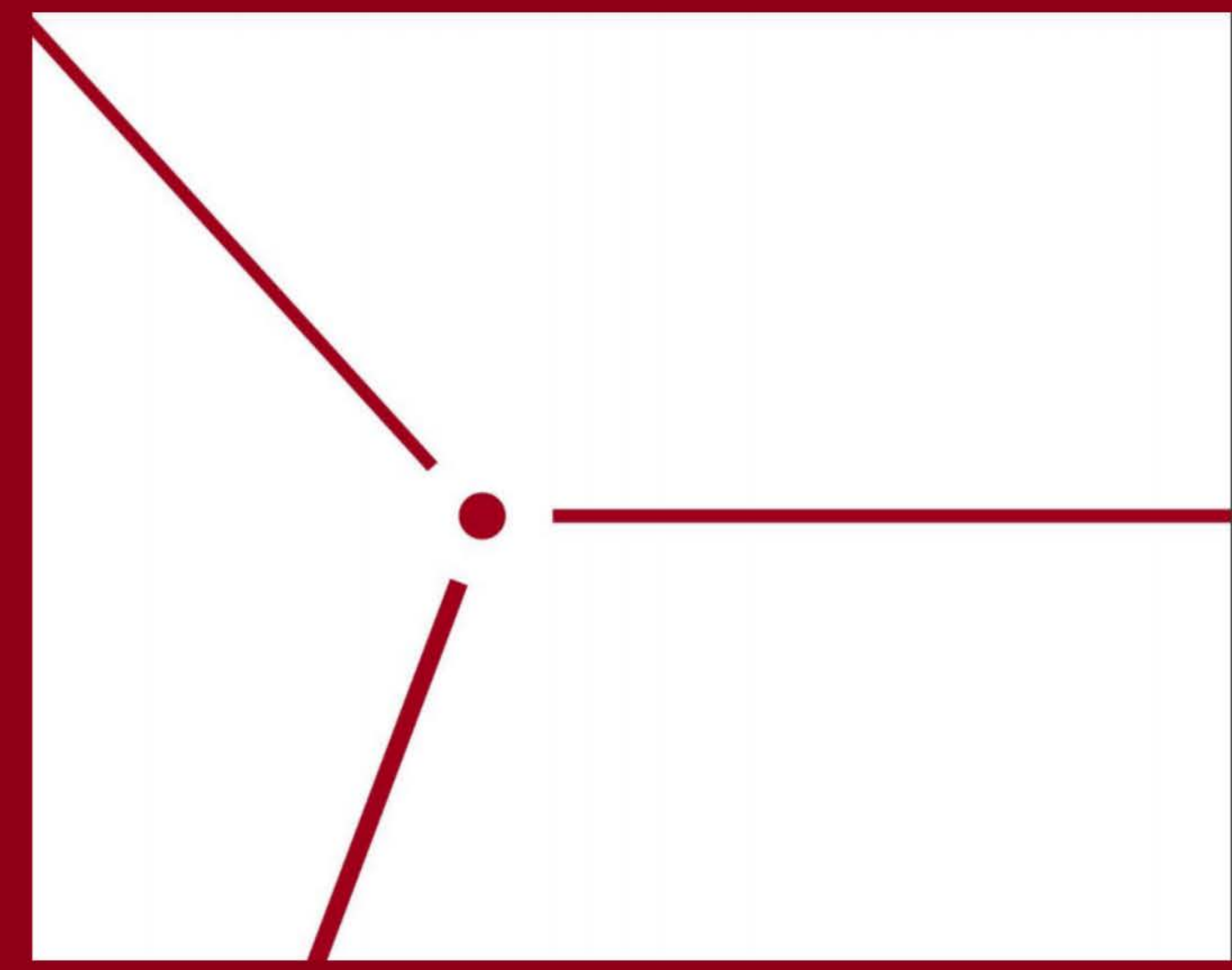
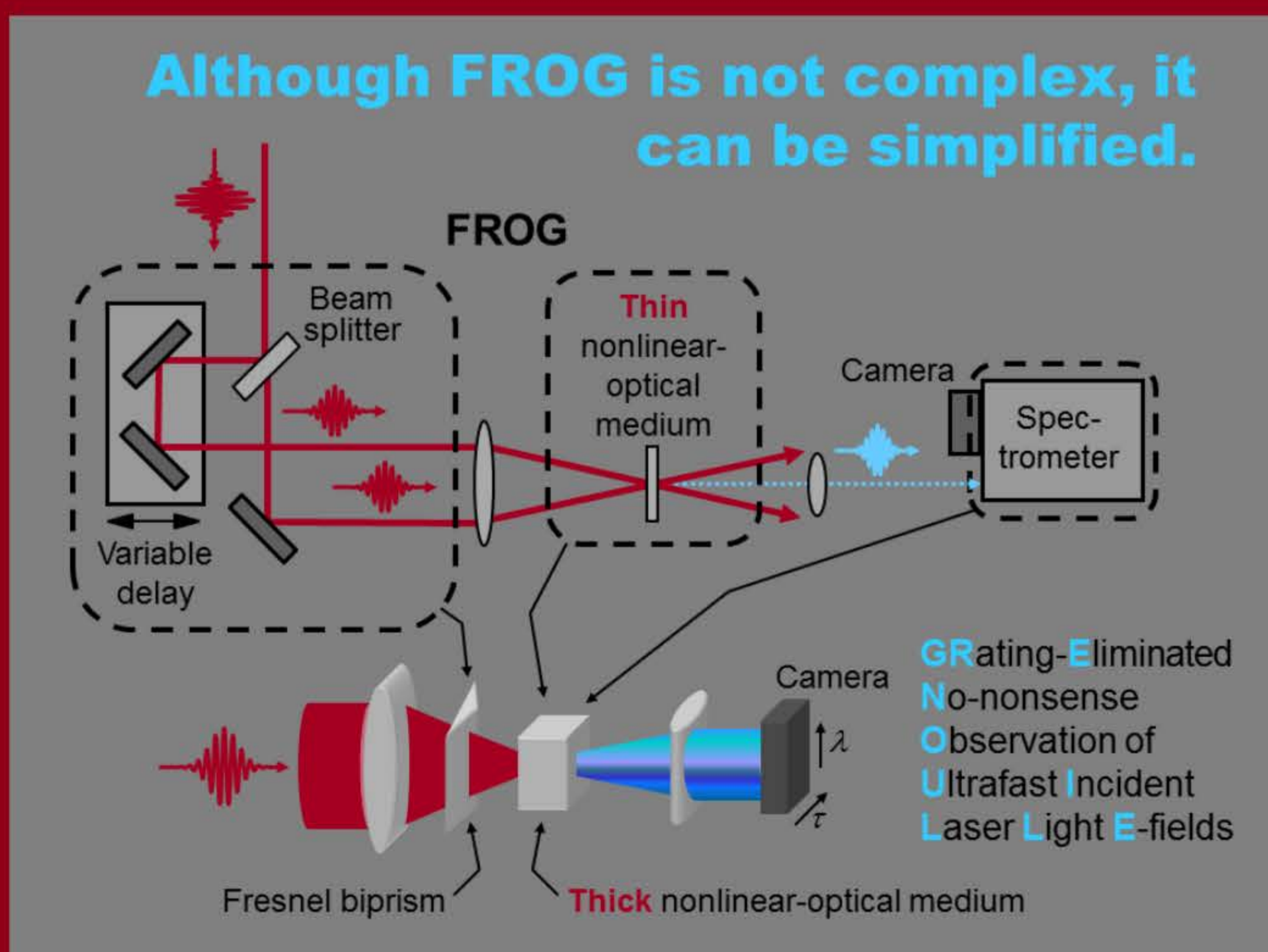


FROG Trace Data Analysis with Python – Elyse Cabrera



Introduction

At SLAC, ultrashort optical laser pulses are needed for a variety of experiments due to their high energy levels, from changing the composition of a glass lens internally to imaging medical treatments at the atomic level. As part of LCLS II, we have been trying to make the pulses shorter in time and therefore more powerful. To measure the duration as well as the wavelength distribution and phase of each pulse, we use a Frequency-Resolved Optical Gating device (FROG).



This summer, the goal of my work was to write a Python program to compile and analyze the FROG data into a more streamlined format.

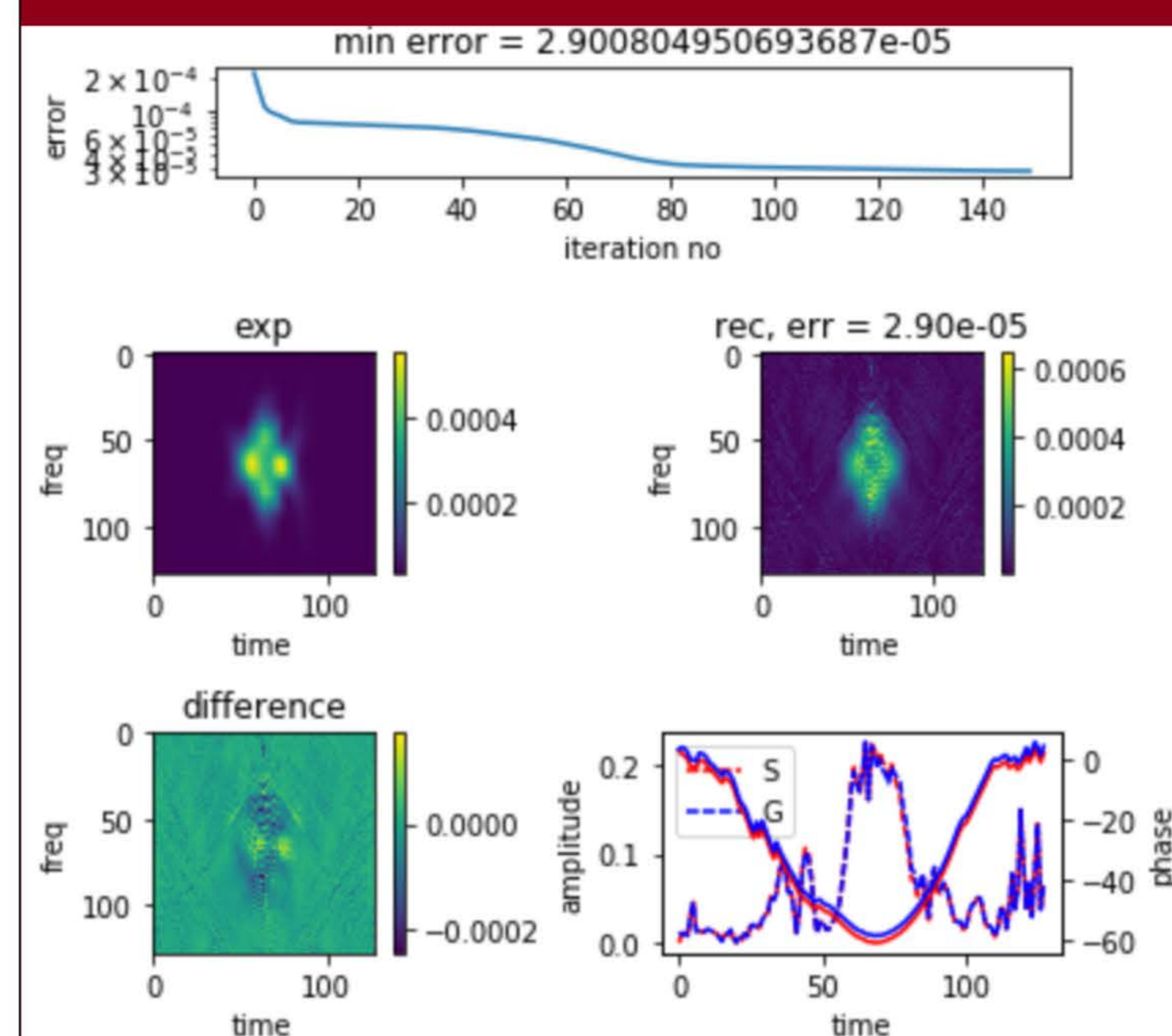
Process

Froglib is a library to reconstruct the complex optical field of short optical pulses from FROG. It offers setup to generate two types of artificial traces as well as reconstruction loops to be used with actual FROG data. The most important thing it offers however, is basic implementation of Rick Trebino's Principle Components Generalized Projection (PCGP) algorithm which utilizes the Fourier Transform to interpret the component wavelengths from the composite wave data returned by the FROG. At the beginning of the project, I created artificial traces and plugged them into the reconstruction algorithm to explore the capabilities of the library. After that, I moved on to real data. My program parses the files and appends their content into a list before cropping that list around the trace to be put into the reconstruction algorithm.

Results

For each trace, the data list must be manually cropped to center it to be properly reconstructed.

```
x_lower = 42 #
x_upper = 170
y_lower = 405
y_upper = 533
```



As a next step, I would find a way to automate the cropping function to simplify the program.

Acknowledgements

I would like to thank Kai LaFortune, Ruairidh Forbes, and Alan Fry for supporting my work this summer.

<https://github.com/xmhk/froglib>