Contents

Chapter 1

Introduction

A calculation can be carried out directly via the GOAT library. Sometimes it is useful to save the scene information externally. A separate GOAT::XML::xmlReader class is available for this purpose. It is based on an XML file, the structure of which is described in more detail in this documentation. In the following, we will given an overview on the structure of a XML file, which can be used as an input for the setup of the scene and for a couple of different calculation processes. Firstly, the principle structure will be discussed. Please note that all lengths are given in µm. After the usual preamble of the XML file, everything is encapsulated in the Root entry, i.e. <Root> ... <\Root>. In the following, the term "section" is used for a closed part which is encapulated by <Sectionname> ... <\Sectionname>. Within the root section, there are two possible subsections, the Scene section, which describes the whole structure of the scene with all light sources, objects etc. and the calculation section where the calculation parameters are described. As values there are strings, integer or floating point numbers, threedimensional vectors and complex numbers. **Threedimensional vectors** are given by the x,y and z component. In the following sections, the parameters that apply equally to all elements of the same type (light sources, detectors, etc.) are specified first. Then only the specific parameters are specified.
examples:

Threedimensional Vector:

```
<Position x="2.3" y="4.5" z="-5.6" />
```

A **complex number** is given its real and its imaginary part:

```
<n real="1.5" imag="0.1"/>
```

The default values for the threedimensional vector is (0,0,0) and for the complex number (0,0) unless otherwise specified.

Chapter 2

Scene

Within this section, all elements like light sources, objects and detectors are described. The scene has the parameter: the radius of the calculation space r0.

2.1 Light sources

All light sources have some entries which are used for all types. All parameters are optional. If one parameter is missing, it will be set to its default value, which is given in a table below.

Parameters used for all light sources

Parameter name	description	possible value	default value
Type	type of the light source	"plane","gaussian","ring", "tophat","plane_mc", "gaussian_mc","ring_mc", "gaussian_ring_mc"	–
Position	position of the light source (center of the area)	3D vector	(0,0,0)
NumRays	Number of rays per calculation step	integer number	100
Size	width of the light source	floating point number	10.0
Wavelength[1]	Wavelength of the light source	floating point number	1.0
Polarisation	Polarisation	3D Vector	(1,0,0)

[1] For pulsed calculation, this wavelength will be overwritten

Note: As described in more detail in the documentation for the LightSrc class, the polarization here also refers to the case where the light source points in the z direction. The polarization is rotated according to the direction defined in “Direction”.

2.1.1 Plane wave

type: **plane**
This type is the most simplest type of light source. This is a plane wave in which the rays are emitted equally distributed. The number of rays here only refers to one direction, i.e. the total number of rays is the square ($Numrays^2$). The distance between two adjacent rays is ${}^{Size}/_{NumRays}$. The only special parameter is

Parameter name	description	possible value
Direction	Direction of the plane wave	3D Vector

2.1.2 Plane wave (mc)

type: **plane_mc**
All light sources denoted wset (gca,'position',[0.1 0.1 0.75 0.75],'xlim',[-1000 1000],'ylim',[-1000 1000]);ith "mc" are those where the rays are arbitrarily distributed. Unlike in the case of the plane wave, the total number of rays is equal to NumRays. Also here, the only special parameter is "Direction".

2.1.3 Gaussian wave

type: **gaussian**
A gaussian wave describe a wave which is focused towards the focal point and has a gaussian radial intensity distribution. The direction of the wave is given by the position of the light source and the focal point. For the description of the gaussian distribution, one can either give the (virtual) waist width at the focal point, w0 or the numerical aperture NA. If both are given, the numerical aperture is used. Please note, that the width (full width at half maximum) refers to the electric field:

$$E = E_0 \cdot e^{\frac{-x^2}{2\sigma^2}} \Rightarrow \sigma^2 = \frac{\delta x_E^2}{8 \ln 2} \tag{2.1}$$

$$I = I_0 \cdot e^{\frac{-2x^2}{2\sigma^2}} \tag{2.2}$$

$$\boxed{\delta x_I = \frac{\delta x_E}{\sqrt{2}}} \tag{2.3}$$

with the width of the electric field δx_E and the corresponding width of the intensity profile δ_I. Special parameters:

Parameter name	description	possible value
w0	(virtual) waist at the focal point	floating point number
NA[2]	numerical aperture	floating point number

2.1.4 Gaussian wave (mc)

type: **gaussian_mc**
The same as the normal gaussian wave, except that the distribution of the rays is arbitrary and the intensity distribution is given by the density of the rays.

2.1.5 Ring shaped light source

type: **ring**
A light source, shaped like a ring with equally distributed rays. The ring is described by its inner and outer radius.

Parameter name	description	possible value
rmin	inner radius (≥ 0)	floating point number
rmax	outer radius	floating point number
Direction	direction of the light rays	3D vector

2.1.6 Ring shaped light source (mc)

type: **ring_mc**
Like ring shaped light source, except that the rays are distributed arbitrarily.

2.1.7 Gaussian ring shaped light source (mc)

type: **gaussian_ring_mc** The light source consists of a Gaussian beam profile from which a ring is cut out. In contrast to the Gaussian beam, the rays are aligned in parallel. Like all light sources denoted with "mc", the rays are distributed arbitrarily. The intensity is given by the density of the rays.

Parameter name	description	possible value	default value
rmin	inner radius	double ($0 \leq$ rmax)	0
rmax	outer radius	double	100
width	FWHM	double	rmax (100)
Direction	direction of the beam	3D-Vector	–

[2] if w0 and NA are given, NA is used

Example for a scene with one light source

```
 <?xml version="1.0" encoding="utf-8"?>
<Root>
  <Scene r0="4E+4">
    <nS imag="0.0" real="1.0" />
    <LightSources>
      <LightSource NumRays="100000" Size="1500" Type="ring_mc"
       Wavelength="1.0"
       rmax="500" rmin="0">
        <Position x="0.0" y="0" z="-1E+4" />
        <Direction x="0.0" y="0" z="1" />
      </LightSource>
    </LightSources>
 </Root>
```

Beside these general parameters, every type of light sources have there own special parameters

2.1.8 Line shaped light source

For some calculations a twodimensional calculation is sufficient. For this purpose a line shaped light source is introduced. Beside the position, this source is defined by its direction, which gives the direction of the rays and the orientation (lateral direction).

2.2 Objects

Like the light sources, all objects have some general parameters. Those parameters can be seen in the following table

Parameter name	description	possible value	default value
Pos	Position of the object[3]	3D Vector	(0,0,0)
Type	Type of the object	ellipsoid, surface, cone, aspheric_lens, box	–
Direction	direction of the light rays	3D vector	(0,0,0)

[3]Reference point for the object differ from shape to shape

Alpha	rotation angle around x-axis (in radiants)	double	0
Beta	rotation angle around y-axis (in radiants)	double	0
Gamma	rotation angle around z-axis (in radiants)	double	0
IsActive	Fields inside the object will be stored (this concerns e.g. the pulsed calculation and the inelastic scattering)	true/false	false
n	refractive index	complex value	(1.0,0.0)

2.2.1 Ellipsoid

type: **ellipsoid**
This object type describes an elliptic object, defined by the three semi-axis, according to

$$\frac{(x-x_c)^2}{a_x^2}+\frac{(y-y_c)^2}{a_y^2}+\frac{(z-z_c)^2}{a_z^2}=1 \tag{2.4}$$

The position is defined by the center of the ellipsoid (x_c,y_c,z_c).

Parameter	description	possible value	default value
Dimension	Vector which holds semi-axis along the x-, y- and z-direction	3D Vector	(10,10,10)

2.2.2 Box

type: **box**
This object type describes a cuboid, defined by the edge lengths. The center of the cuboid is used as position (reference point).

Parameter	description	possible value	default value
Dimension	Vector which holds the edge lengths along the x-, y- and z-direction	3D Vector	(10,10,10)

2.2.3 Cylinder

type: **cylinder** This object type is a spheric cylinder.

Parameter	description	possible value	default value
height	Height of the cylinder	double	1
radius	Radius of the cylinder	double	1

2.2.4 Surface

type: **surface**
A the surface of a surface object is described by triangles stored in a file. Up to now two file formats are supported:

- **SRF**-files: A proprietary file format. The first entry is the number of triangles stored in the file, followed by a list of the triangles. This file format is described in more detail in the documentation of the GOAT library.
- **STL**-files: Here only binary stl-files are supported.

Parameter	description	possible value	default value
FileType	String, which holds the file type	".srf",".stl"	–
FileName	String, which holds the file name	string	–

2.2.5 Spheric lens

type: **spheric_lens**
A spheric lens is described by the two side faces (see also documentation of the GOAT-library for details). It has two child elements named "left" and "right" for the two corresponding surfaces.

Parameter	description	possible value	default value
R	radius of the lens	double	0
left			

2.2.6 Vortex phase plate

type: **vortex_plate**
A vortex phase plate as shown in Fig.2.1 is an object to create a vortex

beam. It is defined by its height dh and the topological charge m, which gives the number of subdivisions. The full object is described by a cylinder with a given height and on top,the spiral vortex structure.

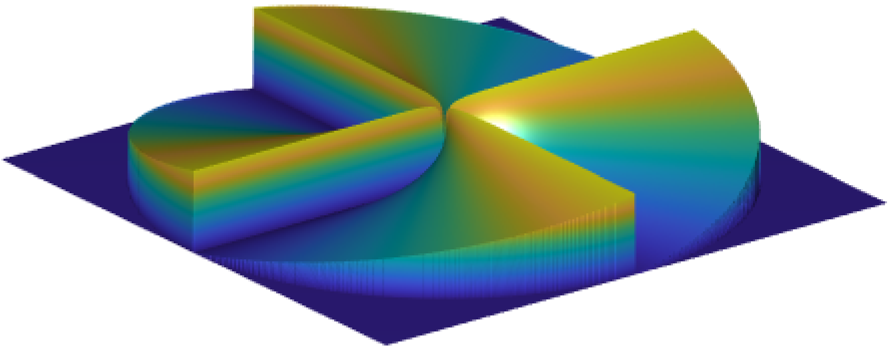

Figure 2.1: Vortex phase plate with $m = 4$

Parameter	description	possible value	default value
height	height of the base cylinder	double	1
radius	radius of the object	double	1
m	topological charge	integer	1
dh	height of the vortex	double	1

2.3 Detectors

A detector is a special twodimensional object in which the electric field is stored. Like for the light sources and the objects the section is encapsulated in a tag with sectioname "Detectors".

Parameter	description	possible value	default value
Type	type of the detector	"plane"	–
Position	Center of the detector	3D-vector	–
Direction	normal vector	3D-vector	–
Filename	name of the file to store the data	string	–

2.3.1 Plane detector

type: **plane** This is square plane detector object.
Special parameters:

Parameter	description	possible value	default value
d	width of the detector	double	1
n	no. of elements per direction	int	1

Chapter 3

Calculations

Also, calculations can be defined and started with help of the XML file. Calculations are given in the section "Calculations". Similar to the definition of the light sources, every single entry for a calculation starts with "Calculation". Depending on the attribute "Type" there are different calculations possible:

type name	description
pure	Raytracing will be done without any further calculation will be done. Only detectors are considered.
path	Raytracing will be performed, path of the rays can be stored in a file.
pulse	Calculation with pulsed light source will be performed.

Parameters for all calculations

Parameter name	description	possible value	default value
Type	type of the calculation	"pure","path", "pulse"	–
Inactive	if true, this calculation will be skipped	true, false	false

3.1 Calculation with detectors only

If one want only the electric field distribution at certain areas, this calculation type "**pure**" is the right choice. Here, beside light sources and objects, only the detectors will be considered. Therefore no further special parameters are needed.

3.2 Calculation of the ray paths

Sometimes it is necessary to know the path of the rays through the scene. This can be done with the calculation type "**path**". In the file, the starting and ending positions of each step, i.e. from surface to surface (or from light source to surface). The corresponding parameters are

Parameter name	description	possible value	default value
Filename	Name of the file in which the paths are stored. This parameter is essential	string	–
numRays	Number of rays[1]	integer	1
numReflex	Number of reflections	integer	0

Sometimes it is convenient to switch off calculations without erasing it. For this reason, the property "Inactive" can be set to "true".

3.3 Calculations with pulsed laser sources

Here, all light sources are considered as mode-locked lasers. For this calculation, two different methods for the Fourier transform can be applied. On the one hand, a pure raytracing calculation can be performed, here attribute "Method" is set to "rtonly". On the other hand a mixture of raytracing and an integral can be used (attribute "Method" is not set). Here, the frequency range used for the calculation is divided into frequency divisions. For each division, one raytracing at the center frequency is performed.

The parameters are as follows:

Parameter name	description	possible value	default value
EstimateTimeForObject	The time when the pulse hits the object with the given number. This can only be used with one light source. If a time is given, it is relative to this estimated time. Can only be used without Method "rtonly"!	number of the corresponding object	–

Filename	Prefix of the filename used to store the result. A file for each active object will be created in the form <Filename><object no.>.dat	any filename	–
Spatial_Resolution	resolution of the grid in µm	double > 0	1
Wavelength	Central wavelength in µm	double	1 µm
NumReflexions	Number of reflections	integer ≥ 0	INEL_MAX_NREFLEX (1)
NumLoops	Specifies how often the calculation should be repeated (-1: calculation is repeated infinitely often)	integer ≥-1	-1
Pulse_width	Pulse width (FWHM) in fs	double	100 fs
Method	For the calculation a pure raytracing calculation will be performed (the fields for all wavelengths will be calculated by raytracing) or a mixed method can be applied	"rtonly","mixed"	"mixed"
NumSpectralRanges	Number of spectral ranges considered for calculation	integer ≥ 1	4

Additionally, a list with the materials is required. For each object, the material is given by the name of the material. Up to now the following materials are available:

Material	value
Air	air
Glass	glass (allways refractive index 1.5)
N-BK7 glass	bk7
N-LASF55 glass	lasf55
Vacuum	vacuum

For vacuum and glass constant refractive indices were used (1 for vacuum and 1.5 for glass). The other dispersion curves were taken from `https://www.refractiveindex.info`.