Solver Guide for the MATLAB gas-filled-fiber pulse propagation

Yi-Hao Chen

Applied and Engineering Physics, Cornell University

February 8, 2024

Contents

Cornell University

Chapter 1

Overview

This package contains all the code needed to simulate pulse propagation in a gas-filled hollow-core fiber. Pulse follows the multimode unidirectional pulse propagation equation (MM-UPPE) which is shown in the following:

$$\partial_z A_p(z, \Omega) = i \left[\beta_p(\omega) - \left(\beta_{(0)} + \beta_{(1)}\Omega \right) \right] A_p(z, \Omega)$$

$$+ \frac{i\omega}{4} \sum_{\ell mn} \left\{ \left(\frac{3}{4} \epsilon_0 \chi_{\text{electronic}}^{(3)} \right) Q_{p\ell mn}^K \mathfrak{F}\left[A_\ell A_m A_n^* \right] \right.$$

$$\left. + \left\{ Q_{p\ell mn}^{R_a} \mathfrak{F}\left[A_\ell \left[\left(\frac{1}{2} \mathtt{R}_a \right) * (A_m A_n^*) \right] \right] + Q_{p\ell mn}^{R_b} \mathfrak{F}\left[A_\ell \left[\left(\frac{1}{2} \mathtt{R}_b \right) * (A_m A_n^*) \right] \right] \right\} \right\}$$

$$+ \frac{i\omega}{4} \sum_m Q_{pm}^I(\omega) P_{I,m}(z, \Omega), \tag{1.1}$$

which includes dispersion, as well as instantaneous electronic and delayed Raman nonlinearities. $A_p(z, t)$ is the electric field ($\sqrt{\text{W}}$) of mode p, whose Fourier Transform is $A_p(z, \Omega) = \mathfrak{F}[A_p(z, T)]$. The Fourier Transform is applied with respect to angular frequency $\Omega = \omega - \omega_0$, where ω_0 is the center angular frequency of the numerical frequency window required to cover the investigated physical phenomena. β_p is the propagation constant of the mode p, obtained with the dispersion formula in [1] for anti-resonant fiber platform or in [2] for capillaries; $\beta_{(0)}$ and $\beta_{(1)}$ are to reduce the propagating global-phase increment to facilitate simulations, $\beta_{(1)}$ is the inverse group velocity of the moving frame, which introduces the delayed time $T = t - \beta_{(1)} z$. $\chi_{\text{electronic}}^{(3)}(\omega)$ is the third-order nonlinear susceptibility of the electronic response (m^2/V^2); $\mathtt{R}_a(t)$ and $\mathtt{R}_b(t)$ are isotropic and anisotropic Raman response functions. $P_{I,m}(z, \Omega)$ is the photoionization-induced polarization. p, ℓ, m, and n are the eigenmode indices; $Q_{p\ell mn}^K$, $Q_{p\ell mn}^{R_a}$, and $Q_{p\ell mn}^{R_b}$ are overlap integrals of eigenmode fields $\vec{F}_j(\vec{r}_\perp)$ (1/m), where $\vec{r}_\perp = (x, y)$:

$$Q_{p\ell mn}^K = \frac{2}{3} Q_{p\ell mn}^{R_a} + \frac{1}{3} Q_{p\ell mn}^k \qquad , Q_{p\ell mn}^k = \frac{\int \left(\vec{F}_p^* \cdot \vec{F}_n^* \right) \left(\vec{F}_\ell \cdot \vec{F}_m \right) \mathrm{d}x\,\mathrm{d}y}{N_p N_\ell N_m N_n} \tag{1.2a}$$

$$Q_{p\ell mn}^{R_a} = \frac{\int \left(\vec{F}_p^* \cdot \vec{F}_\ell \right) \left(\vec{F}_m \cdot \vec{F}_n^* \right) \mathrm{d}x\,\mathrm{d}y}{N_p N_\ell N_m N_n} \tag{1.2b}$$

$$Q_{p\ell mn}^{R_b} = \frac{1}{2} \left[\frac{\int \left(\vec{F}_p^* \cdot \vec{F}_m \right) \left(\vec{F}_\ell \cdot \vec{F}_n^* \right) \mathrm{d}x\,\mathrm{d}y}{N_p N_\ell N_m N_n} + Q_{p\ell mn}^k \right] = \frac{1}{2} \left(Q_{p\ell mn}^{r_b} + Q_{p\ell mn}^k \right), \tag{1.2c}$$

where $\vec{F}_p$ is the pth spatial eigenmode ($\vec{F}_p \in \mathbb{R}$), with the normalization condition, $\int \vec{F}_m^* \vec{F}_n \, \mathrm{d}^2 x = \delta_{mn}$ and $N_p = \sqrt{\frac{\epsilon_0 n_{\mathrm{eff},p} c}{2}}$. n_{eff} and $n_{i,\mathrm{eff}}$ in each N_i ($i \in \{p, \ell, m, n\}$) is assumed to follow

$$\frac{\epsilon_0^2 n_{\mathrm{eff}}^2 c^2}{N_p N_\ell N_m N_n} = 4. \tag{1.3}$$

The Raman response functions are

$$\frac{1}{2}\mathsf{R}_a = R_a = R^{\mathrm{vib}} - 2R^{\mathrm{rot}} \tag{1.4a}$$

$$\frac{1}{2}\mathsf{R}_b = R_b = 6R^{\mathrm{rot}}, \tag{1.4b}$$

where

$$R^{\mathrm{vib}} = \Theta(t) N_g \frac{\left(\frac{\mathrm{d}\alpha}{\mathrm{d}\mathbb{Q}} \right)_0^2}{4\mu} e^{-\gamma_2^{\mathrm{vib}} t} \sum_J (2J+1) \rho_J^{(0)} \frac{1}{\omega_{1,J;0,J}} \sin\left(\omega_{1,J;0,J} t \right) \tag{1.5a}$$

$$R^{\mathrm{rot}} = \Theta(t) N_g \frac{(\triangle\alpha)^2}{60\hbar} e^{-\gamma_2^{\mathrm{rot}} t} \sum_J \left(\rho_J^{(0)} - \rho_{J+2}^{(0)} \right) \frac{(J+1)(J+2)}{(2J+3)} \sin\left(\omega_{0,J+2;0,J} t \right). \tag{1.5b}$$

$\Theta(t)$ is the Heaviside step function, $\left(\frac{\mathrm{d}\alpha}{\mathrm{d}\mathbb{Q}} \right)_0$ is the polarizability derivative at equilibrium, μ is the reduced mass of a molecule. γ_2^{vib} and γ_2^{rot} are dephasing rates of vibrational and rotational SRS, respectively. $\omega_{\nu_2,J_2;\nu_1,J_1} = \omega_{\nu_2,J_2} - \omega_{\nu_1,J_1}$ is the transition frequency of state (ν_1, J_1) to (ν_2, J_2). ν and J are the vibrational and rotational quantum numbers, respectively.

It is worth noting that the field A_p is defined as

$$\vec{\mathbb{E}}(\vec{x}, t) = \frac{1}{2} \left[\vec{\mathcal{E}}(\vec{x}, t) + \mathrm{c.c.} \right] \quad , \vec{\mathcal{E}} \text{ is the analytic signal of } \vec{\mathbb{E}}$$

$$= \sum_p \int \mathrm{d}\omega \frac{1}{2} \left\{ \frac{\vec{F}_p(x, y, \omega)}{N_p(\omega)} A_p(z, \omega) e^{i[\beta_p(\omega)z - \omega t]} + \mathrm{c.c.} \right\}. \tag{1.6}$$

This makes MATLAB "ifft" become the Fourier Transform and "fft" become the *inverse* Fourier Transform. The convolution theorem also becomes different with different conventions. In our package, we follow this convention [Eq. (1.6)]. For example, to see the spectrum, please use

```
c = 299792.458; % nm/ps
wavelength = c./f; % nm
Nt = size(field,1);
dt = t(2)-t(1); % ps
factor_correct_unit = (Nt*dt)^2/1e3; % to make the spectrum of the correct unit
    "nJ/THz"
                                % "/1e3" is to make pJ into nJ
spectrum = abs(fftshift(ifft(field),1)).^2*factor_correct_unit; % in frequency
    domain
```

Cornell University

Use "fftshift" to shift the spectrum from small frequency to large frequency. Note that it is not "ifftshift". They differ when the number of points is odd. To understand this, think about what the first data point is in "ifft(field):" it is the zero-frequency component, so we need to use "fftshift" with the frequency defining as

```
f = f0+(−Nt/2:Nt/2−1)'/(Nt*dt); % THz
```

About the photoionization term, we implement only the single-mode case at this moment.

$$
\begin{aligned}
\partial_z A(z,\Omega) = {} & i\left[\beta(\omega) - \left(\beta_{(0)} + \beta_{(1)}\Omega\right)\right] A(z,\Omega) \\
& + \frac{i\omega}{4\left[N(\omega)\right]^4 A_{\text{eff}}(\omega)} \left(\frac{3}{4}\epsilon_0\chi_{\text{electronic}}^{(3)}\mathfrak{F}\left[|A|^2 A\right] + \mathfrak{F}\left[A\left(R*|A|^2\right)\right]\right) \\
& - i\frac{e^2}{4\left[N(\omega)\right]^2 \omega m_e}\mathfrak{F}\left[n_e(T)A(T)\right] - \frac{E_b A_{\text{eff}}}{4}\mathfrak{F}\left[\frac{\partial n_e}{\partial T}\frac{A(T)}{|A(T)|^2}\right],
\end{aligned}
\tag{1.7}
$$

where $e = 1.602\,176\,63 \times 10^{-19}$ C is the electron charge, $m_e = 9.109\,383\,7 \times 10^{-31}$ kg is the electron mass, and E_b is the ionization energy. n_e is the free-electron density that satisfies

$$
\partial_t n_e = W\left(N_g - n_e\right) \approx W N_g
\tag{1.8}
$$

with W being the ionization rate and N_g being the total neutral-gas density, such that

$$
n_e(t) \approx N_g \int_{-\infty}^{t} W(\tau)\,\mathrm{d}\tau \quad, n_e(t \to -\infty) = 0
\tag{1.9}
$$

The ionization rate W can be calculated with the Perelomov-Popov-Terent'ev model.

For more details, please read our papers on soliton self-frequency shift [3], long-wave-infrared generation [4], and unified theory of Raman scattering. In particular, their supplements include all the details required for gas-based fiber simulations.

1.1   High-level understanding of this package

This package is designed for both single-spatial(transverse) mode or multi-spatial modes. Not only scalar but also polarized fields can be simulated, as well as Raman scattering. The package exhibits an adaptive control of the step size. In addition to CPU, highly parallelized cuda computation with a Nvidia GPU is implemented, which is strongly recommended for running with multimodes. In single-mode simulations, for sampling numbers less than approximately 2^{25}, they can still run faster with CPU than with GPU. The package uses "RK4IP" (Runge-Kutta in the interaction picture) for single mode [5, 6] and "MPA" (Massively Parallel Algorithm) for multimode [7].

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

Cornell University

Chapter 2

Use

Some examples of how to use the this package to simulate pulses can be found in the "Examples" folder. Generally, the workflow is as follows:

1. **Determine the number of discrete points N_t, time spacing $\mathrm{d}t$, and center frequency f_0.**

   They determine the time window and the frequency window. Because oftentimes the simulation covers a huge spectral window, the center frequency of the window cannot be at the pulse center frequency; otherwise, the edge of the frequency window can become negative. UPPE is based on *analytic signal* that includes only signals with positive frequencies. Because simulating with frequency window into the negative frequencies involves the interaction of these negative-frequency spectral components, it can generate a wrong result. It is thus important to shift the center frequency to maintain positiveness of the frequency window.

2. **Setup simulation parameters (three structure variables): sim, fiber, and gas.**

   "sim" includes simulation parameters such as whether to use GPU or not, whether to center the pulse temporally during propagation (the delay is saved for users to recover), what the moving speed of a time window is, and so on.
   "fiber" includes the fiber parameters such as fiber length and propagation constants of the involving modes.
   "gas" includes gas parameters such as gas pressure, gas type, and all the other relevant parameters of hollow-core fibers.

3. **Setup the initial condition which corresponds to input pulses.**

4. **Run the UPPE_propagate() function to start the pulse propagation.**

Chapter 3

Before I go deeply into details

3.1 Introduction

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

Below is how to call this function in general.

```
1  prop_output = UPPE_propagate(fiber, ...
2                               initial_condition, ...
3                               sim, ...
4                               gas)
```

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

fiber
> It contains the information of the fiber, such as β and S^R, etc.

initial_condition
> It contains the information of the input field.

Figure 3.1: Initial conditions

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

gas
> It contains the information required to run gas simulations, such as gas nonlinear refractive index and the information of its Raman response functions.

Chapter 4

Input arguments

Below, I use N_t as the number of time/frequency sampling points, N_m as the number of modes, and N_{sm} as the number of spatial modes. If there is no polarized mode, $N_m = N_{sm}$; otherwise, $N_m = 2N_{sm}$.

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 labelled in blue the parameters required all the time.

4.1 fiber

betas

> **betas** are column vectors of $\beta(\omega)$, which follows
>
> $$\begin{bmatrix} \beta_1 & \beta_2 & \cdots \end{bmatrix}, \tag{4.1}$$
>
> where each β_m is a column vector of the propagation constant of the mth mode. It's a function of frequencies, ordered from small to large. Don't take "ifftshift" for this variable. If the simulation is run with N_t time/frequency points, this β_m should have the length of N_t as well.

SR

> It's the overlap integral $S^R_{p\ell mn} = \int F_p F_\ell F_m F_n \, \mathrm{d}^2 x$ in scalar UPPE. Its dimension is N^4_{sm}.
>
> Since $N_p N_\ell N_m N_n = \frac{\epsilon_0^2 n_{\mathrm{eff}}^2 c^2}{4}$ is a constant value independent of modes, only the integral part needs to be calculated. Furthermore, all overlap integrals can be calculated from $S^R_{p\ell mn}$. For example, in scalar situations, $Q^K_{p\ell mn} = S^K / \left(N_p N_\ell N_m N_n \right) = S^R_{p\ell mn} / \left(N_p N_\ell N_m N_n \right)$ if the field is linearly polarized or $\frac{2}{3} S^R_{p\ell mn} / \left(N_p N_\ell N_m N_n \right)$ is it's circularly polarized. The unit is m^{-2}. It's more complicated with a polarized field, whose $Q^R_{p\ell mn}$ and $Q^K_{p\ell mn}$ can also be calculated from the $S^R_{p\ell mn}$. This will be done by UPPE_propagate() automatically if "sim.scalar=false." For details, check Chap.6.

L0

> This is the fiber length. The unit is meter.

4.2   initial_condition

dt

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

fields

     This is the input field amplitude under the time domain. It has the unit of $\sqrt{W}$. Its size is $N_t \times N_m$.

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

4.3   sim

Below are the most basic parameters for a simualtion.

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 beating, the size of the z-step cannot be too large.

     This "betas" is a 2×1 column vector.

$$\begin{bmatrix} \beta_{(0)} \\ \beta_{(1)} \end{bmatrix} \tag{4.2}$$

     The "betas" is found such that it minimizes the $\max\left(\left|\beta_1(\omega) - \left(\beta_{(0)} + \beta(1)\omega\right)\right|\right)$ of the 1st mode in the frequency window if this "sim.betas" is left empty (sim.betas=[]).

f0

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

midx

     The mode index. It's an integer array. It's 1 for scalar simulations.

X3

     It's the electronic third-order susceptibility of the gas.

save_period

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

Cornell University

4.3.1 Mode info

Here are the information for mode profiles.

mode_profiles.mode_profiles
> 2D mode profiles of each mode.

mode_profiles.norms
> This is the norms of the mode profiles.

mode_profiles.r

mode_profiles.dr

mode_profiles.dtheta

4.3.2 MPA

Here are the parameters if the simulation uses MPA step method [7]. All parameters are contained within a "sim.MPA" structure.

MPA.M
> This is the parallel extent for MPA. 1 is no parallelization. 5-20 is recommended; there are strongly diminishing returns after 5-10. 10 is recommended.

MPA.n_tot_max
> The maximum number of iterations for MPA. This doesn't really matter because if the step size is too large, the algorithm will diverge after a few iterations. 20 is a typical number for this.

MPA.n_tot_min
> The minimum number of iterations for MPA. Use 2 for this parameter.

MPA.tol
> The tolerance of convergence for MPA, which is related to the values of the average NRMSE between consecutive itertaions in MPA at which the step is considered converged.

4.3.3 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

> If the simulation is solved with "sim.scalar=true," the input field takes only the scalar fields, e.g.,

$$\begin{bmatrix} \text{mode 1} & \text{mode 2} & \text{mode 3} & \dots \end{bmatrix}.$$

Otherwise, the input field of each polarized mode needs to be specified in the order of

$$\left[\text{mode } 1_+ \quad \text{mode } 1_- \quad \text{mode } 2_+ \quad \text{mode } 2_- \quad \dots \right],$$

where (+,-) can be (x,y), (right-handed circular, left-handed circular), or any orthogonally polarized modes.

Based on whether to include polarization-mode coupling, S^R and S^K are automatically calculated to its polarized version by UPPE_propagate().

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)

4.3.4 Adaptive-step method

Here are the parameters for adaptive-step method. All parameters are contained within a "sim.adaptive_deltaZ" structure.

adaptive_deltaZ.threshold

The threshold of the adaptive-step method. It controls the accuracy of the simulation and determines whether to increase or decrease the step size. I typically use 10^{-8}.

adaptive_deltaZ.max_deltaZ

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

4.3.5 Algorithm to use

gpu_yes

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

Raman_model

0	ignore Raman effect
1	use Raman model for Raman-active gases

photoionization_model

Photoionization model is implemented currently only in single-mode scenarios.

0	ignore the photoionization effect
1	include the photoionization effect

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

Cornell University

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

num_photon_noise_per_bin

This controls the number of photon noise to include per spectral discretization bin. It's default to 1.

$$P_{\text{noise}} = hf / (N \triangle t), \tag{4.3}$$

whose relation depends on the convention of Fourier Transform. For convention of the laser field, Fourier Transform is defined as MATLAB's "ifft," which leads to the relation here. With a different convention, the multiplication factor might be different from $1/(N\triangle t)$.

cuda_dir_path

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

gpuDevice.Index

The GPU to use. It's typically 1 if the computer has only one GPU. MATLAB starts the index with 1.

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 UPPE shown on the progress bar. If not set (no "sim.progress_bar_name"), it uses a default empty string, ".

Cornell University

Chapter 5

Output arguments

fields

The $N_t \times N_m \times N_z$ output fields.

dt

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

z

This is the positions of each saved field.

deltaZ

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.

delta_permittivity

Raman-induced permittivity change $\triangle \epsilon$. Its size is $N_t \times N_m \times N_R \times N_z$, where N_R is the number of Raman types. This is implemented only in scalar simulations.

We have implemented a few gases whose N_R dimension follows:

$$H_2, N_2, O_2 \quad \left[\triangle \epsilon^{\text{vib}} \quad \triangle \epsilon^{\text{rot}}\right]$$

$$\text{air} \quad \left[\triangle \epsilon^{\text{vib (N}_2)} \quad \triangle \epsilon^{\text{rot (N}_2)} \quad \triangle \epsilon^{\text{vib (O}_2)} \quad \triangle \epsilon^{\text{rot (O}_2)}\right]$$

$$\text{CH}_4 \quad \left[\triangle \epsilon^{\text{vib}}\right]$$

$\triangle \epsilon^{\text{vib}}$ is calculated from Eq. (1.5a) of each gas, while $\triangle \epsilon^{\text{rot}}$ is calculated from Eq. (1.5b) for circularly polarized fields but its 4 times for linearly polarized fields. See our paper of a unified theory for details.

relative_Ne
   The ionized electron density ($1/m^3$).

Chapter 6

Polarization modes

If the simulation is solved with "sim.scalar=true", the input field takes only the scalar fields, e.g.,

$$\begin{bmatrix} \text{mode 1} & \text{mode 2} & \text{mode 3} & \dots \end{bmatrix}.$$

Otherwise, the input field of each polarized mode needs to be specified in the order of

$$\begin{bmatrix} \text{mode } 1_+ & \text{mode } 1_- & \text{mode } 2_+ & \text{mode } 2_- & \dots \end{bmatrix},$$

where (+,-) can be (x,y), (right-handed circular, left-handed circular), or any orthogonally polarized modes.

If the input β has a dimension of only the number of spatial modes, N_{sm}, I assume there's no significant influence from birefringence; thus, it's expanded into $2N_{sm}$ dimension with each i and j (polarization) modes being degenerate by UPPE_propagate(). For polarized fields,

$$S_{plmn}^R = \frac{\int \mathrm{d}x\,\mathrm{d}y \left[\mathbf{F}_p^* \cdot \mathbf{F}_l\right]\left[\mathbf{F}_m^* \cdot \mathbf{F}_n\right]}{\left[\left(\int \mathrm{d}x\,\mathrm{d}y|\mathbf{F}_p|^2\right)\left(\int \mathrm{d}x\,\mathrm{d}y|\mathbf{F}_l|^2\right)\left(\int \mathrm{d}x\,\mathrm{d}y|\mathbf{F}_m|^2\right)\left(\int \mathrm{d}x\,\mathrm{d}y|\mathbf{F}_n|^2\right)\right]^{1/2}} \tag{6.1}$$

$$S_{plmn}^K = \frac{2}{3}S_{plmn}^R + \frac{1}{3}\frac{\int \mathrm{d}x\,\mathrm{d}y \left[\mathbf{F}_p^* \cdot \mathbf{F}_n^*\right]\left[\mathbf{F}_m \cdot \mathbf{F}_l\right]}{\left[\left(\int \mathrm{d}x\,\mathrm{d}y|\mathbf{F}_p|^2\right)\left(\int \mathrm{d}x\,\mathrm{d}y|\mathbf{F}_l|^2\right)\left(\int \mathrm{d}x\,\mathrm{d}y|\mathbf{F}_m|^2\right)\left(\int \mathrm{d}x\,\mathrm{d}y|\mathbf{F}_n|^2\right)\right]^{1/2}} \tag{6.2}$$

Therefore, S_{plmn}^R isn't zero as (p,l) and (m,n) both have the same polarization, and we get four possibilities for (p,l,m,n), (0,0,0,0), (0,0,1,1), (1,1,0,0), and (1,1,1,1), with their values directly derived from the scalar S_{plmm}^R. For S_{plmn}^K, in addition to the permutations of S_{plmn}^R, we need to consider those from the fraction above which isn't zero as (p,l,m,n) is (0,0,0,0), (0,1,1,0), (1,0,0,1), and (1,1,1,1). Notice that some of them can add up with S_{plmn}^R while some of them can't, so the value has a prefactor of $1, \frac{2}{3}, \frac{1}{3}$.

The above generalization of the scalar S^R to polarized S^R, S^K needs each $\mathbf{F}_p$ to be either parallel or orthogonal to one another, so (i,j) has to be an orthogonal group in 2D, e.g., (x, y) or (σ_+, σ_-).

Chapter 7

Information and computation of gas and hollow-core fiber

This package implements two types of hollow-core fibers: anti-resonant fiber and capillary. Dispersion and mode-profile calculations of anti-resonant fiber is based on the poorman's model by Bache *et al.* [1], while that of capillary is based on the equation derived by Marcatili [2].

7.1   Gas info in simulations

To use the model of anti-resonant fiber, follow (the numeric values are chosen as examples. Modify them based on your needs.):

```matlab
%% Gas info
[fiber,sim] = load_default_UPPE_propagate(fiber,sim);

gas.core_radius = 15e-6; % um
gas.temperature = 288; % K
gas.pressure = 15*1.01325e5; % Pa
gas.wavelength_order = 6; % keep this 6 all the time
gas.mode_profile_wavelength = 1030e-9; % m
gas.gas_material = 'H2';
gas.delta = 300e-9; % m; wall thickness of anti-resonant fibers
gas.f_FEM = 2e-3; % loss factor (check poorman's paper for detail)
gas.fiber_type = 'AR_HC_PCF';
gas.xy_sampling = 101; % the number of sampling points for the mode profiles,
    which affects calculationo of "norms" and "SR"

[fiber,sim,gas] = gas_info(fiber,sim,gas,lambda*1e-9);
```

We have implemented a few gases (for "gas.gas_material"): H_2, N_2, O_2, air, Xe, Ar, Ne, He, Kr, and CH_4. Raman models have been implemented for Raman-active gases: H_2, N_2, O_2, air, and CH_4. Photoionization has been implemented for all gases except air.

To use the model of capillary, follow (the numeric values are chosen as examples. Modify them based on your needs.):

```matlab
%% Gas info
[fiber,sim] = load_default_UPPE_propagate(fiber,sim);

gas.core_radius = 250e-6; % m
```

```
5   gas.temperature = 300; % K
6   gas.pressure = 6.2*1.01325e5; % Pa
7   gas.wavelength_order = 6;
8   gas.mode_profile_wavelength = 1030e-9; % m
9   gas.gas_material = 'N2';
10  gas.fiber_type = 'no_coating';
11  gas.xy_sampling = 101; % the number of sampling points for the mode profiles,
        which affects calculationo of "norms" and "SR"
12
13  [fiber,sim,gas] = gas_info(fiber,sim,gas,lambda*1e-9);
```

In Marcatili's model, we can change the dielectric material of capillary. Based on this, we have implemented a few coatings, so "gas.fiber_type" has a few options:

'Ag_coating'	The capillary material is Ag
'no_coating'	The capillary material is silica
'MWLW_coating'	The capillary material is also silica, but the final "waveguide" loss is later entirely neglected.
'waveguide_loss_only'	The capillary material is silica whose loss is initially neglected. The result includes waveguide loss without exacerbated by silica loss.

7.2   Hollow-core fiber

It's sometimes helpful to know the fiber parameters separately from simulations. For example, the simulations compute based on propagation constants that are functions of frequency, but we sometimes would like to know the dispersion, $\frac{\mathrm{d}^2\beta}{\mathrm{d}\omega^2}$.

To compute these parameters, use the scripts with the name, "solve_for_EH_...()" in the folder "HCF/."

Use "calc_betaN()" to compute for dispersion.

Cornell University

Chapter 8

Simulations with multiple spatial modes

This package allows users to simulate multimode pulse propagation. This is enabled by setting "sim.midx":

```
1  sim.midx = 1:6; % mode 1 to 6 (LP01,LP02,...,LP06) are included for example
```

The order of mode indices follows the decreasing magnitudes of propagation constants. By default, this package is restricted to the first six circularly symmetric modes ($LP_{01} \ldots LP_{06}$). The restriction can be easily modified in the HCF functions:

1. HCF/helper_functions/solve_for_EH_Ag_coating_beta_func.m

2. HCF/helper_functions/solve_for_EH_AR_HC_PCF_beta_func.m

3. HCF/helper_functions/solve_for_EH_MWLW_coating_beta_func.m

4. HCF/helper_functions/solve_for_EH_no_coating_beta_func.m

They have the following lines

```
1  % (n,m) modes to solve for
2  user_midx = [1,4,9,17,28,40]; % a maximum of 6 circular symmetric modes included
        here
3  user_midx = user_midx(sim.midx);
4  num_modes = length(user_midx);
```

Feel free to modify "user_midx" to include other modes. For example, user_midx=2 and 3 corresponds to LP_{11a} and LP_{11b}. Readers can plot the fields out with "solve_for_EH_...()" in the folder "HCF/" to see if they pick those they want. They have the same lines that can be modified as below:

```
1  % (n,m) modes to solve for
2  % [1,4,9,17,28,40] are the first six radial EH0m modes
3  user_midx = 1;%[1,4,9,17,28,40];
4  num_modes = length(user_midx);
```

The examples of multimode simulations are in "Examples/validation of the code/Extreme Raman redshift (Carpeggiani's Optica, 2020) (use capillary)," which indludes the first five circularly symmetric modes.

Chapter 9

Dechirper and Stretcher

This chapter gives the phase accumulated after propagating through a grating dechirper or a stretcher. There are four configurations: Treacy [8], prism [9, 10], Offner [11], and Martinez [12, 13]. Although they can be found in papers, the studies in them might deviate people's attention. Here, I focus only on showing the full phase accumulation $\phi(\omega) = k\ell(\omega) + \phi_g(\omega)$, where k is the wave vector, ℓ is the path length, and ϕ_g is the grating phase. It varies with the light angular frequency ω and is implemented numerically to dechirp or stretch a pulse. If readers are interested in their group delay dispersion, $\frac{\mathrm{d}^2\phi}{\mathrm{d}\omega^2}$, or third-order dispersion, $\frac{\mathrm{d}^3\phi}{\mathrm{d}\omega^3}$, please refer to their papers. They are widely used in stretching and dechirping pulses mentioned throughout this thesis. Typically, the grating is designed to be a blazed grating worked under the Littrow configuration whose diffraction order $m = -1$ is only considered.

9.1 Treacy type

In this section, both reflective and transmissive Treacy grating dechirpers/stretchers are introduced. They can add negative chirp (corresponding to anomalous dispersion) to a pulse.

9.1.1 Reflective Treacy type

The single-pass optical path length (Fig. 9.1) is

$$
\begin{aligned}
\ell = \ell_1 + \ell_2 &= d \sec\theta_{\text{out}} \left[1 + \cos(\theta_{\text{in}} + \theta_{\text{out}}) \right] \\
&= d \sec\theta_{\text{out}} \left(1 + \cos\theta_{\text{in}} \cos\theta_{\text{out}} - \sin\theta_{\text{in}} \sin\theta_{\text{out}} \right) \\
&= d \left(\sec\theta_{\text{out}} + \cos\theta_{\text{in}} - \sin\theta_{\text{in}} \tan\theta_{\text{out}} \right),
\end{aligned} \tag{9.1}
$$

where

$$
\Lambda \left(\sin\theta_{\text{out}} - \sin\theta_{\text{in}} \right) = m\lambda, \tag{9.2}
$$

Λ is the grating line spacing, and m is the diffraction order.

For the grating, the accumulated phase considers not only the geometric path length but also the grating phase. The first grating does not add any grating phase because all spectral components are diffracted at the same position. However, the second one imposes a grating phase because different spectral components are now diffracted at different positions of the grating. To calculate the grating phase, only the relative position matters.
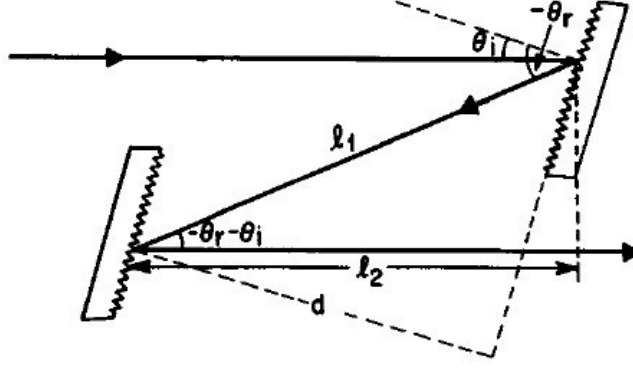
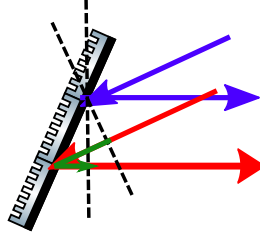Figure 9.1: Reflective grating dechirper/stretcher [14].

Figure 9.2: The grating phase of a Treacy-type reflective grating dechirper/stretcher.

Because of diffraction after the first grating, the red light propagates farther than the blue light (Fig. 9.2). This extra propagation phase (from the green line in Fig. 9.2) needs to be canceled so that it is directed horizontally after the grating. As a result, the grating phase follows

$$\phi_g(x) = \pi + m\frac{x}{\Lambda}2\pi. \tag{9.3}$$

π is due to the reflection and will be taken out if we use transmissive gratings. x is the relative position on the grating plane, which in the case of a reflective grating dechirper/stretcher, becomes $d\tan(-\theta_{out})$. Therefore, the single-pass phase $\phi_{single-pass}$ becomes

$$
\begin{aligned}
\phi_{single-pass} &= k\ell + \phi_g \\
&= k\ell + \left[\pi + m\frac{2\pi d\tan(-\theta_{out})}{\Lambda}\right] \\
&= kd\left(\sec\theta_{out} + \cos\theta_{in} - \sin\theta_{in}\tan\theta_{out}\right) + \left(\pi - m\frac{2\pi d\tan\theta_{out}}{\Lambda}\right) \\
&= kd\left(\sec\theta_{out} + \cos\theta_{in}\right) + \pi - d\tan\theta_{out}\left(k\sin\theta_{in} + m\frac{2\pi}{\Lambda}\right) \\
&= kd\left(\sec\theta_{out} + \cos\theta_{in}\right) + \pi - kd\tan\theta_{out}\sin\theta_{out} \\
&= kd\left(\cos\theta_{out} + \cos\theta_{in}\right) + \pi. \tag{9.4}
\end{aligned}
$$

To avoid spatial chirp, a mirror is introduced after the first pass of the grating pair so that all spectral components propagate back to where they are, eliminating the spatial chirp. Due to this extra reflecting propagation, the total phase is two times larger, that is,

$$\phi_{double-pass} = 2\left[kd\left(\cos\theta_{out} + \cos\theta_{in}\right) + \pi\right]. \tag{9.5}$$

Cornell University

9.1.2   Transmissive Treacy type

The derivation follows the reflective grating dechirper/stretcher discussed previously. The total optical path length (Fig. 9.3) is

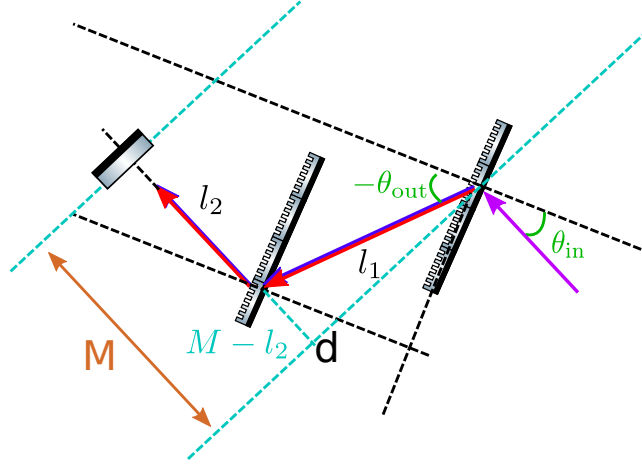$$\ell = \ell_1 + \ell_2 = \ell_1 + \left[M - (M - \ell_2) \right]. \tag{9.6}$$

Figure 9.3: Transmissive grating dechirper/stretcher.

Since M is independent of frequency, I keep only the parameters relevant in pulse dechirping/stretching. Hence, ℓ becomes

$$\begin{aligned}
\ell &= \ell_1 + \left[M - (M - \ell_2) \right] \\
&\to \ell_1 - (M - \ell_2) \\
&= \ell_1 - \ell_1 \sin\left(\frac{\pi}{2} - \theta_{\text{in}} + \theta_{\text{out}} \right) \\
&= \ell_1 - \ell_1 \cos\left(\theta_{\text{in}} - \theta_{\text{out}} \right) \\
&= d \sec\theta_{\text{out}} \left[1 - \cos(\theta_{\text{in}} - \theta_{\text{out}}) \right] \\
&= d \sec\theta_{\text{out}} \left(1 - \cos\theta_{\text{in}} \cos\theta_{\text{out}} - \sin\theta_{\text{in}} \sin\theta_{\text{out}} \right) \\
&= d \left(\sec\theta_{\text{out}} - \cos\theta_{\text{in}} - \sin\theta_{\text{in}} \tan\theta_{\text{out}} \right).
\end{aligned} \tag{9.7}$$

Similar to the reflective gratings, the grating phase from the transmissive grating is added to calculate the single-pass total phase.

$$\begin{aligned}
\phi_{\text{single-pass}} &= k\ell + \phi_g = k\ell + m \frac{2\pi d \tan(-\theta_{\text{out}})}{\Lambda} \\
&= kd \left(\cos\theta_{\text{out}} - \cos\theta_{\text{in}} \right).
\end{aligned} \tag{9.8}$$

To eliminate the spatial chirp, the total phase after introducing a mirror becomes

$$\phi_{\text{double-pass}} = 2 \left[kd \left(\cos\theta_{\text{out}} - \cos\theta_{\text{in}} \right) \right]. \tag{9.9}$$

Cornell University

9.1.3 Group delay dispersion of Treacy dechirper/stretcher

Here, group delay dispersion (GDD) of both reflective and transmissive grating dechirpers/stretchers are shown. They are used to calculate an initial guess of the grating separation, d, for the subsequent grating-pair optimization schemes.

$$\frac{\mathrm{d}^2\phi}{\mathrm{d}\omega^2} = -\frac{m^2\lambda^3 d}{2\pi c^2\Lambda^2}\left[1-\left(-m\frac{\lambda}{\Lambda}-\sin\theta_{\text{in}}\right)^2\right]^{-3/2}, \tag{9.10}$$

where m is the diffraction order and is typically -1, λ is the pulse center wavelength, d is the grating separation, and Λ is the grating line spacing.

9.2 Prism type

As grating pair, prism pair that disperses color can be used as a dechirper or stretcher. It operates as in Fig. 9.4. The light should hit the prism as close to the apex as possible.
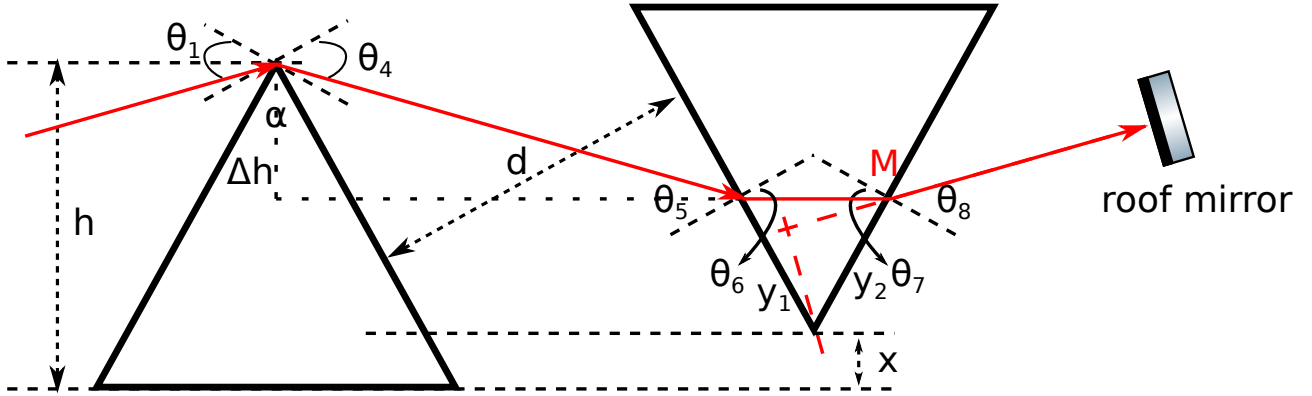
Figure 9.4: Prism dechirper/stretcher.

In principle, the α angle can be varied to tune the dispersion properties of a prism dechirper/stretcher. In practice, however, the geometry is chosen such that the incident and refracted beam have the same angle at the central wavelength of the spectrum to be dechirped/stretched. This configuration is known as the "angle of minimum deviation," and is easier to align than arbitrary angles (Fig. 9.5). In this configuration, the angles of refraction through the prism are symmetric, that is in Fig. 9.5,

$$\theta_1 = \theta_4 \tag{9.11a}$$

$$\theta_2 = \theta_3 = \frac{\alpha}{2}. \tag{9.11b}$$

To calculate the phase added to the pulse, the incident angle needs to be determined so that the beam of its center wavelength λ_0 enters the configuration of "angle of minimum deviation:"

$$\sin\theta_{\text{in}} = n(\lambda_0)\sin\theta_2, \quad \text{with } \theta_2 = \frac{\alpha}{2}$$

$$\Rightarrow \theta_{\text{in}} = \sin^{-1}\left(n(\lambda_0)\sin\frac{\alpha}{2}\right). \tag{9.12}$$
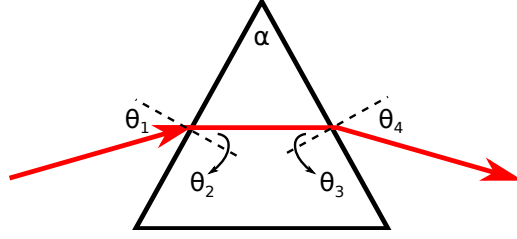
Cornell University

Figure 9.5: Prism operation with a minimum beam deviation.

Next, we calculate each angle in Fig. 9.5:

$$n(\lambda) \sin \theta_2 = \sin \theta_{\text{in}} \tag{9.13a}$$

$$\Rightarrow \theta_3 = \alpha - \theta_2 \tag{9.13b}$$

$$\Rightarrow \sin \theta_4 = n(\lambda) \sin \theta_3. \tag{9.13c}$$

With calculated θ_4 for each wavelength, we subsequently obtain

$$\theta_5 = \theta_4 \tag{9.14a}$$

$$\Rightarrow n(\lambda) \sin \theta_6 = \sin \theta_5 \tag{9.14b}$$

$$\Rightarrow \theta_7 = \alpha - \theta_6 \tag{9.14c}$$

$$\Rightarrow \sin \theta_8 = n(\lambda) \sin \theta_7. \tag{9.14d}$$

These lead to

$$\theta_5 = \theta_4 \tag{9.15a}$$

$$\theta_6 = \theta_3 \tag{9.15b}$$

$$\theta_7 = \theta_2 \tag{9.15c}$$

$$\theta_8 = \theta_1 = \theta_{\text{in}}. \tag{9.15d}$$

Eq. (9.15d) shows that all the colors exit the second prism with the same angle such that they can all be reflected back to their original paths after the roof mirror.

To find the path lengths after entering the second prism, we need to calculate the change of height of the beam when hitting the second prism. It is $\triangle h = \ell_s \cos\left(\frac{\pi}{2} - \theta_4 + \frac{\alpha}{2}\right) = \ell_s \sin\left(\theta_4 - \frac{\alpha}{2}\right)$, where the path length between two prisms is $\ell_s = d \sec(\theta_4)$, leading to $\triangle h = d\left(\tan \theta_4 \cos \frac{\alpha}{2} - \sin \frac{\alpha}{2}\right)$. The distance between where the beam hitting the prism and the apex of the second prism is $y_1 = (h - \triangle h - x) \sec \frac{\alpha}{2}$. It allows us to find the path length to travel inside the second prism ℓ_p and the length on the hypotenuse y_2 through the following relation:

$$\frac{y_1}{\sin\left(\frac{\pi}{2} - \theta_7\right)} = \frac{\ell_p}{\sin \alpha} = \frac{y_2}{\sin\left(\frac{\pi}{2} - \theta_6\right)}. \tag{9.16}$$

And thus

$$\ell_p = \frac{y_1 \sin \alpha}{\cos \theta_2} \tag{9.17a}$$

$$y_2 = \frac{y_1 \cos \theta_3}{\cos \theta_2} \tag{9.17b}$$

Cornell University

y_2 contributes to the change of path length to the roof mirror by $\ell_M = M - y_2 \cos\left(\frac{\pi}{2} - \theta_8\right) = M - y_2 \sin\theta_{\text{in}} \sim -y_2 \sin\theta_{\text{in}}$.

In conclusion, the path lengths to consider are

$$\ell_s = d \sec\theta_4 \tag{9.18a}$$

$$\ell_p = \frac{y_1 \sin\alpha}{\cos\theta_2} = \frac{(h - \triangle h - x)\sec\frac{\alpha}{2}\sin\alpha}{\cos\theta_2} = \frac{2(h - \triangle h - x)\sin\frac{\alpha}{2}}{\cos\theta_2} \tag{9.18b}$$

$$\ell_M = -y_2 \sin\theta_{\text{in}} = -\frac{y_1 \cos\theta_3}{\cos\theta_2}\sin\theta_{\text{in}} = -\frac{(h - \triangle h - x)\cos\theta_3}{\cos\frac{\alpha}{2}\cos\theta_2}\sin\theta_{\text{in}} \tag{9.18c}$$

$$= -\frac{(h - \triangle h - x)(\cos\alpha + \sin\alpha\tan\theta_2)}{\cos\frac{\alpha}{2}}\sin\theta_{\text{in}} \tag{9.18d}$$

As a result, the total phase of this prism dechirper/stretcher is

$$\phi_{\text{double-pass}} = 2k\left(\ell_s + \ell_p + \ell_M\right). \tag{9.19}$$

9.2.1   Group delay dispersion of prism dechirper/stretcher

From [10], the path length that determines the GDD added to the pulse is

$$P = 2\ell\cos\beta, \tag{9.20}$$

where ℓ is the length between apexes of two prisms, and β is the angle of the beam with respect to the line connecting two apexes (Fig. 9.6). The corresponding GDD is

$$\begin{aligned}
\frac{\mathrm{d}^2\phi}{\mathrm{d}\omega^2} &= \frac{\lambda^3}{2\pi c^2}\frac{\mathrm{d}^2 P(\lambda)}{\mathrm{d}\lambda^2} \\
&= \frac{\lambda^3}{2\pi c^2}4\ell\left\{\left[\frac{\mathrm{d}^2 n}{\mathrm{d}\lambda^2} + \left(2n - \frac{1}{n^3}\right)\left(\frac{\mathrm{d}n}{\mathrm{d}\lambda}\right)^2\right]\sin\beta - 2\left(\frac{\mathrm{d}n}{\mathrm{d}\lambda}\right)^2\cos\beta\right\},
\end{aligned} \tag{9.21}$$

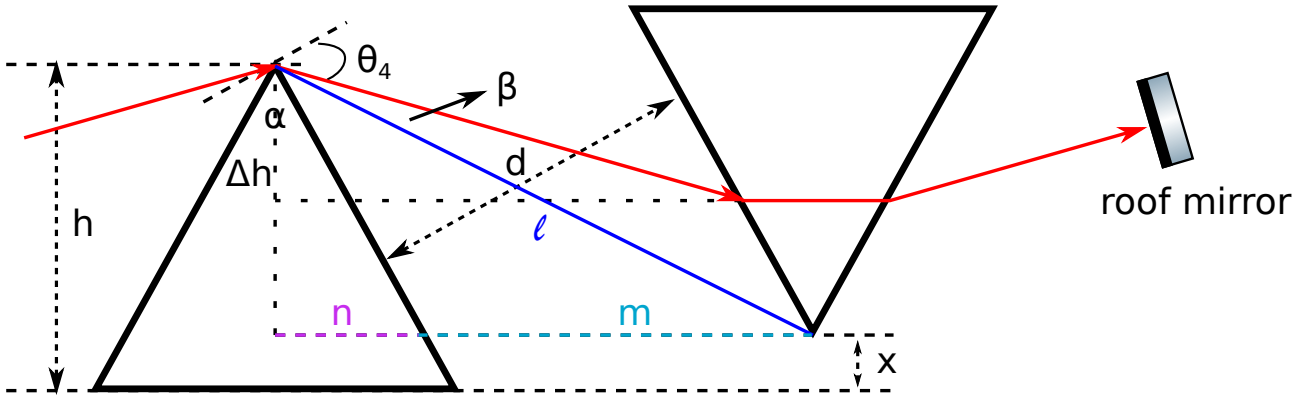where n is the refractive index of the prism material.

Figure 9.6: Schematic for calculating the prism GDD.

To find the value of Eq. (9.21), we need to determine ℓ and β that depend on the prism separation d, pulse bluest wavelength λ_b, and its center wavelength λ_0. In prism operations, the blue edge of the light is, in principle, put near the apex of the second prism, which leads to

$$\ell = \ell_s(\lambda_b) = d\sec\theta_4(\lambda_b) \tag{9.22a}$$

$$\beta = \theta_4(\lambda_b) - \theta_4(\lambda_0). \tag{9.22b}$$

Finally, we obtain

$$\frac{\mathrm{d}^2\phi}{\mathrm{d}\omega^2}(\lambda_0) = \frac{\lambda_0^3}{2\pi c^2}\frac{\mathrm{d}^2 P(\lambda_0)}{\mathrm{d}\lambda^2}$$

$$= \frac{\lambda_0^3}{2\pi c^2}4d\sec\theta_4(\lambda_b)$$

$$\times \left\{ \left[\frac{\mathrm{d}^2 n}{\mathrm{d}\lambda^2}(\lambda_0) + \left(2n(\lambda_0) - \frac{1}{\left(n(\lambda_0)\right)^3} \right)\left(\frac{\mathrm{d}n}{\mathrm{d}\lambda}(\lambda_0) \right)^2 \right] \sin\beta - 2\left(\frac{\mathrm{d}n}{\mathrm{d}\lambda}(\lambda_0) \right)^2 \cos\beta \right\}.$$
$$\tag{9.23}$$

9.3 Martinez type

Unlike the Treacy and prism types that add only negative chirp, Martinez type can add an arbitrary sign of chirp based on the relationship between the focal length f, and the distance ℓ, between the grating and the lens (Fig. 9.7). It adds positive chirp (corresponding to normal dispersion) when $\ell < f$ and negative chirp (corresponding to anomalous dispersion) when $\ell > f$; nothing happens when $\ell = f$ which is simply a 4f-telescope. It is often used as a pulse stretcher for chirped pulse amplification while a Treacy type is for latter pulse dechirper to cancel the chirp added by the Martinez stretcher.
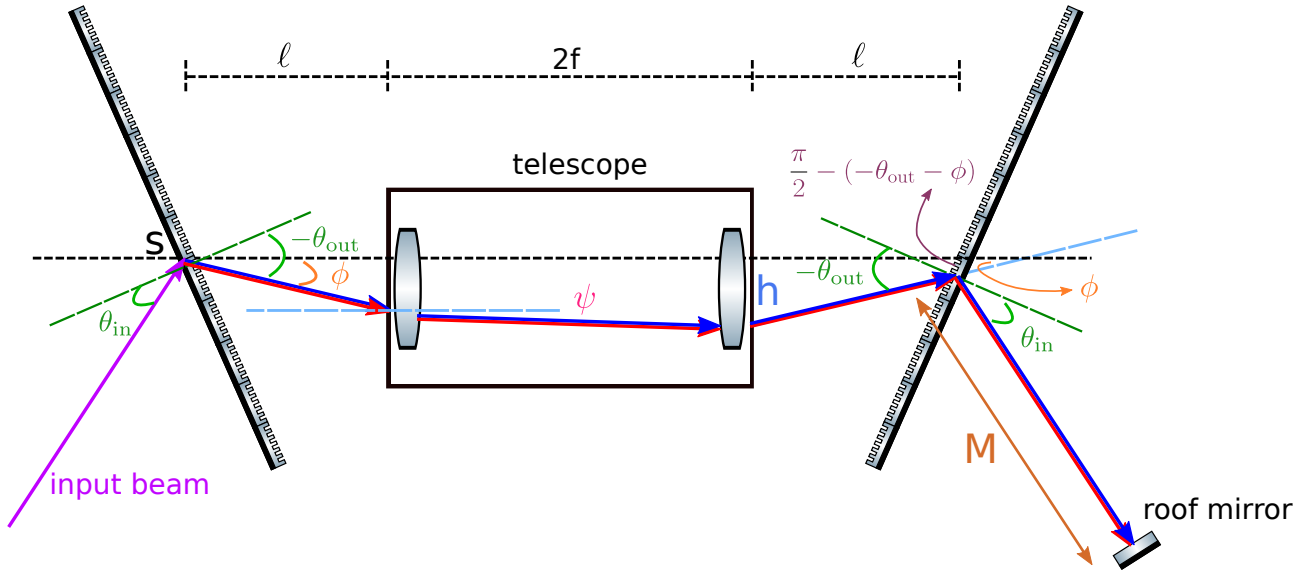

Figure 9.7: Martinez dechirper/stretcher.

There are four path lengths required in the phase calculation: ℓ_1, the first grating to the first lens; ℓ_2, the distance between two lenses; ℓ_3, the second lens to the second grating; ℓ_4, the second grating to the roof mirror for a second pass of the entire system to cancel the introduced spatial chirp after a single pass.

To calculate the first path length $\ell_1 = \ell\sec\phi$, we need to calculate the relative diffraction angle ϕ. Because the alignment of the telescope is determined by the center wavelength of the

Cornell University

input signal, we need to first determine the "center" diffraction angle to calculate ϕ. From here, we can see that the actual value of ϕ can be quite arbitrary.

To calculate the second path length $\ell_2 = 2f \sec \psi$, we calculate ψ with ABCD matrices.

$$T_{\text{lens}} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{f} & 1 \end{bmatrix} \begin{bmatrix} 1 & \ell \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \ell \\ -\frac{1}{f} & -\frac{\ell}{f} + 1 \end{bmatrix} \tag{9.24a}$$

$$T_{\text{lens}} \begin{bmatrix} 0 \\ \phi \end{bmatrix} = \begin{bmatrix} \ell\phi \\ \left(1 - \frac{\ell}{f}\right)\phi \end{bmatrix} = \begin{bmatrix} \ell \tan \phi \\ \left[1 - \tan^{-1}\left(\frac{\ell}{f}\right)\right]\phi \end{bmatrix}, \tag{9.24b}$$

which leads to

$$\psi = \left(1 - \frac{\ell}{f}\right)\phi = \left[1 - \tan^{-1}\left(\frac{\ell}{f}\right)\right]\phi, \ \tan^{-1} \text{ is more accurate} \tag{9.25}$$

Before we calculate the third path length ℓ_3, we are interested in the light passing through the telescope.

$$T = \begin{bmatrix} 1 & 0 \\ -\frac{1}{f} & 1 \end{bmatrix} \begin{bmatrix} 1 & 2f \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{1}{f} & 1 \end{bmatrix} \begin{bmatrix} 1 & \ell \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 2f - \ell \\ 0 & -1 \end{bmatrix} \tag{9.26a}$$

$$\Rightarrow \quad T \begin{bmatrix} 0 \\ \phi \end{bmatrix} = \begin{bmatrix} (2f - \ell)\phi \\ -\phi \end{bmatrix} \tag{9.26b}$$

This shows that the light maintains the same output angle as the input angle but with a vertical offset $h = (2f - \ell)\phi = (2f - \ell)\tan \phi$.

To calculate the third path length $\ell_3 = h \csc \phi - x$, we need to calculate x (Fig. 9.8).
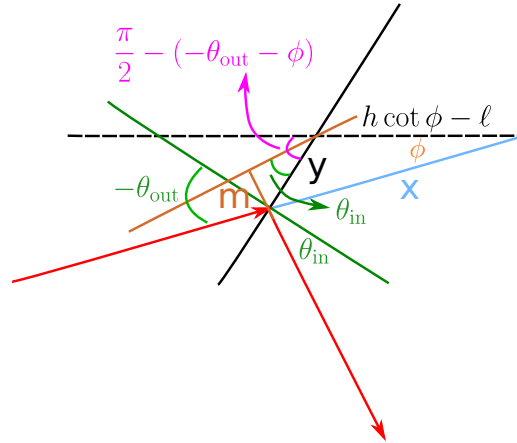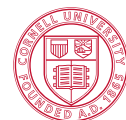

Figure 9.8: Diagram of a Martinez dechirper/stretcher to calculate the propagation length.

$$\frac{h \cot \phi - \ell}{\sin\left(\frac{\pi}{2} - (-\theta_{\text{out}} - \phi) - \phi\right)} = \frac{x}{\sin\left(\frac{\pi}{2} - (-\theta_{\text{out}} - \phi)\right)} = \frac{y}{\sin \phi}$$

$$\Rightarrow \quad \frac{h \cot \phi - \ell}{\cos \theta_{\text{out}}} = \frac{x}{\cos(\theta_{\text{out}} + \phi)} = \frac{y}{\sin \phi} \tag{9.27}$$

Cornell University

However, $h \cot \phi$ term can create NaN ("not a number" in MATLAB) during computation. When $\phi = 0$, $h = 0$; this term becomes $0 \times \infty$. To avoid the ambiguity, it is preferable to use $h \cot \phi = 2f - \ell$. Thus, the relation above becomes

$$\frac{2(f - \ell)}{\cos \theta_{\text{out}}} = \frac{x}{\cos(\theta_{\text{out}} + \phi)} = \frac{y}{\sin \phi}, \tag{9.28}$$

which gives

$$x = \frac{2(f - \ell)}{\cos \theta_{\text{out}}} \cos(\theta_{\text{out}} + \phi) \tag{9.29a}$$

$$y = \frac{2(f - \ell)}{\cos \theta_{\text{out}}} \sin \phi. \tag{9.29b}$$

The last path length is $\ell_4 = M - m$, where $m = y \sin \theta_{\text{in}}$.

Therefore, the single-pass path length is

$$\ell_{\text{single-pass}} = \ell \sec \phi + 2f \sec \psi + h \csc \phi - x + M - y \sin \theta_{\text{in}}$$
$$\to \ell \sec \phi + 2f \sec \psi + h \csc \phi - x - y \sin \theta_{\text{in}}. \tag{9.30}$$

Similarly, there is a grating phase (Fig. 9.9)

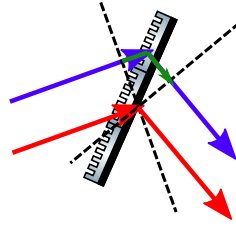$$\phi_g = -m \frac{y}{\Lambda} 2\pi. \tag{9.31}$$

Figure 9.9: The grating phase of an Martinez-type transmissive grating dechirper/stertcher.

Besides the above obvious optical path length and grating phase, there is another phase required to be taken into account: the lens phase, or the lens-thickness optical path length. This explains why the light direction changes after it passes through a lens with Huygens principle. It adds a phase to the light incident on different positions of a lens and thus bends the light, similar to a grating. Since the light from the focus travels in collimation (having a flat phase front) after a lens, the lens path length can be calculated as

$$\text{The 1st lens: } \ell_{\text{lens 1}} = -\sqrt{h_{\text{lens 1}}^2 + f^2} = -\sqrt{(\ell \tan \phi)^2 + f^2} \tag{9.32a}$$

$$\text{The 2nd lens: } \ell_{\text{lens 2}} = -\sqrt{h_{\text{lens 2}}^2 + f^2} = -\sqrt{h^2 + f^2}. \tag{9.32b}$$

$$\ell_{\text{lens}} = \ell_{\text{lens 1}} + \ell_{\text{lens 2}} = -\sqrt{(\ell \tan \phi)^2 + f^2} - \sqrt{h^2 + f^2}. \tag{9.33}$$

Thus, the single-pass total phase is

$$\phi_{\text{single-pass}} = k\ell_{\text{single-pass}} + k\ell_{\text{lens}} + \phi_g. \tag{9.34}$$

To eliminate spatial chirp, two passes are required. Finally, the double-pass total phase is

$$\phi_{\text{double-pass}} = 2 \left(k\ell_{\text{single-pass}} + k\ell_{\text{lens}} + \phi_g \right). \tag{9.35}$$

Cornell University

9.4 Offner type

A typical Martinez stretcher relies on a telescope (Fig. 9.7) but these refractive components can introduce aberration for broadband pulses [15]; therefore, Offner stretcher is preferred due to the use of all reflective optical components (Fig. 9.10).
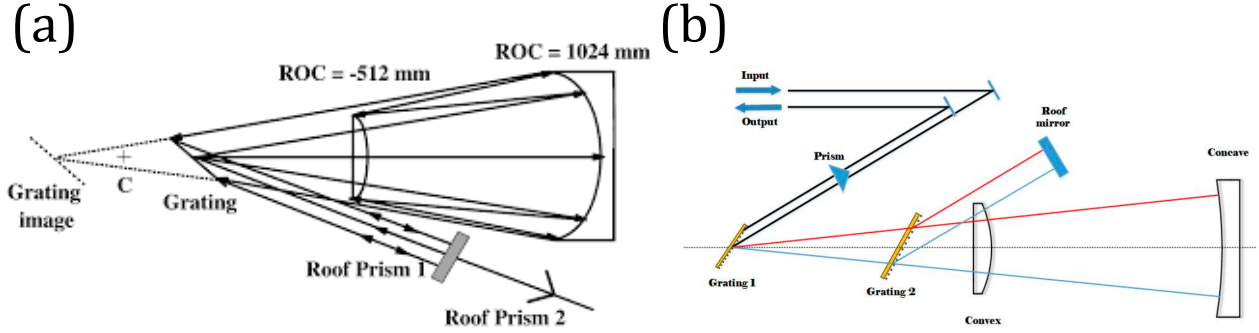

Figure 9.10: (a) Single-grating [15] and (b) double-grating reflective Offner dechirpers/stretchers [16].

9.4.1 Transmissive single-grating Offner type

I first calculate the accumulated phase of an Offner stretcher with a single transmissive grating (Fig. 9.11). There are two concentric convex and concave mirrors, and the transmissive grating is slightly deviated from the center of circles, which introduces abberation.

Suppose the deviation of the transmissive grating from the spherical center is s, and the convex and concave radius of curvature are R and $2R$. From Fig. 9.11 and 9.12, we have

$$\Lambda\left(\sin\theta_{\text{out}} - \sin\theta_{\text{in}}\right) = m\lambda \tag{9.36a}$$

$$\frac{2R}{\sin\left(\theta_{\text{in}} - \theta_{\text{out}} - \frac{\pi}{2}\right)} = \frac{s}{\sin\theta} = \frac{\ell_1}{\sin\phi} \tag{9.36b}$$

$$\frac{R}{\sin\theta} = \frac{2R}{\sin\psi} = \frac{\ell_2}{\sin\left(\psi - \theta\right)} \tag{9.36c}$$

$$\theta + \phi = \theta_{\text{in}} - \theta_{\text{out}} - \frac{\pi}{2}. \tag{9.36d}$$

Note that if $\theta_{\text{in}} - \theta_{\text{out}} - \frac{\pi}{2} < 0$, the diffracted beam goes to the upper-half plane, instead of going downward to the lower-half plane as in Fig. 9.11. Some angles may thus become negative.

To calculate x (Fig. 9.12), we need the following relation,

$$\frac{2\left[\ell_2\sin\psi - \ell_1\sin(\psi - 2\theta)\right]}{\sin\left(2\theta_{\text{in}} - \theta_{\text{out}} - \frac{\pi}{2} - 2\left[(\psi - \theta) + \phi\right]\right)} = \frac{x}{\sin\left(\left[(\psi - \theta) + \phi\right] + \left(\frac{\pi}{2} - \theta_{\text{in}}\right)\right)} \tag{9.37a}$$

$$= \frac{y}{\sin\left(\theta_{\text{in}} - \theta_{\text{out}} - \left[(\psi - \theta) + \phi\right]\right)}. \tag{9.37b}$$

Cornell University
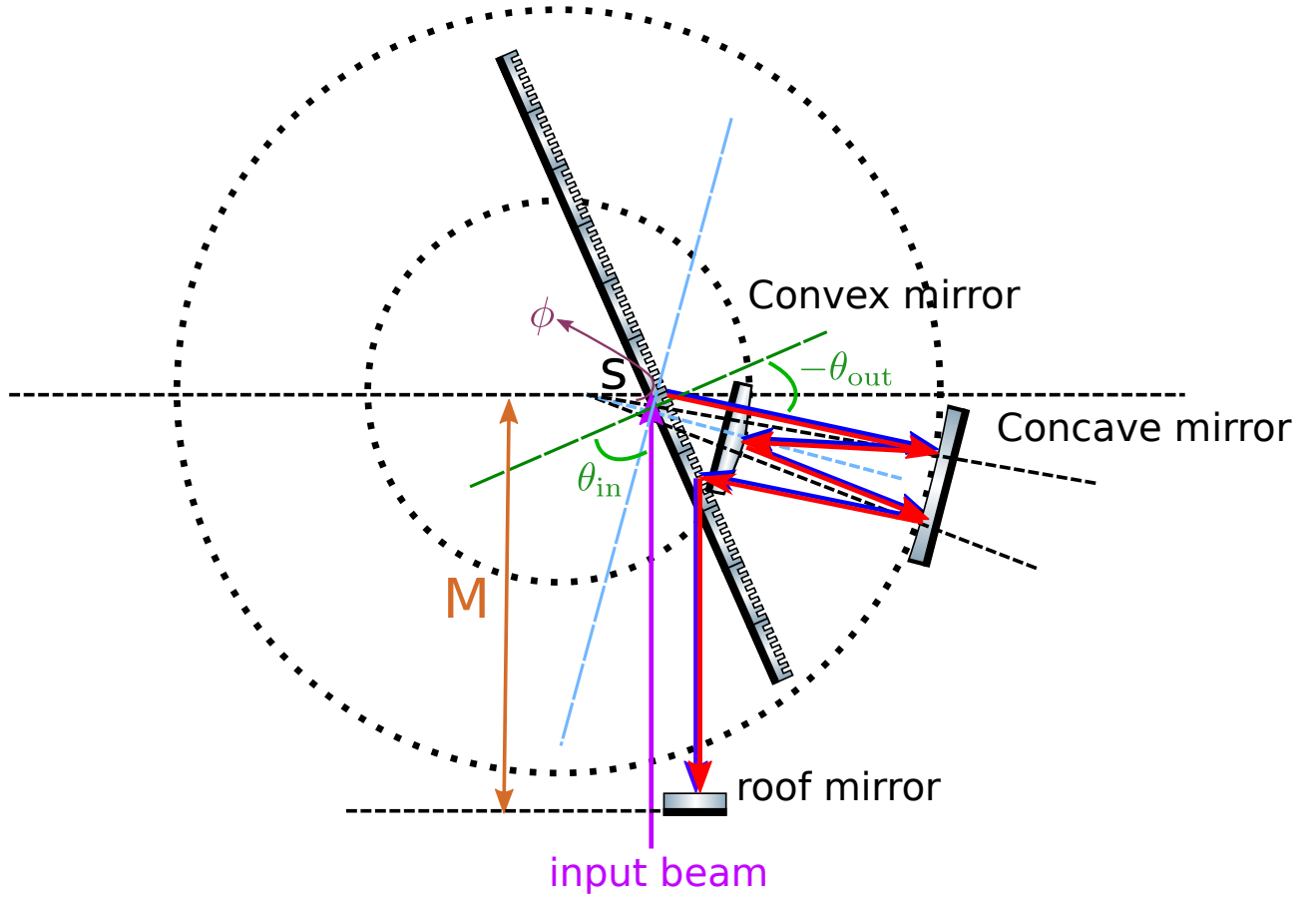
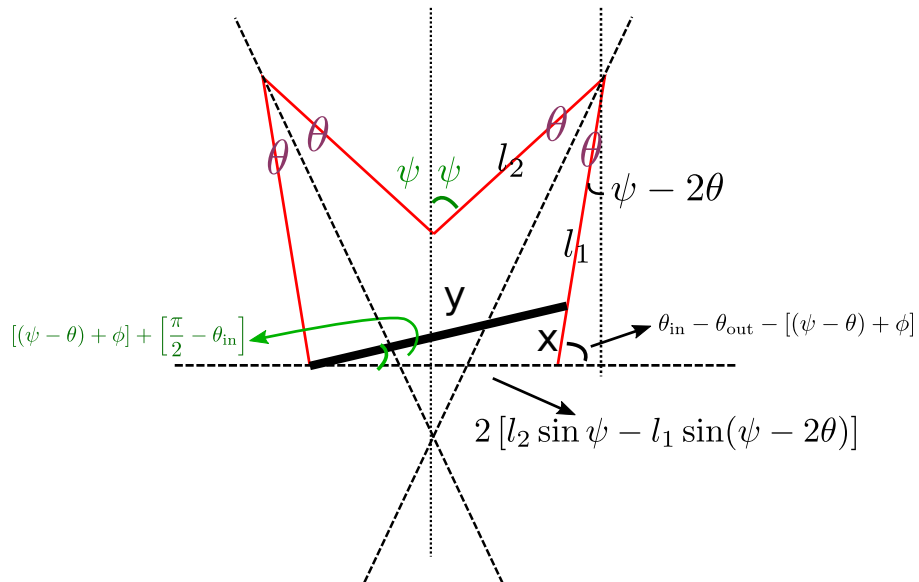Figure 9.11: Diagram of a single-grating transmissive Offner dechirper/stretcher.

Figure 9.12: Diagram of an Offner dechirper/stretcher to calculate the propagation length.

Cornell University

If the deviation of the transmissive grating from the spherical center is small, different colors that go toward the roof mirror are almost parallel to the input beam. The total optical path length thus becomes

$$
\begin{aligned}
\ell &= 2\left(\ell_1 + \ell_2\right) - x + M - y\cos\left(\frac{\pi}{2} - \theta_{\text{in}}\right) \\
&= 2\left(\ell_1 + \ell_2\right) - x + M - y\sin\theta_{\text{in}} \\
&\rightarrow 2\left(\ell_1 + \ell_2\right) - x - y\sin\theta_{\text{in}}, \quad \text{after ignoring } M.
\end{aligned} \tag{9.38}
$$

Here, we go through some algebra. Eq. (9.36b) leads to

$$
\frac{2R}{-\cos(\theta_{\text{in}} - \theta_{\text{out}})} = \frac{s}{\sin\theta} = \frac{\ell_1}{\sin\phi} \quad \Rightarrow \quad \sin\theta = -\frac{s}{2R}\cos(\theta_{\text{in}} - \theta_{\text{out}}) \tag{9.39}
$$

Eq. (9.36d) leads to

$$
\begin{aligned}
\sin\phi = \sin\left(\theta_{\text{in}} - \theta_{\text{out}} - \frac{\pi}{2} - \theta\right) &= -\cos(\theta_{\text{in}} - \theta_{\text{out}} - \theta) \\
&= -\cos(\theta_{\text{in}} - \theta_{\text{out}})\cos\theta - \sin(\theta_{\text{in}} - \theta_{\text{out}})\sin\theta.
\end{aligned} \tag{9.40}
$$

With Eq. (9.40), Eq. (9.36b) leads to

$$
\begin{aligned}
\ell_1 &= \frac{2R}{-\cos(\theta_{\text{in}} - \theta_{\text{out}})}\sin\phi \\
&= 2R\left[\cos\theta + \tan(\theta_{\text{in}} - \theta_{\text{out}})\sin\theta\right].
\end{aligned} \tag{9.41}
$$

With Eq. (9.39), Eq. (9.36c) leads to

$$
\sin\psi = 2\sin\theta = -\frac{s}{R}\cos(\theta_{\text{in}} - \theta_{\text{out}}). \tag{9.42}
$$

With Eq. (9.42), Eq. (9.36c) leads to

$$
\begin{aligned}
\ell_2 &= \frac{R}{\sin\theta}\sin(\psi - \theta) \\
&= \frac{R}{\sin\theta}\left(\sin\psi\cos\theta - \cos\psi\sin\theta\right) \\
&= \frac{R}{\sin\theta}\left(2\sin\theta\cos\theta - \cos\psi\sin\theta\right) \\
&= R\left(2\cos\theta - \cos\psi\right).
\end{aligned} \tag{9.43}
$$

To calculate x and y, we use Eq. (9.39) to find $(\psi - \theta) + \phi$.

$$
(\psi - \theta) + \phi = (\psi - 2\theta) + \left(\theta_{\text{in}} - \theta_{\text{out}} - \frac{\pi}{2}\right). \tag{9.44}
$$

It is then put into Eq. (9.37).

$$
\frac{2\left[\ell_2\sin\psi - \ell_1\sin(\psi - 2\theta)\right]}{-\cos\left(2\theta_{\text{in}} - \theta_{\text{out}} - 2\left[(\psi - \theta) + \phi\right]\right)} = \frac{x}{\sin\left((\psi - 2\theta) - \theta_{\text{out}}\right)} = \frac{y}{\sin\left(\frac{\pi}{2} - (\psi - 2\theta)\right)}
$$

$$\Rightarrow \quad \frac{2\left[\ell_2 \sin\psi - \ell_1 \sin(\psi - 2\theta)\right]}{\cos\left(\theta_{\text{out}} - 2\left(\psi - 2\theta\right)\right)} = \frac{x}{\sin\left((\psi - 2\theta) - \theta_{\text{out}}\right)} = \frac{y}{\cos(\psi - 2\theta)}. \tag{9.45}$$

Finally, this leads to

$$x = \frac{2\left[\ell_2 \sin\psi - \ell_1 \sin(\psi - 2\theta)\right]}{\cos\left(\theta_{\text{out}} - 2\left(\psi - 2\theta\right)\right)} \sin\left((\psi - 2\theta) - \theta_{\text{out}}\right)$$
$$= \frac{2\left[\ell_2 \sin\psi - \ell_1 \sin(\psi - 2\theta)\right]}{\cos(\psi - 2\theta)\cot\left((\psi - 2\theta) - \theta_{\text{out}}\right) - \sin(\psi - 2\theta)} \tag{9.46a}$$

$$y = \frac{2\left[\ell_2 \sin\psi - \ell_1 \sin(\psi - 2\theta)\right]}{\cos\left(\theta_{\text{out}} - 2\left(\psi - 2\theta\right)\right)} \cos(\psi - 2\theta)$$
$$= \frac{2\left[\ell_2 \sin\psi - \ell_1 \sin(\psi - 2\theta)\right]}{\cos\left((\psi - 2\theta) - \theta_{\text{out}}\right) - \tan(\psi - 2\theta)\sin\left((\psi - 2\theta) - \theta_{\text{out}}\right)} \tag{9.46b}$$

With ℓ_1, ℓ_2, x, and y, we can calculate the optical path length ℓ [Eq. (9.38)].

Recall that the grating phase needs to be considered for a total accumulated phase. Unlike Treacy type, the blue light propagates farther than the red light (Fig. 9.13). The larger the relative position y, the more grating phase needs to be added to redirect the light vertically. Thus, the single-pass total phase is

$$\phi_{\text{single-pass}} = k\ell - m\frac{y}{\Lambda}2\pi. \tag{9.47}$$

The double-pass total phase is

$$\phi_{\text{double-pass}} = 2\phi_{\text{single-pass}} = 2k\ell - 4m\pi\frac{y}{\Lambda}. \tag{9.48}$$
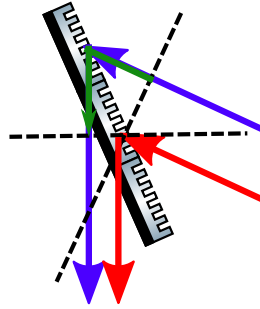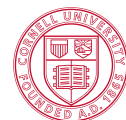

Figure 9.13: The grating phase of an Offner-type transmissive grating dechirper/stertcher.

9.4.2 Reflective single-grating Offner type

Its optical path length is similar to the transmissive one except a reflective π phase. The double-pass total phase is

$$\phi_{\text{double-pass}} = 2k\ell + 2\left(\pi - m\frac{y}{\Lambda}2\pi\right). \tag{9.49}$$

Cornell University

9.4.3 Aberration-free transmissive Offner type

The Offner dechirper/stretcher discussed previously (Fig. 9.11) introduces an off-center distance to mitigate the difficulty of aligning two parallel gratings; however, this introduces aberration. Here, for broadband pulses, an aberration-free design (Fig. 9.14) is preferred to avoid distortion during the dechirping process.
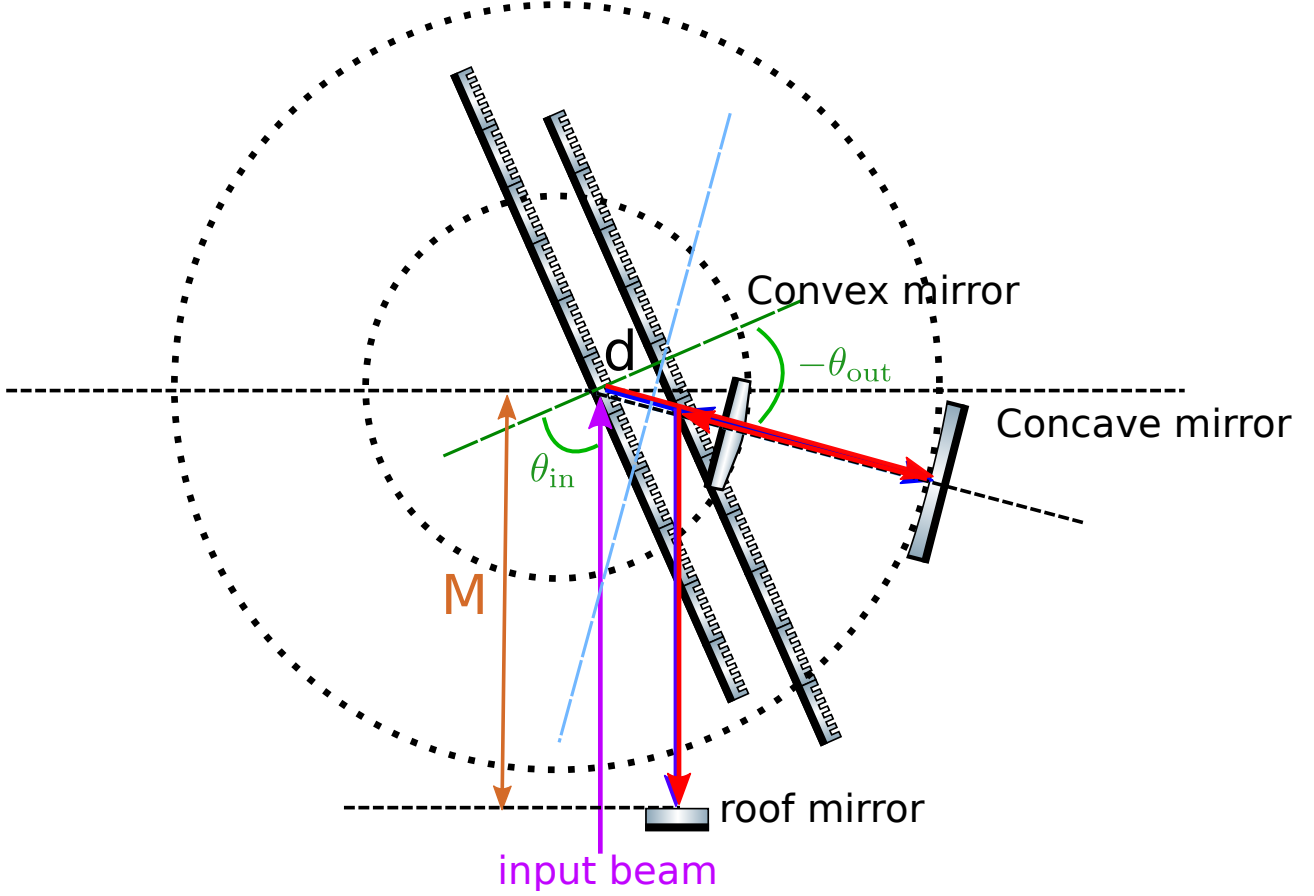

Figure 9.14: Aberration-free double-grating transmissive Offner dechirper/stretcher.

Assume the offset of two gratings is d, the path lengths to travel are, in order,

1. $\ell_1 = 2R$

2. $x = d\sec(-\theta_{\text{out}}) \quad \Rightarrow \quad \ell_2 = 2R - x$

3. $\phi = -\theta_{\text{out}} - \left(\frac{\pi}{2} - \theta_{\text{in}}\right) \quad \Rightarrow \quad \ell_3 = 2\left(M - x\sin\phi\right)$

4. ℓ_2

5. ℓ_1

The single-pass optical path length ℓ is

$$\ell = 2\left(\ell_1 + \ell_2\right) + \ell_3$$
$$= 8R - 2x + 2M - 2x\sin\phi$$

Cornell University

$$\to -2x\left[1 + \sin\phi\right] \tag{9.50}$$

Thus, the single-pass phase, including the grating phase, is

$$\phi_{\text{single-pass}} = k\ell - m\frac{d\tan(-\theta_{\text{out}})}{\Lambda}2\pi \tag{9.51}$$

The double-pass total phase is

$$\phi_{\text{double-pass}} = 2\phi_{\text{single-pass}} = 2k\ell - 4m\pi\frac{d\tan(-\theta_{\text{out}})}{\Lambda}. \tag{9.52}$$

Fig. 9.15 shows how a real Offner dechirper/stretcher is aligned.
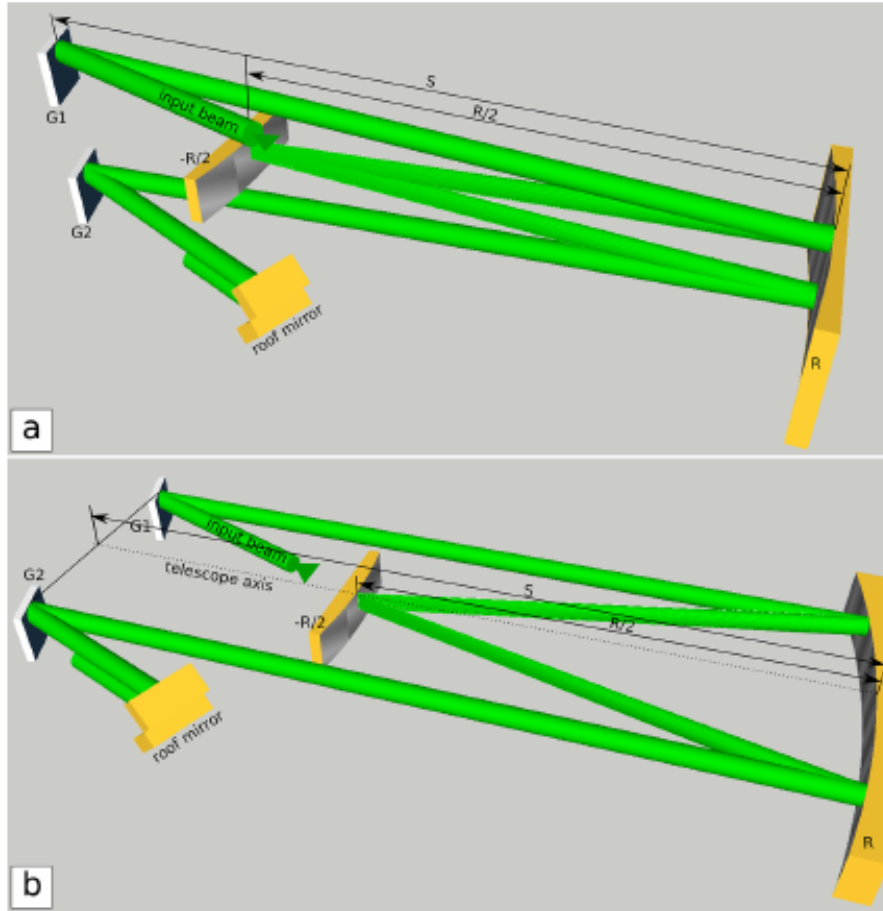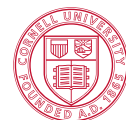

Figure 9.15: Two real configurations of an aberration-free double-grating Offner dechirper/stretcher [17].

Bibliography

[1] M. Bache, M. S. Habib, C. Markos, and J. Lægsgaard, "Poor-man's model of hollow-core anti-resonant fibers", J. Opt. Soc. Am. B **36**, 69–80 (2019).

[2] E. A. J. Marcatili and R. A. Schmeltzer, "Hollow metallic and dielectric waveguides for long distance optical transmission and lasers", Bell Syst. Tech. J. **43**, 1783–1809 (1964).

[3] Y.-H. Chen, P. Sidorenko, E. Antonio-Lopez, R. Amezcua-Correa, and F. W. Wise, "Efficient soliton self-frequency shift in hydrogen-filled hollow-core fiber", Opt. Lett. **47**, 285–288 (2022).

[4] Y.-H. Chen, J. Moses, and F. Wise, "Femtosecond long-wave-infrared generation in hydrogen-filled hollow-core fiber", J. Opt. Soc. Am. B **40**, 796–806 (2023).

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

[6] 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).

[7] L. G. Wright, Z. M. Ziegler, P. M. Lushnikov, Z. Zhu, M. A. Eftekhar, D. N. Christodoulides, and F. W. Wise, "Multimode Nonlinear Fiber Optics: Massively Parallel Numerical Solver, Tutorial, and Outlook", IEEE J. Sel. Top. Quantum Electron. **24**, 1–16 (2018).

[8] E. Treacy, "Optical pulse compression with diffraction gratings", IEEE J. Quantum Electron. **5**, 454–458 (1969).

[9] W. Dietel, J. J. Fontaine, and J.-C. Diels, "Intracavity pulse compression with glass: a new method of generating pulses shorter than 60 fsec", Opt. Lett. **8**, 4–6 (1983).

[10] R. L. Fork, O. E. Martinez, and J. P. Gordon, "Negative dispersion using pairs of prisms", Opt. Lett. **9**, 150–152 (1984).

[11] A. Offner, "Unit power imaging catoptric anastigmat", U.S. pat. 3748015 (June 1971).

[12] O. E. Martinez, R. L. Fork, and J. P. Gordon, "Theory of passively mode-locked lasers including self-phase modulation and group-velocity dispersion", Opt. Lett. **9**, 156–158 (1984).

[13] O. E. Martinez, R. L. Fork, and J. P. Gordon, "Theory of passively mode-locked lasers for the case of a nonlinear complex-propagation coefficient", J. Opt. Soc. Am. B **2**, 753–760 (1985).

[14] G. P. Agrawal, "Chapter 6 - pulse compression", in *Applications of nonlinear fiber optics (second edition)* (Academic Press, Burlington, 2008), pp. 245–300.

[15] G. Cheriaux, P. Rousseau, F. Salin, J. P. Chambaret, B. Walker, and L. F. Dimauro, "Aberration-free stretcher design for ultrashort-pulse amplification", Opt. Lett. **21**, 414–416 (1996).

[16] X. Liu, C. Wang, X. Wang, X. Lu, P. Bai, Y. Liu, Y. Li, K. Liu, L. Yu, Y. Leng, and R. Li, "Dispersion Management in 10-PW Laser Front End", Optics **1**, 191–201 (2020).

[17] D. Shvydkoy and V. Trunov, "Negatively chirped pulse compressor with internal telescope for 1.4 µm range", Appl. Phys. B **126**, 116 (2020).

Cornell University