**SnagPy User Guide**



[Introduction 4](#_Toc120815146)

[**GD module** 5](#_Toc120815147)

[**Basic** 5](#_Toc120815148)

[**class gd** 5](#_Toc120815149)

[**function edit\_gd** 5](#_Toc120815150)

[**function x\_gd** 5](#_Toc120815151)

[**function gd2dict** 6](#_Toc120815152)

[**function dict2gd** 6](#_Toc120815153)

[**function show\_gd** 6](#_Toc120815154)

[**Set functions** 7](#_Toc120815155)

[**set\_gd(ingd,fun,par1=1,par2=0.1,par3=0):** 7](#_Toc120815156)

[**fun == 'delt': # par1 Amp, par2 position** 7](#_Toc120815157)

[**fun == 'step': # par1 Amp, par2 position** 7](#_Toc120815158)

[**fun == 'ramp': # par1 first value, par2 step (positive or negative)** 7](#_Toc120815159)

[**fun == 'cos': # par1 Amp, par2 fr, par3 ph** 7](#_Toc120815160)

[**fun == 'sin': # par1 Amp, par2 fr, par3 ph** 7](#_Toc120815161)

[**fun == 'cexp': # par1 Amp, par2 fr, par3 ph** 7](#_Toc120815162)

[**fun == 'exp': # par1 Amp, par2 tau** 7](#_Toc120815163)

[**fun == 'power': # par1 Amp, par2 exponent, par3 scale** 7](#_Toc120815164)

[**fun == 'rect': # par1 Amp, par2 length on, par3 length off** 7](#_Toc120815165)

[**rand\_gd(ingd,dist,par1=0,par2=1,par3=0)** 8](#_Toc120815166)

[**dist == 'unif': # par1 min, par2 max, par3 seed** 8](#_Toc120815167)

[**dist == 'norm': # par1 mu, par2 sigma, par3 seed** 8](#_Toc120815168)

[**dist == 'cauchy': # par1 mu, par2 scale, par3 seed** 8](#_Toc120815169)

[**Modification functions** 9](#_Toc120815170)

[**modif\_gd(ingd,fun,par1=1,par2=0.1,par3=0)** 9](#_Toc120815171)

[**fun == 'abs'** 9](#_Toc120815172)

[**fun == 'real'** 9](#_Toc120815173)

[**fun == 'imag'** 9](#_Toc120815174)

[**fun == 'log10'** 9](#_Toc120815175)

[**fun == 'xlog10'** 9](#_Toc120815176)

[**fun == 'loglog10'** 9](#_Toc120815177)

[**rota\_gd(ingd,n)** 9](#_Toc120815178)

[**Simple functions** 10](#_Toc120815179)

[**stat\_gd(ingd,nbins=20)** 10](#_Toc120815180)

[**fft\_gd(ingd,fif=1)** 10](#_Toc120815181)

[**Plot functions** 11](#_Toc120815182)

[**plot\_gd(ingd)** 11](#_Toc120815183)

[**semilogx\_gd(ingd)** 11](#_Toc120815184)

[**semilogy\_gd(ingd)** 11](#_Toc120815185)

[**loglog\_gd(ingd)** 11](#_Toc120815186)

[**close\_fig()** 11](#_Toc120815187)

[**gridon()** 11](#_Toc120815188)

[**xlog()** 11](#_Toc120815189)

[**ylog()** 11](#_Toc120815190)

[**xlin()** 11](#_Toc120815191)

[**ylin()** 11](#_Toc120815192)

[**Calc functions** 12](#_Toc120815193)

[**minmax\_gd(ingd)** 12](#_Toc120815194)

[**GD2 module** 13](#_Toc120815195)

[**STAT module** 14](#_Toc120815196)

[**SERV module** 15](#_Toc120815197)

[**GDPROC module** 16](#_Toc120815198)

[**IMAGE module** 17](#_Toc120815199)

[**GUISNAG module** 18](#_Toc120815200)

[**ML\_PY module** 19](#_Toc120815201)

[**ASTROTIM module** 20](#_Toc120815202)

[**GWDATA module** 21](#_Toc120815203)

[**PSS module** 22](#_Toc120815204)

[**BSD module** 23](#_Toc120815205)

[**PARGPU module** 24](#_Toc120815206)

# Introduction

SnagPy is a Python package for data analysis mainly in the gravitational signal field. It is a quasi-porting of the Snag Matlab toolbox developed by the Virgo Rome Data Analysis group.

To start:

* Put the files contained in the zip where you want
* Set the PYTHONPATH in the environment variables (how to do this depends on the operative system. See for example [How to set python path (net-informations.com)](http://net-informations.com/python/intro/path.htm) or anything similar googling ‘’set path python”).

# **GD module**

It imports:

mport numpy as np

import scipy

import cmath as cm

import matplotlib.pyplot as plt

import matplotlib as mpl

import copy

## **Basic**

### **class gd**

To create a gd object. Any gd can be modified by edit\_gd. A gd can be created giving just the number of samples, or transform a 1-dimensional array to a gd, or give the other information. The attributes are:

* y the “ordinate” (basic data)
* n the length
* ini initial abscissa (used in type 1 gds)
* dx sampling step (used in type 1 gds)
* x abscissas (used in type 2 gds)
* typ determine type 1 (virtual abscissa) or type 2 (real abscissa)
* capt caption (a string)
* cont a control variable (in the case of a bsd, a special structure)
* unc ordinate uncertainty (typically not used)
* uncx abscissa uncertainty (used for other)

### **function edit\_gd**

To change gd attributes.

### **function x\_gd**

Extract abscissa, in both real or virtual case.

### **function gd2dict**

Creates a dictionary from a gd.

### **function dict2gd**

Creates a gd from a dictionary.

### **function show\_gd**

Shows gd’s attributes.

## **Set functions**

### **set\_gd(ingd,fun,par1=1,par2=0.1,par3=0):**

#### **fun == 'delt': # par1 Amp, par2 position**

#### **fun == 'step': # par1 Amp, par2 position**

#### **fun == 'ramp': # par1 first value, par2 step (positive or negative)**

#### **fun == 'cos': # par1 Amp, par2 fr, par3 ph**

#### **fun == 'sin': # par1 Amp, par2 fr, par3 ph**

#### **fun == 'cexp': # par1 Amp, par2 fr, par3 ph**

#### **fun == 'exp': # par1 Amp, par2 tau**

#### **fun == 'power': # par1 Amp, par2 exponent, par3 scale**

#### **fun == 'rect': # par1 Amp, par2 length on, par3 length off**

**# if Amp < 0, begins witn off**

### **rand\_gd(ingd,dist,par1=0,par2=1,par3=0)**

#### **dist == 'unif': # par1 min, par2 max, par3 seed**

#### **dist == 'norm': # par1 mu, par2 sigma, par3 seed**

#### **dist == 'cauchy': # par1 mu, par2 scale, par3 seed**

## **Modification functions**

### **modif\_gd(ingd,fun,par1=1,par2=0.1,par3=0)**

#### **fun == 'abs'**

#### **fun == 'real'**

#### **fun == 'imag'**

#### **fun == 'log10'**

#### **fun == 'xlog10'**

#### **fun == 'loglog10'**

### **rota\_gd(ingd,n)**

## **Simple functions**

### **stat\_gd(ingd,nbins=20)**

### **fft\_gd(ingd,fif=1)**

## **Plot functions**

### **plot\_gd(ingd)**

### **semilogx\_gd(ingd)**

### **semilogy\_gd(ingd)**

### **loglog\_gd(ingd)**

### **close\_fig()**

### **gridon()**

### **xlog()**

### **ylog()**

### **xlin()**

### **ylin()**

## **Calc functions**

### **minmax\_gd(ingd)**

# **GD2 module**

# **STAT module**

# **SERV module**

# **GDPROC module**

# **IMAGE module**

# **GUISNAG module**

# **ML\_PY module**

This module deals with operations with Matlab. The most important part is the management of the mat files. Some mat files could not be read because contain Matlab tables: in such case the solution is saving the Matlab f\table as a csv file (by the writetable internal function or table2cvs Snag function).

# **ASTROTIM module**

# **GWDATA module**

# **PSS module**

# **BSD module**

# **PARGPU module**