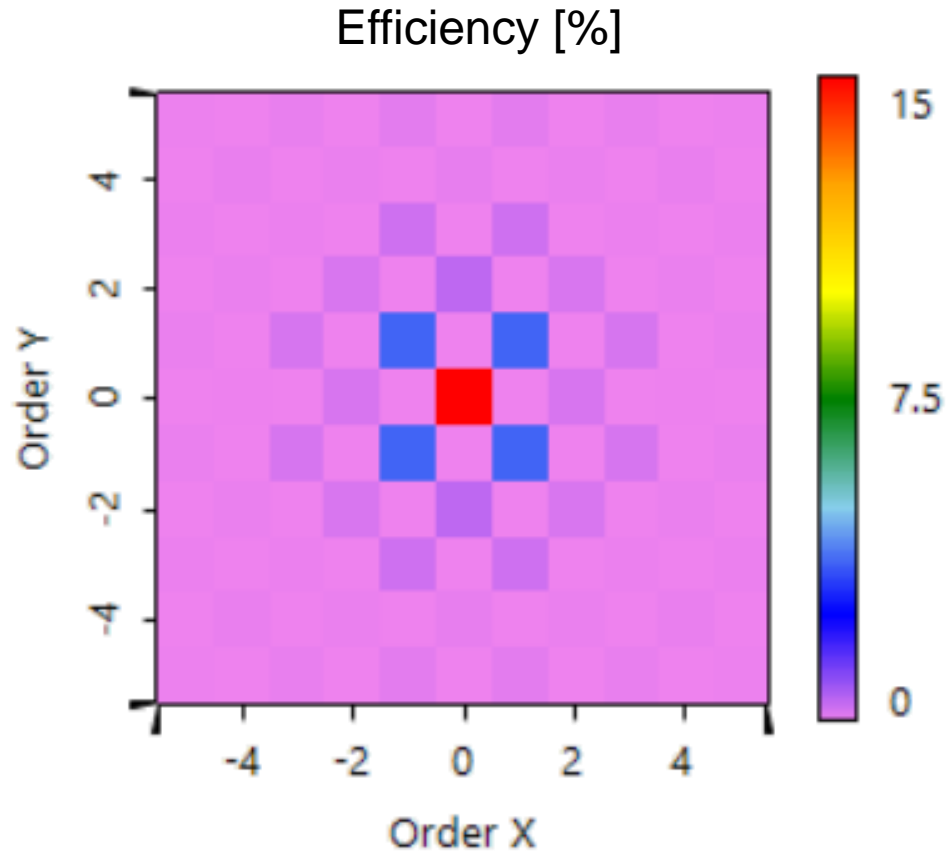


# Programmable Grating Analyzer

# Abstract



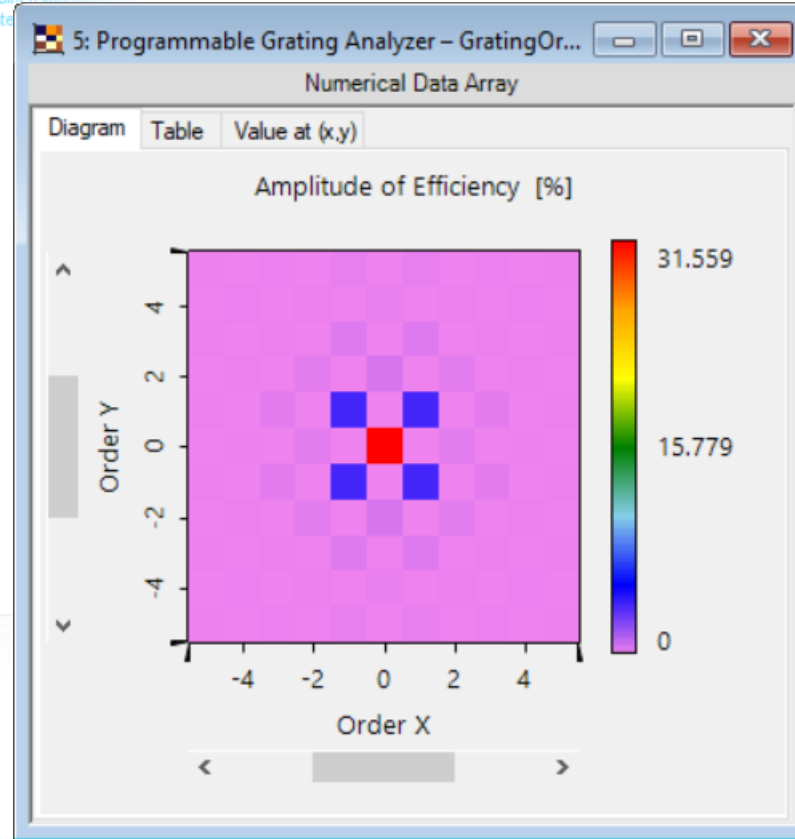
For different grating applications, one may like to access the grating diffraction properties in different manners. Besides the standard Grating Order Analyzer, VirtualLab Fusion provides a fully customizable analyzer. In this example, we show how to access the complete grating diffraction information, 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

```
Source Code Editor
Source Code Global Parameters Snippet Help Advanced Settings

1 DetectorResultObject[] detectorResults = new DetectorResult
2
3 //Get the efficiency into a ComplexField
4 ComplexField Efficiency_cf = TransmissionResults.ConvertEf
5
6 //Calculate the Starting Order Index
7 SamplingParameters SP = new SamplingParameters(Efficiency_
8 VectorD FirstOrderIndexXY = CoordinateTransformations.Poin
9 //Display
10 DataArray2D Efficiency = new DataArray2D(new ComplexFieldA
11     new PhysicalProperty[]{PhysicalProperty.Percentage},
12     new string[]{"Efficiency"},
13     1.0,
14     FirstOrderIndexXY.X,
15     PhysicalProperty.NoUnit,
16     "Order X",
17     1.0,
18     FirstOrderIndexXY.Y,
19     PhysicalProperty.NoUnit,
20     "Order Y");
21
22
```

C# programming



2D data matrix of the efficiencies of all propagating orders

Sub - Detector	Result
Value #1: Efficiency of Order	31.559 %

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.

Messages Detector Results

# Initialization

2: Optical Setup View #1 (Programmable Grating Analyzer)\*

Filter by...

- Grating Order Analyzer
- Programmable Grating Analyzer**
- Polarization Analyzer

Ideal Plane Wave 0

Pillar Grating 1

Raw Data Det 600

Raw Data Det 601

Programmable Grating Analyzer 801

Grating Order Analyzer 800

Algorithm

Validity: ✓

Source Code Editor

```
1 DetectorResultObject[] detectorResults = new DetectorResultObject[...];
2
3
4 /***** INSERT YOUR CODE HERE *****/
5
6 *****/
7 detectorResults[0] = new DetectorResultObject(new PhysicalVal...);
8
9 return detectorResults;
```

Check Consistency Validity

OK Cancel Help

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

# Basic Parameters

The image shows a Source Code Editor window with a code snippet and a Global Parameters dialog box. The code snippet is as follows:

```
1 DetectorResultObject[] detector
2
3
4 /******
5 ***** INSERT YOUR CODE HERE *****
6 *****
7
8 detectorResults[0]
9 return detectorResults
```

The Global Parameters dialog box is open, showing the following content:

Standard Usings

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.IO;

using VirtualLab.Programming;
using VirtualLabAPI.Core.BasicFunctions;
using VirtualLabAPI.Core.Common;
using VirtualLabAPI.Core.DataVisualization;
using VirtualLabAPI.Core.FieldRepresentations;
using VirtualLabAPI.Core.Functions;
using VirtualLabAPI.Core.GeometryDescription;
```

Additional Usings

```
using VirtualLabAPI.Core.RigorousAnalysis;
```

Buttons: Add, Edit, Remove

Annotations:

- A red box highlights the following parameters in the code snippet:

```
TransmissionResults [RigorousSimulationResult2D]
ReflectionResults [RigorousSimulationResult2D]
AssociatedSystem [Lightpath]
```

An arrow points from this box to the text: "order information can be accessed from the two global parameters"
- A red box highlights the line `using VirtualLabAPI.Core.RigorousAnalysis;` in the Additional Usings section of the dialog. An arrow points from this box to the text: "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 public double Efficiency
- Equals
- GetHashCode
- GetType
- OrderNumber
- RayleighCoefficients
- ToString

```
//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;
```

well guidance of C# code.

Simulation Engine 802: Programmable Grating Analyzer

Go!



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

# Efficiencies 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
```

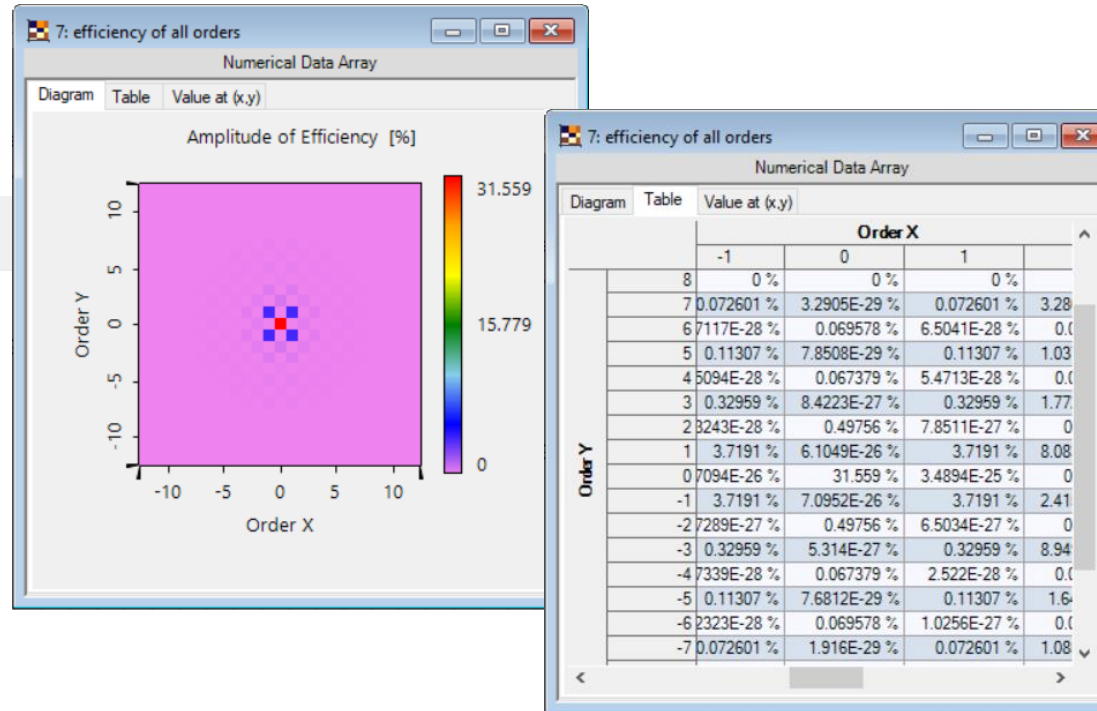
# Efficiencies of All Grating Orders

C# Code: display the efficiency matrix

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

Simulation Engine 802: 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

Parameter Specification

Set up the parameter(s) to be varied.

You can select one or more parameters which shall be varied as well as the resulting number of iterations. Several [modes](#) are available specifying how the parameters are varied per iteration.

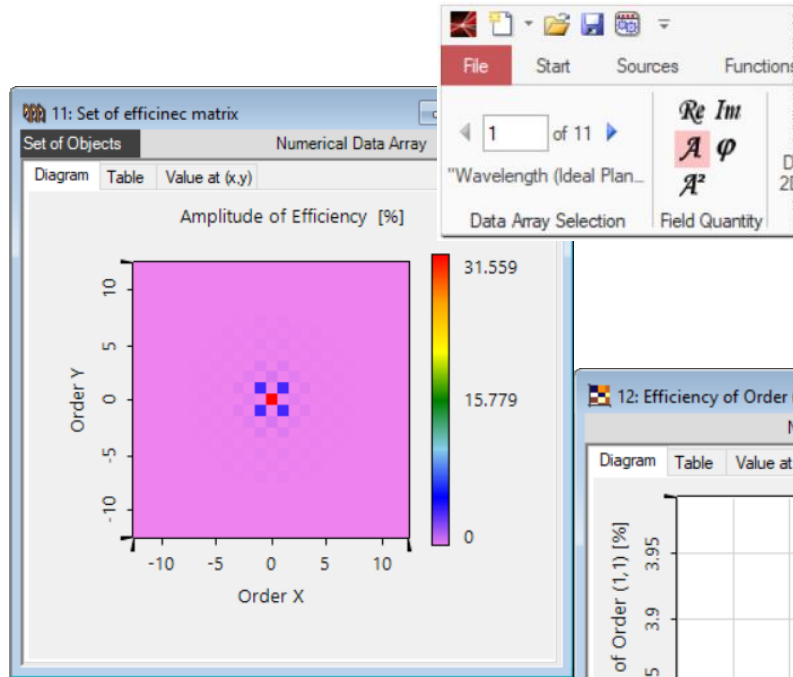
Usage Mode: Standard

wave

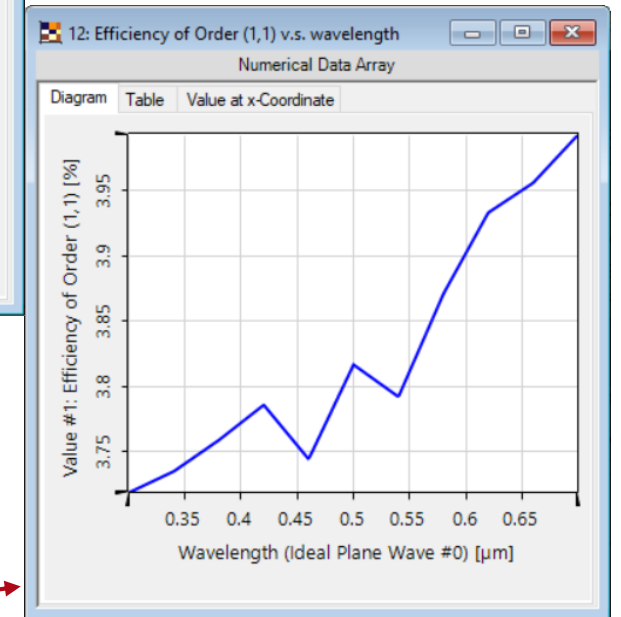
1	2	Object	Category	Parameter	Vary	From	To	Steps	Step Size	Original
		Ideal Plane Wave #0	Medium at "..." Cha...	Material (Air) I...	<input type="checkbox"/>	0	1E+300	1	1E+300	0
				Wavelength	<input checked="" type="checkbox"/>	300 nm	700 nm	11	40 nm	300 nm
				Weight	<input type="checkbox"/>	0	1E+300	1	1E+300	1
				Polarization An...	<input type="checkbox"/>	0°	360°	1	360°	0°

Use Cached Results for Next Run

		Iteration Step			
Detector	Subdetector	Combined Output	8	9	10
Varied Parameters	Wavelength (Ideal Plane W...	Data Array	580 nm	620 nm	660 nm
Grating Order Efficiency	Programmable Grating An...	2D Data Ara	ata Array	2D Data Array	2D Data Array
Programmable Grating An...	Value #1: Efficiency of Ord...	Data Array	3.8711 %	3.9334 %	3.9562 %



11 layers of 2D data matrix/graph



efficiency (1,1) v.s. wavelength

# Peek into VirtualLab

Source Code Editor

```
8 detectorResults[0] = new DetectorResultObject(new PhysicalValue(Efficiency.  
9 PhysicalPr  
10 "Efficiency:  
11 "Programmable Grating Analyzer"  
12  
13  
14 //Get all efficiencies  
15 ComplexField Efficiency_cf = TransmissionResults.ConvertEfficienciesToComp.  
16  
17 //Calculate the index of order (OrderX_min, OrderY_  
18 SamplingParameters SP = new SamplingParameters(Effi  
19 new Vec  
20 VectorD FirstOrderIndexXY = CoordinateTransformatio  
21  
22 //2D order efficiencies matrix  
23 DataArray2D Efficiency = new DataArray2D(new Comple  
24 new PhysicalProperty[] {PhysicalProperty.Percent  
25 new string[] {"Efficiency"}, // "title of the dat  
26 1.0, //sampling distance  
27 FirstOrderIndexXY.X, //OrderX_min  
28 PhysicalProperty.NoUnit, //index has no unit  
--
```

7: efficiency of all orders  
Numerical Data Array  
Diagram Table Value at (x,y)  
Amplitude of Efficiency [%]  
Order Y  
Order X

Parameter Specification  
Set up the parameter(s) to be varied.  
Usage Mode Standard

1	2	Object	Category	Parameter	Vary	From	To	Steps	Step Size	Original
		Ideal Plane Wave #0	Medium at "-" Cha...	Material (Air) [...]	<input type="checkbox"/>	0	1E+300	1	1E+300	0
				Wavelength	<input checked="" type="checkbox"/>	300 nm	700 nm	11	40 nm	300
				Weight	<input type="checkbox"/>	0	1E+300	1	1E+300	1
				Polarization An...	<input type="checkbox"/>	0°	360°	1	360°	0°

Parameter Run to scanning values of wavelength

C# code in the Programmable Grating Analyzer

# 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	<ul style="list-style-type: none"><li>- <a href="#">Customized Detector for Lightguide Coupling Grating Evaluation</a></li><li>- <a href="#">Source Code Editor</a></li></ul>