KALLISTOS

An automatic optimisation tool

OmniSim tutorial

34 Leopold St
Oxford, OX4 1TW
United Kingdom
Phone + 44 1865 324998
Fax + 44 1865 324991
info@lambda-tek.com
www.lambda-tek.com

Distributed By
Photon Design
34 Leopold St
Oxford, OX4 1TW
United Kingdom

Phone +44 1865 324 990
Fax +44 1865 324 991
Email: info@photond.com

CONTENTS

Chapter 1 Introduction

1.1 What is Kallistos?

KALLISTOS is a novel optimisation tool capable of automatically improving existing designs of photonic devices with minimum intervention from the user. Using KALLISTOS, design cycle times of a new device can be dramatically reduced.

Considerable work has gone into all aspects of the product, resulting in a tool incorporating state of the art optimisation algorithms, combined with a powerful graphical user interface making it easy for the user to set up and run a design optimisation calculation, and a comfortable visual interface which allows the user to fully explore the solutions found.

Under the hood, KALLISTOS is equipped with several robust optimisers for the global and local optimisation of continuous functions. Moreover much work has gone into KALLISTOS to make it as efficient as possible. For example, in calculating the sensitivities used by the local optimiser, KALLISTOS takes full advantage of the special structure of the wave equations describing light propagation through photonic devices.

1.2 How it works

In the first iteration of a product design, the user typically starts with an initial guess of a design, such as a linear ring (see the examples). Selected *parameters* and *constraints* defining the structure (e.g. governing the ring shape) are then declared within KALLISTOS, as well as the quantity that needs to be optimised (the *objective function*) e.g. Power Transmission. Finally the calculation is launched with one of the optimisation algorithms available in KALLISTOS.

1.3 The function parser

Although KALLISTOS comes with a variety of predefined *objective functions*, KALLISTOS also comes with a powerful built-in function parser making it possible to define arbitrarily complex parameter constraints and *objective functions.*

1.4 The optimisation algorithms

Finding solutions to complex optimisation problems can be a difficult task. In general there is no optimisation algorithm that works in all cases. KALLISTOS comes with efficient algorithms based on recent advances in mathematical optimisation, each with complementing strengths:

- An efficient local descent routine, ideal for large, computationally intensive structures. An optimum can be found with relatively few iterations.

- A Deterministic global optimisation techniques. This converges more slowly than the above, but is more likely, if not guaranteed, to find a global optimum.

1.5 The monitoring interface

Far from adopting a black box approach, the software comes with a powerful graphical monitoring interface for following the progress of multidimensional optimisation calculations. These prove to be particularly useful for detecting potentially optimal designs when using a global optimiser. For example, the user can quickly locate a point of interest in the parameter space and "home in" on it with a local optimiser.

1.6 Highly speed optimised

Optimisation calculations can be very lengthy for complex, 3D structures. We have exploited to the full, the mathematical structure of the wave equations, for example, by incorporating an analytical procedure for calculation of the sensitivities, essential to the optimisation process. This and other improvements make KALLISTOS an extremely fast optimisation tool for the design of photonic devices.

Chapter

2

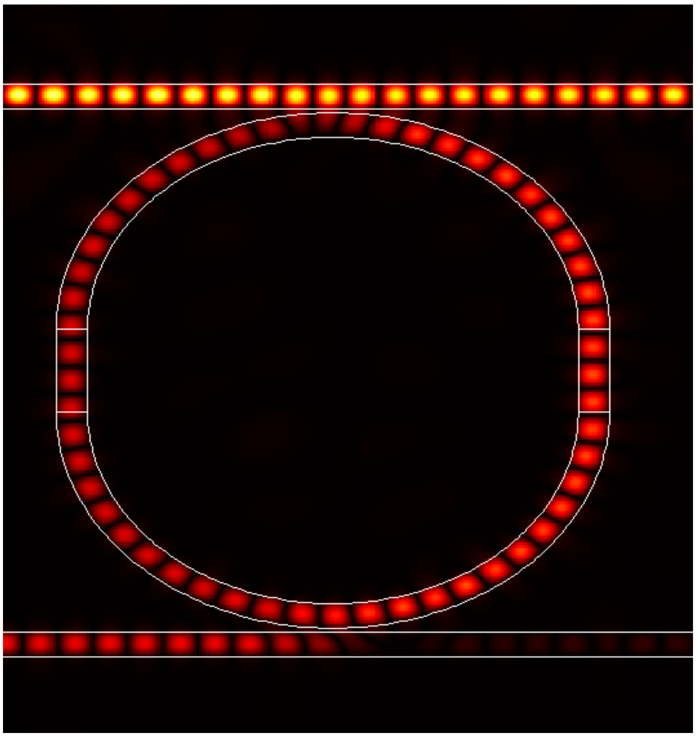
Using Kallistos to tune a 2D ring resonator

This tutorial assumes that you are familiar with OMNISIM. In the example that follows we will use the project "`Learn_Kallistos_OS`.prj", which contains a ready-made OMNISIM Device to be optimised. For the completed examples refer to the file "`Complete_Learn_Kallistos_OS`.prj"

2.1 Introducing the problem

We will now use KALLISTOS to tune a ring resonator so that there is maximum transmission at a chosen wavelength and a maximum extinction ratio.

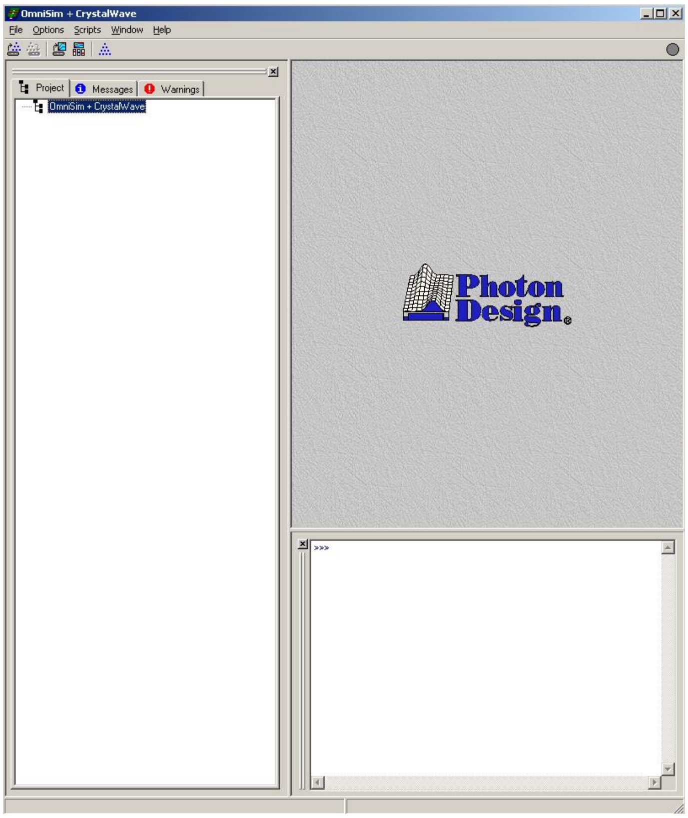
Suppose we have an initial ring resonator structure….



An initial ring resonator structure

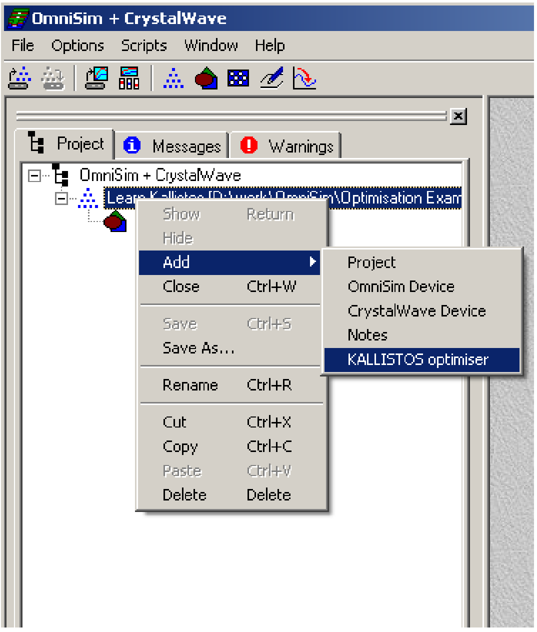
We wish to optimise this structure in order to tune the resonator transmission to a chosen wavelength while maximising the extinction ratio.

- ➢ First open OMNISIM.



OMNISIM

- Within OMNISIM click on the icon and open the project containing the photonic crystal device. (`Learn_Kallistos_OS`.prj)
- Now we need to create a KALLISTOS optimiser. Once the project has been opened right click on the icon in the *project tree*:



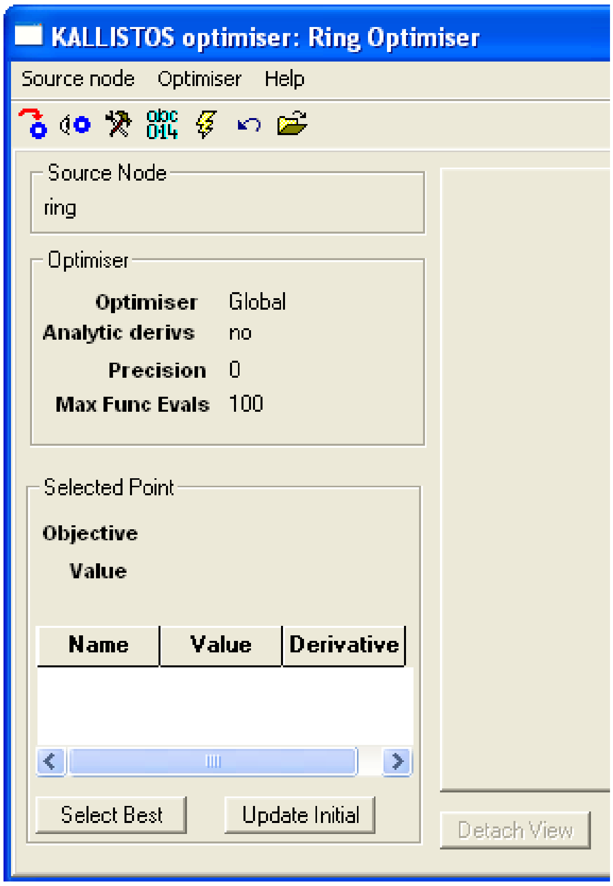
Adding a KALLISTOS Optimiser to the project

- In the popup menu that appears, select /**Add/KALLISTOS optimiser**
- Alternatively you can click on the icon on the toolbar to add an optimiser

➢ In the New Node dialog box that appears type the name "Ring Optimiser" as the name of your optimiser.

A KALLISTOS optimiser node will be added to your project. It will appear as the icon in the *project tree*.

➢ Double click on the optimiser node to open the optimiser. You should see the following window

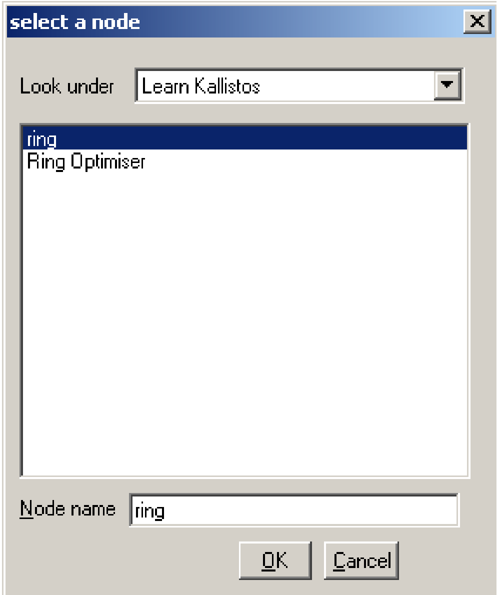


Optimiser Main Window

We want to specify the node "RING" as the source node to be optimised.

➢ Clicking on the toolbar icon will allow you to change the source node within the project that you wish to optimise. Alternatively you can select **source node/set source node** from the top menu

You should see the following dialog window:



Setting the source node

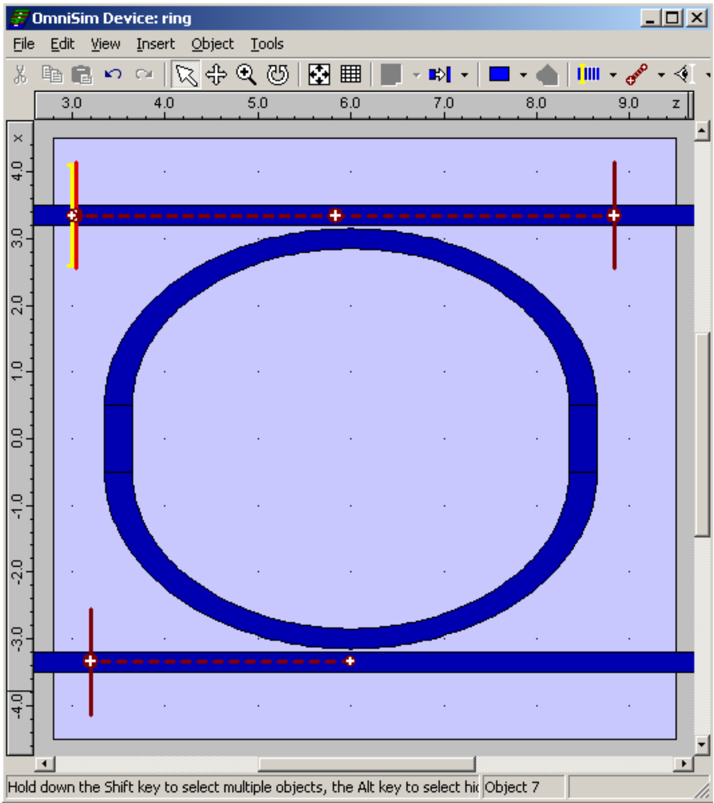
You can set the source node by clicking on an item in the list. The drop down box can be used to select the main node. The items with […] have sub nodes, double clicking on these will allow you select subnode. However in our case we do not need to do this.

- Set the source node to be "ring" in the text box and press the **OK** button

We can view the node that we have chosen to optimise. This will allow us to check that we have the correct node.

- Click on the view source node tool on the toolbar in the Main Window. This will show the node to be optimised. Alternatively you can select **/source node/view source node** from the top menu

You should see the initial device prior to the optimisation process as shown below

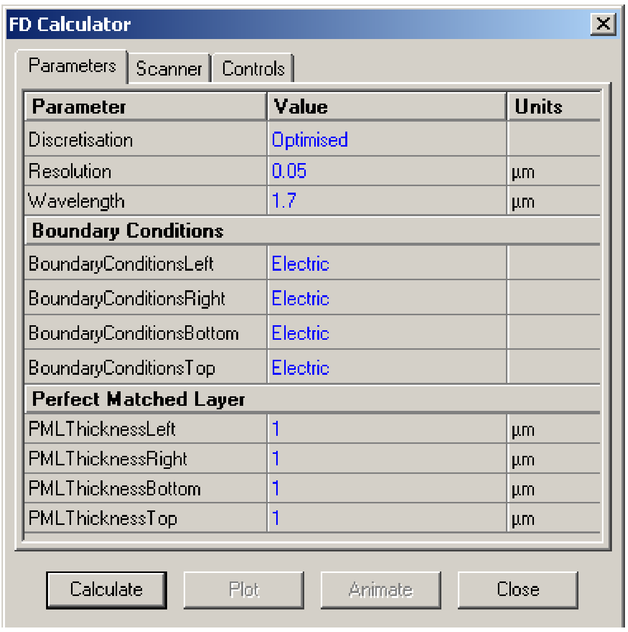


The initial design in OMNISIM

In this example, the waveguides have a refractive index of 3.2; (the background is air). This is currently set up for a 2D simulation

- Select the **Tools/Calculator/FD Calculator…** menu item (or, if the only engine option you have is the FD option, **Tools/FD Calculator…**).

The FD Calculator will appear:



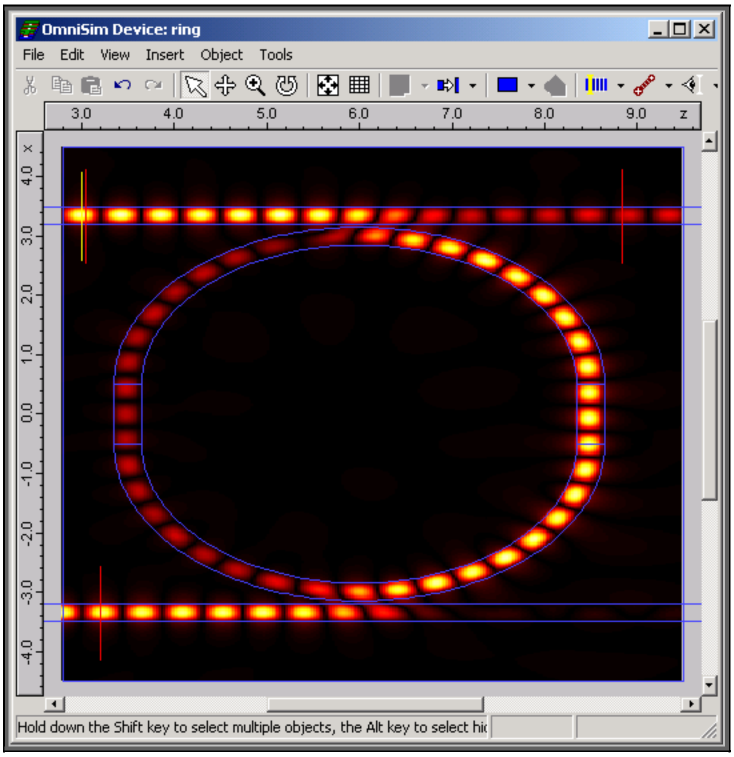
The FD Calculator

Unlike the FDTD calculator this is a Frequency Domain tool and as such, a working wavelength needs to be specified. Here it is set to be 1.7um.

- Press **Calculate**

After only a few seconds the FD Calculator has solved the system. Notice that the FD Calculator is extremely fast. (This is why it can be very powerful when used in conjunction with the KALLISTOS optimiser. hundreds of iterations can be performed in an hour!)

- On the Controls panel, make sure **Hy** is selected from the drop-down choice at the top, the radio button to the left of the drop-down choice is selected, and the **real** radio button is selected.

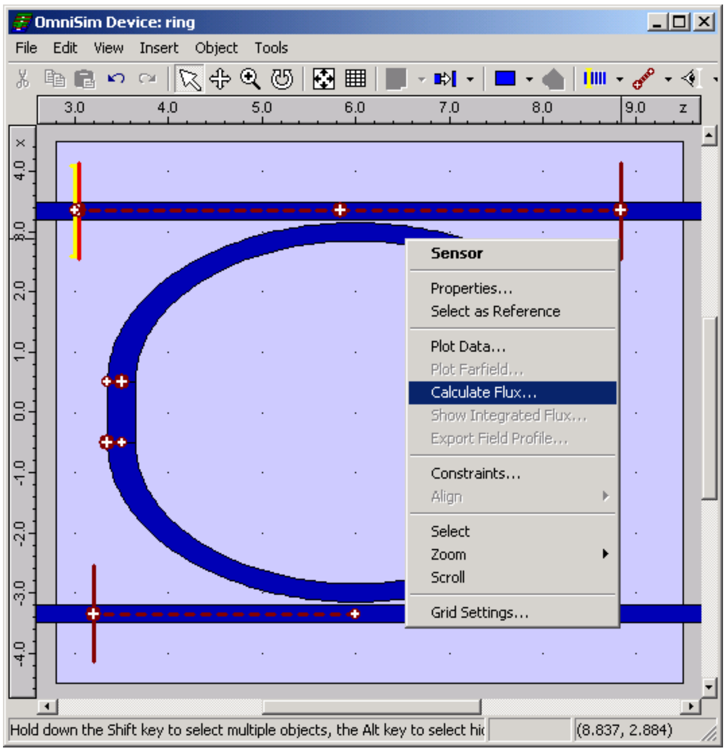


Hy field plot of the non-optimised ring

It is obvious from the field plot that the light is shared between the throughput and the drop.

➢ Close the FD Calculator

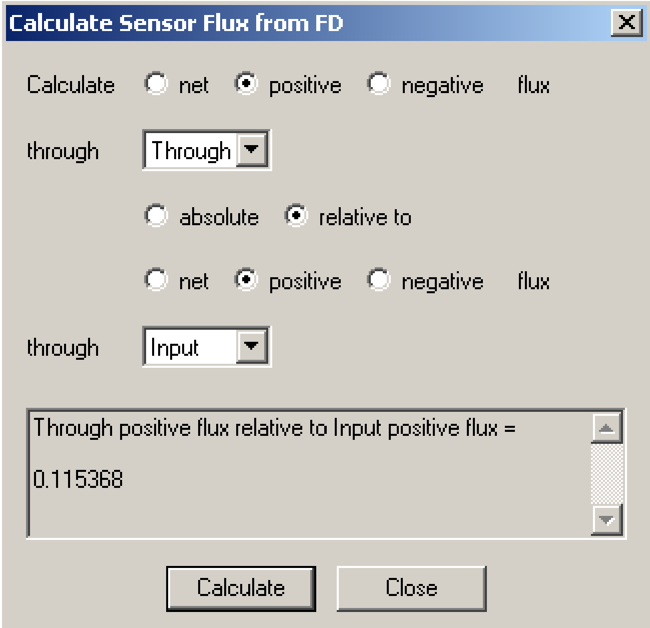
➢ To find out how much is transmitted along the input waveguide, right-click on the sensor in the top-right corner and select **/Calculate Flux…**



Selecting Calculate Flux…

➢ In the **Calculate Sensor Flux from FD** dialog that appears, press **Calculate**

The sensor flux will be calculated. Note that the positive flux (i.e. the throughput) is around 12%.



The calculated Sensor Flux from FD

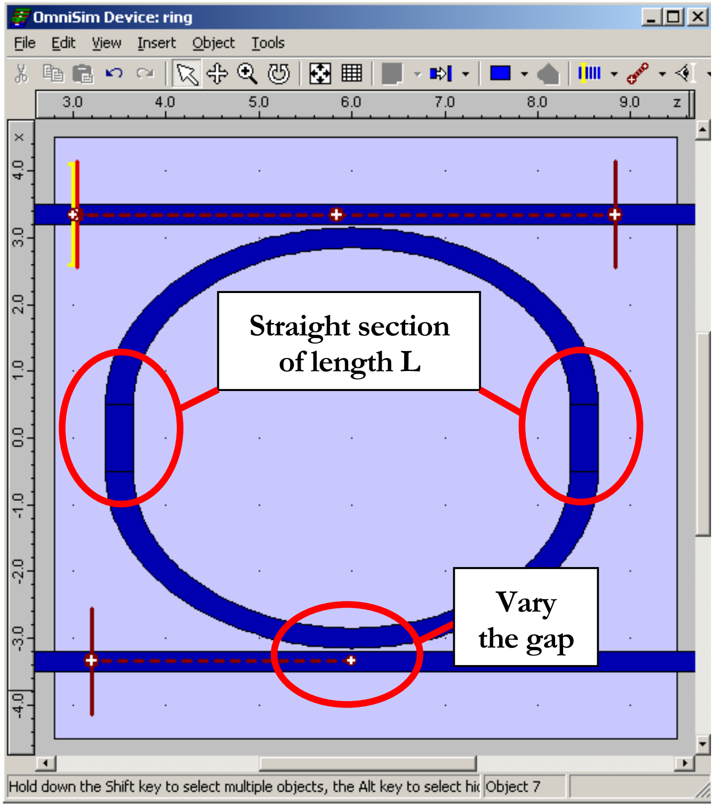
Similarly, by right-clicking on the sensor in the bottom left corner and selecting **/Calculate Flux…**, you can show that the negative flux for the drop is around 88%.

In the following optimisation we will maximise the drop at 1.7um and maximise the throughput at 1.72um, thus maximising the extinction ratio.

2.2 Parameterising the ring resonator

In order to maximise the transmission at a 1.7um and the extinction ratio, we are going to tune different parameters that we first need to introduce in the structure.

1) We wish to optimise the drop for our chosen wavelength by varying the path length around the ring. This is done by inserting a straight waveguide section of variable length between the two arcs.
2) We wish to optimise the extinction ratio by varying the gap between the waveguides and the ring.

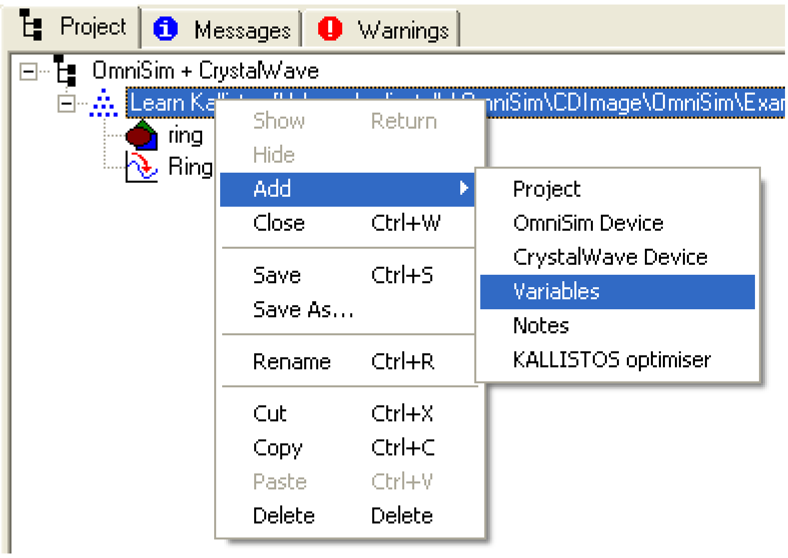


The two variables in the optimisation

Variable Name	Function
length	The length of the two straight waveguide sections inserted between the two arcs that describe the ring.
gap	The distance from the input and output waveguides to the ring.

The named variable system is ideal for parameterising the taper.

➢ Right-click on the project icon to add a new *Variables* node:



➢ Type "Ring Variables" for the *Variables* node name.

➢ Open the “Ring Variables” *Variables* node from the *Project Tree* by double clicking on it.

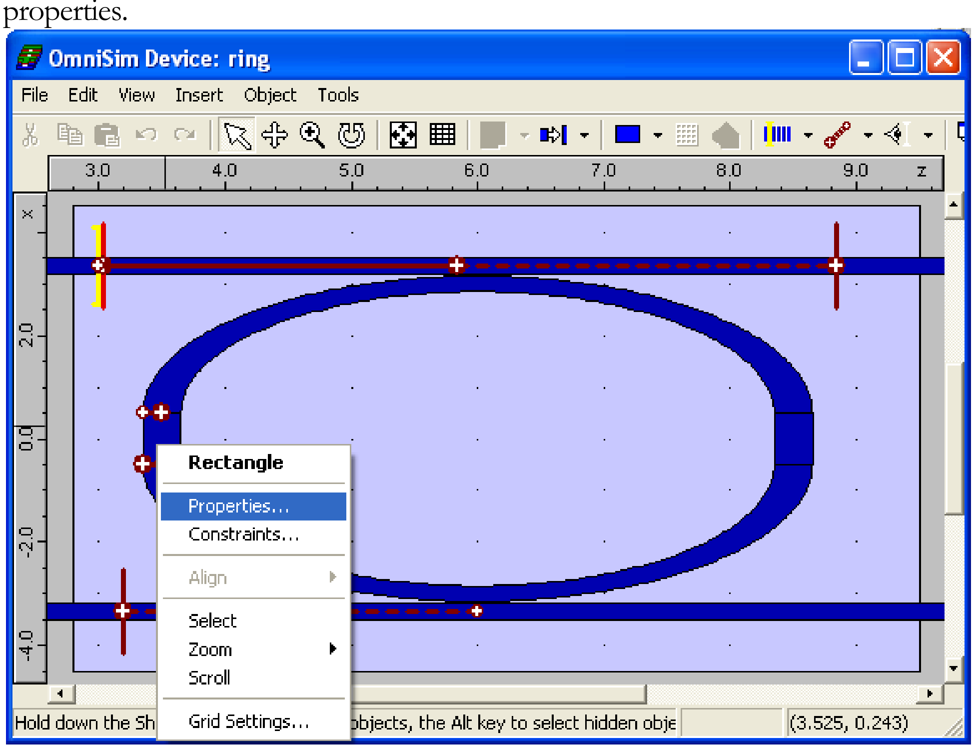
➢ The position of the input and output guides are determined not only by the gap parameter, but the size of the ring, hence we need to create variables for them that will be function of the other ones. Add 4 new *Variables*, named *length, gap, positionXinput and positionXoutput.*

➢ Set their values according to the following window.

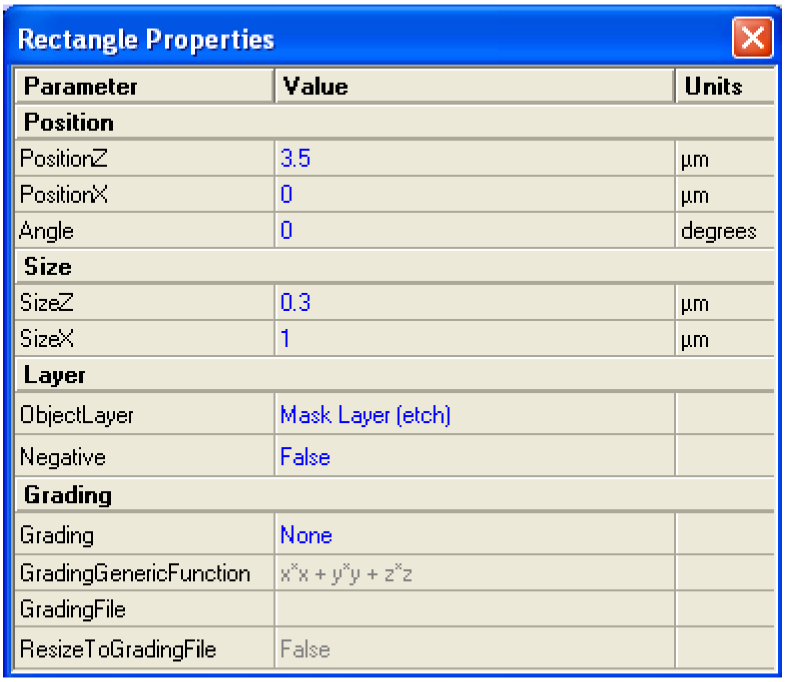
The next step is to change the device.

➢ Open the OmniSim Device “ring” by double clicking on it at the Project Tree.

➢ Right click on one of the straight section in the middle of the ring to edit its properties.



You should see the following window:



Rectangle Properties

Parameter	Value	Units
Position		
PositionZ	3.5	µm
PositionX	0	µm
Angle	0	degrees
Size		
SizeZ	0.3	µm
SizeX	1	µm
Layer		
ObjectLayer	Mask Layer (etch)	
Negative	False	
Grading		
Grading	None	
GradingGenericFunction	x*x + y*y + z*z	
GradingFile		
ResizeToGradingFile	False	

- In front of *SizeX*, type *length*. As shown below:

Size		
SizeZ	0.3	µm
SizeX	length	µm

- Close the window and repeat the operation with the other straight section.
- Now right click on the input waveguide to edit its properties and set *PositionX* to "positionXinput". Close the window and repeat with the operation for the output waveguide, setting *PositionX* to "positionXoutput".
- You should see the x-position of each of waveguide change as you set up these parameters.

The structure is now fully parameterised, and we can start setting up the optimiser.

2.3 Setting up the optimiser

In order to run our optimisation we will need to set the appropriate settings for our problem.

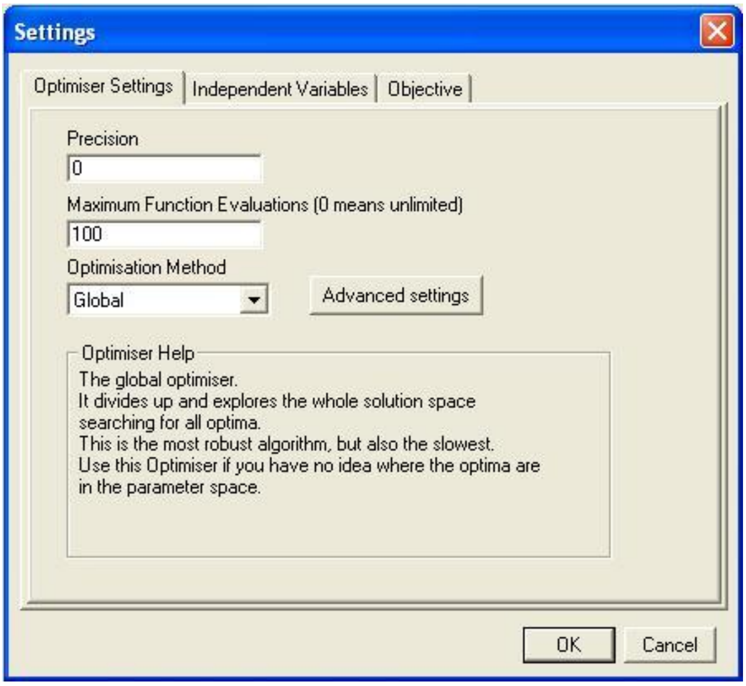
- Open the node of the optimiser.
- Click on the settings  tool on the toolbar in the main window. Alternatively you can select **Optimiser/Settings** from the top menu

A dialog should appear with three tabbed panels.

2.3.1 Choosing the Optimiser

- In the *Optimiser Settings* tab, set the **Optimisation Method** to *Global*. We do not know beforehand what the optimum configuration might be, therefore we want to search the whole of the solution space so this method is ideal. (The Help box tells you a little information about the optimiser that you have selected to use.)
- Set the **Precision** to *0*.
- We want to set the **Maximum Function Evaluations** to 100 so that the run continues until this number of iterations is completed.

You should now have the following:



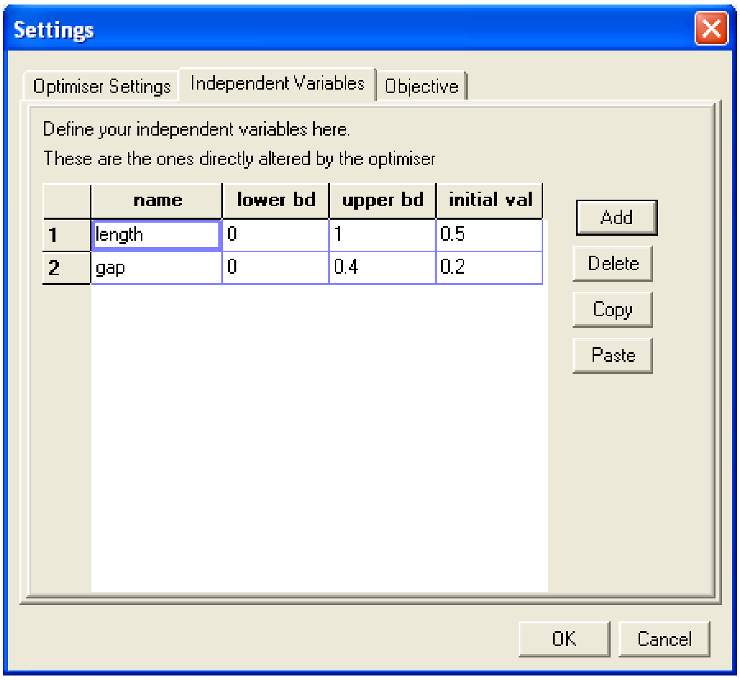
The Settings Panel for the ring optimisation

2.3.2 Setting up the Independent Variables

Next, we set up the *Independent Variables* for our problem. These are the variables that the optimiser will vary.

This optimisation will have two *Independent Variables*, which are *length* and *gap*.

- Using the **Add** button, insert these parameters into the table, giving them the lower, upper bounds and initial values as shown below.

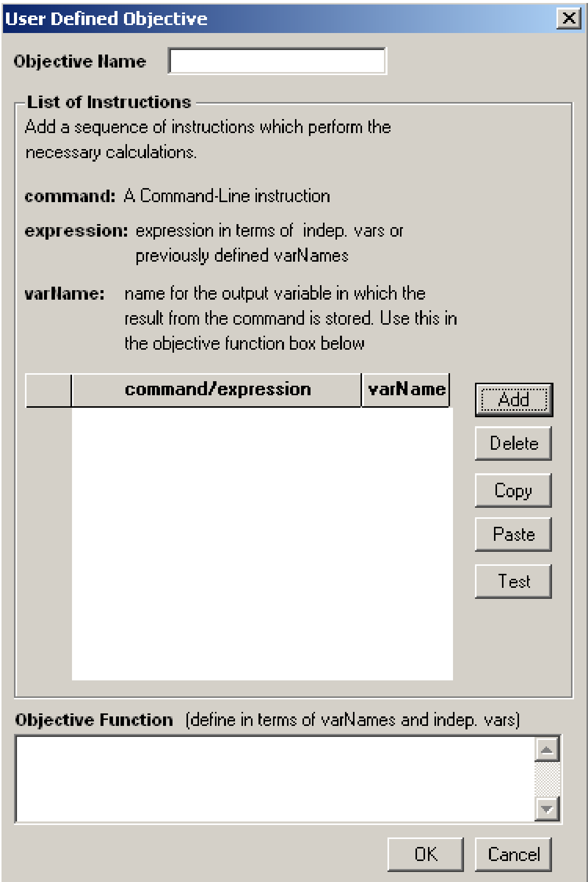


Setting up the independent variables for the ring optimisation

2.3.3 Choosing the Objective Function

Once we have finished setting up the *Independent Variables* we need to define the objective function that we wish to optimise.

- Click on the *Objective* tab.
- We need to create a User Defined Objective function. So Click on the **New...** button. You should see the following.



- First, we should give the Objective function a name. Type findpeak-droplambda in the **Objective Name** box.

We wish for the optimiser to send the following commands for each iteration.

1. Set wavelength = 1.70 um
2. Run the FD Calculator
3. Get the positive flux from the throughput sensor
4. Get the negative flux (flux going from right to left) from the drop sensor.
5. Change wavelength to 1.72 um
6. Repeat steps 2,3,4

- These commands can be found by performing the actions via the GUI. Open the FD Calculator and set the wavelength to 1.72. Note that in the Command-Line Window these three commands have been generated:

app.subnodes[1].subnodes[1].fsdevice.fdparameters.startchange()

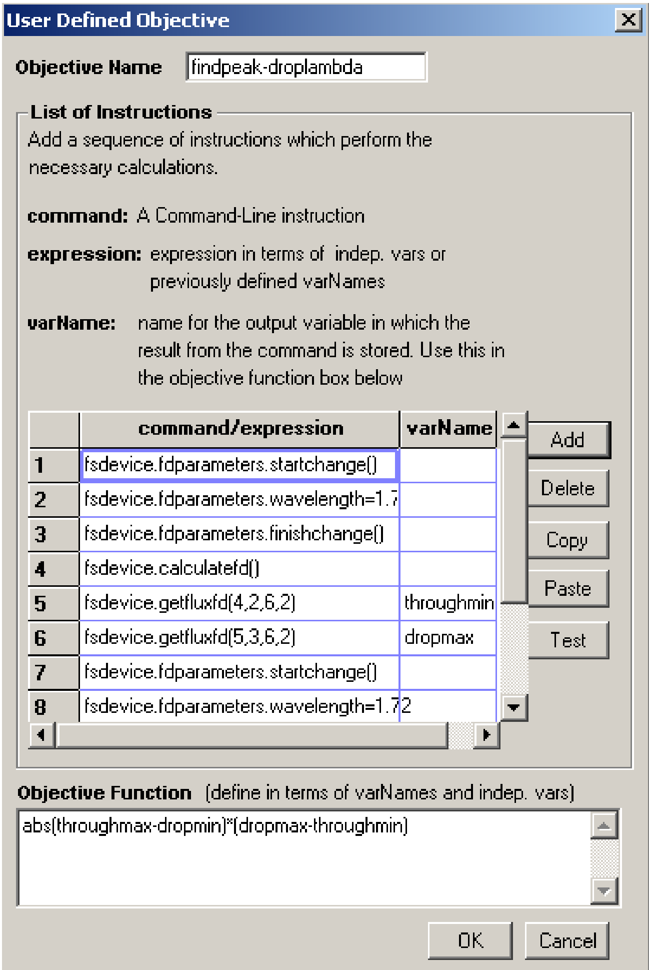
app.subnodes[1].subnodes[1].fsdevice.fdparameters.wavelength=1.72

app.subnodes[1].subnodes[1].fsdevice.fdparameters.finishchange()

However, in the KALLISTOS dialog we only need to give the command from the source node root (i.e. we can remove the *app.subnodes[1].subnodes[1].* – the position of the source node in the project tree). Thus the commands we must enter into the Settings Panel are:

fsdevice.fdparameters.startchange()
fsdevice.fdparameters.wavelength=1.7
fsdevice.fdparameters.finishchange()
fsdevice.calculatefd()
fsdevice.getfluxfd(4,2,6,2)
fsdevice.getfluxfd(5,3,6,2)
fsdevice.fdparameters.startchange()
fsdevice.fdparameters.wavelength=1.72
fsdevice.fdparameters.finishchange()
fsdevice.calculatefd()
fsdevice.getfluxfd(4,2,6,2)
fsdevice.getfluxfd(5,3,6,2)

➢ Insert these commands using the **Add**, **Delete**, **Copy** and **Paste** buttons accordingly. You should see the following.

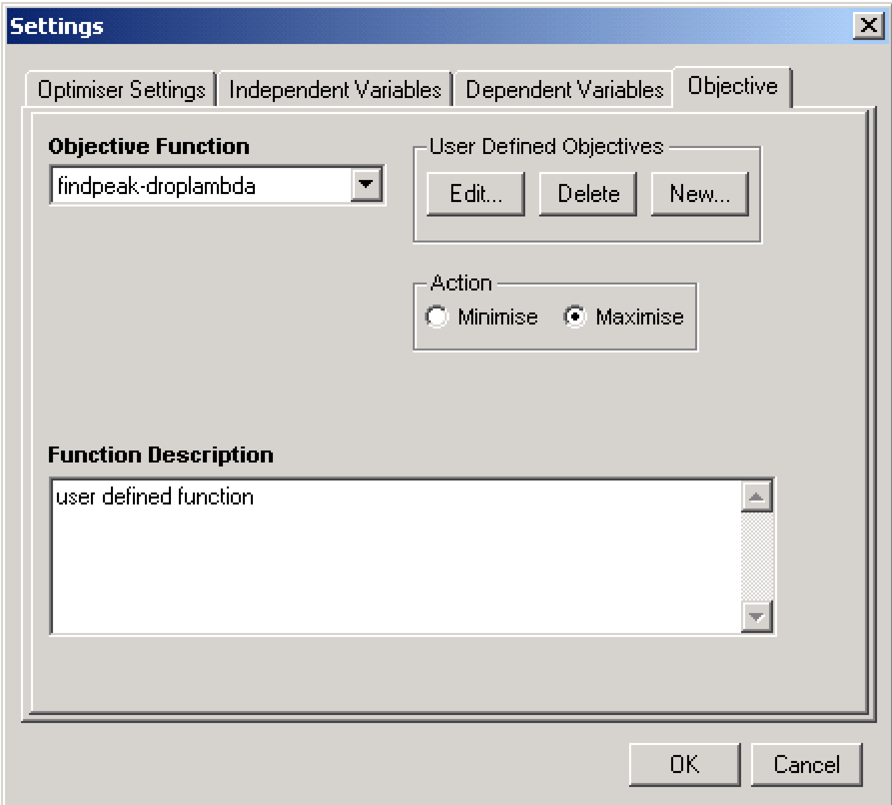


Notice that we have also assigned the names throughmin, dropmax, throughmax and dropmin for the flux values. We will use these for defining our Objective function.

We wish to minimise the throughput (maximise the drop) at 1.7µm while maximising the throughput (minimising the drop) at 1.72µm. Thus an appropriate objective function is:

abs(throughmax-dropmin)(dropmax-throughmin)*

- Type this into the **Objective Function** box
- Click **OK**.
- We wish to maximise this objective function so make sure **Action** is set to **Maximise.** You should see the following:



Setting the objective function for the ring optimisation

We have now set all the settings that we need for the optimisation.

- Click the **OK** button on the settings dialog to accept all these settings.

We are now ready to run the optimiser.

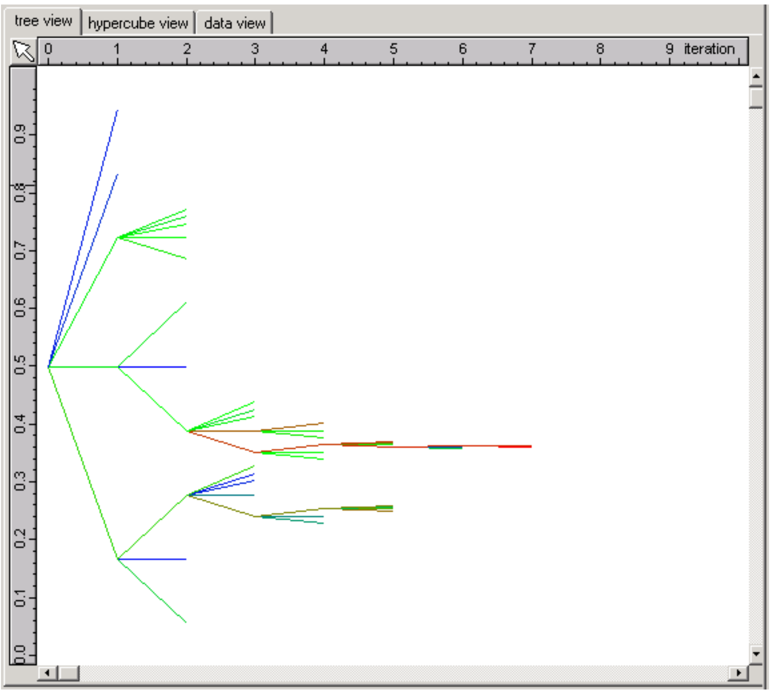
2.4 Running the optimiser

- Click on the Run tool  on the toolbar in the main window (or select **/Optimiser/Run** from the top menu).

The optimiser will start to run. The green dial in OMNISIM will turn. As the optimiser runs it will update the graphical views, so you should be able to gauge its progress as it runs.

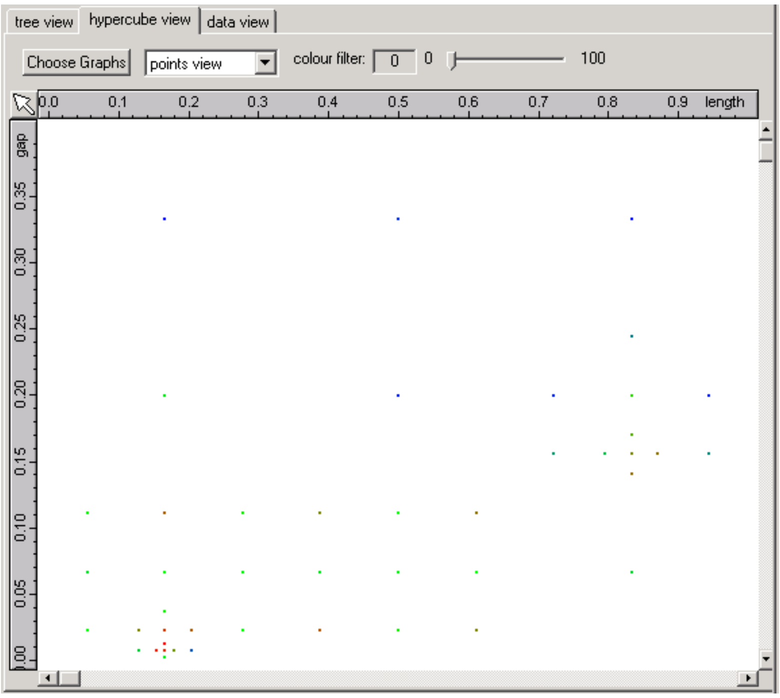
- You can monitor the progress of the optimisation by looking at the various tabbed views. If click on the *data view* you can follow the numerical quantities themselves.
- Now click on the *Tree view* tab.

The *Tree view* display is a graphical representation of how the global optimiser systematically divides up and explores the solution space. Each branch represents a different region in the parameter space. The longer the branches, the more the optimiser tried to explore there, so these tend to correspond to local minima. The line colour indicates how good the function value is. It goes from blue for the worst function value, through green, to red for the best one. So the red shoots represent the better local optima. The longest red shoot is the global optimum found so far. For example, on completion of the optimisation you should see the following view:



The Tree View for the ring optimisation

- Click on the *hypercube view* to see the optimisation evolving through the parameter space.



The Hypercube View for the ring optimisation

On the *hypercube view* we can see the local optima. Again the colours vary from blue for the worst function values, through green, to red for the best ones. Notice the clustering of points. This corresponds to a local minima that the optimiser explored in more detail. Indeed this gives an indication of how the optimiser works: it explores the entire parameter space, but preferentially around the "better" points found so far. As a consequence the clusters will correspond to the "interesting regions" in the parameter space. Again, you can click on these to view the local solution.

- You can look at the various *views* simultaneously by clicking on the **Detach View** button. This will put the currently selected *view* into a separate window. Click on the **x** button on the top right to put the *view* back in a tabbed window.
- Clicking on any data point in any *view* will highlight the corresponding data point in the **Selected Point** box and all other *views*.
- You can also inspect the best point found so far by pressing **Select Best**. This will also highlight the point in all the *views*.

Note that you can stop the optimisation at any time by clicking the running symbol in the OMNISIM main window. Clicking the run tool again will continue the run from where it left off.

The FD Engine is extremely fast, so each iteration takes only 30 seconds or so to evaluate. Thus the whole optimisation – 100 points in two-parameter space – should take around 1 hour to complete.

2.5 Analysing the results

Looking at the final result

Once the optimisation has finished, we can analyse the optimisation process and look at the optimum solution that the optimiser has found. KALLISTOS provides graphical views in order to assist the user in analysing the optimisation process.

- Click on the **Select Best** button of the main window. The point with best function value will be highlighted in all views.

On the *tree view* the selected point is highlighted with a large black dot.

On the *hypercube view* the selected point is highlighted in each graph with a large black dot.

On the *data view* the selected row corresponding to the selected point is highlighted in blue.

The information concerning this point will be displayed in the **Selected Point** table on the left-hand panel of the main window as shown below.

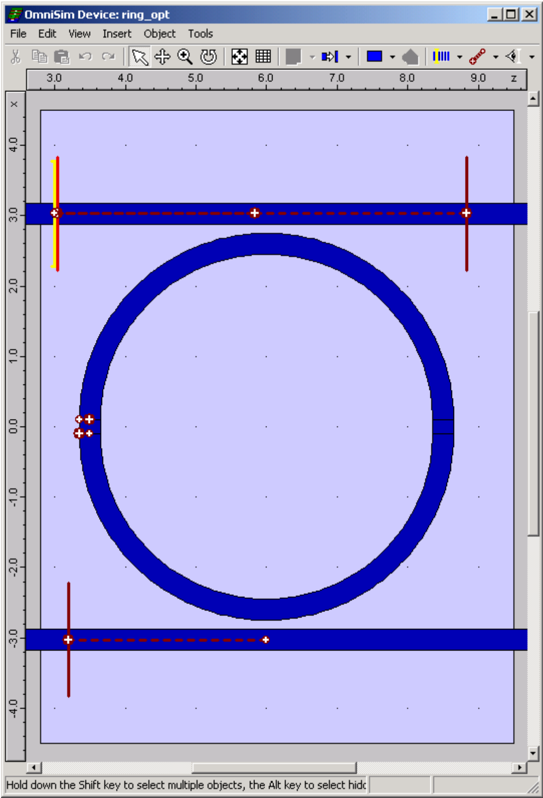
Name	Value	Derivative
throughmin	0.011536	
dropmax	0.93647	
throughmax	0.93301	
dropmin	0.062921	
length	0.82922	
gap	0.16049	

Selecting the best point

The corresponding values for each of the parameters are also listed for this solution in the table.

For this example, it is not obvious how "good" this optimum is. To get a clearer idea of what this solution looks like we can view it in OMNISIM as we did for our starting solution.

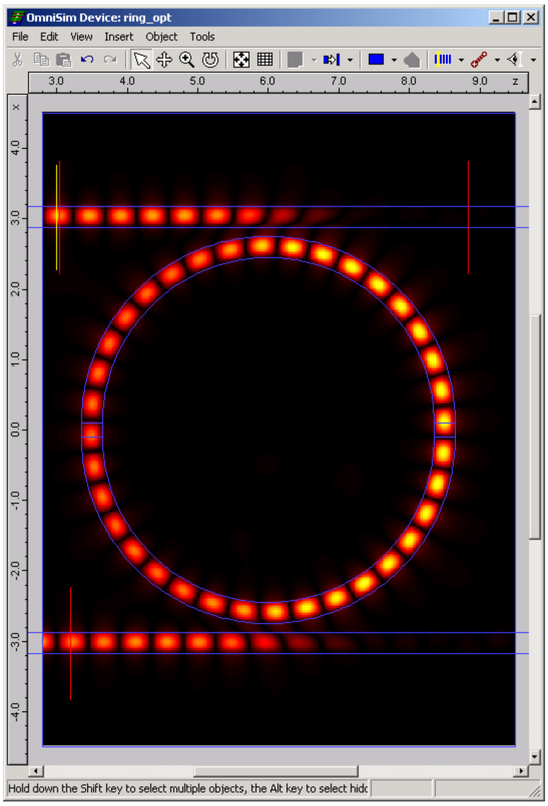
- Click on the **Update Initial** button of the main window. The node is updated to match the optimal value.
- Double click on the node (if not already displayed).



Viewing the optimal device

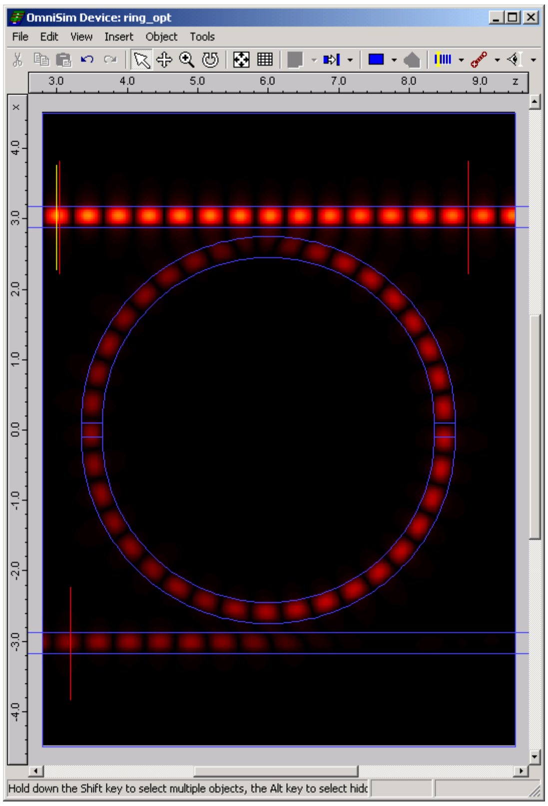
- Select **Tools/Calculator/FD Calculator…** from the menu of the node.
- On the FD Calculator dialog that appears, press **Calculate**
- On the Controls panel, make sure **Hy** is selected from the drop-down choice at the top, the radio button to the left of the drop-down choice is selected, and the **real** radio button is selected.

You should get the following:



The field intensity for the optimal device at 1.72µm

On repeating the calculation with the Wavelength set to 1.7µm, you should get the following:



The field intensity for the optimal device at 1.7µm

The optimiser has therefore managed to find an structure that drops a large fraction of power at 1.72µm, and very little at 1.7µm. The extinction ratio has been maximised.

Looking at the intermediate results

Here we have used a deterministic algorithm, meaning that if you run it long enough, it is **eventually** guaranteed to find an optimum. This is in contrast with stochastic type algorithms (e.g. genetic, evolutionary, simulated annealing strategies) which may at times prove quicker, but are by no means guaranteed to find the global optimum. This algorithm follows a systematic splitting strategy: it divides all the areas of the parameter space, but sub divisions happen more frequently in those areas deemed more likely to have an optimum. Eventually all points in the parameter space are covered, which is why it is deterministic.

One significant advantage that this method has over genetic or evolutionary strategies is that all history is stored. Thus allowing us to look at all the local minima that have been found in the solution space.

- Again click on the tab headed *Data view*
- Click on the Fvalue tab at the top of the table – this will re-order the iterations from best to worst.
- Click on the list and use the arrow keys to scroll down the list while making sure that you can see the node at the same time. You will see that there are many different "close-to-optimal" configurations that you may wish to consider in regard to some other criteria (E.g. easier to fabricate).