

EGO - Virgo



A Novel Non-Sequential 3D Gaussian Beam Tracer: Theia

Raphaël Duque, EGO Optics Group

Virgo Week July 2017

Wednesday, July 19th 2017

What is theia?

- Synthesis of existing Gaussian optics simulation tools (OptoCAD, gtrace, IfoCad)
- + enhanced flexibility → Scripting
- + enhanced user interface → 3D visualization and navigation
- + enhanced user environment → support and extensibility
- + specific aims in optics design

What are the specific aims?

- Ghost beam hunting
- Large-scale optical benches

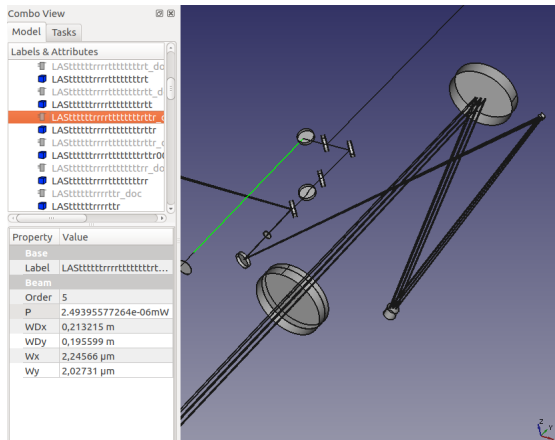


Figure: Working environment of theia

- 1 Scope and validation of theia
- 2 Operation of theia
- 3 The extensive user environment
- 4 Quick tutorial

- 1 Scope and validation of theia
- 2 Operation of theia
- 3 The extensive user environment
- 4 Quick tutorial

Scope and validation of theia

Features

Scope of theia: Non-sequential 3D Gaussian beam tracer *and* visualization

No	Yes (July 14 th)	Not yet	Extensions
Higher order modes	3D general astigmatic Gaussian beams	All 2 nd order surfaces	Polarization
Grating surfaces	Spherical surfaces (mirrors, lenses)	High-level 3D visualization	Surface action specification
Response in GW environment	{Non-, }sequential tracing	Beam tree navigation interface (and transverse section inspection)	2-way communication with CAD
	Low-level 3D visualization	Interferences	
		Cavities	
		Beam clipping	

Table: The functionalities of theia v0.1.1

Approximations:

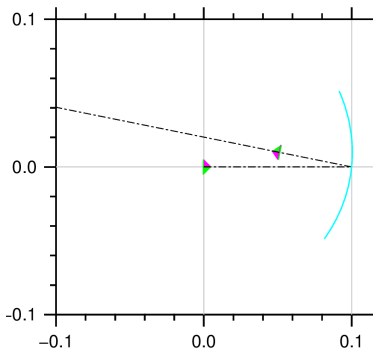
- Geometrical optics: none
- Gaussian data:
ROC(beam) \gg ROC(surface)
(+ paraxial)

Test cases:

- 2D: confronted with OptoCAD
- 3D: comparison with experimental data
- See tutorial files in project

Scope and validation of theia

Validating theia: 2D



```
Beam # 1:  
rs  s2  asoi  ang[deg]  z1t[m]  w0t[m]  z0t[m]  
0   0   n      0.0       0.0     1.000E-3  2.953E+0  
1   1   r      0.0       0.0     1.000E-3  2.953E+0  
2   -3  d      168.6E+0  -49.77E-3  16.85E-6  838.1E-6
```

```
(M1, HR) [Open] Beam0r {  
  Waist Pos: (0.05020495529598723, 0.049832118784703998)mm  
  Waist Size: (0.0169934117386, 0.0168671777452)mm  
  Direction: (80.1130682652, 180.0)deg  
}
```

Figure: Results comparison between OptoCAD (top) and theia (bottom).

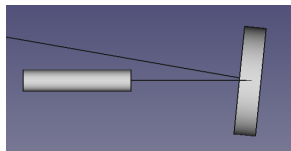


Figure: Simple setup for Gaussian optics testing (top: OptoCAD, bottom: theia).

OptoCAD and theia in agreement:

- Size of waist: $\leq 0.1\%$
- Position of waist: $\leq 0.1\%$

Scope and validation of theia

Validating theia: 3D

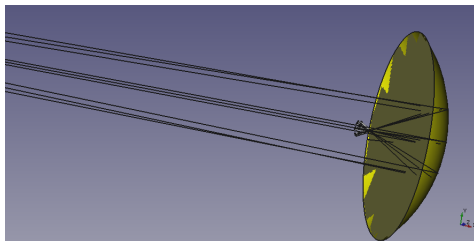


Figure: Interaction of a pencil of beams with a spherical reflecting surface

- Test against geometrical optics results
- Test against experimental results

- 1 Scope and validation of theia
- 2 Operation of theia**
- 3 The extensive user environment
- 4 Quick tutorial

Operation of theia

Algorithms

Algorithms + Data Structures = Programs

What low-level features allow the ghost beam hunting functionality?

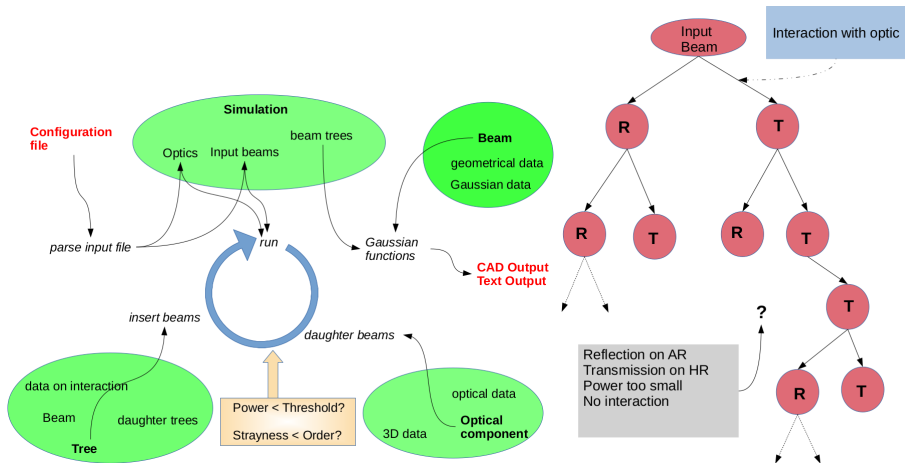


Figure: Left: the algorithm implemented in theia, right: the beam tree data structure.

Operation of theia

User interface

Text input

#Threshold and order each on their own line

```
order = 1  
threshold = 1.*mW
```

input beam

```
#bm Wx Wy WDistx WDisty Wl P X Y Z Theta Phi Alpha Name Re  
bm 0.045*mm, 0.094*mm, -43*mm, -45.0*mm, 1064*mm, 1*W, Y = 0*mm, Z = 0.,
```

thin lens (X,Y,Z is the center of the thin lens)

```
#th X Y Z Theta Phi Focal Diameter R T KeepI Name Ref  
th 11*cm, 0, 0, 90 * deg, arccos(-1.), sqrt(144)*cm, 5*cm, 0.1, 0.9, True, Ref = 'L1'  
th 53*cm, -15*cm, 0, 90*deg, 180 * deg, -150*mm, 5*cm, Ref = 'L2'  
th 90*cm, -15*cm, 0, 90*deg, 180 * deg, -300*mm, 5*cm, Ref = 'L3'
```

Text output

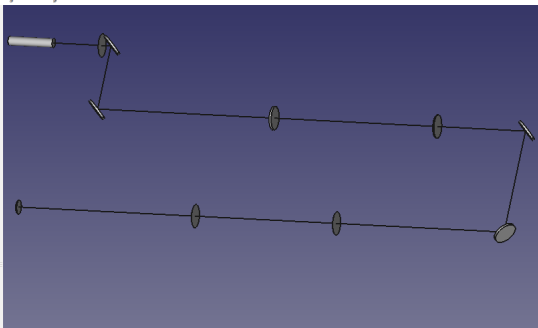
```
(H2, HR) 0.3999999999m (L2, HR) TBttrr {  
Waist Pos: (0.224742415377119777, 0.40587482437740047)m  
Waist Size: (0.269864950223, 0.171918229273)mm  
Direction: (90.0, -1.90833280888e-14)deg
```

Model Tasks

Labels & Attributes

- TBtt_doc
- TBtt
- TBttr_doc
- TBttr
- TBttrr_doc
- TBttrr
- TBttrrrt_doc
- TBttrrrt
- TBttrrrt_doc
- TBttrrrt

Property	Value
Base	
Label	TBttrr_doc
Beam	
Order	0
P	801.9mW
WDx	0,555875 m
WDy	0,374742 m
Wx	0,171918 mm
Wy	0,269865 mm



3D CAD output

- 1 Scope and validation of theia
- 2 Operation of theia
- 3 The extensive user environment**
- 4 Quick tutorial

<http://theia.hopto.org:56000>

- Quick start guide
- User documentation
- API documentation
- Tutorials
- Releases
- Contact with maintainer
- Access to git repo

Table of Contents

[Everything](#)

Modules

[theia](#)
[theia.helpers](#)
[theia.helpers.geometry](#)
[theia.helpers.interaction](#)
[theia.helpers.settings](#)
[theia.helpers.tools](#)
[theia.helpers.units](#)
[theia.main](#)
[theia.optics](#)

Everything

All Classes

[abc.ABCMeta](#)
[theia.helpers.tools.InputError](#)
[theia.helpers.tools.TotalRefle](#)
[theia.optics.beam.GaussianBe](#)
[theia.optics.beamdump.Beam](#)
[theia.optics.component.Setup](#)
[theia.optics.ghost.Ghost](#)
[theia.optics.lens.Lens](#)
[theia.optics.mirror.Mirror](#)
[theia.optics.optic.Optic](#)
[theia.optics.thicklens.ThickLe](#)
[theia.optics.thinlens.ThinLens](#)
[theia.rendering.features.FCB](#)
[theia.rendering.features.FCL](#)
[theia.rendering.features.FCM](#)
[theia.rendering.features.FCO](#)
[theia.running.simulation.Sim](#)
[theia.tree.beamtree.BeamTre](#)

All Functions

[theia.helpers.geometry.basis](#)
[theia.helpers.geometry.lineCy](#)
[theia.helpers.geometry.linePl](#)

```
1 '''Defines the Mirror class for theia.'''
2
3 # Provides:
4 class Mirror
5     #   init
6     #   lines
7     #   isHit
8     #   hit
9     #   hitHR
10    #   hitAR
11
12 import numpy as np
13 from ..helpers import geometry, settings
14 from ..helpers.units import deg, cm, pi
15 from ..optic import Optic
16 from ..beam import GaussianBeam
17
18 class Mirror(Optic):
19     ...
20
21     # Mirror class.
22
23     # This class represents semi reflective mirrors composed of two faces (HR, AR)
24     # and with a wedge angle. These are the objects with which the beams will
25     # interact during the ray tracing. Please see the documentation for details
26     # on the geometric construction of these mirrors.
27
28     # --- Attributes ---
29     SetupCount (inherited): class attribute, counts all setup components.
30         [integer]
31     OptCount (inherited): class attribute, counts optical components. [integer]
32     Name: class attribute. [string]
33     HRCenter (inherited): reference string (for keeping track with the lab). [3D vector]
34     HRNorm (inherited): unitary normal to the 'chord' of the HR (always pointing
35     towards the outside of the component). [3D vector]
36     Thick (inherited): thickness of the optic, counted in opposite direction to
37     HRNorm. [float]
38     Dia (inherited): diameter of the component. [float]
39     Ref (inherited): reference string (for keeping track with the lab). [string]
40     ARCenter (inherited): center of the 'chord' of the AR surface. [3D vector]
41     ARNorm (inherited): unitary normal to the 'chord' of the AR (always pointing
42     towards the outside of the component). [3D vector]
43     N (inherited): refraction index of the material. [float]
44     HRK, ARK (inherited): curvature of the HR, AR surfaces. [float]
45     HRr, HRt, ARr, ARt (inherited): power reflectance and transmission
46     coefficients of the HR and AR surfaces. [float]
47     KeepI (inherited): whether or not to keep data of rays for interference
48     calculations on the HR. [boolean]
49     Wedge: wedge angle of the mirror, please refer to the documentation for
50     detail on the geometry of mirrors and their implementation here.
51     [float]
52     Alpha: rotation angle used in the geometrical construction of the mirror
53     (see doc, it is the angle between the projection of Ex on the AR plane
54     and the vector from ARCenter to the point where the cylinder and the AR
55     face meet). [float]
56
```

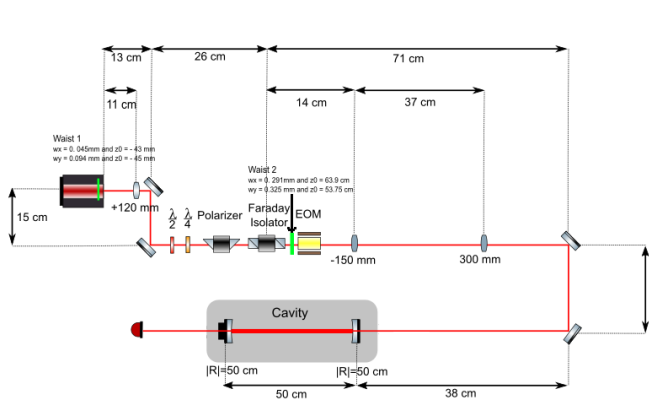
Figure: theia online API documentation

- 1 Scope and validation of theia
- 2 Operation of theia
- 3 The extensive user environment
- 4 Quick tutorial**

Quick tutorial

CLI Tutorial

- Typical telescope/cavity setup, used as a test case for experimental validation
- Order 1 simulation for ghost beam hunting: **Will transmitted beams interfere with PD?**



Plane	Sagittal	Tangential
Exp.	260 μm @ 22 cm	170 μm @ 41 cm
Sim.	320 μm @ 26 cm	290 μm @ 36 cm

Table: Experimental and simulation data for this telescope/cavity setup (beam waist @ waist position)

Overall agreement:

- Size of waist: $\leq 50\%$
- Position of waist: $\leq 20\%$

Figure: Typical simulation setup (courtesy Léa Lhopital, Optics Group)

Conclusions and perspectives

- New 3D simulation tool to respond to specific aims: **Ghost beam hunting and large-scale setups**
- Next steps: **confirm test cases, implement following features, communicate** (theia.hopto.org:56000)

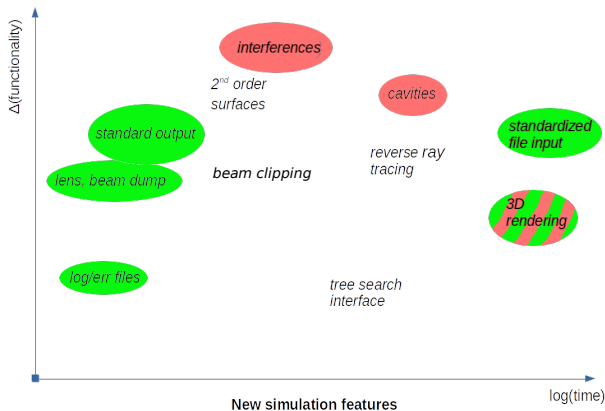


Figure: The near future of theia

- [1] E. Kochkina, G. Wanner, D. Schmelzer, M. Tröbs, G. Heinzel: *Modeling of the General Astigmatic Gaussian Beam and its Propagation through 3D Optical Systems*, Applied Optics 24 (2013)

- Optimum placement of lens $L2$ for maximum width of transmitted beam? For position of waist?

⇒ Small script



Figure: Optical setup (left) and Python script (right) for an optimization case

```
#simulation object
simu = simulation.Simulation(FName = 'optimization')
simu.LName = 'Optimizing with theta!'
simu.Order = 0          Setup
simu.Threshold = 0.5*mW

#optics, the first L1 lens doesn't move
L1 = thinLens.ThinLens(X = 0*cm, Y = 0., Z = 0., Focal = 20.*cm,
    Diameter = 3.*cm, Phi1 = 180.*deg, Ref = 'L1')

bm = beam.userGaussianBeam(1.*mm, 1.*mm, 0., 0, 1004*mm, 1*W,
    X = -30*cm, Phi1 = 0, Ref = 'Beam')

# this is a list of centers for the second lens we want to try (it is around
#70 cm = L1.X + 2*Focal to respect the 2F configuration). We're trying n
#configurations around L2.X = 50
n = 500
centers = [ 40.*cm + 2.*cm*(float(i)/n) for i in range(-n, n)]
waistSizes = []
waistPositions = []      Optical data specification

# load beam
simu.InBeams = [bm]

#run the simulations in sequence!
for center in centers:
    dic = {'X': center, 'Y': 0., 'Z': 0., 'Focal': 20.*cm,
        'Diameter': 3.*cm, 'Theta': 90*deg, 'Phi1': 180.*deg, 'Ref': 'L2'}
    simu.OptList = [L1, thinLens.ThinLens(**dic)]

    #go theta go!          Optimization sequence
    simu.run()
    output = simu.BeanTreeList[0].T.T.T.T.Root

    #save output data for plotting
    waistSizes.append(output.waistSize()[0])
    waistPositions.append(output.waistPos()[0])

simu.writeCAD()

#plot the results      Plotting
plt.figure(1)
plt.subplot(211)
plt.plot(centers, waistSizes, 'r')
plt.ylabel('waistSize (m)')
plt.xlabel('center of second lens (m)')
plt.subplot(212)
plt.plot(centers, waistPositions, 'g')
plt.ylabel('waistPos (m)')
plt.xlabel('center of second lens (m)')
plt.show()
```

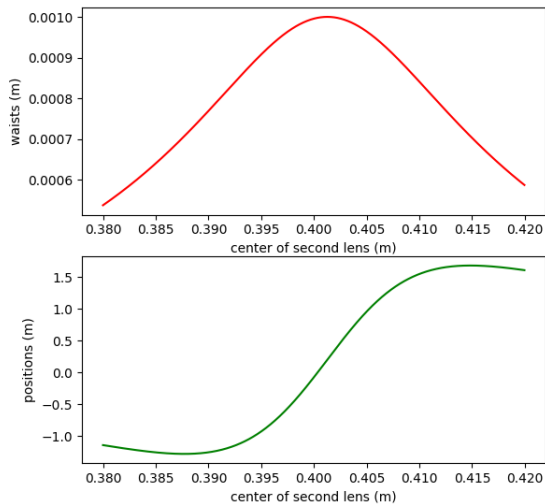


Figure: Width (*top*) and position (*bottom*) of the waist of the output beam as calculated by *theia*

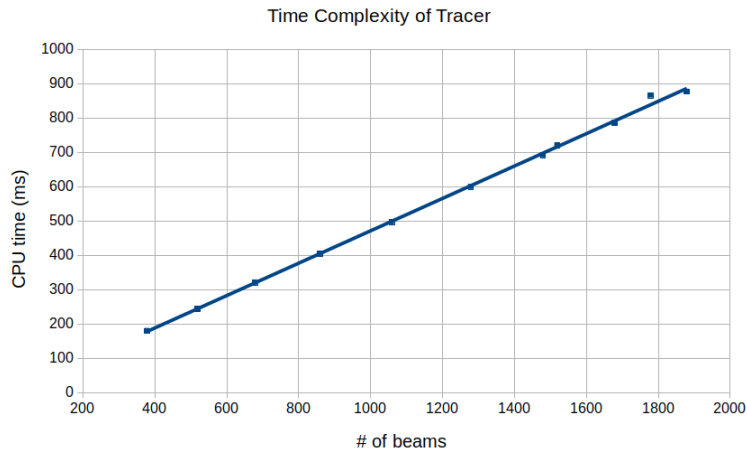
- General astigmatic Gaussian beam in an orthogonal basis (k, e_1, e_2):

$$E(\vec{r}, t) = \exp[i\eta(z) - i\frac{k}{2}t(x, y)Q(z)(x, y)]e^{i(\omega t - kz)}$$

- (x, y) is the transversal coordinate in the (e_1, e_2) basis, Q is a symmetrical tensor:

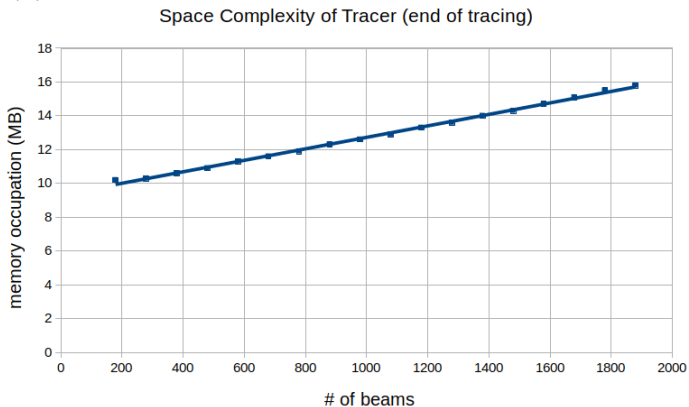
$$\begin{pmatrix} \frac{\cos^2 \theta}{q_x(z)} + \frac{\sin^2 \theta}{q_y(z)} & \frac{1}{2} \sin 2\theta \left(\frac{1}{q_x(z)} - \frac{1}{q_y(z)} \right) \\ \frac{1}{2} \sin 2\theta \left(\frac{1}{q_x(z)} - \frac{1}{q_y(z)} \right) & \frac{\sin^2 \theta}{q_x(z)} + \frac{\cos^2 \theta}{q_y(z)} \end{pmatrix}$$

- Specification parameters: $\theta, q_{x,y} \in \mathbb{C}$, (e_1, e_2) basis.
- **Approximations:** ROC(beam) \gg ROC(surface) (+ paraxial)
- Geometric optics: no approximation



- $\text{CPU} = 0.47\text{ms} \times (\# \text{ beams})$ ($R^2 = 99.95\%$)

mem (MB)



- Mem. = 9,3MB + 3,4kB/beam ($R^2 = 99.76\%$)