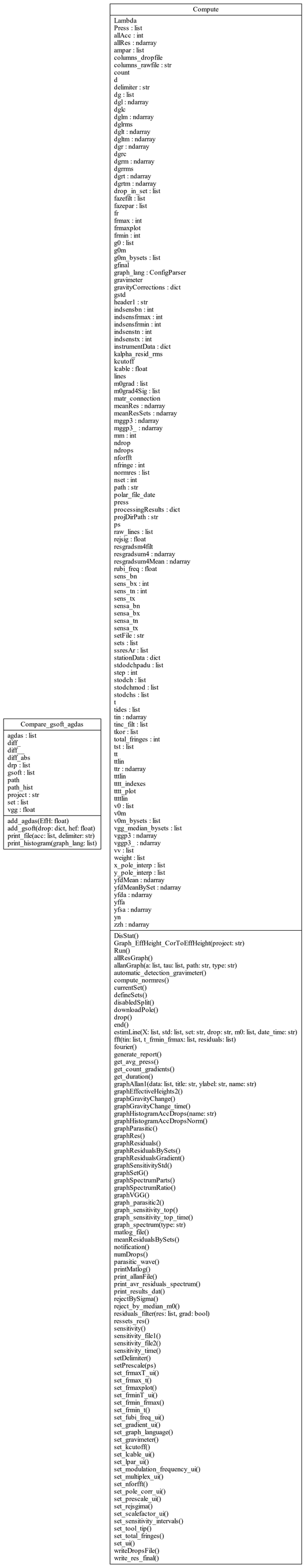
Compute
Lambda
Press : list
allAcc : int
allRes : ndarray
ampar : list
columns_dropfile
columns_rawfile : str
count
d
delimiter : str
dg : list
dgl : ndarray
dglc
dglm : ndarray
dglrms
dglt : ndarray
dgltm : ndarray
dgr : ndarray
dgrc
dgrm : ndarray
dgrrms
dgrt : ndarray
dgrtm : ndarray
drop_in_set : list
fazefilt : list
fazepar : list
fr
frmax : int
frmaxplot
frmin : int
g0 : list
g0m
g0m_bysets : list
gfinal
graph_lang : ConfigParser
gravimeter
gravityCorrections : dict
gstd
header1 : str
indsensbn : int
indsensfrmax : int
indsensfrmin : int
indsenstn : int
indsenstx : int
instrumentData : dict
kalpha_resid_rms
kcutoff
lcable : float
lines
m0grad : list
m0grad4Sig : list
matr_connection
meanRes : ndarray
meanResSets : ndarray
mggp3 : ndarray
mggp3_ : ndarray
mm : int
ndrop
ndrops
nforfft
nfringe : int
normres : list
nset : int
path : str
polar_file_date
press
processingResults : dict
projDirPath : str
ps
raw_lines : list
rejsig : float
resgradsm4filt
resgradsum4 : ndarray
resgradsum4Mean : ndarray
rubi_freq : float
sens_bn
sens_bx : int
sens_tn : int
sens_tx
sensa_bn
sensa_bx
sensa_tn
sensa_tx
setFile : str
sets : list
ssresAr : list
stationData : dict
stdodchpadu : list
step : int
stodch : list
stodchmod : list
stodchs : list
t
tides : list
tin : ndarray
tinc_filt : list
tkor : list
total_fringes : int
tst : list
tt
ttlin
ttr : ndarray
tttlin
tttt_indexes
tttt_plot
ttttlin
v0 : list
v0m
v0m_bysets : list
vgg_median_bysets : list
vggp3 : ndarray
vggp3_ : ndarray
vv : list
weight : list
x_pole_interp : list
y_pole_interp : list
yfdMean : ndarray
yfdMeanBySet : ndarray
yfda : ndarray
yffa
yfsa : ndarray
yn
zzh : ndarray
DisStat()
Graph_EffHeight_CorToEffHeight(project: str)
Run()
allResGraph()
allanGraph(a: list, tau: list, path: str, type: str)
automatic_detection_gravimeter()
compute_normres()
currentSet()
defineSets()
disabledSplit()
downloadPole()
drop()
end()
estimLine(X: list, std: list, set: str, drop: str, m0: list, date_time: str)
fft(tin: list, t_frmin_frmax: list, residuals: list)
fourier()
generate_report()
get_avg_press()
get_count_gradients()
get_duration()
graphAllan1(data: list, title: str, ylabel: str, name: str)
graphEffectiveHeights2()
graphGravityChange()
graphGravityChange_time()
graphHistogramAccDrops(name: str)
graphHistogramAccDropsNorm()
graphParasitic()
graphRes()
graphResiduals()
graphResidualsBySets()
graphResidualsGradient()
graphSensitivityStd()
graphSetG()
graphSpectrumParts()
graphSpectrumRatio()
graphVGG()
graph_parasitic2()
graph_sensitivity_top()
graph_sensitivity_top_time()
graph_spectrum(type: str)
matlog_file()
meanResidualsBySets()
notification()
numDrops()
parasitic_wave()
printMatlog()
print_allanFile()
print_avr_residuals_spectrum()
print_results_dat()
rejectBySigma()
reject_by_median_m0()
residuals_filter(res: list, grad: bool)
ressets_res()
sensitivity()
sensitivity_file1()
sensitivity_file2()
sensitivity_time()
setDelimiter()
setPrescale(ps)
set_frmaxT_ui()
set_frmax_t()
set_frmaxplot()
set_frminT_ui()
set_frmin_frmax()
set_frmin_t()
set_fubi_freq_ui()
set_gradient_ui()
set_graph_language()
set_gravimeter()
set_kcutoff()
set_lcable_ui()
set_lpar_ui()
set_modulation_frequency_ui()
set_multiplex_ui()
set_nforffft()
set_pole_corr_ui()
set_prescale_ui()
set_rejsgima()
set_scalefactor_ui()
set_sensitivity_intervals()
set_tool_tip()
set_total_fringes()
set_ui()
writeDropsFile()
write_res_final()
Compare_gsoft_agdas
agdas : list
diff_
diff__
diff_abs
drp : list
gsoft : list
path
path_hist
project : str
set : list
vgg : float
add_agdas(EfH: float)
add_gsoft(drop: dict, hef: float)
print_file(acc: list, delimiter: str)
print_histogram(graph_lang: list)
Fall
Acable : float
Grad
Lambda : float
Lcable : float
Lpar : float
Pcable
Resgrad4
c : float
effectiveZ
fmod : float
fringe : float64
frmax : int
frmin
g0
g0_Gr
gTop
gTopCor
gradient
h
htop
kdis : bool
keys : list
kimp : bool
kpar : bool
ksae : bool
ksol : float
ksol_k : bool
m02
m02_grad
m0grad4
m0gradient
multiplex : float
res_grad1
resgrad4
rubiFreq : float
scaleFactor : float
ssres
std : ndarray
stdGradX
stdX
std_grad
stdstd : ndarray
tt : ndarray
v0
x : tuple
x_grad : tuple
xef : tuple
xgrad4 : tuple
z0
LST()
checkKDIS()
checkKIMP()
checkKSAE()
computeLST(A: np.ndarray, z: np.ndarray, frmin: int, frmax: int)
effectiveHeight()
effectiveHeightTop()
effectivePosition()
gTop()
gTopCor(tide, load, baro, polar)
setAcable(Acable: float)
setFrRange(frmin: int, frmax: int)
setFringe(times: list)
setGradient(grad: str)
setKpar(kpar: bool)
setLambda(Lambda: str)
setLcable(Lcable: float)
setLpar(Lpar: float)
setModulFreq(fmod: float)
setMultiplex(multiplex: str)
setPcable(Pcable)
setRubiFreq(freq: str)
setScaleFactor(scaleFactor: str)
set_ksol(ksol: float)
set_ksol_k(ksol_k: bool)
Graph
columns_name : list
gr
hist_data : list
name : str
path : str
project : str
title : str
x : list
x_err : list
x_label : str
y : list
y_err : list
y_label : str
yerr : list
decimal(decimal_number: int)
error_bar(x_err: list, y_err: list, yerr: list, color_err: str, ms, capsize)
histogram(hist_data: list, fit: bool)
plotXY(x: list, y: list, mark: list, columns_name, legend, lw)
save()
text(x: list, y: list, t: list, c: list)
x_ax_int()
ylim(lim: list)
Install
Main
newProjectWin
result
Computing()
closeproject()
finals_folder()
last_version()
new_Project()
set_info()
splashScreen()
viewData()
NewProject
gravityCorrections
header1
header2
instrumentData
names
path
pathDir
processingResults
projectfile
rawfile
rawfilepath
rawlines
setFile
stationData
succ : bool
units
accept()
load_files()
load_test_data()
saveDir()
Sumarize
gravityCorrections : dict
instrumentData : dict
names : dict
processingResults : dict
rawheader : str
stationData : dict
units : dict
accept()
load()
string(dictionary: dict)
Warning
dropFile
drop_lines
dropfile
dropHeader4()
dropLines()
read()
estim
f
printResult(X, std, set, drop, m0)
matr_db
cursor
matr_db
close()
commit()
get(query: str)
insert(query: str)
projectFile
file : list
file_lines
names : list
names_summary : dict
projectfile
units : dict
createDictionary()
insert_to_names(dict: dict, i: str)
insert_to_units(dict, i)
read()
rawFile
raw_lines
rawfile : str
rawHeader1()
rawHeader2()
rawLines()
read()
res_final
delimiter : str
f
close()
printResult(line: list)
write_line(line: str)