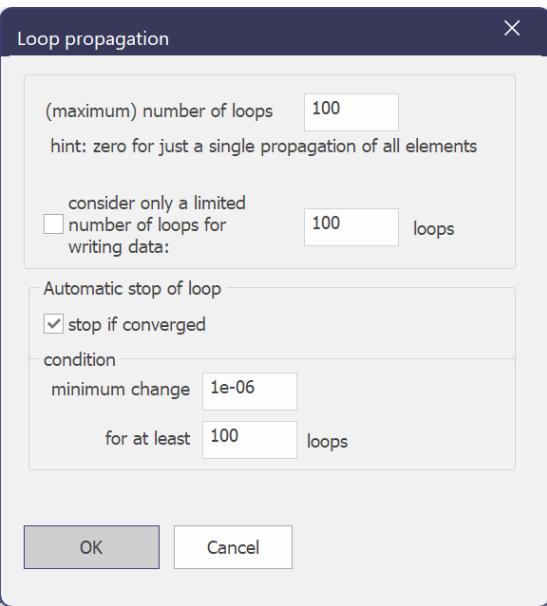
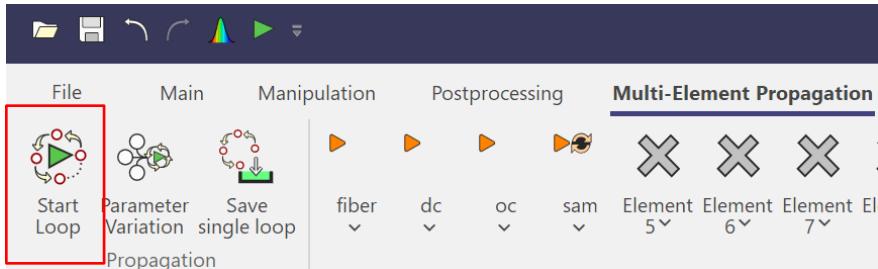
So, create the initial field with the right parameters for the final field and usually, for such short pulse oscillators, the initial condition is nothing else then quantum noise.



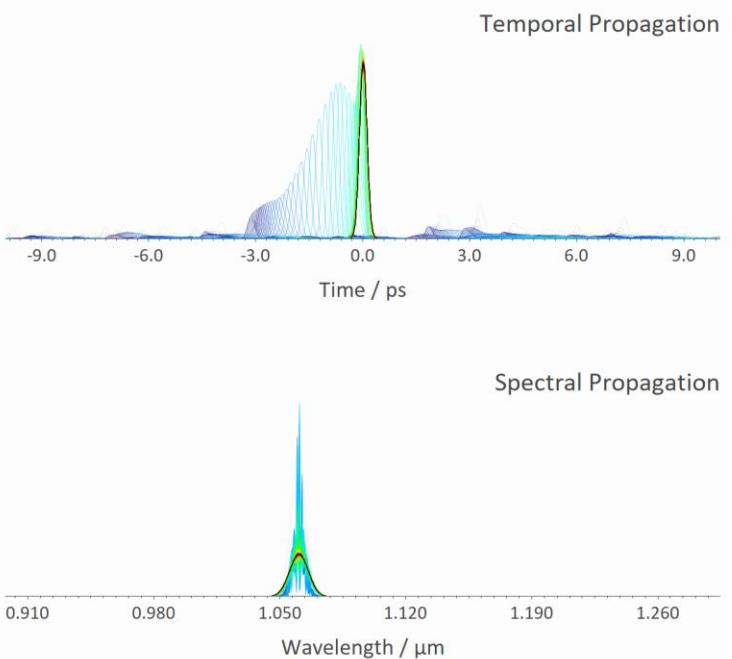
Initial field in temporal and spectrum domain.

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:

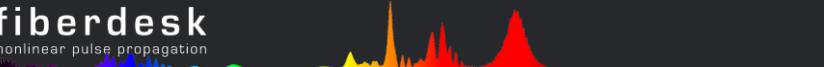
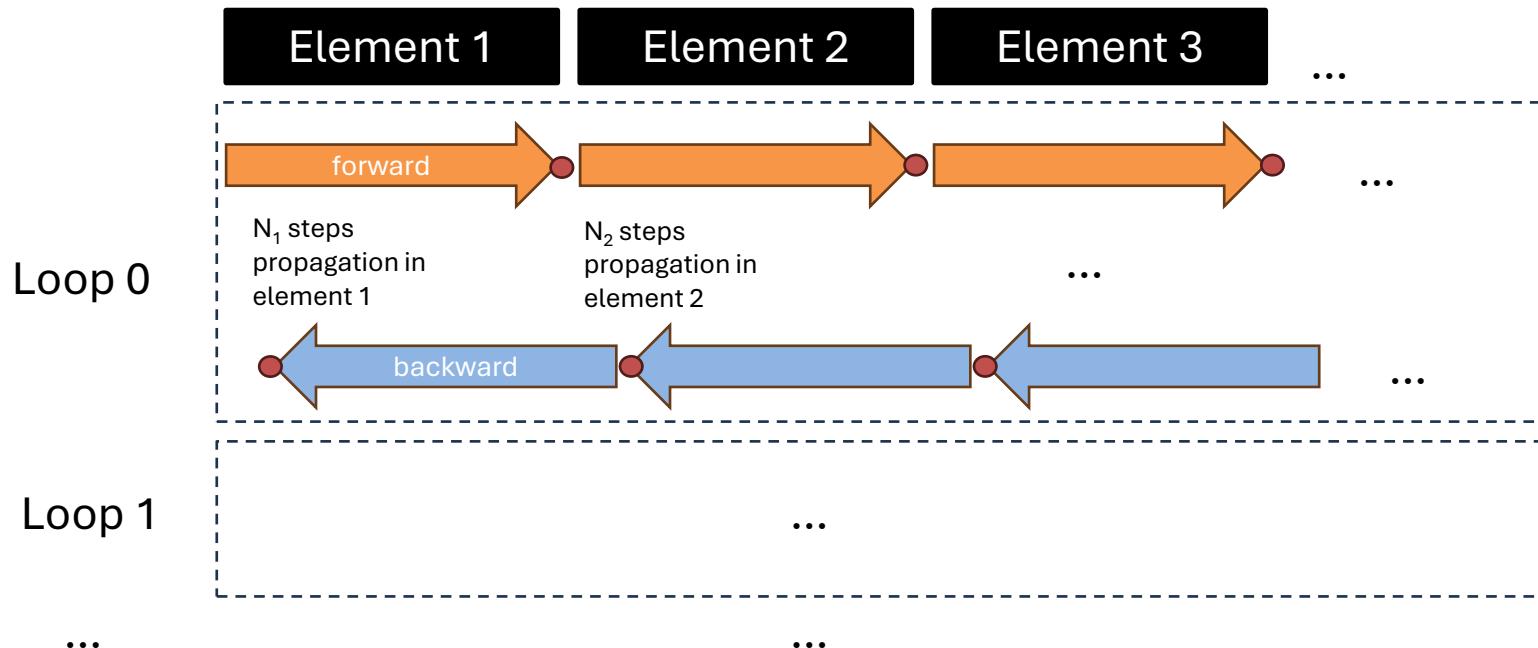
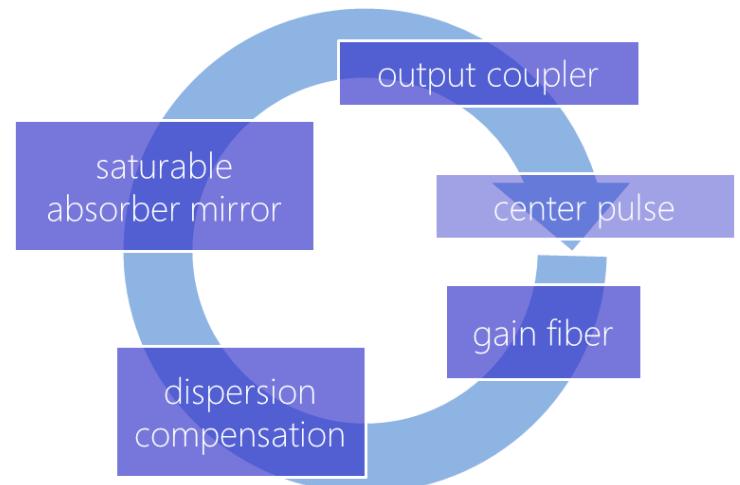


Multi-Element Propagation: Short Pulse Fiber Lasers

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

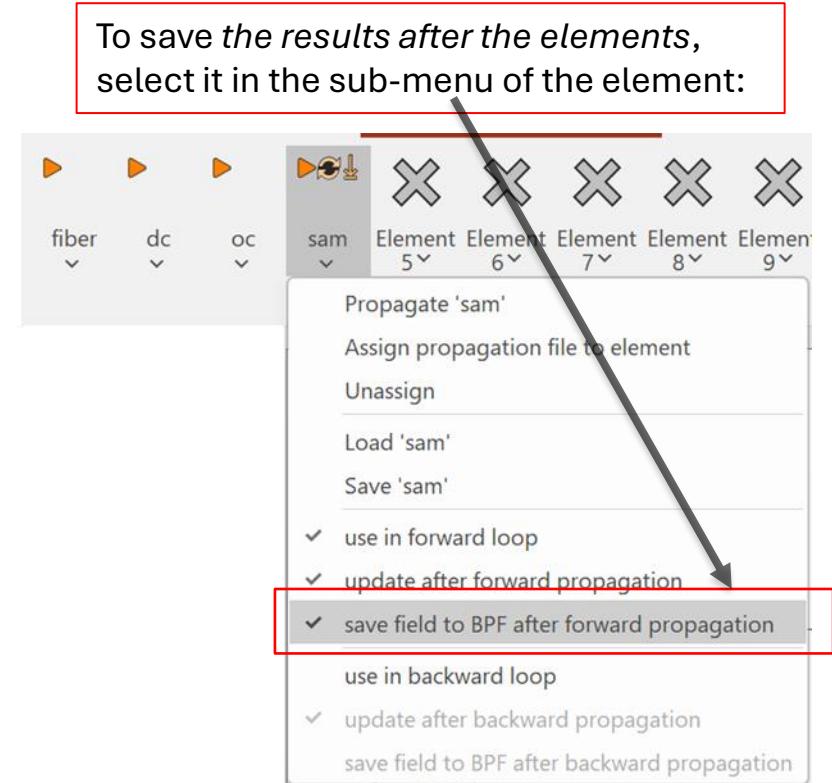
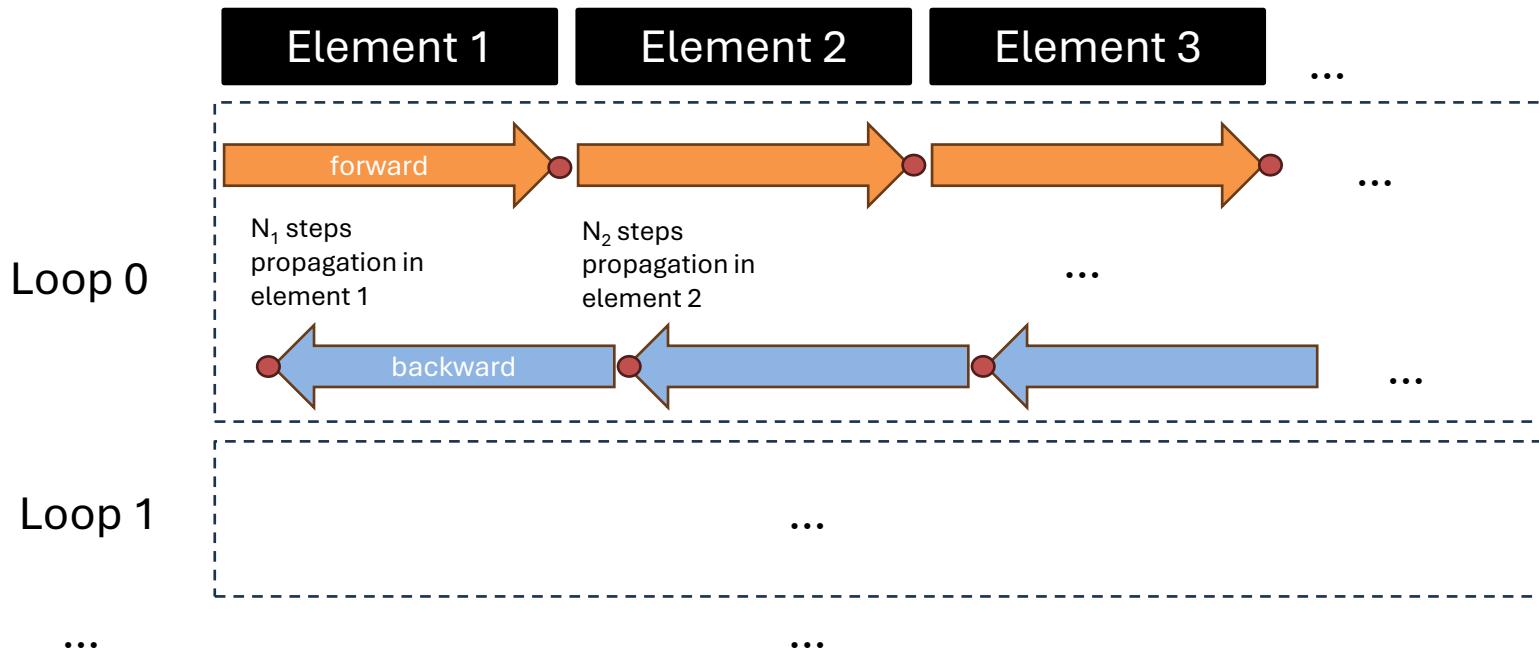
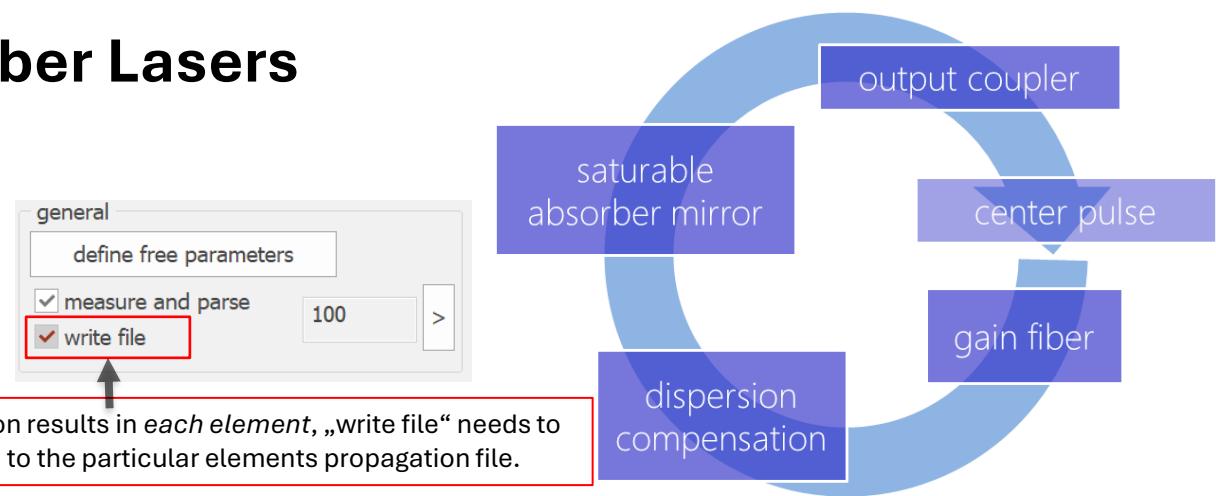


Multi-Element Propagation: Short Pulse Fiber Lasers

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



Multi-Element Propagation: Short Pulse Fiber Lasers

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

Example:

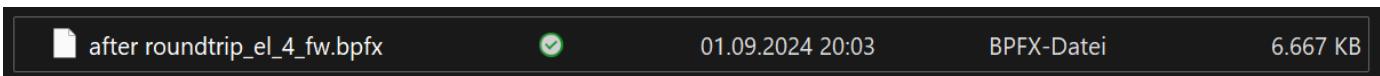
- Only the result at the end of each element



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.

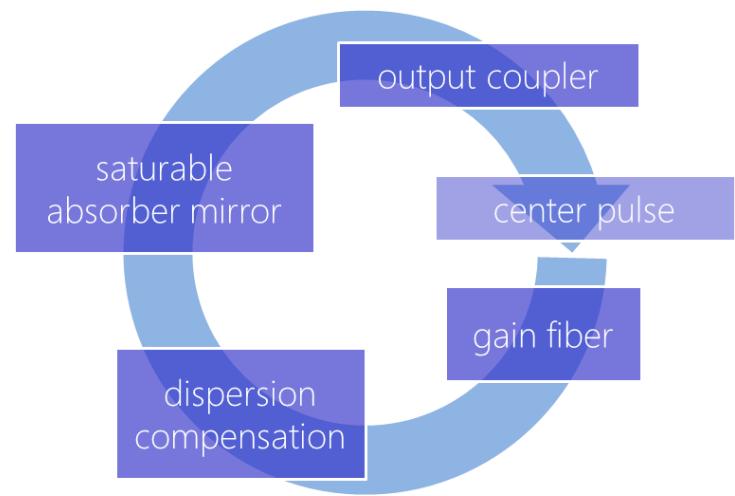
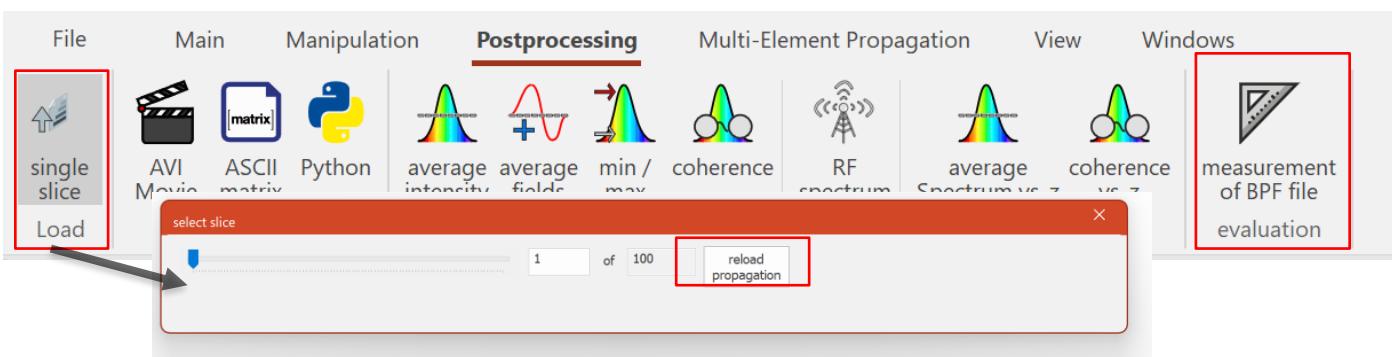
In my case, I used „after_propagation“ as the filename

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

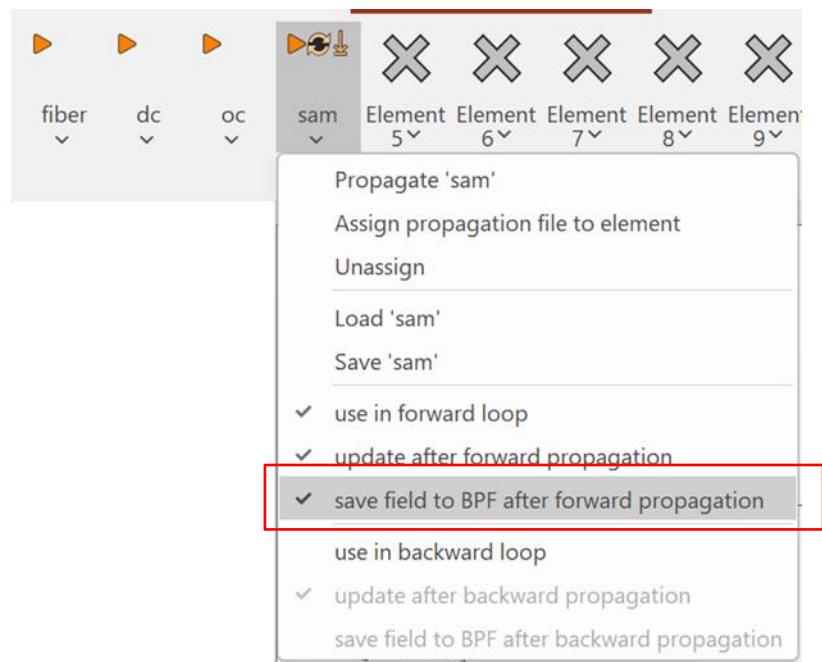


You can reload the field depending on the loop count, and also reload it into fiberdesk's 2D plot.

Also consider measurements with this propagation file.

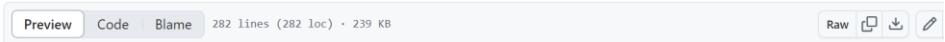


To save the results after the elements, select it in the sub-menu of the element:



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.



Example Reading propagation files - bpxf

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
```

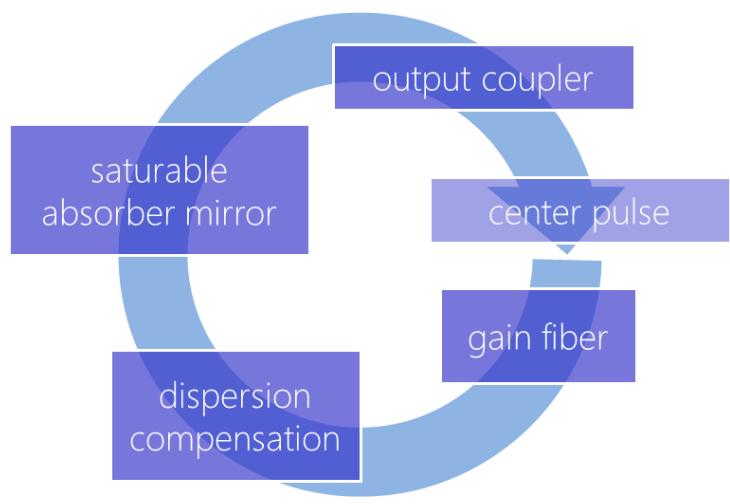
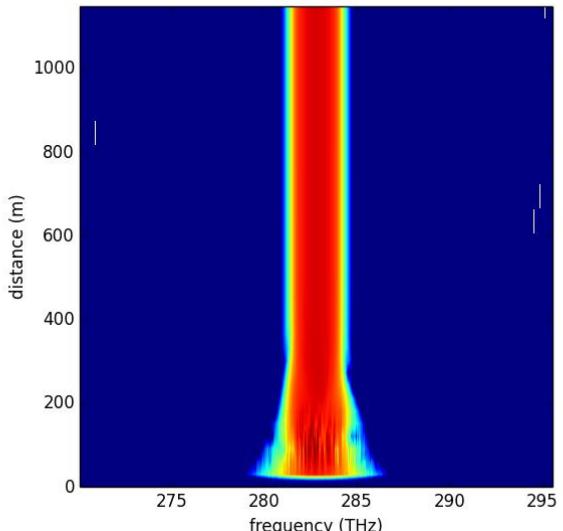
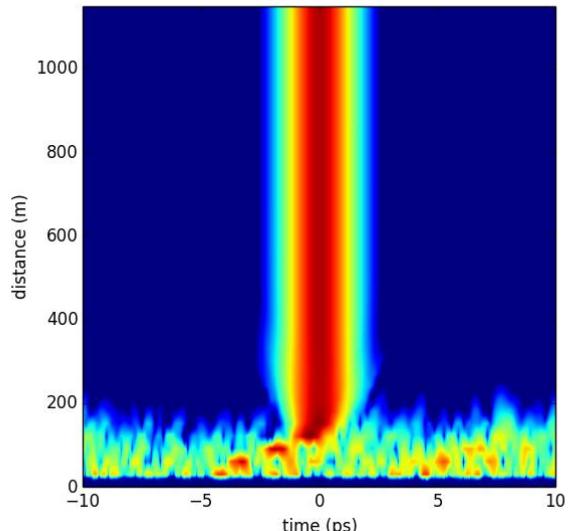
Now define the following class that load the fiberdesk propagation file (bpxf) that is actually an xml file, so pandas read_xml is used.

```
In [6]: class fiberdesk_bpxf:
    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)

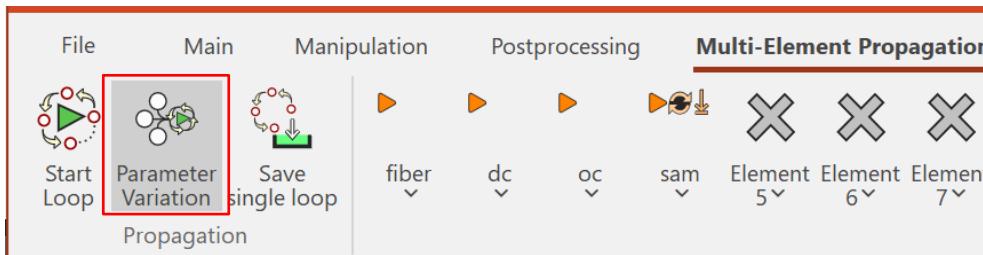
    # read axis from first entry (index 0), assume that it does no change along propagation
    if "freq_intervall" in self.df.columns:
        self.freq_intervall = self.df.at[0,"freq_intervall"]
    if "time_intervall" in self.df.columns:
        self.time_intervall = self.df.at[0,"time_intervall"]
    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

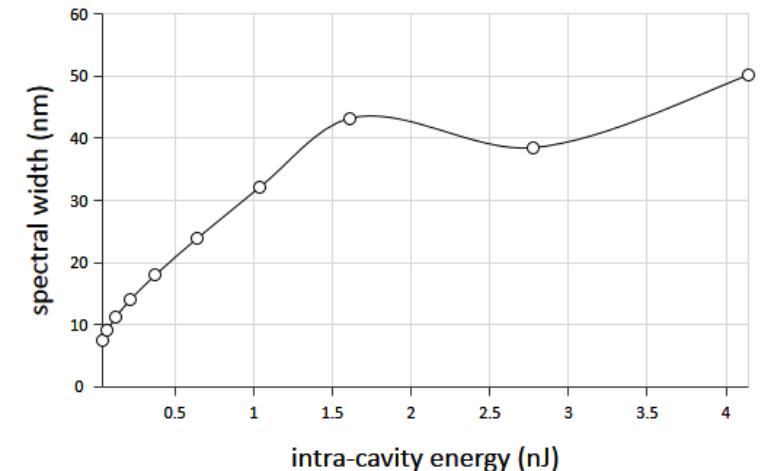
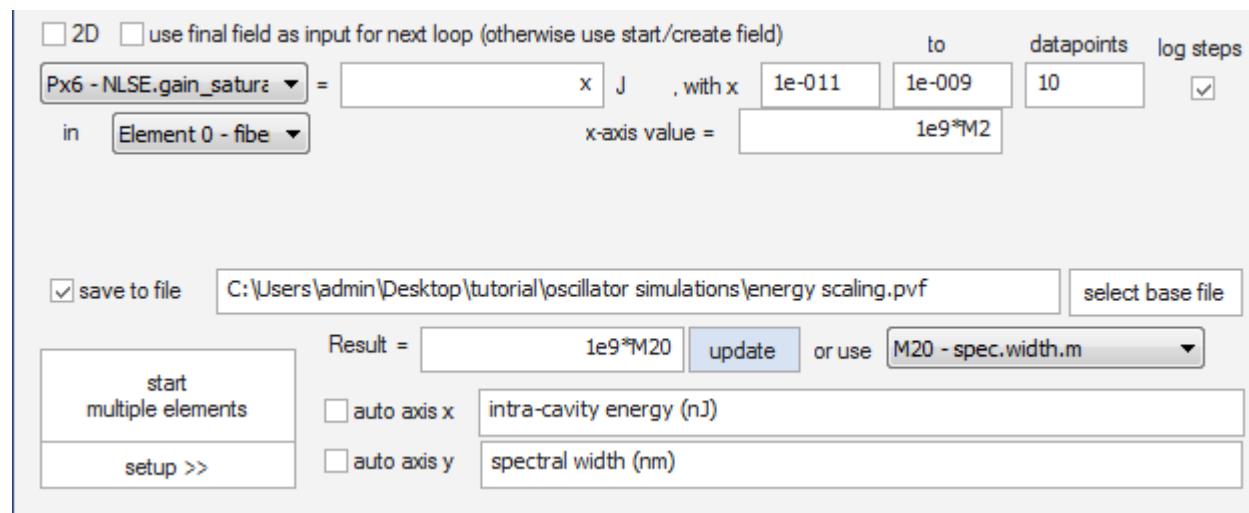
Multi-Element Propagation:

Example: Short Pulse Fiber Lasers



Multi-element > Parameter variation

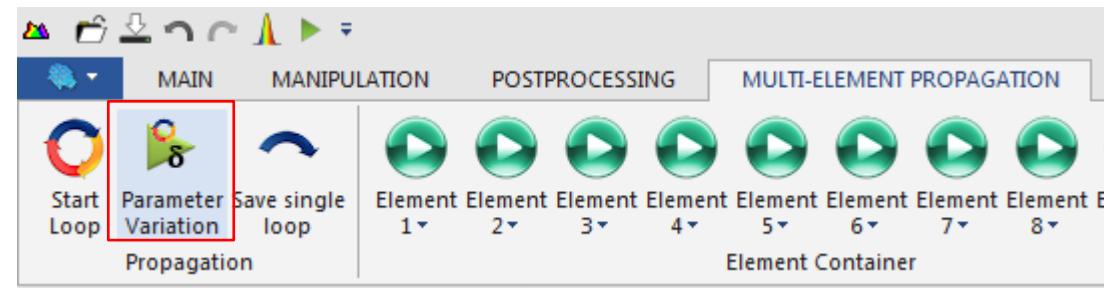
we change the gain saturation to increase the energy (remark: intracavity energy!)



lecture 5

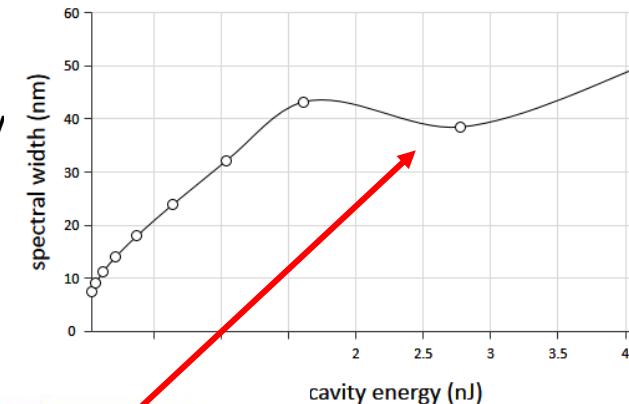
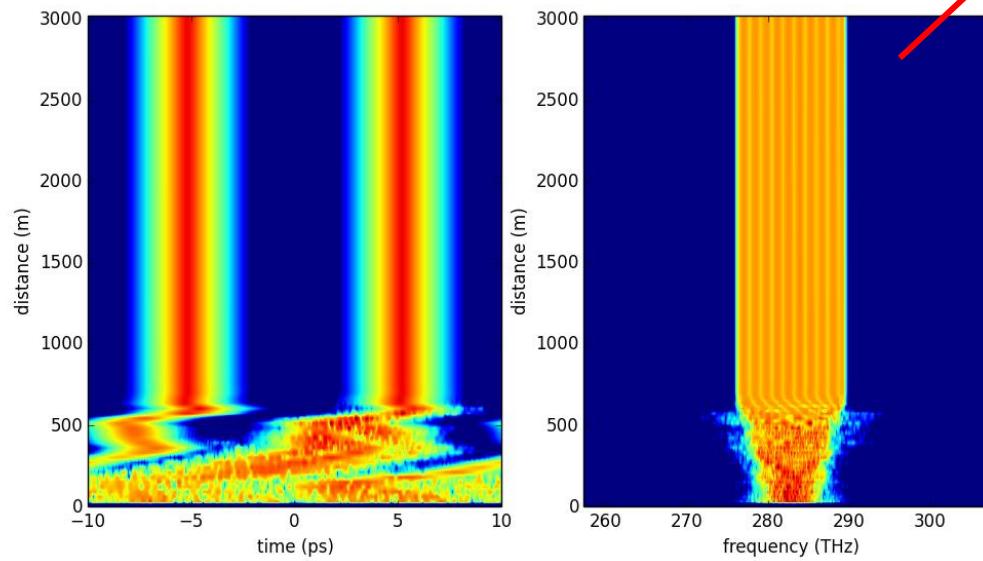
Multi-Element Propagation:

Example: Short Pulse Fiber Lasers



Multi-element > Parameter variation

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



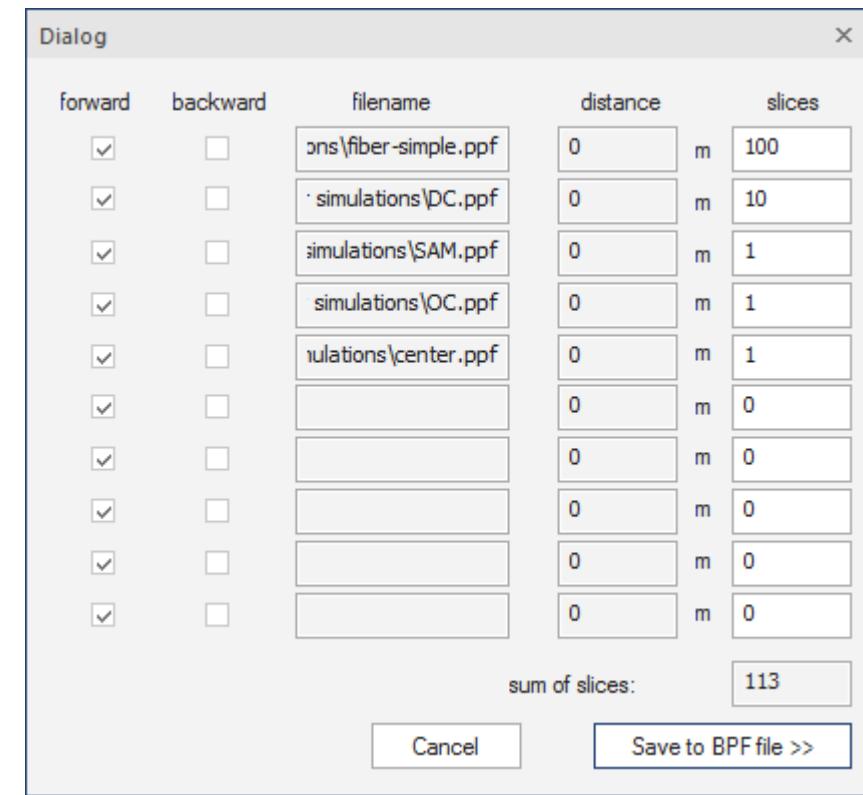
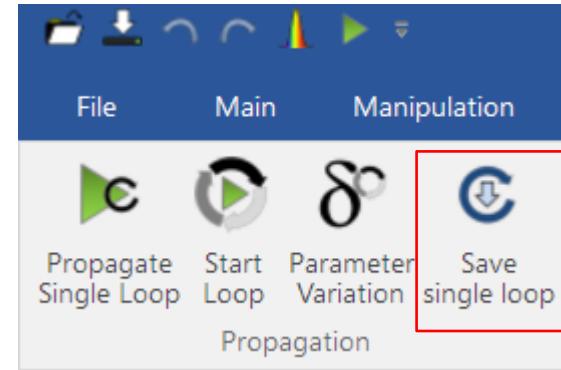
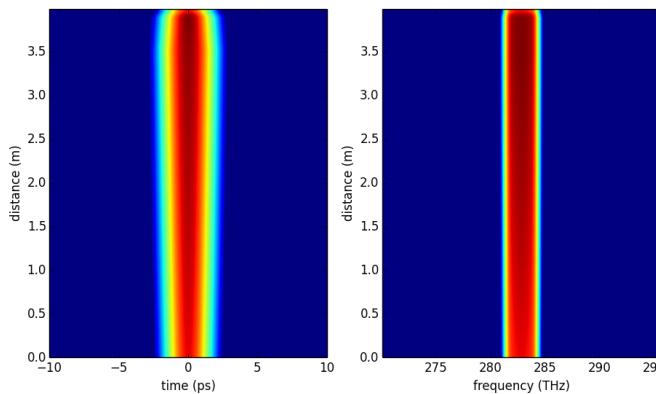
lecture 5

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

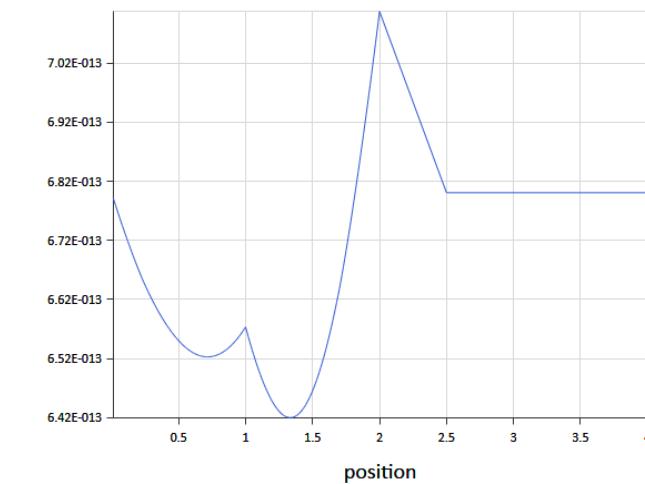
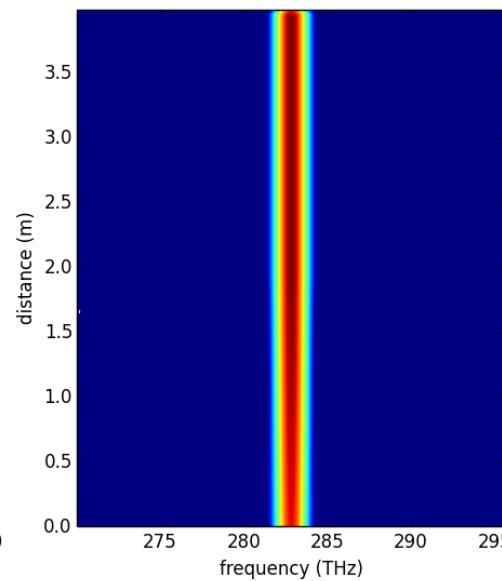
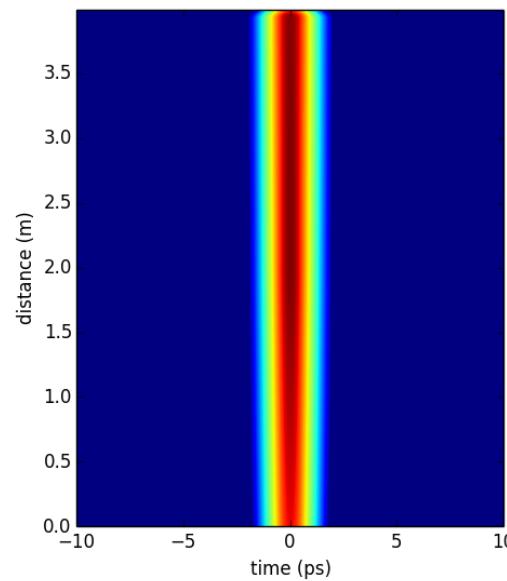
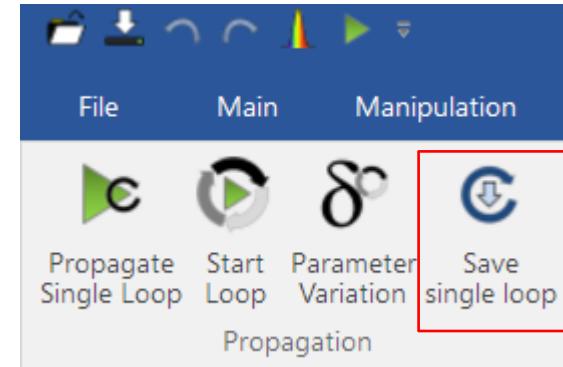


lecture 5

Multi-Element Propagation:

Example: Short Pulse Fiber Lasers

soliton solution: $\beta_2 @ DC = -0.06 \text{ ps}^2$

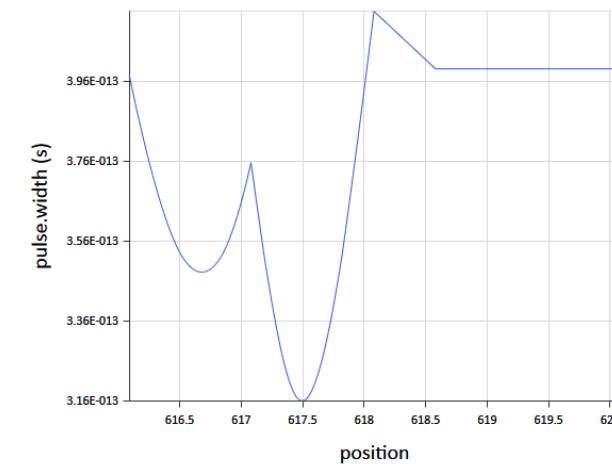
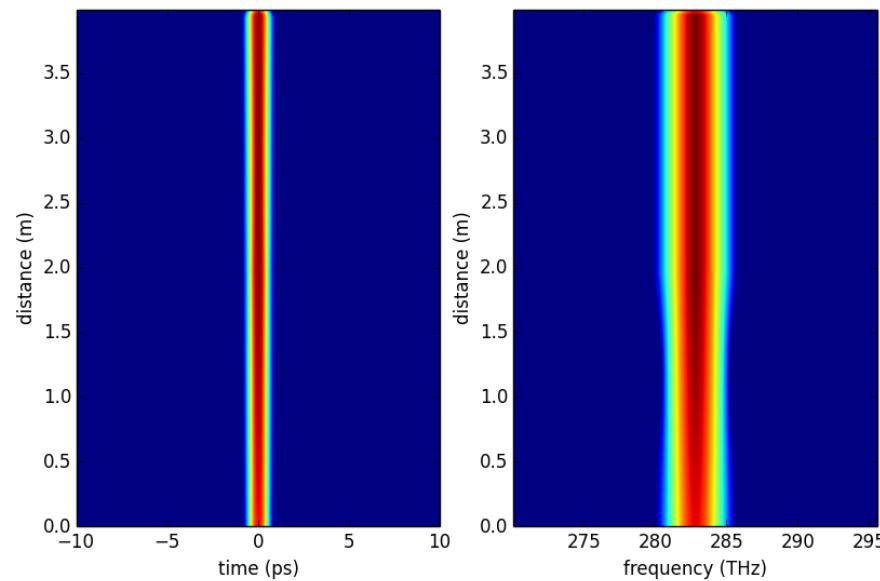
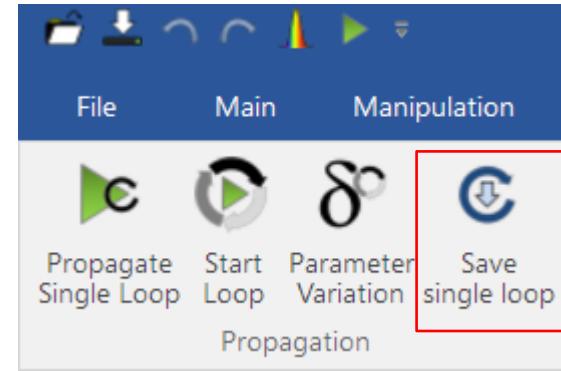


lecture 5

Multi-Element Propagation:

Example: Short Pulse Fiber Lasers

soliton solution: $\beta_2@DC = -0.04 \text{ ps}^2$

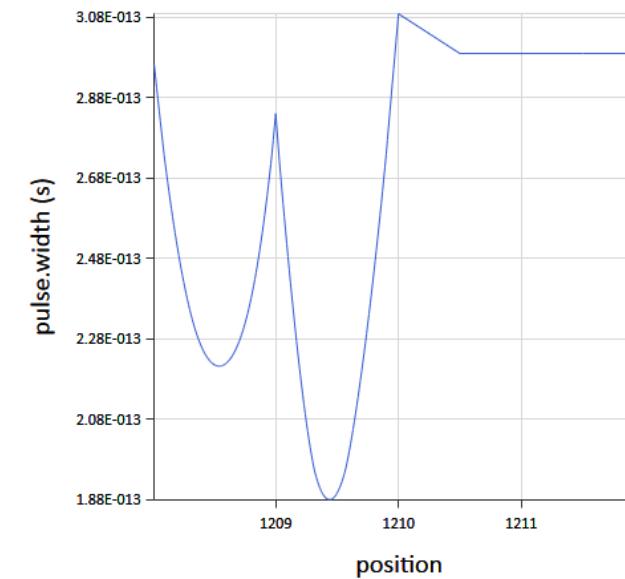
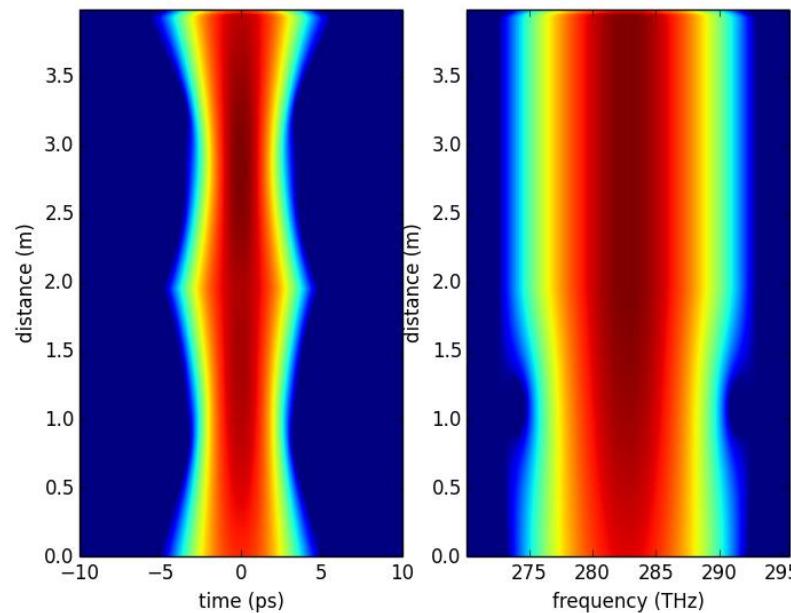
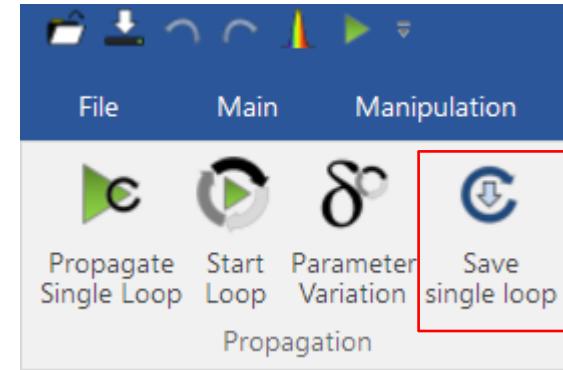


lecture 5

Multi-Element Propagation:

Example: Short Pulse Fiber Lasers

toward stretched pulse: $\beta_2@DC = -0.03 \text{ ps}^2$

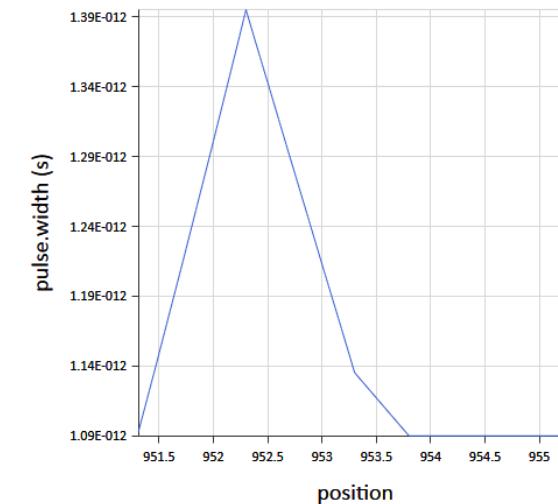
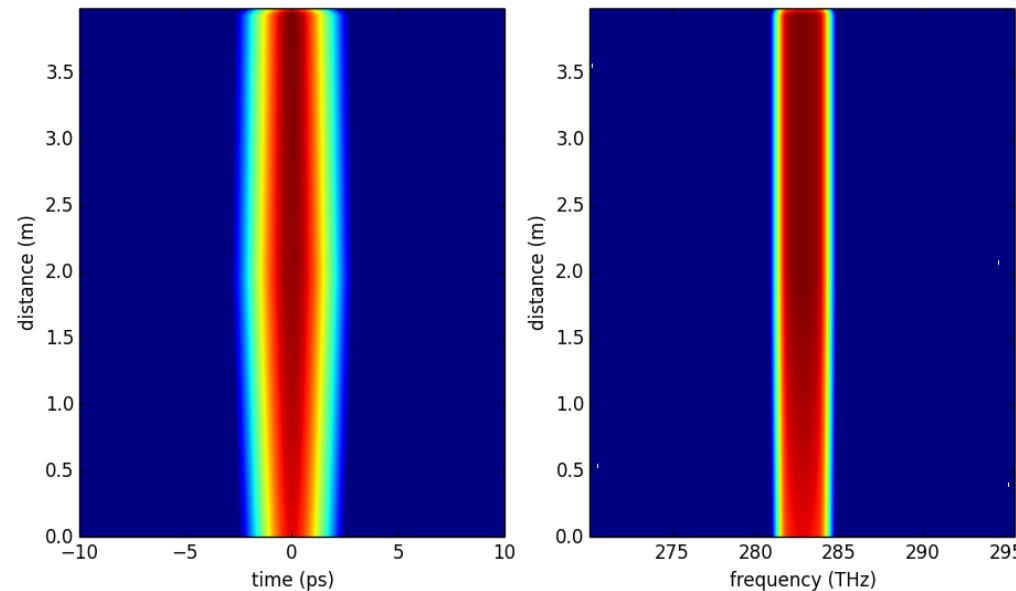
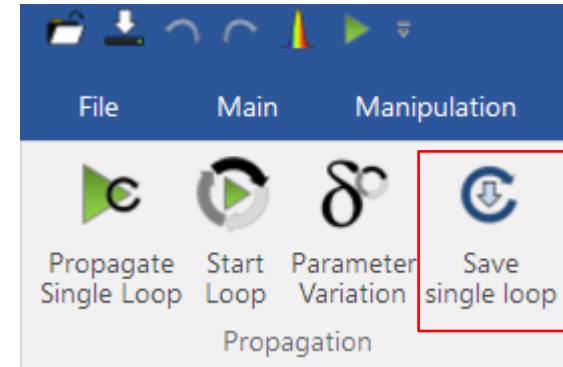


lecture 5

Multi-Element Propagation:

Example: Short Pulse Fiber Lasers

similariton: $\beta_2@DC = -0.02 \text{ ps}^2$

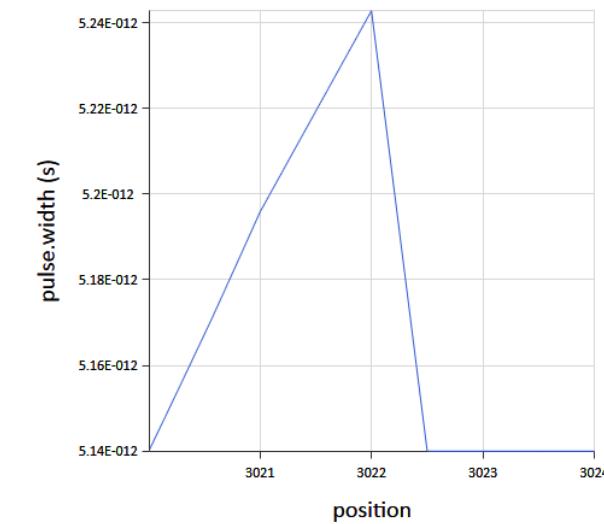
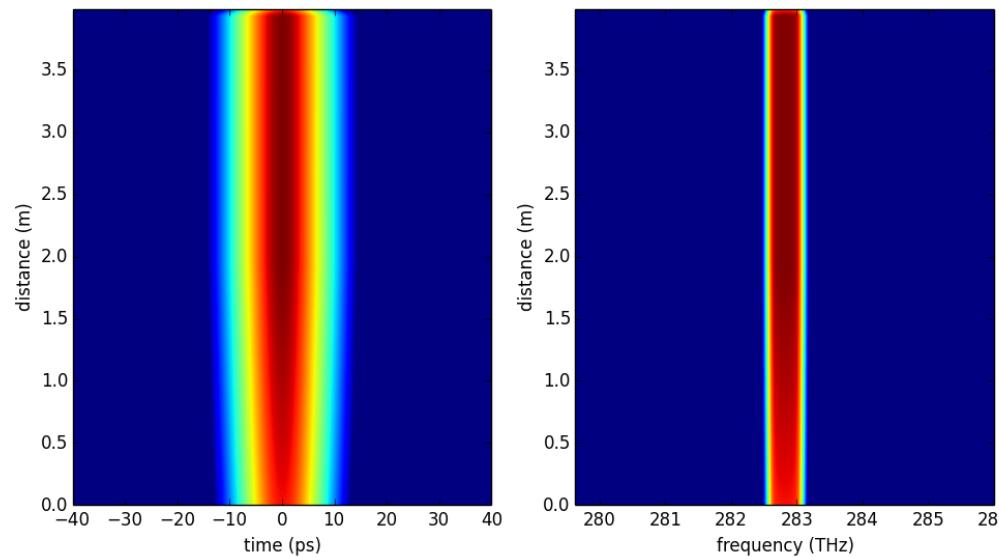
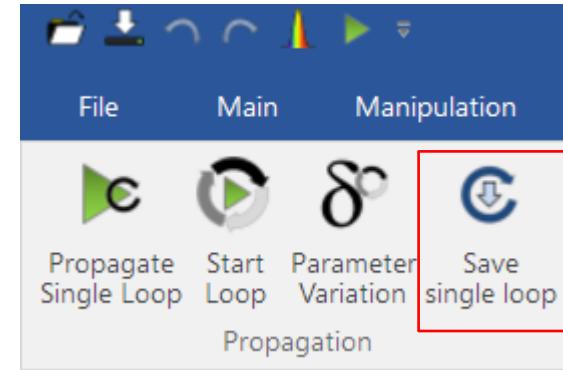


lecture 5

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:

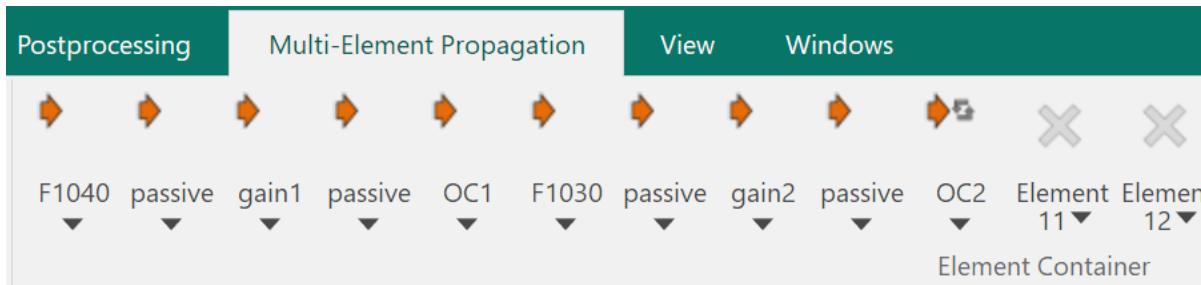
Micro Comb Generator



Multi-Element Propagation: Mamyshev Oscillator

See 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: **1040.ppf**

Manipulation

Create Pulse

change repetition rate to Hz

Create double pulse

delay of pulses (will be ps)

take phase shift into account (Mach-Zehnder equival...)

Delay Pulse (temporal shift) ps

Center Pulse

Complex Multiplication Temporal Domain

t - time in sec

helper variable h

i

Complex Multiplication Spectral Domain

wl - wavelength in m, f - frequency in Hz

helper variable h

i

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: **1030.ppf**

Manipulation

Create Pulse

change repetition rate to Hz

Create double pulse

delay of pulses (will be ps)

take phase shift into account (Mach-Zehnder equival...)

Delay Pulse (temporal shift) ps

Center Pulse

Complex Multiplication Temporal Domain

t - time in sec

helper variable h

i

Complex Multiplication Spectral Domain

wl - wavelength in m, f - frequency in Hz

helper variable h

i

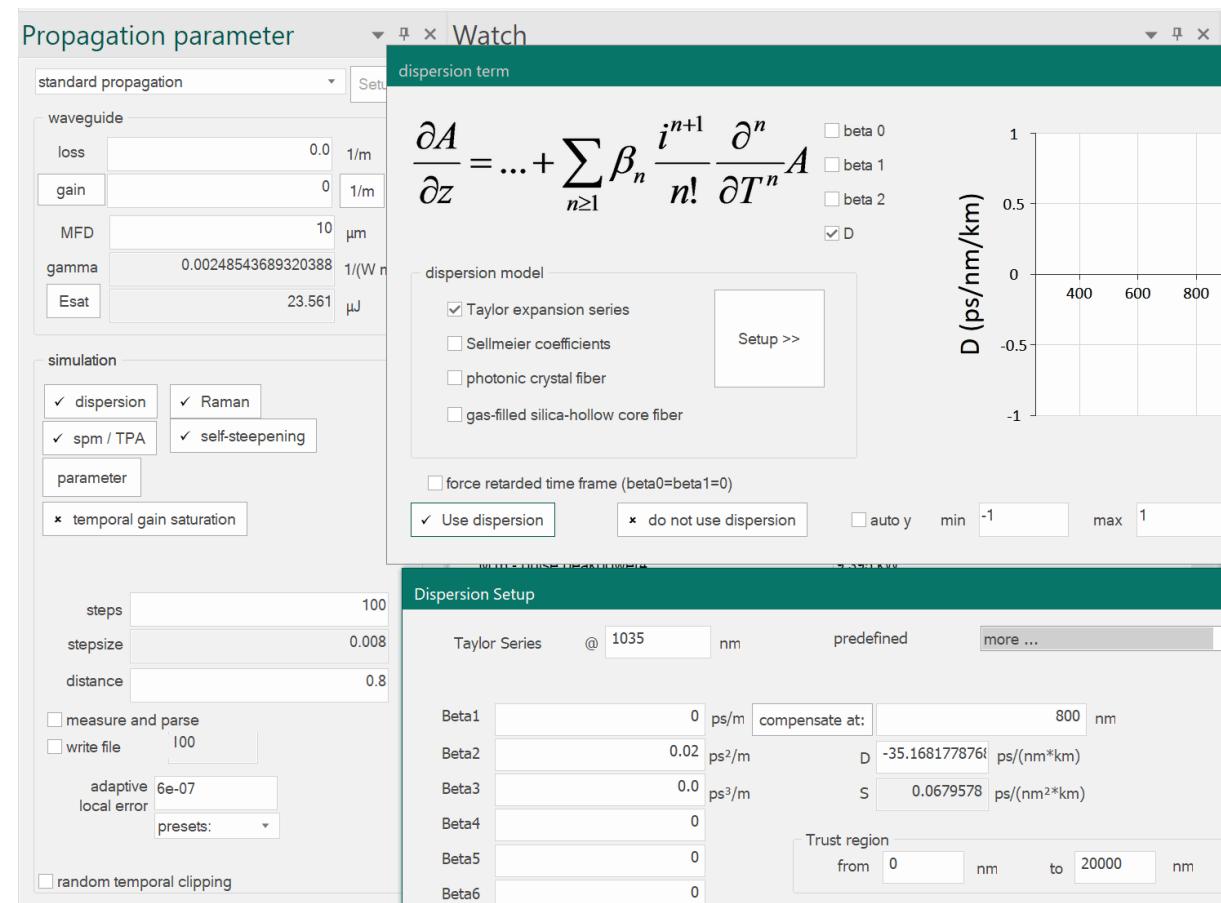
Similar, but centered at 1030 nm.

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.ppf**



Multi-Element Propagation: Mamyshev Oscillator

The output coupler is again defined as a pulse manipulation:

Pulse manipulation propagation

Saved as: OC1.ppf

Manipulation

Create Pulse
 change repetition rate to Hz
 Create double pulse
delay of pulses (will be ps)
 take phase shift into account (Mach-Zehnder equival...
 Delay Pulse (temporal shift) ps
 Center Pulse
 Complex Multiplication Temporal Domain
t - time in sec
helper variable h
 $\sqrt{0.5}$
i
 Complex Multiplication Spectral Domain
wl - wavelength in m, f - frequency in Hz
helper variable h
 0
i

50% Transmission.

Pulse manipulation propagation

Saved as: OC2.ppf

Manipulation

Create Pulse
 change repetition rate to Hz
 Create double pulse
delay of pulses (will be ps)
 take phase shift into account (Mach-Zehnder equival...
 Delay Pulse (temporal shift) ps
 Center Pulse
 Complex Multiplication Temporal Domain
t - time in sec
helper variable h
 $\sqrt{0.12}$
i
 Complex Multiplication Spectral Domain
wl - wavelength in m, f - frequency in Hz
helper variable h
 0
i

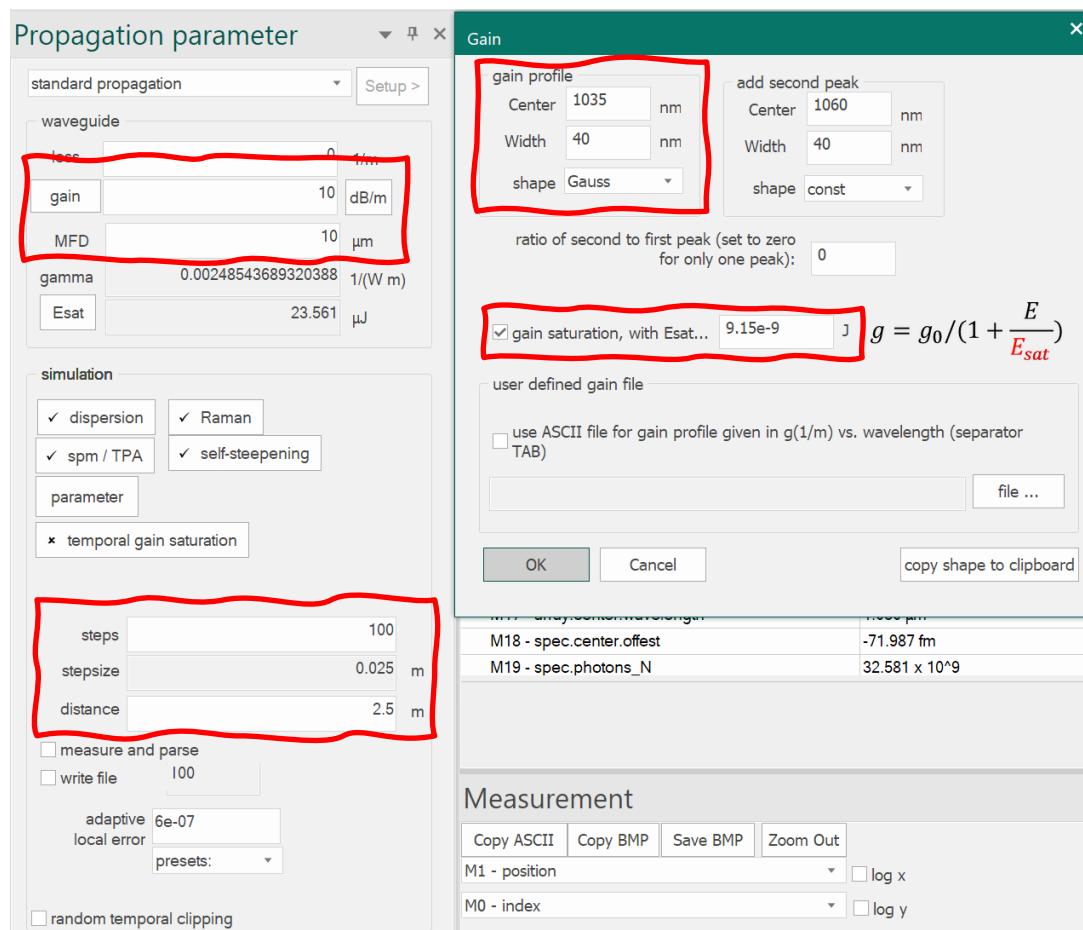
12% Transmission. (88% out-coupling)

Multi-Element Propagation: Mamyshev Oscillator

The first gain fiber is defined as a standard propagation:

Standard propagation

Saved as: gain1.ppf

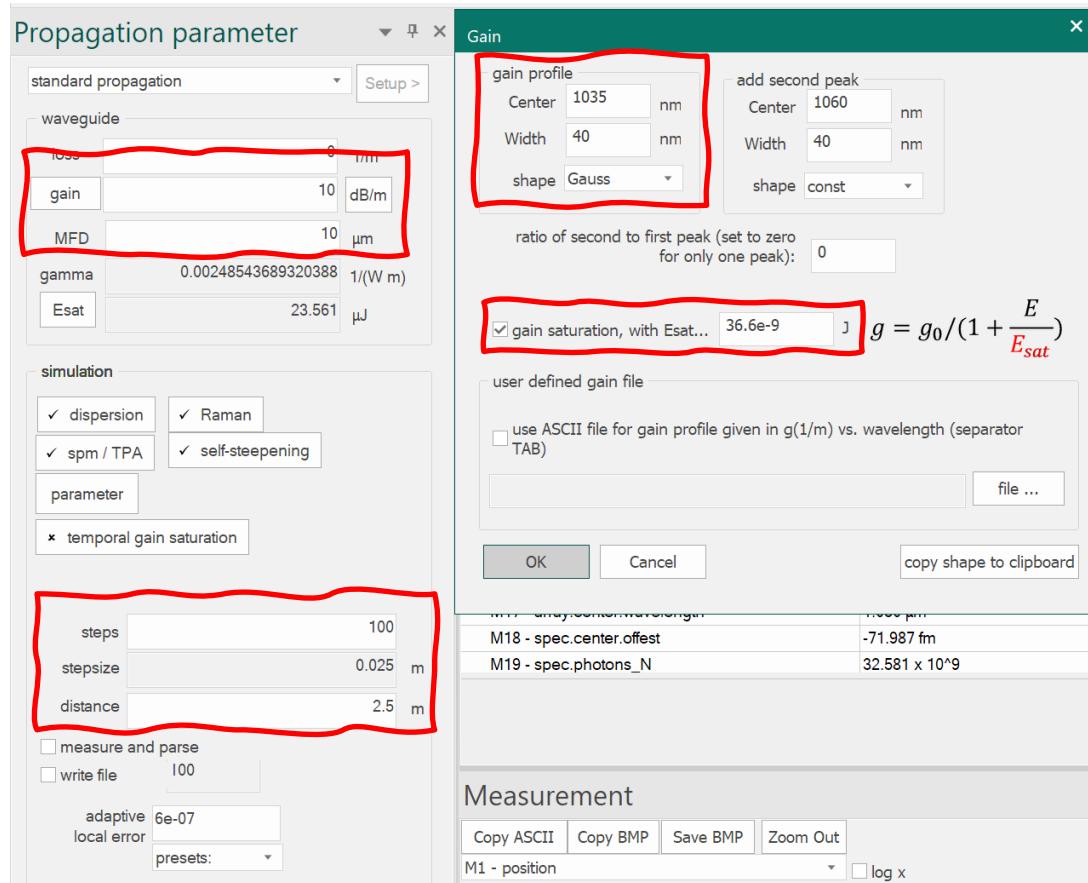


Multi-Element Propagation: Mamyshev Oscillator

The second gain fiber is defined as a standard propagation:

Standard propagation

Saved as: gain2.ppf



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: 2k (2¹¹) nm

array center wavelength: 1035 nm

half interval: 10 ps

field profile definition

Type: Gauss

FWHM: 0.1 ps

TempShift: 0 ps

phase: 0 rad

wavelength: 1035 nm

2nd order spectral phase: 0 fs²

3rd order: 0 fs³

energy: 1e-09 J

average power: 0.1 W

repetition rate: 100e6 Hz

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

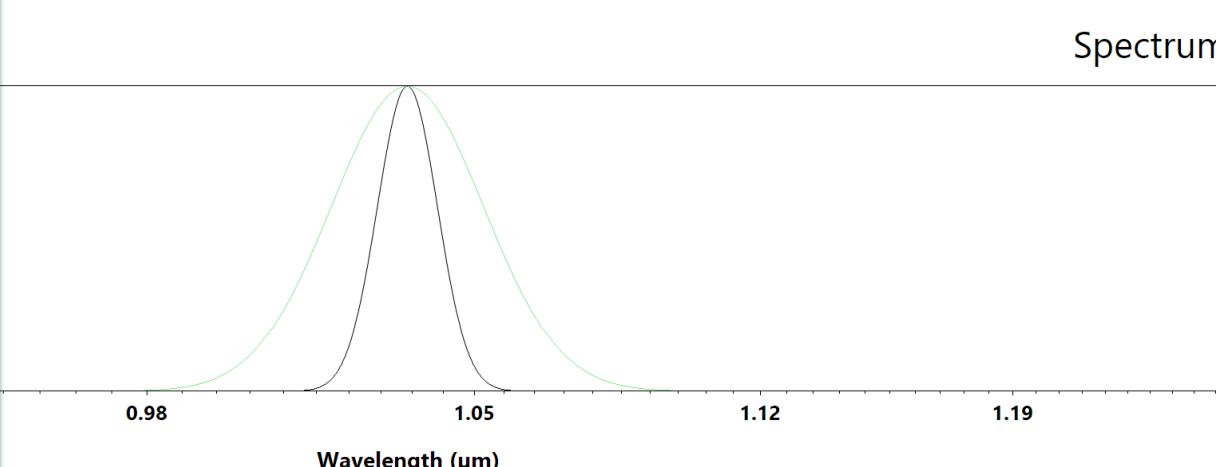
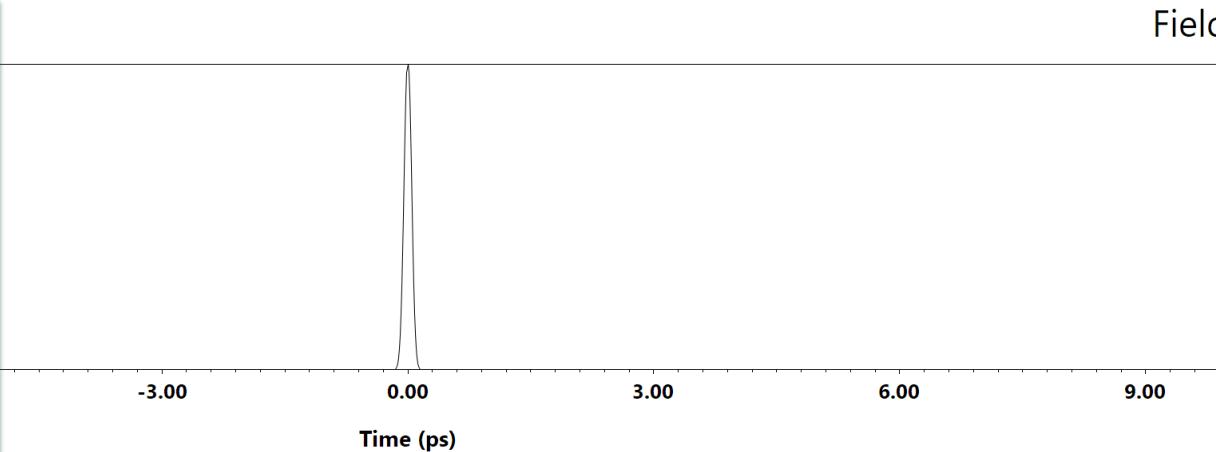
create field in data array 1

create field in data array 2

add field to data array 1

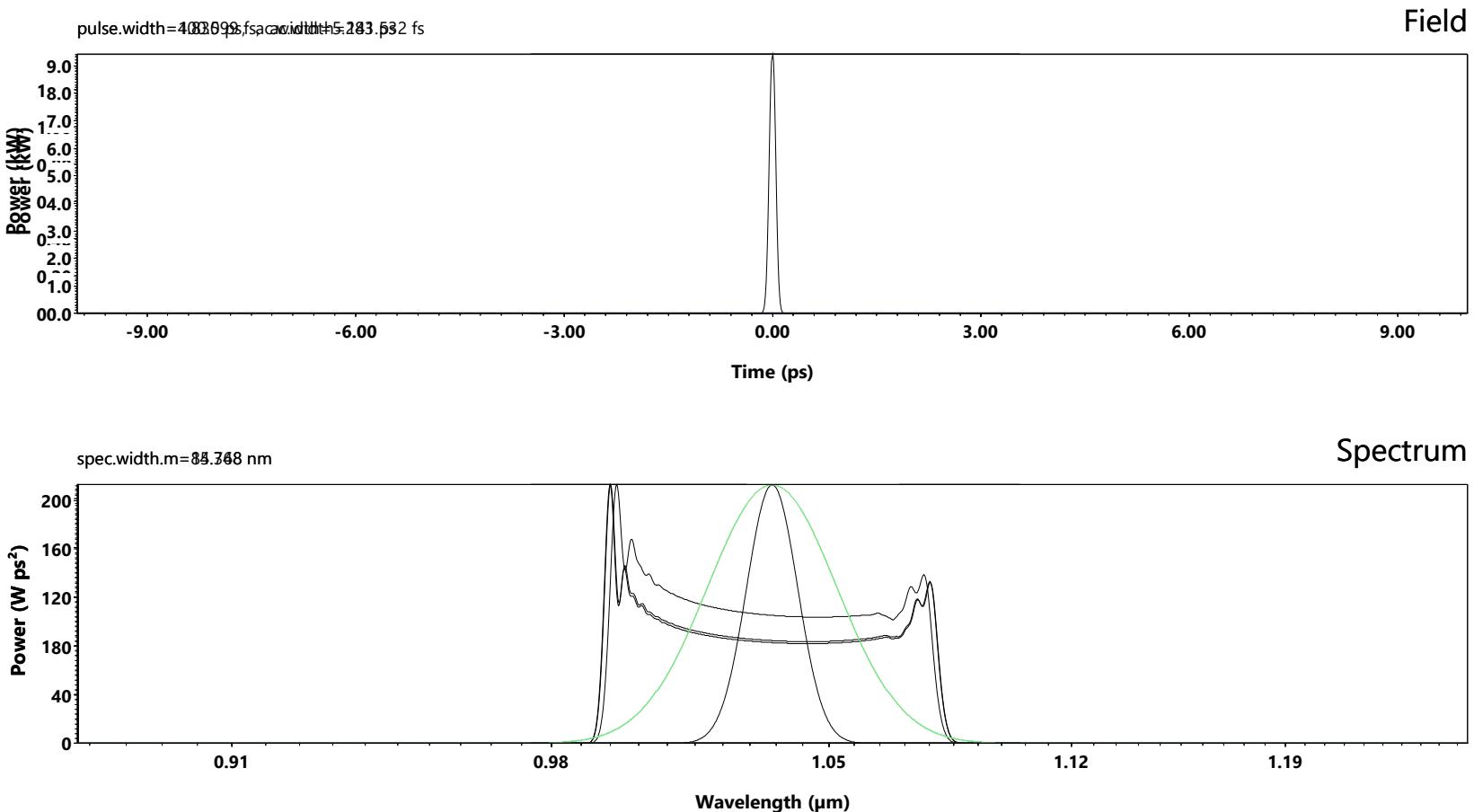
add field to data array 2

OK Apply Cancel reset



Multi-Element Propagation: Mamyshev Oscillator

After some roundtrips, we see the pulse converging. However, after many roundtrips, the pulse might destabilize.



Lecture 5

Multi-Element Propagation

Part 1:

Short Pulse Fiber Laser

Part 2:

Mamyshev Oscillator

Part 3:

Micro Comb Generator



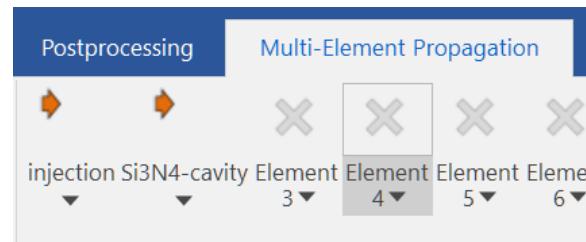
Multi-Element Propagation: Micro Comb Generator

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:

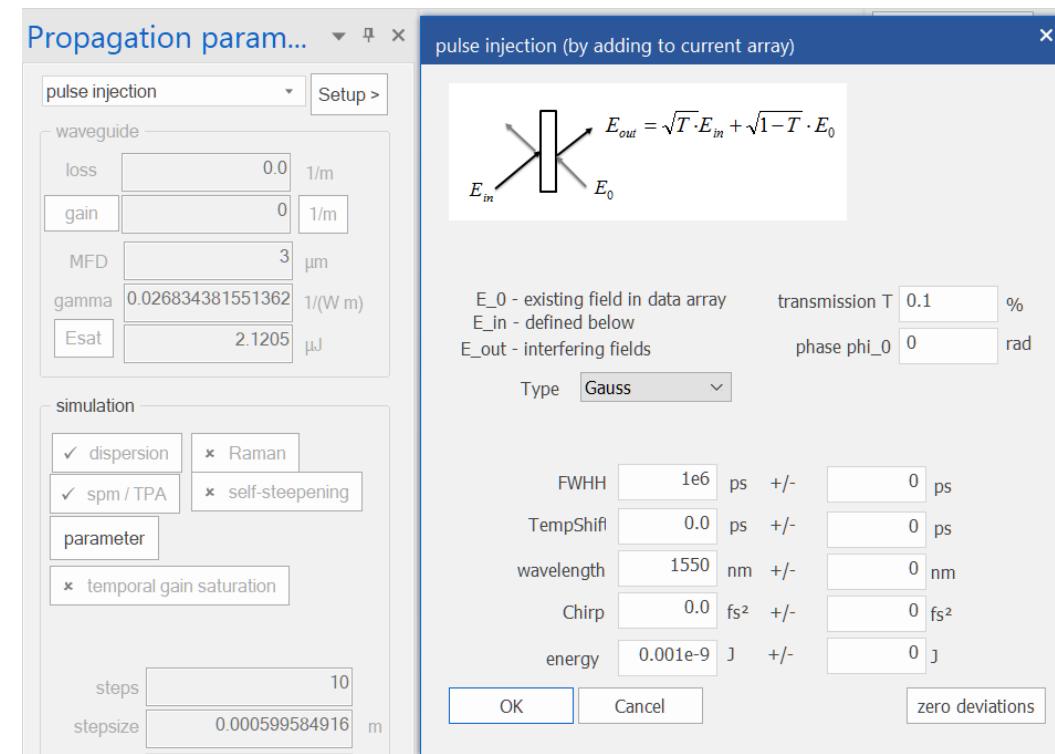


Please also note that the average propagation equation (LLE) can be used for simulating Micro Comb Generation but are intended for another tutorial.

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



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 ² /m
Beta3	0.0 ps ³ /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

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

n2	2.3e-19 m ² /V
f_R	0.15
TPA	0 m/W

TPA is experimental so far ...

saturate SPM saturation power density 1.0 GW/cm²

use SPM and TPA exclude SPM

Propagation param...

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

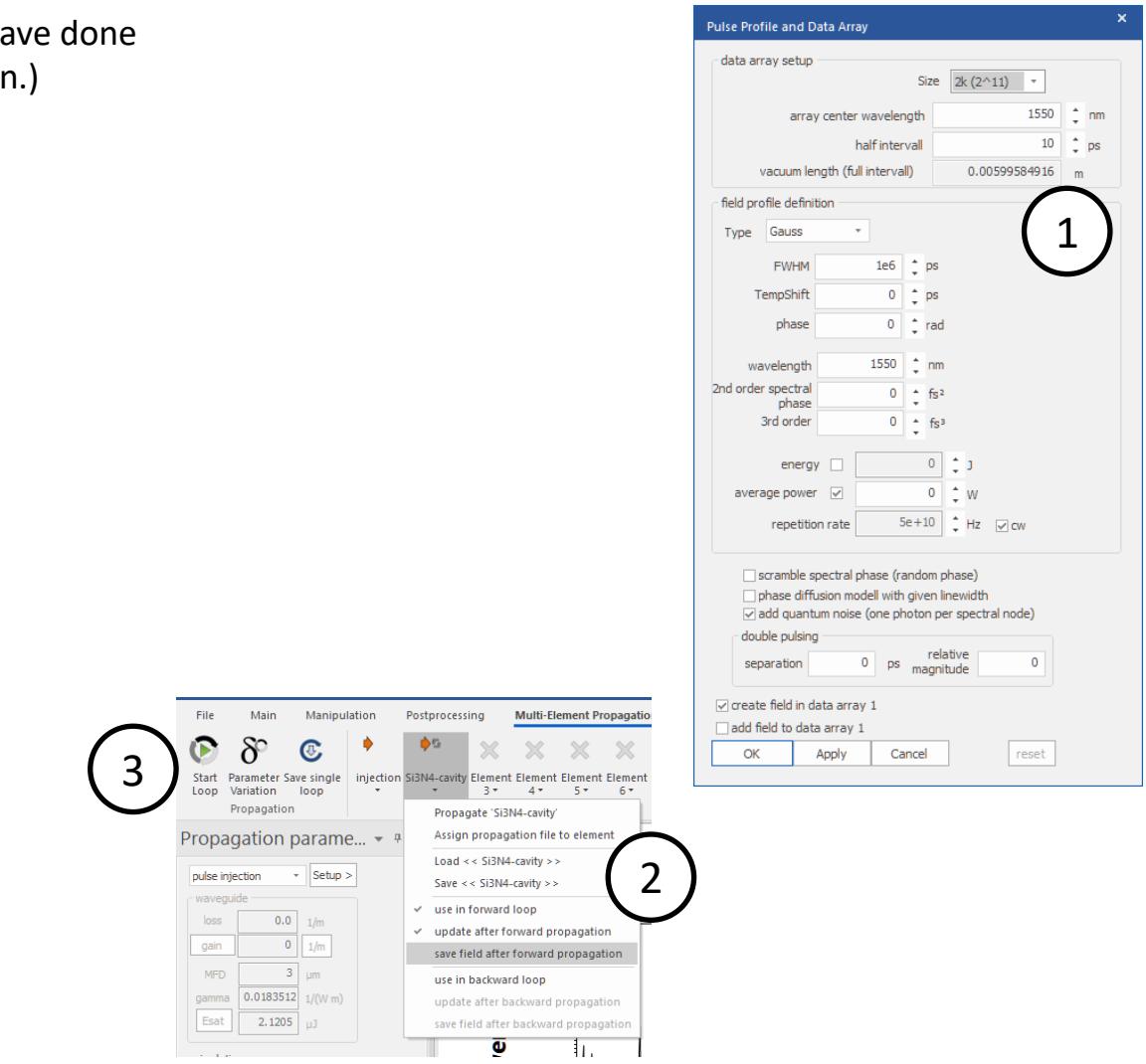
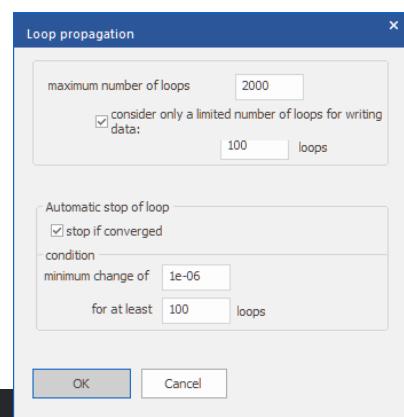
presets:

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 te 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.

