System optimization, ray tracing and analysis using ZOSAPI and Matlab

The ZOSAPI allows Matlab (external program) to perform any operation which can be performed using the OpticStudio graphical user interface. This includes but not limited to changing the system parameters, performing ray tracing, defining merit functions, running the system optimization, running the system analysis tools and getting the analysis results directly from Matlab. In this exercise we demonstrate this by designing and analyzing a classical achromate using the ZOSAPI-Matlab interface.

a) Establish the OpticStudio Matlab connection in the Interactive Extension mode.

b) Write a matlab script which constructs a doublet lens with the following initial parameters using the ZOSAPI_Application object returned in (a).
   - System aperture: Entrance pupil diameter of 20 mm
   - System wavelength: F-d-C (Visible)
   - Lens thickness: 5 mm and 3 mm
   - Glasses: BK7 and SF2
   - Radius of 1st surface: 100 mm
   - Radius of 2nd and 3rd surface: Inf

c) Optimize the doublet in (b) with classical achromate criteria and the effective focal length of 100 mm
   - System variables: The three radii and last thickness (image distance)
   - Merit functions:
     - EFFL = 100
     - AXCL between wavelength 1 and 3 = 0
     - REAY at last surface for marginal ray = 0
     - PARY at last surface for marginal ray = 0

   Plot the longitudinal aberration of the achromate to see the fulfillment of the requirements.

d) Plot spot diagram of the system by tracing 1500 real rays randomly distributed on the entrance pupil from the on-axis field point using the batch ray tracing tool of ZOSAPI.

e) Using ZOSAPI open the analysis window for the PSF cross section and change the settings to plot scale = 20 and sample size = 256 x 256. Then get results for the PSF cross section for each of the three wavelengths and plot them on a single Matlab figure.

Solution:

a) First generate the template code from the OpticStudio by following Programming tab → ZOS-API.NET Application Builders → MATLAB → Interactive Extension

This generates a matlab script with the template code for the Interactive Extension and saves it under the “...\Documents\Zemax\ZOS-API Projects\MATLABZOSConnection”.

Now in OpticStudio turn on the Interactive Extension mode by following Programming tab → ZOS-API.NET Applications → Interactive Extension.
This makes the OpticStudio application to be blocked and wait for the connection from an external application like Matlab.

Next open the template script (generated earlier) in Matlab. The template script has functions to initialize the connection with the OpticStudio instance.

Running the template script creates the connection between Matlab and OpticStudio and returns “TheApplication” which is ZOS application object. The application object contains all data and functions for the subsequent interaction of Matlab with the OpticStudio. The following figure shows the first few elements of the application object returned.

```
>> TheApplication = MATLABZOSConnection();
Found OpticStudio error: program files/zemax/opticstudio 18.4
TheApplication =

ZOSAPI_Application with properties:

    LicenseStatus: PremiumEdition
    IsValidLicenseForAPI: 1
    IsInitializationErrors: [1x1 System.String]
    InitializationErrorString: [1x1 System.String]
    ConnectionMode: Extension
    Mode: Plugin
    ClientProcess: 9916
    OpticStudioVersion: 180308
    SerialNumber: 103087
    ExecutionCode: [1x1 System.String]
    Expiration: [1x1 System.DateTime]
    ExpirationYear: 2018
    ExpirationMonth: 10
    ExpirationDay: 23
    PrimarySystem: [1x1 ZemaxUI.ZOSAPI.ZemaxSystem]

b) Create a Matlab function accepting the ZOSAPI application object as input and import the ZOSAPI class.

```

```
function [r] = ClassicalAchromateUsingZOSAPIDemo(TheApplication)
    if nargin < 1
        error('The function needs TheApplication object as input');
    end
    import ZOSAPI.*;

Now get the optical system object and create an empty new system in the sample optical systems directory with a given name.

```

```
% Set up primary optical system
TheSystem = TheApplication.PrimarySystem;
```
sampleDir = TheApplication.SamplesDir;
% Make new file
achromate_Initial= System.String.Concat(sampleDir, "\Sequential\Objectives\ Achromate ZOS API Initial Design.zmx");
achromate_Optimized= System.String.Concat(sampleDir, "\Sequential\Objectives\ Achromate ZOS API Optimized Design.zmx");
TheSystem.New(false);

Get the system data and setup the system aperture value, the fields and the wavelengths.

% Aperture
TheSystemData = TheSystem.SystemData;
TheSystemData.Aperture.ApertureValue = 20;
% Wavelength preset FdC
TheSystemData.Wavelengths.SelectWavelengthPreset(ZOSAPI.SystemData.WavelengthPreset. FdC_Visible);

Now get the lens data editor object and add two more surfaces with their corresponding parameters.

% Lens data
TheLDE = TheSystem.LDE;
TheLDE.InsertNewSurfaceAt(2);
TheLDE.InsertNewSurfaceAt(2);
TheLDE.InsertNewSurfaceAt(2);

Surface_1 = TheLDE.GetSurfaceAt(1);
Surface_2 = TheLDE.GetSurfaceAt(2);
Surface_3 = TheLDE.GetSurfaceAt(3);
Surface_4 = TheLDE.GetSurfaceAt(4);

Surface_1.Thickness = 5.0;
Surface_2.Radius = 100.0; % Just initial guess
Surface_2.Thickness = 5.0;
Surface_2.Comment = 'front';
Surface_2.Material = 'BK7';
Surface_3.Radius = Inf; % Just initial guess
Surface_3.Thickness = 3.0;
Surface_3.Comment = 'cemented';
Surface_3.Material = 'SF2';
Surface_4.Radius = Inf; % Just initial guess
Surface_4.Thickness = 100.0;

TheSystem.SaveAs(achromate_Initial);

The Lens data editor with the initial lens data now looks like

c) Now make the three radii and the final thickness cell variable and define the merit function using the following code snippet.
% Variables
Surface_2.RadiusCell.MakeSolveVariable();
Surface_3.RadiusCell.MakeSolveVariable();
Surface_4.RadiusCell.MakeSolveVariable();
Surface_4.ThicknessCell.MakeSolveVariable();

% Merit functions
TheMFE = TheSystem.MFE;
TheMFE.ShowEditor;
Operand_1=TheMFE.GetOperandAt(1);  Operand_1.ChangeType(ZOSAPI.Editors.MFE.MeritOperandType.EFFL);
Operand_1.GetCellAt(3).IntegerValue = int32(2);  % Wave
Operand_1.Target = 100.0;
Operand_1.Weight = 1.0;
Operand_2=TheMFE.InsertNewOperandAt(2);
Operand_2.ChangeType(ZOSAPI.Editors.MFE.MeritOperandType.AXCL);  Operand_2.GetCellAt(2).IntegerValue = int32(1);
Operand_2.GetCellAt(3).IntegerValue = int32(3);  % Wave 1
Operand_2.Target = 0.0;
Operand_2.Weight = 1.0;
Operand_3=TheMFE.AddOperand();  Operand_3.ChangeType(ZOSAPI.Editors.MFE.MeritOperandType.REAY);
Operand_3.GetCellAt(2).IntegerValue = int32(5);  % Surf
Operand_3.GetCellAt(3).IntegerValue = int32(2);  % Wave
Operand_3.GetCellAt(7).DoubleValue = 1;  % Py
Operand_3.Target = 0.0;
Operand_3.Weight = 1.0;
Operand_4=TheMFE.AddOperand();  Operand_4.ChangeType(ZOSAPI.Editors.MFE.MeritOperandType.PARY);
Operand_4.GetCellAt(2).IntegerValue = int32(5);  % Surf
Operand_4.GetCellAt(3).IntegerValue = int32(2);  % Wave
Operand_4.GetCellAt(7).DoubleValue = 1;  % Py
Operand_4.Target = 0.0;
Operand_4.Weight = 1.0;

The resulting merit function editor would be

[Image of the resulting merit function editor]

Now perform the local optimization using the optimization tool.

% Local optimisation till completion
LocalOpt = TheSystem.Tools.OpenLocalOptimization();
LocalOpt.NumberOfCores = 8;
initial_MF = TheSystem.MFE.CalculateMeritFunction
LocalOpt.RunAndWaitForCompletion();
final_MF = TheSystem.MFE.CalculateMeritFunction
LocalOpt.Close();
% Save final optimized system
TheSystem.SaveAs(achromate_Optimized);

The initial and final merit function values calculated before and after the optimization are 46 and $5.7 \times 10^{-11}$ respectively.

To observe the longitudinal aberration plot of the system, use the ZOSAPI to access the corresponding analysis tool by using code snippet given below.

```matlab
% Longitudinal Aberration Plots analysis data
longitudinalAberration = TheSystem.Analyses.New_LongitudinalAberration();
longitudinalAberration.ApplyAndWaitForCompletion();
longAberrPlotResult = longitudinalAberration.GetResults();
dataSeries = longAberrPlotResult.DataSeries(1);

% Access the x and y data of the analysis window result
py = dataSeries.XData.Data.double; % Py
long_shift = dataSeries.YData.Data.double; % Longitudinal aberration

Plot the longitudinal aberration data in matlab plot.

figure;
plot(long_shift(:,1),py,'-','color','b','Linewidth',2); hold on;
plot(long_shift(:,2),py,'-','color','g','Linewidth',2);
plot(long_shift(:,3),py,'-','color','r','Linewidth',2);

title('Longitudinal Aberration');
xlabel('Milimeters');
ylabel('Py (Normalized)');
legend('486 nm', '588 nm', '656 nm');
maxX = max(max((abs(long_shift))));
xlim([-maxX,maxX]*1.2);
grid on;
set(gca,'FontSize',22);
```

The longitudinal aberration plot is plotted in Matlab shows that the classical achromate requirements are fulfilled by the system.

d) Setup the batch ray tracing tool for unpolarized ray defined using the normalized pupil coordinates.
% Set up Batch Ray Trace
rayTracer = TheSystem.Tools.OpenBatchRayTrace();
nSurf = TheSystem.LDE.NumberOfSurfaces;
nRays = 1500;
realRayType = ZOSAPI.Tools.RayTrace.RaysType.Real;
normUnPolData = rayTracer.CreateNormUnpol(nRays, realRayType, nSurf);

Define the normalized field coordinates and pupil coordinates for randomly distributed rays in the entrance pupil.

% Field coordinate
hx = 0; hy = 0;
% Normalized pupil coordinate
for r = 1:nRays
    rand_px = rand() * 2 - 1;
    rand_py = rand() * 2 - 1;
    while (rand_px^2 + rand_py^2 > 1)
        rand_py = rand() * 2 - 1;
    end
    px(r) = rand_px;
    py(r) = rand_py;
end

Now add individual rays to the batch ray tracer tool and trace for each of the three wavelengths. Save the final (x,y) locations of the rays in a matrix.

% initialize x/y image plane arrays
nWaves = double(TheSystem.SystemData.Wavelengths.NumberOfWaveLengths);
final_x = zeros(nRays, nWaves);
final_y = zeros(nRays, nWaves);
dot_color = {'b', 'g', 'r'};
for wave = 1:nWaves
% Adding Rays to Batch
    normUnPolData.ClearData();
    waveNumber = wave;
    calcOPD = ZOSAPI.Tools.RayTrace.OPDMode.None;
    for r = 1:nRays
        normUnPolData.AddRay(waveNumber, hx, hy, px(r), py(r), calcOPD);
    end
% Run Batch Ray Trace
    rayTracer.RunAndWaitForCompletion();
% Read batch raytrace and record results
    normUnPolData.StartReadingResults();
    [success, rayNumber, errCode, vigCode, x, y, ~, ~, ~, ~, ~, ~, ~, ~] = normUnPolData.ReadNextResult();
    while success
        if (errCode == 0) && (vigCode == 0)
            final_x(rayNumber, wave) = x;
            final_y(rayNumber, wave) = y;
        end
        [success, rayNumber, errCode, vigCode, x, y, ~, ~, ~, ~, ~, ~, ~, ~] = normUnPolData.ReadNextResult();
    end
end

Finally plot the final positions as spot diagram using the following snippet.
Plot the results as spot diagram
maxX = max(abs(final_x(:)));  
maxY = max(abs(final_y(:)));  
for wave = 1:nWaves  
    subplot(1,nWaves,wave);  
    plot(final_x(:,wave), final_y(:,wave), '.', 'MarkerSize', 4, 'color', dot_color{wave});  
    title([num2str(TheSystem.SystemData.Wavelen
gths.GetWavelength(wave).Wavelength),' mum']);  
    axis('square');  
    xlim([-1.5*maxX,1.5*maxX]);  
    ylim([-1.5*maxY,1.5*maxY]);  
    grid on;  
end

The resulting spot diagram is shown in the following figure.

**e)** The first step is to create an analysis tool for window for the FFT PSF cross-sectional analysis window and change the settings.

```matlab
% Create analysis
TheAnalyses  = TheSystem.Analyses;  
newWin = TheAnalyses.New_FftPsfCrossSection();  
% Settings  
newWin_Settings = newWin.GetSettings();  
newWin_Settings.SampleSize = ZOSAPI.Analysis.SampleSizes.S_256x256;  
newWin_Settings.PlotScale = 20;
```

Then run the analysis tool and get results for each of the three wavelengths defined in the optical system.

```matlab
% Create analysis
figure;  
line_color = {'b','g','r'};  
for p = 1:3  
    newWin_Settings.Wavelength.SetWavelengthNumber(p);  
    % Run Analysis  
    newWin.ApplyAndWaitForCompletion();  
    newWin_Results = newWin.GetResults();  
    dataSeries = newWin_Results.DataSeries(1);  
    % Access the x and y data of the analysis window result  
    xlin = dataSeries.XData.Data.double;  
    psf = dataSeries.YData.Data.double;  
    plot(xlin,psf,'-','Color',line_color(p),'Linewidth',2);  
end
```

```matlab
title('FFT PSF');  
xlabel('X (mum)');  
ylabel('Intensity');  
legend('486 nm', '588 nm', '656 nm');```
grid on;
set(gca,'FontSize',22);

The resulting Matlab plot is shown below.
Adaptive Optics

Setup an afocal 4f-system with focal length $f = 100$ mm for both groups at a wavelength of $\lambda = 546$ nm. The numerical aperture is $NA = 0.15$. The first lens should be a biconvex lens of thickness 5 mm and is made of $n$-BK7, the second lens is an achromate out of a vendor catalog. The pupil is located in the mid-plane of the common focus point.

a) Establish the system with the adjusted distances for an object point on axis. Calculate the Zernike coefficients and determine the surface contributions of Seidel for spherical aberrations.

b) Now the system should be corrected by an adaptive mirror. First a plane mirror with an incidence angle large enough is inserted at the position of the system pupil. The tilt angle should be chosen in a way, that there is no collision between the two lenses. The mirror surface is defined to be a Zernike-Surface now. What is the necessary normalization diameter of the surface? First we have only 3rd order corrections in mind. How many Zernike coefficients are necessary for this task? Hint: for the optimization, don’t let the tilt coefficients be free.

c) Now the Zernike coefficients are used to correct the system on axis. Perform this optimization. Is the system now diffraction limited? Which are the dominant coefficients, explain the result. Draw a plot of the surface topology.

d) Increase the surface description to a 5th order Zernike polynomial and retry the optimization. Is the result now diffraction limited? What are the dominant terms of the surface shape?

e) Why is the Zernike approach not an optimal approach from first principles? Draw the illumination area at the mirror. Now try to describe the mirror as an extended polynomial by a Taylor expansion with 16 terms. Is the correction on axis now improved in comparison to d)?

Solution:

a) The biconvex lens is defined by setting the second radius as a pick up. The first radius is optimized with the EFFL 100 in the merit function. The first distance is optimized with angular radius as merit function, the distance behind the lens is found by a solve with chief ray height zero. The following data are obtained for the first part of the system.

<table>
<thead>
<tr>
<th>Surf. Type</th>
<th>Comment</th>
<th>Radius</th>
<th>Thickness</th>
<th>Glass</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJ Standard</td>
<td>Infinity</td>
<td>93.9635769</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STO Standard</td>
<td>102.084123</td>
<td>5.0000000</td>
<td>N-BK7</td>
<td></td>
</tr>
<tr>
<td>2 Standard</td>
<td>-102.08413</td>
<td>98.3610600</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>3A Standard</td>
<td>Infinity</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If an achromate with $f = 100$ mm is searched in the catalogs, the necessary diameter should be at least $D = 2 NA f + 2y = 50$ mm. In the catalog of CV Melles Griot the lens LAO - 100.1 - 50.0 fulfills this requirement. The final distance is found by quick focus, the distance before the achromate is obtained by an optimization, if for a small field of $y = 1$ mm the chief ray angle in the image space is zero. This corresponds to the operand REAB at surface 7 in the merit function. Then the final data of the linear setup are as follows.

<table>
<thead>
<tr>
<th>Surf. Type</th>
<th>Comment</th>
<th>Radius</th>
<th>Thickness</th>
<th>Glass</th>
<th>Semi-Diameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJ Standard</td>
<td>Infinity</td>
<td>93.9435769</td>
<td>1.0000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 Standard</td>
<td>102.08412</td>
<td>5.0000000</td>
<td>N-BK7</td>
<td>22.0000000</td>
<td></td>
</tr>
<tr>
<td>2 Standard</td>
<td>-102.08413</td>
<td>98.3610600</td>
<td></td>
<td>22.0000000</td>
<td></td>
</tr>
<tr>
<td>STO Standard</td>
<td>Infinity</td>
<td>67.7633264</td>
<td>Y</td>
<td>18.0000000</td>
<td></td>
</tr>
<tr>
<td>4 Standard</td>
<td>LAO-100.1-50.0</td>
<td>69.7660900</td>
<td>14.0000000</td>
<td>N-BAK4</td>
<td>25.0000000</td>
</tr>
<tr>
<td>5 Standard</td>
<td>-59.0579000</td>
<td>5.5900000</td>
<td>N-FO</td>
<td>25.0000000</td>
<td></td>
</tr>
<tr>
<td>6 Standard</td>
<td>-139.1779000</td>
<td>92.8763588</td>
<td></td>
<td>25.0000000</td>
<td></td>
</tr>
<tr>
<td>1A Standard</td>
<td>Infinity</td>
<td>-</td>
<td></td>
<td>1.514095</td>
<td></td>
</tr>
</tbody>
</table>
The Aberrations are

It is seen, that the spherical aberration at the first surface dominates the residual performance and has a value in the range of 6 \( \lambda \). The higher orders have no considerable contribution.

b) The insertion of a bending mirror at the stop position gives the following data. The incidence angle should be larger than 13°, 15° is chosen here.

Now the surface type Zernike fringe sag is chosen at the mirror. In the Extra data editor, the normalization radius 16 mm is introduced and 9 Zernike terms are selected.
c) In this step, the default merit function for spot diagram is selected and the 9 Zernike coefficients are chosen as variables. Only the axis point is chosen for the field. The coefficients 2 and 3 are set to 0 and are fixed. The optimization gives the following result.

The dominant residual aberration is astigmatism, this can be seen in the spot diagram and the Zernike coefficients. The surface sag is nearly rotational symmetric.

d) If now the surface is extended to 5th order, the optimization gives a diffraction limited result. The cross sections through the surface sag shows, that the 5th order spherical aberration is important. A small astigmatism can be seen, that gives slightly differences between the two sections. The dominant Zernike contributions of the mirror surface are astigmatism (terms 5 and 12), coma (term 8) and spherical aberration (terms 9 and 16).
e) At the mirror, the cross section of the beam is not circular in shape. Therefore the Zernikes are not based on a circular homogeneous intensity in the pupil. But in reality, the effects are small, as can be seen in the cross section of the illumination. This comes from the fact, that \( \cos 15^\circ = 0.966 \) is near to one.

If the extended polynomial surface type is used, the result is worse. An additional disadvantage is the non-orthogonality of the Taylor expansion.
Fit of Refractive Index

The following data are measured values of the ordinary refractive index of a special material. The first column gives the wavelengths in \( \mu \text{m} \), the second column the corresponding 9 indices.

\[
\begin{array}{ll}
0.337 & 1.800 \\
0.442 & 1.780 \\
0.488 & 1.775 \\
0.532 & 1.772 \\
0.590 & 1.768 \\
0.633 & 1.766 \\
0.670 & 1.764 \\
0.755 & 1.761 \\
0.780 & 1.760 \\
\end{array}
\]

In this exercise it should be demonstrated, how a new private glass catalog can be established in Zemax and how to include a new glass according to the data above is installed.

a) Select an arbitrary catalog with only a few glasses, for instance the catalog 'ISUZU.AGF'. If there is more than one glass in the catalog, delete all glasses except one by the command 'cut'. Save the new private catalog by 'Save as' with an appropriate new name cat_new.agf.

b) Press 'Fit Index Data', an Excel-type sheet occurs. Fill in the data listed above (first generate the necessary number of lines). Fit the data by a Sellmeier formula and save the new material under glass-new. Delete the remaining old glass of the former catalog.

c) Add the new catalog into the list of Material catalogs. Show the dispersion curve of the new glass.

d) Write an ASCII file with wavelength (in micron) and index in one row with the same data and one more digit. Locate this file with the extension glass new2.dat in the folder ‘Glasscat’ of Zemax. For the decimals, a point is necessary. Compare the dispersion curves.

Solution:

a) We select the catalog ISUZU.AGF, Only one glass is inside, therefore it is not necessary to remove glasses. With ‘Save as’ and the new name ‘cat new’ we save the private material catalog.

b) By filling the table with the wavelength and index data, we get
Fit of the data:

Adding the new material and deleting the old glass gives by ‘add to catalog’ and ‘Save catalog’: 
c) The new glass catalog is added in the System explorer

The dispersion curve of the new material is visualized
d) ASCII file, it is important, that the decimal is separated by a point.

Activate: ‘Fit index data’. Load the new data file. Fit the index, the same result is obtained.

Due to slightly different data, the dispersion curve is a little bit changing.
**Stock lens matching**

With the option ‘Stock Lens Matching’ Zemax is able to make a proposal to substitute given lenses in a system by available catalog lenses. The basis is the definition of an appropriate merit function of the desired system properties and an accepted tolerance range considering the deviation of the focal length and the diameter of the component.

a) Establish a single lens of K5 with an incoming collimated beam diameter of 20 mm at 550 nm and an image distance of 87.80405 mm. Optimize the radii for a focal length of 100 mm and a minimal spot diameter. Determine the current spot diameter. Fix the half diameter of the lens to be 22 mm. Activate the stock-lens-matching feature in the optimization menu. Select the vendor catalogs of CVI Melles Griot, Newport and Thorlabs in the lens matching option for a 10% deviation of the focal length and a 20% deviation in diameter. The distances can be compensated. The merit function must be filled to give the program a measure of quality. Select ‘save the best combination’, in this case a file ‘Lens_SLM’ is created in the current folder.

b) Introduce the proposed lens by open lens_SLM and check the performance in comparison to the previous result. The merit function is preserved.

c) The problem with the solution is the posed problem: it is usually hard to find a solution with a meniscus shaped lens for a predefined image distance. Therefore now change the lens half diameter to 12 mm and optimize both radii and the final distance and then look for a solution in all catalogs.

d) Now add one more lens and optimize the doublet. Is the result diffraction limited? What happens, if the lens exchange tool is used now? Establish the proposed system.

**Solution:**

a) Initial system

The spot diameter is 63 μm.
Call of Stock lens matching routine:
The selected vendors are marked by pressed STRG.

Result of search:

b) The suggested lens ‘LENS_SLM’ is loaded and gives the following data:
The spot diameter is growing to 126 \( \mu \text{m} \).

c) Now we have only a change in diameter by 5%. 
d) Now with 2 lenses, the performance is diffraction limited with a spot size of 3.38 \( \mu m \).

The exchange tool first looks for the best matching single components and then calculates the best combination of the selected lenses.

With the best combination of the stock lenses, the result is a factor of two below the optimized result.

The automatic substitution does not work in any case (!?).
The proposed system (LENS_SLM_SLM) has a spot size of 5.16 μm.
Macro focal lengths

Write a macro, which calculates the focal lengths of all lenses in the system (in air).

Solution:

```plaintext
! get number of surfaces Ns
Ns = nsur()
! declare N, double, 1, Ns+1
! get refractive indices
FOR j = 1, Ns+1, 1
   N(j) = index(j-1)
ext
! number of lenses
Nl = 0
FOR j = 1, Ns+1, 1
   IF ( N(j) > 1 ) THEN Nl = Nl + 1
next
! header line: number of surfaces and glasses
print ""
print "lenses ", Nl
! focal lengths
print "focal lengths 
print ""
! print result
k = 0
FOR j = 1, Ns-1, 1
   IF ( N(j) == 1 ) THEN goto 1
   k = k + 1
   c1 = curv(j-1)
   c2 = curv(j)
   t = thic(j-1)
   ni = N(j)
   F = (ni-1)*(c1-c2) + (ni-1)*(ni-1)/ni*t*c1*c2
   foc = 0
   IF ( F != 0 ) THEN foc = 1/F
print "L ", k, " (" j-1, ", j ") f = ",
format 10.4
print foc
label 1
next
```

Typical output:

```
Executing C:\Gross\Zemax\MACROS\macro

lenses 3
focal lengths
L 1 ( 3 - 4) f = -25.1762
L 2 ( 8 - 9) f = 125.8811
L 3 (13-14) f = 151.0515
```
Macro telecentricity

Write a macro, which calculates the telecentricity error of a given set of sampled chief rays. Create also a plot of \( w(y) \).

Solution:

```plaintext
! sampling number over field
nfield = 21
!
! Set primary wavelength
jwave = pwav()
!
! define considered field index
jfield = 2
!
! save old field height
hyold = FLDY(jfield)
!
! calculate field increment
dw = 1/(nfield-1)
!
! fix ray data in pupil and x-field
hx = 0
px = 0
py = 0
!
! define surface of evaluation
n = nsur()
!
! header line
print " w-rel  w-abs  ws"
!
! raytrace of chief rays for all field heights and print results
DECLARE yfov, DOUBLE, 1, nfield
declare tele, double, 1, nfield
FOR j = 1,nfield,1
    hy    = (j-1) * dw
    hyabs = (j-1) * dw * hyold
    SYSP 103, jfield, hy
!
    update
    raytrace hx, hy, px, py, jwave
delw = raym(n)
!
FORMAT 10.5
PRINT hy, hyabs, delw
```
! yfov(j) = hyabs
   tele(j) = delw

NEXT!
!
! recover data
!
SYSP 103, jfield, hyold
!
! plot dw(y)
!
style = 0
options = 0
!
plot new
plot formatx, "%.2f"
plot title,"Telecentricity violation"
plot titlex,"FoV"
plot titley,"dwy(CR)"
!
plot rangex,0,sdi
plot rangey,0,waves_per_mm(iter_num)
!
ncolor = 3
PLOT data, yfov, tele, nfield, ncolor, style, options
FORMAT 5.2
plot go
!

Typical output:

```
Executing C:\Gross\Temax\MACROS\ma

<table>
<thead>
<tr>
<th>w-rel</th>
<th>w-abs</th>
<th>uw</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>-0.00817</td>
</tr>
<tr>
<td>0.0500</td>
<td>0.1000</td>
<td>-0.00621</td>
</tr>
<tr>
<td>0.1000</td>
<td>0.2000</td>
<td>-0.00832</td>
</tr>
<tr>
<td>0.1500</td>
<td>0.3000</td>
<td>-0.00650</td>
</tr>
<tr>
<td>0.2000</td>
<td>0.4000</td>
<td>-0.00876</td>
</tr>
<tr>
<td>0.2500</td>
<td>0.5000</td>
<td>-0.01010</td>
</tr>
<tr>
<td>0.3000</td>
<td>0.6000</td>
<td>-0.00950</td>
</tr>
<tr>
<td>0.3500</td>
<td>0.7000</td>
<td>-0.00998</td>
</tr>
<tr>
<td>0.4000</td>
<td>0.8000</td>
<td>-0.01054</td>
</tr>
<tr>
<td>0.4500</td>
<td>0.9000</td>
<td>-0.01117</td>
</tr>
<tr>
<td>0.5000</td>
<td>1.0000</td>
<td>-0.01188</td>
</tr>
<tr>
<td>0.5500</td>
<td>1.1000</td>
<td>-0.01265</td>
</tr>
<tr>
<td>0.6000</td>
<td>1.2000</td>
<td>-0.01351</td>
</tr>
<tr>
<td>0.6500</td>
<td>1.3000</td>
<td>-0.01444</td>
</tr>
<tr>
<td>0.7000</td>
<td>1.4000</td>
<td>-0.01544</td>
</tr>
<tr>
<td>0.7500</td>
<td>1.5000</td>
<td>-0.01652</td>
</tr>
<tr>
<td>0.8000</td>
<td>1.6000</td>
<td>-0.01767</td>
</tr>
<tr>
<td>0.8500</td>
<td>1.7000</td>
<td>-0.01890</td>
</tr>
<tr>
<td>0.9000</td>
<td>1.8000</td>
<td>-0.02020</td>
</tr>
<tr>
<td>0.9500</td>
<td>1.9000</td>
<td>-0.02158</td>
</tr>
<tr>
<td>1.0000</td>
<td>2.0000</td>
<td>-0.02303</td>
</tr>
</tbody>
</table>
```

![Graph showing telecentricity violation with data points and trend line.](image-url)