API documentation

Generated by Doxygen 1.9.4

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

Module Documentation

2.1 MG-MAMPOSSt main ingredients

Functions/Subroutines

- subroutine mamposst (di, ve, eve, rso, vso, npg1)
- function sigmar1 (tt)
- function sigmar2 (tt)
- function sigmar3 (tt)
- function sr2int (alr)
- function sr2out (t)
- subroutine vmaxlik (nfv, xfv, f)
- function fa (tlog)

2.1.1 Detailed Description

List of the main functions and subroutines needed for the execution of MG-MAMPOSSt. This block includes all the fundamental routines and functions on which the MG-MAMPOSSt method is based. The central core is represented by the mamposst() subroutine, which computes the tabulated $-\ln \mathcal{L}$.

All the routines in this block need external inputs as described below. In particular, the COMMON block of parameters "paramsoptS.i" should always be included.

2.1.2 Function/Subroutine Documentation

2.1.2.1 fa()

```
function fa (
              tlog )
```

Authors

A. Biviano, G. Mamon, G. Boue', L. Pizzuti

Integrand function for the determination of $N(R) * \sigma^2_{LOS}(R)$, given beta(r), M(r) and

It requires the inclusion of the COMMON parameters in `paramsoptS.i`, the values of all the paramters `r200, rs, rc, cbe, cbe0, tmass, screen, rcut, al, kani, kmp, knfit, kscr, v200=10.*hz*r200`. if `kmp=7` and `kscr>0`, one needs to specify also nhs, the exponent for the Hu&Sawicki model of $f(R)$, and the redshift parameter zb. A special case corresponds to the number of steps in the anisotropy parameter nbs=-1, where the Hansen & Moore model is selected (the anisotropy depends on the mass density profile $\beta(r) = a + bd\rho(r)/d\ln r$). In this case, the parameter `ahm=a` and `bhm=b` should be specified externally. NOTE: this function cannot be used as it is, unless a previous computation of spline coefficients for the radial velocity dispersion profile $\sigma^2_r(r)$ is done. Indeed fa(tlog) include a cubic spline interpolation of $\sigma^2_r(r)$. The spline is computed by using the routines by John Burkardt, `spline_cubic_set` and `spline_cubic_val` (see `spline_cubic.f90` in folder Gaml for source codes, credits and usage).

Parameters

in	*tlog*	natural logarithm of the projected radial distance at which the profile is computed.

Example of usage:

```
 !compute spline coeffs for later interpolation of sigma_r
 call spline_cubic_set(ninterp,xris,yris,2,0.d0,2,0.d0,ypp2)
 !xris(ninterp) and yris(ninterp) are global variables storing the
 !log-radii and the values of sigma^2_r computed before.
 nbs=1
!  Values of the parameters of Hansen & Moore's (2006)
!     beta = a + b dln(rho)/dln(r) relation
 ahm=-0.15
 bhm=-0.192
 r200=1.41
 v200=r200*10*85
 rc=0.3
 rs=0.3
 cbe=1.2
 cbe0=1.0  !not used unless kani=21,41
 tmass=1.0
 screen=0.4
 kmp=7     !mass profile model. kmp=7 corresponds to mNFW_LH
 knfit=1   !number density model. knfit=1 corresponds to pNFW
 kani=4    !anisotropy model. kani=4 corresponds to Tiret profile
 kscr=1
 rcut=1.0
 zb=0.3
 nhs=2

 tlog=dlog(0.5d0)
 svdp=fa(tlog)
 write(*,*) svdp
```

2.1.2.2 mamposst()

```
subroutine mamposst (
              dimension(npg1) di,
```

```
            dimension(npg1) ve,
            dimension(npg1) eve,
            dimension(npg1) rso,
            dimension(npg1) vso,
             npg1 )
```

Authors

A. Biviano, G. Mamon, G. Boue', L. Pizzuti

MAMPOSSt subroutine computes the Log-likelihood in the projected phase space. This is the core of the MG-MAMPOSSt code. Along with the input data, mamposst() requires as external input the values of all the input parameters/Options of MG-MAMPOSSt along with the file names. In particular:

- name and id number of the output likelihood file: `iu60`
- name and id number of the output file of projected position and velocity with the associated probability p(R_i,v_i): `iu20`
- name and id number of the output file containing the best fit of projected number density profile: `iu30`
- input parameters in `test_pars.txt` described in the documentation and named as in the COMMON block `paramsoptS.i`
- The `Options.txt` file, read within the subroutine

In the case of `Nclust>1` in `Options.txt`, the mamposst() subroutine requires the additional data-files: `data/datphys_<i>.dat`, where "i" is an integer number larger than 1.

Parameters

in	*di(npg1)*	dimension REAL∗8 projected distance in Mpc
in	*ve(npg1)*	dimension REAL∗8 line-of-sight velocity [km/s]
in	*eve(npg1)*	dimension REAL∗8 velocity error [km/s]
in	*npg1*	INTEGER∗4 number of galaxies in the phase space
out	*rso(npg1)*	dimension REAL∗8 projected distance in Mpc of the galaxies considered in the fit
out	*vso(npg1)*	dimension REAL∗8 line-of-sight velocity [km/s] of the galaxies considered in the fit

Example of usage, assuming to have read data and all the input parameters in pars_test.txt and Options.txt:

```
open(20,file="rnvn.dat",status='unknown')
open(30,file="Npar.dat",status='unknown')
open(60,file="MaxLik.dat",status='unknown')
iu20=20
iu30=30
iu60=60
call mamposst(di,ve,eve,rso,vso,npg1)
close(30)
close(60)
```

Note that the file "iu20" is closed within the mamposst subroutine.

2.1.2.3 sigmar1()

```
function sigmar1 (
             tt )
```

Authors

A. Biviano, G. Mamon, G. Boue'

N(R) - NFW-model. This function computes the projected cumulative number profile for a NFW model which is used to calculate the radial velocity dispersion profile (VDP). It is based on eq. (41) and (42) of Lokas & Mamon (2001) It requires the inclusion of the external file paramsoptS.i and the values of the number density scale radius rc and of the
virial radius of the mass model, r200.

Note that the normalization is not considered as it simplifies in the equation of the VDP.

Parameters

in	*tt*	(projected) radial distance at which the profile is computed, in unit of Mpc.

Example of usage:
```
rc=0.3
r200=1.41
tt=0.5
tnnfw=sigmar1(tt)
write(*,*) tnnfw
```

2.1.2.4 sigmar2()

```
function sigmar2 (
                tt )
```

Authors

A. Biviano, G. Mamon, G. Boue'

N(R) - Hernquist model. This function computes the projected cumulative number profile for a Hernquist model which is used to calculate the radial velocity dispersion profile (VDP). See Hernquist (1990).

It requires the inclusion of the external file paramsoptS.i and the values of the number density scale radius rc.

Note that the normalization is not considered as it simplifies in the equation of the VDP.

Parameters

in	*tt*	(projected) radial distance at which the profile is computed, in unit of Mpc.

Example of usage:
```
rc=0.3
tt=0.5
tnher=sigmar2(tt)
write(*,*) tnher
```

2.1.2.5 sigmar3()

```
function sigmar3 (
```

```
            tt )
```

Authors

A. Biviano, G. Mamon, G. Boue'

N(R) - beta-function. This function computes the projected cumulative number profile for a beta-model which is used to calculate the radial velocity dispersion profile (VDP). It requires the inclusion of the external file paramsoptS.i and the values of the number density scale radius rc and of the exponent of the beta-model al.

Note that the normalization is not considered as it simplifies in the equation of the VDP.

Parameters

in	*tt*	(projected) radial distance at which the profile is computed.

Example of usage:

```
rc=0.3
tt=0.5
tnbeta=sigmar3(tt)
write(*,*) tnbeta
```

2.1.2.6 sr2int()

```
function sr2int (
            alr )
```

Authors

A. Biviano, G. Mamon, G. Boue', L. Pizzuti

Integrand function for the determination of $\nu(r) * \sigma_r^2(r)$, given beta(r) and M(r) from eqs. A3, A4, A6 in Mamon & Lokas (2005) or from eq.(A3) in Mamon, Biviano & Boue' (2013). It includes the modified gravity parametrizations of the mass profile described in Pizzuti et al., 2021

It requires the inclusion of the COMMON parameters defined in paramsoptS.i, the values of all the paramters r200, rs, rc, cbe, cbe0, tmass, screen, rcut, al, kani, kmp, knfit, kscr, v200=10.$*$hz$*$r200. if kmp=7 and kscr$>$0, one needs to specify also nhs, the exponent for the Hu&Sawicki model of $f(R)$, and the redshift parameter zb. A special case corresponds to the number of steps in the anisotropy parameter nbs=-1, where the Hansen & Moore model is selected (the anisotropy depends on the mass density profile $\beta(r) = a + b d\rho(r)/d\ln r$). In this case, the parameter ahm=a and bhm=b should be specified externally.

Parameters

in	*alr*	natural logarithm of the radial distance at which the profile is computed.

Example of usage:

```
      nbs=1
!     Values of the parameters of Hansen & Moore's (2006)
!     beta = a + b dln(rho)/dln(r) relation
```

```
ahm=-0.15
bhm=-0.192
r200=1.41
rc=0.3
rs=0.3
cbe=1.2
cbe0=1.0 !not used unless kani=21, 41
tmass=0.10
screen=0.4
kmp=7    !mass profile model. kmp=7 corresponds to mNFW_LH
v200=r200*10*85
knfit=1  !Number density profile model. knfit=1 Corresponds to pNFW
kani=4   !Anisotropy model. kani=4 corresponds to Tiret profile
kscr=1   !screening option
rcut=1.0
zb=0.3
nhs=2
alr=dlog(2.5d0)
sint=sr2int(alr)
write(*,*)  sint
```

2.1.2.7 sr2out()

```
function sr2out (
                t )
```

Authors

A. Biviano, G. Mamon, G. Boue',

This function computes the factor outside the integral for the σ_r^2 formula.

It requires the inclusion of the COMMON parameters in file paramsoptS.i, the values of r200, rc, cbe, cbe0, al, kani, knfit.

A special case corresponds to the number of steps in the anisotropy parameter nbs=-1, where the Hansen & Moore model is selected (the anisotropy depends on the mass density profile $\beta(r) = a + bd\rho(r)/d\ln r$). In this case, the parameter ahm=a and bhm=b should be specified externally.

Parameters

in	*t*	radial distance at which the profile is computed.

Example of usage:

```
nbs=1
!Values of the parameters of Hansen & Moore's (2006)
!beta = a + b dln(rho)/dln(r) relation
ahm=-0.15
bhm=-0.192
r200=1.41
rc=0.3
cbe=1.2
cbe0=1.0 !not used unless kani=21, 41
knfit=2  !number density profile model. knfit=2 corresponds to pHer
kani=4  !velocity anisotropy model. kani=4 corresponds to Tiret profile
alr=dlog(0.5)
sout=sr2out(t)
write(*,*) sout
```

2.1.2.8 vmaxlik()

```
subroutine vmaxlik (
             nfv,
            dimension(nfv) xfv,
             f )
```

Authors

A. Biviano, G. Mamon, G. Boue', L. Pizzuti

this subroutine compute the value of the likelihood $-\ln\mathcal{L}(\theta)$, where θ is the vector of values for the model parameters. It requires all the values of the input parameters in `pars_test.txt`, and in `Options.txt`, as well as the COMMON blocks: `paramsoptS.i`, `sr.i`, `vlos.i`, `datarv.i` `probs.i`.

Data of projected positions (*in Mpc*), velocities (*in km/s*) and and velocity uncertainties (*in km/s*) should be passed as a common set of vectors `r(ndata), v(ndata) e(ndata), w(ndata)`, where `ndata` is the total number of data points and `w(ndata)` represents the weights (assumed to be 1 in general)

Parameters

`in`	*nfv=2*	dimension of the vector xfv
`in`	*xfv=(rs,cbe)*	two-dimensional vector with the values of rs and cbe
`out`	*f*	value of the (-)log likelihood

2.2 Modified Gravity

Functions/Subroutines

- function frlin (x)
- function field (x)
- function dphidr (x)

2.2.1 Detailed Description

This set of functions are used to compute some quantities needed for the kinematic analysis in modified gravity

2.2.2 Function/Subroutine Documentation

2.2.2.1 dphidr()

```
function dphidr (
              x )
```

Author

L. Pizzuti

compute the term in the effective mass due to chameleon field It requires the parameters in the COMMON block paramsoptS.i, in particular:

- (effective) mass profile parameters: `r200, rs, tmass, screen`. The background field value `tmass` is given by tmass (in unit of 1e-5)
- cosmological parameters: hubble constant `h0` (in km/s/Mpc), density parameters `Omegam, Omegal`, redshift `za`
- mass model `kmp`
- number of steps in the second MG parameter `nhone`. If equal to -1 it forces f(R) chameleon case.

Parameters

in	*x*	REAL∗8, value at which the function is computed

Example of usage:

```
h0=70
omegam=0.3
omegal=0.7
kmp=9
r200=1.2
rs=0.5
tmass=100.23
nhone=10
screen=0.3
write(*,*) dphidr(2.12d0)
```

2.2.2.2 field()

```
function field (
              x )
```

Author

L. Pizzuti

It computes the value of the scalaron perturbations \f $\delta f_R

2.2.2.3 frlin()

```
function frlin (
            x )
```

Author

L. Pizzuti

It computes the contribution to the effective mass profile produced by a linear Horndeski modification of gravity (i.e. no screening) assuming that the mass of the field is given by the COMMON parameter `tmass` and the coupling by the COMMON parameter `screen`. The source of the mass distribution is modeled as a NFW profile expressed by the virial radius `r200` and the scale radius `rs`. It requires the inclusion of the block of COMMON parameters `paramsoptS.i` and defined values of the parameters `r200, rs, tmass, screen, kscr`. The function is scaled by $M_{200}/f200$, where $f200 = (\log(1 + r_{200}/r_s) - (r_{200}/r_s)/(1 + r_{200}/r_s))$.

Parameters

in	*x*	values of the radius (in Mpc) from the cluster center at which the function if computed

Example of usage:
```
kscr=-1  !screening parameter. kscr=-1 corresponds to linear Horndeski gravity
screen=1.2
r200=1.2
rs=0.5
tmass=0.8
write(*,*) frlin(2.12d0)
```

2.3 Lensing Simulation

Functions/Subroutines

- function fmg (x, ya, yb)
- function gmg (x, ya, yb)
- function f_nfw (x)
- function g_nfw (x)
- function plens (r2x, rsx)
- function gt_true (R)
- function gt_mod (R, r2, rss, ya, yb)
- subroutine likelens_bh (npoint, plen)

2.3.1 Detailed Description

This set of routines and functions are needed to compute the lensing likelihood in Vainsthein screening and Chameleon screening

2.3.2 Function/Subroutine Documentation

2.3.2.1 f_nfw()

```
function f_nfw (
              x )
```

Author

L. Pizzuti

This function computes the analytic expression for the radial dependence of the projected surface density of a NFW profile in GR $f_{NFW}(x)$, defined by $\Sigma_{NFW}(R) = 2\rho_s r_s \times f_{NFW}(R/r_s)$. See e.g. Bartelmann (1996).

Parameters

in	*x*	REAL*8, dimensionless radius $x = r/r_s$

Example of usage:

```
write(*,*) f_nfw(1.2d0)
```

2.3.2.2 fmg()

```
function fmg (
              x,
              ya,
              yb )
```

Author

L. Pizzuti

This function computes the analytic expression for the radial dependence of the projected surface density of a NFW profile in DHOST gravity $f_{mg}(x, Y_1, Y_2)$, defined by $\Sigma_{mg}(R) = 2\rho_s r_s \times f_{mg}(R/r_s)$. The expression has been obtained by integrating eq. (36) with the surface density (41) in Pizzuti et al. (2021).

Parameters

in	*x*	REAL*8 dimensionless radius $x = r/r_s$
in	*ya*	REAL*8 first modified gravity parameter

in	*yb*	REAL*8 second modifed gravity parameter Example of usage:
		`ya=0.5d0 yb=-2.0d0`
		`write(*,*) fmg(1.2d0,ya,yb)`

2.3.2.3 g_nfw()

```
function g_nfw (
              x )
```

Author

L. Pizzuti

This function computes the analytic expression for the radial dependence of the projected mass density of a NFW profile in GR $g_{NFW}(x)$, defined by $M_{P,NFW}(R) = 2\rho_s r_s \times g_{NFW}$. See e.g. Bartelmann (1996).

Parameters

in	*x*	REAL∗8, dimensionless radius $x = r/r_s$

Example of usage:
```
write(*,*) f_nfw(1.2d0)
```

2.3.2.4 gmg()

```
function gmg (
              x,
              ya,
              yb )
```

Author

L. Pizzuti

This function computes the expression for the radial dependence of the projected mass density of a NFW profile in DHOST gravity $g_{mg}(x, Y_1, Y_2)$, defined by $M_{P,mg}(R) = 2\rho_s r_s \times g_{mg}(R/r_s)$. The expression has been obtained by integrating eq. (37) with the surface density (41) in Pizzuti et al. (2021).

Parameters

in	*REAL∗8,x*	dimensionless radius $x = r/r_s$
in	*REAL∗8,ya*	first modified gravity parameter
in	*REAL∗8,yb*	second modifed gravity parameter Example of usage:

```
ya=0.5d0
yb=-2.0d0
write(*,*) gmg(1.2d0,ya,yb)
```

2.3.2.5 gt_mod()

```
function gt_mod (
              R,
              r2,
              rss,
              ya,
              yb )
```

Authors

L. Pizzuti, G. Mamon

This routine computes the average tangential shear profile, eq. (34) in Pizzuti et al., 2021, at the projected radius R for a modified NFW profile in Vainsthein screening (eq. (41) in Pizzuti et al., 2021).

Parameters

in	*R*	REAL∗8, projected radius from the center of the mass distribution
in	*r2*	REAL∗8, virial radius r200 of the NFW model
in	*rss*	REAL∗8, scale radius rs of the NFW model
in	*ya*	REAL∗8, first MG coupling
in	*yb*	REAL∗8, second MG coupling example of usage:

```
r2=1.41
rss=0.3
ya=0.1
yb=0.5
write(*,*) gt_mod(1.2d0,r2,rss,ya,yb)
```

2.3.2.6 gt_true()

```
function gt_true (
              R )
```

Authors

L. Pizzuti, G. Mamon

This routine computes the average tangential shear profile, eq. (34) in Pizzuti et al., 2021, at the projected radius R for a NFW model described by the external parameters `r200t, rst`.

Parameters

in	*R*	REAL∗8, projected radius from the center of the mass distribution

Additional requirements are the mass profile parameters `r200t, rst` and the inclusion of `paramsoptS.i`.

Example of usage:

```
r200t=1.41
rst=0.3
write(*,*) gt_true(1.2d0)
```

2.3.2.7 likelens_bh()

```
subroutine likelens_bh (
              npoint,
              plen )
```

This subroutine computes the simulated lensing likelihood for a galaxy cluster in Vainsthein screening gravity. the "true" mass profile is assumed to be a NFW model.

Parameters

in	*npoint*	INTEGER∗4, number of points where the likelihood is evaluated
out	*plen*	REAL∗8, value of the -log(likelihood)

The redshift of the halo is fixed to z=0.44, as an average redshift for optimal lensing observations (see Pizzuti et al., 2022) Note that in this version of the code, the lensing simulation is assumed to be at a fixed cosmology with $H_0 = 67.5$ Mpc, $\Omega_m = 0.32, \Omega_\Lambda = 0.68$

It requires additional external parameters from `pars_test.txt`, from `Options.txt` and the COMMON block `paramsoptS.i`. In particular:

- the mass profile parameters `r200, rs, tmass, screen`
- maximum radius of the kinematic analysis `rupin`
- lensing "true" value of the mock profile `rst, r200t`
- lensing uncertainties: sigma_ellipticity, sigma_lss, Number of galaxies per arcmin^2 (min 10)

  Example of usage:

```
r200=1.41
rs=0.3
cbe=1.2
tmass=1.2
screen=0.4
rupin=r200
rst=0.3    !lensing "true" value of the scale radius
r200t=1.41 !lensing "true" value of the virial raidius
wr2=0.25  !intrinsic ellipticity
wr=0.005  !lss noise
wx=30     !number of galaxies per arcmin^2
call  likelens_bh(10,plen)
write(*,*) plen
```

2.3.2.8 plens()

```
function plens (
              r2x,
              rsx )
```

Author

L. Pizzuti

This funcion computes the logarithm of a bivariate gaussian probability distribution centered on the values r2x and rsx, with relative standard deviations and correlation given externally by `wr2, wr, wx` respectively, defined in the COMMON block `paramsoptS.i`. "Relative" means that e.g. the variance of `r2x` is given by `r2x*wr2`.

Parameters

in	*r2x*	REAL∗8, central value of the first parameter
in	*rsx*	REAL∗8, central value of the second parameter

Example of usage:

```
wr2=0.2
wr=0.1
wx=0.5
write(*,*) plens(r2x,rsx)
```

Index