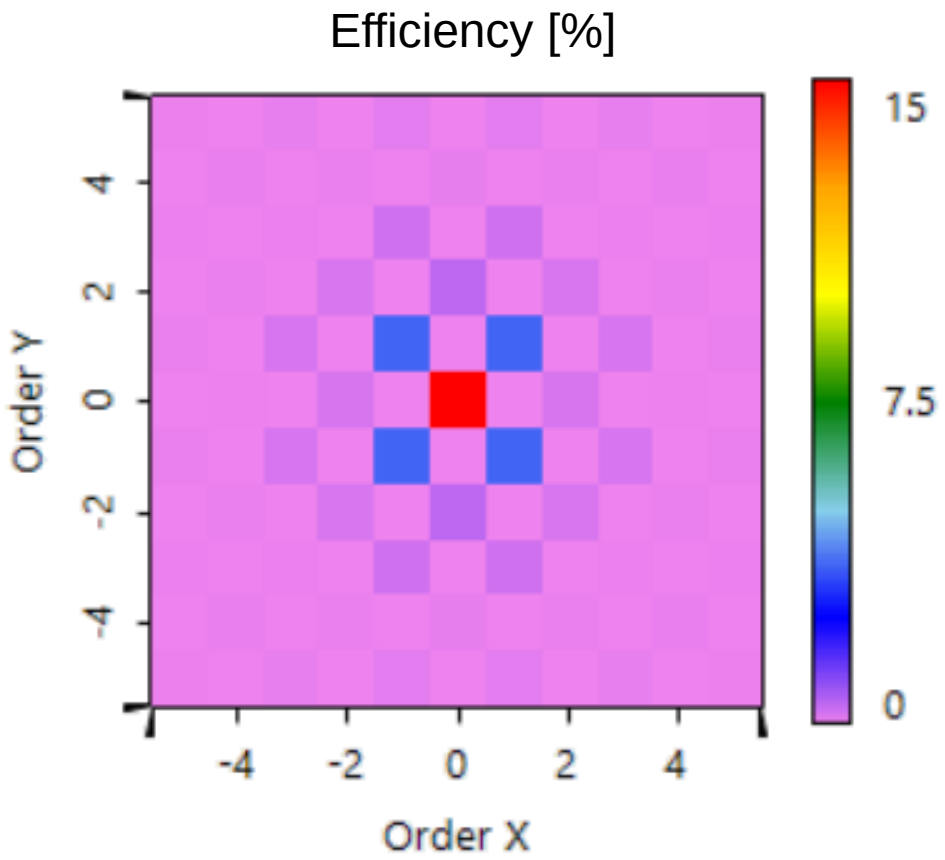
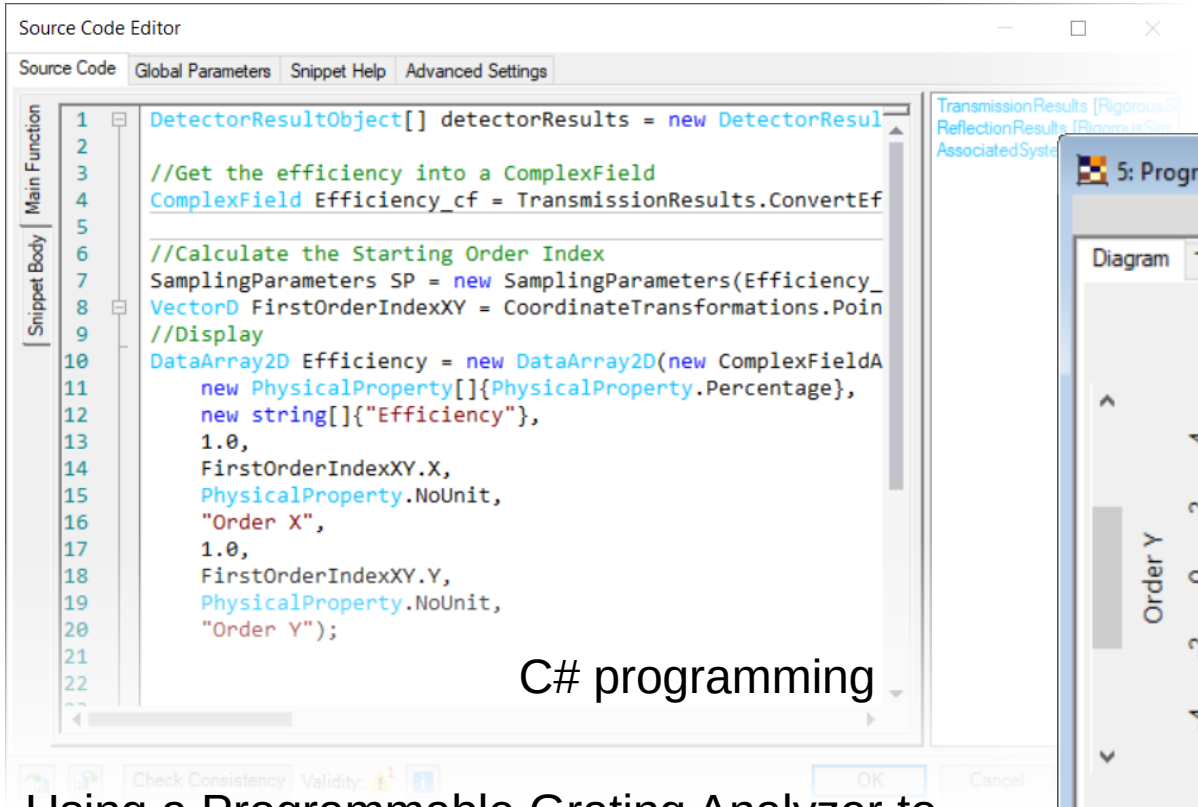
Programmable Grating Analyzer

Abstract

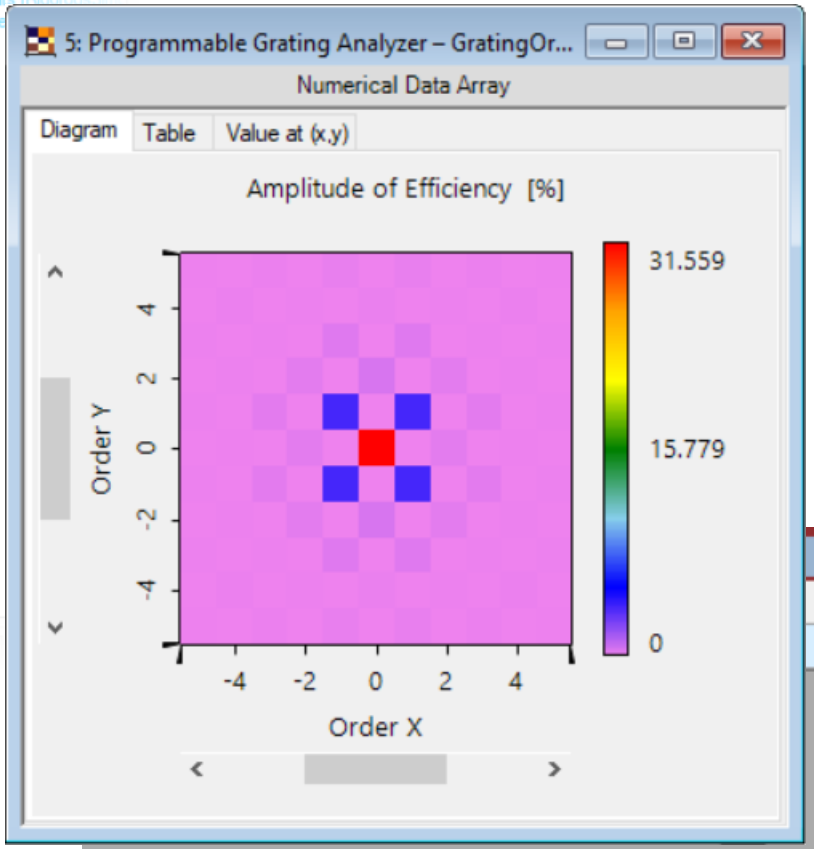


For different grating applications, one may like to access the grating diffraction properties in different manners. Besides the standard Grating Order Analyzer, VirtualLab Fusion provides a fully customizable analyzer. In this example, we show how to access the complete grating diffraction information, to display them, and to use it for further analysis or optimization. We use a pillar grating for illustration and show how to access the result of interest via Parameter Run.

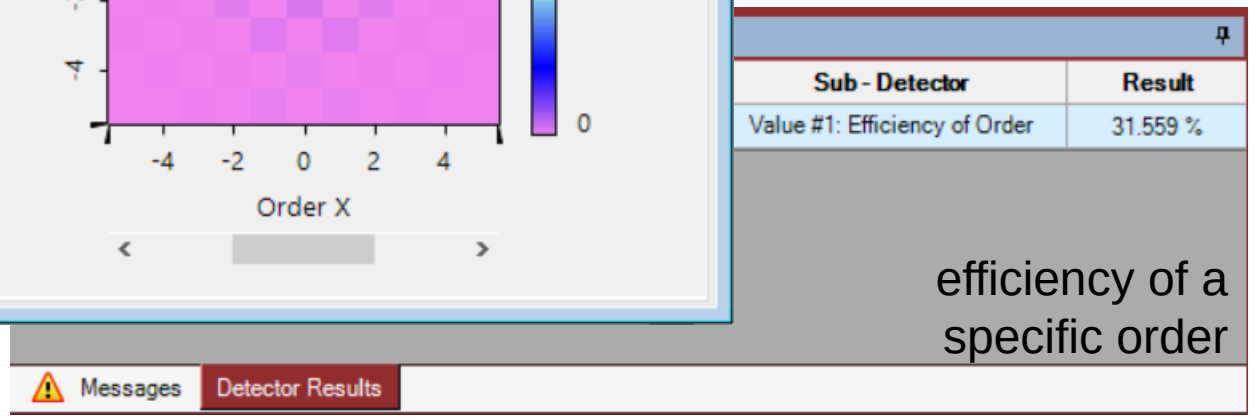
Programming Task



C# programming



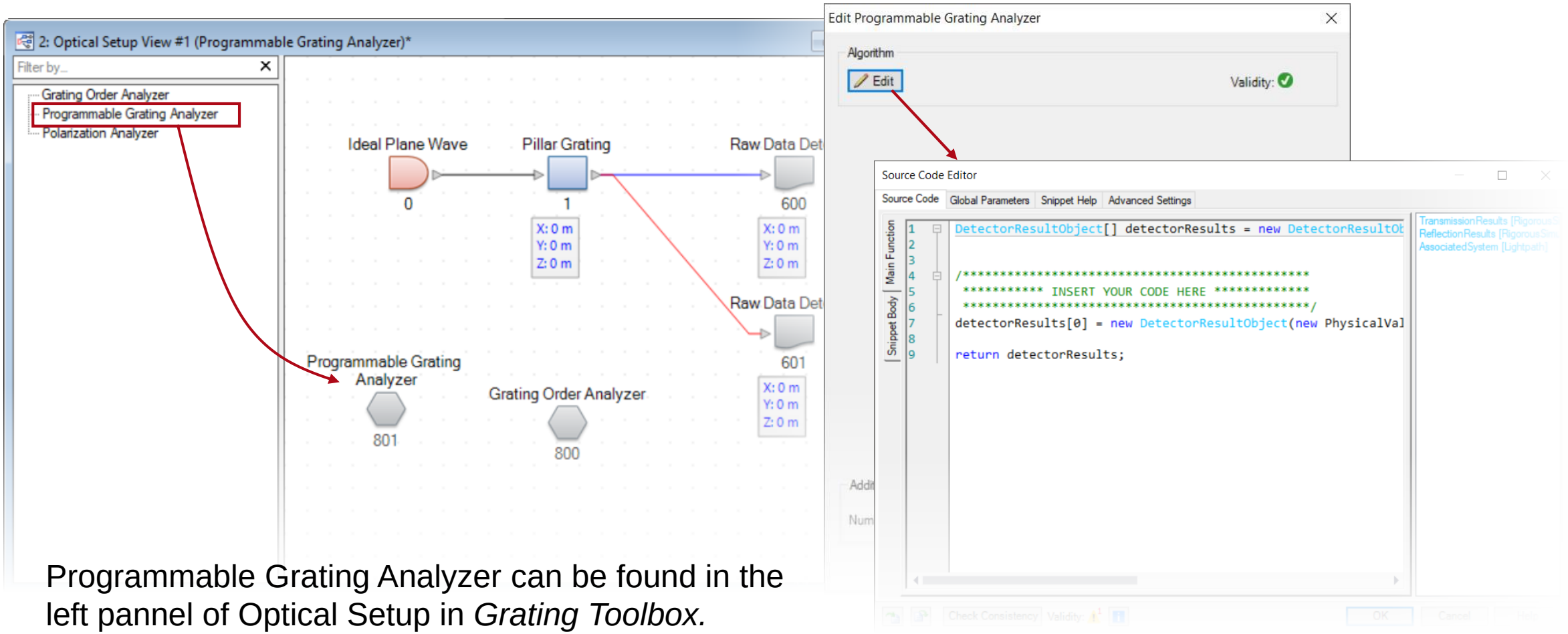
2D data matrix of the efficiencies of all propagating orders



efficiency of a specific order

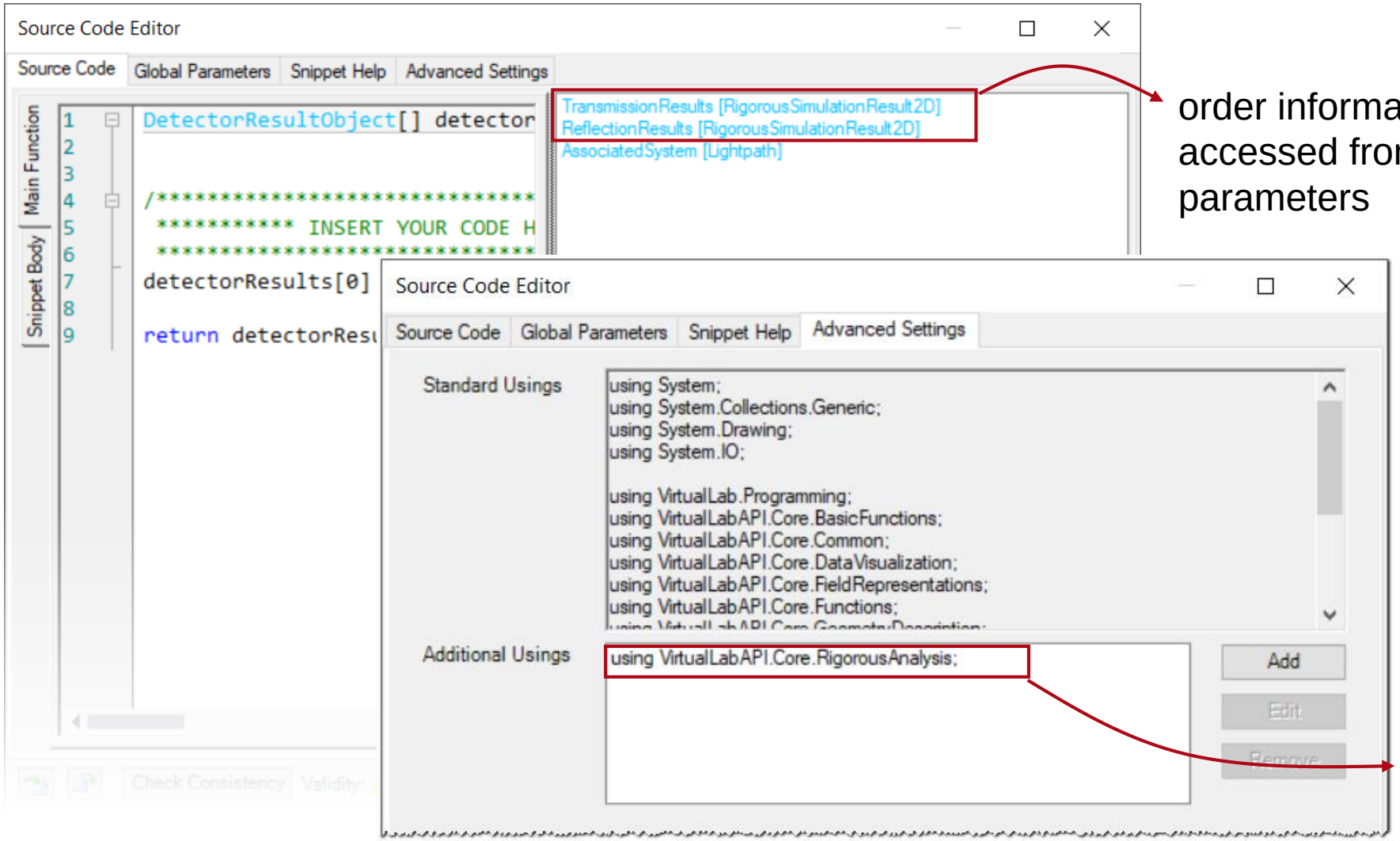
Using a Programmable Grating Analyzer to access the grating diffraction information, and to display the efficiency of a specified order, as well as the efficiencies of all propagating order as a 2D matrix/graph.

Initialization



Programmable Grating Analyzer can be found in the left pannel of Optical Setup in *Grating Toolbox.*

Basic Parameters



order information can be accessed from the two global parameters

additional using must be included here

Efficiency of a Specified Order

C# Code: get the efficiency of a specified order and display of it

```
DetectorResultObject[] detectorResults = new DetectorResultObject[1];

//Get the efficiency of Order (1,1)
OrderInfo Order_11 = TransmissionResults.GetOrder(1, 1);
double Efficiency_11 = Order_11.
```

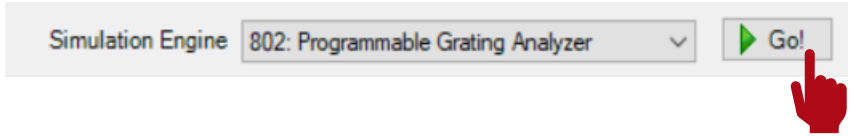
Efficiency
Equals
GetHashCode
GetType
OrderNumber
RayleighCoefficients
ToString

public double Efficiency

well guidance of C# code.

```
//display of efficiency of specific order
detectorResults[0] = new DetectorResultObject(new PhysicalValue(Efficiency_11,              //order efficiency [double]
                                                                PhysicalProperty.Percentage, //property of efficiency[PhysicalProperty]
                                                                "Efficiency of Order (1,1)"), //title of the display[string]
                                              "Programmable Grating Analyzer");               //name of this analyzer[string]


return detectorResults;
```



Detector Results

	Date/Time	Detector	Sub - Detector	Result
1	12/19/2018 14:32:58	Programmable Grating Analyzer	Value #1: Efficiency of Order (1,1)	3.7191 %

Messages | Detector Results

Eifficiencies of All Grating Orders

C# Code: get the efficiency matrix

```
//Get all efficiencies
ComplexField Efficiency_cf = TransmissionResults.ConvertEfficienciesToComplexField();//Convert efficiency into a 2D matrix
```

C# Code: formulate the display matrix with setting the physical properties of x-/y- axis and efficiency

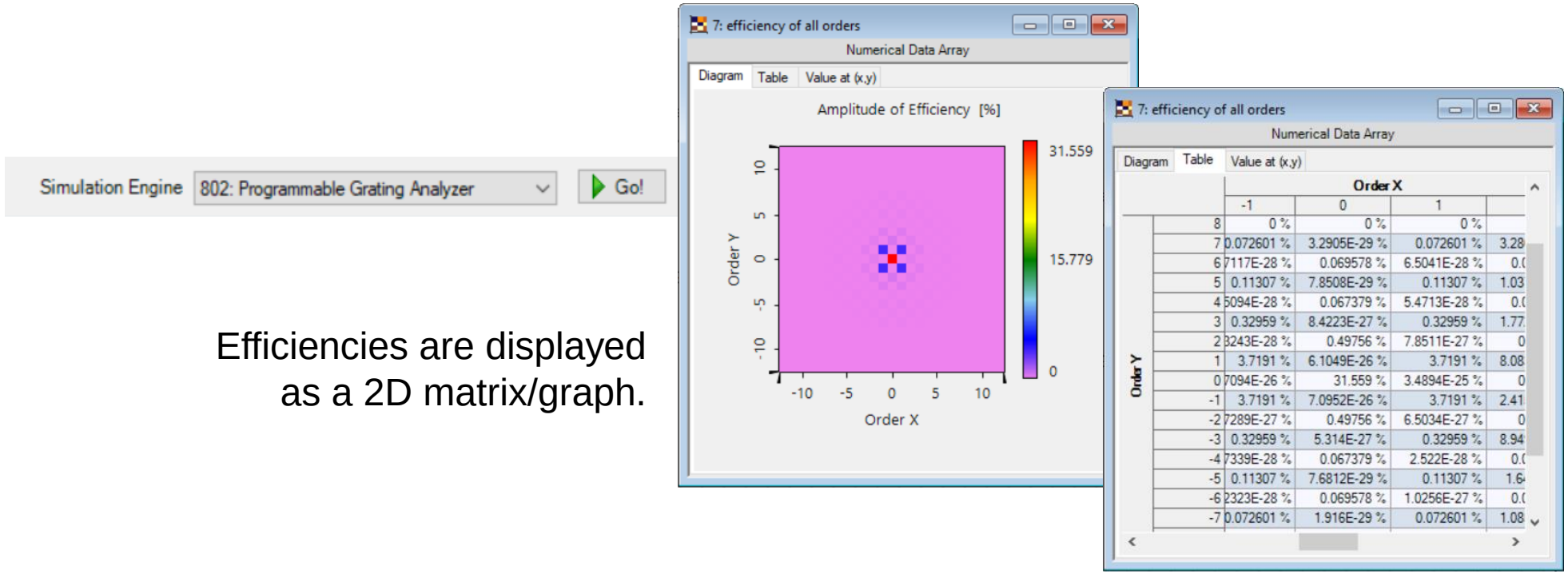
```
//Calculate the index of order (OrderX_min, OrderY_min)
SamplingParameters SP = new SamplingParameters(Efficiency_cf.SamplingPoints, //order numbers
                                               new VectorD(1, 1));// order index is integer, so sampling distance is 1 along both x and y direction
VectorD FirstOrderIndexXY = CoordinateTransformations.PointFromPixelToPhysicalCoordinates(new Vector(0, 0), SP); //(OrderX_min, OrderY_min)

//2D order efficiencies matrix
DataArray2D Efficiency = new DataArray2D(new ComplexFieldArray(Efficiency_cf), //efficiency matrix
    new PhysicalProperty[]{PhysicalProperty.Percentage}, // unit of efficiency is percentage
    new string[]{"Efficiency"}, //"title of the data array"
    1.0,                        //sampling distance of order index is 1 along x direction
    FirstOrderIndexXY.X,        //OrderX_min
    PhysicalProperty.NoUnit,    //index has no unit
    "Order X",                  //label of x-axis
    1.0,                        //sampling distance of order index is 1 along y direction
    FirstOrderIndexXY.Y,        //OrderY_min
    PhysicalProperty.NoUnit,    //index has no unit
    "Order Y");                 //label of y-axis
```

Eifficiencies of All Grating Orders

C# Code: display the efficiency matrix

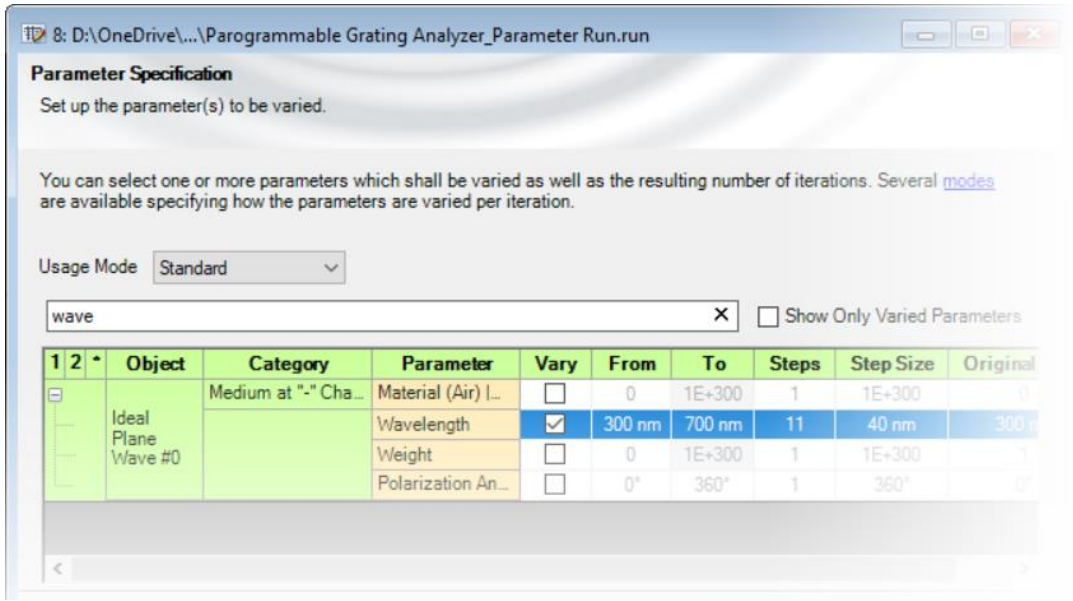
```
//Display the efficiency data array 2D
detectorResults[1] = new DetectorResultObject(Efficiency, "Grating Order Efficiency", "Programmable Grating Analyzer");
```

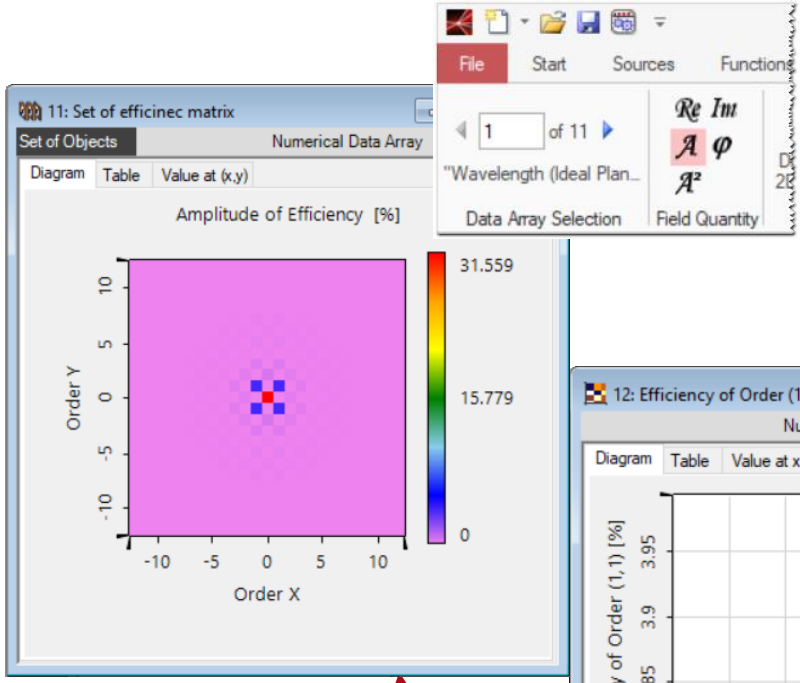


Efficiencies are displayed as a 2D matrix/graph.

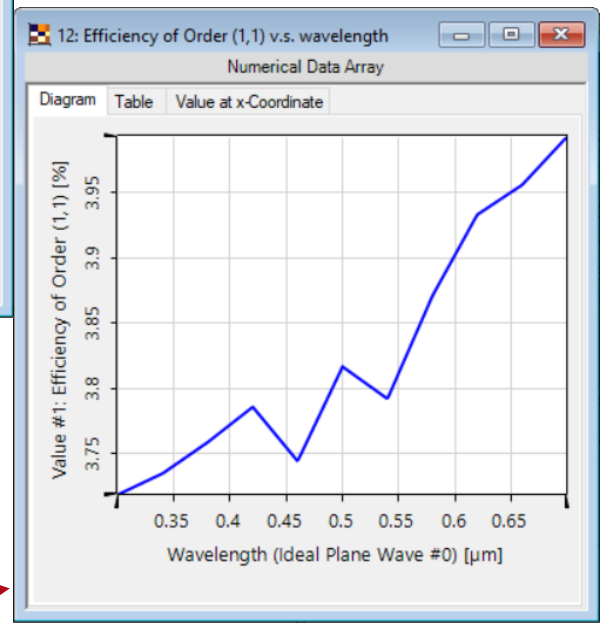
Parameter Run – Varing Wavelength

varing wavelength from 300 nm to 700 nm for 11 steps



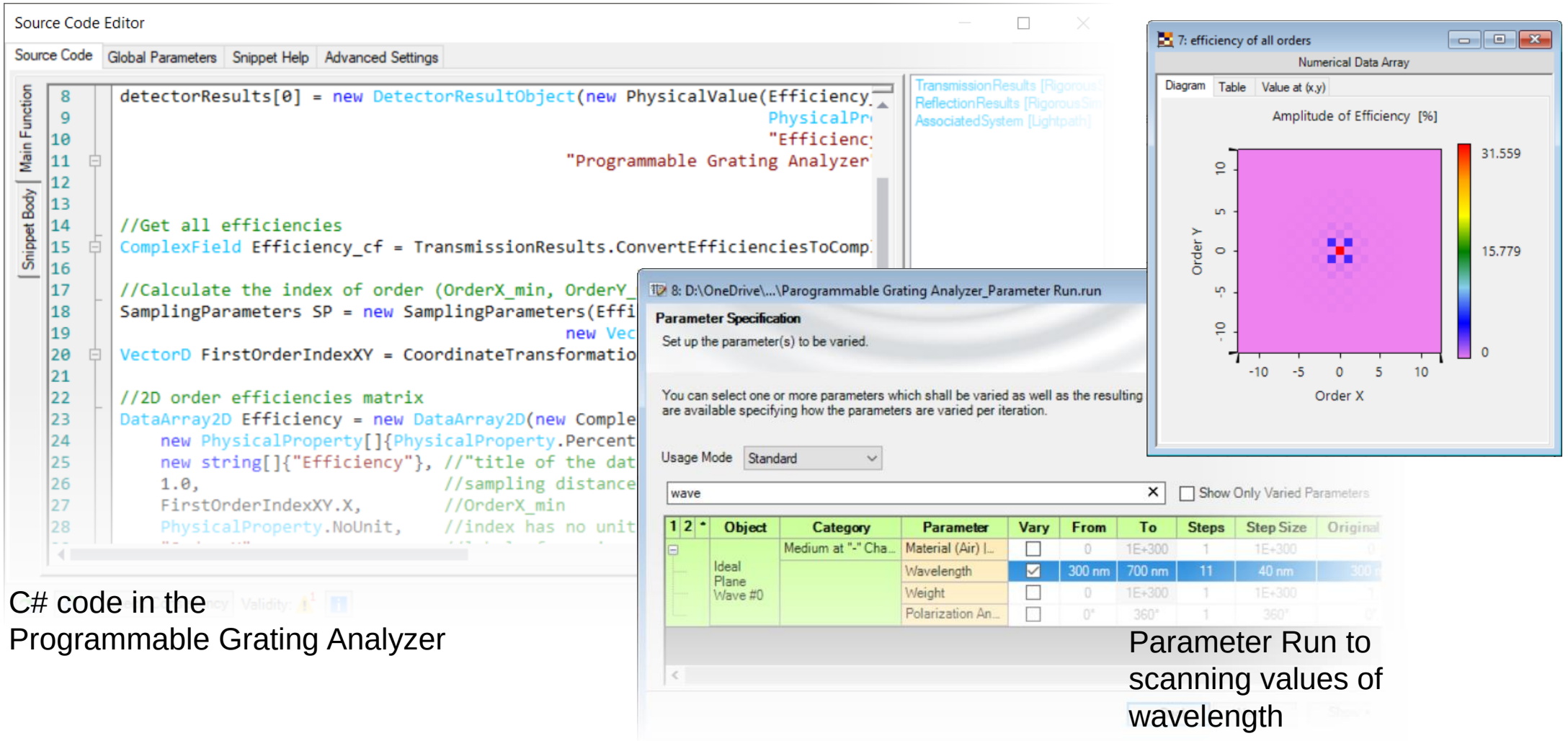


11 layers of 2D data matrix/graph



efficiency (1,1) v.s. wavelength

Peek into VirtualLab



C# code in the Programmable Grating Analyzer

Parameter Run to scanning values of wavelength

Document Information

title	Programmable Grating Analyzer
document code	CZT.0109
version	1.0
toolbox(es)	Grating Toolbox
VL version used for simulations	7.6.1.18
category	Feature Use Case
further reading	- Customized Detector for Lightguide Coupling Grating Evaluation - Source Code Editor