# OPT212 Final Project Fall 2024

## Topic Briefings

**Introduction:**

You and your group will choose a topic for your final project from several different final project topic options. In addition to the topics described here, you have the opportunity to propose a final project topic as long as it results in a GUI tool. Independent topics will be reviewed to ensure the topic is feasible and, additionally, relevant and not too simple.

To propose your own topic, please speak with Greg ASAP. Since it is possible your independent topic will not be approved, you should also come up with a preferred topic from the list provided here.

The final project will have following general requirements:

- MUST be completed using AppDesigner
- Create a GUI application to solve a mathematical, physical, or optical problem
- Compile the GUI script for distribution as a Windows and/or Mac application
- Create an informal final report describing the process of script development, pitfalls found, lessons learned, etc.
- Demonstrate the application with a set of interesting and instructive examples

The final application requirements (as far as functionality is concerned) will remain somewhat open-ended. This way, if a group or groups has/have a hard time completing the project given reasonable time constraints and the project description (more detail will be provided soon), the group will not be penalized for an inability to complete a project that was too ambitious in its design.

On the other hand, if a project turns out to be too simple and a group or groups finish the project too quickly, additional functionality requirements may be added in order to assure each topic is approximately equal in difficulty.

## Topic 1: GUI Curve Fitting Tool

You are going to create a curve fitting tool which will:

- Import a data file from Excel or a text document
- Plot the data on a set of axes
- Fit a curve to the data following user input
- Calculate a statistical error for the fit and display
- Contain built in curve fitting tools for linear, polynomial, and single-exponential fitting
- Allow for a user defined function and starting conditions for custom, non-linear curve fitting
- Suppress all command line output and additional figure window output. (Everything should display within a single GUI window)
- Perform a non-linear optimization-based curve fitting algorithm when the custom function option is selected by the user

## Topic 2: GUI Interferogram Simulator

You are going to create an interferogram simulator which will:

- Allow the user to generate a live-updating, grayscale, circular interferogram on a black field (or within a circular plot)
- Allow the user to input a variable amount of tilt and defocus
- Allow the user to input variable amounts of the third-order Seidel aberrations
- Allow the user to switch to a Zernike Polynomial aberration scheme through third order (First 16 using the Wyant description of the Zernike-described aberrations https://spie.org/publications/spie-publication-resources/optipedia-free-optics-information/fg10_p24-25_zernike-polynomials#_=_) and input relevant coefficients
- Plot the tangential and sagittal wave fan across the pupil or half pupil as relevant
- Allow the user to view an animated version of the interferogram as tilt or defocus (or, potentially, any aberration) is varied in real-time.

## Topic 3: Waveform/Interference Simulator Tool

You are going to create a plane and spherical wave propagation tool which will:

- Accept an input of a single plane wave or single spherical wave
- Accept an input of two plane waves or two spherical waves and properly interfere them
- Accept an input of wave amplitude, wavelength, index of refraction for the medium, wave phase, plane wave angle, and spherical wave spacing, as is applicable
- Show an animation of the propagation of the wave(s) through time
- Ensure appropriate time durations are used given the incredibly fast speed of light
- Allow the user the option of viewing a static image of the waves given an input phase without running the animation

### Topic 4: Far-Field Diffraction Simulator Tool

You are going to create a far-field diffraction simulator which will:

- Allow the user to choose an aperture of interest from a given list
- Allow the user to change basic properties of the aperture regarding aperture size
- Show the user the aperture
- Compute the far-field diffraction pattern in two spatial dimensions
- Show the far-field diffraction pattern
- Show any relevant cross sections for the given aperture
- Code for efficiency

### Topic 5: Refraction/Dispersion Simulator Tool

You are going to create a refraction and dispersion simulator tool which will:

- Allow the user to choose from several standard optical elements or interfaces including an air/glass interface, glass/air interface, plane parallel plate, 60-degree prism, and (maybe) a simple positive lens.
- Use Snell's law and standard equations for visible-wavelength dispersion should be used to properly show a ray of a given angle transmitting through the interface or object (but this should go short of real ray tracing)
- Allow the user to input a starting angle for the ray coming into the system
- Include common optical glasses (go to the Thorlabs website and look at the glasses they use for standard optical elements as well as high-dispersion visible prisms). Include at minimum, 5 glass types for the user to choose from.
- Allow the user to choose between white light or red, yellow, green, or blue monochromatic light as the incident light source.
- Show the differing ray paths for each color of the rainbow (red, orange, yellow, green, blue, indigo, and violet) when white light is the desired input.
- Optionally include a slider which will change the input ray angle and plot the desired output in real time.
- Optionally compute the critical angle and post an error when the critical angle has been exceeded.