Solver Guide for the MATLAB bulk-solid pulse propagation

Yi-Hao Chen
Applied and Engineering Physics, Cornell University

January 25, 2025


CORNELL UNIVERSITY
FOUNDED A.D. 1865

Contents


CORNELL UNIVERSITY
FOUNDED A.D. 1865

Chapter 1

Overview

1.1 Mathematical background

This package aims to solve the 3D unidirectional pulse propagation equation (3D-UPPE):

$$\begin{aligned}\partial_z\vec{\mathbb{A}}(\vec{k}_\perp,z,\Omega) = i\left[K_z-\left(\beta_{(0)}+\beta_{(1)}\Omega\right)\right]\vec{\mathbb{A}}(\vec{k}_\perp,z,\Omega)\\ +i\frac{1}{2k_\omega}\left(1+\frac{\left|\vec{k}_\perp\right|^2}{2k_\omega^2}\right)\mathfrak{F}_{k_\perp}\left[k_W^2(\vec{r}_\perp,\omega)\vec{\mathbb{A}}(\vec{r}_\perp,z,\Omega)\right]\\ +i\frac{\omega^2}{2k_\omega c^2}\left(1+\frac{\left|\vec{k}_\perp\right|^2}{2k_\omega^2}\right)\frac{\vec{P}(\vec{k}_\perp,z,\Omega)}{\epsilon_0},\end{aligned} \tag{1.1}$$

where $\vec{\mathbb{A}}$ is the field envelope (in $\sqrt{\mathrm{W/m^2}}$), with $\vec{\mathbb{A}}(\vec{k}_\perp)=\mathfrak{F}_{k_\perp}\left[\vec{\mathbb{A}}(\vec{r}_\perp)\right]=\mathfrak{F}_{k_x}\left[\mathfrak{F}_{k_y}\left[\vec{\mathbb{A}}(\vec{r}_\perp)\right]\right]$ representing the spatial frequency component after the spatial Fourier transform $[\mathfrak{F}_{k_x}[A](k_x)=C_{\mathfrak{F}_{k_x}}\int_{-\infty}^{\infty}A(x)\mathrm{e}^{-ik_xx}\,\mathrm{d}x$ and $\vec{r}_\perp=(x,y)]$, and $\vec{\mathbb{A}}(\Omega)=\mathfrak{F}\left[\vec{\mathbb{A}}(T)\right]$ representing the spectral component after spectral Fourier transform. Note that the spectral Fourier transform follow a different convention from that in mathematics due to following $(k_zz-\omega t)$. Due to the second-order spatial derivative in $\vec{r}_\perp$, the convention of spatial Fourier transform $\mathfrak{F}_{k_\perp}$ does not affect the result. To be consistent with K_z, we choose the convention of spatial Fourier transform such that it follows the same convention as that in mathematics, opposite to the spectral Fourier transform. The spectral Fourier transform is applied with respect to $\Omega=\omega-\omega_0$, where ω_0 is the center frequency of the numerical frequency window. $K_z(\vec{k}_\perp,\Omega)=\sqrt{k_\omega^2(\omega)-\left|\vec{k}_\perp\right|^2}$ represents both the (spectral) dispersion (from k_ω), and the (spatial) diffraction [from $\vec{k}_\perp=(k_x,k_y)$] terms. $k_W^2(\vec{r}_\perp,\omega)$ determines the waveguide effect; it is zero if the medium is homogeneous. They are determined by splitting the material's $k^2(\vec{r}_\perp,\omega)=\omega^2\mu_0\epsilon_0\epsilon_r(\vec{r}_\perp,\omega)$ into its spatially-independent part $k_\omega^2(\omega)$ and spatially-dependent part $k_W^2(\vec{r}_\perp,\omega)$, where $k_\omega^2(\omega)$ dominates the frequency variation in $k^2(\vec{r}_\perp,\omega)$:

$$k^2(\vec{r}_\perp,\omega)=k_\omega^2(\omega)+k_W^2(\vec{r}_\perp,\omega). \tag{1.2}$$

Because the dispersion and diffraction operators $\hat{D}_{\text{is}} + \hat{D}_{\text{if}} \sim K_z$ are not commutable with the waveguide operator $\hat{W} \sim k_W^2(\vec{r}_\perp, \omega)$, this separation [Eq. (1.2)] reduces the error resulting from incommutableness due to the Baker–Campbell–Hausdorff formula [1]. $\beta_{(0)}$ and $\beta_{(1)}$ are to reduce the propagating global phase increment to facilitate simulations in which $\beta_{(1)}$ is the inverse group velocity of the moving reference frame that introduces the delayed time $T = t - \beta_{(1)}z$. $\vec{P}(\vec{r}_\perp, z, \Omega) = \mathfrak{F}\left[\vec{P}(\vec{r}_\perp, z, T)\right]$ is the nonlinear term that follows

$$P^i(\vec{r}_\perp, z, \Omega) = \frac{2\epsilon_0 n(\omega) n_2}{3}\mathfrak{F}\Bigg[\sum_{j=x,y}\Bigg\{(1-f_R)\left[\left(\mathbb{A}^j\right)^2\left(\mathbb{A}^i\right)^* + 2\left|\mathbb{A}^j\right|^2\mathbb{A}^i\right] + f_R\Bigg\{f_a\left[3\mathbb{A}^i(T)\int_0^\infty h_a(\tau)\left|\mathbb{A}^j(T-\tau)\right|^2 \mathrm{d}\tau\right] + f_b\left[\mathbb{A}^j(T)\int_0^\infty \frac{3}{2}h_b(\tau)\left(\mathbb{A}^i(\mathbb{A}^j)^* + (\mathbb{A}^i)^*\mathbb{A}^j\right)(T-\tau)\,\mathrm{d}\tau\right]\Bigg\}\Bigg\}\Bigg], \tag{1.3}$$

(for $i = x, y$) which becomes

$$\begin{aligned} P(\vec{r}_\perp, z, \Omega) &= 2\epsilon_0 n(\omega) n_2 \mathfrak{F}\Big[(1-f_R)\left|\mathbb{A}\right|^2\mathbb{A} \\ &\qquad + f_R\mathbb{A}(T)\int_0^\infty (f_a h_a + f_b h_b)(\tau)\left|\mathbb{A}(T-\tau)\right|^2 \mathrm{d}\tau\Big] \\ &= 2\epsilon_0 n(\omega) n_2 \mathfrak{F}\left[(1-f_R)\left|\mathbb{A}\right|^2\mathbb{A} + f_R\left[[f_a h_a + f_b h_b] * \left|\mathbb{A}\right|^2\right]\mathbb{A}\right] \end{aligned} \tag{1.4}$$

for scalar (singly-polarized) fields. n_2 is the material nonlinear refractive index (in m^2/W). f_R is the Raman fraction representing the contribution of the Raman response of all nonlinearities where f_a and f_b are Raman fractions of the total Raman response for isotropic and anisotropic Raman responses, respectively ($f_a + f_b = 1$); $h_a(t)$ and $h_b(t)$ are isotropic and anisotropic Raman response functions [2, 3].

1.2 High-level understanding of this package

This package is designed to solve for nonlinear pulse propagation in bulk space (*e.g.*, crystals or free space). It's originally designed to solve for multimode fibers, but it can also be used for general bulk medium with a customized index profile. In addition to a full field based on 2D $x - y$ dimensions, a radially-symmetric scheme is also implemented, which relies on the Hankel transform. This reduces the system by one dimension, significantly releasing the memory constraint of the 3D (x,y,t/ω) field due to the use of a radially-symmetric (r,t/ω) field. Not only scalar but also polarized fields can be simulated. The package exhibits an adaptive control of the step size. In addition to CPU, highly parallelized cuda computation with a Nvidia GPU is implemented. The package uses "RK4IP" (Runge-Kutta in the interaction picture) [4, 5].

The fastest way to learn how to use this code is to start with the example codes in the package.

Chapter 2

Before I go deeply into details

2.1 Introduction

This document describes how to use the UPPE3D_propagate() MATLAB function.

Below is how to call this function in general.

```
prop_output = UPPE3D_propagate(fiber, ...
                               initial_condition, ...
                               sim)
```

prop_output
: It contains the information of the output field after propagating through the medium, such as the field amplitudes and the positions of each saved field, etc.

fiber
: It contains the information of the fiber, such as the index profile.

initial_condition
: It contains the information of the input.

sim
: It contains a multitude of information about the simulation, such as the algorithm to use and the center wavelength, etc.


CORNELL UNIVERSITY
FOUNDED A.D. 1865

Chapter 3

Input arguments

Below, I use N_t as the number of time/frequency sampling points, N_x and N_y as the number of spatial sampling points in x and y dimensions, respectively. N_r is the number of spatial (radial) sampling points if the simulation employs its radially-symmetric scheme. N_p is the number of polarizations. N_z is the number of saved fields after the propagation. Overall, the dimension of simulation arrays follows $N_t \times N_x \times N_y \times N_z \times N_p$ in the xy scheme or $N_t \times N_r \times N_z \times N_p$ in the radially-symmetric scheme.

I recommend to use the information below as a reference guide if you're confused. Start with an example script is always better than reading this first.

Some parameters are required only when you enable some settings. Below I labeled in blue the parameters required all the time.

3.1 fiber

fiber

This specifies the fiber type. It is used only with the index-creating function "calc_index_profile()," so in general, this parameter is not required depending on simulations.

n

This is the index profile of the medium. Its size is $N_t \times N_x \times N_y \times 1 \times N_p$ or $N_t \times N_r \times 1 \times N_p$ in the radially-symmetric scheme.

It can be used with the "calc_index_profile_i()" (i="xy," or "r" for the radially-symmetric scheme) function to generate the index profile of fibers. For bulk crystals, just use "repmat()" to create a homogeneous index profile:

```
n = repmat(n_f,1,Nx,Ny); % n_f is the (Nf,1) frequency-dependent refractive index
```

Below is the comment of "calc_index_profile_xy():"

```
%CALC_INDEX_PROFILE_XY It calculates the 2D index profile of the fiber.
%
%   fiber: a structure with
%
%       Below you can use the fiber in the repository (in "fiber_collections.m")
%       or set all parameters yourself
%       (1) Set it yourself
%           fiber_type - a string with either 'step', 'Nufern', or 'GRIN'
%           material - fiber material; usually 'silica'
%           diameter - fiber core diameter (um)
```

```
%           NA - numerical aperture
%           extra_params - extra parameters for different fiber types
%
%       (2) Use fiber repository
%           fiber - the name of the fiber
%                   Currently I have '1060XP',
%                                    'YB1200-4_125',
%                                    'YB1200-6_125DC', 'YB1200-6_125DC-PM',
%                                    'YB1200-10_125DC', 'YB1200-10_125DC-PM',
%                                    'YB1200-20_400DC',
%                                    'YB1200-25_250DC', 'YB1200-25_250DC-PM',
%                                    'ER30-4_125', 'ER110-4_125',
%                                    'ER16-8_125', 'ER80-8_125',
%                                    'M5-980-125', 'M12-980-125',
%                                    'OM1','OM2','OM3','OM4',
%                                    'FUD-7005',
%                                    'PLMA-YDF-30_400-VIII'.
%
% ------------------------------------------------------------------------
% (1) step fiber:
%     extra_params = [];
% (2) GRIN fiber:
%     extra_params.alpha = 2.08; % Shape parameter
% ------------------------------------------------------------------------
%
%   Nx - number of spatial grid points
%   spatial_window - full spatial window size (um) (usually set to 100 um)
%   f - (Nt,1); frequency points (from small to large) (THz)
```

The operation is similar for the radially-symmetric "calc_index_profile_r():"

```
function index_profile = calc_index_profile_r(fiber,r,f)
%CALC_INDEX_PROFILE_R It calculates the 2D radially-symmetric index profile
% of the fiber. Only 1D radial information is generated.
%
......
%
%   r - (1,Nr); radial sampling positions (m)
%   f - (Nt,1); frequency points (from small to large) (THz)
```

n2

It's the nonlinear coefficient of the fiber. By default, UPPE3D_propagate() uses $2.3 \times 10^{-20}\,\mathrm{m}^2/\mathrm{W}$ assuming we use a silica fiber around 1 µm if fiber.n2 is left empty.

L0

This is the fiber length. The unit is meter.

material

This specifies the fiber material. It's 'silica' by default. It's used only for specifying which Raman model to use. It's either 'silica', 'chalcogenide', or 'ZBLAN'.

3.2 initial_condition

dt

This is the time sampling step $\triangle t$ with a unit of ps.

For the xy scheme,

dx

This is the spatial sampling step $\triangle x$ with a unit of m.

dy

This is the spatial sampling step $\triangle y$ with a unit of m.

field

This is the temporal amplitude of the input field ($\mathbb{A}(t)$ with the unit $\sqrt{\mathrm{W/m^2}}$). Its size is $N_t \times N_x \times N_y \times 1 \times N_p$.

If its size is $N_t \times N_x \times N_y \times N_z \times N_p$, only the last N_z is taken as the input field.

For the radially-symmetric scheme,

r

This is the radial sampling points with a unit of m. In the Hankel transform with high accuracy (FHATHA), the spatial radial dimension is sampled non-uniformly, following an exponential sampling strategy with denser sampling around the center [Eqs. (6.11) and (6.12)].

field

This is the temporal amplitude of the input field ($\mathbb{A}(t)$ with the unit $\sqrt{\mathrm{W/m^2}}$). Its size is $N_t \times N_r \times 1 \times N_p$.

If its size is $N_t \times N_r \times N_z \times N_p$, only the last N_z is taken as the input field.

3.3 sim

Below are the most basic parameters for a simulation.

betas

In UPPE, we not only create a moving frame that follows the pulse with the inverse velocity $\beta_{(1)}$ but extract out the reference propagation constant $\beta_{(0)}$. The benefit of extracting $\beta_{(0)}$ is that it reduces the rate of global phase increment such that the simulation can run with a larger step. This is similar to the limitation of multimode simulations that different spatial modes have different propagation constants that generate beating. To resolve the multimode beating, the size of the z-step cannot be too large.

This "betas" is a 2×1 column vector.

$$\begin{bmatrix} \beta_{(0)} \\ \beta_{(1)} \end{bmatrix} \tag{3.1}$$

f0

The center frequency (THz). It's a scalar.

save_period

The length between saved fields (m). If it's zero, it's equivalent to save_period=fiber.L0 that saves only the input and output fields.

Be aware that this number needs to be a divisor of the fiber length, fiber.L0; otherwise, UPPE3D_propagate() will throw an error. I have the maximum step size set as the $\frac{1}{10}$ of the save_period and the position of the saved fields will be chosen as the one that first passes through each saved point.

3.3.1 Polarization modes

Here are the parameters if the simulation includes polarization modes.

scalar

false (0)	includes polarization-mode coupling
true (1)	don't include polarization-mode coupling

ellipticity

The ellipticity of the polarization modes. Please refer to "Nonlinear Fiber Optics, eq (6.1.18) Agrawal" for the equations.

0	linear polarization	(+,-)=(x,y)
1	circular polarization	(+,-)=(right,left)

3.3.2 Adaptive-step method

Here are the parameters for the adaptive-step method which this code always uses. All parameters are contained within a "sim.adaptive_dz" structure. The user doesn't need to specify whether to use adaptive-step method or not; the code determines itself. With the adaptive-step method, the initial step size is set to a small 10^{-6} m.

adaptive_dz.threshold

The nonlinear threshold of the upper-level adaptive-step method (see Ch. 7). It controls the accuracy of the simulation and determines whether to increase or decrease the step size. I typically use 10^{-6}.

adaptive_dz.DW_threshold

The dispersion+diffraction threshold of the nested adaptive-step method (see Ch. 7). It controls the accuracy of the simulation and determines whether to increase or decrease the step size. I typically use 10^{-6}.

adaptive_dz.max_dz

The maximum z-step size (m) of the adaptive-step method. It's 1/10 the save_period by default.

3.3.3 Hankel transform

Here are the parameters required to run the radially-symmetric scheme. They are generated beforehand with the function "Hankel_info()" and stored in the "Hankel" stucture for simulations as

Hankel.r

The radial sampling points with a unit of m [Eq. (6.10b)]. Its dimension is $(1, N_r)$.

Hankel.kr
The radial sampling points in $k_\perp$-space with a unit of 2π/m [Eq. (6.10b)]. Its dimension is $(1, N_r)$.

Hankel.l0
The ℓ_0 scalar constant used in FHATHA [Eq. (6.15)].

Hankel.exp_prefactor
The exponential prefactor in P or R in FHATHA [Eq. (6.19a)].

Hankel.Q
The Q parameter in FHATHA [Eq. (6.19b)].

3.3.4 Algorithm to use

gpu_yes

true (1)	use GPU
false (0)	don't use GPU

include_Raman

false(0)	ignore Raman effect
true(1)	Raman model including the anisotropic contribution ("Ch. 2.3, p. 43" and "Ch. 8.5, p. 340," Nonlinear Fiber Optics (5th), Agrawal)

Typically, only isotropic Raman is considered, which is based on a single vibrational Raman mode of molecules (Ch. 2.3, p.42, Nonlinear Fiber Optics (5th), Agrawal). Here, we include the anisotropic part if there is an existing model, such as the one in silica ("Ch. 2.3, p. 43" and "Ch. 8.5, p. 340," Nonlinear Fiber Optics (5th), Agrawal).

For more details about anisotropic Raman, please read "Raman response function for silica fibers," by Q. Lin and Govind P. Agrawal (2006). Besides silica, chalcogenide and ZBLAN are also included.

pulse_centering
Because the pulse will evolve in the fiber, it's hard to have the moving frame always move with the same speed as the pulse. As a result, the pulse will go out of the time window and come back from the other side due to the use of periodic assumption of discrete Fourier transform. The shift in time is saved in "prop_output.t_delay" so that you don't lose the information

When enabling pulse_centering, the pulse will be centered to the center of the time window based on the moment of the field intensity ($|A|^2$).

true (1)	center the pulse according to the time window
false (0)	don't center the pulse

cuda_dir_path
The path to the cuda directory into which ptx files will be compiled and stored. This is "/UPPE3D/cuda/."

gpuDevice.Index

The GPU to use. It's typically 1 if the computer has only one GPU. MATLAB starts the index with 1. By starting multiple MATLAB sessions, we can run simulations on different GPUs simultaneously if different sessions are set with different GPU indices here.

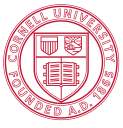
Here are the parameters for the progress bar used in the simulation. It's useful in general to see how a simulation progresses.

progress_bar

true (1)	show progress bar
false (0)	don't show progress bar

progress_bar_name

The name of the GMMNLSE shown on the progress bar. If not set (no "sim.progress_bar_name"), it uses a default empty string, ''.



Cornell University

Chapter 4

Output arguments

field
The $N_t \times N_x \times N_y \times N_z \times N_p$ output field.

dt
This is the time sampling step $\triangle t$ with a unit of ps.

z
This is the positions of each saved field.

dz
The z-step size (m).

This contains the step size at each saved point. You can see how the step size evolves through the propagation with this parameter.

betas
The "sim.betas," $[\beta_{(0)}; \beta_{(1)}]$, used in this propagation.

t_delay
The time delay of the pulse at each saved point due to pulse centering.

seconds
The time spent for this simulation.

For the xy scheme,

dx
This is the spatial sampling step $\triangle x$ with a unit of m.

dy
This is the spatial sampling step $\triangle y$ with a unit of m.

For the radially-symmetric scheme,

r

This is the radial sampling points with a unit of m. In the Hankel transform with high accuracy (FHATHA), the spatial radial dimension is sampled non-uniformly, following an exponential sampling strategy with denser sampling around the center [Eqs. (6.11) and (6.12)].

Chapter 5

load_default_UPPE3D_propagate()

Because of the overwhelming parameters of input arguments, I've created a function that loads the default value for each parameter. If a user has specified the value already, the user's value precedes over the default one.

Here is a typical way of calling this function.

```
[fiber,sim] = load_default_UPPE3D_propagate(input_fiber,...
                                            input_sim])
```

input_fiber and input_sim are user-defined parameters. Below are some examples.

```
% User-defined parameters
fiber.L0 = 3; % m

% Incorporate default settings
[fiber,sim] = load_default_UPPE3D_propagate(fiber,[]);

% If there are "sim" settings
sim.gpu_yes = false;
[fiber,sim] =  load_default_UPPE3D_propagate(fiber,sim);

% Use only user-defined "sim", not "fiber"
[fiber,sim] = load_default_UPPE3D_propagate([],sim);
```

Besides loading the default values, this function gives a user more options to obtain several parameters. This function transforms them into the allowed parameters of UPPE3D_propagate(). I list them below. If both equivalence are specified unfortunately, the allowed UPPE3D_propagate() input has the higher priority.

Description	Allowed UPPE3D_propagate()'s input	Equivalent input arguments for this function
center frequency/wavelength	sim.f0 (THz)	sim.lambda0 (m)

Below is the process flow of this "load_default_UPPE3D_propagate()" function. Read this if you're not sure whether your input will be used or overwritten. Because user-defined parameters take precedence, overwritten should happen only for (f0,lambda0) mentioned above.

```
% <-- Uncorrelated parameters are loaded directly -->

sim.f0 - depend on input f0 or lambda0
```

```
4           If no input f0 or lambda0, f0=3e5/1030e-9 (THz)
5
6   % If there's a user-defined one, use user's instead for the parameters below.
7   % Below I list the default values -->
8   fiber.fiber = '1060XP';
9   fiber.material = 'silica';
10  fiber.n2 = 2.3e-20;
11
12  sim.save_period = 0;
13
14  sim.ellipticity = 0; % linear polarization
15  sim.scalar = true;
16
17  sim.adaptive_dz.DW_threshold = 1e-6;
18  sim.adaptive_dz.threshold = 1e-6;
19
20  sim.gpu_yes = true;
21  sim.pulse_centering = true;
22  sim.include_Raman = true;
23  sim.gpuDevice.Index = 1;
24  sim.progress_bar = true;
25  sim.progress_bar_name = '';
26  sim.cuda_dir_path = 'UPPE3D/cuda';
```

Chapter 6

Radially-symmetric system

In scenarios that require a full-field simulation, pulse propagation most likely exhibits radial symmetry, such as in a multipass cell and a multiplate compressor. Although I develop this code initially for simulating nonlinear propagation in a multimode fiber that follows a non-radially-symmetric propagation, it can be useful to develop its radially-symmetric version for other uses.

6.1 Introduction

To develop the radially-symmetric version of the code, it is crucial to first introduce the Hankel transform, which naturally appears in the two-dimensional Fourier transform [6] of a radially-symmetric function:

$$\begin{aligned}
A(k_\perp) = \mathfrak{F}_{k_\perp}[A(\vec{r}_\perp = r_\perp)] &\equiv C_{\mathfrak{F}}^2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} A(r_\perp) \mathrm{e}^{ic_s\left(k_x x + k_y y\right)} \,\mathrm{d}x\,\mathrm{d}y \\
&= C_{\mathfrak{F}}^2 \int_0^{2\pi}\int_0^{\infty} A(r_\perp) \mathrm{e}^{ic_s\left|\vec{k}_\perp\right||\vec{r}_\perp|\cos\theta} |\vec{r}_\perp| \,\mathrm{d}|\vec{r}_\perp| \,\mathrm{d}\theta \quad , \vec{k}_\perp \cdot \vec{r}_\perp = \left|\vec{k}_\perp\right||\vec{r}_\perp|\cos\theta \\
&= C_{\mathfrak{F}}^2 \int_0^{\infty} A(r_\perp)|\vec{r}_\perp| \left(\int_0^{2\pi} \mathrm{e}^{ic_s\left|\vec{k}_\perp\right||\vec{r}_\perp|\cos\theta} \,\mathrm{d}\theta\right) \mathrm{d}|\vec{r}_\perp| \\
&= 2\pi C_{\mathfrak{F}}^2 \int_0^{\infty} A(r_\perp) J_0(|c_s|k_\perp r_\perp) r_\perp \,\mathrm{d}r_\perp \\
&\equiv 2\pi C_{\mathfrak{F}}^2 \mathfrak{H}[A](k_\perp) \quad \because |c_s| = 1,
\end{aligned} \tag{6.1}$$

where $J_0(z) = \frac{1}{2\pi}\int_0^{2\pi} \mathrm{e}^{iz\cos\theta}\,\mathrm{d}\theta$ is employed. Notations are simplified with $k_\perp = \left|\vec{k}_\perp\right| = \left|(k_x, k_y)\right|$ and $r_\perp = |\vec{r}_\perp| = \left|(x, y)\right|$. Similarly, the inverse Fourier transform becomes

$$\begin{aligned}
A(r_\perp) = \mathfrak{F}_{k_\perp}^{-1}[A(k_\perp)] &\equiv C_{\mathfrak{IF}}^2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} A(k_\perp) \mathrm{e}^{-ic_s\left(k_x x + k_y y\right)} \,\mathrm{d}k_x\,\mathrm{d}k_y \\
&= 2\pi C_{\mathfrak{IF}}^2 \int_0^{\infty} A(k_\perp) J_0(|c_s|k_\perp r_\perp) k_\perp \,\mathrm{d}k_\perp \\
&\equiv 2\pi C_{\mathfrak{IF}}^2 \mathfrak{H}^{-1}[A](k_\perp).
\end{aligned} \tag{6.2}$$

The minus sign in the exponent disappears due to the integral over a full 2π angle; as a result, the Hankel transform displays the same formulation whatever the Fourier-transform convention is. In general, Hankel transform is defined for $\nu \geq -1/2$; however, because we are more interested in using it to solve radially-symmetric systems, discussion is limited only to Hankel transform of order 0 (*i.e.*, $\nu = 0$). The Hankel transform of order 0 is defined as

$$\begin{aligned} A_H(k_\perp) = \mathfrak{H}[A(r_\perp)] &\equiv \int_0^\infty A(r_\perp) J_0(k_\perp r_\perp) r_\perp \, \mathrm{d}r_\perp \\ A(r_\perp) = \mathfrak{H}^{-1}[A_H(k_\perp)] &\equiv \int_0^\infty A_H(k_\perp) J_0(k_\perp r_\perp) k_\perp \, \mathrm{d}k_\perp. \end{aligned} \tag{6.3}$$

With the orthogonality relation

$$\int_0^\infty J_\nu(kr) J_\nu(k'r) r \, \mathrm{d}r = \frac{\delta(k-k')}{k}, \quad k, k' > 0, \tag{6.4}$$

Eq. (6.3) forms a Hankel-transform pair with $\mathfrak{H}\mathfrak{H}^{-1} = \mathfrak{H}^{-1}\mathfrak{H} = 1$. This relation can also be verified with one-dimensional $\mathfrak{F}\mathfrak{F}^{-1} = \mathfrak{F}^{-1}\mathfrak{F} = 1$ by noting that $C_\mathfrak{F} C_{\mathfrak{I}\mathfrak{F}} = \frac{1}{2\pi}$ with the following Fourier-transform representation:

$$\mathfrak{F}_{k_\perp} = 2\pi C_\mathfrak{F}^2 \mathfrak{H} \tag{6.5a}$$

$$\mathfrak{F}_{k_\perp}^{-1} = 2\pi C_{\mathfrak{I}\mathfrak{F}}^2 \mathfrak{H}^{-1}. \tag{6.5b}$$

Hankel transform satisfies a similar convolution theorem to the Fourier transform, which follows

$$\mathfrak{H}[A **B] = 2\pi \mathfrak{H}[A]\mathfrak{H}[B] \tag{6.6a}$$

$$\mathfrak{H}^{-1}[A **B] = 2\pi \mathfrak{H}^{-1}[A]\mathfrak{H}^{-1}[B], \tag{6.6b}$$

derived by applying Eq. (6.5) to the two-dimensional version of the Fourier-transform convolution theorem:

$$\mathfrak{F}_{k_\perp}[A **B] = \frac{1}{C_\mathfrak{F}^2} \mathfrak{F}_{k_\perp}[A]\mathfrak{F}_{k_\perp}[B],$$

with $A **B = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} A(x,y) B(X-x, Y-y) \, \mathrm{d}x \, \mathrm{d}y$ represents the two-dimensional convolution

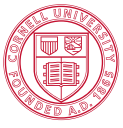
$= \int_0^{2\pi}\int_0^{\infty} A(r) B(\left|\vec{R} - \vec{r}\right|) r \, \mathrm{d}r \, \mathrm{d}\theta$, if A and B are radially symmetric

$\Rightarrow A **B$ is also radially symmetric. (Fig. 6.1)

With Eq. (6.5a), it is transformed into Eq. (6.6a). Similar process can be applied to derive Eq. (6.6b).

Hankel transform satisfies the Parseval's theorem as well. The two-dimensional Parseval's theorem of the Fourier transform is

$$\frac{1}{C_{\mathfrak{I}\mathfrak{F}}^2} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left|A(\vec{r}_\perp)\right|^2 \mathrm{d}x \, \mathrm{d}y = \frac{1}{C_\mathfrak{F}^2} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left|A(\vec{k}_\perp)\right|^2 \mathrm{d}k_x \, \mathrm{d}k_y. \tag{6.7}$$



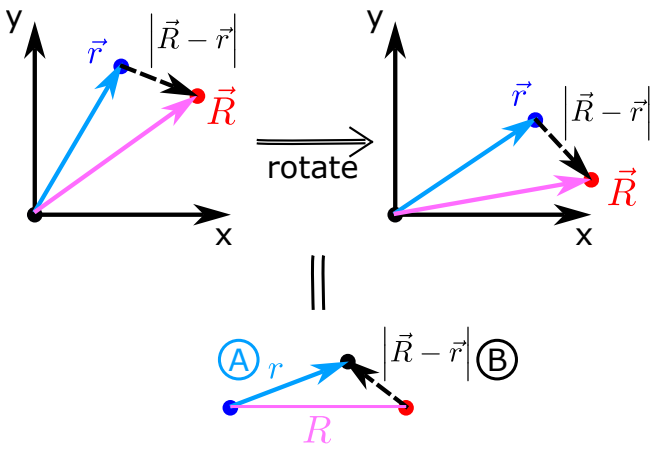


Figure 6.1: Two-dimensional convolution $A**B$ of two radially-symmetric functions $A(r)$ and $B(r)$. Because A depends only on $\vec{r}$ and B only on $\left|\vec{R}-\vec{r}\right|$, the convolution $(A**B)(R) = \int_0^{2\pi}\int_0^{\infty} A(r)B(\left|\vec{R}-\vec{r}\right|)r\,\mathrm{d}r\,\mathrm{d}\theta$ is also radially symmetric. This convolution can be treated as multiplying values depending on the black dot moving over the entire space (bottom figure), whose only independent variable is the distance R between the blue and red dots.

If A is radially symmetric (and thus its 2D Fourier transform $A(\vec{k}_\perp)$ is also radially symmetric [Eq. (6.1)]), with Eq. (6.5a), it becomes

$$\frac{1}{C_{\mathfrak{Jg}}^2}2\pi\int_0^\infty \left|A(r_\perp)\right|^2 r_\perp\,\mathrm{d}r_\perp = \frac{1}{C_{\mathfrak{g}}^2}2\pi\int_0^\infty \left|\left(2\pi C_{\mathfrak{g}}^2\right)A_H(k_\perp)\right|^2 k_\perp\,\mathrm{d}k_\perp, \tag{6.8}$$

which leads to the Parseval's theorem for the Hankel transform:

$$\int_0^\infty \left|A(r_\perp)\right|^2 r_\perp\,\mathrm{d}r_\perp = \int_0^\infty \left|A_H(k_\perp)\right|^2 k_\perp\,\mathrm{d}k_\perp. \tag{6.9}$$

6.2 Numerical computation: Fast Hankel transform of high accuracy (FHATHA)

In this section, we will discuss how to numerically compute the Hankel transform with the "fast Hankel transform of high accuracy," which Magni *et al.* abbreviated as FHATHA [7]. There are a few numerical procedures to compute the Hankel transform, among which one of the most popular is the quasi-fast Hankel transform (QFHT) [8, 9]. FHATHA outperforms QFHT by exhibiting two orders of magnitude lower error while relying on the well-established fft that offers computational efficiency.

First, both the fields in real $r_\perp$-space and $k_\perp$-space are sampled at

$$r_{\perp,n} = r_\perp^{\max}\zeta_n \tag{6.10a}$$
$$k_{\perp,n} = k_\perp^{\max}\zeta_n, \tag{6.10b}$$

where ζ_n represents the normalized coordinate that follows

$$\zeta_n = \zeta_0 \mathrm{e}^{\alpha n}, \quad n = 0, 1, \cdots, N_\perp - 1 \tag{6.11}$$

with $\zeta_0 = \frac{1+e^{\alpha}}{2} e^{-\alpha N_\perp}$ such that $\zeta_n, \forall n \geq 1$ are in the center of each interval $[\xi_n, \xi_{n+1}]$. ξ_n is the normalized coordinate following

$$\xi_n = \begin{cases} 0, & n = 0 \\ \xi_0' e^{\alpha n}, & n = 1, 2, \cdots, N_\perp \end{cases}, \tag{6.12}$$

where $\xi_0' = e^{-\alpha N_\perp}$ (Fig. 6.2). α is given such that the widths of the first and the last intervals are the same:

$$\begin{aligned} &\xi_1 - \xi_0 = \xi_{N_\perp} - \xi_{N_\perp - 1} \\ &\Rightarrow e^{-\alpha(N_\perp - 1)} = 1 - e^{-\alpha}, \end{aligned} \tag{6.13}$$

which can be easily solved numerically. During computing the Hankel transform, the function needs to be evaluated at the center of each interval. However, ζ_0 is not at the center of $[\xi_0, \xi_1]$, as we redefine ξ_0 from ξ_0' to 0 [Eq. (6.12)]. Therefore, $A(r_{\perp,0} = r_\perp^{\max}\zeta_0)$ needs to be replaced with $A(r'_{\perp,0} = r_\perp^{\max}\zeta_0')$ (with $\zeta_0' = \xi_1/2$ being at the center of $[\xi_0 = 0, \xi_1]$), which is approximated through extrapolation of $A(r_{\perp,0})$ and $A(r_{\perp,1})$ with a parabola with zero derivative at the origin, a reasonable approximation curve for a radially-symmetric function. In FHATHA, an extra evaluation at $\zeta_{N_\perp}$, beyond the range of Eq. (6.11), is required, which is simply treated with $A(\zeta_{N_\perp}) = 0$. If we denote A_n as $A(r_{\perp,n})$, eventually the A that is used in FHATHA is

$$\tilde{A}_n = \begin{cases} A(r'_{\perp,0}) = \ell_0 (A_0 - A_1) + A_1, & n = 0 \\ A_n, & n = 1, 2, \cdots, N_\perp - 1 \\ 0, & n = N_\perp \end{cases} \tag{6.14}$$

where

$$ell_0 = \frac{e^{\alpha}(2 + e^{\alpha})}{(1 + e^{\alpha})^2 (1 - e^{-2\alpha})}. \tag{6.15}$$

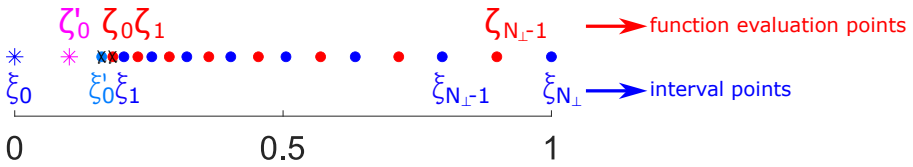


Figure 6.2: Numerical sampling in FHATHA.

FHATHA solves the Hankel transform by summing over Hankel-transform integral of each interval, whose function is evaluated at ζ_n [Eq. (6.11)] (with slight modification at $n = 0$ [Eq. (6.14)]) but interval is defined with ξ_n [Eq. (6.12)]. By use of the relation $\int J_0(u) u \, du = u J_1(u) + C$, it leads to

$$\begin{aligned} A_{H,m} = A_H(k_{\perp,m}) &= \frac{1}{k_{\perp,m}} \sum_{n=0}^{N_\perp - 1} \ell_n (A_n - A_{n+1}) (r_\perp^{\max}\xi_{n+1}) J_1(k_{\perp,m} (r_\perp^{\max}\xi_{n+1})) \\ &= \frac{r_\perp^{\max}}{k_{\perp,m}} \sum_{n=0}^{N_\perp - 1} \left[\ell_n (A_n - A_{n+1}) e^{\alpha(n+1-N_\perp)} \right] J_1(k_\perp^{\max} r_\perp^{\max} \zeta_0 e^{\alpha(m+n+1-N_\perp)}), \end{aligned} \tag{6.16}$$

where $\ell_n = 1$ if $n \neq 0$, and J_1 is the Bessel function of order 1.

If we define the discrete cross convolution $\star_D$ as

$$(P \star_D Q)_m \equiv \sum_n \overline{P_n} Q_{m+n}. \tag{6.17}$$

It satisfies

$$\begin{aligned}\mathfrak{F}_D[P \star_D Q] &= \frac{1}{C_{\mathfrak{F}_D}} \overline{\mathfrak{F}_D[P]} \mathfrak{F}_D[Q] = \frac{1}{C_{\mathfrak{F}_D}} \left(\frac{C_{\mathfrak{F}_D}}{C_{\mathfrak{I}\mathfrak{F}_D}} \mathfrak{F}_D^{-1}[\overline{P}] \right) \mathfrak{F}_D[Q] \\ &= \frac{1}{C_{\mathfrak{I}\mathfrak{F}_D}} \mathfrak{F}_D^{-1}[\overline{P}] \mathfrak{F}_D[Q]. \end{aligned} \tag{6.18}$$

Since DFT inherently exhibits periodicity, zero-padding is required if P or Q does not display periodicity beyond the sampling window; otherwise, aliasing will occur by direct application of Eq. (6.18).

If we let

$$P_n = \overline{R}_n = \begin{cases} \ell_n \left(\overline{A}_n - \overline{A}_{n+1} \right) \mathrm{e}^{\alpha(n+1-N_\perp)}, & n = 0, 1, \cdots, N_\perp - 1 \\ 0, & n = N_\perp, N_\perp + 1, \cdots, T^w_{\text{extended}} \quad \text{(zero-padding)} \end{cases} \tag{6.19a}$$

$$Q_n = J_1(k_\perp^{\max} r_\perp^{\max} \zeta_0 \mathrm{e}^{\alpha(n+1-N_\perp)}), \qquad n = 0, 1, \cdots, T^w_{\text{extended}} \tag{6.19b}$$

Eq. (6.16) becomes

$$A_{H,m} = \frac{r_\perp^{\max}}{k_{\perp,m}} \frac{1}{C_{\mathfrak{I}\mathfrak{F}_D}} \mathfrak{F}_D^{-1} \left[\mathfrak{F}_D^{-1}[R] \mathfrak{F}_D[Q] \right] \tag{6.20a}$$

$$\overset{\text{or also}}{=} \frac{r_\perp^{\max}}{k_{\perp,m}} \frac{1}{C_{\mathfrak{F}_D}} \mathfrak{F}_D \left[\mathfrak{F}_D[R] \mathfrak{F}_D^{-1}[Q] \right], \tag{6.20b}$$

which is meaningful only for $m = 0, 1, \cdots, (N_\perp - 1)$ with others discarded after FHATHA. The zero-padding range is determined by $T^w_{\text{extended}} \geq 2N_\perp - 2$ [so that there are a total of $(2N_\perp - 1)$ sampling points]. This zero-padding procedure is similar to the convolution operation with clipped extended signals. However, since the summation in Eq. (6.16) includes Q_n up to $n = 2N_\perp - 2$, Q_n must be analytically computed at least for $n = 0, 1, \cdots, (2N_\perp - 2)$, rather than simple zero-padding as in the convolution theorem. Note that Q can be precomputed only once before the simulation. In the physical Fourier-transform convention ($C_{\mathfrak{F}_D} = \frac{1}{\mathfrak{N}}$ and thus $C_{\mathfrak{I}\mathfrak{F}_D} = 1$), $A_{H,m} = \frac{r_\perp^{\max}}{k_{\perp,m}} \mathfrak{F}_D^{-1} \left[\mathfrak{F}_D^{-1}[R] \mathfrak{F}_D[Q] \right]$, which is (in MATLAB syntax below)

```
A_H = r_max./k.*fft(fft(R).*ifft(Q));
```

Since Hankel transform and inverse Hankel transform exhibit the same form [Eq. (6.3)], inverse FHATHA follows Eq. (6.20) by swapping the symbols $r \leftrightarrow k$. Note that Q is independent of Hankel or inverse Hankel transforms due to its $(r_\perp, k_\perp)$ symmetry. Fig. 6.3 shows two verification examples that demonstrate its excellent consistency with the analytical formula.

The exponential sampling strategy [Eqs. (6.11) and (6.12)] of FHATHA introduces an advantage over typical uniform sampling. Mostly, the radially-symmetric field is the strongest at the spatial center. In particular, in cases such as self-focusing for a high-peak-power pulse

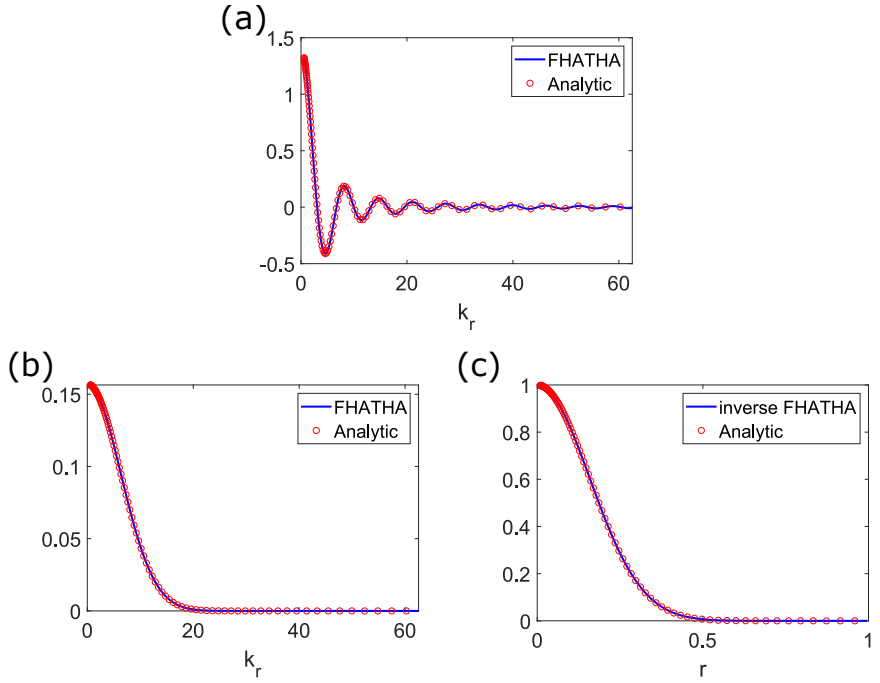


Figure 6.3: Verifications of FHATHA. Hankel transforms of (a) $f(r) = \sqrt{\frac{5}{2\pi}} r^2$ and (b) $f(r) = \mathrm{e}^{-20r^2}$, which are $\sqrt{10\pi}\frac{2\eta J_0(\eta)+(\eta^2-4)J_1(\eta)}{\eta^3}$ (with $\eta = r_\perp^{\max} k_\perp$) and $\frac{\pi}{20}\mathrm{e}^{-k_\perp^2/80}$, respectively. (c) Inverse Hankel transform of (b). This is used to further verify the operation of inverse Hankel transform and see whether the signal can be recovered.

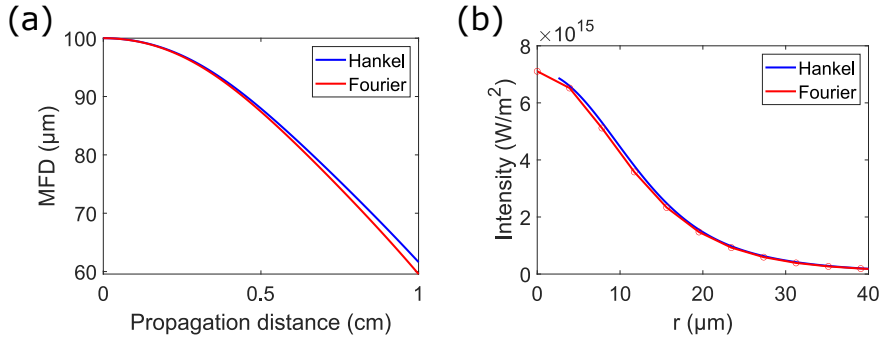


Figure 6.4: (a) Mode-field-diameter (MFD) evolution of a pulse propagating in a 1-cm-long bulk silica, simulated with the Hankel transform or 2D spatial Fourier transform. The input pulse is 1-ps-long and has 8-μJ pulse energy, with a Gaussian spatial profile of 100-μm MFD. (b) The intensity profile of the pulse.

(Fig. 6.4), the spatial extent of the field decreases as it propagates. This can lead to undersampling of the field, which is resolved by FHATHA's exponential sampling that densely samples around the center.

Unlike the frequency window in the Fourier transform that is determined by the temporal sampling period, the $k_\perp$ window can be chosen quite arbitrarily. Its value is determined by the signal's maximum $\vec{k}_\perp$ vector.

Chapter 7

Diagram of the calling sequence

It's not necessary to know how or when each function is called. I keep it here for documentation or in case someone wants to modify the code.

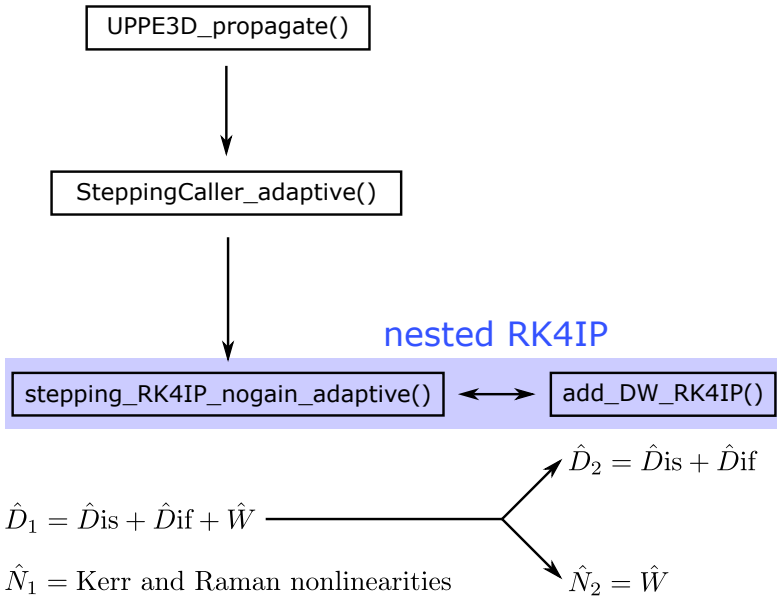
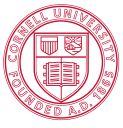


Figure 7.1: Diagram of the calling sequence.

The code uses "RK4IP" (Runge-Kutta in the interaction picture) [4, 5] with an adaptive step-size control, whose dispersion operation $\hat{D}_1$ contains dispersion $\hat{D}$is, diffraction $\hat{D}$if, and waveguide $\hat{W}$ effects, and nonlinear operation $\hat{N}_1$ contains Kerr and Raman nonlinearities. However, due to the incommutableness of the dispersion+diffraction $\hat{D}\text{is} + \hat{D}\text{if} \sim K_z$ and the waveguide $\hat{W} \sim k_W^2$ operators, we apply a second RK4IP for them whenever we need to apply $\hat{D}_1$. As typical dispersion and nonlinear operators that are incommutable, the nested RK4IP has a dispersion operator $\hat{D}_2 = \hat{D}\text{is} + \hat{D}\text{if}$ and a nonlinear operator $\hat{N}_2 = \hat{W}$. Because of two adaptive-step RK4IP, there are two threshold parameters to control each behavior, which by default are both set to 10^{-6}.


CORNELL UNIVERSITY
FOUNDED A.D. 1865

Chapter 8

Derivation

In this chapter, I show the derivation of 3D-UPPE [Eq. (1.1)]. Although it has been shown in the paper [10], I realized that there are a few mistakes in equations. Filing an errata might be a future option, but it's good for me to have one document that I can easily edit and update.

In fiber optics, light propagates in a dielectric medium, or gas if it is a hollow-core fiber, with its waveguide boundary conditions. Since there are no charge and current sources, such electromagnetic fields are governed by the following Maxwell's equations:

$$\nabla \times \vec{\mathbb{E}}(\vec{x},t) = -\mu_0 \partial_t \vec{\mathbb{H}}(\vec{x},t) \tag{8.1a}$$

$$\nabla \times \vec{\mathbb{H}}(\vec{x},t) = \epsilon_0 \partial_t \left[\epsilon_r(\vec{x},t) * \vec{\mathbb{E}}(\vec{x},t)\right] + \partial_t \vec{\mathbb{P}}(\vec{x},t), \tag{8.1b}$$

where $\vec{\mathbb{E}}$ and $\vec{\mathbb{H}}$ are electric and magnetic fields, respectively; $\vec{\mathbb{P}}$ is the induced nonlinear polarization; ϵ_r is the relative dielectric constant of the medium; ϵ_0 and μ_0 are permittivity and permeability in vacuum. We can do spectral Fourier transform to the equations and consider only their positive-frequency components (analytic signal).

$$\mu_0 i\omega \vec{\mathcal{H}} = \nabla \times \vec{\mathcal{E}} \tag{8.2a}$$

$$\vec{\mathcal{J}} - i\omega\vec{\mathcal{P}} - i\omega\epsilon_0\epsilon_r\vec{\mathcal{E}} = \nabla \times \vec{\mathcal{H}}, \tag{8.2b}$$

which leads to

$$\begin{aligned} \nabla \times \nabla \times \vec{\mathcal{E}} &= \nabla\left(\nabla \cdot \vec{\mathcal{E}}\right) - \nabla^2\vec{\mathcal{E}} \\ &= \mu_0 i\omega\vec{\mathcal{J}} + \mu_0\omega^2\vec{\mathcal{P}} + \omega^2\mu_0\epsilon_0\epsilon_r\vec{\mathcal{E}} \end{aligned} \tag{8.3}$$

$\vec{\mathcal{E}}$, $\vec{\mathcal{H}}$, $\vec{\mathcal{J}}$, and $\vec{\mathcal{P}}$ are the complex-valued analytic signal of their corresponding real-valued counterparts. We assume that the electric field and the medium response (current $\vec{\mathcal{J}}$, nonlinear polarization $\vec{\mathcal{P}}$) are transverse, *i.e.*, perpendicular to the propagation direction determined by the wave number $\vec{k}$. This standard assumption in propagation of electromagnetic fields means that the term $\nabla\left(\nabla \cdot \vec{\mathcal{E}}\right)$ can be neglected. This remains valid as long as beams are not too strongly focused. When beam numerical aperture exceeds a few percent, a small longitudinal component $\vec{\mathcal{E}}$ may develop close to the focus and makes this approximation invalid. In addition, we also assume that there is no current $\vec{\mathcal{J}} = 0$. After these assumptions, we obtain

$$\nabla^2\vec{\mathcal{E}}(\vec{r}_\perp,z,\omega) + k^2(\vec{r}_\perp,\omega)\vec{\mathcal{E}}(\vec{r}_\perp,z,\omega) = -\mu_0\omega^2\vec{\mathcal{P}}(\vec{r}_\perp,z,\omega), \tag{8.4}$$

where $k^2 = \omega^2\mu_0\epsilon_0\epsilon_r$ and $\vec{r}_\perp = (x, y)$. Note that $\epsilon_r = \epsilon_r(\vec{r}_\perp, \omega)$ depends on x and y and is independent of z. This accounts for the variation in refractive index of the cross-sectional profile. We can split k^2 into its frequency-dependent part $k_\omega^2(\omega)$ and spatially-dependent (but frequency-independent) part $k_W^2(\vec{r}_\perp, \omega)$:

$$k^2(\vec{r}_\perp, \omega) = k_\omega^2(\omega) + k_W^2(\vec{r}_\perp, \omega). \tag{8.5}$$

$k_W^2(\vec{r}_\perp, \omega)$ determines the waveguide effect. It is zero if the medium is homogeneous.

By transforming Eq. (8.4) into the cross-sectional $k_\perp$-space, we obtain

$$\left[\partial_z^2 + \left(k_\omega^2(\omega) - \left|\vec{k}_\perp\right|^2\right)\right]\vec{\mathcal{E}}(\vec{k}_\perp, z, \omega) + \mathfrak{F}_{k_\perp}\left[k_W^2(\vec{r}_\perp, \omega)\vec{\mathcal{E}}(\vec{r}_\perp, z, \omega)\right] = -\mu_0\omega^2\vec{\mathcal{P}}(\vec{k}_\perp, z, \omega). \tag{8.6}$$

Its $\left[\partial_z^2 + \left(k_\omega^2(\omega) - \left|\vec{k}_\perp\right|^2\right)\right]$ can be rewritten into $\left[\left(\partial_z + iK_z(\vec{k}_\perp, \omega)\right)\left(\partial_z - iK_z(\vec{k}_\perp, \omega)\right)\right]$, where $K_z(\vec{k}_\perp, \omega) = \sqrt{k_\omega^2(\omega) - \left|\vec{k}_\perp\right|^2}$. Assume that the forward propagating component is larger than the backward propagating one: $\partial_z \to iK_z$. The above equation becomes

$$\partial_z\vec{\mathcal{E}}(\vec{k}_\perp, z, \omega) = iK_z\vec{\mathcal{E}}(\vec{k}_\perp, z, \omega) + i\frac{1}{2K_z}\mathfrak{F}_{k_\perp}\left[k_W^2(\vec{r}_\perp, \omega)\vec{\mathcal{E}}(\vec{r}_\perp, z, \omega)\right] + i\frac{\omega^2}{2K_z c^2}\frac{\vec{\mathcal{P}}(\vec{k}_\perp, z, \omega)}{\epsilon_0}. \tag{8.7}$$

By introducing the envelope of the analytic signal:

$$\begin{aligned}
\vec{\mathbb{E}}(\vec{x}, t) &= \frac{1}{2}\left[\vec{\mathcal{E}}(\vec{x}, t) + \text{c.c.}\right] \quad , \vec{\mathcal{E}} \text{ is the analytic signal of } \vec{\mathbb{E}} \\
&= \frac{1}{2}\left[\vec{E}(t)\mathrm{e}^{i\left(\beta_{(0)}z - \omega_0 t\right)} + \text{c.c.}\right] \qquad (8.8a) \\
\vec{\mathbb{P}}(\vec{x}, t) &= \frac{1}{2}\left[\vec{\mathcal{P}}(\vec{x}, t) + \text{c.c.}\right] \quad , \vec{\mathcal{P}} \text{ is the analytic signal of } \vec{\mathbb{P}} \\
&= \frac{1}{2}\left[\vec{P}(t)\mathrm{e}^{i\left(\beta_{(0)}z - \omega_0 t\right)} + \text{c.c.}\right], \qquad (8.8b)
\end{aligned}$$

Eq. (8.7) becomes

$$\partial_z\vec{E}(\vec{k}_\perp, z, \omega) = i\left(K_z - \beta_{(0)}\right)\vec{E}(\vec{k}_\perp, z, \omega) + i\frac{1}{2K_z}\mathfrak{F}_{k_\perp}\left[k_W^2(\vec{r}_\perp, \omega)\vec{E}(\vec{r}_\perp, z, \omega)\right] + i\frac{\omega^2}{2K_z c^2}\frac{\vec{P}(\vec{k}_\perp, z, \omega)}{\epsilon_0}. \tag{8.9}$$

Kerr and Raman nonlinearity

Note that, in general, the nonlinear polarization has the following form [3]:

$$\vec{\mathbb{P}}(t) = \int_{-\infty}^{\infty}\chi^{(3)}(t_1, t_2, t_3) \vdots \vec{\mathbb{E}}(t - t_1)\vec{\mathbb{E}}(t - t_2)\vec{\mathbb{E}}(t - t_3)\,\mathrm{d}t_1\,\mathrm{d}t_2\,\mathrm{d}t_3, \tag{8.10}$$

where

$$\chi^{(3)}(t_1, t_2, t_3) = \delta(t_1)\delta(t_2 - t_3)\hat{R}^{ijk\ell}(t_3) \tag{8.11a}$$

$$\hat{R}^{ijk\ell}(t) = \epsilon_0\chi^{(3)}_{\text{electronic}}\frac{\delta^{ij}\delta^{k\ell} + \delta^{ik}\delta^{j\ell} + \delta^{i\ell}\delta^{jk}}{3}\delta(t) + \mathtt{R}_a(t)\delta^{ij}\delta^{k\ell} + \mathtt{R}_b(t)\frac{\delta^{ik}\delta^{j\ell} + \delta^{i\ell}\delta^{jk}}{2} \tag{8.11b}$$

includes the instantaneous Kerr nonlinearity and the delayed isotropic ($\mathtt{R}_a$) and anisotropic ($\mathtt{R}_b$) Raman nonlinearities; therefore,

$$\mathbb{P}^i(t) = \sum_{jk\ell} \mathbb{E}^j(t) \int_{-\infty}^{\infty} \hat{R}^{ijk\ell}(\tau)\mathbb{E}^k(t-\tau)\mathbb{E}^\ell(t-\tau)\,\mathrm{d}\tau, \qquad \text{by applying } (t_3 \to \tau). \tag{8.12}$$

First, we solve for the Kerr nonlinearity:

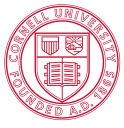
$$\begin{aligned}
\mathbb{P}^i(\vec{r},t) &= \sum_{jkl} \epsilon_0 \frac{1}{2}\left[E^j(t)\mathrm{e}^{i\left(\beta_{(0)}z-\omega_0 t\right)} + \text{c.c.}\right] \chi^{(3)}_{\text{electronic}} \frac{\delta^{ij}\delta^{k\ell} + \delta^{ik}\delta^{j\ell} + \delta^{i\ell}\delta^{jk}}{3} \\
&\qquad \times \frac{1}{2}\left[E^k(t)\mathrm{e}^{i\left(\beta_{(0)}z-\omega_0 t\right)} + \text{c.c.}\right] \frac{1}{2}\left[E^\ell(t)\mathrm{e}^{i\left(\beta_{(0)}z-\omega_0 t\right)} + \text{c.c.}\right] \\
&= \frac{\epsilon_0 \chi^{(3)}_{\text{electronic}}}{8} \sum_k \Bigg\{ \left(E^i(E^k)^2\mathrm{e}^{3i\left(\beta_{(0)}z-\omega_0 t\right)} + \text{c.c.}\right) \\
&\qquad + \left[\left((E^i)^*(E^k)^2 + 2E^i\left|E^k\right|^2\right)\mathrm{e}^{i\left(\beta_{(0)}z-\omega_0 t\right)} + \text{c.c.}\right]\Bigg\}.
\end{aligned} \tag{8.13}$$

Next, we solve for the isotropic Raman term:

$$\begin{aligned}
\mathbb{P}^i(\vec{r},t) &= \frac{1}{2}\left[E^i(t)\mathrm{e}^{i\left(\beta_{(0)}z-\omega_0 t\right)} + \text{c.c.}\right] \int \mathtt{R}_a(\tau) \sum_k \left\{\frac{1}{2}\left[E^k(t-\tau)\mathrm{e}^{i\left[\beta_{(0)}z-\omega_0(t-\tau)\right]} + \text{c.c.}\right]\right\}^2 \\
&= \frac{1}{8}\sum_k \Bigg\{ \left[E^i(t)\mathrm{e}^{3i\left(\beta_{(0)}z-\omega_0 t\right)} \int \mathtt{R}_a(\tau)\left(E^k(t-\tau)\right)^2 \mathrm{e}^{2i\omega_0\tau}\,\mathrm{d}\tau + \text{c.c.}\right] \\
&\qquad + \left[2E^i(t)\mathrm{e}^{i\left(\beta_{(0)}z-\omega_0 t\right)} \int \mathtt{R}_a(\tau)\left|E^k(t-\tau)\right|^2 \mathrm{d}\tau + \text{c.c.}\right] \\
&\qquad + \left[\left(E^i(t)\right)^* \mathrm{e}^{i\left(\beta_{(0)}z-\omega_0 t\right)} \int \mathtt{R}_a(\tau)\left(E^k(t-\tau)\right)^2 \mathrm{e}^{2i\omega_0\tau}\,\mathrm{d}\tau + \text{c.c.}\right]\Bigg\},
\end{aligned} \tag{8.14}$$

and the anisotropic term:

$$\begin{aligned}
\mathbb{P}^i(\vec{r},t) &= \sum_j \mathbb{E}^j(t) \int \mathtt{R}_b(\tau)\mathbb{E}^i(t-\tau)\mathbb{E}^j(t-\tau)\,\mathrm{d}\tau \\
&= \sum_j \frac{1}{2}\left[E^j(t)\mathrm{e}^{i\left(\beta_{(0)}z-\omega_0 t\right)} + \text{c.c.}\right] \\
&\qquad \int \mathtt{R}_b(\tau)\frac{1}{2}\left\{E^i(t-\tau)\mathrm{e}^{i\left[\beta_{(0)}z-\omega_0(t-\tau)\right]} + \text{c.c.}\right\}\frac{1}{2}\left\{E^j(t-\tau)\mathrm{e}^{i\left[\beta_{(0)}z-\omega_0(t-\tau)\right]} + \text{c.c.}\right\}\mathrm{d}\tau \\
&= \frac{1}{8}\sum_j \Bigg\{ \left[E^j(t)\mathrm{e}^{3i\left(\beta_{(0)}z-\omega_0 t\right)} \int \mathtt{R}_b(\tau)E^i(t-\tau)E^j(t-\tau)\mathrm{e}^{2i\omega_0\tau} + \text{c.c.}\right] \\
&\qquad + \left[E^j(t)\mathrm{e}^{i\left(\beta_{(0)}z-\omega_0 t\right)} \int \mathtt{R}_b(\tau)\left(E^i(E^j)^* + (E^i)^*E^j\right)(t-\tau)\,\mathrm{d}\tau + \text{c.c.}\right]
\end{aligned}$$



$$+\left[(E^j)^*\mathrm{e}^{i\left(\beta_{(0)}z-\omega_0 t\right)}\int \mathtt{R}_b(\tau)E^i(t-\tau)E^j(t-\tau)\mathrm{e}^{2i\omega_0\tau}\,\mathrm{d}\tau+\mathrm{c.c.}\right]\Bigg\}. \tag{8.15}$$

By ignoring the 3rd harmonic terms, which are terms with $\mathrm{e}^{3i\left(\beta_{(0)}z-\omega_0 t\right)}$, the polarization can be expressed, in terms of its envelope, as

$$\begin{aligned}P^i(\vec{r},t)=\frac{1}{4}\sum_j\Bigg\{&\epsilon_0\chi^{(3)}_{\text{electronic}}\left[(E^i)^*(E^j)^2+2E^i\left|E^j\right|^2\right]\\&+2E^i(t)\int \mathtt{R}_a(\tau)\left|E^j(t-\tau)\right|^2\mathrm{d}\tau+(E^i)^*(t)\int \mathtt{R}_a(\tau)(E^j)^2(t-\tau)\mathrm{e}^{2i\omega_0\tau}\,\mathrm{d}\tau\\&+E^j(t)\int \mathtt{R}_b(\tau)\left(E^i(E^j)^*+(E^i)^*E^j\right)(t-\tau)\,\mathrm{d}\tau\\&+(E^j(t))^*\int \mathtt{R}_b(\tau)\left(E^iE^j\right)(t-\tau)\mathrm{e}^{2i\omega_0\tau}\,\mathrm{d}\tau\Bigg\}.\end{aligned} \tag{8.16}$$

We further ignore the delay terms, and obtain

$$\begin{aligned}P^i(\vec{r},t)=\frac{1}{4}\sum_j\Bigg\{&\epsilon_0\chi^{(3)}_{\text{electronic}}\left[(E^i)^*(E^j)^2+2E^i\left|E^j\right|^2\right]\\&+2E^i(t)\int \mathtt{R}_a(\tau)\left|E^j(t-\tau)\right|^2\mathrm{d}\tau\\&+E^j(t)\int \mathtt{R}_b(\tau)\left(E^i(E^j)^*+(E^i)^*E^j\right)(t-\tau)\,\mathrm{d}\tau\Bigg\}.\end{aligned} \tag{8.17}$$

Since $\vec{E}$ above has the unit of V/m, we transform it into $\sqrt{\mathrm{W/m^2}}$, a more commonly used physical quantity such that $\left|\vec{\mathbb{A}}\right|^2$ directly corresponds to the intensity (in $\mathrm{W/m^2}$). The transformation relies on $\left|\vec{\mathbb{A}}\right|^2=\frac{\epsilon_0 n_{\text{eff}}c}{2}\left|\vec{E}\right|^2$, where $\vec{\mathbb{A}}$ has a unit of $\sqrt{\mathrm{W/m^2}}$. The transformed equation looks the same as Eq. (8.9) by replacing $\vec{E}$ with $\vec{\mathbb{A}}$, but the nonlinear polarization $\vec{P}$ becomes

$$\begin{aligned}P^i(\vec{r},t)=\frac{1}{2\epsilon_0 n_{\text{eff}}c}\sum_j\Bigg\{&\epsilon_0\chi^{(3)}_{\text{electronic}}\left[(\mathbb{A}^i)^*(\mathbb{A}^j)^2+2\mathbb{A}^i\left|\mathbb{A}^j\right|^2\right]\\&+2\mathbb{A}^i(t)\int \mathtt{R}_a(\tau)\left|\mathbb{A}^j(t-\tau)\right|^2\mathrm{d}\tau\\&+\mathbb{A}^j(t)\int \mathtt{R}_b(\tau)\left(\mathbb{A}^i(\mathbb{A}^j)^*+(\mathbb{A}^i)^*\mathbb{A}^j\right)(t-\tau)\,\mathrm{d}\tau\Bigg\}.\end{aligned} \tag{8.18}$$

Consider the moving frame:

$$T=t-\beta_{(1)}z \quad\Rightarrow\quad \partial_z\mathbb{A}(\vec{r},t)=\partial_z\mathbb{A}(\vec{r},T)-\beta_{(1)}\partial_T\mathbb{A}(\vec{r},T), \tag{8.19}$$

where the ∂_z on the left-hand side is actually a total derivative. Also, we apply the Fourier transform with respect to the offset frequency $\Omega = \omega - \omega_0$.

$$\partial_z \vec{\mathbb{A}}(\vec{k}_\perp, z, \Omega) = i\left[K_z - \left(\beta_{(0)} + \beta_{(1)}\Omega\right)\right]\vec{\mathbb{A}}(\vec{k}_\perp, z, \Omega) + i\frac{1}{2K_z}\mathfrak{F}_{k_\perp}\left[k_W^2(\vec{r}_\perp, \omega)\vec{\mathbb{A}}(\vec{r}_\perp, z, \Omega)\right] + i\frac{\omega^2}{2K_z c^2}\frac{\vec{P}(\vec{k}_\perp, z, \Omega)}{\epsilon_0}. \quad (8.20)$$

If we apply the model commonly used in solid-core silica fiber, where

$$\epsilon_0\chi^{(3)}_{\text{electronic}} = (1 - f_R)\,\epsilon_0\chi^{(3)}_{xxxx} = (1 - f_R)\,\frac{4\epsilon_0^2 n_{\text{eff}}^2 c}{3}n_2 \quad (8.21a)$$

$$\mathtt{R}_a = f_R f_a \epsilon_0 \chi^{(3)}_{xxxx}\frac{3}{2}h_a = 2 f_R f_a \epsilon_0^2 n_{\text{eff}}^2 c n_2 h_a(t) \quad (8.21b)$$

$$\mathtt{R}_b = f_R f_b \epsilon_0 \chi^{(3)}_{xxxx}\frac{3}{2}h_b = 2 f_R f_b \epsilon_0^2 n_{\text{eff}}^2 c n_2 h_b(t), \quad (8.21c)$$

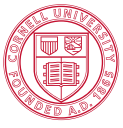
we obtain Eq. (1.3).

In our derivation of nonlinearities, we assume that the electronic nonlinear coefficient $\chi^{(3)}_{\text{electronic}}$ and the effective index n_{eff} are constants. Their frequency dependence is artificially recovered later after the spectral Fourier transform [Eq. (1.3)], which seems not justifiable *a priori* but has been shown to improve the result [11–13].

Because K_z can be close to 0 as $|k_\perp|^2$ is huge, which can easily happen under a decently-huge $k_\perp$-space (with fine spatial sampling), the computation of the nonlinear term will easily blow up due to the $1/K_z$ term, artificially producing unreasonable results. Therefore, to simplify the simulation process, K_z in the denominator of the nonlinear term is replaced with k_ω. This assumption fails when the light has high-$\vec{k}_\perp$ components, such as tightly-focusing. However, during the derivation, we have already assumed that all the field and the responses are transverse to make $\nabla\left(\nabla\cdot\vec{\mathcal{E}}\right) = 0$, so the assumption for $K_z \to k_\omega$ makes sense. For potentially higher accuracy, this replacement can be applied with

$$\frac{1}{K_z} = \frac{1}{\sqrt{k_\omega^2(\omega) - \left|\vec{k}_\perp\right|^2}} = \frac{1}{k_\omega\sqrt{1 - \frac{\left|\vec{k}_\perp\right|^2}{k_\omega^2}}} \approx \frac{1}{k_\omega}\left(1 + \frac{\left|\vec{k}_\perp\right|^2}{2k_\omega^2}\right). \quad (8.22)$$

Eq. (8.22) approximates $\frac{1}{K_z}$ well as $\left|\vec{k}_\perp\right| \ll k_\omega$. Although it induces deviation at large $\left|\vec{k}_\perp\right|$, it does not blow up to infinity as the original $\frac{1}{K_z}$ but rather only slowly increases (such as approaching 1.5 as $\left|\vec{k}_\perp\right| \to k_\omega$). As a result, the replacement with Eq. (8.22) not only introduces more accuracy but is also numerically stable. After all, we achieve deriving the final 3D-UPPE in Eq. (1.1).



Bibliography

[1] B. C. Hall, "The baker—campbell—hausdorff formula", in *Lie groups, lie algebras, and representations: an elementary introduction* (Springer New York, New York, NY, 2003), pp. 63–90.

[2] R. H. Stolen, J. P. Gordon, W. J. Tomlinson, and H. A. Haus, "Raman response function of silica-core fibers", J. Opt. Soc. Am. B **6**, 1159–1166 (1989).

[3] Q. Lin and G. P. Agrawal, "Raman response function for silica fibers", Opt. Lett. **31**, 3086–3088 (2006).

[4] A. M. Heidt, "Efficient Adaptive Step Size Method for the Simulation of Supercontinuum Generation in Optical Fibers", J. Light. Technol. **27**, 3984–3991 (2009).

[5] S. Balac and F. Mahé, "Embedded Runge-Kutta scheme for step-size control in the interaction picture method", Comput. Phys. Commun. **184**, 1211–1219 (2013).

[6] Y.-H. Chen, "Tutorial of Fourier transform for ultrafast optics", arXiv preprint arXiv: 2412.20698 (2025).

[7] V. Magni, G. Cerullo, and S. D. Silvestri, "High-accuracy fast Hankel transform for optical beam propagation", J. Opt. Soc. Am. A **9**, 2031–2033 (1992).

[8] A. E. Siegman, "Quasi fast Hankel transform", Opt. Lett. **1**, 13–15 (1977).

[9] G. P. Agrawal and M. Lax, "End correction in the quasi-fast Hankel transform for optical-propagation problems", Opt. Lett. **6**, 171–173 (1981).

[10] W. Wang, Y. Eisenberg, Y.-H. Chen, C. Xu, and F. Wise, "Efficient temporal compression of 10-μJ pulses in periodic layered Kerr media", Opt. Lett. **49**, 5787–5790 (2024).

[11] F. Poletti and P. Horak, "Description of ultrashort pulse propagation in multimode optical fibers", J. Opt. Soc. Am. B **25**, 1645–1654 (2008).

[12] B. Kibler, J. M. Dudley, and S. Coen, "Supercontinuum generation and nonlinear pulse propagation in photonic crystal fiber: influence of the frequency-dependent effective mode area", Appl. Phys. B **81**, 337–342 (2005).

[13] J. Lægsgaard, "Mode profile dispersion in the generalized nonlinear schrödinger equation", Opt. Express **15**, 16110–16123 (2007).