
gtrace

Release 0.2.1

Yoichi Aso

Apr 30, 2021

CONTENTS:

1	Introduction	1
2	Basic Concepts	3
2.1	2D plane	3
2.2	Direction	3
2.3	Beam	4
2.4	Mirror	5
3	Propagation of a beam	7
3.1	Hitting a mirror	7
4	Tutorial	9
4.1	gtrace Tutorial	9
5	API Reference	21
5.1	gtrace	21
6	Indices and tables	159
	Python Module Index	161
	Index	163

INTRODUCTION

`gtrace` is a python package to trace the propagation of Gaussian beams among optical components such as mirrors and lenses. The features of `gtrace` include:

- Automatically track the Gaussian beam propagation, i.e. q-parameter change, using the ABCD matrix method.
- Reflection and refraction at interface surfaces are properly treated.
- Automatically track the optical distance traveled by a beam through dielectric media.
- Sequential or non-sequential trace modes are available.
- Exporting the results to DXF files

The main motivation behind the development of `gtrace` was to help the design of the optical layout for the [KAGRA](#) interferometer. The task is not so trivial because we had to satisfy many constraints at the same time. Manually placing mirrors on a CAD and adjusting the distances between the mirrors and their orientations was not an option. We needed a way to automatically adjust the optical layout with a computer to satisfy our requirements. In order to do this, we needed a tool to represent an optical layout convenient for computer processings. `gtrace` represents mirrors and beams as class objects in python. Then the propagation of the beams can be treated as interactions between the beams and the mirrors. This way, we can automate the optimization of the [KAGRA](#) optical layout.

The main ingredients of `gtrace` are mirrors and beams. A mirror is represented as an instance of `Mirror` class. You place mirrors in a 2D plane. You can set various properties of a mirror such as size, curvature, reflectivities, wedge angle etc. Then you launch a beam, an instance of `GaussianBeam` class, from a certain point in the 2D plane. As the beam hits mirrors, it is divided into several sub-beams, such as reflections and deflected beams. These newly generated beam objects are available to you for further propagation. In the non-sequential mode, these sub-beams are automatically propagated until one of the termination conditions is satisfied. The termination conditions include, not hitting any mirror, power is below a certain threshold and so on. At the end of the propagation, you will have a collection of beam objects generated by the collision with the mirrors. You can export the mirrors and the beams into a DXF file.

`gtrace` is at this moment limited to 2D optical layouts. This limitation might be lifted in the future.

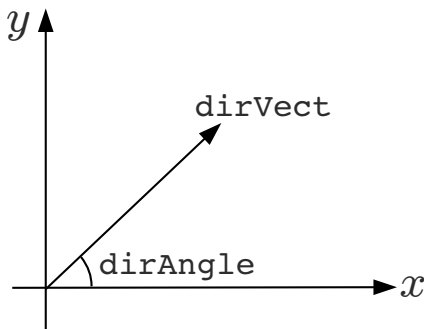
BASIC CONCEPTS

In this section, basic concepts of gtrace will be introduced.

2.1 2D plane

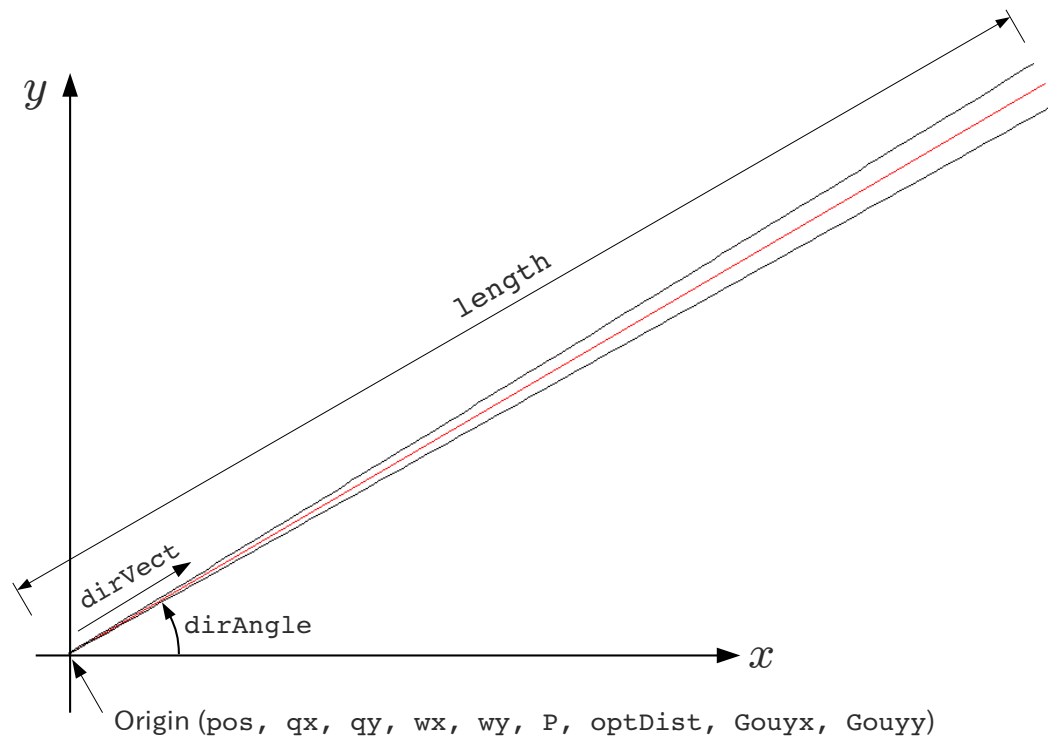
In gtrace world, an optical system will be placed on a two dimensional plane. A location on the plane is specified by a set of Cartesian coordinates (x, y). This just a normal x-y plane. The origin of the axes is at the lower left of the plane. The X-axis extends horizontally to the right. The Y-axis goes up vertically. Nothing more to add here.

2.2 Direction



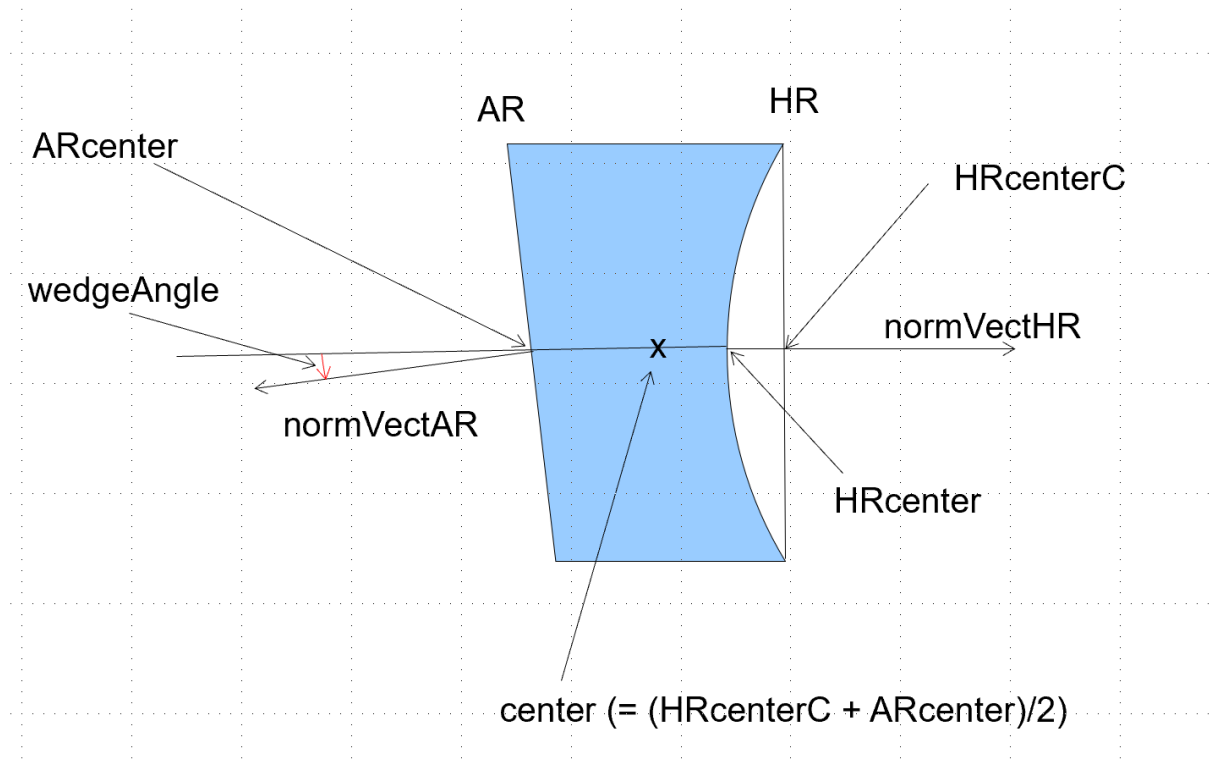
While working with optical layouts, one often has to specify a direction in the 2D plane such as the orientation of a mirror or the propagation direction of a beam. In gtrace, in most cases, a direction can be specified in two ways. One way is to use an angle measured from the X-axis in counter clockwise (`dirAngle` in the figure above). The other way is to use a 2D vector of length 1. If a direction can be specified either way, you only have to specify it in one of those methods. For example, the *GaussianBeam* class has an attribute called `dirVect`. It holds a 2D vector in the form of `numpy.Array`. The *GaussianBeam* class also has an attribute called `dirAngle`, which holds the angle of the beam propagation direction measured from the X-axis in radian. When one of the two attributes is changed, the other is updated automatically to be consistent with the modification. Therefore, you don't have to worry about the consistency. For the direction vector, it is also automatically normalized. Therefore, you can assign it a vector of any norm.

2.3 Beam



A Gaussian beam is represented by an instance of *GaussianBeam* class. The most fundamental properties of a beam is its position (pos) and the direction of propagation (dirVect or dirAngle).

2.4 Mirror



Mirror is a basic optical component in gtrace. Even though the name is "Mirror", it can represent a transparent optical window, a prism, a spherical lense, light absorbing plate (like black glass) and so on. A mirror object has two surfaces, called HR and AR. These surfaces can be flat or curved. Curved surfaces are spherical. If you need a cylindrical surface, use `CyMirror` class instead.

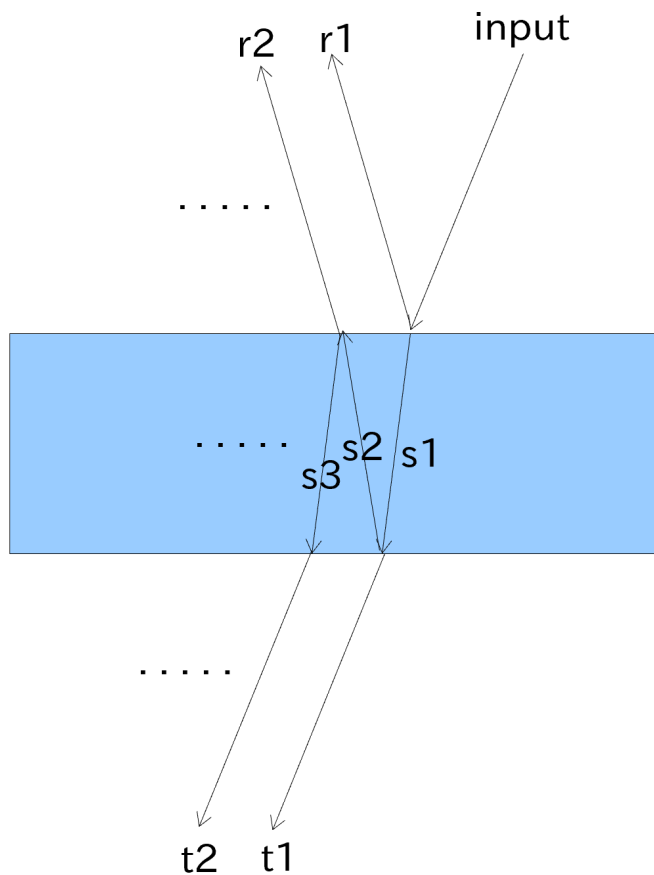
The parameters of a Mirror object are shown in the figure above.

PROPAGATION OF A BEAM

A beam can be propagated either by manually propagate for a certain distance or tell gtrace to propagate it until hitting a particular mirror.

The manual propagation can be performed by calling, `beam.propagate(d)` where `d` is the distance to propagate.

3.1 Hitting a mirror



By calling `Mirror.hitFromHR(beam)`, you can tell gtrace to propagate the beam until it hits the mirror. If the beam indeed hits the mirror, gtrace will generate a set of beam objects produced by the interactions (reflection and refraction) of the incident beam with the mirror.

The generated beams are given the names indicated in the figure above. The beam objects will be returned as a dictionary with the name of a beam as a key.

TUTORIAL

You can find a basic tutorial of gtrace in its [GitHub repository](#).

You can download the gtrace source tree by:

```
git clone https://github.com/asoy01/gtrace.git
```

Or the direct link to the tutorial file is here: <https://github.com/asoy01/gtrace/blob/master/Manuals/Tutorial/gtrace-tutorial.ipynb>

Or you can find it directly here:

4.1 gtrace Tutorial

This Jupyter notebook walks you through how to use the gtrace package.

4.1.1 Import modules

First, we import related modules

```
[1]: import gtrace.beam as beam # Gaussian beam module
import gtrace.optcomp as opt # Optical components
import gtrace.draw as draw # A module to draw results into CAD files
from gtrace.draw.tools import drawAllBeams, drawAllOptics, transAll, rotateAll
↪ #Utility functions for drawing
import gtrace.draw.renderer as renderer
from gtrace.unit import * # A convenience module to represent various modules
import gtrace.optics.gaussian as gauss # A utility module for Gaussian beams
from gtrace.nonsequential import non_seq_trace #Non-sequential trace
from gtrace.optics.geometric import vector_normalize #A function to normalize a vector

import numpy as np #Numpy
pi = np.pi #Just for brevity
```

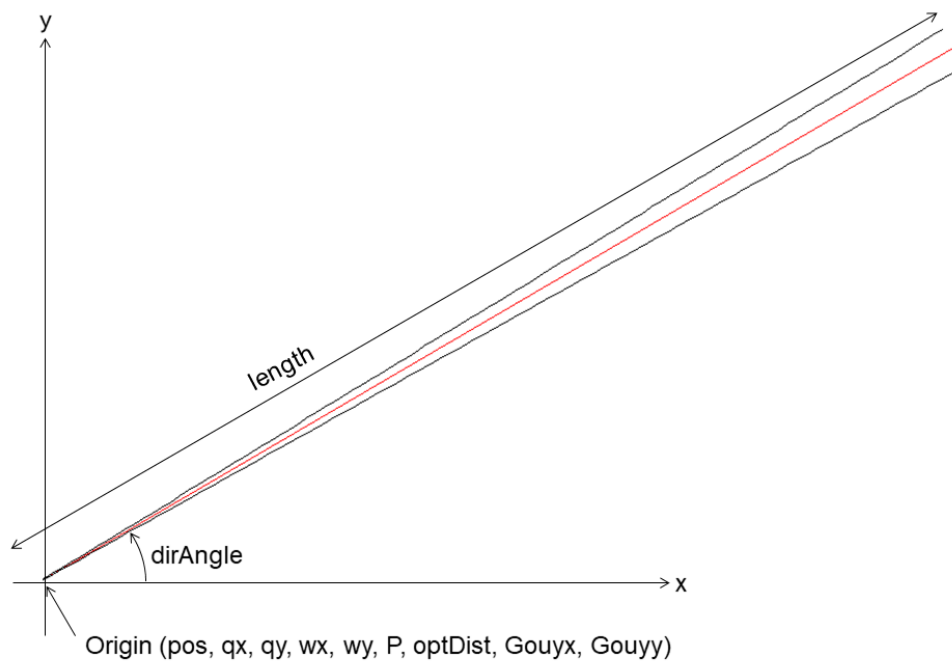
4.1.2 Coordinates and directions

In gtrace, everything is put on a 2D plane and we use a Cartesian coordinate system with x- and y-axis.

Sometimes, you may want to specify a direction in the 2D space, such as a beam propagation direction or the direction of a mirror surface represented by its normal vector. You can use either a unit vector or an angle to specify a direction. When using an angle, it is always measured from the X-axis in counter-clockwise. Therefore, a unit vector having a direction angle θ has components $(\cos \theta, \sin \theta)$.

4.1.3 Gaussian beam object

A Gaussian beam object:



```
[2]: #q-parameter of the beam
q0 = gauss.Rw2q(ROC=np.inf, w=0.3*mm)

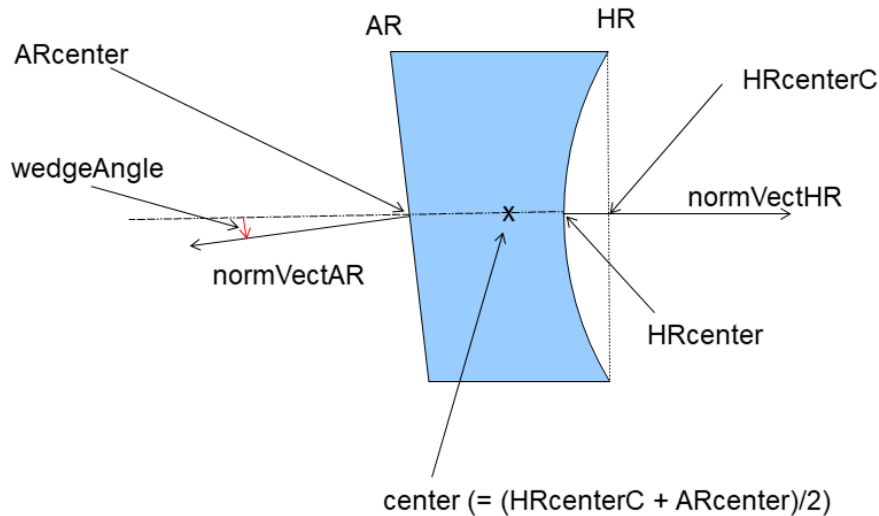
#Create a GaussianBeam object.
b0 = beam.GaussianBeam(q0=q0, wl=1064*nm, length=30*cm, P=1.0)

#Set the direction angle of the beam to 10deg from the global x-axis.
b0.dirAngle = deg2rad(10)

#Set the position of the origin of the beam
b0.pos = (0.0, 0.0)
```

4.1.4 Define mirrors

Mirror object:



```
[5]: M1 = opt.Mirror(HRcenter=[50*cm, 10*cm], normAngleHR=pi,
                    diameter=25*cm, thickness=10*cm,
                    wedgeAngle=deg2rad(0.25), inv_ROC_HR=1./(120*cm),
                    inv_ROC_AR=0,
                    Refl_HR=0.9, Trans_HR=1-0.9,
                    Refl_AR=500*ppm, Trans_AR=1-500*ppm,
                    n=1.45, name='M1')

M2 = opt.Mirror(HRcenter=[0*cm, 18*cm], normAngleHR=deg2rad(5.0),
                diameter=15*cm, thickness=5*cm,
                wedgeAngle=deg2rad(0.25), inv_ROC_HR=-1./(350*cm),
                inv_ROC_AR=0,
                Refl_HR=0.9, Trans_HR=1-0.9,
                Refl_AR=500*ppm, Trans_AR=1-500*ppm,
                n=1.45, name='M2')

M3 = opt.Mirror(HRcenter=[30*cm, 30*cm], normAngleHR=deg2rad(21.3),
                diameter=15*cm, thickness=5*cm,
                wedgeAngle=deg2rad(1), inv_ROC_HR=1./(350*cm),
                inv_ROC_AR=0,
                Refl_HR=0.9, Trans_HR=1-0.9,
                Refl_AR=500*ppm, Trans_AR=1-500*ppm,
                n=1.45, name='M3')
```

4.1.5 Sequential beam trace

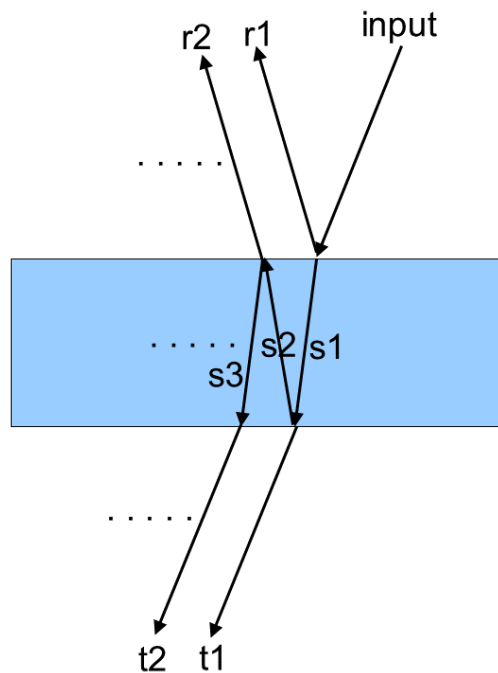
Prepare a dictionary to save beams

```
[6]: beamDict = {}
```

Hit the mirror M1 with the beam b1 from the HR side

```
[7]: beams = M1.hitFromHR(b0, order=2)
```

Returned object (beams) is a dictionary containing the resulting beams. The names of the beams are the following.



Trace the remaining beam path

```
[9]: #Save the incident beam to the beam dictionary
beamDict['input_beam'] = beams['input']

#Save other beams
beamDict['M1s1'] = beams['s1']
beamDict['M1t1'] = beams['t1']
beamDict['M1s2'] = beams['s2']
beamDict['M1s3'] = beams['s3']
beamDict['M1t2'] = beams['t2']

#Reflected beam from M1
b = beams['r1']

#Hit M2
beams = M2.hitFromHR(b, order=2)

#Save the beam from M1 to M2
beamDict['M1toM2'] = beams['input']
```

(continues on next page)

(continued from previous page)

```

beamDict['M2s1'] = beams['s1']
beamDict['M2t1'] = beams['t1']

#Reflected beam from M2
b = beams['r1']

#Hit M3 from AR
beams = M3.hitFromAR(b, order=2)

#Save beams
beamDict['M2toM3'] = beams['input']
beamDict['M3s1'] = beams['s1']
beamDict['M3t1'] = beams['t1']

```

4.1.6 Draw the results

```

[10]: #Create a canvas object
cnv = draw.Canvas()

#Add a layer to the canvas
cnv.add_layer("main_beam", color=(0,0,0))

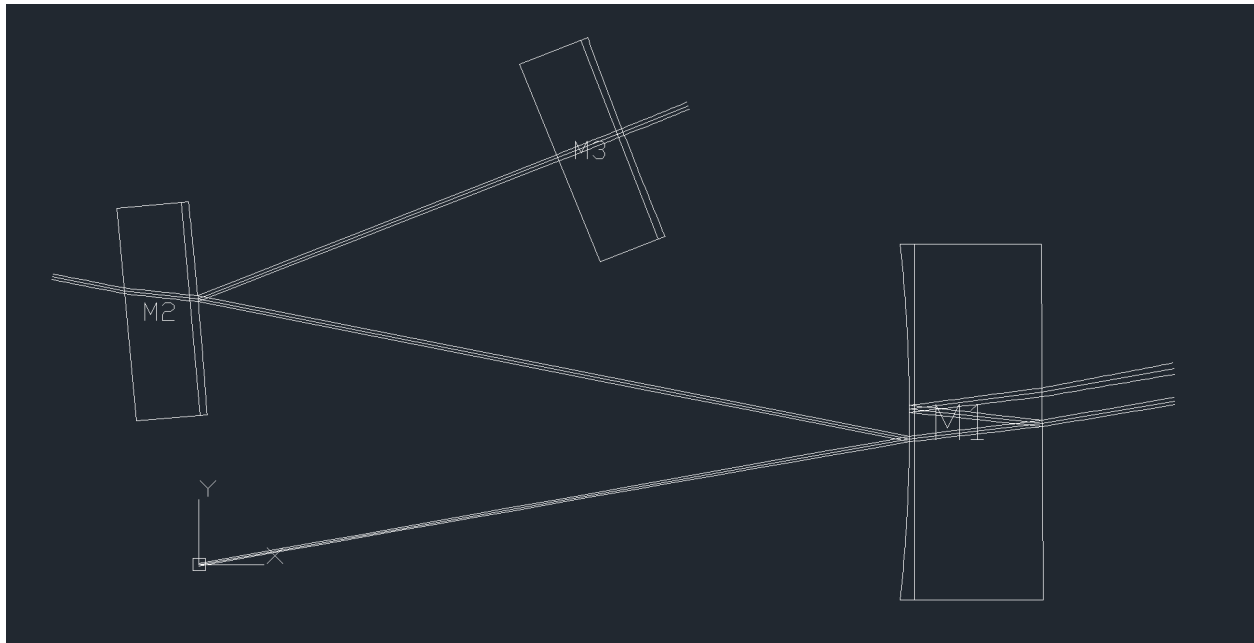
#Draw all the beams in beamDict
drawAllBeams(cnv, list(beamDict.values()), drawWidth=True, sigma=3.0, drawPower=False,
              drawROC=False, drawGouy=False, drawOptDist=False, layer='main_beam',
              fontSize=0.01)

#Draw the mirror
drawAllOptics(cnv, [M1,M2,M3])

#Save the result as a DXF file
renderer.renderDXF(cnv, 'SeqTrace.dxf')

```

The generated DXF file looks like this.



4.1.7 Non-sequential trace

In the above example, we instructed which beam should hit which optic explicitly. Here, we only specify an input beam and a set of optics. Then let the beam go around until specified criteria are met. This is called “Non-Sequential Trace”.

Perform non-sequential trace

```
[11]: # Trace beams until the power is less than 1e-6 or 30th order internal reflections
beams = non_seq_trace([M1,M2,M3], b0, order=30, power_threshold=1e-6)
```

Draw results

```
[13]: #Create a canvas object
cnv = draw.Canvas()

#Add a layer to the canvas
cnv.add_layer("main_beam", color=(0,0,0))

#Draw all beams
drawAllBeams(cnv, beams, drawWidth=True, sigma=3.0, drawPower=False,
             drawROC=False, drawGouy=False, drawOptDist=False, layer='main_beam',
             fontSize=0.01)

#Draw the mirror
drawAllOptics(cnv, [M1,M2,M3])

#Save the result as a DXF file
renderer.renderDXF(cnv, 'NonSeq.dxf')
```

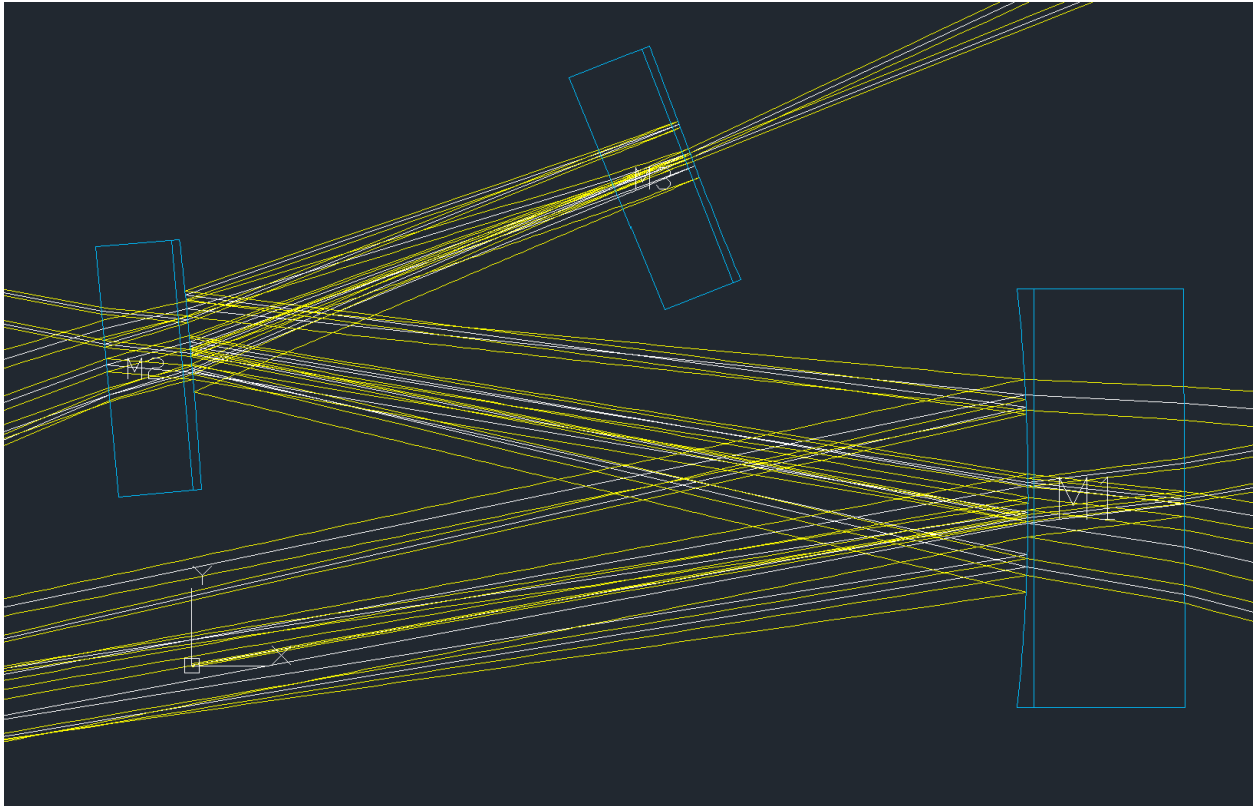
```
[49]: a=cnv.layers['main_beam']

[52]: b=a.shapes[0]

[54]: b.stop

[54]: (0.4999415205870579, 0.0881531788559148)
```

The generated DXF file looks like this.



4.1.8 KAGRA Input Mode Cleaner

Parameters

```
[14]: nsilica = 1.44967

MC_Dia = 10.0*cm
MC_Thick = 3.0*cm
MCe_ROC = 37.3

MCi_Refl = 0.9937
MCo_Refl = 0.9937
MCe_Refl = 0.9999
AR_Refl = 0.1/100

pos_MCi = np.array([-0.25, 0.0])
```

(continues on next page)

(continued from previous page)

```
pos_MCo = np.array([0.25, 0.0])
pos_MCe = np.array([0, 2.63986994e+01])
```

Define MC mirrors

```
[15]: MCI = opt.Mirror(HRcenter=[0,0], normAngleHR=0.0,
                      diameter=MC_Dia, thickness=MC_Thick,
                      wedgeAngle=-deg2rad(2.5), inv_ROC_HR=0.0,
                      Refl_HR=MCI_Refl, Trans_HR=1-MCI_Refl,
                      Refl_AR=AR_Refl, Trans_AR=1-AR_Refl,
                      n=nsilica, name='MCI')

MCo = opt.Mirror(HRcenter=[0,0], normAngleHR=0.0,
                 diameter=MC_Dia, thickness=MC_Thick,
                 wedgeAngle=deg2rad(2.5), inv_ROC_HR=0.0,
                 Refl_HR=MCo_Refl, Trans_HR=1-MCo_Refl,
                 Refl_AR=AR_Refl, Trans_AR=1-AR_Refl,
                 n=nsilica, name='MCo')

MCe = opt.Mirror(HRcenter=[0,0], normAngleHR=0.0,
                 diameter=MC_Dia, thickness=MC_Thick,
                 wedgeAngle=deg2rad(2.5), inv_ROC_HR=1.0/MCe_ROC,
                 Refl_HR=MCe_Refl, Trans_HR=1-MCe_Refl,
                 Refl_AR=AR_Refl, Trans_AR=1-AR_Refl,
                 n=nsilica, name='MCe')
```

Put MC mirrors

```
[16]: #Put mirrors in position
MCI.HRcenter = pos_MCI
MCo.HRcenter = pos_MCo
MCe.HRcenter = pos_MCe

#Align the mirrors to form a triangular cavity
v1 = vector_normalize(pos_MCI - pos_MCo)
v2 = vector_normalize(pos_MCe - pos_MCo)
MCo.normVectHR = (v1+v2)/2

v2 = vector_normalize(pos_MCe - pos_MCI)
MCI.normVectHR = (-v1+v2)/2

MCe.normVectHR = np.array([0.0, -1.0])
```

4.1.9 MC eigen mode

Now we compute the eigen mode of the MC by tracing a beam round trip of the MC then extracting the ABCD matrix

```
[17]: #Test beam (beam parameters does not matter)
b = beam.GaussianBeam(q0=gauss.Rw2q(ROC=np.inf, w=1*mm), wl=1064*nm)

#Put the beam on the surface of MCi
b.pos = MCi.HRcenter

#Direct the beam to the center of MCo
b.dirVect = MCo.HRcenter - b.pos

#Hit MCo
beams = MCo.hitFromHR(b)
b = beams['r1']

#Hit MCo
beams = MCo.hitFromHR(b)
b = beams['r1']

#Hit MCi
beams = MCi.hitFromHR(b)
b = beams['r1']

#Extract the round trip ABCD matrix in the horizontal direction
A = b.Mx[0,0]
B = b.Mx[0,1]
C = b.Mx[1,0]
D = b.Mx[1,1]
#q-parameter of the MC eigenmode beam
qxMC = 1.0 / ((D-A) / (2*B) - 1j*np.sqrt(4-(A+D)**2) / (2*B))

#Extract the round trip ABCD matrix in the vertical direction
A = b.My[0,0]
B = b.My[0,1]
C = b.My[1,0]
D = b.My[1,1]
#q-parameter of the MC eigenmode beam
qyMC = 1.0 / ((D-A) / (2*B) - 1j*np.sqrt(4-(A+D)**2) / (2*B))

#Update the q-parameter of b
b.qx = qxMC
b.qy = qyMC

#Name it bMC
bMC = b.copy()
```

Waist size and position of the MC eigenmode (measured from the MCi HR surface)

```
[18]: bMC.waist()
[18]: {'Waist Size': (0.002388588001545699, 0.002388775565011432),
      'Waist Position': (0.250000000000000416, 0.250000000000000505)}
```

Trace the beam in the MC again

```
[19]: beamDict = {}

#Hit MCo
beams = MCo.hitFromHR(b, order=1)
beamDict['MCitoMCo'] = beams['input']
beamDict['MCos1'] = beams['s1']
beamDict['MCot1'] = beams['t1']

b = beams['r1']

#Hit MCE
beams = MCE.hitFromHR(b, order=1)
beamDict['MCotoMCE'] = beams['input']
beamDict['MCes1'] = beams['s1']
beamDict['MCet1'] = beams['t1']

b = beams['r1']

#Hit MCI
beams = MCI.hitFromHR(b, order=1)
beamDict['MCetoMCI'] = beams['input']
beamDict['MCis1'] = beams['s1']
beamDict['MCit1'] = beams['t1']
```

4.1.10 Draw MC beams

```
[20]: #Create a canvas object
cnv = draw.Canvas()

#Add a layer to the canvas
cnv.add_layer("main_beam", color=(0,0,0))

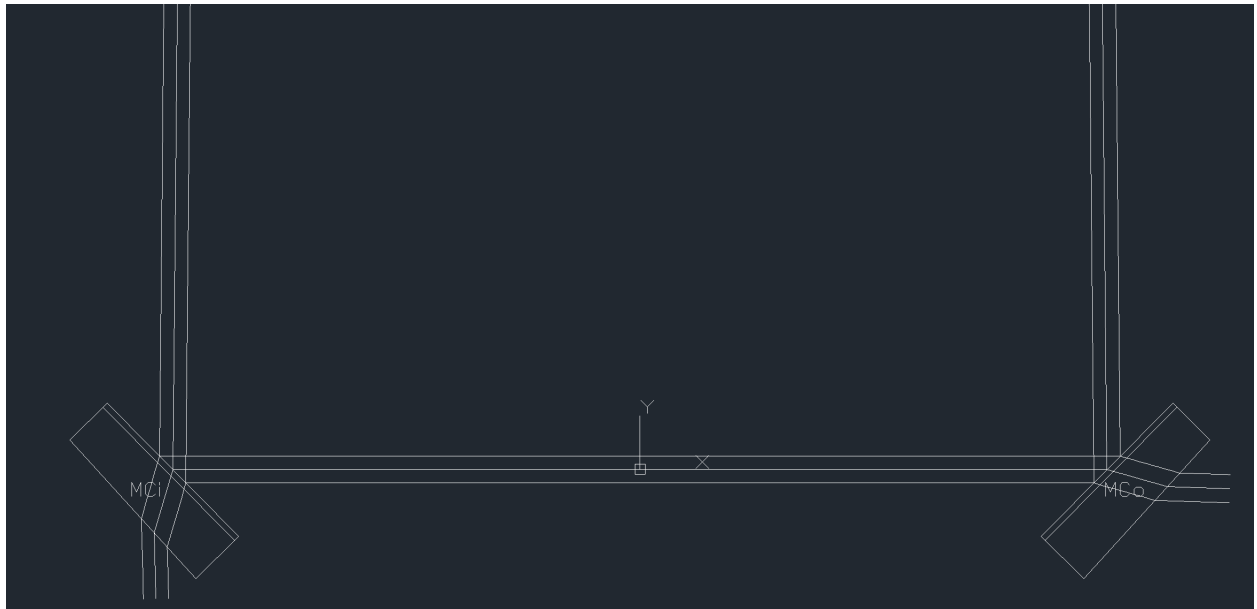
#Draw all beams

drawAllBeams(cnv, list(beamDict.values()), drawWidth=True, sigma=3.0, drawPower=False,
              drawROC=False, drawGouy=False, drawOptDist=False, layer='main_beam',
              fontSize=0.01)

#Draw the mirror
drawAllOptics(cnv, [MCI,MCo,MCE])

#Save the result as a DXF file
renderer.renderDXF(cnv, 'MC.dxf')
```

The generated DXF file looks like this:



[]:

API REFERENCE

This page contains auto-generated API reference documentation¹.

5.1 gtrace

This package provides necessary classes and functions for tracing the propagation of Gaussian beams among mirrors and lenses.

5.1.1 Subpackages

`gtrace.draw`

Submodules

`gtrace.draw.draw`

Drawing classes for gtrace

Module Contents

Classes

<i>Canvas</i>	Canvas class
<i>Layer</i>	Layer class
<i>Shape</i>	Shape class
<i>Line</i>	Line class
<i>PolyLine</i>	A light weight poly-line
<i>Rectangle</i>	A rectangle
<i>Circle</i>	A circle
<i>Arc</i>	An arc
<i>Text</i>	Text

¹ Created with sphinx-autoapi

Attributes

pi

__author__

__copyright__

__credits__

__license__

__version__

__maintainer__

__email__

__status__

```
gtrace.draw.draw.pi
gtrace.draw.draw.__author__ = Yoichi Aso
gtrace.draw.draw.__copyright__ = Copyright 2011-2021, Yoichi Aso
gtrace.draw.draw.__credits__ = ['Yoichi Aso']
gtrace.draw.draw.__license__ = BSD
gtrace.draw.draw.__version__ = 0.2.1
gtrace.draw.draw.__maintainer__ = Yoichi Aso
gtrace.draw.draw.__email__ = yoichi.aso@nao.ac.jp
gtrace.draw.draw.__status__ = Beta
class gtrace.draw.draw.Canvas (unit='m')
    Bases: object
    Canvas class
    add_layer (self, name, color=(0, 0, 0))
    add_shape (self, shape, layername)
class gtrace.draw.draw.Layer (name, color=(0, 0, 0))
    Bases: object
    Layer class
    add_shape (self, shape)
class gtrace.draw.draw.Shape
    Bases: object
    Shape class
```

class gtrace.draw.draw.**Line** (*start, stop, thickness=0*)

Bases: *Shape*

Line class

exception gtrace.draw.draw.**NumberOfElementError**

Bases: *BaseException*

Common base class for all exceptions

__init__ (*self, message*)

class gtrace.draw.draw.**PolyLine** (*x, y, thickness=0*)

Bases: *Shape*

A light weight poly-line

class gtrace.draw.draw.**Rectangle** (*point, width, height, thickness=0*)

Bases: *Shape*

A rectangle

class gtrace.draw.draw.**Circle** (*center, radius, thickness=0*)

Bases: *Shape*

A circle

class gtrace.draw.draw.**Arc** (*center, radius, startangle, stopangle, thickness=0, angle_in_rad=True*)

Bases: *Shape*

An arc

Note that angles are stored in rad.

class gtrace.draw.draw.**Text** (*text, point, height=1.0, rotation=0.0, angle_in_rad=True*)

Bases: *Shape*

Text

Note that angles are stored in rad.

gtrace.draw.dxf

dxf.py - a DXF export library for python

Sample code:

```
import dxf
d = dxf.DXF('test.dxf')
d.add_layer('ABC', color=5)
d.add_entity(dxf.Line((1.5,5), (56,-89)), 'ABC')
d.save_to_file()
```

Module Contents

Classes

<i>DXF</i>	A DXF file class.
<i>Layer</i>	Layer class
<i>Entity</i>	A graphic entity
<i>Line</i>	A line entity

continues on next page

Table 3 – continued from previous page

<i>LwPolyLine</i>	A light weight poly-line
<i>Rectangle</i>	A rectangle
<i>Circle</i>	A circle entity
<i>Arc</i>	An arc entity
<i>Text</i>	Text

Functions

<i>color_encode</i> (color)	Given a set of RGB values for a color, find the closest matching
<i>test_func</i> ()	

Attributes

<i>pi</i>
<i>color_table</i>
<i>header_template1</i>
<i>header_template2</i>
<i>header_template3</i>
<i>classes_template</i>
<i>tables_template1</i>
<i>vport_template</i>
<i>linetype_template</i>
<i>layer_template1</i>
<i>layer_template2</i>
<i>style_template</i>
<i>view_template</i>
<i>misctable_template</i>
<i>blockrecords_template</i>
<i>tables_template2</i>

continues on next page

Table 5 – continued from previous page

blocks_template

entities_template1

entities_template0

objects_template

`gtrace.draw.dxf.pi``class gtrace.draw.dxf.DXF (filename='drawing.dxf')``Bases: object``A DXF file class.``add_layer (self, name, color=1, ltype='Continuous')``add_entity (self, entity, layername)``save_to_file (self)``Save the DXF file``class gtrace.draw.dxf.Layer (name, handle, color=1, ltype='Continuous')``Bases: object``Layer class``add_entity (self, entity)``class gtrace.draw.dxf.Entity``Bases: object``A graphic entity``set_handle (self, handle)``report_min_max (self)``Return the coordinates of the lower left and the upper right corners of the drawing. Return value: ((xmin, ymin), (xmax, ymax))``draw (self)``class gtrace.draw.dxf.Line (start, stop, thickness=0)``Bases: Entity``A line entity``report_min_max (self)``Return the coordinates of the lower left and the upper right corners of the drawing. Return value: ((xmin, ymin), (xmax, ymax))``draw (self, layername)``exception gtrace.draw.dxf.NumberOfElementError``Bases: BaseException``Common base class for all exceptions``__init__ (self, message)`

```
class gtrace.draw.dxf.LwPolyLine (x, y, thickness=0)
    Bases: Entity
    A light weight poly-line

    report_min_max (self)
        Return the coordinates of the lower left and the upper right corners of the drawing. Return value: ((xmin,
        ymin), (xmax, ymax))

    draw (self, layername)

class gtrace.draw.dxf.Rectangle (point, width, height, thickness=0)
    Bases: Entity
    A rectangle

    report_min_max (self)
        Return the coordinates of the lower left and the upper right corners of the drawing. Return value: ((xmin,
        ymin), (xmax, ymax))

    draw (self, layername)

class gtrace.draw.dxf.Circle (center, radius, thickness=0)
    Bases: Entity
    A circle entity

    report_min_max (self)
        Return the coordinates of the lower left and the upper right corners of the drawing. Return value: ((xmin,
        ymin), (xmax, ymax))

    draw (self, layername)

class gtrace.draw.dxf.Arc (center, radius, startangle, stopangle, thickness=0)
    Bases: Entity
    An arc entity

    report_min_max (self)
        Return the coordinates of the lower left and the upper right corners of the drawing. Return value: ((xmin,
        ymin), (xmax, ymax))

    draw (self, layername)

class gtrace.draw.dxf.Text (text, point, height=1.0, rotation=0.0)
    Bases: Entity
    Text

    report_min_max (self)
        Return the coordinates of the lower left and the upper right corners of the drawing. Return value: ((xmin,
        ymin), (xmax, ymax))

    draw (self, layername)

gtrace.draw.dxf.color_encode (color)
    Given a set of RGB values for a color, find the closest matching one from the pre-defined colors in the DXF
    specification. Then return its color code (an integer in 1 to 255).

    = Input = color: A tuple of three numbers in the range of 0-255, i.e. (R,G,B)

    = Return = best_color_num: integer

gtrace.draw.dxf.color_table

gtrace.draw.dxf.header_template1 = Multiline-String
```

```

1      0
2  SECTION
3      2
4  HEADER
5      9
6  $ACADVER
7      1
8  AC1024
9      9
10 $ACADMAINTVER
11     70
12         6
13     9
14 $DWGCODEPAGE
15     3
16 ANSI_1252
17     9
18 $INSBASE
19     10
20 0.0
21     20
22 0.0
23     30
24 0.0

```

`gtrace.draw.dxf.header_template2 = Multiline-String`

```

1      9
2  $EXTMIN
3      10
4  1.0
5      20
6  0.0
7      30
8  0.0
9      9
10 $EXTMAX
11     10
12 73.0
13     20
14 41.0
15     30
16 0.0
17     9
18 $LIMMIN
19     10
20 0.0
21     20
22 0.0
23     9
24 $LIMMAX
25     10
26 1000.0
27     20
28 1000.0

```

`gtrace.draw.dxf.header_template3 = Multiline-String`

```
1      9
2  $ORTHOMODE
3      70
4      0
5      9
6  $REGENMODE
7      70
8      1
9      9
10 $FILLMODE
11      70
12      1
13      9
14 $QTEXTMODE
15      70
16      0
17      9
18 $MIRRTEXT
19      70
20      1
21      9
22 $LTSCALE
23      40
24      1.0
25      9
26 $ATTMODE
27      70
28      1
29      9
30 $TEXTSIZE
31      40
32      2.5
33      9
34 $TRACEWID
35      40
36      1.0
37      9
38 $TEXTSTYLE
39      7
40 Standard
41      9
42 $CLAYER
43      8
44 Outline
45      9
46 $CELTYPE
47      6
48 ByLayer
49      9
50 $CECOLOR
51      62
52      256
53      9
54 $CELTSCALE
55      40
56      1.0
57      9
```

(continues on next page)

(continued from previous page)

```
58 $DISPSILH
59   70
60     0
61     9
62 $DIMSCALE
63   40
64   1.0
65     9
66 $DIMASZ
67   40
68   2.5
69     9
70 $DIMEXO
71   40
72   0.625
73     9
74 $DIMDLI
75   40
76   3.75
77     9
78 $DIMRND
79   40
80   0.0
81     9
82 $DIMDLE
83   40
84   0.0
85     9
86 $DIMEXE
87   40
88   1.25
89     9
90 $DIMTP
91   40
92   0.0
93     9
94 $DIMTM
95   40
96   0.0
97     9
98 $DIMTXT
99   40
100  2.5
101     9
102 $DIMCEN
103   40
104  2.5
105     9
106 $DIMTSZ
107   40
108   0.0
109     9
110 $DIMTOL
111   70
112     0
113     9
114 $DIMLIM
```

(continues on next page)

(continued from previous page)

```
115 70
116 0
117 9
118 $DIMTIH
119 70
120 0
121 9
122 $DIMTOH
123 70
124 0
125 9
126 $DIMSE1
127 70
128 0
129 9
130 $DIMSE2
131 70
132 0
133 9
134 $DIMITAD
135 70
136 1
137 9
138 $DIMZIN
139 70
140 8
141 9
142 $DIMBLK
143 1
144
145 9
146 $DIMASO
147 70
148 1
149 9
150 $DIMSHO
151 70
152 1
153 9
154 $DIMPOST
155 1
156
157 9
158 $DIMAPOST
159 1
160
161 9
162 $DIMALT
163 70
164 0
165 9
166 $DIMALTD
167 70
168 3
169 9
170 $DIMALTF
171 40
```

(continues on next page)

(continued from previous page)

```
172 0.03937007874016
173 9
174 $DIMLFAC
175 40
176 1.0
177 9
178 $DIMTOFL
179 70
180 1
181 9
182 $DIMTVP
183 40
184 0.0
185 9
186 $DIMTIX
187 70
188 0
189 9
190 $DIMSOXD
191 70
192 0
193 9
194 $DIMSAH
195 70
196 0
197 9
198 $DIMBLK1
199 1
200
201 9
202 $DIMBLK2
203 1
204
205 9
206 $DIMSTYLE
207 2
208 ISO-25
209 9
210 $DIMCLRD
211 70
212 0
213 9
214 $DIMCLRE
215 70
216 0
217 9
218 $DIMCLRT
219 70
220 0
221 9
222 $DIMTFAC
223 40
224 1.0
225 9
226 $DIMGAP
227 40
228 0.625
```

(continues on next page)

(continued from previous page)

```
229     9
230 $DIMJUST
231     70
232     0
233     9
234 $DIMSD1
235     70
236     0
237     9
238 $DIMSD2
239     70
240     0
241     9
242 $DIMENTOLJ
243     70
244     0
245     9
246 $DIMENTZIN
247     70
248     8
249     9
250 $DIMALTZ
251     70
252     0
253     9
254 $DIMALTTZ
255     70
256     0
257     9
258 $DIMUPT
259     70
260     0
261     9
262 $DIMDEC
263     70
264     2
265     9
266 $DIMENTDEC
267     70
268     2
269     9
270 $DIMALTU
271     70
272     2
273     9
274 $DIMALTTD
275     70
276     3
277     9
278 $DIMTXSTY
279     7
280 Standard
281     9
282 $DIMAUNIT
283     70
284     0
285     9
```

(continues on next page)

(continued from previous page)

```

286 $DIMADEC
287   70
288     0
289     9
290 $DIMALTRND
291   40
292  0.0
293     9
294 $DIMAZIN
295   70
296     0
297     9
298 $DIMDSEP
299   70
300   44
301     9
302 $DIMATFIT
303   70
304     3
305     9
306 $DIMFRAC
307   70
308     0
309     9
310 $DIMLDRBLK
311    1
312
313     9
314 $DIMLUNIT
315   70
316     2
317     9
318 $DIMLWD
319   70
320   -2
321     9
322 $DIMLWE
323   70
324   -2
325     9
326 $DIMTMOVE
327   70
328     0
329     9
330 $DIMFXL
331   40
332  1.0
333     9
334 $DIMFXLON
335   70
336     0
337     9
338 $DIMJOGANG
339   40
340  0.7853981633974483
341     9
342 $DIMTFILL

```

(continues on next page)

(continued from previous page)

```
343 70
344 0
345 9
346 $DIMTFILLCLR
347 70
348 0
349 9
350 $DIMARCSYM
351 70
352 0
353 9
354 $DIMLTYPE
355 6
356
357 9
358 $DIMLTEX1
359 6
360
361 9
362 $DIMLTEX2
363 6
364
365 9
366 $DIMTXTDIRECTION
367 70
368 0
369 9
370 $LUNITS
371 70
372 2
373 9
374 $LUPREC
375 70
376 4
377 9
378 $SKETCHINC
379 40
380 1.0
381 9
382 $FILLETRAD
383 40
384 10.0
385 9
386 $AUNITS
387 70
388 0
389 9
390 $AUPREC
391 70
392 0
393 9
394 $MENU
395 1
396 .
397 9
398 $ELEVATION
399 40
```

(continues on next page)

(continued from previous page)

```
400 0.0
401 9
402 $PELEVATION
403 40
404 0.0
405 9
406 $THICKNESS
407 40
408 0.0
409 9
410 $LIMCHECK
411 70
412 0
413 9
414 $CHAMFERA
415 40
416 10.0
417 9
418 $CHAMFERB
419 40
420 10.0
421 9
422 $CHAMFERC
423 40
424 20.0
425 9
426 $CHAMFERD
427 40
428 0.0
429 9
430 $SKPOLY
431 70
432 0
433 9
434 $TDCREATE
435 40
436 2456231.748506945
437 9
438 $TDUCREATE
439 40
440 2456231.373506945
441 9
442 $TDUPDATE
443 40
444 2456231.748506945
445 9
446 $TDUUPDATE
447 40
448 2456231.373506945
449 9
450 $TDINDWG
451 40
452 0.0000000116
453 9
454 $TDUSRTIMER
455 40
456 0.0000000116
```

(continues on next page)

(continued from previous page)

```
457     9
458 $USRTIMER
459     70
460     1
461     9
462 $ANGBASE
463     50
464     0.0
465     9
466 $ANGDIR
467     70
468     0
469     9
470 $PDMODE
471     70
472     0
473     9
474 $PDSize
475     40
476     0.0
477     9
478 $PLINEWID
479     40
480     0.0
481     9
482 $SPLFRAME
483     70
484     0
485     9
486 $SPLINETYPE
487     70
488     6
489     9
490 $SPLINESEGS
491     70
492     8
493     9
494 $HANDSEED
495     5
496 6F
497     9
498 $SURFTAB1
499     70
500     6
501     9
502 $SURFTAB2
503     70
504     6
505     9
506 $SURFTYPE
507     70
508     6
509     9
510 $SURFU
511     70
512     6
513     9
```

(continues on next page)

(continued from previous page)

```
514 $SURFV
515 70
516 6
517 9
518 $UCSBASE
519 2
520
521 9
522 $UCSNAME
523 2
524
525 9
526 $UCSORG
527 10
528 0.0
529 20
530 0.0
531 30
532 0.0
533 9
534 $UCSXDIR
535 10
536 1.0
537 20
538 0.0
539 30
540 0.0
541 9
542 $UCSYDIR
543 10
544 0.0
545 20
546 1.0
547 30
548 0.0
549 9
550 $UCSORTHOREF
551 2
552
553 9
554 $UCSORTHOVIEW
555 70
556 0
557 9
558 $UCSORGTOP
559 10
560 0.0
561 20
562 0.0
563 30
564 0.0
565 9
566 $UCSORGBOTTOM
567 10
568 0.0
569 20
570 0.0
```

(continues on next page)

(continued from previous page)

```
571 30
572 0.0
573 9
574 $UCSORGLEFT
575 10
576 0.0
577 20
578 0.0
579 30
580 0.0
581 9
582 $UCSORGRIGHT
583 10
584 0.0
585 20
586 0.0
587 30
588 0.0
589 9
590 $UCSORGFRONT
591 10
592 0.0
593 20
594 0.0
595 30
596 0.0
597 9
598 $UCSORGBACK
599 10
600 0.0
601 20
602 0.0
603 30
604 0.0
605 9
606 $PUCSBASE
607 2
608
609 9
610 $PUCSNAME
611 2
612
613 9
614 $PUCSORG
615 10
616 0.0
617 20
618 0.0
619 30
620 0.0
621 9
622 $PUCSXDIR
623 10
624 1.0
625 20
626 0.0
627 30
```

(continues on next page)

(continued from previous page)

```
628 0.0
629 9
630 $PUCSYDIR
631 10
632 0.0
633 20
634 1.0
635 30
636 0.0
637 9
638 $PUCSORTHOREF
639 2
640
641 9
642 $PUCSORTHOVIEW
643 70
644 0
645 9
646 $PUCSORGTOP
647 10
648 0.0
649 20
650 0.0
651 30
652 0.0
653 9
654 $PUCSORGBOTTOM
655 10
656 0.0
657 20
658 0.0
659 30
660 0.0
661 9
662 $PUCSORGGLEFT
663 10
664 0.0
665 20
666 0.0
667 30
668 0.0
669 9
670 $PUCSORGRIGHT
671 10
672 0.0
673 20
674 0.0
675 30
676 0.0
677 9
678 $PUCSORGFRONT
679 10
680 0.0
681 20
682 0.0
683 30
684 0.0
```

(continues on next page)

(continued from previous page)

```
685     9
686 $PUCSORGBACK
687     10
688     0.0
689     20
690     0.0
691     30
692     0.0
693     9
694 $USERI1
695     70
696         0
697     9
698 $USERI2
699     70
700         0
701     9
702 $USERI3
703     70
704         0
705     9
706 $USERI4
707     70
708         0
709     9
710 $USERI5
711     70
712         0
713     9
714 $USERR1
715     40
716     0.0
717     9
718 $USERR2
719     40
720     0.0
721     9
722 $USERR3
723     40
724     0.0
725     9
726 $USERR4
727     40
728     0.0
729     9
730 $USERR5
731     40
732     0.0
733     9
734 $WORLDVIEW
735     70
736         1
737     9
738 $SHADEGE
739     70
740         3
741     9
```

(continues on next page)

(continued from previous page)

```

742 $SHADEDIF
743   70
744   70
745   9
746 $TILEMODE
747   70
748   1
749   9
750 $MAXACTVP
751   70
752   64
753   9
754 $PINSBASE
755   10
756  0.0
757   20
758  0.0
759   30
760  0.0
761   9
762 $PLIMCHECK
763   70
764   0
765   9
766 $PEXTMIN
767   10
768  1.0000000000000000E+20
769   20
770  1.0000000000000000E+20
771   30
772  1.0000000000000000E+20
773   9
774 $PEXTMAX
775   10
776 -1.0000000000000000E+20
777   20
778 -1.0000000000000000E+20
779   30
780 -1.0000000000000000E+20
781   9
782 $PLIMMIN
783   10
784  0.0
785   20
786  0.0
787   9
788 $PLIMMAX
789   10
790  420.0
791   20
792  297.0
793   9
794 $UNITMODE
795   70
796   0
797   9
798 $VISRETAIN

```

(continues on next page)

(continued from previous page)

```
799 70
800 1
801 9
802 $PLINEGEN
803 70
804 0
805 9
806 $PSLTSCALE
807 70
808 1
809 9
810 $TREEDEPTH
811 70
812 3020
813 9
814 $CMLSTYLE
815 2
816 Standard
817 9
818 $CMLJUST
819 70
820 0
821 9
822 $CMLSCALE
823 40
824 20.0
825 9
826 $PROXYGRAPHICS
827 70
828 1
829 9
830 $MEASUREMENT
831 70
832 1
833 9
834 $CELWEIGHT
835 370
836 -1
837 9
838 $ENDCAPS
839 280
840 0
841 9
842 $JOINSTYLE
843 280
844 0
845 9
846 $LWDISPLAY
847 290
848 0
849 9
850 $INSUNITS
851 70
852 4
853 9
854 $HYPERLINKBASE
855 1
```

(continues on next page)

(continued from previous page)

```

856
857     9
858 $STYLESHEET
859     1
860
861     9
862 $XEDIT
863 290
864     1
865     9
866 $CEPSNTYPE
867 380
868     0
869     9
870 $PSTYLEMODE
871 290
872     1
873     9
874 $FINGERPRINTGUID
875     2
876 {92905EEA-4798-D3AF-0738-BBE039DCC74E}
877     9
878 $VERSIONGUID
879     2
880 {FAEB1C32-E019-11D5-929B-00C0DF256EC4}
881     9
882 $EXTNAMES
883 290
884     1
885     9
886 $PSVPSCALE
887 40
888 0.0
889     9
890 $OLESTARTUP
891 290
892     0
893     9
894 $SORTENTS
895 280
896     127
897     9
898 $INDEXCTL
899 280
900     0
901     9
902 $HIDETEXT
903 280
904     1
905     9
906 $XCLIPFRAME
907 280
908     2
909     9
910 $HALOGAP
911 280
912     0

```

(continues on next page)

(continued from previous page)

```
913     9
914 $OBSCOLOR
915     70
916     257
917     9
918 $OBSLTYPE
919 280
920     0
921     9
922 $INTERSECTIONDISPLAY
923 280
924     0
925     9
926 $INTERSECTIONCOLOR
927     70
928     257
929     9
930 $DIMASSOC
931 280
932     2
933     9
934 $PROJECTNAME
935     1
936
937     9
938 $CAMERADISPLAY
939 290
940     0
941     9
942 $LENSLENGTH
943     40
944     50.0
945     9
946 $CAMERAHEIGHT
947     40
948     0.0
949     9
950 $STEPSPERSEC
951     40
952     2.0
953     9
954 $STEPSize
955     40
956     6.0
957     9
958 $3DDWFPREC
959     40
960     2.0
961     9
962 $PSOLWIDTH
963     40
964     5.0
965     9
966 $PSOLHEIGHT
967     40
968     80.0
969     9
```

(continues on next page)

(continued from previous page)

```

970 $LOFTANG1
971 40
972 1.570796326794897
973 9
974 $LOFTANG2
975 40
976 1.570796326794897
977 9
978 $LOFTMAG1
979 40
980 0.0
981 9
982 $LOFTMAG2
983 40
984 0.0
985 9
986 $LOFTPARAM
987 70
988 7
989 9
990 $LOFTNORMALS
991 280
992 1
993 9
994 $LATITUDE
995 40
996 37.795
997 9
998 $LONGITUDE
999 40
1000 -122.394
1001 9
1002 $NORTHDIRECTION
1003 40
1004 0.0
1005 9
1006 $TIMEZONE
1007 70
1008 -8000
1009 9
1010 $LIGHTGLYPHDISPLAY
1011 280
1012 1
1013 9
1014 $TILEMODELIGHTSYNCH
1015 280
1016 1
1017 9
1018 $CMATERIAL
1019 347
1020 3C
1021 9
1022 $$SOLIDHIST
1023 280
1024 1
1025 9
1026 $SHOWHIST

```

(continues on next page)

(continued from previous page)

```

1027 280
1028     1
1029     9
1030 $DWFFRAME
1031 280
1032     2
1033     9
1034 $DGNFRAME
1035 280
1036     2
1037     9
1038 $REALWORLDSCALE
1039 290
1040     1
1041     9
1042 $INTERFERECOLOR
1043    62
1044    256
1045     9
1046 $CSHADOW
1047 280
1048     0
1049     9
1050 $SHADOWPLANELOCATION
1051    40
1052    0.0
1053     0
1054 ENDSEC

```

`gtrace.draw.dxf.classes_template = Multiline-String`

```

1      0
2  SECTION
3      2
4  CLASSES
5      0
6  CLASS
7      1
8  ACDBDICTIONARYWDFLT
9      2
10 AcDbDictionaryWithDefault
11      3
12 ObjectDBX Classes
13     90
14         0
15     91
16         4
17 280
18     0
19 281
20     0
21     0
22 CLASS
23     1
24 VISUALSTYLE
25     2
26 AcDbVisualStyle

```

(continues on next page)

(continued from previous page)

```

27     3
28 ObjectDBX Classes
29     90
30         4095
31     91
32         4
33 280
34     0
35 281
36     0
37     0
38 CLASS
39     1
40 MATERIAL
41     2
42 AcDbMaterial
43     3
44 ObjectDBX Classes
45     90
46         1153
47     91
48         4
49 280
50     0
51 281
52     0
53     0
54 CLASS
55     1
56 SUN
57     2
58 AcDbSun
59     3
60 SCENEOE
61     90
62         1024
63     91
64         4
65 280
66     0
67 281
68     0
69     0
70 ENDSEC

```

`gtrace.draw.dxf.tables_template1 = Multiline-String`

```

1     0
2 SECTION
3     2
4 TABLES

```

`gtrace.draw.dxf.vport_template = Multiline-String`

```

1     0
2 TABLE
3     2

```

(continues on next page)

(continued from previous page)

```
4  VPORT
5    5
6    8
7    330
8    0
9    100
10  AcDbSymbolTable
11    70
12      1
13      0
14  VPORT
15    5
16    29
17    330
18    8
19    100
20  AcDbSymbolTableRecord
21    100
22  AcDbViewportTableRecord
23    2
24  *Active
25    70
26      0
27    10
28    0.0
29    20
30    0.0
31    11
32    1.0
33    21
34    1.0
35    12
36    360.5069142256372
37    22
38    77.37740091485402
39    13
40    0.0
41    23
42    0.0
43    14
44    10.0
45    24
46    10.0
47    15
48    10.0
49    25
50    10.0
51    16
52    0.0
53    26
54    0.0
55    36
56    1.0
57    17
58    0.0
59    27
60    0.0
```

(continues on next page)

(continued from previous page)

61	37
62	0.0
63	40
64	643.00113180099
65	41
66	2.116173120728929
67	42
68	50.0
69	43
70	0.0
71	44
72	0.0
73	50
74	0.0
75	51
76	0.0
77	71
78	0
79	72
80	100
81	73
82	1
83	74
84	3
85	75
86	0
87	76
88	0
89	77
90	0
91	78
92	0
93	281
94	0
95	65
96	1
97	110
98	0.0
99	120
100	0.0
101	130
102	0.0
103	111
104	1.0
105	121
106	0.0
107	131
108	0.0
109	112
110	0.0
111	122
112	1.0
113	132
114	0.0
115	79
116	0
117	146

(continues on next page)

(continued from previous page)

```
118 0.0
119 60
120 3
121 61
122 5
123 292
124 1
125 282
126 1
127 141
128 0.0
129 142
130 0.0
131 63
132 250
133 361
134 3F
135 0
136 ENDTAB
```

`gtrace.draw.dxf.linetype_template = Multiline-String`

```
1 0
2 TABLE
3 2
4 LTYPE
5 5
6 5
7 330
8 0
9 100
10 AcDbSymbolTable
11 70
12 7
13 0
14 LTYPE
15 5
16 14
17 330
18 5
19 100
20 AcDbSymbolTableRecord
21 100
22 AcDbLinetypeTableRecord
23 2
24 ByBlock
25 70
26 0
27 3
28
29 72
30 65
31 73
32 0
33 40
34 0.0
35 0
```

(continues on next page)

(continued from previous page)

```

36  LTYPE
37    5
38  15
39  330
40    5
41  100
42  AcDbSymbolTableRecord
43  100
44  AcDbLinetypeTableRecord
45    2
46  ByLayer
47    70
48      0
49    3
50
51    72
52      65
53    73
54      0
55    40
56    0.0
57      0
58  LTYPE
59    5
60  16
61  330
62    5
63  100
64  AcDbSymbolTableRecord
65  100
66  AcDbLinetypeTableRecord
67    2
68  Continuous
69    70
70      0
71    3
72  Solid line
73    72
74      65
75    73
76      0
77    40
78    0.0
79      0
80  LTYPE
81    5
82  40
83  330
84    5
85  100
86  AcDbSymbolTableRecord
87  100
88  AcDbLinetypeTableRecord
89    2
90  DOT
91    70
92      0

```

(continues on next page)

(continued from previous page)

```

93      3
94 Dot . . . . .
95      72
96      65
97      73
98      2
99      40
100 2.032
101 49
102 1.016
103 74
104      0
105 49
106 -1.016
107 74
108      0
109      0
110 LTYPE
111      5
112 41
113 330
114 5
115 100
116 AcDbSymbolTableRecord
117 100
118 AcDbLinetypeTableRecord
119      2
120 CENTER
121 70
122      0
123      3
124 Center _____
125 72
126      65
127 73
128      4
129 40
130 13.208
131 49
132 8.128
133 74
134      0
135 49
136 -1.524
137 74
138      0
139 49
140 2.032
141 74
142      0
143 49
144 -1.524
145 74
146      0
147      0
148 LTYPE
149      5

```

(continues on next page)

(continued from previous page)

```

150 42
151 330
152 5
153 100
154 AcDbSymbolTableRecord
155 100
156 AcDbLinetypeTableRecord
157 2
158 HIDDEN2
159 70
160 0
161 3
162 Hidden (.5x) _ _ _ _ _
163 72
164 65
165 73
166 2
167 40
168 4.7624999999999999
169 49
170 3.175
171 74
172 0
173 49
174 -1.5875
175 74
176 0
177 0
178 LTYPE
179 5
180 43
181 330
182 5
183 100
184 AcDbSymbolTableRecord
185 100
186 AcDbLinetypeTableRecord
187 2
188 DASHED
189 70
190 0
191 3
192 Dashed _ _ _ _ _
193 72
194 65
195 73
196 2
197 40
198 9.652
199 49
200 8.128
201 74
202 0
203 49
204 -1.524
205 74
206 0

```

(continues on next page)

(continued from previous page)

```

207 0
208 LTYPE
209 5
210 44
211 330
212 5
213 100
214 AcDbSymbolTableRecord
215 100
216 AcDbLinetypeTableRecord
217 2
218 PHANTOM2
219 70
220 0
221 3
222 Phantom (.5x) ____ _ _ _ _ _ _ _ _ _ _
223 72
224 65
225 73
226 6
227 40
228 14.732
229 49
230 8.128
231 74
232 0
233 49
234 -1.524
235 74
236 0
237 49
238 1.016
239 74
240 0
241 49
242 -1.524
243 74
244 0
245 49
246 1.016
247 74
248 0
249 49
250 -1.524
251 74
252 0
253 0
254 LTYPE
255 5
256 45
257 330
258 5
259 100
260 AcDbSymbolTableRecord
261 100
262 AcDbLinetypeTableRecord
263 2

```

(continues on next page)

(continued from previous page)

```

264 DASHDOT
265   70
266     0
267     3
268 Dash dot  _ . _ . _ . _ . _ . _
269   72
270     65
271   73
272     4
273   40
274 11.176
275   49
276  8.128
277   74
278     0
279   49
280 -1.524
281   74
282     0
283   49
284   0.0
285   74
286     0
287   49
288 -1.524
289   74
290     0
291     0
292 ENDTAB

```

`gtrace.draw.dxf.layer_template1 = Multiline-String`

```

1      0
2  TABLE
3      2
4  LAYER
5      5
6      2
7      330
8      0
9      100
10  AcDbSymbolTable
11      70
12      7
13      0
14  LAYER
15      5
16      10
17      330
18      2
19      100
20  AcDbSymbolTableRecord
21      100
22  AcDbLayerTableRecord
23      2
24      0
25      70

```

(continues on next page)

(continued from previous page)

```

26      0
27    62
28      5
29      6
30 Continuous
31    370
32     -3
33    390
34    F
35    347
36    3E
37      0
38 LAYER
39      5
40    46
41    330
42      2
43    100
44 AcDbSymbolTableRecord
45    100
46 AcDbLayerTableRecord
47      2
48 Default
49      70
50      0
51    62
52      255
53      6
54 Continuous
55    370
56     -3
57    390
58    F
59    347
60    3E

```

`gtrace.draw.dxf.layer_template2 = Multiline-String`

```

1      0
2    ENDTAB

```

`gtrace.draw.dxf.style_template = Multiline-String`

```

1      0
2 TABLE
3      2
4 STYLE
5      5
6      3
7    330
8      0
9    100
10 AcDbSymbolTable
11      70
12      33
13      0
14 STYLE

```

(continues on next page)

(continued from previous page)

```

15      5
16     11
17    330
18     3
19    100
20    AcDbSymbolTableRecord
21    100
22    AcDbTextStyleTableRecord
23     2
24    Standard
25     70
26      0
27     40
28    0.0
29     41
30    1.0
31     50
32    0.0
33     71
34      0
35     42
36    2.5
37     3
38    txt
39     4
40
41     0
42    ENDTAB

```

`gtrace.draw.dxf.view_template = Multiline-String`

```

1      0
2     TABLE
3      2
4     VIEW
5      5
6      6
7    330
8      0
9    100
10   AcDbSymbolTable
11    70
12      0
13      0
14   ENDTAB

```

`gtrace.draw.dxf.misctable_template = Multiline-String`

```

1      0
2     TABLE
3      2
4     UCS
5      5
6      7
7    330
8      0
9    100

```

(continues on next page)

(continued from previous page)

```

10  AcDbSymbolTable
11    70
12      0
13      0
14  ENDTAB
15    0
16  TABLE
17    2
18  APPID
19    5
20    9
21   330
22    0
23   100
24  AcDbSymbolTable
25    70
26      1
27      0
28  APPID
29    5
30   12
31   330
32    9
33   100
34  AcDbSymbolTableRecord
35   100
36  AcDbRegAppTableRecord
37    2
38  ACAD
39    70
40      0
41      0
42  ENDTAB
43    0
44  TABLE
45    2
46  DIMSTYLE
47    5
48  A
49   330
50    0
51   100
52  AcDbSymbolTable
53    70
54      1
55   100
56  AcDbDimStyleTable
57    0
58  DIMSTYLE
59   105
60   27
61   330
62  A
63   100
64  AcDbSymbolTableRecord
65   100
66  AcDbDimStyleTableRecord

```

(continues on next page)

(continued from previous page)

```

67      2
68 ISO-25
69      70
70      0
71      41
72      2.5
73      42
74      0.625
75      43
76      3.75
77      44
78      1.25
79      73
80      0
81      74
82      0
83      77
84      1
85      78
86      8
87      140
88      2.5
89      141
90      2.5
91      143
92      0.03937007874016
93      147
94      0.625
95      171
96      3
97      172
98      1
99      178
100     0
101     271
102     2
103     272
104     2
105     274
106     3
107     278
108     44
109     283
110     0
111     284
112     8
113     340
114     11
115     0
116 ENDTAB

```

`gtrace.draw.dxf.blockrecords_template = Multiline-String`

```

1      0
2 TABLE
3      2
4 BLOCK_RECORD

```

(continues on next page)

(continued from previous page)

```

5      5
6      1
7      330
8      0
9      100
10     AcDbSymbolTable
11      70
12      1
13      0
14     BLOCK_RECORD
15      5
16     1F
17      330
18      1
19      100
20     AcDbSymbolTableRecord
21      100
22     AcDbBlockTableRecord
23      2
24     *Model_Space
25      340
26      22
27      70
28      0
29      280
30      1
31      281
32      0
33      0
34     BLOCK_RECORD
35      5
36     1B
37      330
38      1
39      100
40     AcDbSymbolTableRecord
41      100
42     AcDbBlockTableRecord
43      2
44     *Paper_Space
45      340
46     1E
47      70
48      0
49      280
50      1
51      281
52      0
53      0
54     BLOCK_RECORD
55      5
56      23
57      330
58      1
59      100
60     AcDbSymbolTableRecord
61      100

```

(continues on next page)

(continued from previous page)

```

62  AcDbBlockTableRecord
63      2
64  *Paper_Space0
65  340
66  26
67      70
68      0
69  280
70      1
71  281
72      0
73      0
74  ENDTAB

```

`gtrace.draw.dxf.tables_template2 = Multiline-String`

```

1      0
2  ENDSEC

```

`gtrace.draw.dxf.blocks_template = Multiline-String`

```

1      0
2  SECTION
3      2
4  BLOCKS
5      0
6  BLOCK
7      5
8      20
9      330
10     1F
11     100
12  AcDbEntity
13     8
14     0
15     100
16  AcDbBlockBegin
17     2
18  *Model_Space
19     70
20     0
21     10
22     0.0
23     20
24     0.0
25     30
26     0.0
27     3
28  *Model_Space
29     1
30
31     0
32  ENDBLK
33     5
34     21
35     330
36     1F

```

(continues on next page)

(continued from previous page)

```
37 100
38 AcDbEntity
39 8
40 0
41 100
42 AcDbBlockEnd
43 0
44 BLOCK
45 5
46 1C
47 330
48 1B
49 100
50 AcDbEntity
51 67
52 1
53 8
54 0
55 100
56 AcDbBlockBegin
57 2
58 *Paper_Space
59 70
60 0
61 10
62 0.0
63 20
64 0.0
65 30
66 0.0
67 3
68 *Paper_Space
69 1
70 0
71 0
72 ENDBLK
73 5
74 1D
75 330
76 1B
77 100
78 AcDbEntity
79 67
80 1
81 8
82 0
83 100
84 AcDbBlockEnd
85 0
86 BLOCK
87 5
88 24
89 330
90 23
91 100
92 AcDbEntity
93 8
```

(continues on next page)

(continued from previous page)

```

94 0
95 100
96 AcDbBlockBegin
97 2
98 *Paper_Space0
99 70
100 0
101 10
102 0.0
103 20
104 0.0
105 30
106 0.0
107 3
108 *Paper_Space0
109 1
110
111 0
112 ENDBLK
113 5
114 25
115 330
116 23
117 100
118 AcDbEntity
119 8
120 0
121 100
122 AcDbBlockEnd
123 0
124 ENDSEC

```

`gtrace.draw.dxf.entities_template1 = Multiline-String`

```

1 0
2 SECTION
3 2
4 ENTITIES

```

`gtrace.draw.dxf.entities_template0 = Multiline-String`

```

1 0
2 LINE
3 5
4 7C
5 330
6 1F
7 100
8 AcDbEntity
9 8
10 Default
11 100
12 AcDbLine
13 10
14 1.0
15 20
16 0.0

```

(continues on next page)

(continued from previous page)

```
17 30
18 0.0
19 11
20 73.0
21 21
22 41.0
23 31
24 0.0
25 0
26 ENDSEC
```

`gtrace.draw.dxf.objects_template = Multiline-String`

```
1 0
2 SECTION
3 2
4 OBJECTS
5 0
6 DICTIONARY
7 5
8 C
9 330
10 0
11 100
12 AcDbDictionary
13 281
14 1
15 3
16 ACAD_GROUP
17 350
18 D
19 3
20 ACAD_LAYOUT
21 350
22 1A
23 3
24 ACAD_MATERIAL
25 350
26 3B
27 3
28 ACAD_MLEADERSTYLE
29 350
30 6E
31 3
32 ACAD_MLINESYLE
33 350
34 17
35 3
36 ACAD_PLOTSETTINGS
37 350
38 19
39 3
40 ACAD_PLOTSTYLENAME
41 350
42 E
43 3
44 ACAD_TABLESTYLE
```

(continues on next page)

(continued from previous page)

```

45 350
46 6D
47 3
48 ACAD_VISUALSTYLE
49 350
50 2A
51 0
52 SUN
53 5
54 3F
55 330
56 29
57 100
58 AcDbSun
59 90
60 1
61 290
62 0
63 63
64 7
65 421
66 16777215
67 40
68 1.0
69 291
70 1
71 91
72 2455826
73 92
74 54000000
75 292
76 0
77 70
78 2
79 71
80 256
81 280
82 1
83 0
84 DICTIONARY
85 5
86 D
87 102
88 {ACAD_REACTORS
89 330
90 C
91 102
92 }
93 330
94 C
95 100
96 AcDbDictionary
97 281
98 1
99 0
100 DICTIONARY
101 5

```

(continues on next page)

(continued from previous page)

```
102 1A
103 102
104 {ACAD_REACTORS
105 330
106 C
107 102
108 }
109 330
110 C
111 100
112 AcDbDictionary
113 281
114     1
115     3
116 Layout1
117 350
118 1E
119     3
120 Layout2
121 350
122 26
123     3
124 Model
125 350
126 22
127     0
128 DICTIONARY
129     5
130 3B
131 102
132 {ACAD_REACTORS
133 330
134 C
135 102
136 }
137 330
138 C
139 100
140 AcDbDictionary
141 281
142     1
143     3
144 ByBlock
145 350
146 3D
147     3
148 ByLayer
149 350
150 3C
151     3
152 Global
153 350
154 3E
155     0
156 DICTIONARY
157     5
158 6E
```

(continues on next page)

(continued from previous page)

```

159 102
160 {ACAD_REACTORS
161 330
162 C
163 102
164 }
165 330
166 C
167 100
168 AcDbDictionary
169 281
170     1
171     0
172 DICTIONARY
173     5
174 17
175 102
176 {ACAD_REACTORS
177 330
178 C
179 102
180 }
181 330
182 C
183 100
184 AcDbDictionary
185 281
186     1
187     3
188 Standard
189 350
190 18
191     0
192 DICTIONARY
193     5
194 19
195 102
196 {ACAD_REACTORS
197 330
198 C
199 102
200 }
201 330
202 C
203 100
204 AcDbDictionary
205 281
206     1
207     0
208 ACDBDICTIONARYWDFLT
209     5
210 E
211 102
212 {ACAD_REACTORS
213 330
214 C
215 102

```

(continues on next page)

(continued from previous page)

```

216 }
217 330
218 C
219 100
220 AcDbDictionary
221 281
222     1
223     3
224 Normal
225 350
226 F
227 100
228 AcDbDictionaryWithDefault
229 340
230 F
231     0
232 DICTIONARY
233     5
234 6D
235 102
236 {ACAD_REACTORS
237 330
238 C
239 102
240 }
241 330
242 C
243 100
244 AcDbDictionary
245 281
246     1
247     0
248 DICTIONARY
249     5
250 2A
251 102
252 {ACAD_REACTORS
253 330
254 C
255 102
256 }
257 330
258 C
259 100
260 AcDbDictionary
261 281
262     1
263     3
264 2dWireframe
265 350
266 2F
267     3
268 3D Hidden
269 350
270 31
271     3
272 3dWireframe

```

(continues on next page)

(continued from previous page)

```

273 350
274 30
275 3
276 Basic
277 350
278 32
279 3
280 Brighten
281 350
282 36
283 3
284 ColorChange
285 350
286 3A
287 3
288 Conceptual
289 350
290 34
291 3
292 Dim
293 350
294 35
295 3
296 Facepattern
297 350
298 39
299 3
300 Flat
301 350
302 2B
303 3
304 FlatWithEdges
305 350
306 2C
307 3
308 Gouraud
309 350
310 2D
311 3
312 GouraudWithEdges
313 350
314 2E
315 3
316 Linepattern
317 350
318 38
319 3
320 Realistic
321 350
322 33
323 3
324 Thicken
325 350
326 37
327 0
328 LAYOUT
329 5

```

(continues on next page)

(continued from previous page)

```
330 1E
331 102
332 {ACAD_REACTORS
333 330
334 1A
335 102
336 }
337 330
338 1A
339 100
340 AcDbPlotSettings
341 1
342
343 2
344 none_device
345 4
346
347 6
348
349 40
350 0.0
351 41
352 0.0
353 42
354 0.0
355 43
356 0.0
357 44
358 0.0
359 45
360 0.0
361 46
362 0.0
363 47
364 0.0
365 48
366 0.0
367 49
368 0.0
369 140
370 0.0
371 141
372 0.0
373 142
374 1.0
375 143
376 1.0
377 70
378 688
379 72
380 0
381 73
382 0
383 74
384 5
385 7
386
```

(continues on next page)

(continued from previous page)

```

387 75
388 16
389 76
390 0
391 77
392 2
393 78
394 300
395 147
396 1.0
397 148
398 0.0
399 149
400 0.0
401 100
402 AcDbLayout
403 1
404 Layout1
405 70
406 1
407 71
408 1
409 10
410 0.0
411 20
412 0.0
413 11
414 420.0
415 21
416 297.0
417 12
418 0.0
419 22
420 0.0
421 32
422 0.0
423 14
424 1.0000000000000000E+20
425 24
426 1.0000000000000000E+20
427 34
428 1.0000000000000000E+20
429 15
430 -1.0000000000000000E+20
431 25
432 -1.0000000000000000E+20
433 35
434 -1.0000000000000000E+20
435 146
436 0.0
437 13
438 0.0
439 23
440 0.0
441 33
442 0.0
443 16

```

(continues on next page)

(continued from previous page)

```
444 1.0
445 26
446 0.0
447 36
448 0.0
449 17
450 0.0
451 27
452 1.0
453 37
454 0.0
455 76
456 0
457 330
458 1B
459 0
460 LAYOUT
461 5
462 26
463 102
464 {ACAD_REACTORS
465 330
466 1A
467 102
468 }
469 330
470 1A
471 100
472 AcDbPlotSettings
473 1
474
475 2
476 none_device
477 4
478
479 6
480
481 40
482 0.0
483 41
484 0.0
485 42
486 0.0
487 43
488 0.0
489 44
490 0.0
491 45
492 0.0
493 46
494 0.0
495 47
496 0.0
497 48
498 0.0
499 49
500 0.0
```

(continues on next page)

(continued from previous page)

```

501 140
502 0.0
503 141
504 0.0
505 142
506 1.0
507 143
508 1.0
509 70
510 688
511 72
512 0
513 73
514 0
515 74
516 5
517 7
518
519 75
520 16
521 76
522 0
523 77
524 2
525 78
526 300
527 147
528 1.0
529 148
530 0.0
531 149
532 0.0
533 100
534 AcDbLayout
535 1
536 Layout2
537 70
538 1
539 71
540 2
541 10
542 0.0
543 20
544 0.0
545 11
546 0.0
547 21
548 0.0
549 12
550 0.0
551 22
552 0.0
553 32
554 0.0
555 14
556 0.0
557 24

```

(continues on next page)

(continued from previous page)

```
558 0.0
559 34
560 0.0
561 15
562 0.0
563 25
564 0.0
565 35
566 0.0
567 146
568 0.0
569 13
570 0.0
571 23
572 0.0
573 33
574 0.0
575 16
576 1.0
577 26
578 0.0
579 36
580 0.0
581 17
582 0.0
583 27
584 1.0
585 37
586 0.0
587 76
588 0
589 330
590 23
591 0
592 LAYOUT
593 5
594 22
595 102
596 {ACAD_REACTORS
597 330
598 1A
599 102
600 }
601 330
602 1A
603 100
604 AcDbPlotSettings
605 1
606
607 2
608 none_device
609 4
610
611 6
612
613 40
614 0.0
```

(continues on next page)

(continued from previous page)

```
615 41
616 0.0
617 42
618 0.0
619 43
620 0.0
621 44
622 0.0
623 45
624 0.0
625 46
626 0.0
627 47
628 0.0
629 48
630 0.0
631 49
632 0.0
633 140
634 0.0
635 141
636 0.0
637 142
638 1.0
639 143
640 1.0
641 70
642 1712
643 72
644 0
645 73
646 0
647 74
648 0
649 7
650
651 75
652 0
653 76
654 0
655 77
656 2
657 78
658 300
659 147
660 1.0
661 148
662 0.0
663 149
664 0.0
665 100
666 AcDbLayout
667 1
668 Model
669 70
670 1
671 71
```

(continues on next page)

(continued from previous page)

```
672      0
673     10
674    0.0
675     20
676    0.0
677     11
678   420.0
679     21
680  297.0
681     12
682    0.0
683     22
684    0.0
685     32
686    0.0
687     14
688    1.0
689     24
690    0.0
691     34
692    0.0
693     15
694   73.0
695     25
696   41.0
697     35
698    0.0
699   146
700    0.0
701     13
702    0.0
703     23
704    0.0
705     33
706    0.0
707     16
708    1.0
709     26
710    0.0
711     36
712    0.0
713     17
714    0.0
715     27
716    1.0
717     37
718    0.0
719     76
720      0
721   330
722   1F
723   331
724   29
725     0
726 MATERIAL
727     5
728   3D
```

(continues on next page)

(continued from previous page)

```

729 102
730 {ACAD_REACTORS
731 330
732 3B
733 102
734 }
735 330
736 3B
737 100
738 AcDbMaterial
739 1
740 ByBlock
741 72
742 1
743 94
744 127
745 0
746 MATERIAL
747 5
748 3C
749 102
750 {ACAD_REACTORS
751 330
752 3B
753 102
754 }
755 330
756 3B
757 100
758 AcDbMaterial
759 1
760 ByLayer
761 72
762 1
763 94
764 127
765 0
766 MATERIAL
767 5
768 3E
769 102
770 {ACAD_REACTORS
771 330
772 3B
773 102
774 }
775 330
776 3B
777 100
778 AcDbMaterial
779 1
780 Global
781 72
782 1
783 94
784 127
785 0

```

(continues on next page)

(continued from previous page)

```
786 MLINESTYLE
787     5
788     18
789     102
790     {ACAD_REACTORS
791     330
792     17
793     102
794     }
795     330
796     17
797     100
798     AcDbMlineStyle
799         2
800     Standard
801         70
802             0
803             3
804
805             62
806             256
807             51
808             90.0
809             52
810             90.0
811             71
812                 2
813             49
814             0.5
815             62
816             256
817             6
818     BYLAYER
819         49
820         -0.5
821         62
822             256
823             6
824     BYLAYER
825         0
826     ACDBPLACEHOLDER
827         5
828     F
829     102
830     {ACAD_REACTORS
831     330
832     E
833     102
834     }
835     330
836     E
837         0
838     VISUALSTYLE
839         5
840     2F
841     102
842     {ACAD_REACTORS
```

(continues on next page)

(continued from previous page)

```

843 330
844 2A
845 102
846 }
847 330
848 2A
849 100
850 AcDbVisualStyle
851 2
852 2dWireframe
853 70
854 4
855 177
856 2
857 291
858 0
859 71
860 0
861 176
862 1
863 72
864 2
865 176
866 1
867 73
868 0
869 176
870 1
871 90
872 0
873 176
874 1
875 40
876 -0.6
877 176
878 1
879 41
880 -30.0
881 176
882 1
883 62
884 5
885 63
886 7
887 421
888 16777215
889 176
890 1
891 74
892 1
893 176
894 1
895 91
896 4
897 176
898 1
899 64

```

(continues on next page)

(continued from previous page)

900	7
901	176
902	1
903	65
904	257
905	176
906	1
907	75
908	1
909	176
910	1
911	175
912	1
913	176
914	1
915	42
916	1.0
917	176
918	1
919	92
920	0
921	176
922	1
923	66
924	257
925	176
926	1
927	43
928	1.0
929	176
930	1
931	76
932	1
933	176
934	1
935	77
936	6
937	176
938	1
939	78
940	2
941	176
942	1
943	67
944	7
945	176
946	1
947	79
948	5
949	176
950	1
951	170
952	0
953	176
954	1
955	171
956	0

(continues on next page)

(continued from previous page)

```

957 176
958     1
959 290
960     0
961 176
962     1
963    93
964         1
965 176
966     1
967    44
968    0.0
969 176
970     1
971 173
972     0
973 176
974     1
975     0
976 VISUALSTYLE
977     5
978    31
979   102
980 {ACAD_REACTORS
981 330
982 2A
983 102
984 }
985 330
986 2A
987 100
988 AcDbVisualStyle
989     2
990 3D Hidden
991    70
992     6
993 177
994     2
995 291
996     0
997    71
998     1
999 176
1000     1
1001    72
1002     2
1003 176
1004     1
1005    73
1006     2
1007 176
1008     1
1009    90
1010         0
1011 176
1012     1
1013    40

```

(continues on next page)

(continued from previous page)

1014	-0.6		
1015	176		
1016		1	
1017	41		
1018	-30.0		
1019	176		
1020		1	
1021	62		
1022		5	
1023	63		
1024		7	
1025	421		
1026	16777215		
1027	176		
1028		1	
1029	74		
1030		2	
1031	176		
1032		1	
1033	91		
1034		2	
1035	176		
1036		1	
1037	64		
1038		7	
1039	176		
1040		1	
1041	65		
1042	257		
1043	176		
1044		1	
1045	75		
1046		2	
1047	176		
1048		1	
1049	175		
1050		1	
1051	176		
1052		1	
1053	42		
1054	40.0		
1055	176		
1056		1	
1057	92		
1058		0	
1059	176		
1060		1	
1061	66		
1062	257		
1063	176		
1064		1	
1065	43		
1066	1.0		
1067	176		
1068		1	
1069	76		
1070		1	

(continues on next page)

(continued from previous page)

```

1071 176
1072     1
1073    77
1074     6
1075 176
1076     1
1077    78
1078     2
1079 176
1080     1
1081    67
1082     7
1083 176
1084     1
1085    79
1086     3
1087 176
1088     1
1089 170
1090     0
1091 176
1092     1
1093 171
1094     0
1095 176
1096     1
1097 290
1098     0
1099 176
1100     1
1101    93
1102         1
1103 176
1104     1
1105    44
1106    0.0
1107 176
1108     1
1109 173
1110     0
1111 176
1112     1
1113     0
1114 VISUALSTYLE
1115     5
1116    30
1117   102
1118 {ACAD_REACTORS
1119 330
1120 2A
1121 102
1122 }
1123 330
1124 2A
1125 100
1126 AcDbVisualStyle
1127     2

```

(continues on next page)

(continued from previous page)

```

1128 3dWireframe
1129   70
1130     5
1131  177
1132     2
1133  291
1134     0
1135   71
1136     0
1137  176
1138     1
1139   72
1140     2
1141  176
1142     1
1143   73
1144     0
1145  176
1146     1
1147   90
1148       0
1149  176
1150     1
1151   40
1152  -0.6
1153  176
1154     1
1155   41
1156 -30.0
1157  176
1158     1
1159   62
1160     5
1161   63
1162     7
1163  421
1164  16777215
1165  176
1166     1
1167   74
1168     1
1169  176
1170     1
1171   91
1172       4
1173  176
1174     1
1175   64
1176     7
1177  176
1178     1
1179   65
1180  257
1181  176
1182     1
1183   75
1184     1

```

(continues on next page)

(continued from previous page)

1185	176		
1186		1	
1187	175		
1188		1	
1189	176		
1190		1	
1191	42		
1192	1.0		
1193	176		
1194		1	
1195	92		
1196			0
1197	176		
1198		1	
1199	66		
1200		257	
1201	176		
1202		1	
1203	43		
1204	1.0		
1205	176		
1206		1	
1207	76		
1208		1	
1209	176		
1210		1	
1211	77		
1212		6	
1213	176		
1214		1	
1215	78		
1216		2	
1217	176		
1218		1	
1219	67		
1220		7	
1221	176		
1222		1	
1223	79		
1224		5	
1225	176		
1226		1	
1227	170		
1228		0	
1229	176		
1230		1	
1231	171		
1232		0	
1233	176		
1234		1	
1235	290		
1236		0	
1237	176		
1238		1	
1239	93		
1240			1
1241	176		

(continues on next page)

(continued from previous page)

```
1242      1
1243    44
1244    0.0
1245    176
1246      1
1247    173
1248      0
1249    176
1250      1
1251      0
1252    VISUALSTYLE
1253      5
1254    32
1255    102
1256    {ACAD_REACTORS
1257    330
1258    2A
1259    102
1260    }
1261    330
1262    2A
1263    100
1264    AcDbVisualStyle
1265      2
1266    Basic
1267      70
1268      7
1269    177
1270      2
1271    291
1272      1
1273      71
1274      1
1275    176
1276      1
1277      72
1278      0
1279    176
1280      1
1281      73
1282      1
1283    176
1284      1
1285      90
1286      0
1287    176
1288      1
1289      40
1290    -0.6
1291    176
1292      1
1293      41
1294    -30.0
1295    176
1296      1
1297      62
1298      5
```

(continues on next page)

(continued from previous page)

1299	63	
1300	7	
1301	421	
1302	16777215	
1303	176	
1304	1	
1305	74	
1306	0	
1307	176	
1308	1	
1309	91	
1310		4
1311	176	
1312	1	
1313	64	
1314	7	
1315	176	
1316	1	
1317	65	
1318	257	
1319	176	
1320	1	
1321	75	
1322	1	
1323	176	
1324	1	
1325	175	
1326	1	
1327	176	
1328	1	
1329	42	
1330	1.0	
1331	176	
1332	1	
1333	92	
1334		8
1335	176	
1336	1	
1337	66	
1338	7	
1339	176	
1340	1	
1341	43	
1342	1.0	
1343	176	
1344	1	
1345	76	
1346	1	
1347	176	
1348	1	
1349	77	
1350	6	
1351	176	
1352	1	
1353	78	
1354	2	
1355	176	

(continues on next page)

(continued from previous page)

```

1356      1
1357    67
1358      7
1359   176
1360      1
1361    79
1362      5
1363   176
1364      1
1365   170
1366      0
1367   176
1368      1
1369   171
1370      0
1371   176
1372      1
1373   290
1374      0
1375   176
1376      1
1377    93
1378          1
1379   176
1380      1
1381    44
1382    0.0
1383   176
1384      1
1385   173
1386      0
1387   176
1388      1
1389      0
1390  VISUALSTYLE
1391      5
1392    36
1393   102
1394  {ACAD_REACTORS
1395   330
1396   2A
1397   102
1398   }
1399   330
1400   2A
1401   100
1402  AcDbVisualStyle
1403      2
1404  Brighten
1405    70
1406      12
1407   177
1408      2
1409   291
1410      1
1411    71
1412      2

```

(continues on next page)

(continued from previous page)

1413	176		
1414		1	
1415	72		
1416		2	
1417	176		
1418		1	
1419	73		
1420		0	
1421	176		
1422		1	
1423	90		
1424			0
1425	176		
1426		1	
1427	40		
1428	-0.6		
1429	176		
1430		1	
1431	41		
1432	-30.0		
1433	176		
1434		1	
1435	62		
1436		5	
1437	63		
1438		7	
1439	421		
1440	16777215		
1441	176		
1442		1	
1443	74		
1444		1	
1445	176		
1446		1	
1447	91		
1448			4
1449	176		
1450		1	
1451	64		
1452		7	
1453	176		
1454		1	
1455	65		
1456		257	
1457	176		
1458		1	
1459	75		
1460		1	
1461	176		
1462		1	
1463	175		
1464		1	
1465	176		
1466		1	
1467	42		
1468	1.0		
1469	176		

(continues on next page)

(continued from previous page)

1470	1	
1471	92	
1472		8
1473	176	
1474	1	
1475	66	
1476	7	
1477	176	
1478	1	
1479	43	
1480	1.0	
1481	176	
1482	1	
1483	76	
1484	1	
1485	176	
1486	1	
1487	77	
1488	6	
1489	176	
1490	1	
1491	78	
1492	2	
1493	176	
1494	1	
1495	67	
1496	7	
1497	176	
1498	1	
1499	79	
1500	5	
1501	176	
1502	1	
1503	170	
1504	0	
1505	176	
1506	1	
1507	171	
1508	0	
1509	176	
1510	1	
1511	290	
1512	0	
1513	176	
1514	1	
1515	93	
1516		1
1517	176	
1518	1	
1519	44	
1520	50.0	
1521	176	
1522	1	
1523	173	
1524	0	
1525	176	
1526	1	

(continues on next page)

(continued from previous page)

```

1527     0
1528 VISUALSTYLE
1529     5
1530 3A
1531 102
1532 {ACAD_REACTORS
1533 330
1534 2A
1535 102
1536 }
1537 330
1538 2A
1539 100
1540 AcDbVisualStyle
1541     2
1542 ColorChange
1543     70
1544         16
1545 177
1546     2
1547 291
1548     1
1549     71
1550     2
1551 176
1552     1
1553     72
1554     2
1555 176
1556     1
1557     73
1558     3
1559 176
1560     1
1561     90
1562         0
1563 176
1564     1
1565     40
1566 -0.6
1567 176
1568     1
1569     41
1570 -30.0
1571 176
1572     1
1573     62
1574     5
1575     63
1576     8
1577 421
1578 8421504
1579 176
1580     1
1581     74
1582     1
1583 176

```

(continues on next page)

(continued from previous page)

1584	1
1585	91
1586	4
1587	176
1588	1
1589	64
1590	7
1591	176
1592	1
1593	65
1594	257
1595	176
1596	1
1597	75
1598	1
1599	176
1600	1
1601	175
1602	1
1603	176
1604	1
1605	42
1606	1.0
1607	176
1608	1
1609	92
1610	8
1611	176
1612	1
1613	66
1614	8
1615	424
1616	8421504
1617	176
1618	1
1619	43
1620	1.0
1621	176
1622	1
1623	76
1624	1
1625	176
1626	1
1627	77
1628	6
1629	176
1630	1
1631	78
1632	2
1633	176
1634	1
1635	67
1636	7
1637	176
1638	1
1639	79
1640	5

(continues on next page)

(continued from previous page)

```

1641 176
1642     1
1643 170
1644     0
1645 176
1646     1
1647 171
1648     0
1649 176
1650     1
1651 290
1652     0
1653 176
1654     1
1655    93
1656         1
1657 176
1658     1
1659    44
1660    0.0
1661 176
1662     1
1663 173
1664     0
1665 176
1666     1
1667     0
1668 VISUALSTYLE
1669     5
1670    34
1671   102
1672 {ACAD_REACTORS
1673 330
1674 2A
1675 102
1676 }
1677 330
1678 2A
1679 100
1680 AcDbVisualStyle
1681     2
1682 Conceptual
1683    70
1684     9
1685 177
1686     2
1687 291
1688     0
1689    71
1690     3
1691 176
1692     1
1693    72
1694     2
1695 176
1696     1
1697    73

```

(continues on next page)

(continued from previous page)

1698	0
1699	176
1700	1
1701	90
1702	0
1703	176
1704	1
1705	40
1706	-0.6
1707	176
1708	1
1709	41
1710	-30.0
1711	176
1712	1
1713	62
1714	5
1715	63
1716	7
1717	421
1718	16777215
1719	176
1720	1
1721	74
1722	2
1723	176
1724	1
1725	91
1726	2
1727	176
1728	1
1729	64
1730	7
1731	176
1732	1
1733	65
1734	257
1735	176
1736	1
1737	75
1738	1
1739	176
1740	1
1741	175
1742	1
1743	176
1744	1
1745	42
1746	40.0
1747	176
1748	1
1749	92
1750	8
1751	176
1752	1
1753	66
1754	7

(continues on next page)

(continued from previous page)

1755	176	
1756	1	
1757	43	
1758	1.0	
1759	176	
1760	1	
1761	76	
1762	1	
1763	176	
1764	1	
1765	77	
1766	6	
1767	176	
1768	1	
1769	78	
1770	2	
1771	176	
1772	1	
1773	67	
1774	7	
1775	176	
1776	1	
1777	79	
1778	3	
1779	176	
1780	1	
1781	170	
1782	0	
1783	176	
1784	1	
1785	171	
1786	0	
1787	176	
1788	1	
1789	290	
1790	0	
1791	176	
1792	1	
1793	93	
1794	1	
1795	176	
1796	1	
1797	44	
1798	0.0	
1799	176	
1800	1	
1801	173	
1802	0	
1803	176	
1804	1	
1805	0	
1806	VISUALSTYLE	
1807	5	
1808	35	
1809	102	
1810	{ACAD_REACTORS	
1811	330	

(continues on next page)

(continued from previous page)

```

1812 2A
1813 102
1814 }
1815 330
1816 2A
1817 100
1818 AcDbVisualStyle
1819 2
1820 Dim
1821 70
1822 11
1823 177
1824 2
1825 291
1826 1
1827 71
1828 2
1829 176
1830 1
1831 72
1832 2
1833 176
1834 1
1835 73
1836 0
1837 176
1838 1
1839 90
1840 0
1841 176
1842 1
1843 40
1844 -0.6
1845 176
1846 1
1847 41
1848 -30.0
1849 176
1850 1
1851 62
1852 5
1853 63
1854 7
1855 421
1856 16777215
1857 176
1858 1
1859 74
1860 1
1861 176
1862 1
1863 91
1864 4
1865 176
1866 1
1867 64
1868 7

```

(continues on next page)

(continued from previous page)

1869	176	
1870		1
1871	65	
1872	257	
1873	176	
1874		1
1875	75	
1876		1
1877	176	
1878		1
1879	175	
1880		1
1881	176	
1882		1
1883	42	
1884	1.0	
1885	176	
1886		1
1887	92	
1888		8
1889	176	
1890		1
1891	66	
1892		7
1893	176	
1894		1
1895	43	
1896	1.0	
1897	176	
1898		1
1899	76	
1900		1
1901	176	
1902		1
1903	77	
1904		6
1905	176	
1906		1
1907	78	
1908		2
1909	176	
1910		1
1911	67	
1912		7
1913	176	
1914		1
1915	79	
1916		5
1917	176	
1918		1
1919	170	
1920		0
1921	176	
1922		1
1923	171	
1924		0
1925	176	

(continues on next page)

(continued from previous page)

```
1926      1
1927    290
1928      0
1929    176
1930      1
1931     93
1932        1
1933    176
1934      1
1935     44
1936   -50.0
1937    176
1938      1
1939    173
1940      0
1941    176
1942      1
1943      0
1944  VISUALSTYLE
1945      5
1946     39
1947    102
1948  {ACAD_REACTORS
1949    330
1950    2A
1951    102
1952    }
1953    330
1954    2A
1955    100
1956  AcDbVisualStyle
1957      2
1958  Facepattern
1959     70
1960      15
1961    177
1962      2
1963    291
1964      1
1965     71
1966      2
1967    176
1968      1
1969     72
1970      2
1971    176
1972      1
1973     73
1974      0
1975    176
1976      1
1977     90
1978        0
1979    176
1980      1
1981     40
1982   -0.6
```

(continues on next page)

(continued from previous page)

1983	176	
1984		1
1985	41	
1986	-30.0	
1987	176	
1988		1
1989	62	
1990		5
1991	63	
1992		7
1993	421	
1994	16777215	
1995	176	
1996		1
1997	74	
1998		1
1999	176	
2000		1
2001	91	
2002		4
2003	176	
2004		1
2005	64	
2006		7
2007	176	
2008		1
2009	65	
2010	257	
2011	176	
2012		1
2013	75	
2014		1
2015	176	
2016		1
2017	175	
2018		1
2019	176	
2020		1
2021	42	
2022	1.0	
2023	176	
2024		1
2025	92	
2026		8
2027	176	
2028		1
2029	66	
2030		7
2031	176	
2032		1
2033	43	
2034	1.0	
2035	176	
2036		1
2037	76	
2038		1
2039	176	

(continues on next page)

(continued from previous page)

```
2040      1
2041      77
2042      6
2043     176
2044      1
2045      78
2046      2
2047     176
2048      1
2049      67
2050      7
2051     176
2052      1
2053      79
2054      5
2055     176
2056      1
2057     170
2058      0
2059     176
2060      1
2061     171
2062      0
2063     176
2064      1
2065     290
2066      0
2067     176
2068      1
2069      93
2070          1
2071     176
2072      1
2073      44
2074     0.0
2075     176
2076      1
2077     173
2078      0
2079     176
2080      1
2081      0
2082     VISUALSTYLE
2083      5
2084     2B
2085     102
2086     {ACAD_REACTORS
2087     330
2088     2A
2089     102
2090     }
2091     330
2092     2A
2093     100
2094     AcDbVisualStyle
2095      2
2096     Flat
```

(continues on next page)

(continued from previous page)

2097	70
2098	0
2099	177
2100	2
2101	291
2102	1
2103	71
2104	2
2105	176
2106	1
2107	72
2108	1
2109	176
2110	1
2111	73
2112	1
2113	176
2114	1
2115	90
2116	2
2117	176
2118	1
2119	40
2120	-0.6
2121	176
2122	1
2123	41
2124	30.0
2125	176
2126	1
2127	62
2128	5
2129	63
2130	7
2131	421
2132	16777215
2133	176
2134	1
2135	74
2136	0
2137	176
2138	1
2139	91
2140	4
2141	176
2142	1
2143	64
2144	7
2145	176
2146	1
2147	65
2148	257
2149	176
2150	1
2151	75
2152	1
2153	176

(continues on next page)

(continued from previous page)

2154	1	
2155	175	
2156	1	
2157	176	
2158	1	
2159	42	
2160	1.0	
2161	176	
2162	1	
2163	92	
2164		8
2165	176	
2166	1	
2167	66	
2168	7	
2169	176	
2170	1	
2171	43	
2172	1.0	
2173	176	
2174	1	
2175	76	
2176	1	
2177	176	
2178	1	
2179	77	
2180	6	
2181	176	
2182	1	
2183	78	
2184	2	
2185	176	
2186	1	
2187	67	
2188	7	
2189	176	
2190	1	
2191	79	
2192	5	
2193	176	
2194	1	
2195	170	
2196	0	
2197	176	
2198	1	
2199	171	
2200	0	
2201	176	
2202	1	
2203	290	
2204	0	
2205	176	
2206	1	
2207	93	
2208		13
2209	176	
2210	1	

(continues on next page)

(continued from previous page)

```

2211 44
2212 0.0
2213 176
2214 1
2215 173
2216 0
2217 176
2218 1
2219 0
2220 VISUALSTYLE
2221 5
2222 2C
2223 102
2224 {ACAD_REACTORS
2225 330
2226 2A
2227 102
2228 }
2229 330
2230 2A
2231 100
2232 AcDbVisualStyle
2233 2
2234 FlatWithEdges
2235 70
2236 1
2237 177
2238 2
2239 291
2240 1
2241 71
2242 2
2243 176
2244 1
2245 72
2246 1
2247 176
2248 1
2249 73
2250 1
2251 176
2252 1
2253 90
2254 2
2255 176
2256 1
2257 40
2258 -0.6
2259 176
2260 1
2261 41
2262 30.0
2263 176
2264 1
2265 62
2266 5
2267 63

```

(continues on next page)

(continued from previous page)

2268	7
2269	421
2270	16777215
2271	176
2272	1
2273	74
2274	1
2275	176
2276	1
2277	91
2278	4
2279	176
2280	1
2281	64
2282	7
2283	176
2284	1
2285	65
2286	257
2287	176
2288	1
2289	75
2290	1
2291	176
2292	1
2293	175
2294	1
2295	176
2296	1
2297	42
2298	1.0
2299	176
2300	1
2301	92
2302	0
2303	176
2304	1
2305	66
2306	257
2307	176
2308	1
2309	43
2310	1.0
2311	176
2312	1
2313	76
2314	1
2315	176
2316	1
2317	77
2318	6
2319	176
2320	1
2321	78
2322	2
2323	176
2324	1

(continues on next page)

(continued from previous page)

```

2325 67
2326 7
2327 176
2328 1
2329 79
2330 5
2331 176
2332 1
2333 170
2334 0
2335 176
2336 1
2337 171
2338 0
2339 176
2340 1
2341 290
2342 0
2343 176
2344 1
2345 93
2346 13
2347 176
2348 1
2349 44
2350 0.0
2351 176
2352 1
2353 173
2354 0
2355 176
2356 1
2357 0
2358 VISUALSTYLE
2359 5
2360 2D
2361 102
2362 {ACAD_REACTORS
2363 330
2364 2A
2365 102
2366 }
2367 330
2368 2A
2369 100
2370 AcDbVisualStyle
2371 2
2372 Gouraud
2373 70
2374 2
2375 177
2376 2
2377 291
2378 1
2379 71
2380 2
2381 176

```

(continues on next page)

(continued from previous page)

2382	1
2383	72
2384	2
2385	176
2386	1
2387	73
2388	1
2389	176
2390	1
2391	90
2392	2
2393	176
2394	1
2395	40
2396	-0.6
2397	176
2398	1
2399	41
2400	30.0
2401	176
2402	1
2403	62
2404	5
2405	63
2406	7
2407	421
2408	16777215
2409	176
2410	1
2411	74
2412	0
2413	176
2414	1
2415	91
2416	4
2417	176
2418	1
2419	64
2420	7
2421	176
2422	1
2423	65
2424	257
2425	176
2426	1
2427	75
2428	1
2429	176
2430	1
2431	175
2432	1
2433	176
2434	1
2435	42
2436	1.0
2437	176
2438	1

(continues on next page)

(continued from previous page)

2439	92	
2440		0
2441	176	
2442		1
2443	66	
2444		7
2445	176	
2446		1
2447	43	
2448	1.0	
2449	176	
2450		1
2451	76	
2452		1
2453	176	
2454		1
2455	77	
2456		6
2457	176	
2458		1
2459	78	
2460		2
2461	176	
2462		1
2463	67	
2464		7
2465	176	
2466		1
2467	79	
2468		5
2469	176	
2470		1
2471	170	
2472		0
2473	176	
2474		1
2475	171	
2476		0
2477	176	
2478		1
2479	290	
2480		0
2481	176	
2482		1
2483	93	
2484		13
2485	176	
2486		1
2487	44	
2488	0.0	
2489	176	
2490		1
2491	173	
2492		0
2493	176	
2494		1
2495	0	

(continues on next page)

(continued from previous page)

```

2496 VISUALSTYLE
2497     5
2498     2E
2499     102
2500     {ACAD_REACTORS
2501     330
2502     2A
2503     102
2504     }
2505     330
2506     2A
2507     100
2508     AcDbVisualStyle
2509     2
2510     GouraudWithEdges
2511     70
2512         3
2513     177
2514         2
2515     291
2516         1
2517         71
2518         2
2519     176
2520         1
2521         72
2522         2
2523     176
2524         1
2525         73
2526         1
2527     176
2528         1
2529         90
2530             2
2531     176
2532         1
2533         40
2534     -0.6
2535     176
2536         1
2537         41
2538     30.0
2539     176
2540         1
2541         62
2542         5
2543         63
2544         7
2545     421
2546     16777215
2547     176
2548         1
2549         74
2550         1
2551     176
2552         1

```

(continues on next page)

(continued from previous page)

2553	91	
2554		4
2555	176	
2556		1
2557	64	
2558		7
2559	176	
2560		1
2561	65	
2562		257
2563	176	
2564		1
2565	75	
2566		1
2567	176	
2568		1
2569	175	
2570		1
2571	176	
2572		1
2573	42	
2574	1.0	
2575	176	
2576		1
2577	92	
2578		0
2579	176	
2580		1
2581	66	
2582		257
2583	176	
2584		1
2585	43	
2586	1.0	
2587	176	
2588		1
2589	76	
2590		1
2591	176	
2592		1
2593	77	
2594		6
2595	176	
2596		1
2597	78	
2598		2
2599	176	
2600		1
2601	67	
2602		7
2603	176	
2604		1
2605	79	
2606		5
2607	176	
2608		1
2609	170	

(continues on next page)

(continued from previous page)

```
2610      0
2611    176
2612      1
2613    171
2614      0
2615    176
2616      1
2617    290
2618      0
2619    176
2620      1
2621     93
2622      13
2623    176
2624      1
2625     44
2626    0.0
2627    176
2628      1
2629    173
2630      0
2631    176
2632      1
2633      0
2634  VISUALSTYLE
2635      5
2636     38
2637    102
2638  {ACAD_REACTORS
2639    330
2640    2A
2641    102
2642    }
2643    330
2644    2A
2645    100
2646  AcDbVisualStyle
2647      2
2648  Linepattern
2649     70
2650      14
2651    177
2652      2
2653    291
2654      1
2655     71
2656      2
2657    176
2658      1
2659     72
2660      2
2661    176
2662      1
2663     73
2664      0
2665    176
2666      1
```

(continues on next page)

(continued from previous page)

2667	90	
2668		0
2669	176	
2670		1
2671	40	
2672	-0.6	
2673	176	
2674		1
2675	41	
2676	-30.0	
2677	176	
2678		1
2679	62	
2680		5
2681	63	
2682		7
2683	421	
2684	16777215	
2685	176	
2686		1
2687	74	
2688		1
2689	176	
2690		1
2691	91	
2692		4
2693	176	
2694		1
2695	64	
2696		7
2697	176	
2698		1
2699	65	
2700	257	
2701	176	
2702		1
2703	75	
2704		7
2705	176	
2706		1
2707	175	
2708		7
2709	176	
2710		1
2711	42	
2712	1.0	
2713	176	
2714		1
2715	92	
2716		8
2717	176	
2718		1
2719	66	
2720		7
2721	176	
2722		1
2723	43	

(continues on next page)

(continued from previous page)

```
2724 1.0
2725 176
2726     1
2727    76
2728     1
2729 176
2730     1
2731    77
2732     6
2733 176
2734     1
2735    78
2736     2
2737 176
2738     1
2739    67
2740     7
2741 176
2742     1
2743    79
2744     5
2745 176
2746     1
2747 170
2748     0
2749 176
2750     1
2751 171
2752     0
2753 176
2754     1
2755 290
2756     0
2757 176
2758     1
2759    93
2760         1
2761 176
2762     1
2763    44
2764 0.0
2765 176
2766     1
2767 173
2768     0
2769 176
2770     1
2771     0
2772 VISUALSTYLE
2773     5
2774    33
2775   102
2776 {ACAD_REACTORS
2777    330
2778    2A
2779   102
2780 }
```

(continues on next page)

(continued from previous page)

```

2781 330
2782 2A
2783 100
2784 AcDbVisualStyle
2785 2
2786 Realistic
2787 70
2788 8
2789 177
2790 2
2791 291
2792 0
2793 71
2794 2
2795 176
2796 1
2797 72
2798 2
2799 176
2800 1
2801 73
2802 0
2803 176
2804 1
2805 90
2806 0
2807 176
2808 1
2809 40
2810 -0.6
2811 176
2812 1
2813 41
2814 -30.0
2815 176
2816 1
2817 62
2818 5
2819 63
2820 7
2821 421
2822 16777215
2823 176
2824 1
2825 74
2826 1
2827 176
2828 1
2829 91
2830 0
2831 176
2832 1
2833 64
2834 7
2835 176
2836 1
2837 65

```

(continues on next page)

(continued from previous page)

2838	257
2839	176
2840	1
2841	75
2842	1
2843	176
2844	1
2845	175
2846	1
2847	176
2848	1
2849	42
2850	1.0
2851	176
2852	1
2853	92
2854	8
2855	176
2856	1
2857	66
2858	8
2859	424
2860	7895160
2861	176
2862	1
2863	43
2864	1.0
2865	176
2866	1
2867	76
2868	1
2869	176
2870	1
2871	77
2872	6
2873	176
2874	1
2875	78
2876	2
2877	176
2878	1
2879	67
2880	7
2881	176
2882	1
2883	79
2884	5
2885	176
2886	1
2887	170
2888	0
2889	176
2890	1
2891	171
2892	0
2893	176
2894	1

(continues on next page)

(continued from previous page)

```

2895 290
2896     0
2897 176
2898     1
2899    93
2900        13
2901 176
2902     1
2903    44
2904    0.0
2905 176
2906     1
2907 173
2908     0
2909 176
2910     1
2911     0
2912 VISUALSTYLE
2913     5
2914    37
2915   102
2916 {ACAD_REACTORS
2917 330
2918 2A
2919 102
2920 }
2921 330
2922 2A
2923 100
2924 AcDbVisualStyle
2925     2
2926 Thicken
2927    70
2928        13
2929 177
2930     2
2931 291
2932     1
2933    71
2934     2
2935 176
2936     1
2937    72
2938     2
2939 176
2940     1
2941    73
2942     0
2943 176
2944     1
2945    90
2946        0
2947 176
2948     1
2949    40
2950   -0.6
2951 176

```

(continues on next page)

(continued from previous page)

2952	1
2953	41
2954	-30.0
2955	176
2956	1
2957	62
2958	5
2959	63
2960	7
2961	421
2962	16777215
2963	176
2964	1
2965	74
2966	1
2967	176
2968	1
2969	91
2970	4
2971	176
2972	1
2973	64
2974	7
2975	176
2976	1
2977	65
2978	257
2979	176
2980	1
2981	75
2982	1
2983	176
2984	1
2985	175
2986	1
2987	176
2988	1
2989	42
2990	1.0
2991	176
2992	1
2993	92
2994	12
2995	176
2996	1
2997	66
2998	7
2999	176
3000	1
3001	43
3002	1.0
3003	176
3004	1
3005	76
3006	1
3007	176
3008	1

(continues on next page)

(continued from previous page)

```
3009 77
3010 6
3011 176
3012 1
3013 78
3014 2
3015 176
3016 1
3017 67
3018 7
3019 176
3020 1
3021 79
3022 5
3023 176
3024 1
3025 170
3026 0
3027 176
3028 1
3029 171
3030 0
3031 176
3032 1
3033 290
3034 0
3035 176
3036 1
3037 93
3038 1
3039 176
3040 1
3041 44
3042 0.0
3043 176
3044 1
3045 173
3046 0
3047 176
3048 1
3049 0
3050 ENDSEC
```

```
gtrace.draw.dxf.test_func()
```

gtrace.draw.renderer

Renderer module for gtrace.draw

Module Contents

Functions

<code>renderDXF(canvas, filename)</code>	Render a canvas into a DXF file
--	---------------------------------

Attributes

<code>__author__</code>
<code>__copyright__</code>
<code>__credits__</code>
<code>__license__</code>
<code>__version__</code>
<code>__maintainer__</code>
<code>__email__</code>
<code>__status__</code>

```
gtrace.draw.renderer.__author__ = Yoichi Aso
gtrace.draw.renderer.__copyright__ = Copyright 2011-2021, Yoichi Aso
gtrace.draw.renderer.__credits__ = ['Yoichi Aso']
gtrace.draw.renderer.__license__ = BSD
gtrace.draw.renderer.__version__ = 0.2.1
gtrace.draw.renderer.__maintainer__ = Yoichi Aso
gtrace.draw.renderer.__email__ = yoichi.aso@nao.ac.jp
gtrace.draw.renderer.__status__ = Beta
```

exception gtrace.draw.renderer.UnknownShapeError

Bases: BaseException

Common base class for all exceptions

`__init__(self, message)`

gtrace.draw.renderer.renderDXF(*canvas*, *filename*)

Render a canvas into a DXF file

gtrace.draw.tools**Module Contents****Functions**

`drawOptSys(optList, beamList, filename, font-
Size=False)`

`drawAllBeams(d, beamList, sigma=3.0,
drawWidth=True, drawPower=False, drawROC=False,
drawGouy=False, drawOptDist=False, layer=None,
mode='x', fontSize=0.01)`

`drawAllOptics(d, opticsList, drawName=True,
layer=None)`

`transAll(objList, transVect)`

`rotateAll(objList, angle, center)`

Attributes

`__author__`

`__copyright__`

`__credits__`

`__license__`

`__version__`

`__maintainer__`

`__email__`

`__status__`

`gtrace.draw.tools.__author__ = Yoichi Aso`
`gtrace.draw.tools.__copyright__ = Copyright 2011–2021, Yoichi Aso`
`gtrace.draw.tools.__credits__ = ['Yoichi Aso']`
`gtrace.draw.tools.__license__ = BSD`
`gtrace.draw.tools.__version__ = 0.2.1`
`gtrace.draw.tools.__maintainer__ = Yoichi Aso`
`gtrace.draw.tools.__email__ = yoichi.aso@nao.ac.jp`
`gtrace.draw.tools.__status__ = Beta`

```
gtrace.draw.tools.drawOptSys (optList, beamList, filename, fontSize=False)
gtrace.draw.tools.drawAllBeams (d, beamList, sigma=3.0, drawWidth=True, drawPower=False,  
                                drawROC=False, drawGouy=False, drawOptDist=False,  
                                layer=None, mode='x', fontSize=0.01)
gtrace.draw.tools.drawAllOptics (d, opticsList, drawName=True, layer=None)
gtrace.draw.tools.transAll (objList, transVect)
gtrace.draw.tools.rotateAll (objList, angle, center)
```

Package Contents

Classes

<i>Canvas</i>	Canvas class
<i>Layer</i>	Layer class
<i>Shape</i>	Shape class
<i>Line</i>	Line class
<i>PolyLine</i>	A light weight poly-line
<i>Rectangle</i>	A rectangle
<i>Circle</i>	A circle
<i>Arc</i>	An arc
<i>Text</i>	Text

Attributes

<i>__author__</i>
<i>__copyright__</i>
<i>__credits__</i>
<i>__license__</i>
<i>__version__</i>
<i>__maintainer__</i>
<i>__email__</i>
<i>__status__</i>
<i>pi</i>
<i>__author__</i>
<i>__copyright__</i>

continues on next page

Table 11 – continued from previous page

<code>__credits__</code>
<code>__license__</code>
<code>__version__</code>
<code>__maintainer__</code>
<code>__email__</code>
<code>__status__</code>

```

gtrace.draw.__author__ = Yoichi Aso
gtrace.draw.__copyright__ = Copyright 2011–2021, Yoichi Aso
gtrace.draw.__credits__ = ['Yoichi Aso']
gtrace.draw.__license__ = BSD
gtrace.draw.__version__ = 0.2.1
gtrace.draw.__maintainer__ = Yoichi Aso
gtrace.draw.__email__ = yoichi.aso@nao.ac.jp
gtrace.draw.__status__ = Beta
gtrace.draw.pi
gtrace.draw.__author__ = Yoichi Aso
gtrace.draw.__copyright__ = Copyright 2011–2021, Yoichi Aso
gtrace.draw.__credits__ = ['Yoichi Aso']
gtrace.draw.__license__ = BSD
gtrace.draw.__version__ = 0.2.1
gtrace.draw.__maintainer__ = Yoichi Aso
gtrace.draw.__email__ = yoichi.aso@nao.ac.jp
gtrace.draw.__status__ = Beta
class gtrace.draw.Canvas (unit='m')
    Bases: object
    Canvas class
    add_layer (self, name, color=(0, 0, 0))
    add_shape (self, shape, layername)
class gtrace.draw.Layer (name, color=(0, 0, 0))
    Bases: object
    Layer class
    add_shape (self, shape)

```

class gtrace.draw.**Shape**

Bases: object

Shape class

class gtrace.draw.**Line** (*start, stop, thickness=0*)

Bases: *Shape*

Line class

exception gtrace.draw.**NumberOfElementError**

Bases: BaseException

Common base class for all exceptions

`__init__` (*self, message*)

class gtrace.draw.**PolyLine** (*x, y, thickness=0*)

Bases: *Shape*

A light weight poly-line

class gtrace.draw.**Rectangle** (*point, width, height, thickness=0*)

Bases: *Shape*

A rectangle

class gtrace.draw.**Circle** (*center, radius, thickness=0*)

Bases: *Shape*

A circle

class gtrace.draw.**Arc** (*center, radius, startangle, stopangle, thickness=0, angle_in_rad=True*)

Bases: *Shape*

An arc

Note that angles are stored in rad.

class gtrace.draw.**Text** (*text, point, height=1.0, rotation=0.0, angle_in_rad=True*)

Bases: *Shape*

Text

Note that angles are stored in rad.

gtrace.optics

gtrace.optics

This package provides utility functions and classes for Gaussian optics.

Submodules

gtrace.optics.cavity

cavity.py - A Cavity class and related functions for representing a Fabry-Perot cavity

Module Contents

Classes

<i>Cavity</i>	A class to represent a Fabry-Perot cavity.
---------------	--

Functions

<i>finesse</i> (r1, r2, power=False)	Returns the finesse of a cavity
--------------------------------------	---------------------------------

Attributes

<i>pi</i>
<i>c</i>
<i>sqrt</i>

gtrace.optics.cavity.**pi**

gtrace.optics.cavity.**c** = 299792458.0

gtrace.optics.cavity.**sqrt**

class gtrace.optics.cavity.**Cavity** (*r1=0.9, r2=0.99, L=1.0, R1=- 1.5, R2=1.5, wl=1.064e-06, power=False*)

Bases: traits.api.HasTraits

A class to represent a Fabry-Perot cavity.

r1
Input mirror reflectivity (amplitude)
Type float

r2
End mirror reflectivity (amplitude)
Type float

rp1
Input mirror reflectivity (power)
Type float

rp2
End mirror reflectivity (power)
Type float

L
Length
Type float

R1
ROC of the input mirror (positive when concave to incident light, i.e. convex seen from inside the cavity)
Type float

R2
ROC of the end mirror (positive when concave to incident light, i.e. concave seen from inside the cavity)
Type float

w1
Wavelength
Type float

r1

r2

rp1

rp2

tp1

tp2

t1

t2

L

R1

R2

g1

g2

w1

_R1_changed (*self, old, new*)

_R2_changed (*self, old, new*)

_L_changed (*self, old, new*)

_r1_changed (*self, old, r1*)

_r2_changed (*self, old, r2*)

finesse (*self*)
Returns the finesse of the cavity.

storageTime (*self*)
Storage time

pole (*self*)
Cavity pole frequency [Hz]

Nbounce (*self*)
Bounce number

powerGain (*self*)
Ratio of the intra-cavity power to the input power.

FSR (*self*)

Returns the free spectral range of the cavity.

modeSpacing (*self*)

Return the transverse mode spacing of the cavity (commonly called gamma). It is a fractional number defined by $\gamma = (\text{mode spacing frequency})/\text{FSR}$.

waist (*self*, *size=False*)

Return the q-parameter or the radius of the beam at the cavity waist.

Parameters *size* (*boolean*, *optional*) – if set to true, the first element of the returned tuple will be the waist size, rather than the q-parameter.

Returns (*q0*, *d*) – This function returns a tuple with two elements. The first element is the q-parameter of the cavity mode at the cavity waist. If *size=True* is given, it becomes the waist size ($1/e^2$ radius). The second element is the distance of the cavity waist from the input mirror.

Return type (complex, float)

spotSize (*self*)

Returns the beam spot sizes on the input and end mirrors as a tuple (*w1*, *w2*).

trans (*self*, *f=0*, *d=0*)

Returns the amplitude transmissivity of the cavity. It assumes the cavity was locked to the incident light first. Then computes the amplitude transmissivity for the light with a frequency shift *f* from the original light with the cavity length changed by *d* from the initial state.

Parameters

- *f* (*float*, *optional*) – Frequency shift of the light in Hz. Defaults 0.
- *d* (*float*, *optional*.) – Cavity length detuning in m. Defaults 0.

Returns The amplitude transmissivity of the cavity (a complex number).

Return type complex

refl (*self*, *f=0*, *d=0*)

Returns the amplitude reflectivity of the cavity. It assumes the cavity was locked to the incident light first. Then computes the amplitude reflectivity for the light with a frequency shift *f* from the original light with the cavity length changed by *d* from the initial state.

Parameters

- *f* (*float*, *optional*) – Frequency shift of the light in Hz. Defaults 0.
- *d* (*float*, *optional*.) – Cavity length detuning in m. Defaults 0.

Returns The amplitude reflectivity of the cavity (a complex number).

Return type complex

intra (*self*, *f=0*, *d=0*)

Returns the intra cavity field amplitude. It assumes the cavity was locked to the incident light first. Then computes the intra-cavity field amplitude for the light with a frequency shift *f* from the original light with the cavity length changed by *d* from the initial state.

Parameters

- *f* (*float*, *optional*) – Frequency shift of the light in Hz. Defaults 0.
- *d* (*float*, *optional*.) – Cavity length detuning in m. Defaults 0.

Returns The intra-cavity field amplitude at the input mirror surface (a complex number).

Return type complex

`gtrace.optics.cavity.finesse(r1, r2, power=False)`

Returns the finesse of a cavity

Parameters

- **r1** (*float*) – Reflectivity of the first mirror.
- **r2** (*float*) – Reflectivity of the second mirror.
- **power** (*boolean, optional*) – If True, r1 and r2 are treated as power reflectivities. Otherwise, r1 and r2 are regarded as amplitude reflectivities. Defaults False.

`gtrace.optics.consts`

Module Contents

Functions

<code>sellmeier(wl, B1, B2, B3, C1, C2, C3)</code>	Calculate index of refraction using Sellmeiers equation
<code>n_fused_silica(wl)</code>	Calculate the index of refraction of fused silica for a given wavelength.
<code>n_sapphire_ordinary(wl)</code>	Calculate the index of refraction of Sapphire ordinary axis for a given wavelength.
<code>n_sapphire_extraordinary(wl)</code>	Calculate the index of refraction of Sapphire extraordinary axis for a given wavelength.

Attributes

<code>n_fused_silica_532nm</code>
<code>n_fused_silica_1064nm</code>
<code>n_fused_silica_1550nm</code>
<code>no_sapphire_532nm</code>
<code>ne_sapphire_532nm</code>
<code>no_sapphire_1064nm</code>
<code>ne_sapphire_1064nm</code>
<code>no_sapphire_1550nm</code>
<code>ne_sapphire_1550nm</code>

`gtrace.optics.consts.sellmeier(wl, B1, B2, B3, C1, C2, C3)`

Calculate index of refraction using Sellmeiers equation

$$n^2 = 1 + B1 * w^2 / (w^2 - C1) + B2 * w^2 / (w^2 - C2) + B3 * w^2 / (w^2 - C3)$$

See below for the coefficients for specific materials. http://www.cvimellesgriot.com/products/Documents/Catalog/Dispersion_Equations.pdf

`gtrace.optics.consts.n_fused_silica(wl)`

Calculate the index of refraction of fused silica for a given wavelength.

`gtrace.optics.consts.n_sapphire_ordinary(wl)`

Calculate the index of refraction of Sapphire ordinary axis for a given wavelength.

`gtrace.optics.consts.n_sapphire_extraordinary(wl)`

Calculate the index of refraction of Sapphire extraordinary axis for a given wavelength.

`gtrace.optics.consts.n_fused_silica_532nm = 1.46071`

`gtrace.optics.consts.n_fused_silica_1064nm = 1.45`

`gtrace.optics.consts.n_fused_silica_1550nm = 1.444`

`gtrace.optics.consts.no_sapphire_532nm = 1.7717`

`gtrace.optics.consts.ne_sapphire_532nm = 1.76355`

`gtrace.optics.consts.no_sapphire_1064nm = 1.75449`

`gtrace.optics.consts.ne_sapphire_1064nm = 1.74663`

`gtrace.optics.consts.no_sapphire_1550nm = 1.74618`

`gtrace.optics.consts.ne_sapphire_1550nm = 1.73838`

`gtrace.optics.gaussian`

gaussian - Gaussian Optics Module

This module contains several utility functions for gaussian optics.

Module Contents

Functions

<code>modeSpacing(g1, g2)</code>	
<code>q2zr(q)</code>	Convert a q-parameter to Rayleigh range.
<code>q2w(q, wl=1064 * nm)</code>	Convert a q-parameter to the beam size
<code>q2R(q)</code>	Convert a q-parameter to the ROC
<code>Rw2q(ROC=1.0, w=1.0, wl=1.064e-06)</code>	Get the q-parameter from the ROC and w.
<code>InvROCandW2q(invROC=0.0, w=1.0, wl=1.064e-06)</code>	Get the q-parameter from the inverse ROC and w.
<code>zr2w0(zr, wl=1064 * nm)</code>	Convert Rayleigh range to the waist size
<code>w02zr(w0, wl=1064 * nm)</code>	Convert waist size to Rayleigh range
<code>modeMatching(q1, q2x, q2y=False)</code>	Mode matching between two beams with different q-parameters.
<code>modeMatchingElliptic(q1x, q1y, q2x, q2y)</code>	Mode matching between two elliptic beams.
<code>optimalMatching(q1, q2)</code>	Returns a mode (q-parameter) which best matches the given

continues on next page

Table 17 – continued from previous page

<code>qToRadius(q, wl=1.064e-06)</code>	Convert a q-parameter to the beam size
<code>qToROC(q)</code>	Convert a q-parameter to radius of curvature.
<code>ROCandWtoQ(ROC=1.0, w=1.0, wl=1.064e-06)</code>	Convert radius of curvature and beam width to q-parameter
<code>beamClip(a=1.0, w=3.0)</code>	Beam clip
<code>appertureCut(r=1.0, w=3.0)</code>	Apperture cut.

Attributes

`pi`

`sqrt`

`gtrace.optics.gaussian.pi`

`gtrace.optics.gaussian.sqrt`

`gtrace.optics.gaussian.modeSpacing(g1, g2)`

`gtrace.optics.gaussian.q2zr(q)`

Convert a q-parameter to Rayleigh range.

Parameters `q` (*complex*) – Beam parameter.

Returns `zr` – Rayleigh range.

Return type float

`gtrace.optics.gaussian.q2w(q, wl=1064 * nm)`

Convert a q-parameter to the beam size

Parameters

- `q` (*complex*) – Beam parameter.
- `wl` (*float, optional*) – Wavelength. Defaults 1064*nm.

Returns `w` – Beam size.

Return type float

`gtrace.optics.gaussian.q2R(q)`

Convert a q-parameter to the ROC

Parameters `q` (*complex*) – Beam parameter.

Returns Radius of curvature.

Return type float

`gtrace.optics.gaussian.Rw2q(ROC=1.0, w=1.0, wl=1.064e-06)`

Get the q-parameter from the ROC and w.

ROC [float, optional] Radius of curvature.

w [float, optional] Beam size. Defaults 1.0.

wl [float, optional] Wavelength. Defaults 1064*nm.

Returns Beam parameter.

Return type complex

`gtrace.optics.gaussian.InvROCandW2q(invROC=0.0, w=1.0, wl=1.064e-06)`

Get the q-parameter from the inverse ROC and w.

Parameters

- **invROC** (*float, optional*) – Inverse of the ROC. Defaults 0.0.
- **w** (*float, optional*) – Beam size. Defaults 1.0.
- **wl** (*float, optional*) – Wavelength. Defaults 1064*nm.

Returns Beam parameter.

Return type complex

`gtrace.optics.gaussian.zr2w0(zr, wl=1064 * nm)`

Convert Rayleigh range to the waist size

Parameters

- **zr** (*float*) – Rayleigh range.
- **wl** (*float, optional*) – Wavelength. Defaults 1064*nm.

Returns Waist size.

Return type float

`gtrace.optics.gaussian.w02zr(w0, wl=1064 * nm)`

Convert waist size to Rayleigh range

Parameters

- **w0** (*float*) – Waist size.
- **wl** (*float, optional*) – Wavelength. Defaults 1064*nm.

`gtrace.optics.gaussian.modeMatching(q1, q2x, q2y=False)`

Mode matching between two beams with different q-parameters. The axes of the two beams are assumed to be matched.

Parameters

- **q1** (*complex*) – q-parameter of the first beam. This beam is assumed to be circular.
- **q2x** (*complex*) – q-parameter of the second beam in x-direction. If the second beam is also circular, omit the next argument.
- **q2y** (*complex, optional*) – q-parameter of the second beam in y-direction. Specify this parameter if the second beam is elliptic. Defaults False.

`gtrace.optics.gaussian.modeMatchingElliptic(q1x, q1y, q2x, q2y)`

Mode matching between two elliptic beams.

Parameters

- **q1x** (*complex*) – q-parameter of the first beam in x-direction.
- **q1y** (*complex*) – q-parameter of the first beam in y-direction.
- **q2x** (*complex*) – q-parameter of the second beam in x-direction.
- **q2y** (*complex*) – q-parameter of the second beam in y-direction.

Returns**Return type** float`gtrace.optics.gaussian.optimalMatching(q1, q2)`

Returns a mode (q-parameter) which best matches the given two q-parameters, q1 and q2.

Parameters

- **q1** (*complex*) – q-parameter of the first beam. This beam is assumed to be circular.
- **q2** (*complex*) – q-parameter of the second beam. This beam is assumed to be circular.

Returns

(q, match)

q: The best matching q-parameter

match: Mode matching rate

Return type (complex, match?)`gtrace.optics.gaussian.qToRadius(q, wl=1.064e-06)`

Convert a q-parameter to the beam size

Parameters

- **q** (*complex*) – Beam parameter.
- **wl** (*float, optional*) – Wavelength. Defaults 1064e-9.

Returns Radius.**Return type** float`gtrace.optics.gaussian.qToROC(q)`

Convert a q-parameter to radius of curvature.

Parameters **q** (*complex*) – Beam parameter.**Returns** Radius of curvature.**Return type** float`gtrace.optics.gaussian.ROCandWtoQ(ROC=1.0, w=1.0, wl=1.064e-06)`

Convert radius of curvature and beam width to q-parameter

Parameters

- **ROC** (*float, optional*) – Radius of curvature. Defaults to 1.0.
- **w** (*float, optional*) – Beam width. Defaults to 1.0.
- **wl** (*float, optional*) – Wavelength. Defaults to 1064e-9

Returns q-parameter**Return type** complex`gtrace.optics.gaussian.beamClip(a=1.0, w=3.0)`

Beam clip

Parameters

- **a** (*float*) –
- **w** (*float*) –

Returns

Return type float

`gtrace.optics.gaussian.apertureCut` ($r=1.0$, $w=3.0$)
Aperture cut.

Parameters

- **r** (*float*) –
- **w** (*float*) –

Returns

Return type float

`gtrace.optics.geometric`

Module Contents

Functions

<code>deflection_angle</code> (<i>theta</i> , <i>n1</i> , <i>n2</i> , <i>deg=True</i>)	Calculate deflection angle according to Snell's law.
<code>line_plane_intersection</code> (<i>pos</i> , <i>dirVect</i> , <i>plane_center</i> , <i>normalVector</i> , <i>diameter</i>)	Compute the intersection point between a line
<code>line_arc_intersection</code> (<i>pos</i> , <i>dirVect</i> , <i>chord_center</i> , <i>chordNormVect</i> , <i>invROC</i> , <i>diameter</i> , <i>verbose=False</i>)	Compute the intersection point between a line
<code>vector_rotation_2D</code> (<i>vect</i> , <i>angle</i>)	Rotate a 2D vector by an angle.
<code>vector_normalize</code> (<i>vect</i>)	Normalize a vector
<code>normSpheric</code> (<i>normAngle</i> , <i>invROC</i> , <i>dist_from_center</i>)	Returns the local normal angle of a spheric mirror
<code>refl_defl_angle</code> (<i>beamAngle</i> , <i>normAngle</i> , <i>n1</i> , <i>n2</i> , <i>invROC=None</i>)	Returns a tuples of reflection and deflection angles.
<code>cyl_refl_defl_angle</code> (<i>beamAngle</i> , <i>normAngle</i> , <i>n1</i> , <i>n2</i> , <i>invROC=None</i> , <i>curve_direction='h'</i>)	Returns a tuples of reflection and deflection angles for incidence of a beam into a cylindrical surface.
<code>vc_deflect</code> (<i>theta</i> , <i>theta1</i> , <i>n1</i> , <i>n2</i>)	Deflection angle helper function for VariCAD.
<code>vc_reflect</code> (<i>theta</i> , <i>theta1</i>)	Convert <i>theta</i> and <i>theta1</i> to 0-360 format.

Attributes

pi

`gtrace.optics.geometric.pi`

`gtrace.optics.geometric.deflection_angle` (*theta*, *n1*, *n2*, *deg=True*)
Calculate deflection angle according to Snell's law.

Parameters

- **theta** (*float*) – Angle of incidence.
- **n1** (*float*) – Refractive index of the first medium.

- **n2** (*float*) – Refraction index of the second medium.
- **deg** (*boolean, optional*) – True if theta is specified in degrees.

`gtrace.optics.geometric.line_plane_intersection` (*pos, dirVect, plane_center, normalVector, diameter*)

Compute the intersection point between a line and a plane

Parameters

- **pos** (*array*) – The position of the end point of the line.
- **dirVect** (*array*) – The directional vector specifying the line.
- **plane_center** (*array*) – The position of the center of the plane.
- **normalVector** (*array*) – The normal vector of the plane.
- **diameter** (*float*) – The diameter of the plane.

Returns The returned value is a dictionary with the following keys: “Intersection Point”: numpy array of the coordinates of the intersection point. “isHit”: A boolean value of whether the line intersects with the plane or not. “distance”: Distance between the origin of the line and the intersection point. “distance from center”: Distance between the center of the plane and the intersection point.

Return type dict

`gtrace.optics.geometric.line_arc_intersection` (*pos, dirVect, chord_center, chordNormVect, invROC, diameter, verbose=False*)

Compute the intersection point between a line and an arc.

Parameters

- **pos** (*array*) – Origin of the line.
- **dirVect** (*array*) – Direction of the line.
- **chord_center** (*array*) – The center of the chord made by the arc.
- **chordNormVect** (*array*) – Normal vector of the chord.
- **invROC** (*float*) – Inverse of the ROC of the arc. Positive for concave surface.
- **diameter** (*float*) – Length of the chord.
- **verbose** (*boolean, optional*) – Prints useful information.

Returns The returned value is a dictionary with the following keys: “Intersection Point”: numpy array of the coordinates of the intersection point. “isHit”: A boolean value of whether the line intersects with the plane or not. “distance”: Distance between the origin of the line and the intersection point. “localNormVect”: localNormVect, “localNormAngle”: localNormAngle.

Return type dict

`gtrace.optics.geometric.vector_rotation_2D` (*vect, angle*)

Rotate a 2D vector by an angle.

Parameters

- **vect** (*array*) – A 2D vector.
- **angle** (*float*) – Angle of rotation in radians.

Returns The rotated vector.

Return type array

`gtrace.optics.geometric.vector_normalize(vect)`

Normalize a vector

Parameters `vect` (*array*) – The vector to be normalized

Returns The normalized vector.

Return type `array`

`gtrace.optics.geometric.normSpheric(normAngle, invROC, dist_from_center)`

Returns the local normal angle of a spheric mirror at a distance from the center.

Parameters

- **normAngle** (*float*) – The angle formed by the normal vector of the mirror at the center and the x-axis.
- **invROC** (*float*) – $1/R$, where R is the ROC of the mirror.
- **dist_from_center** (*float*) – The distance from the center of the point where the local normal is requested. This is a signed value. For a mirror facing +x (the normal vector points towards positive x direction), this distance is positive for points with positive y coordinate, and negative for points with negative y coordinate.

Returns The local normal angle of a spheric mirror at a distance from the center.

Return type `float`

`gtrace.optics.geometric.refl_defl_angle(beamAngle, normAngle, n1, n2, invROC=None)`

Returns a tuples of reflection and deflection angles.

Parameters

- **beamAngle** (*float*) – The angle formed by the propagation direction vector of the incident beam and the x-axis.
- **normAngle** (*float*) – The angle formed by the normal vector of the surface and the x-axis.
- **n1** (*float*) – Index of refraction of the incident side medium.
- **n2** (*float*) – Index of refraction of the transmission side medium.
- **invROC** (*float or None, optional*) – Inverse of the radius of curvature.

Returns

- *6-tuple or 2-tuple*
- (*reflAngle, deflAngle, Mrx, Mry, Mtx, Mty*) or (*reflAngle, deflAngle*)

`gtrace.optics.geometric.cyl_refl_defl_angle(beamAngle, normAngle, n1, n2, invROC=None, curve_direction='h')`

Returns a tuples of reflection and deflection angles for incidence of a beam into a cylindrical surface.

Parameters

- **beamAngle** (*float*) – The angle formed by the propagation direction vector of the incident beam and the x-axis.
- **normAngle** (*float*) – The angle formed by the normal vector of the surface and the x-axis.
- **n1** (*float*) – Index of refraction of the incident side medium.
- **n2** (*float*) – Index of refraction of the transmission side medium.
- **invROC** (*float or None, optional*) – Inverse of the radius of curvature.

- **curve_direction** (*str*, *optional*) – Direction of curvature. Either ‘h’ or ‘v’.

`gtrace.optics.geometric.vc_deflect` (*theta*, *theta1*, *n1*, *n2*)

Deflection angle helper function for VariCAD.

Parameters

- **theta** (*float*) – Angle of the surface measured from right.
- **theta1** (*float*) – Angle of the incident beam measured from right.
- **n1** (*float*) – Index of refraction of the incident side medium.
- **n2** (*float*) – Index of refraction of the transmission side medium.

Returns **phi2** – Angle of the deflected beam measured from right.

Return type `float`

`gtrace.optics.geometric.vc_reflect` (*theta*, *theta1*)

Convert theta and theta1 to 0-360 format.

Parameters

- **theta** (*float*) – Angle of the surface measured from right.
- **theta1** (*float*) – Angle of the incident beam measured from right.

Returns

Return type `float`

`gtrace.optics.unit`

Module Contents

Functions

`rad2deg`(*rad*)

`deg2rad`(*deg*)

Attributes

`pi`

`km`

`m`

`cm`

`mm`

continues on next page

Table 22 – continued from previous page

<i>um</i>
<i>nm</i>
<i>kW</i>
<i>W</i>
<i>mW</i>
<i>uW</i>
<i>THz</i>
<i>GHz</i>
<i>MHz</i>
<i>kHz</i>
<i>Hz</i>
<i>mHz</i>
<i>uHz</i>
<i>ppm</i>

```

gtrace.optics.unit.pi
gtrace.optics.unit.km = 1000.0
gtrace.optics.unit.m = 1.0
gtrace.optics.unit.cm = 0.01
gtrace.optics.unit.mm = 0.001
gtrace.optics.unit.um = 1e-06
gtrace.optics.unit.nm = 1e-09
gtrace.optics.unit.kW = 1000.0
gtrace.optics.unit.W = 1.0
gtrace.optics.unit.mW = 0.001
gtrace.optics.unit.uW = 1e-06
gtrace.optics.unit.THz = 1000000000000.0
gtrace.optics.unit.GHz = 1000000000.0
gtrace.optics.unit.MHz = 1000000.0
gtrace.optics.unit.kHz = 1000.0
gtrace.optics.unit.Hz = 1.0

```

```
gtrace.optics.unit.mHz = 0.001
gtrace.optics.unit.uHz = 1e-06
gtrace.optics.unit.ppm = 1e-06
gtrace.optics.unit.rad2deg(rad)
gtrace.optics.unit.deg2rad(deg)
```

Package Contents

```
gtrace.optics.__author__ = Yoichi Aso
gtrace.optics.__copyright__ = Copyright 2011-2021, Yoichi Aso
gtrace.optics.__credits__ = ['Yoichi Aso']
gtrace.optics.__license__ = BSD
gtrace.optics.__version__ = 0.2.1
gtrace.optics.__maintainer__ = Yoichi Aso
gtrace.optics.__email__ = yoichi.aso@nao.ac.jp
gtrace.optics.__status__ = Beta
```

5.1.2 Submodules

gtrace.beam

gtrace.beam

A module to define GaussianBeam class.

Module Contents

Classes

<i>GaussianBeam</i>	This is a class to represent a Gaussian beam.
---------------------	---

Functions

<i>optFunForStartPointR</i> (phi, Mrot, R, q0, k, sigma, side)	A function to return the distance between the point on
<i>optimStartPointR</i> (theta, R, q0, k, sigma)	Calculate optimal starting point.
<i>optFunForEndPointR</i> (phi, Mrot, R, q0, k, sigma, side)	A function to return the distance between the point on
<i>optimEndPointR</i> (theta, R, q0, k, sigma)	Calculate optimal end point.
<i>optFunForFlat</i> (a, Mrot, q0, k, sigma, side)	A function to return the distance between the point on
<i>optimCrossPointFlat</i> (theta, q0, k, sigma)	Calculate optimal cross point.

Attributes

*pi**array**sqr**__author__**__copyright__**__credits__**__license__**__version__**__maintainer__**__email__**__status__*gtrace.beam.**pi**gtrace.beam.**array**gtrace.beam.**sqr**gtrace.beam.**__author__** = Yoichi Asogtrace.beam.**__copyright__** = Copyright 2011-2021, Yoichi Asogtrace.beam.**__credits__** = ['Yoichi Aso']gtrace.beam.**__license__** = BSDgtrace.beam.**__version__** = 0.2.1gtrace.beam.**__maintainer__** = Yoichi Asogtrace.beam.**__email__** = yoichi.aso@nao.ac.jpgtrace.beam.**__status__** = Beta

```
class gtrace.beam.GaussianBeam(q0=1j * 2 * pi / 1064 * nm * 1e-06 / 2, q0x=False,
                               q0y=False, pos=[0.0, 0.0], length=1.0, dirAngle=0.0,
                               dirVect=None, wl=1064 * nm, P=1 * W, n=1.0, name='Beam',
                               layer='main_beam')
```

Bases: traits.api.HasTraits

This is a class to represent a Gaussian beam. A GaussianBeam object has its origin (pos) and a propagation direction (dirVect or dirAngle). A GaussianBeam is characterized by q-parameter(s) at its origin. The beam can be either circular or elliptic. In order to deal with elliptic beams, some parameters are stored in pairs like (q0x, q0y). x and y denote the axes of the cross section of the beam. x-axis is parallel to the paper and the y-axis is perpendicular to the paper.

A beam object can be propagated through a free space or made to interact with an optics.

As a beam propagate through optical system, optical distance and Gouy phase are accumulated.

q
q-parameter of the beam. If the beam is elliptic, q is the q-parameter of the best matching circular mode.

Type complex

qx
q-parameter of the beam in the x-direction.

Type complex

qy
q-parameter of the beam in the y-direction.

Type complex

pos
Position of the beam origin (x, y).

Type array

dirVect
Propagation direction vector.

Type array

dirAngle
Propagation direction angle measured from the positive x-axis.

Type float

length
Length of the beam (used for DXF export)

Type float

layer
Layer name of the beam when exported to a DXF file.

Type str

name
Name of the beam

Type str

wl
Wavelength in vacuum. Not the wavelength in the medium.

Type float

n
Index of refraction of the medium the beam is passing through.

Type float

P
Power.

Type float

wx
Beamwidth in x-direction.

Type float

wy

Beamwidth in y-direction.

Type float

optDist

Accumulated optical distance.

Type float

Gouyx

Accumulated Gouy phase in x-direction.

Type float

Gouyy

Accumulated Gouy phase in y-direction.

Type float

Mx

ABCD matrix in x-direction. This is a 2x2 matrix representing the product of ABCD transformations applied to this beam. It defaults to an identity matrix. Whenever a beam experience an ABCD matrix transformation, such as propagation in the space or reflection by a curved mirror, the applied ABCD matrix is multiplied to this matrix, so that we can keep track of what kind of transformations were made during beam propagation.

Type array

My

ABCD matrix in y-direction. The meaning is the same as Mx.

Type array

departSurfAngle

The angle formed by x-axis and the normal vector of the surface from which the beam is departing. Default is None. Used by the drawing routine.

Type None

departSurfInvROC

Inverse of the ROC of the surface from which the beam is departing. The ROC is positive for a concave surface seen from the beam side. Default is None. Used by the drawing routine.

Type None

incSurfAngle

The angle formed by the x-arm and the normal vector of the surface to which the beam is incident. Default is None. Used by the drawing routine.

Type None

incSurfInvROC

Inverse of the ROC of the surface to which the beam is incident. The ROC is positive for a concave surface seen from the beam side. Default is None. Used by the drawing routine.

Type None

stray_order

An integer indicating if this beam is a stray light or not. The default value is 0. Every time a beam is reflected by an AR surface or transmits an HR surface, this counter is increased by 1.

Type int

name
wl
P
q
qx
qy
qrx
qry
Gouyx
Gouyy
wx
wy
n
pos
length
layer
dirVect
dirAngle
optDist
Mx
My

copy (*self*)
Make a deep copy.

propagate (*self*, *d*)
Propagate the beam by a distance *d* from the current position. *self.n* is used as the index of refraction. During this process, the optical distance traveled is added to *self.optDist*. *self.Goux* and *self.Gouyy* are also updated to record the Gouy phase change.

Parameters *d* (*float*) – Distance.

ABCDTrans (*self*, *ABCDx*, *ABCDy=None*)
Apply ABCD transformation to the beam.

Parameters

- **ABCDx** (*array*) – ABCD matrix for x-direction.
- **ABCDy** (*array or None, optional.*) – ABCD matrix for y-direction. Defaults None. If None, set to ABCDx.

rotate (*self*, *angle*, *center=False*)
Rotate the beam around ‘center’. If center is not given, the beam is rotated around *self.pos*.

Parameters

- **angle** (*float*) – Rotation angle in radians.

- **center** (*array or boolean.*) – Center of rotation. Should be an array of shape(2,). Defaults False.

translate (*self, trVect*)

Translate the beam by the direction and the distance specified by a vector.

Parameters **trVect** (*array*) – A vector to specify the translation direction and distance. Should be an array of shape(2,)

flip (*self, flipDirVect=True*)

Change the propagation direction of the beam by 180 degrees. This is equivalent to the reflection of the beam by a spherical mirror with the same ROC as the beam.

If optional argument flipDirVect is set to False, the propagation direction of the beam is not changed.

Parameters **flipDirVect** (*boolean, optional*) – Flip propagation direction. Defaults True.

width (*self, dist*)

Returns the beam width at a distance dist from the origin of the beam. The width is the radius where the light power becomes $1/e^2$.

Parameters **dist** (*float*) – Distance.

Returns The width of the beam in x and y direction.

Return type (float, float)

R (*self, dist=0.0*)

Returns the beam ROC at a distance dist from the origin of the beam.

Parameters **dist** (*float, optional*) – Distance.

Returns Beam ROC.

Return type (float, float)

waist (*self*)

Return the tuples of waist size and distance

Returns {“Waist Size”: (float, float), “Waist Position”: (float, float)}

Return type dict

draw (*self, cv, sigma=3.0, mode='x', drawWidth=True, fontSize=False, drawPower=False, drawROC=False, drawGouy=False, drawOptDist=False, drawName=False, debug=False*)
Draw the beam into a DXF object.

Parameters

- **cv** (*gtrace.draw.draw.Canvas*) – gtrace canvas.
- **sigma** (*float, optional*) – The width of the beam drawn is $\sigma * (1/e^2 \text{ radius of the beam})$. The default is $\sigma = 3$. $\sigma = 2.7$ gives 1ppm diffraction loss. Defaults 3.
- **mode** (*str, optional*) – ‘avg’, ‘x’, or ‘y’. A beam can have different widths for x- and y- directions. If ‘avg’ is specified, the average of them are drawn. ‘x’ and ‘y’ specifies to show the width of the respective directions. Defaults ‘x’.
- **fontSize** (*float, optional*) – Size of the font used to show supplemental informations. Defaults False.
- **drawWidth** (*boolean, optional*) – Whether to draw width or not. Defaults True.

- **drawPower** (*boolean, optional*) – Whether to show the beam power. Defaults False.
- **drawROC** (*boolean, optional*) – Whether to show the ROC or not. Defaults False.
- **drawGouy** (*boolean, optional*) – Whether to show the Gouy phase or not. Defaults False.
- **drawOptDist** (*boolean, optional*) – Whether to show the accumulated optical distance or not. Defaults False.
- **drawName** (*boolean, optional*) – Whether draw the name of the beam or not. Defaults False.
- **debug** (*boolean, optional*) – Debug.

drawWidth (*self, cv, sigma, mode*)

Draw width on canvas.

Parameters

- **cv** (*gtrace.draw.draw.Canvas*) – The canvas.
- **sigma** (*float*) – The width of the beam drawn is $\sigma * (1/e^2 \text{ radius of the beam})$. The default is $\sigma = 3$. $\sigma = 2.7$ gives 1ppm diffraction loss.
- **mode** (*str*) – ‘avg’, ‘x’, or ‘y’. A beam can have different widths for x- and y- directions. If ‘avg’ is specified, the average of them are drawn. ‘x’ and ‘y’ specifies to show the width of the respective directions.

_dirAngle_changed (*self, old, new*)

_dirVect_changed (*self, old, new*)

_qx_changed (*self, old, new*)

_qy_changed (*self, old, new*)

_qrx_changed (*self, old, new*)

_qry_changed (*self, old, new*)

_n_changed (*self, old, new*)

gtrace.beam.optFunForStartPointR (*phi, Mrot, R, q0, k, sigma, side*)

A function to return the distance between the point on the spherical surface at an angle phi and the beam width at the same z.

Parameters

- **phi** (*float*) – phi
- **Mrot** (*array*) – Rotational transformation.
- **R** (*float*) – R
- **q0** (*complex*) – Beam parameter.
- **k** (*float*) – k
- **sigma** (*float*) – Beam width
- **side** – side

Returns Distance between the point on the spherical surface at an angle phi and the beam width at the same z.

Return type float

`gtrace.beam.optImStartPointR(theta, R, q0, k, sigma)`

Calculate optimal starting point.

theta [float] theta

R [float] R

q0 [complex] Beam parameter.

k [float] k

sigma [float] Beam width.

Returns Optimal starting point.

Return type (float, float)

`gtrace.beam.optFunForEndPointR(phi, Mrot, R, q0, k, sigma, side)`

A function to return the distance between the point on the spherical surface at an angle phi and the beam width at the same z.

Parameters

- **phi** (*float*) – phi
- **Mrot** (*array*) – Rotational transformation.
- **R** (*float*) – R
- **q0** (*complex*) – Beam parameter.
- **k** (*float*) – k
- **sigma** (*float*) – Beam width
- **side** – side

Returns Distance between the point on the spherical surface at an angle phi and the beam width at the same z.

Return type float

`gtrace.beam.optImEndPointR(theta, R, q0, k, sigma)`

Calculate optimal end point.

theta [float] theta

R [float] R

q0 [complex] Beam parameter.

k [float] k

sigma [float] Beam width.

Returns Optimal end point.

Return type (float, float)

`gtrace.beam.optFunForFlat(a, Mrot, q0, k, sigma, side)`

A function to return the distance between the point on the spherical surface (flat?) at an angle phi and the beam width at the same z.

Parameters

- **a** (*float*) – a

- **Mrot** (*array*) – Rotational transformation.
- **q0** (*complex*) – Beam parameter.
- **k** (*float*) – k
- **sigma** (*float*) – Beam width
- **side** – side

Returns Distance between the point on the spherical surface at an angle phi and the beam width at the same z.

Return type float

`gtrace.beam.optimCrossPointFlat` (*theta, q0, k, sigma*)

Calculate optimal cross point.

theta [float] theta

R [float] R

q0 [complex] Beam parameter.

k [float] k

sigma [float] Beam width.

Returns Optimal end point.

Return type (float, float)

`gtrace.nonsequential`

`gtrace.nonsequential`

A module to perform non-sequential trace of a beam in an optical system.

Module Contents

Functions

<code>non_seq_trace</code>	(<code>optList</code> , <code>src_beam</code> , <code>order=10</code> , <code>power_threshold=0.1</code> , <code>open_beam_length=1.0</code>)	Perform non-sequential trace of the source beam, <code>src_beam</code> ,
----------------------------	---	--

Attributes

`__author__`

`__copyright__`

`__credits__`

`__license__`

continues on next page

Table 27 – continued from previous page

<code>__version__</code>
<code>__maintainer__</code>
<code>__email__</code>
<code>__status__</code>

```

gtrace.nonsequential.__author__ = Yoichi Aso
gtrace.nonsequential.__copyright__ = Copyright 2011-2021, Yoichi Aso
gtrace.nonsequential.__credits__ = ['Yoichi Aso']
gtrace.nonsequential.__license__ = BSD
gtrace.nonsequential.__version__ = 0.2.1
gtrace.nonsequential.__maintainer__ = Yoichi Aso
gtrace.nonsequential.__email__ = yoichi.aso@nao.ac.jp
gtrace.nonsequential.__status__ = Beta
gtrace.nonsequential.non_seq_trace(optList, src_beam, order=10, power_threshold=0.1,
                                   open_beam_length=1.0)
    Perform non-sequential trace of the source beam, src_beam, through the optical system represented by a collec-
    tion of optics, optList.

```

Parameters

- **optList** (*list of gtrace.optcomp.Optics*) – List of optical components.
- **src_beam** (*gtrace.beam.GaussianBeam*) – The source beam object.
- **order** (*int, optional*) – An integer to specify how many times the internal reflections are computed. Defaults to 10.
- **power_threshold** (*float, optional*) – The power threshold for internal reflection calculation. If the power of an auxiliary beam falls below this threshold, further propagation of this beam will not be performed. Defaults to 0.1.
- **open_beam_length** (*float, optional*) – The default length for beams that are not hitting anything. Defaults to 1.0.

Returns `terminated_beam_list` – A list of beams.

Return type list of `gtrace.beam.GaussianBeam`

gtrace.optcomp

Define optical components for gtrace.

Module Contents

Classes

<i>Optics</i>	A general optics class from which other specific
<i>Mirror</i>	Representing a partial reflective mirror.
<i>CyMirror</i>	Representing a partial reflective cylindrical mirror. Note that both HR and AR surfaces are treated as cylindrical if you specify non-zero ROC for them. The curve directions of the two surfaces must be the same.

Attributes

<i>pi</i>
<i>array</i>
<i>sqrt</i>
<i>__author__</i>
<i>__copyright__</i>
<i>__credits__</i>
<i>__license__</i>
<i>__version__</i>
<i>__maintainer__</i>
<i>__email__</i>
<i>__status__</i>

```
gtrace.optcomp.pi
gtrace.optcomp.array
gtrace.optcomp.sqrt
gtrace.optcomp.__author__ = Yoichi Aso
gtrace.optcomp.__copyright__ = Copyright 2011-2021, Yoichi Aso
gtrace.optcomp.__credits__ = ['Yoichi Aso']
gtrace.optcomp.__license__ = BSD
gtrace.optcomp.__version__ = 0.2.1
gtrace.optcomp.__maintainer__ = Yoichi Aso
```

```
gtrace.optcomp.__email__ = yoichi.aso@nao.ac.jp
```

```
gtrace.optcomp.__status__ = Beta
```

```
class gtrace.optcomp.Optics
```

```
    Bases: traits.api.HasTraits
```

A general optics class from which other specific optics classes are derived.

name

Name of the optics.

Type str

center

Center position of the optics. array of shape(2,).

Type array

rotationAngle

This angle defines the orientation of the optics.

Type float

name

center

rotationAngle

isHit (*beam*)

A function to see if a beam hits this optics or not.

Parameters **beam** (`gtrace.beam.GaussianBeam`) – A GaussianBeam object to be interacted by the optics.

Returns

The return value is a dictionary with the following keys: `isHit`, `position`, `distance`, `face`

`isHit`: This is a boolean to answer whether the beam hit the optics or not.

`position`: A numpy array containing the coordinate values of the intersection point between the beam and the optics. If `isHit` is False, this parameter does not mean anything.

`distance` The distance between the beam origin and the intersection point.

`face`: An optional string identifying which face of the optics was hit. For example, `face` can be either “HR” or “AR” for a mirror. `face` can also be “side”, meaning that the beam hits a side of the optics, which is not meant to be used, e.g. the side of a mirror. In this case, the beam have reached a dead end.

Return type Dict

hit (*beam*, *order=0*, *threshold=0.0*)

A function to hit the optics with a beam.

This function attempts to hit the optics with the source beam, `beam`.

Parameters

- **beam** (`gtrace.beam.GaussianBeam`) – A GaussianBeam object to be interacted by the optics.
- **order** (*int*, *optional*) – An integer to specify how many times the internal reflections are computed. Defaults 0.

- **threshold**(*float, optional*) – The power threshold for internal reflection calculation. If the power of an auxiliary beam falls below this threshold, further propagation of this beam will not be performed. Defaults 0.0.

Returns

(isHit, beamDict, face)

isHit This is a boolean to answer whether the beam hit the optics or not.

beamDict A dictionary containing resultant beams.

face: An optional string identifying which face of the optics was hit. For a mirror, face is any of “HR”, “AR” or “side”.

Return type {boolean, dict, str}

`_isHitSurface_`(*self, beam, surface_center, normal_vector, surface_size=1.0, inv_ROC=0.0*)

Determine if a beam hit a surface

Parameters **beam** (`gtrace.beam.GaussianBeam`) – A GaussianBeam object to be interacted by the optics.

Returns

ans – A dictionary with the following keys: “isHit”: A boolean value whether the beam hit the surface or not. “Intersection Point”: numpy array of the coordinates of the intersection point. “distance”: Distance between the origin of the beam and the intersection point. “localNormVect”: A numpy array representing the normal vector of the surface

at the intersection point.

”localNormAngle”: The angle of the localNormVect.

Return type dict

class `gtrace.optcomp.Mirror` (*HRcenter=[0.0, 0.0], normAngleHR=0.0, normVectHR=None, diameter=25.0 * cm, thickness=15.0 * cm, wedgeAngle=0.25 * pi / 180.0, inv_ROC_HR=1.0 / 7000.0, inv_ROC_AR=0.0, Refl_HR=0.99, Trans_HR=0.01, Refl_AR=0.01, Trans_AR=0.99, n=1.45, name='Mirror', HRtransmissive=False, term_on_HR=False*)

Bases: `Optics`

Representing a partial reflective mirror.

`curve_direction`

Either ‘h’ or ‘v’. If it is ‘h’ the mirror is curved in horizontal plane. If ‘v’, it is vertical.

Type str

`HRcenter`

The position of the center of the arc of the HR surface. shape(2,).

Type array

`HRcenterC`

The position of the center of the chord of the HR surface. shape(2,).

Type array

`normVectHR`

Normal vector of the HR surface. shape(2,)

Type array

normAngleHR

Angle of the HR normal vector. In radians.

Type float

ARcenter

The position of the center of the AR surface. shape(2,)

Type array

normVectAR

Normal vector of the HR surface. shape(2,)

Type array

normAngleAR

Angle of the HR normal vector. In radians.

Type float

HRtransmissive

A boolean value defaults to False. If True, this mirror is supposed to transmit beams on the HR surface. Therefore, for the first encounter of a beam on the HR surface of this mirror will not increase the stray_order. This flag should be set to True for beam splitters and input test masses.

Type boolean

term_on_HR

If this is True, a beam with stray_order <= self.term_on_HR_order will be terminated when it hits on HR. This is to avoid the infinite loop of non-sequential trace by forming a cavity.

Type boolean

term_on_HR_order

Integer to specify the upper limit of the stray order used to judge whether to terminate the non sequential trace or not on HR reflection.

Type int

HRcenter**HRcenterC****sagHR****normVectHR****normAngleHR****ARcenter****ARcenterC****sagAR****normVectAR****normAngleAR****diameter****ARdiameter****thickness****wedgeAngle****n**

`inv_ROC_HR`

`inv_ROC_AR`

`Refl_HR`

`Trans_HR`

`Refl_AR`

`Trans_AR`

`copy (self)`

`get_side_info (self)`

Return information on the sides of the mirror. Returned value is a list of two tuples like [(center1, normVect1, length1), (center2, normVect2, length2)] Each tuple corresponds to a side. center1 is the coordinates of the center of the side line. normVect1 is the normal vector of the side line. length1 is the length of the side line.

Returns

Return type [(float, float, float), (float, float, float)]

`rotate (self, angle, center=False)`

Rotate the mirror. If center is not specified, the center of rotation is HRcenter. If center is given (as a vector), the center of rotation is center. center is a position vector in the global coordinates.

Parameters

- **angle** (*float*) – Angle of rotation.
- **center** (*array or boolean, optional*) – Center of rotation, or False.

`translate (self, trVect)`

`draw (self, cv, drawName=False)`

Draw itself

`isHit (self, beam)`

A function to see if a beam hits this optics or not.

Parameters **beam** (`gtrace.beam.GaussianBeam`) – A GaussianBeam object to be interacted by the optics.

Returns

The return value is a dictionary with the following keys: `isHit`, `position`, `distance`, `face`

`isHit`: This is a boolean to answer whether the beam hit the optics or not.

`position`: A numpy array containing the coordinate values of the intersection point between the beam and the optics. If `isHit` is False, this parameter does not mean anything.

`distance` The distance between the beam origin and the intersection point.

`face`: An optional string identifying which face of the optics was hit. For example, `face` can be either “HR” or “AR” for a mirror. `face` can also be “side”, meaning that the beam hits a side of the optics, which is not meant to be used, e.g. the side of a mirror. In this case, the beam have reached a dead end.

Return type Dict

hit (*self*, *beam*, *order=0*, *threshold=0.0*, *face=False*)

A function to hit the optics with a beam.

This function attempts to hit the optics with the source beam, *beam*.

Parameters

- **beam** (*gtrace.beam.GaussianBeam*) – A GaussianBeam object to be interacted by the optics.
- **order** (*int*, *optional*) – An integer to specify how many times the internal reflections are computed. Defaults 0.
- **threshold** (*float*, *optional*) – The power threshold for internal reflection calculation. If the power of an auxiliary beam falls below this threshold, further propagation of this beam will not be performed. Defaults 0.0.

Returns

(*isHit*, *beamDict*, *face*)

isHit This is a boolean to answer whether the beam hit the optics or not.

beamDict A dictionary containing resultant beams.

face: An optional string identifying which face of the optics was hit. For a mirror, *face* is any of “HR”, “AR” or “side”.

Return type {boolean, dict, str}

hitFromHR (*self*, *beam*, *order=0*, *threshold=0.0*, *verbose=False*)

Compute the reflected and deflected beams when an input beam hit the HR surface.

The internal reflections are computed as long as the number of internal reflections are below the *order* and the power of the reflected beams is over the threshold.

Parameters

- **beam** (*gtrace.beam.GaussianBeam*) – A GaussianBeam object to be interacted by the optics.
- **order** (*int*, *optional*) – An integer to specify how many times the internal reflections are computed. Defaults 0.
- **threshold** (*float*, *optional*) – The power threshold for internal reflection calculation. If the power of an auxiliary beam falls below this threshold, further propagation of this beam will not be performed. Defaults 0.0.
- **verbose** (*boolean*, *optional*) – Print useful information.

Returns *beams* – Dictionary of reflected and deflected beams.

Return type dict

hitFromAR (*self*, *beam*, *order=0*, *threshold=0.0*, *verbose=False*)

Compute the reflected and deflected beams when an input beam hit the AR surface.

The internal reflections are computed as long as the number of internal reflections are below the *order* and the power of the reflected beams is over the threshold.

Parameters

- **beam** (*gtrace.beam.GaussianBeam*) – A GaussianBeam object to be interacted by the optics.

- **order** (*int, optional*) – An integer to specify how many times the internal reflections are computed. Defaults 0.
- **threshold** (*float, optional*) – The power threshold for internal reflection calculation. If the power of an auxiliary beam falls below this threshold, further propagation of this beam will not be performed. Defaults 0.0.
- **verbose** (*boolean, optional*) – Print useful information.

Returns **beams** – Dictionary of reflected and deflected beams.

Return type dict

_normAngleHR_changed (*self, old, new*)

_normVectHR_changed (*self, old, new*)

_HRcenterC_changed (*self, old, new*)

_HRcenter_changed (*self, old, new*)

_center_changed (*self, old, new*)

_wedgeAngle_changed (*self, old, new*)

_inv_ROC_HR_changed (*self, old, new*)

_inv_ROC_AR_changed (*self, old, new*)

```
class gtrace.optcomp.CyMirror (HRcenter=[0.0, 0.0], normAngleHR=0.0, normVectHR=None,
                                diameter=25.0 * cm, thickness=15.0 * cm, wedgeAngle=0.25
                                * pi / 180.0, inv_ROC_HR=1.0 / 7000.0, inv_ROC_AR=0.0,
                                Refl_HR=0.99, Trans_HR=0.01, Refl_AR=0.01, Trans_AR=0.99,
                                n=1.45, name='Mirror', HRtransmissive=False,
                                term_on_HR=False, curve_direction='h')
```

Bases: *Mirror*

Representing a partial reflective cylindrical mirror. Note that both HR and AR surfaces are treated as cylindrical if you specify non-zero ROC for them. The curve directions of the two surfaces must be the same.

curve_direction

Either 'h' or 'v'. If it is 'h' the mirror is curved in horizontal plane. If 'v', it is vertical.

Type str

HRcenter

The position of the center of the arc of the HR surface. shape(2,).

Type array

HRcenterC

The position of the center of the chord of the HR surface. shape(2,).

Type array

normVectHR

Normal vector of the HR surface. shape(2,)

Type array

normAngleHR

Angle of the HR normal vector. In radians.

Type float

ARcenter

The position of the center of the AR surface. shape(2,)

Type array

normVectAR

Normal vector of the HR surface. shape(2,)

Type array

normAngleAR

Angle of the HR normal vector. In radians.

Type float

HRtransmissive

A boolean value defaults to False. If True, this mirror is supposed to transmit beams on the HR surface. Therefore, for the first encounter of a beam on the HR surface of this mirror will not increase the stray_order. This flag should be set to True for beam splitters and input test masses.

Type boolean

term_on_HR

If this is True, a beam with stray_order <= self.term_on_HR_order will be terminated when it hits on HR. This is to avoid the infinite loop of non-sequential trace by forming a cavity.

Type boolean

term_on_HR_order

Integer to specify the upper limit of the stray order used to judge whether to terminate the non sequential trace or not on HR reflection.

Type int

copy (*self*)

get_side_info (*self*)

Return information on the sides of the mirror. Returned value is a list of two tuples like [(center1, normVect1, length1), (center2, normVect2, length2)] Each tuple corresponds to a side. center1 is the coordinates of the center of the side line. normVect1 is the normal vector of the side line. length1 is the length of the side line.

Returns

Return type [(float, float, float), (float, float, float)]

isHit (*self*, *beam*)

A function to see if a beam hits this optics or not.

Parameters **beam** (`gtrace.beam.GaussianBeam`) – A GaussianBeam object to be interacted by the optics.

Returns

The return value is a dictionary with the following keys: `isHit`, `position`, `distance`, `face`

`isHit`: This is a boolean to answer whether the beam hit the optics or not.

`position`: A numpy array containing the coordinate values of the intersection point between the beam and the optics. If `isHit` is False, this parameter does not mean anything.

`distance` The distance between the beam origin and the intersection point.

`face`: An optional string identifying which face of the optics was hit. For example, `face` can be either “HR” or “AR” for a mirror. `face` can also be “side”, meaning that the beam hits a side of the optics, which is not meant to be used, e.g. the side of a mirror. In this case, the beam have reached a dead end.

Return type Dict

draw (*self*, *cv*, *drawName=False*)

Draw itself

hitFromHR (*self*, *beam*, *order=0*, *threshold=0.0*, *verbose=False*)

Compute the reflected and deflected beams when an input beam hit the HR surface.

The internal reflections are computed as long as the number of internal reflections are below the `order` and the power of the reflected beams is over the threshold.

Parameters

- **beam** (`gtrace.beam.GaussianBeam`) – A GaussianBeam object to be interacted by the optics.
- **order** (*int*, *optional*) – An integer to specify how many times the internal reflections are computed. Defaults 0.
- **threshold** (*float*, *optional*) – The power threshold for internal reflection calculation. If the power of an auxiliary beam falls below this threshold, further propagation of this beam will not be performed. Defaults 0.0.
- **verbose** (*boolean*, *optional*) – Print useful information.

Returns **beams** – Dictionary of reflected and deflected beams.

Return type dict

hitFromAR (*self*, *beam*, *order=0*, *threshold=0.0*, *verbose=False*)

Compute the reflected and deflected beams when an input beam hit the AR surface.

The internal reflections are computed as long as the number of internal reflections are below the `order` and the power of the reflected beams is over the threshold.

Parameters

- **beam** (`gtrace.beam.GaussianBeam`) – A GaussianBeam object to be interacted by the optics.
- **order** (*int*, *optional*) – An integer to specify how many times the internal reflections are computed. Defaults 0.
- **threshold** (*float*, *optional*) – The power threshold for internal reflection calculation. If the power of an auxiliary beam falls below this threshold, further propagation of this beam will not be performed. Defaults 0.0.
- **verbose** (*boolean*, *optional*) – Print useful information.

Returns **beams** – Dictionary of reflected and deflected beams.

Return type dict

gtrace.unit

Module Contents

Functions

rad2deg(rad)

deg2rad(deg)

Attributes

pi

__author__

__copyright__

__credits__

__license__

__version__

__maintainer__

__email__

__status__

km

m

cm

mm

um

nm

kW

W

mW

continues on next page

Table 31 – continued from previous page

<i>uW</i>
<i>THz</i>
<i>GHz</i>
<i>MHz</i>
<i>kHz</i>
<i>Hz</i>
<i>mHz</i>
<i>uHz</i>
<i>ppm</i>
<i>percent</i>

```
gtrace.unit.pi
gtrace.unit.__author__ = Yoichi Aso
gtrace.unit.__copyright__ = Copyright 2011, Yoichi Aso
gtrace.unit.__credits__ = ['Yoichi Aso']
gtrace.unit.__license__ = BSD
gtrace.unit.__version__ = 0.2.1
gtrace.unit.__maintainer__ = Yoichi Aso
gtrace.unit.__email__ = yoichi.aso@nao.ac.jp
gtrace.unit.__status__ = Beta
gtrace.unit.km = 1000.0
gtrace.unit.m = 1.0
gtrace.unit.cm = 0.01
gtrace.unit.mm = 0.001
gtrace.unit.um = 1e-06
gtrace.unit.nm = 1e-09
gtrace.unit.kW = 1000.0
gtrace.unit.W = 1.0
gtrace.unit.mW = 0.001
gtrace.unit.uW = 1e-06
gtrace.unit.THz = 1000000000000.0
gtrace.unit.GHz = 1000000000.0
```

```
gtrace.unit.MHz = 1000000.0
gtrace.unit.kHz = 1000.0
gtrace.unit.Hz = 1.0
gtrace.unit.mHz = 0.001
gtrace.unit.uHz = 1e-06
gtrace.unit.ppm = 1e-06
gtrace.unit.percent = 0.01
gtrace.unit.rad2deg(rad)
gtrace.unit.deg2rad(deg)
```

5.1.3 Package Contents

```
gtrace.__author__ = Yoichi Aso
gtrace.__copyright__ = Copyright 2011-2021, Yoichi Aso
gtrace.__credits__ = ['Yoichi Aso']
gtrace.__license__ = BSD
gtrace.__version__ = 0.2.1
gtrace.__maintainer__ = Yoichi Aso
gtrace.__email__ = yoichi.aso@nao.ac.jp
gtrace.__status__ = Beta
```


INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

g

- [gtrace](#), [21](#)
- [gtrace.beam](#), [136](#)
- [gtrace.draw](#), [21](#)
- [gtrace.draw.draw](#), [21](#)
- [gtrace.draw.dxf](#), [23](#)
- [gtrace.draw.renderer](#), [118](#)
- [gtrace.draw.tools](#), [119](#)
- [gtrace.nonsequential](#), [144](#)
- [gtrace.optcomp](#), [145](#)
- [gtrace.optics](#), [122](#)
- [gtrace.optics.cavity](#), [122](#)
- [gtrace.optics.consts](#), [126](#)
- [gtrace.optics.gaussian](#), [127](#)
- [gtrace.optics.geometric](#), [131](#)
- [gtrace.optics.unit](#), [134](#)
- [gtrace.unit](#), [155](#)

Symbols

_HRcenterC_changed() (gtrace.optcomp.Mirror method), 152
 _HRcenter_changed() (gtrace.optcomp.Mirror method), 152
 _L_changed() (gtrace.optics.cavity.Cavity method), 124
 _R1_changed() (gtrace.optics.cavity.Cavity method), 124
 _R2_changed() (gtrace.optics.cavity.Cavity method), 124
 __author__ (in module gtrace), 157
 __author__ (in module gtrace.beam), 137
 __author__ (in module gtrace.draw), 121
 __author__ (in module gtrace.draw.draw), 22
 __author__ (in module gtrace.draw.renderer), 118
 __author__ (in module gtrace.draw.tools), 119
 __author__ (in module gtrace.nonsequential), 145
 __author__ (in module gtrace.optcomp), 146
 __author__ (in module gtrace.optics), 136
 __author__ (in module gtrace.unit), 156
 __copyright__ (in module gtrace), 157
 __copyright__ (in module gtrace.beam), 137
 __copyright__ (in module gtrace.draw), 121
 __copyright__ (in module gtrace.draw.draw), 22
 __copyright__ (in module gtrace.draw.renderer), 118
 __copyright__ (in module gtrace.draw.tools), 119
 __copyright__ (in module gtrace.nonsequential), 145
 __copyright__ (in module gtrace.optcomp), 146
 __copyright__ (in module gtrace.optics), 136
 __copyright__ (in module gtrace.unit), 156
 __credits__ (in module gtrace), 157
 __credits__ (in module gtrace.beam), 137
 __credits__ (in module gtrace.draw), 121
 __credits__ (in module gtrace.draw.draw), 22
 __credits__ (in module gtrace.draw.renderer), 118
 __credits__ (in module gtrace.draw.tools), 119
 __credits__ (in module gtrace.nonsequential), 145
 __credits__ (in module gtrace.optcomp), 146
 __credits__ (in module gtrace.optics), 136
 __credits__ (in module gtrace.unit), 156
 __email__ (in module gtrace), 157
 __email__ (in module gtrace.beam), 137
 __email__ (in module gtrace.draw), 121
 __email__ (in module gtrace.draw.draw), 22
 __email__ (in module gtrace.draw.renderer), 118
 __email__ (in module gtrace.draw.tools), 119
 __email__ (in module gtrace.nonsequential), 145
 __email__ (in module gtrace.optcomp), 146
 __email__ (in module gtrace.optics), 136
 __email__ (in module gtrace.unit), 156
 __initi__ () (gtrace.draw.NumberOfElementError method), 122
 __initi__ () (gtrace.draw.draw.NumberOfElementError method), 23
 __initi__ () (gtrace.draw.dxf.NumberOfElementError method), 25
 __initi__ () (gtrace.draw.renderer.UnknownShapeError method), 118
 __license__ (in module gtrace), 157
 __license__ (in module gtrace.beam), 137
 __license__ (in module gtrace.draw), 121
 __license__ (in module gtrace.draw.draw), 22
 __license__ (in module gtrace.draw.renderer), 118
 __license__ (in module gtrace.draw.tools), 119
 __license__ (in module gtrace.nonsequential), 145
 __license__ (in module gtrace.optcomp), 146
 __license__ (in module gtrace.optics), 136
 __license__ (in module gtrace.unit), 156
 __maintainer__ (in module gtrace), 157
 __maintainer__ (in module gtrace.beam), 137
 __maintainer__ (in module gtrace.draw), 121
 __maintainer__ (in module gtrace.draw.draw), 22
 __maintainer__ (in module gtrace.draw.renderer), 118
 __maintainer__ (in module gtrace.draw.tools), 119
 __maintainer__ (in module gtrace.nonsequential), 145
 __maintainer__ (in module gtrace.optcomp), 146
 __maintainer__ (in module gtrace.optics), 136
 __maintainer__ (in module gtrace.unit), 156
 __status__ (in module gtrace), 157

__status__ (in module *gtrace.beam*), 137
 __status__ (in module *gtrace.draw*), 121
 __status__ (in module *gtrace.draw.draw*), 22
 __status__ (in module *gtrace.draw.renderer*), 118
 __status__ (in module *gtrace.draw.tools*), 119
 __status__ (in module *gtrace.nonsequential*), 145
 __status__ (in module *gtrace.optcomp*), 147
 __status__ (in module *gtrace.optics*), 136
 __status__ (in module *gtrace.unit*), 156
 __version__ (in module *gtrace*), 157
 __version__ (in module *gtrace.beam*), 137
 __version__ (in module *gtrace.draw*), 121
 __version__ (in module *gtrace.draw.draw*), 22
 __version__ (in module *gtrace.draw.renderer*), 118
 __version__ (in module *gtrace.draw.tools*), 119
 __version__ (in module *gtrace.nonsequential*), 145
 __version__ (in module *gtrace.optcomp*), 146
 __version__ (in module *gtrace.optics*), 136
 __version__ (in module *gtrace.unit*), 156
 _center_changed() (*gtrace.optcomp.Mirror*
 method), 152
 _dirAngle_changed() (*gtrace.beam.GaussianBeam*
 method), 142
 _dirVect_changed() (*gtrace.beam.GaussianBeam*
 method), 142
 _inv_ROC_AR_changed() (*gtrace.optcomp.Mirror*
 method), 152
 _inv_ROC_HR_changed() (*gtrace.optcomp.Mirror*
 method), 152
 isHitSurface() (*gtrace.optcomp.Optics* *method*),
 148
 _n_changed() (*gtrace.beam.GaussianBeam* *method*),
 142
 _normAngleHR_changed() (*gtrace.optcomp.Mirror*
 method), 152
 _normVectHR_changed() (*gtrace.optcomp.Mirror*
 method), 152
 _qrx_changed() (*gtrace.beam.GaussianBeam*
 method), 142
 _qry_changed() (*gtrace.beam.GaussianBeam*
 method), 142
 _qx_changed() (*gtrace.beam.GaussianBeam*
 method), 142
 _qy_changed() (*gtrace.beam.GaussianBeam*
 method), 142
 _r1_changed() (*gtrace.optics.cavity.Cavity* *method*),
 124
 _r2_changed() (*gtrace.optics.cavity.Cavity* *method*),
 124
 _wedgeAngle_changed() (*gtrace.optcomp.Mirror*
 method), 152

A

ABCDTrans() (*gtrace.beam.GaussianBeam* *method*),

140
 add_entity() (*gtrace.draw.dxf.DXF* *method*), 25
 add_entity() (*gtrace.draw.dxf.Layer* *method*), 25
 add_layer() (*gtrace.draw.Canvas* *method*), 121
 add_layer() (*gtrace.draw.draw.Canvas* *method*), 22
 add_layer() (*gtrace.draw.dxf.DXF* *method*), 25
 add_shape() (*gtrace.draw.Canvas* *method*), 121
 add_shape() (*gtrace.draw.draw.Canvas* *method*), 22
 add_shape() (*gtrace.draw.draw.Layer* *method*), 22
 add_shape() (*gtrace.draw.Layer* *method*), 121
 appertureCut() (in module *gtrace.optics.gaussian*),
 131
 Arc (class in *gtrace.draw*), 122
 Arc (class in *gtrace.draw.draw*), 23
 Arc (class in *gtrace.draw.dxf*), 26
 ARcenter (*gtrace.optcomp.CyMirror* attribute), 152
 ARcenter (*gtrace.optcomp.Mirror* attribute), 149
 ARcenterC (*gtrace.optcomp.Mirror* attribute), 149
 ARdiameter (*gtrace.optcomp.Mirror* attribute), 149
 array (in module *gtrace.beam*), 137
 array (in module *gtrace.optcomp*), 146

B

beamClip() (in module *gtrace.optics.gaussian*), 130
 blockrecords_template (in module
 gtrace.draw.dxf), 59
 blocks_template (in module *gtrace.draw.dxf*), 61

C

c (in module *gtrace.optics.cavity*), 123
 Canvas (class in *gtrace.draw*), 121
 Canvas (class in *gtrace.draw.draw*), 22
 Cavity (class in *gtrace.optics.cavity*), 123
 center (*gtrace.optcomp.Optics* attribute), 147
 Circle (class in *gtrace.draw*), 122
 Circle (class in *gtrace.draw.draw*), 23
 Circle (class in *gtrace.draw.dxf*), 26
 classes_template (in module *gtrace.draw.dxf*), 46
 cm (in module *gtrace.optics.unit*), 135
 cm (in module *gtrace.unit*), 156
 color_encode() (in module *gtrace.draw.dxf*), 26
 color_table (in module *gtrace.draw.dxf*), 26
 copy() (*gtrace.beam.GaussianBeam* *method*), 140
 copy() (*gtrace.optcomp.CyMirror* *method*), 153
 copy() (*gtrace.optcomp.Mirror* *method*), 150
 curve_direction (*gtrace.optcomp.CyMirror* at-
 tribute), 152
 curve_direction (*gtrace.optcomp.Mirror* attribute),
 148
 cyl_refl_defl_angle() (in module
 gtrace.optics.geometric), 133
 CyMirror (class in *gtrace.optcomp*), 152

D

deflection_angle() (in module *gtrace.optics.geometric*), 131

deg2rad() (in module *gtrace.optics.unit*), 136

deg2rad() (in module *gtrace.unit*), 157

departSurfAngle (*gtrace.beam.GaussianBeam* attribute), 139

departSurfInvROC (*gtrace.beam.GaussianBeam* attribute), 139

diameter (*gtrace.optcomp.Mirror* attribute), 149

dirAngle (*gtrace.beam.GaussianBeam* attribute), 138, 140

dirVect (*gtrace.beam.GaussianBeam* attribute), 138, 140

draw() (*gtrace.beam.GaussianBeam* method), 141

draw() (*gtrace.draw.dxf.Arc* method), 26

draw() (*gtrace.draw.dxf.Circle* method), 26

draw() (*gtrace.draw.dxf.Entity* method), 25

draw() (*gtrace.draw.dxf.Line* method), 25

draw() (*gtrace.draw.dxf.LwPolyLine* method), 26

draw() (*gtrace.draw.dxf.Rectangle* method), 26

draw() (*gtrace.draw.dxf.Text* method), 26

draw() (*gtrace.optcomp.CyMirror* method), 154

draw() (*gtrace.optcomp.Mirror* method), 150

drawAllBeams() (in module *gtrace.draw.tools*), 120

drawAllOptics() (in module *gtrace.draw.tools*), 120

drawOptSys() (in module *gtrace.draw.tools*), 119

drawWidth() (*gtrace.beam.GaussianBeam* method), 142

DXF (class in *gtrace.draw.dxf*), 25

E

entities_template0 (in module *gtrace.draw.dxf*), 63

entities_template1 (in module *gtrace.draw.dxf*), 63

Entity (class in *gtrace.draw.dxf*), 25

F

finesse() (*gtrace.optics.cavity.Cavity* method), 124

finesse() (in module *gtrace.optics.cavity*), 126

flip() (*gtrace.beam.GaussianBeam* method), 141

FSR() (*gtrace.optics.cavity.Cavity* method), 124

G

g1 (*gtrace.optics.cavity.Cavity* attribute), 124

g2 (*gtrace.optics.cavity.Cavity* attribute), 124

GaussianBeam (class in *gtrace.beam*), 137

get_side_info() (*gtrace.optcomp.CyMirror* method), 153

get_side_info() (*gtrace.optcomp.Mirror* method), 150

GHz (in module *gtrace.optics.unit*), 135

GHz (in module *gtrace.unit*), 156

Gouyx (*gtrace.beam.GaussianBeam* attribute), 139, 140

Gouyy (*gtrace.beam.GaussianBeam* attribute), 139, 140

gtrace

- module, 21

gtrace.beam

- module, 136

gtrace.draw

- module, 21

gtrace.draw.draw

- module, 21

gtrace.draw.dxf

- module, 23

gtrace.draw.renderer

- module, 118

gtrace.draw.tools

- module, 119

gtrace.nonsequential

- module, 144

gtrace.optcomp

- module, 145

gtrace.optics

- module, 122

gtrace.optics.cavity

- module, 122

gtrace.optics.consts

- module, 126

gtrace.optics.gaussian

- module, 127

gtrace.optics.geometric

- module, 131

gtrace.optics.unit

- module, 134

gtrace.unit

- module, 155

H

header_template1 (in module *gtrace.draw.dxf*), 26

header_template2 (in module *gtrace.draw.dxf*), 27

header_template3 (in module *gtrace.draw.dxf*), 27

hit() (*gtrace.optcomp.Mirror* method), 150

hit() (*gtrace.optcomp.Optics* method), 147

hitFromAR() (*gtrace.optcomp.CyMirror* method), 154

hitFromAR() (*gtrace.optcomp.Mirror* method), 151

hitFromHR() (*gtrace.optcomp.CyMirror* method), 154

hitFromHR() (*gtrace.optcomp.Mirror* method), 151

HRcenter (*gtrace.optcomp.CyMirror* attribute), 152

HRcenter (*gtrace.optcomp.Mirror* attribute), 148, 149

HRcenterC (*gtrace.optcomp.CyMirror* attribute), 152

HRcenterC (*gtrace.optcomp.Mirror* attribute), 148, 149

HRtransmissive (*gtrace.optcomp.CyMirror* attribute), 153

HRtransmissive (*gtrace.optcomp.Mirror* attribute), 149

Hz (in module *gtrace.optics.unit*), 135

Hz (in module *gtrace.unit*), 157

I

incSurfAngle (*gtrace.beam.GaussianBeam* attribute), 139

incSurfInvROC (*gtrace.beam.GaussianBeam* attribute), 139

intra() (*gtrace.optics.cavity.Cavity* method), 125

inv_ROC_AR (*gtrace.optcomp.Mirror* attribute), 150

inv_ROC_HR (*gtrace.optcomp.Mirror* attribute), 149

InvROCandW2q() (in module *gtrace.optics.gaussian*), 129

isHit() (*gtrace.optcomp.CyMirror* method), 153

isHit() (*gtrace.optcomp.Mirror* method), 150

isHit() (*gtrace.optcomp.Optics* method), 147

K

kHz (in module *gtrace.optics.unit*), 135

kHz (in module *gtrace.unit*), 157

km (in module *gtrace.optics.unit*), 135

km (in module *gtrace.unit*), 156

kW (in module *gtrace.optics.unit*), 135

kW (in module *gtrace.unit*), 156

L

L (*gtrace.optics.cavity.Cavity* attribute), 123, 124

Layer (class in *gtrace.draw*), 121

Layer (class in *gtrace.draw.draw*), 22

Layer (class in *gtrace.draw.dxf*), 25

layer (*gtrace.beam.GaussianBeam* attribute), 138, 140

layer_template1 (in module *gtrace.draw.dxf*), 55

layer_template2 (in module *gtrace.draw.dxf*), 56

length (*gtrace.beam.GaussianBeam* attribute), 138, 140

Line (class in *gtrace.draw*), 122

Line (class in *gtrace.draw.draw*), 22

Line (class in *gtrace.draw.dxf*), 25

line_arc_intersection() (in module *gtrace.optics.geometric*), 132

line_plane_intersection() (in module *gtrace.optics.geometric*), 132

linetype_template (in module *gtrace.draw.dxf*), 50

LwPolyLine (class in *gtrace.draw.dxf*), 25

M

m (in module *gtrace.optics.unit*), 135

m (in module *gtrace.unit*), 156

MHz (in module *gtrace.optics.unit*), 135

mHz (in module *gtrace.optics.unit*), 135

MHz (in module *gtrace.unit*), 156

mHz (in module *gtrace.unit*), 157

Mirror (class in *gtrace.optcomp*), 148

misctable_template (in module *gtrace.draw.dxf*), 57

mm (in module *gtrace.optics.unit*), 135

mm (in module *gtrace.unit*), 156

modeMatching() (in module *gtrace.optics.gaussian*), 129

modeMatchingElliptic() (in module *gtrace.optics.gaussian*), 129

modeSpacing() (*gtrace.optics.cavity.Cavity* method), 125

modeSpacing() (in module *gtrace.optics.gaussian*), 128

module

gtrace, 21

gtrace.beam, 136

gtrace.draw, 21

gtrace.draw.draw, 21

gtrace.draw.dxf, 23

gtrace.draw.renderer, 118

gtrace.draw.tools, 119

gtrace.nonsequential, 144

gtrace.optcomp, 145

gtrace.optics, 122

gtrace.optics.cavity, 122

gtrace.optics.consts, 126

gtrace.optics.gaussian, 127

gtrace.optics.geometric, 131

gtrace.optics.unit, 134

gtrace.unit, 155

mW (in module *gtrace.optics.unit*), 135

mW (in module *gtrace.unit*), 156

Mx (*gtrace.beam.GaussianBeam* attribute), 139, 140

My (*gtrace.beam.GaussianBeam* attribute), 139, 140

N

n (*gtrace.beam.GaussianBeam* attribute), 138, 140

n (*gtrace.optcomp.Mirror* attribute), 149

n_fused_silica() (in module *gtrace.optics.consts*), 127

n_fused_silica_1064nm (in module *gtrace.optics.consts*), 127

n_fused_silica_1550nm (in module *gtrace.optics.consts*), 127

n_fused_silica_532nm (in module *gtrace.optics.consts*), 127

n_sapphire_extraordinary() (in module *gtrace.optics.consts*), 127

n_sapphire_ordinary() (in module *gtrace.optics.consts*), 127

name (*gtrace.beam.GaussianBeam* attribute), 138, 139

name (*gtrace.optcomp.Optics* attribute), 147

Nbounce() (*gtrace.optics.cavity.Cavity* method), 124

ne_sapphire_1064nm (in module *gtrace.optics.consts*), 127

- ne_sapphire_1550nm (in module *gtrace.optics.consts*), 127
 ne_sapphire_532nm (in module *gtrace.optics.consts*), 127
 nm (in module *gtrace.optics.unit*), 135
 nm (in module *gtrace.unit*), 156
 no_sapphire_1064nm (in module *gtrace.optics.consts*), 127
 no_sapphire_1550nm (in module *gtrace.optics.consts*), 127
 no_sapphire_532nm (in module *gtrace.optics.consts*), 127
 non_seq_trace() (in module *gtrace.nonsequential*), 145
 normAngleAR (*gtrace.optcomp.CyMirror* attribute), 153
 normAngleAR (*gtrace.optcomp.Mirror* attribute), 149
 normAngleHR (*gtrace.optcomp.CyMirror* attribute), 152
 normAngleHR (*gtrace.optcomp.Mirror* attribute), 148, 149
 normSpheric() (in module *gtrace.optics.geometric*), 133
 normVectAR (*gtrace.optcomp.CyMirror* attribute), 153
 normVectAR (*gtrace.optcomp.Mirror* attribute), 149
 normVectHR (*gtrace.optcomp.CyMirror* attribute), 152
 normVectHR (*gtrace.optcomp.Mirror* attribute), 148, 149
 NumberOfElementError, 23, 25, 122
- ## O
- objects_template (in module *gtrace.draw.dxf*), 64
 optDist (*gtrace.beam.GaussianBeam* attribute), 139, 140
 optFunForEndPointR() (in module *gtrace.beam*), 143
 optFunForFlat() (in module *gtrace.beam*), 143
 optFunForStartPointR() (in module *gtrace.beam*), 142
 Optics (class in *gtrace.optcomp*), 147
 optimalMatching() (in module *gtrace.optics.gaussian*), 130
 optimCrossPointFlat() (in module *gtrace.beam*), 144
 optimEndPointR() (in module *gtrace.beam*), 143
 optimStartPointR() (in module *gtrace.beam*), 142
- ## P
- P (*gtrace.beam.GaussianBeam* attribute), 138, 140
 percent (in module *gtrace.unit*), 157
 pi (in module *gtrace.beam*), 137
 pi (in module *gtrace.draw*), 121
 pi (in module *gtrace.draw.draw*), 22
 pi (in module *gtrace.draw.dxf*), 25
 pi (in module *gtrace.optcomp*), 146
 pi (in module *gtrace.optics.cavity*), 123
 pi (in module *gtrace.optics.gaussian*), 128
 pi (in module *gtrace.optics.geometric*), 131
 pi (in module *gtrace.optics.unit*), 135
 pi (in module *gtrace.unit*), 156
 pole() (*gtrace.optics.cavity.Cavity* method), 124
 PolyLine (class in *gtrace.draw*), 122
 PolyLine (class in *gtrace.draw.draw*), 23
 pos (*gtrace.beam.GaussianBeam* attribute), 138, 140
 powerGain() (*gtrace.optics.cavity.Cavity* method), 124
 ppm (in module *gtrace.optics.unit*), 136
 ppm (in module *gtrace.unit*), 157
 propagate() (*gtrace.beam.GaussianBeam* method), 140
- ## Q
- q (*gtrace.beam.GaussianBeam* attribute), 138, 140
 q2R() (in module *gtrace.optics.gaussian*), 128
 q2w() (in module *gtrace.optics.gaussian*), 128
 q2zr() (in module *gtrace.optics.gaussian*), 128
 qrx (*gtrace.beam.GaussianBeam* attribute), 140
 qry (*gtrace.beam.GaussianBeam* attribute), 140
 qToRadius() (in module *gtrace.optics.gaussian*), 130
 qToROC() (in module *gtrace.optics.gaussian*), 130
 qx (*gtrace.beam.GaussianBeam* attribute), 138, 140
 qy (*gtrace.beam.GaussianBeam* attribute), 138, 140
- ## R
- R() (*gtrace.beam.GaussianBeam* method), 141
 R1 (*gtrace.optics.cavity.Cavity* attribute), 123, 124
 r1 (*gtrace.optics.cavity.Cavity* attribute), 123, 124
 R2 (*gtrace.optics.cavity.Cavity* attribute), 124
 r2 (*gtrace.optics.cavity.Cavity* attribute), 123, 124
 rad2deg() (in module *gtrace.optics.unit*), 136
 rad2deg() (in module *gtrace.unit*), 157
 Rectangle (class in *gtrace.draw*), 122
 Rectangle (class in *gtrace.draw.draw*), 23
 Rectangle (class in *gtrace.draw.dxf*), 26
 refl() (*gtrace.optics.cavity.Cavity* method), 125
 Refl_AR (*gtrace.optcomp.Mirror* attribute), 150
 refl_defl_angle() (in module *gtrace.optics.geometric*), 133
 Refl_HR (*gtrace.optcomp.Mirror* attribute), 150
 renderDXF() (in module *gtrace.draw.renderer*), 118
 report_min_max() (*gtrace.draw.dxf.Arc* method), 26
 report_min_max() (*gtrace.draw.dxf.Circle* method), 26
 report_min_max() (*gtrace.draw.dxf.Entity* method), 25
 report_min_max() (*gtrace.draw.dxf.Line* method), 25

`report_min_max()` (*gtrace.draw.dxf.LwPolyLine method*), 26
`report_min_max()` (*gtrace.draw.dxf.Rectangle method*), 26
`report_min_max()` (*gtrace.draw.dxf.Text method*), 26
`ROCandWtoQ()` (*in module gtrace.optics.gaussian*), 130
`rotate()` (*gtrace.beam.GaussianBeam method*), 140
`rotate()` (*gtrace.optcomp.Mirror method*), 150
`rotateAll()` (*in module gtrace.draw.tools*), 120
`rotationAngle` (*gtrace.optcomp.Optics attribute*), 147
`rp1` (*gtrace.optics.cavity.Cavity attribute*), 123, 124
`rp2` (*gtrace.optics.cavity.Cavity attribute*), 123, 124
`Rw2q()` (*in module gtrace.optics.gaussian*), 128

S

`sagAR` (*gtrace.optcomp.Mirror attribute*), 149
`sagHR` (*gtrace.optcomp.Mirror attribute*), 149
`save_to_file()` (*gtrace.draw.dxf.DXF method*), 25
`sellmeier()` (*in module gtrace.optics.consts*), 126
`set_handle()` (*gtrace.draw.dxf.Entity method*), 25
`Shape` (*class in gtrace.draw*), 121
`Shape` (*class in gtrace.draw.draw*), 22
`spotSize()` (*gtrace.optics.cavity.Cavity method*), 125
`sqrt` (*in module gtrace.beam*), 137
`sqrt` (*in module gtrace.optcomp*), 146
`sqrt` (*in module gtrace.optics.cavity*), 123
`sqrt` (*in module gtrace.optics.gaussian*), 128
`storageTime()` (*gtrace.optics.cavity.Cavity method*), 124
`stray_order` (*gtrace.beam.GaussianBeam attribute*), 139
`style_template` (*in module gtrace.draw.dxf*), 56

T

`t1` (*gtrace.optics.cavity.Cavity attribute*), 124
`t2` (*gtrace.optics.cavity.Cavity attribute*), 124
`tables_template1` (*in module gtrace.draw.dxf*), 47
`tables_template2` (*in module gtrace.draw.dxf*), 61
`term_on_HR` (*gtrace.optcomp.CyMirror attribute*), 153
`term_on_HR` (*gtrace.optcomp.Mirror attribute*), 149
`term_on_HR_order` (*gtrace.optcomp.CyMirror attribute*), 153
`term_on_HR_order` (*gtrace.optcomp.Mirror attribute*), 149
`test_func()` (*in module gtrace.draw.dxf*), 117
`Text` (*class in gtrace.draw*), 122
`Text` (*class in gtrace.draw.draw*), 23
`Text` (*class in gtrace.draw.dxf*), 26
`thickness` (*gtrace.optcomp.Mirror attribute*), 149
`THz` (*in module gtrace.optics.unit*), 135
`THz` (*in module gtrace.unit*), 156
`tp1` (*gtrace.optics.cavity.Cavity attribute*), 124
`tp2` (*gtrace.optics.cavity.Cavity attribute*), 124
`trans()` (*gtrace.optics.cavity.Cavity method*), 125
`Trans_AR` (*gtrace.optcomp.Mirror attribute*), 150
`Trans_HR` (*gtrace.optcomp.Mirror attribute*), 150
`transAll()` (*in module gtrace.draw.tools*), 120
`translate()` (*gtrace.beam.GaussianBeam method*), 141
`translate()` (*gtrace.optcomp.Mirror method*), 150

U

`uHz` (*in module gtrace.optics.unit*), 136
`uHz` (*in module gtrace.unit*), 157
`um` (*in module gtrace.optics.unit*), 135
`um` (*in module gtrace.unit*), 156
`UnknownShapeError`, 118
`uW` (*in module gtrace.optics.unit*), 135
`uW` (*in module gtrace.unit*), 156

V

`vc_deflect()` (*in module gtrace.optics.geometric*), 134
`vc_reflect()` (*in module gtrace.optics.geometric*), 134
`vector_normalize()` (*in module gtrace.optics.geometric*), 132
`vector_rotation_2D()` (*in module gtrace.optics.geometric*), 132
`view_template` (*in module gtrace.draw.dxf*), 57
`vport_template` (*in module gtrace.draw.dxf*), 47

W

`W` (*in module gtrace.optics.unit*), 135
`W` (*in module gtrace.unit*), 156
`w02zr()` (*in module gtrace.optics.gaussian*), 129
`waist()` (*gtrace.beam.GaussianBeam method*), 141
`waist()` (*gtrace.optics.cavity.Cavity method*), 125
`wedgeAngle` (*gtrace.optcomp.Mirror attribute*), 149
`width()` (*gtrace.beam.GaussianBeam method*), 141
`wl` (*gtrace.beam.GaussianBeam attribute*), 138, 140
`wl` (*gtrace.optics.cavity.Cavity attribute*), 124
`wx` (*gtrace.beam.GaussianBeam attribute*), 138, 140
`wy` (*gtrace.beam.GaussianBeam attribute*), 139, 140

Z

`zr2w0()` (*in module gtrace.optics.gaussian*), 129