Add New Data Sets in CosmoMC

Chia-Hsun Chuang (Albert)
Leibniz institute for astrophysics Potsdam, Germany

School on cosmology tools
March 15th, 2017

More than one way..

A. Run new chains
    1. No extra parameter (HST.f90)
    2. With extra parameters which do not pass through CAMB (supernovae_SLNS.f90)
    3. Generic MCMC sampler (Calclike.f90)
    4. Add new parameters for CAMB

B. Importance sampling (modify old chains)
    1. CosmoMC
    2. Getdist

Use HST.f90
(add new data without additional parameters)

```
module HST
contains
   subroutine HSTLikelihood_Add(LikeList, Ini)
… initialization for new data
   end subroutine HSTLikelihood_Add

   real(mcp) function HST_LnLike(like, CMB)
…
      HST_LnLike = (theoryval -
angdistinvzeffh0)**2/(2*angdistinvzeffh0errsqr)
      HST_LnLike = new_LnLike
   end function  HST_LnLike
end module HST
```

params.ini :
INCLUDE(batch1/HST.ini)

Use supernovae_SNLS.f90 (1/2) (new data with new parameters)

```
  MODULE SNLS
 CONTAINS
  subroutine SNLSLikelihood_Add(LikeList, Ini)
… initialization for new data
  end subroutine SNLSLikelihood_Add

 FUNCTION snls_LnLike(like, CMB, Theory, DataParams)
…
     alpha=DataParams(1)
     beta=DataParams(2)
     snls_LnLike=SNLS_alpha_beta_like(alpha, beta, lumdists)
     snls_LnLike = new_LnLike
  END FUNCTION snls_LnLike
```

Use supernovae_SNLS.f90 (1/2)
(new data with new parameters)

params.ini :

```
INCLUDE(batch1/SNLS.ini)
```

Batch1/SNLS.ini :

```
use_SN = T
use_SNLS = T
param[alpha_SNLS]=1.442 0.6 2.6 0.11 0.11
param[beta_SNLS]=3.262 0.9 4.6 0.11 0.11
```

data/SNLS.paramnames :

```
alpha_SNLS     \alpha_{SNLS}
beta_SNLS      \beta_{SNLS}
```

Calclike.f90
(use CosmoMC as a generic MCMC sampler)

```
 module CalcLike
 contains
...
 function Generic_GetLogLikeMain(this, Params) result(LogLike)!Get -Ln(Likelihood) for chains
class(TGenericLikeCalculator) :: this
   class(TCalculationAtParamPoint) :: Params
   real(mcp) LogLike

   !Used when you want to plug in your own CMB-independent likelihood function:
      !Parameter array is Params%P, so e.g. 2D unit Gaussian would be
      LogLike = (Params%P(1)**2+Params%P(2)**2)/2
      !LogLike = LogZero
      !call MpiStop('Generic_GetLogLikeMain: need to write this function!')

   end function Generic_GetLogLikeMain
```

```
./cosmomc  params_generic.ini
```

Importance sampling (CosmoMC)

1. Pre-processing

Params.ini:

```
file_root = chains/test
#set indep_sample a positive integer to produce .data file
Indep_sample = 5
```

2. Post-processing the .data file

Params.ini :

```
INCLUDE(batch1/HST.ini) (want to add HST to chains)
INCLUDE(batch1/importance_sampling.ini)
file_root = chains/test
action = 1
redo_outroot = chains/test_add_HST
```

Important sampling (getdist)

distparams.ini :

```
file_root=chains/original_chains
out_root=name_of_output
map_params = T
```

Getdist.f90:

```
 subroutine MapParameters(invars)
 real(gp) invars(1:ncols)
 real :: chi2
 chi2 = chi2_from_new_data()
 invars(1) = invars(1)*exp(-chi2/2)
 invars(2) = invars(2) + chi2/2
 end subroutine MapParameters
```

Add new parameters

CosmologyTypes.f90

```
 Type, extends(TTheoryParams) :: CMBParams
     real(mcp) InitPower(max_inipower_params)
           !These  are  fast  paramters  for the initial  power  spectrum
           !Now  remaining  (non-independent)  parameters
      real(mcp) omb, omc, omv, omnu, omk, omdm
      real(mcp) ombh2, omch2, omnuh2, omdmh2
      real(mcp) zre, zre_delta, nufrac
           real(mcp)  h, H0, tau
      real(mcp) w, wa, new_param
      real(mcp) YHe, nnu, iso_cdm_correlated, ALens, Alensf, fdm !fdm is dark matter annihilation, eg,. 0910.3663
      real(mcp) :: omnuh2_sterile = 0._mcp  !note omnhu2 is the sum of this + standard neutrinos
      real(mcp) :: sum_mnu_standard
      real(mcp) reserved(5)
```

Add new parameters

/paramnames/params_CMB.paramnames:

```
…
fdm          \epsilon_0 f_d    #CosmoRec dark matter annihilation parameter, 0910.3663
new_param   new_p  # new parameter
ns           n_s        #beware that pivot scale can change in .ini file
…
```

CosmologyParameterizations.f90:

```
   call  this%SetTheoryParameterNumbers(17,last_power_index)
!!it  was  16
  end subroutine TP_Init
…
  subroutine SetForH
…
       CMB%fdm = Params(16)
       CMB%new_param = Params(17)
```

param.ini :

```
…
param[new_param] = 7
…
```

Test adding parameter with HST.f90

```
module HST
   real(mcp) function HST_LnLike(like, CMB)
…
      HST_LnLike = (theoryval -
angdistinvzeffh0)**2/(2*angdistinvzeffh0errsqr)
      write(*,*) “new parameter =“, CMB%new_param
      stop
   end function  HST_LnLike
end module HST
```

params.ini :

```
INCLUDE(batch1/HST.ini)
```

Stdout:

```
new parameter = 7.00000
```

Pass to CAMB: (see Alessandra Silvestri & Matteo Martinelli’s lectures for modifying CAMB)

Modify subroutine CAMBCalc_CMBToCAMB(this,CMB,P)

in Calculator_CAMB.f90