**SnagPy User Guide**



[Introduction 4](#_Toc119449732)

[**GD module** 5](#_Toc119449733)

[**Basic** 5](#_Toc119449734)

[**class gd** 5](#_Toc119449735)

[**function edit\_gd** 5](#_Toc119449736)

[**function x\_gd** 5](#_Toc119449737)

[**Set functions** 6](#_Toc119449738)

[**set\_gd(ingd,fun,par1=1,par2=0.1,par3=0):** 6](#_Toc119449739)

[**fun == 'delt': # par1 Amp, par2 position** 6](#_Toc119449740)

[**fun == 'step': # par1 Amp, par2 position** 6](#_Toc119449741)

[**fun == 'ramp': # par1 first value, par2 step (positive or negative)** 6](#_Toc119449742)

[**fun == 'cos': # par1 Amp, par2 fr, par3 ph** 6](#_Toc119449743)

[**fun == 'sin': # par1 Amp, par2 fr, par3 ph** 6](#_Toc119449744)

[**fun == 'cexp': # par1 Amp, par2 fr, par3 ph** 6](#_Toc119449745)

[**fun == 'exp': # par1 Amp, par2 tau** 6](#_Toc119449746)

[**fun == 'power': # par1 Amp, par2 exponent, par3 scale** 6](#_Toc119449747)

[**fun == 'rect': # par1 Amp, par2 length on, par3 length off** 6](#_Toc119449748)

[**rand\_gd(ingd,dist,par1=0,par2=1,par3=0)** 7](#_Toc119449749)

[**dist == 'unif': # par1 min, par2 max, par3 seed** 7](#_Toc119449750)

[**dist == 'norm': # par1 mu, par2 sigma, par3 seed** 7](#_Toc119449751)

[**dist == 'cauchy': # par1 mu, par2 scale, par3 seed** 7](#_Toc119449752)

[**Modification functions** 8](#_Toc119449753)

[**modif\_gd(ingd,fun,par1=1,par2=0.1,par3=0)** 8](#_Toc119449754)

[**fun == 'abs'** 8](#_Toc119449755)

[**fun == 'real'** 8](#_Toc119449756)

[**fun == 'imag'** 8](#_Toc119449757)

[**fun == 'log10'** 8](#_Toc119449758)

[**fun == 'xlog10'** 8](#_Toc119449759)

[**fun == 'loglog10'** 8](#_Toc119449760)

[**rota\_gd(ingd,n)** 8](#_Toc119449761)

[**Simple functions** 9](#_Toc119449762)

[**stat\_gd(ingd,nbins=20)** 9](#_Toc119449763)

[**fft\_gd(ingd,fif=1)** 9](#_Toc119449764)

[**Plot functions** 10](#_Toc119449765)

[**plot\_gd(ingd)** 10](#_Toc119449766)

[**semilogx\_gd(ingd)** 10](#_Toc119449767)

[**semilogy\_gd(ingd)** 10](#_Toc119449768)

[**loglog\_gd(ingd)** 10](#_Toc119449769)

[**close\_fig()** 10](#_Toc119449770)

[**gridon()** 10](#_Toc119449771)

[**xlog()** 10](#_Toc119449772)

[**ylog()** 10](#_Toc119449773)

[**xlin()** 10](#_Toc119449774)

[**ylin()** 10](#_Toc119449775)

[**Calc functions** 11](#_Toc119449776)

[**minmax\_gd(ingd)** 11](#_Toc119449777)

[**GD2 module** 12](#_Toc119449778)

[**STAT module** 13](#_Toc119449779)

[**SERV module** 14](#_Toc119449780)

[**GDPROC module** 15](#_Toc119449781)

[**IMAGE module** 16](#_Toc119449782)

[**GUISNAG module** 17](#_Toc119449783)

[**ML\_PY module** 18](#_Toc119449784)

[**ASTROTIM module** 19](#_Toc119449785)

[**GWDATA module** 20](#_Toc119449786)

[**PSS module** 21](#_Toc119449787)

[**BSD module** 22](#_Toc119449788)

[**PARGPU module** 23](#_Toc119449789)

# 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**

### **function edit\_gd**

### **function x\_gd**

## **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**