# Advanced Examples of Use of the GMMNLSE Solver

Logan G. Wright*[1], Zachary M. Ziegler†[1], Zimu Zhu‡[1], Pavel M. Lushnikov[3], and Frank W. Wise[1]

[1]School of Applied and Engineering Physics, Cornell University, Ithaca, New York 14853, USA
[2]Department of Mathematics and Statistics, University of New Mexico, Alberquerque, New Mexico 87131 , USA

August 3, 2017

**Abstract**

We go through some tutorial examples using the GMMNLSE solver. Before working through these, please consult the setup guide in the main folder, called 'README'.

## 1 Example 1: Linear propagation

In this first example, we consider linear propagation in a multimode fiber. The setting is similar to what will be considered in Example 2, where we'll add nonlinearity and observe the formation of multimode solitons.

Before doing any propagation simulations, we need to solve for the modes of the fiber, and calculate the dispersion and mode-coupling tensors. To do this, in the base folder `GMMNLSE-Solver`, run:

```
solve_for_modes_1550.m
```

This step will take awhile. It populates a folder with the 'Fibers' subfolder with the calculated spatial eigenmodes of a GRIN fiber (whose parameters are already input into this script) over a range of wavelengths around a chosen center (1550 nm here). (Although you are free to copy and edit these files as necessary for your own use, in this guide we provide separate pre-written files for each example.)

*lgw32@cornell.edu
†zmz4@cornell.edu
‡zz346@cornell.edu

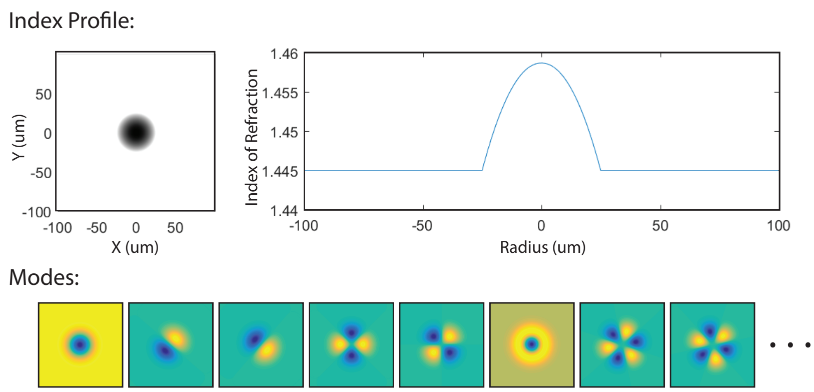


Figure 1: Cross-sectional and lateral views of the index profile for the fiber used in Examples 1 and 2, along with the first 8 modes of the fiber at 1550 nm.

```
calc_dispersion_1550.m
```

The above step calculates the dispersion of the modes. We fit a polynomial function to the measured variation of the propagation constant. Sometimes, MATLAB will give warnings about the fit obtained, especially when the number of points is small. This is often not a problem, though we recommend performing a 'sanity check' of comparing the group velocity dispersion, and third-order dispersion to the values expected for bulk fused silica at the same wavelength (around - 20 $fs^2$/mm and 145 $fs^3$/mm at 1550, for example). If some of the modes are near their cut-off wavelength, you should be very careful with the fitting procedure, and may need to calculate numerous points at a high resolution. Note also that once a mode is no longer guided, the solver essentially solves for the modes of the entire window.

```
calc_SRSK_tensors_1550.m
```

This calculates the mode-coupling tensors from the modes we calculated at the chosen center wavelength.

Once all this is done, we can start our linear propagation simulation. Before we do this, we usually need to add the root folder (which contains most of the basic functions) to MATLAB's path. Once you are in the folder `GMMNLSE-Solver\Examples\MM Propagation\` , you can do this by typing:

```
addpath('../../');}
```

```
GMMNLSE_driver_gpu_1550_linear.m
```

Note that, since we really are just considering linear propagation, it would be sufficient to just apply the dispersion operator to the initial field once. Without any nonlinear term, we can solve the equation exactly in the spectral domain

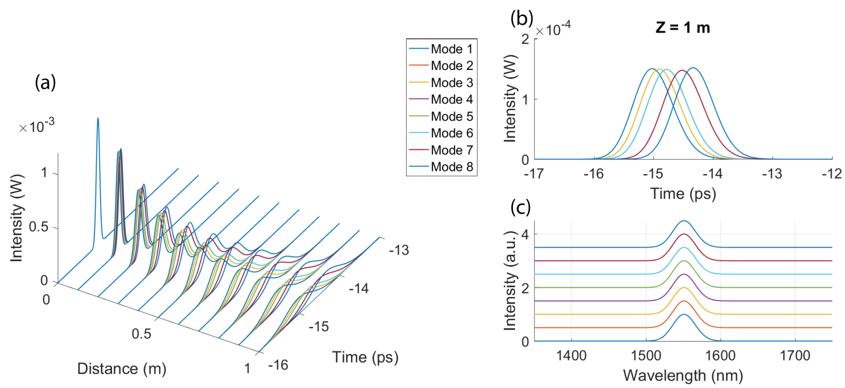


Figure 2: Linear propagation. (a) shows walk-off (modal dispersion) and temporal broadening (group-velocity dispersion) over the first 1 m. (b) and (c) show temporal and spectral profiles at 1 m.

without having to care about the pulse energy. You could verify this by repeating the simulation with a much coarser z resolution.

To plot the result:

```
plotter_1550_linear.m
```

This will generate plots at each save position along the fiber (here we chose 100 evenly distributed along the 5-m length). It's often useful to view these in succession to see a movie of the propagation. Also, plotting the temporal and spectral domain on a log scale over their entire range helps to keep track of the energy near the edges of the windows, or to look for other signs of numerical artifacts. Here we show a slightly different presentation (Fig. 2) of the results to better demonstrate the propagation physics over the initial 1 m of propagation. The above script can be reconfigured to do this by plotting onto one single figure with the plot3 function:

```
figure; hold on;
...
for zi=1:1:Zp
  for idx=1:num_modes
    plot3(zplot*ones(size(t)),t, I_time(:,idx),...
      'Color',col(idx,:))
  end
end
```

## 2 Example 2: Multimode soliton formation

In this example, we consider the formation of multimode solitons. Here we will use the same fiber, for which we already calculated the files we need in Example

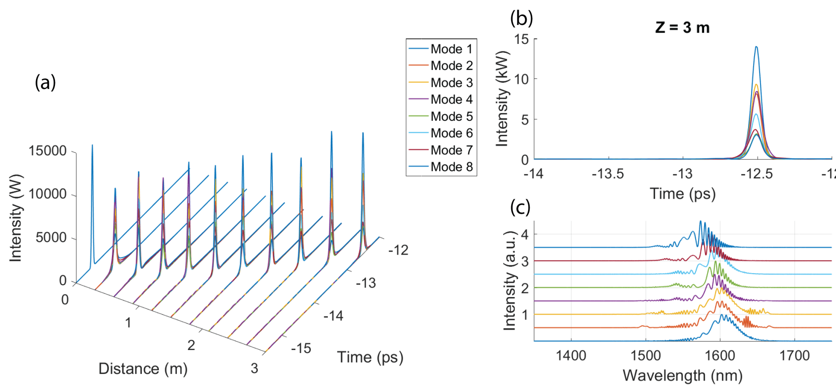


Figure 3: Nonlinear propagation. (a) shows the formation of a multimode soliton over 3 m using an initial pulse duration of 50 fs. (b) and (c) show temporal and spectral profiles at 3 m. Note the shift of the soliton from its initial temporal position (-15 ps).

1 (if you skipped that, go back and run the first 3 scripts).

First, let's look at what the whole GMMNLSE gives when we launch 6 nJ, equally distributed into 50-fs Gaussian pulses in each mode. Since we are interested in solitons, let's propagate over 15 m. If you looked at Example 1, you see that even for 5-m of linear propagation, the pulse broadens significantly due to both modal and chromatic dispersion.

After this, we also want to see what happens if the energy is the same (6 nJ), but the pulse duration is 1 ps.

Sometimes the simulations take awhile, so it's advisable to automate them in advance. In the folder `GMMNLSE-Solver\Examples\MM Propagation\` , try running the following. This runs 2 simulations, 1 as described with 50-fs, and the other with 1 ps duration. The second parameter passed is just a number to keep track of the names of the different simulations.

```
TFWHM=[0.05 1.0]; %duration in ps
for idx=1:length(TFWHM)
GMMNLSE_driver_gpu_1550_MMS(TFWHM(idx),idx);
end
```

Once these are done, you can plot the results.

```
plotter_1550MMSA.m
```

This plots at each save point. Here in Fig. 3 we show only the results using a 50 fs input pulse duration. See for yourself what happens in the 1 ps case!.

Some more interesting kinds of plots are generated by another script that visualize the spatiotemporal field (shown here in Fig. 4). Since they involve a change of basis back to the Cartesian coordinates, they can be slow to run, so I generally do not run these kinds of plots for each save point initially.

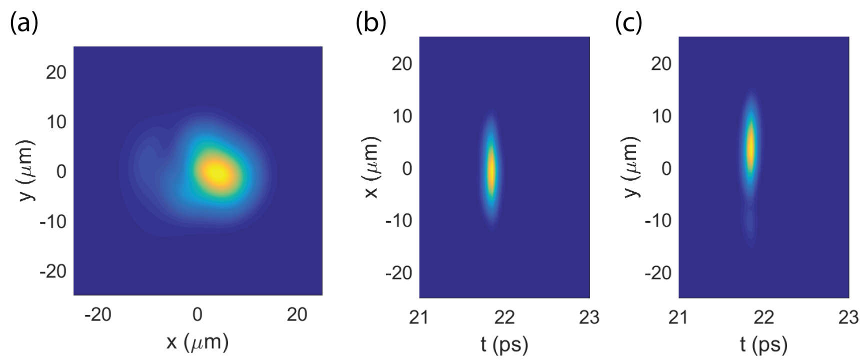


Figure 4: Nonlinear propagation. (a) final average spatial field (integrated over time) after 15 m of propagation. (b) and (c) show the field integrated over y and x respectively.

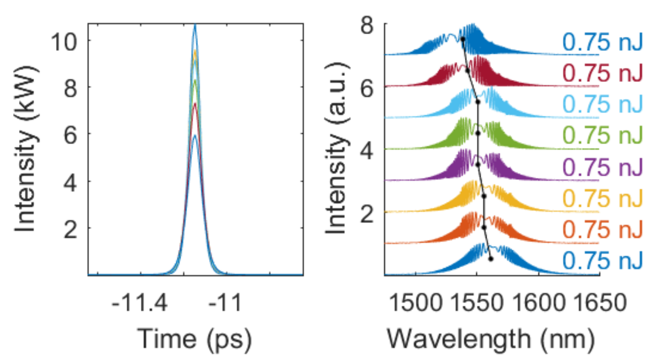


Figure 5: Nonlinear propagation over 15 m for an initial pulse duration of 50 fs and 6 nJ, but with only the Kerr SPM and XPM terms included.

```
multiplotter_1550MMSA.m
```

Now that we know what happens in the full GMMNLSE, we can examine the contribution of different effects. As promised, the first thing we can look at is what happens when only Kerr SPM and XPM terms are included. Again, we want to do it for 50 fs and 1 ps. If you examine the function we're calling, you'll see it has some lines early on that take the SR tensor we calculated and includes only the coefficients corresponding to XPM and SPM in the one that gets passed into the simulations (all the other terms are set to 0). We're also neglecting Raman and self-steepening here, by taking $f_R = 0$ and setting the `sim.sw` parameter to 0.

```
TFWHM=[0.05 1.0]; %duration in ps
for idx=1:length(TFWHM)
GMMNLSE_driver_gpu_1550_MMS_XPM(TFWHM(idx),idx);
```

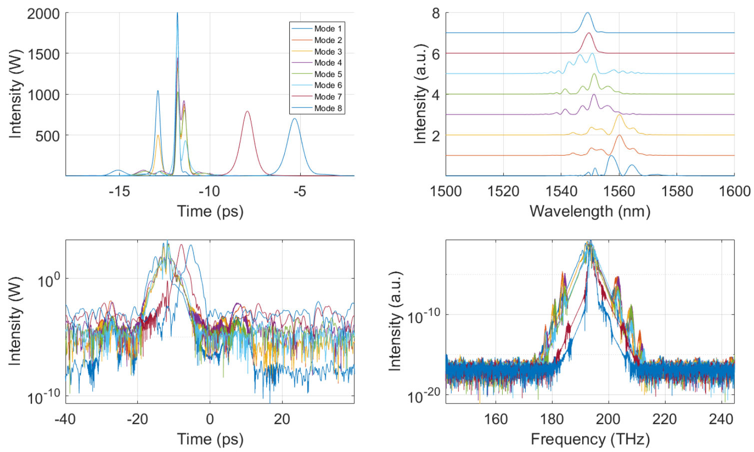


Figure 6: Nonlinear propagation after 5 m for the 1 ps case, including all Kerr terms, but without Raman and self-steepening. Note the emerging spectral sidebands in the bottom right panel.

```
end
```

At this point, we can generate the figure in the manuscript (shown here in Fig. 5).

```
plot_MMS_manuscriptfigure.m in the same folder.
```

If you are interested now, you can easily modify the scripts we just used before to plot the XPM/SPM results.

Another thing to compare to our first result would be the GMMNLSE with all Kerr terms, but without Raman and self-steepening. This means we're including all the Kerr nonlinear terms, i.e. we're solving what is called in the manuscript the MMNLSEs. For the sake of keeping the runtime small, we'll stick to 5 m for these simulations (you'll see why shortly).

```
TFWHM=[0.05 1.0];
for idx=1:length(TFWHM)
GMMNLSE_driver_gpu_1550_MMS_NSS(TFWHM(idx),idx);
end
```

Something that happens in this simulation that I don't like are sidebands in the spectrum. You can see what I mean in the bottom right panel of Fig. 6.

```
plotter_1550MMS_NSS
```

Sidebands like this can correspond to real processes, such as geometric parametric instability or resonant dispersive radiation from the breathing of solitons. But they can also arise as numerical artifacts. Since these sidebands have a much smaller shift from the center frequency than are expected for geometric

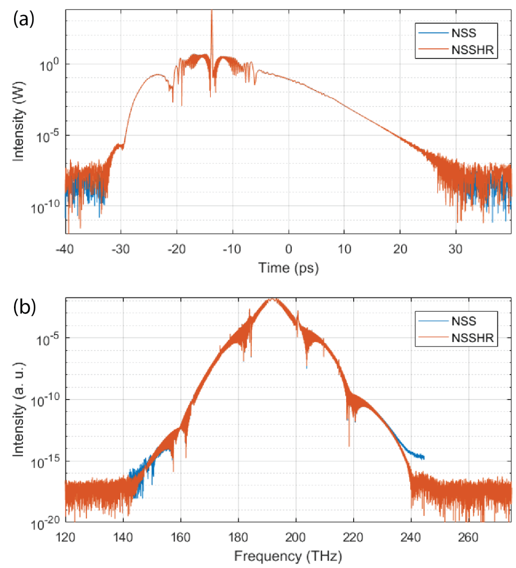


Figure 7: Comparing results of the nonlinear propagation simulations including all terms except Raman and self-steepening for low ($2^{13}$ temporal gridpoints and longitudinal step size of 25 um) and high resolutions ($2^{15}$ temporal gridpoints and longitudinal step size of 12.5 um). (a) shows the final temporal profiles and (b) the spectra of the fundamental mode in both cases.

parametric instability, or for dispersive radiation arising from spatiotemporal oscillations, my instinct is that these are not 'real' (of course, since they don't show up when self-steepening and Raman are included, they are arguably unreal regardless of whether they are an artifact or not).

This next step is optional, since it takes awhile. I increase the number of temporal gridpoints from $2^{13}$ to $2^{15}$ and the longitudinal step size from 25 to 12.5 um.

```
GMMNLSE_driver_gpu_1550_MMS_NSSHR(0.05,1);
compareresults.m
```

In Fig. 7 we plot a comparison of the results of the two cases. For the most part, the comparisons looks fine for the purposes of exploration. You can zoom in on the FIG to see how close the two results are. But in general, I'd still prefer to have theory and experiment that agree with the numerical results (they all support each other). If you have little power at the edge of the windows, and if your small step size (`deltaZ`) is much shorter than the shortest beat length, you should be reasonably confident. However, obviously here we're examining a result that is obviously not physically accurate - it is clearly different from the result containing Raman and self-steepening. As is discussed in the manuscript, the GMMNLSE and MMNLSE are fantastic for understanding propagation qualitatively. Getting quantitative agreement is possible, but should not be taken for granted.

# 3 Example 3: Generation of 1300-nm pulses by self-phase modulation with normal dispersion

## 3.1 GRIN fiber

1300-nm is an attractive wavelength for doing biomedical imaging, but well-developed rare-earth gain media do not cover this range. One approach is to use the self-phase modulation of an initially nearly transform-limited pulse. This produces spectral sidebands, the most red and blueshifted of which correspond to nearly-transform limited pulses when they are spectrally filtered (since for pure self-phase modulation, these sidelobes are the location of a phase inflection).

One way to scale the energy generated by this technique is to try it in multimode fiber, launching into the fundamental mode. We first look at this in a fiber which, like the previous example, is based on the Corning OM4 design (we just choose this because it is a readily-available but high-quality GRIN fiber). Like the last example, we consider a subset of the modes that includes the first 6, plus two higher-order radially symmetric modes. In the last example, this was somewhat arbitrary since we were looking to answer a theoretical/conceptual question. Here, we are trying to ensure accuracy of our simulations. For this purpose, the subset of modes we've chosen is good. These are the modes most likely to be excited when we initially launch the pulse into only the fundamental mode.

First, we need to calculate the modes, coupling tensors and dispersion. In the base folder `GMMNLSE-Solver`:

```
solve_for_modes
```

This populates the folder `GMMNLSE-Solver\Fibers\GRIN_1030` . This step will take awhile, producing quite a few plots. It's worthwhile to examine the modes in the folder, to see what kind of outputs the modesolver is producing.

```
calc_dispersion
```

This gets the dispersion parameters you need, and writes them into a files betas.mat in the folder containing the result of the mode information.

```
calc_SRSK_tensors
```

We are assuming, like in the previous example, that we have a short fiber and are initially exciting a linear polarization. Under this assumption, we take $S_K = S_R$.

Now we are ready to run the first simulations. Open up the folder
`GMMNLSE-Solver\Examples\MM Propagation\` . If you have not already, run the following:

```
addpath('../../');
```

```
GMMNLSE_driver_gpu_1030_SPMGRINA;
```

This launches a 200-fs duration, 600-nJ energy pulse into the fundamental mode. We assume some weak excitation of all the other modes, which would often be the case in an experiment.

```
plotter_SPMGRIN1030A;
```

This plots every save point as in previous examples. Here, in addition to watching the evolution and keeping an eye out for numerical artifacts, we are looking for the point where we'd want to cut the fiber (in an experiment, we'd probably tune the pulse energy for a fixed short fiber length to get a similar effect). For this one, the best point looks to be around 3.6 cm to me, shown at the end of this document in Fig. 9 for direct comparison to step-index fiber.

```
multiplotter_SPMGRIN1030A
```

This plots the field before and after spectral filtering of the 1300-nm sidelobe. It also plots the spatiotemporal field before filtering. Note that, because we are using the modes at 1030-nm to represent the field in spacetime, this representation is probably only approximately correct over this bandwidth. Since these are low-order modes that are guided over the entire range, we expect the approximation is decent. But it's definitely not perfect!

Given the high nonlinearity and peak power, an important thing to check is whether our assumption regarding choice of modes is valid. To this end, we calculate many modes, and run simulations as before with a larger set. Ultimately, we find a pretty typical answer that our choice of a limited subset of modes is reasonably correct concerning some of the major features, but in the details is not accurate. For lower-power simulations, assumptions about energy coupling to initially unoccupied modes is usually better. However, as discussed in the manuscript it is important to keep the limitations of the equation and the choice of modes in mind always. It's not uncommon for four-wave mixing or dispersive waves to cause energy to end up in modes that are excluded from a subset chosen for GMMNLSE simulations.

Go back to the root folder, and run these (which will take awhile):

```
solve_for_modes_many;
calc_dispersion_many;
calc_SRSK_tensors_many;
```

You can now go back to the MM Examples folder and run these:

```
GMMNLSE_driver_gpu_1030_SPMGRINAHMM;
plotter_SPMGRIN1030AHMM;
```

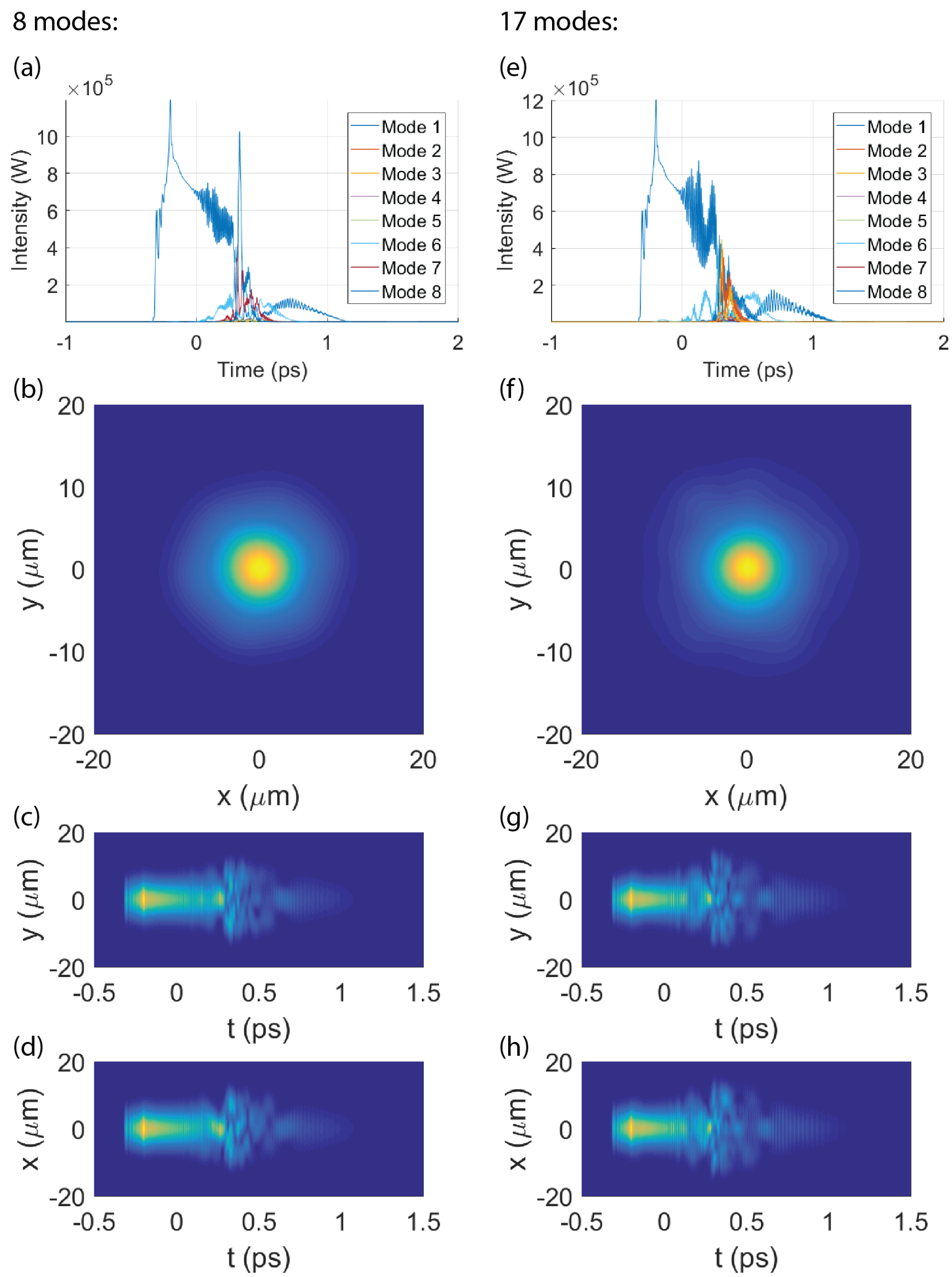


Figure 8: Comparison of results for generation of 1300-nm pulses in GRIN via SPM for the cases of including 8 modes (a)-(d) and 17 modes (e)-(h) in the simulation. (a) and (e) show the resulting fully mode-resolved temporal profiles from each respective simulation, but with only the first 8 modes indicated in the legends. (b) and (f) show average (integrated over time) spatial fields for the two cases, while (c), (d), (g) and (h) show the field integrated over y and x for each case. For mode-resolved spectra, feel free to try the simulations out yourself!

## 3.2 Step-index fiber

One question we can ask is whether a step-index fiber might serve better for this purpose. The greater walk-off of the modes means that coupling to HOM may be less efficient. The fundamental mode is larger compared to a GRIN fiber of the same radius, so we can either support more energy or we can choose a fiber with fewer modes (and this again may help us if our goal is to keep the beam quality high by ensuring the 1300-nm pulse is exclusively in the fundamental mode.

If you try to scale much further with mode area, you will quickly find that the required peak powers exceed the critical power for self-focusing. Therefore, we'll consider a step-index fiber whose fundamental mode is about the same size as the GRIN fiber. In this case, the number of guided modes is small enough, we can model all of them easily in the GMMNLSE.

Go into the root folder and run:

```
solve_for_modes_stepindex2;
calc_dispersion_stepindex2;
calc_SRSK_tensors_stepindex2;
```

Once this is done, we can compare some properties of the two fibers we have solved for at 1030-nm. In the same folder, run the following:

```
%Comparing fiber properties with GRIN fiber
num_modesS=10;
prefix = '\Fibers\STEP_1030S';
load([pwd prefix '\S_tensors_' num2str(num_modesS) 'modes.mat']); % in m^-2
load([pwd prefix '\betas.mat']); % in fs^n/mm

SRstep=SR;
betasstep=betas;

num_modesG=8;
prefix = '\Fibers\GRIN_1030';
load([pwd prefix '\S_tensors_' num2str(num_modesG) 'modes.mat']); % in m^-2
load([pwd prefix '\betas.mat']); % in fs^n/mm

SRgrin=SR;
betasgrin=betas;

%First plot the modal dispersion parameters
f1=figure('Position',[1 1 400 200])
hold on
plot(1:num_modesS,betasstep(2,:),'ko')
plot(1:num_modesG,betasgrin(2,:),'ro')
hold off
axis tight
legend('Step-index', 'GRIN')
```

```
ylabel('\delta\beta^{(p)}_1 (fs/mm)')
xlabel('p (mode index)')
box on

%Retrieve the effective areas, convert to um^2 units
AeffG=zeros(num_modesG,1);
AeffS=zeros(num_modesS,1);
for idx=1:num_modesG
AeffG(idx)=1E12/SRgrin(idx,idx,idx,idx);
end
for idx=1:num_modesS
AeffS(idx)=1E12/SRstep(idx,idx,idx,idx);
end
f2=figure('Position',[1 1 400 200])
hold on
plot(1:num_modesS,AeffS,'ko')
plot(1:num_modesG,AeffG,'ro')
hold off
legend('Step-index', 'GRIN')
ylabel('A_{eff} (\mum^2)')
xlabel('p (mode index)')
box on
```

To save you the time, the outputs are shown in Fig. 9.

Now, we can run a simulation in this fiber. Go into the `Examples\MM Propagation` folder, and run the following:

```
prefix = '../../Fibers/STEP_1030S';
GMMNLSE_driver_gpu_1030_SPMSTEPSA;
plotter_SPMSTEP1030SA;
multiplotter_SPMSTEP1030SA;
```

Finally, you can plot the figure from the manuscript (shown in Fig. 10):

```
plot_SPMGRINSTEP_manuscriptfigure
```

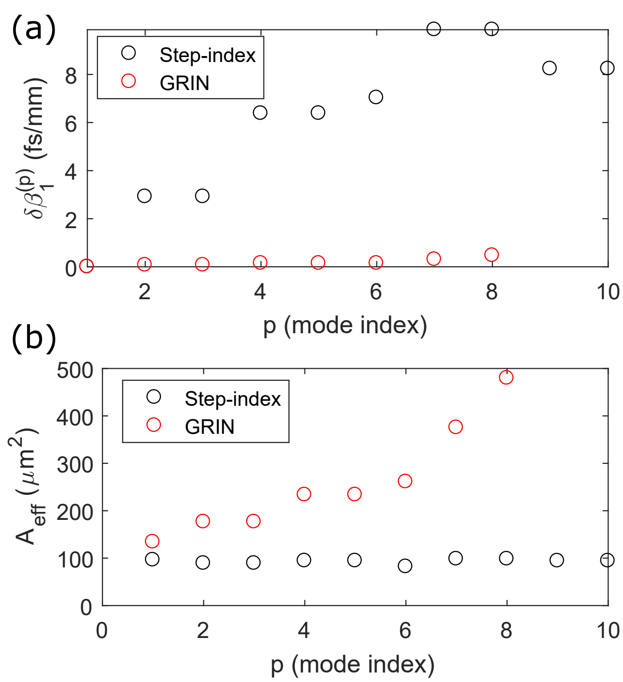


Figure 9: Comparison of the (a) group velocity relative to the fundamental mode, and (b) $A_{eff}$ of modes in a particular GRIN and step-index fiber. This example illustrates how to use the mode solver to analyze different properties of fiber designs.

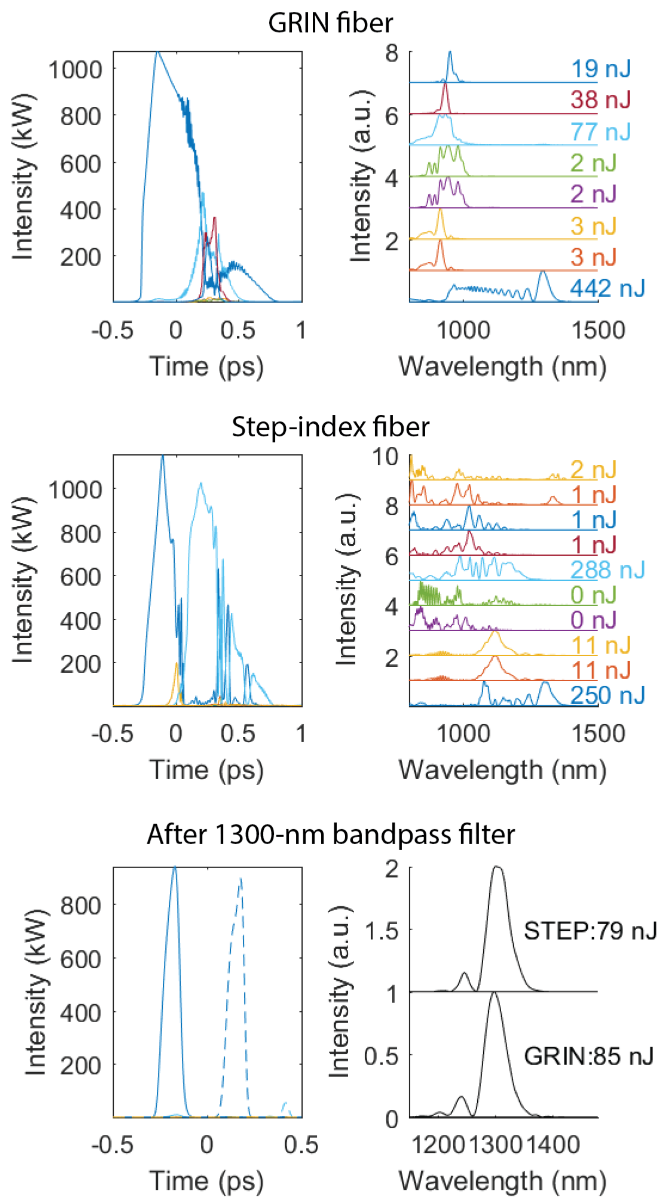


Figure 10: Comparison of results for generation of 1300-nm pulses in GRIN and step-index fiber by self-phase modulation.