lumerical
illuminating the way

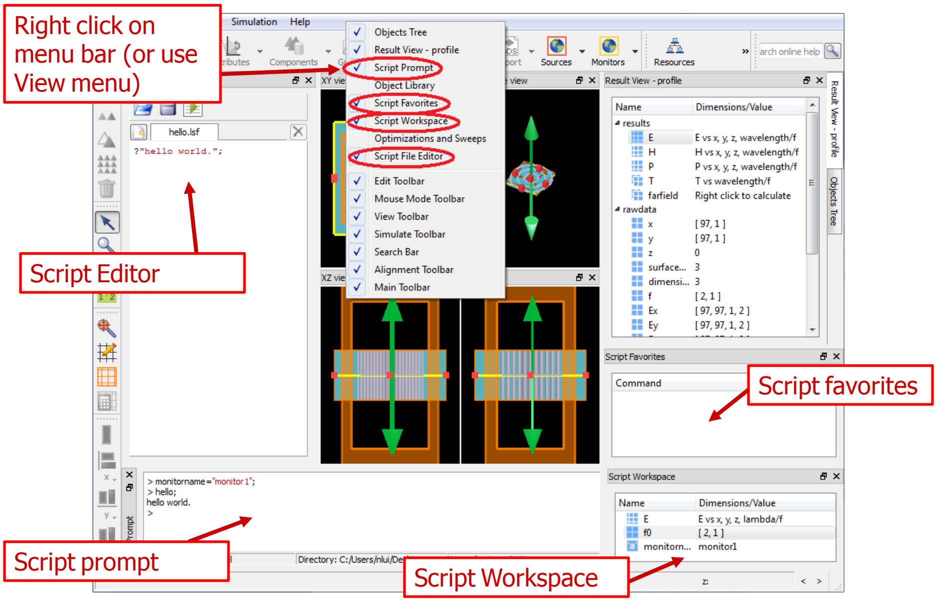
FDTD *Solutions*

Lumerical's Scripting Environment

Outline

- Lumerical's scripting environment
  - Basic mathematics and data visualization
  - Setting up a simulation
  - Running and analyzing simulations
- Creating structure and analysis groups
- Parameterizing your simulation
- Script files and example

Advanced analysis with scripting



Right click on menu bar (or use View menu)

Script Editor

Script favorites

Script prompt

Script Workspace

```
?"hello world.";
```

```
> monitorname="monitor1";
> hello;
hello world.
>
```

Result View - profile

Name	Dimensions/Value
results	
E	E vs x, y, z, wavelength/f
H	H vs x, y, z, wavelength/f
P	P vs x, y, z, wavelength/f
T	T vs wavelength/f
farfield	Right click to calculate
rawdata	
x	[97, 1]
y	[97, 1]
z	0
surface...	3
dimensi...	3
f	[2, 1]
Ex	[97, 97, 1, 2]
Ey	[97, 97, 1, 2]

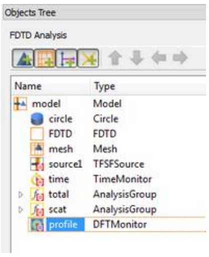
Script Favorites

Command

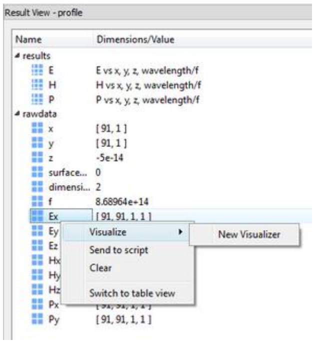
Script Workspace

Name	Dimensions/Value
E	E vs x, y, z, lambda/f
f0	[2, 1]
monitorn...	monitor1

Basic analysis – Results manager



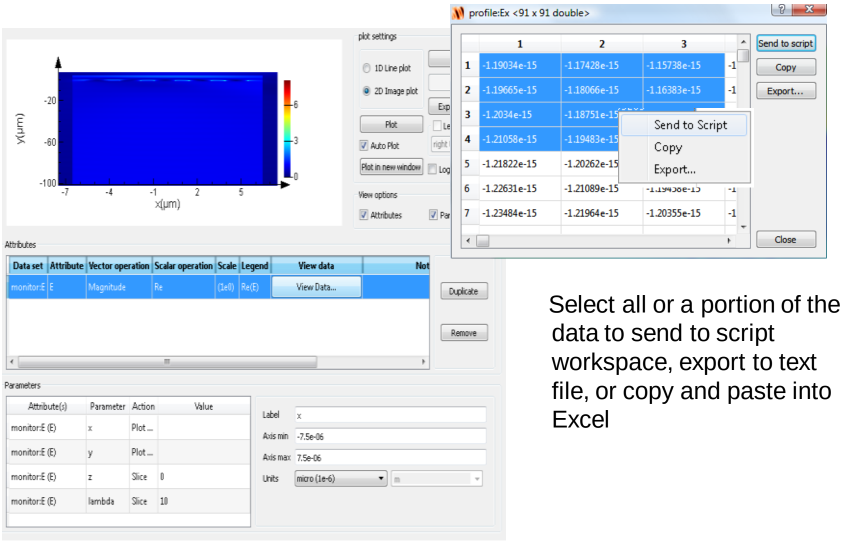
Name	Type
model	Model
circle	Circle
FDTD	FDTD
mesh	Mesh
source1	TFSFSource
time	TimeMonitor
total	AnalysisGroup
scat	AnalysisGroup
profile	DFTMonitor



Name	Dimensions/Value
results	
E	E vs x, y, z, wavelength/f
H	H vs x, y, z, wavelength/f
P	P vs x, y, z, wavelength/f
rawdata	
x	[91, 1]
y	[91, 1]
z	-5e-14
surface...	0
dimensi...	2
f	8.68964e+14
Ex	[91, 91, 1, 1]
Ey	
Ez	
Hx	
Hy	
Hz	
Px	[91, 91, 1, 1]
Py	[91, 91, 1, 1]

- Results View window shows all results for the selected simulation object
- Icons used to distinguish between matrix results, datasets and strings
- Right-click on any result to visualize the data or send the result to the script workspace

Basic analysis - Visualizer



profile:Ex <91 x 91 double>

	1	2	3	
1	-1.19034e-15	-1.17428e-15	-1.15738e-15	-1
2	-1.19665e-15	-1.18066e-15	-1.16383e-15	-1
3	-1.2034e-15	-1.18751e-15	[illegible]	[illegible]
4	-1.21058e-15	-1.19483e-15	[illegible]	[illegible]
5	-1.21822e-15	-1.20262e-15	[illegible]	[illegible]
6	-1.22631e-15	-1.21089e-15	-1.19458e-15	-1
7	-1.23484e-15	-1.21964e-15	-1.20355e-15	-1

Attributes

Data set	Attribute	Vector operation	Scalar operation	Scale	Legend	View data	Not
monitor:E	E	Magnitude	Re	(1e0)	Re(E)	View Data...	

Parameters

Attribute(s)	Parameter	Action	Value
monitor:E (E)	x	Plot ...	
monitor:E (E)	y	Plot ...	
monitor:E (E)	z	Slice	0
monitor:E (E)	lambda	Slice	10

Label: x
Axis min: -7.5e-06
Axis max: 7.5e-06
Units: micro (1e-6) m

Select all or a portion of the data to send to script workspace, export to text file, or copy and paste into Excel

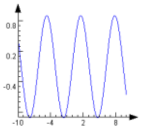
Scripting: Mathematics

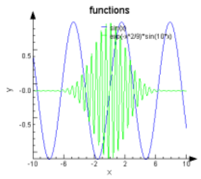
Simple Mathematics: plot some simple functions

> x=linspace(-10,10,500);

> y=sin(x);

> plot(x,y);





> y2=exp(-x^2/9)*sin(10*x);

> plot(x,y,y2,"x","y","functions");

> legend("sin(x)","exp(-x^2/9)*sin(10*x)");

>?size(x);

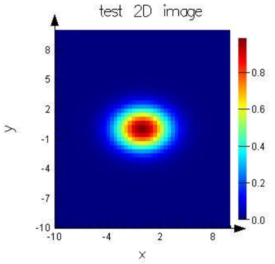
result:

500   1

Scripting: Mathematics

Simple Mathematics (plot a 2D gaussian)

```
> x=linspace(-10,10,50);
> y=linspace(-10,10,50);
> X = meshgridx(x,y);
> Y = meshgridy(x,y);
> ?size(x);          50  1
> ?size(X);          50  50
> E = exp(-X^2/9 – Y^2/4);
> image(x,y,E,"x","y","test 2D image");
```

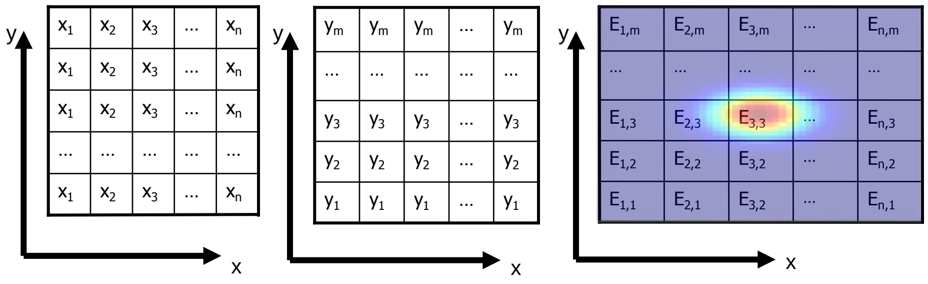


Scripting: Mathematics

- X = meshgridx(x,y);
- x has length n
- X has size n by m

- Y = meshgridy(x,y);
- y has length m
- Y has size n by m

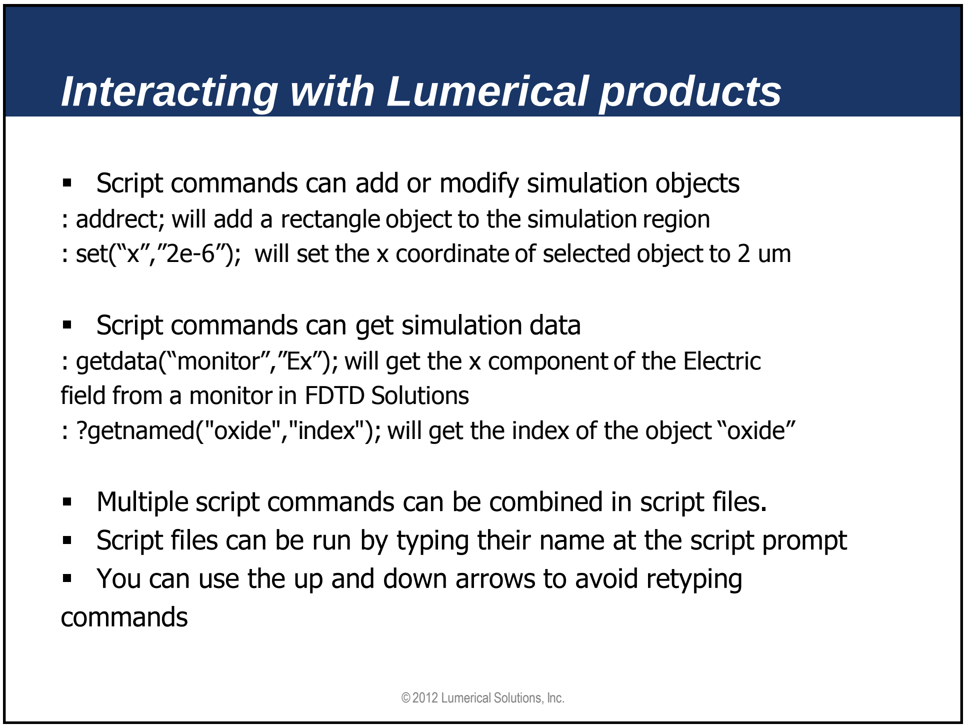
- E = exp(-X^2/9 – Y^2/4);
- E has size n by m



x_1	x_2	x_3	…	x_n
x_1	x_2	x_3	…	x_n
x_1	x_2	x_3	…	x_n
…	…	…	…	…
x_1	x_2	x_3	…	x_n

y_m	y_m	y_m	…	y_m
…	…	…	…	…
y_3	y_3	y_3	…	y_3
y_2	y_2	y_2	…	y_2
y_1	y_1	y_1	…	y_1

$E_{1,m}$	$E_{2,m}$	$E_{3,m}$	…	$E_{n,m}$
…	…	…	…	…
$E_{1,3}$	$E_{2,3}$	$E_{3,3}$	…	$E_{n,3}$
$E_{1,2}$	$E_{2,2}$	$E_{3,2}$	…	$E_{n,2}$
$E_{1,1}$	$E_{2,1}$	$E_{3,2}$	…	$E_{n,1}$



Interacting with Lumerical products

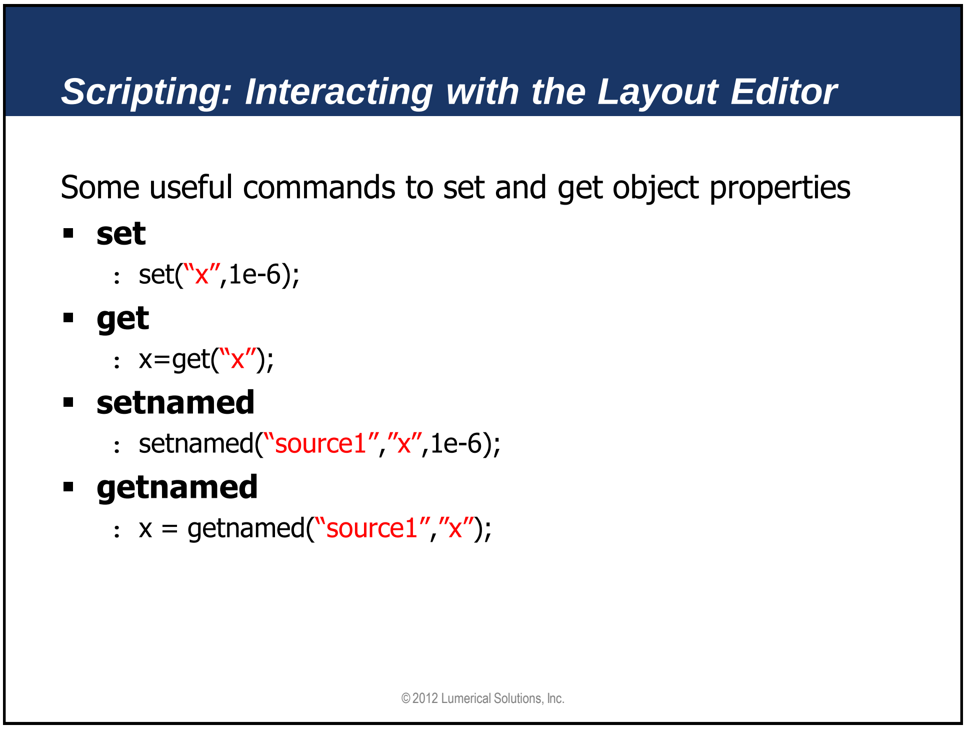
- Script commands can add or modify simulation objects

: addrect; will add a rectangle object to the simulation region
: set(“x”,”2e-6”); will set the x coordinate of selected object to 2 um

- Script commands can get simulation data

: getdata(“monitor”,”Ex”); will get the x component of the Electric field from a monitor in FDTD Solutions
: ?getnamed("oxide","index"); will get the index of the object “oxide”

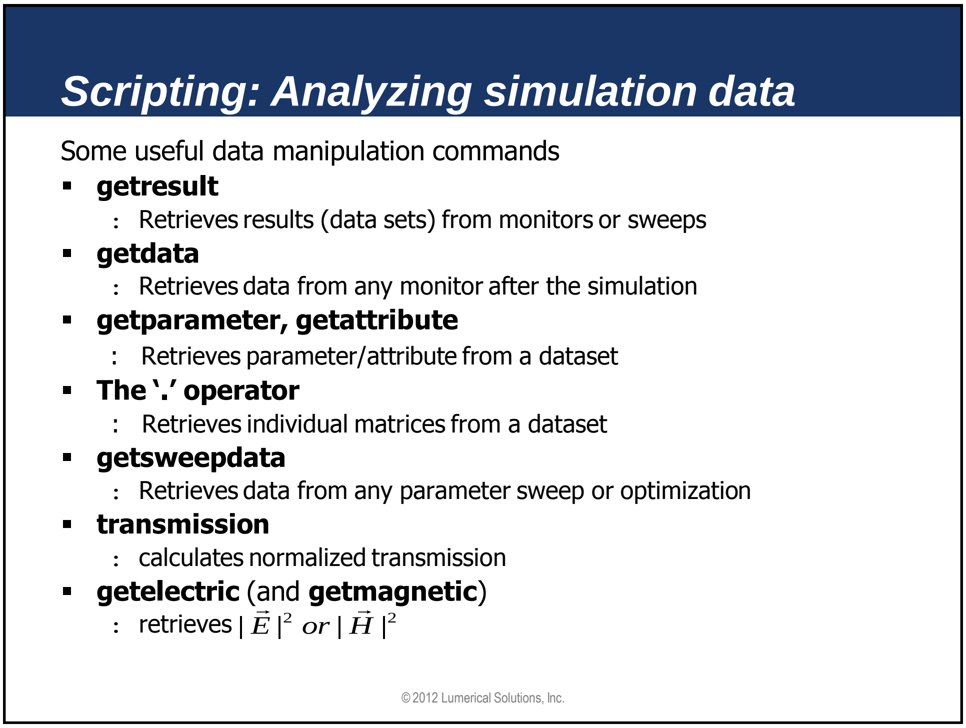
- Multiple script commands can be combined in script files.
- Script files can be run by typing their name at the script prompt
- You can use the up and down arrows to avoid retyping commands



Scripting: Interacting with the Layout Editor

Some useful commands to set and get object properties

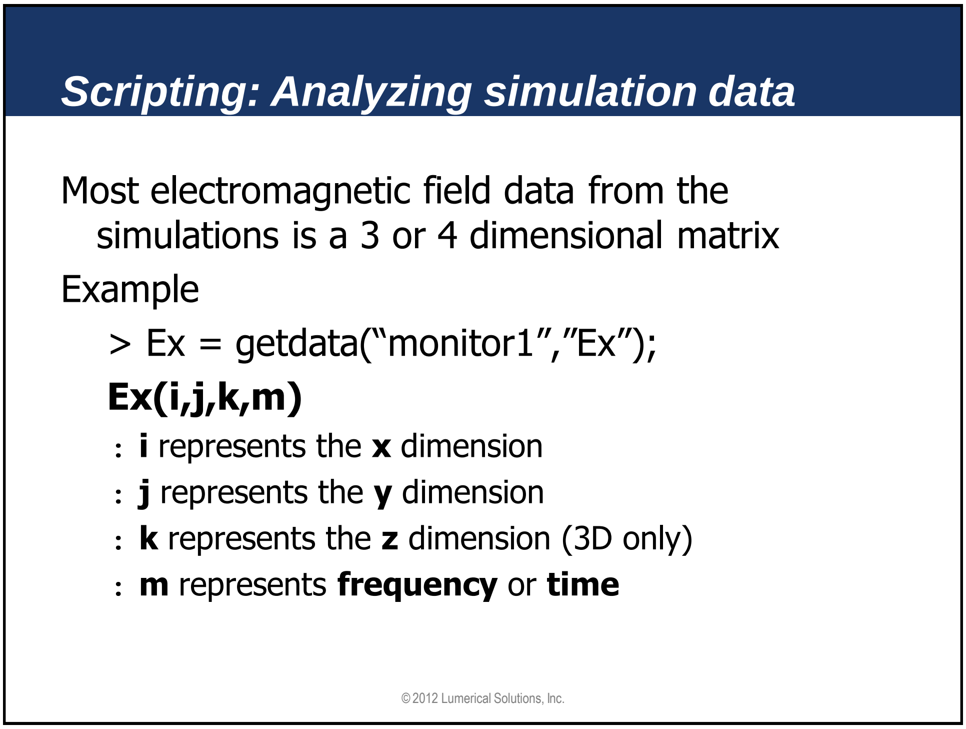
- **set**
  - : set(“x”,1e-6);
- **get**
  - : x=get(“x”);
- **setnamed**
  - : setnamed(“source1”,”x”,1e-6);
- **getnamed**
  - : x = getnamed(“source1”,”x”);



Scripting: Analyzing simulation data

Some useful data manipulation commands

- **getresult**
  - : Retrieves results (data sets) from monitors or sweeps
- **getdata**
  - : Retrieves data from any monitor after the simulation
- **getparameter, getattribute**
  - : Retrieves parameter/attribute from a dataset
- **The '.' operator**
  - : Retrieves individual matrices from a dataset
- **getsweepdata**
  - : Retrieves data from any parameter sweep or optimization
- **transmission**
  - : calculates normalized transmission
- **getelectric** (and **getmagnetic**)
  - : retrieves $|\vec{E}|^2 \ or \ |\vec{H}|^2$



Scripting: Analyzing simulation data

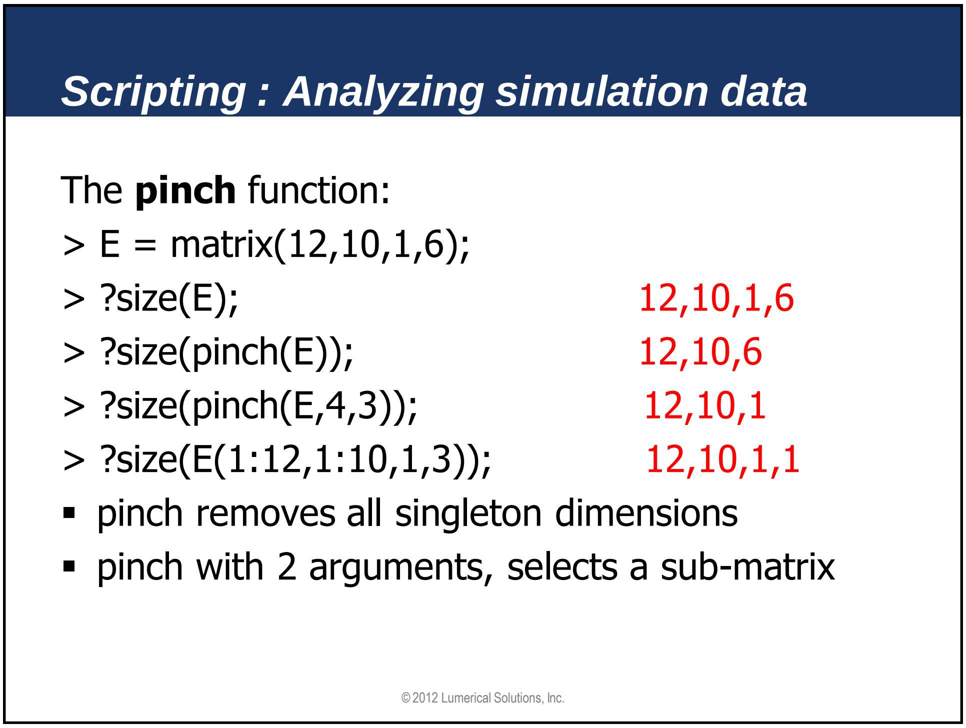
Most electromagnetic field data from the simulations is a 3 or 4 dimensional matrix

Example

```
> Ex = getdata("monitor1","Ex");
```

Ex(i,j,k,m)

- : **i** represents the **x** dimension
- : **j** represents the **y** dimension
- : **k** represents the **z** dimension (3D only)
- : **m** represents **frequency** or **time**

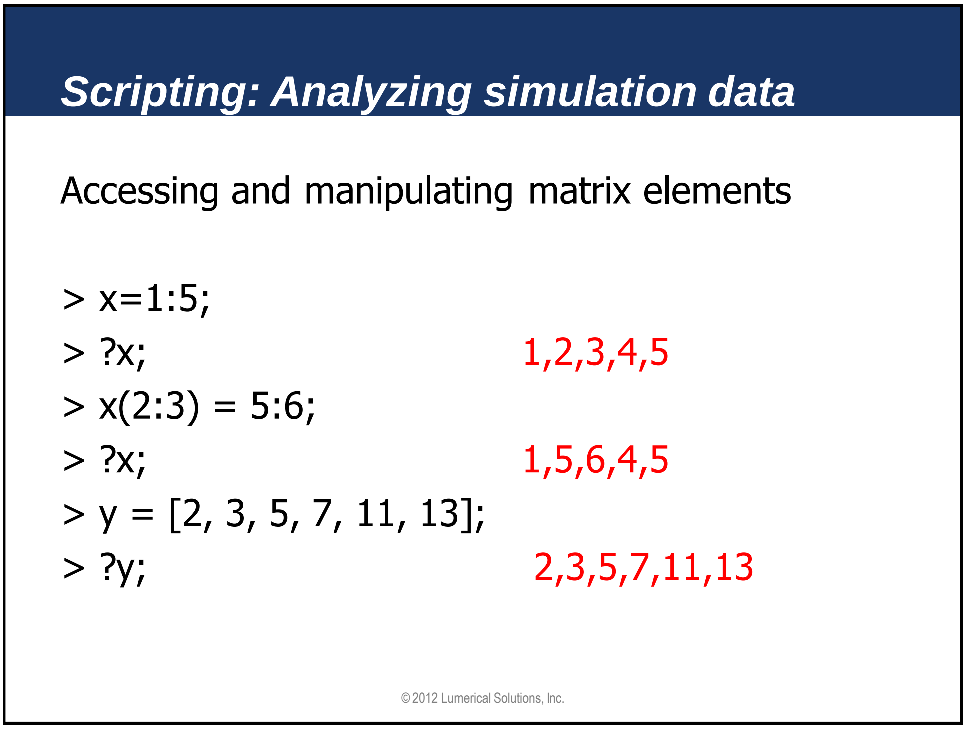


Scripting : Analyzing simulation data

The **pinch** function:

```
> E = matrix(12,10,1,6);
> ?size(E);                        12,10,1,6
> ?size(pinch(E));                 12,10,6
> ?size(pinch(E,4,3));             12,10,1
> ?size(E(1:12,1:10,1,3));         12,10,1,1
```

- pinch removes all singleton dimensions
- pinch with 2 arguments, selects a sub-matrix



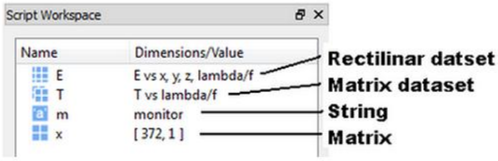
Scripting: Analyzing simulation data

Accessing and manipulating matrix elements

```
> x=1:5;
> ?x;                         1,2,3,4,5
> x(2:3) = 5:6;
> ?x;                         1,5,6,4,5
> y = [2, 3, 5, 7, 11, 13];
> ?y;                         2,3,5,7,11,13
```

Scripting: Analyzing simulation data

Datasets : structured data objects that collect a set of related matrices into a single convenient object.



Script Workspace

Name	Dimensions/Value
E	E vs x, y, z, lambda/f
T	T vs lambda/f
m	monitor
x	[372, 1]

Rectilinar datset

Matrix dataset

String

Matrix

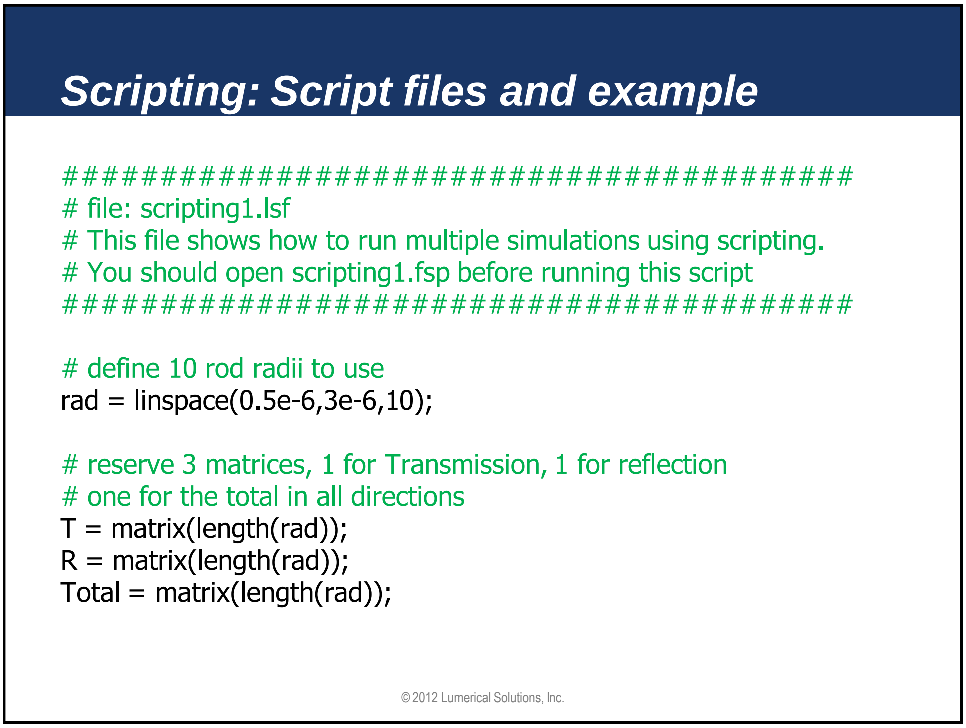
```
?E_field = getresult("monitor","E");
> E vs x, y, z, lambda/f
```

To output the actual attribute or parameter values, do something like:

```
?E_field.x;   # output the 'x' position vector

> result:
> -6.58393e-006
> -6.5442e-006
> -6.50447e-006
> .....
```

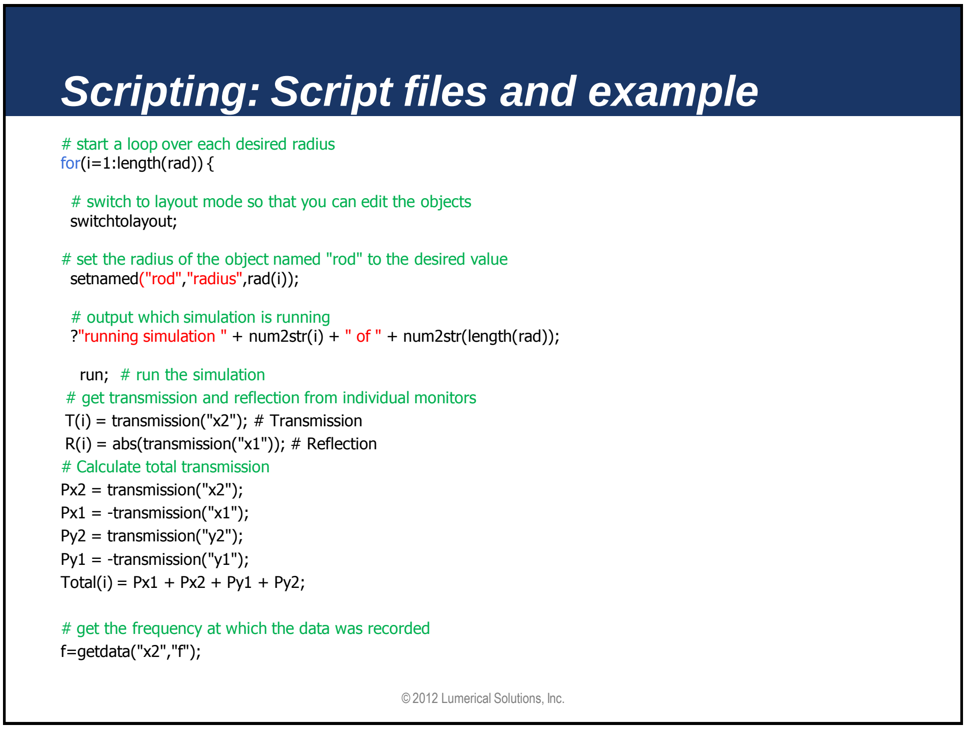
Scripting: Script files and example

- Copy the file scripting1.fsp and scripting1.lsf to your machine
- Open scripting1.fsp and edit scripting1.lsf
- We'll calculate transmission and reflection as a function of the radius of a dielectric rod



Scripting: Script files and example

```
########################################
# file: scripting1.lsf
# This file shows how to run multiple simulations using scripting.
# You should open scripting1.fsp before running this script
########################################

# define 10 rod radii to use
rad = linspace(0.5e-6,3e-6,10);

# reserve 3 matrices, 1 for Transmission, 1 for reflection
# one for the total in all directions
T = matrix(length(rad));
R = matrix(length(rad));
Total = matrix(length(rad));
```



Scripting: Script files and example

```
# start a loop over each desired radius
for(i=1:length(rad)) {

  # switch to layout mode so that you can edit the objects
  switchtolayout;

# set the radius of the object named "rod" to the desired value
  setnamed("rod","radius",rad(i));

  # output which simulation is running
  ?"running simulation " + num2str(i) + " of " + num2str(length(rad));

    run;  # run the simulation
 # get transmission and reflection from individual monitors
 T(i) = transmission("x2"); # Transmission
 R(i) = abs(transmission("x1")); # Reflection
# Calculate total transmission
Px2 = transmission("x2");
Px1 = -transmission("x1");
Py2 = transmission("y2");
Py1 = -transmission("y1");
Total(i) = Px1 + Px2 + Py1 + Py2;

# get the frequency at which the data was recorded
f=getdata("x2","f");
```

Scripting: Script files and example

```
} # end of the main loop over the radius
# plot the final results
plot(rad*1e6,R,T,Total,"radius (microns)","normalized power","transmission
    and reflection at " + num2str(c/f*1e6) + " microns");
legend("R","T","total");
```

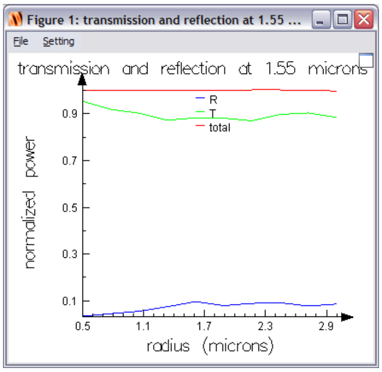


Figure 1: transmission and reflection at 1.55 ...
File
Setting
transmission and reflection at 1.55 microns
R
T
total
normalized power
0.9
0.7
0.5
0.3
0.1
0.5
1.1
1.7
2.3
2.9
radius (microns)

Structure groups and analysis groups

TOPICS

- Creating simple groups
- Using script commands in groups to parameterize structures
- Creating advanced analysis groups to calculate figures of merit
- Using the Object library

Structure groups and analysis groups

Structure groups and analysis groups

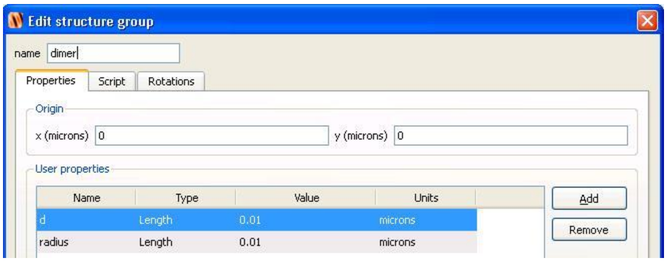
- Powerful objects to parameterize your designs and analysis
- Can be simple grouping of objects, or very complex structures using scripting

Work through an example based on the silver nanowire scattering example from the getting started

- The completed file and script is in the groups folder (nano_dimer.fsp and nano_dimer.lsf)
- Copy the file nanowire.fsp file to your computer and rename nano_dimer.fsp

Structure groups and analysis groups

- Structure group
  - Select the circle and copy it
  - Set the name of one circle to “left” and the other to “right”
  - Select both circles and add them to a new group
    - Name the group “dimer”
  - Edit the group and add properties
    - radius (type=Length) and set to 10nm
    - d (type=Length) and set to 10 nm



Edit structure group

name: dimer

Origin

x (microns): 0

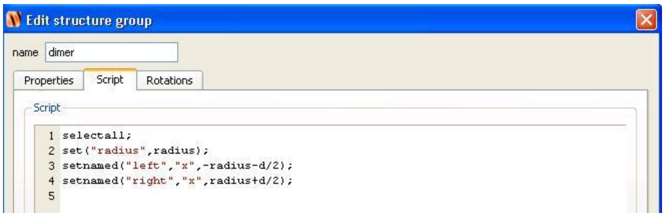
y (microns): 0

User properties

Name	Type	Value	Units
d	Length	0.01	microns
radius	Length	0.01	microns

Structure groups and analysis groups

- Create a script in the group with the following lines
  - selectall;
  - set("radius",radius);
  - setnamed("left","x",-radius-d/2);
  - setnamed("right","x",radius+d/2);



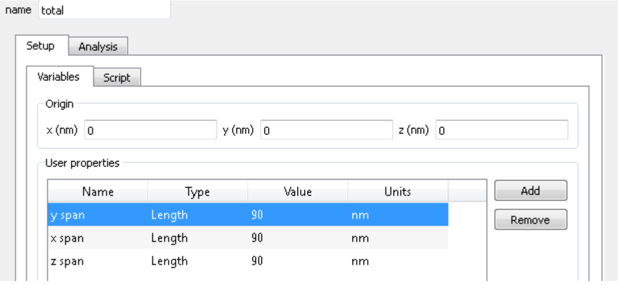
Edit structure group

name: dimer

```
selectall;
set("radius",radius);
setnamed("left","x",-radius-d/2);
setnamed("right","x",radius+d/2);

```

Structure groups and analysis groups

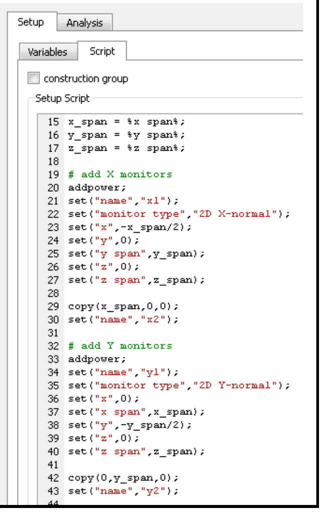
- Look at the analysis groups (e.g., "scat")
  - The group has 4 monitors that make a box around the particles
  - The width can be modified easily



name: total

x (nm): 0 y (nm): 0 z (nm): 0

Name	Type	Value	Units
y span	Length	90	nm
x span	Length	90	nm
z span	Length	90	nm

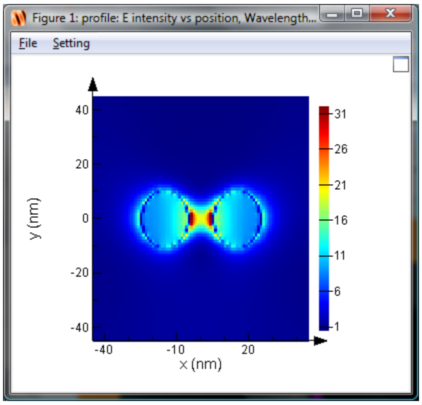


☐ construction group

```
x_span = %x span%;
y_span = %y span%;
z_span = %z span%;

# add X monitors
addpower;
set("name","x1");
set("monitor type","2D X-normal");
set("x",-x_span/2);
set("y",0);
set("y span",y_span);
set("z",0);
set("z span",z_span);

copy(x_span,0,0);
set("name","x2");

# add Y monitors
addpower;
set("name","y1");
set("monitor type","2D Y-normal");
set("x",0);
set("x span",x_span);
set("y",-y_span/2);
set("z",0);
set("z span",z_span);

copy(0,y_span,0);
set("name","y2");
```

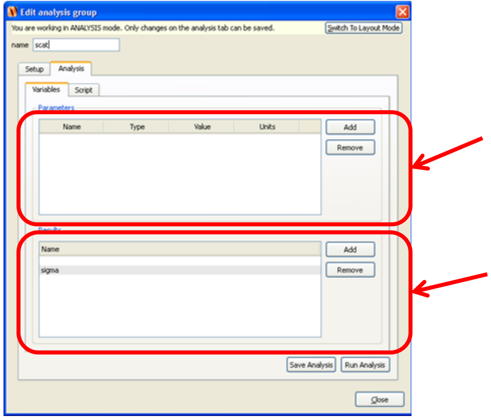
Structure groups and analysis groups

- Run the simulation and look at the field distribution at 345nm



Structure groups and analysis groups

- We can define input parameters and output values for the analysis group object



You are working in ANALYSIS mode. Only changes on the analysis tab can be saved.

name scat

Parameters

Name	Type	Value	Units

Results

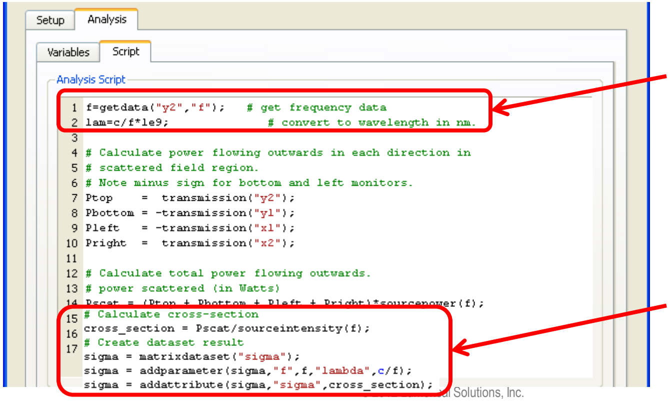
Name
sigma

Input parameters

Output values

Structure groups and analysis groups

- This script calculates the cross section
  - sigma = power / I_s
    - power is the net power flow out of the box (Watts)
    - I_s is the intensity of the source (Watts/m)
  - It also returns the frequencies (f) and wavelength in nm (lam)



```
f=getdata("y2","f");   # get frequency data
lam=c/f*1e9;               # convert to wavelength in nm.

# Calculate power flowing outwards in each direction in
# scattered field region.
# Note minus sign for bottom and left monitors.
Ptop    =  transmission("y2");
Pbottom = -transmission("y1");
Pleft   = -transmission("x1");
Pright  =  transmission("x2");

# Calculate total power flowing outwards.
# power scattered (in Watts)
Pscat = (Ptop + Pbottom + Pleft + Pright)*sourcepower(f);
# Calculate cross-section
cross_section = Pscat/sourceintensity(f);
# Create dataset result
sigma = matrixdataset("sigma");
sigma = addparameter(sigma,"f",f,"lambda",c/f);
sigma = addattribute(sigma,"sigma",cross_section);
```

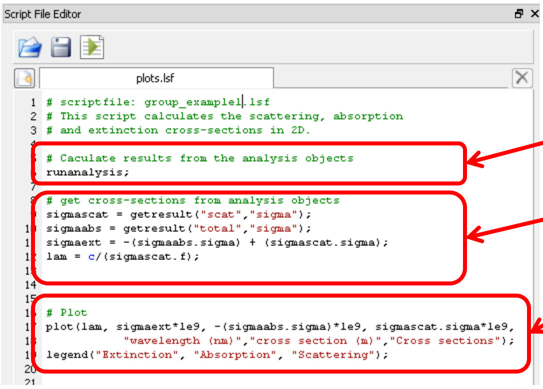
f and lam

sigma

al Solutions, Inc.

Structure groups and analysis groups

- Consider the final analysis when using an analysis object
  - File nano_dimer.lsf



```
# scriptfile: group_example1.lsf
# This script calculates the scattering, absorption
# and extinction cross-sections in 2D.

# Caculate results from the analysis objects
runanalysis;

# get cross-sections from analysis objects
sigmascat = getresult("scat","sigma");
sigmaabs = getresult("total","sigma");
sigmaext = -(sigmaabs.sigma) + (sigmascat.sigma);
lam = c/(sigmascat.f);



# Plot
plot(lam, sigmaext*1e9, -(sigmaabs.sigma)*1e9, sigmascat.sigma*1e9,
           "wavelength (nm)","cross section (m)","Cross sections");
legend("Extinction", "Absorption", "Scattering");
```

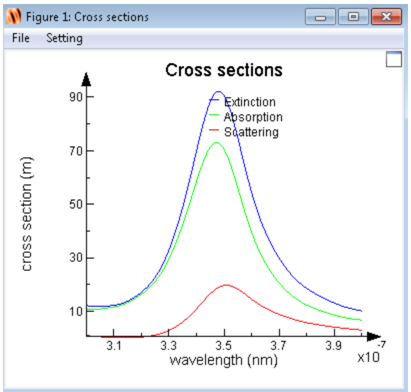
Update the output

Use getresult command to get the output values

Plot the results

Structure groups and analysis groups

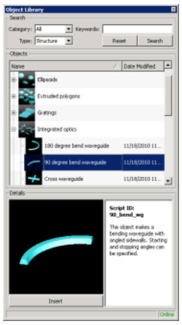
- The final result



© 2012 Lumerical Solutions, Inc.

Structure groups and analysis groups

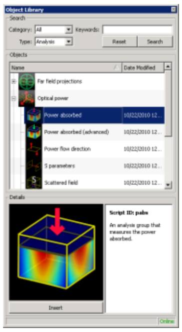
- Where can you get these objects?
  - The object library



Category: All
Type: Structure
Keywords:

Script ID: 90_bend_wg

This object makes a bending waveguide with angled sidewalls. Starting and stopping angles can be specified.



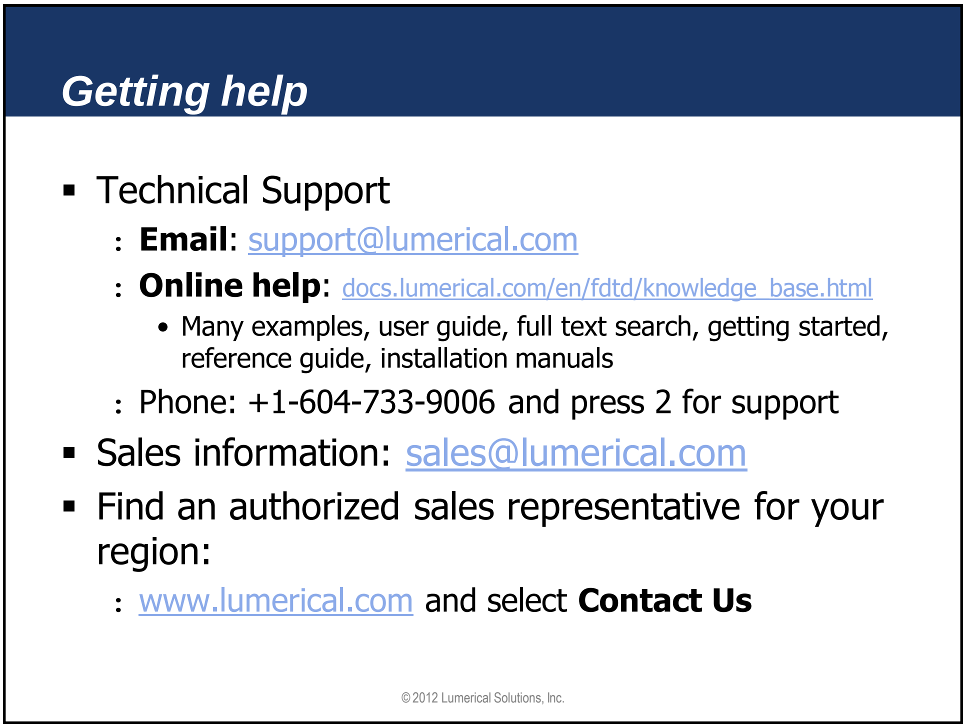
Category: All
Type: Analysis
Keywords:

Script ID: pabs

An analysis group that measures the power absorbed.

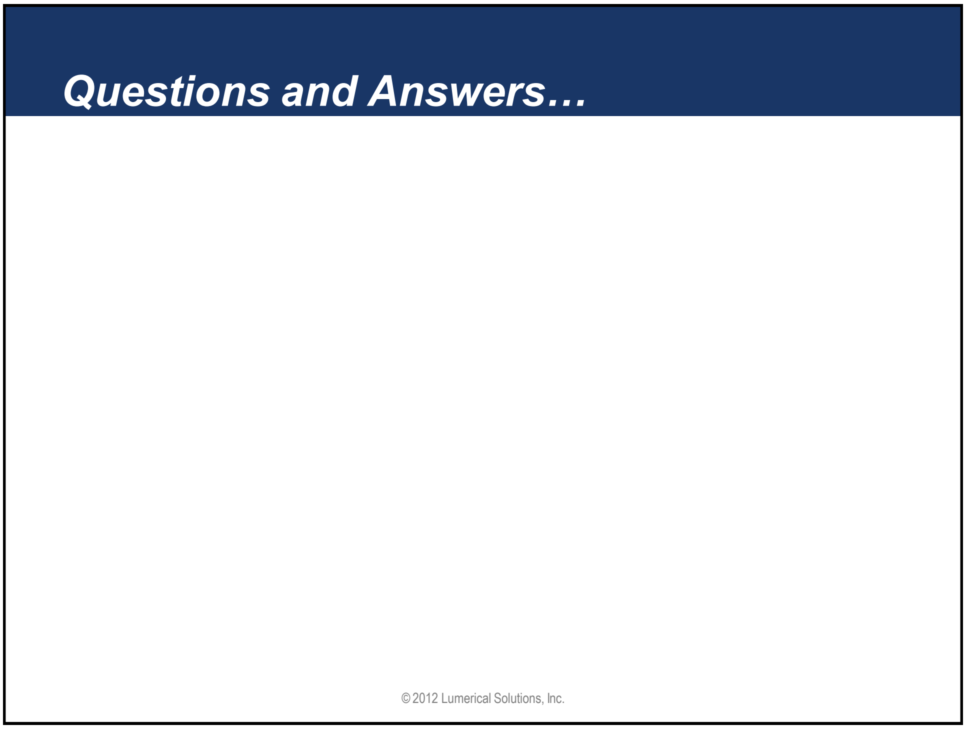
  - Create your own!
    - Email support@lumerical.com for help creating them for your application

© 2012 Lumerical Solutions, Inc.



Getting help

- Technical Support
  - **Email**: support@lumerical.com
  - **Online help**: docs.lumerical.com/en/fdtd/knowledge_base.html
    - Many examples, user guide, full text search, getting started, reference guide, installation manuals
  - Phone: +1-604-733-9006 and press 2 for support
- Sales information: sales@lumerical.com
- Find an authorized sales representative for your region:
  - www.lumerical.com and select **Contact Us**



Questions and Answers…