Introduction into CAMB

to CAMB-Version 10/2013

all texts within “” are extracted out of the CAMB-code or from http://camb.info/readme.html!

Martina Schwind
Institute for Theoretical Physics Heidelberg

1/27/14

Overview

- Running CAMB
- **Fortran**: program, procedure, module
- The program-files of CAMB:
  - **main-program-files**
    - can be used to run different tests
  - useful program-files (independent of **cosmology**)
  - program-files **with cosmology inside**
    - **steering modules**
    - **Recombination, Reionization, Initial Powerspectrum, Bispectrum, Halofit, Lensing**
    - **modules, equations**
- **Example:** inidriver.f90: **program** driver

Running CAMB

Compile with the makefile:
make

includes
makefile_main

choose FFLAGS-Command inside the makefile
fitting to Your technical environment

Compile again with:
make clean && make

Run with ./camb <param-filename>

Fortran: program, procedure, module

- **program**:
  - consists of a sequence of statements: these statements are written on lines that may contain from 0 to 132 characters.
- two kinds of **procedures**:
  - **subroutine:** may be used to perform any computation and is invoked by executing a *call* statements.
  - **function:** looks much like a Fortran program, except that it begins with the keyword *function* instead of the keyword *program.*
- **module**:
  - provide another way of **sharing constants**, **variables**, and **type definitions**.
  - also provide a way of **sharing procedures**, as well as data.
  - is a **program unit** that is **not executed directly**, but contains data specifications and procedures that **may be utilized by other program units** via the ***use*-statement**.

Ref: Programmer's guide to Fortran 90 Brainerd, Goldberg, Adams 3rd edition

All program-files of CAMB

bessels.f90

camb.f90

cmbmain.f90

constants.f90

cosmorec.F90

equations.f90

halofit.f90

hyrec.F90

inidriver.F90

inifile.f90

lensing.f90

Matrix_utils.F90

modules.f90

power_tilt.f90

recfast.f90

reionization.f90

SeparableBispectrum.F90

sigma8.f90

subroutines.f90

tester.f90

utils.F90

writefits.f90

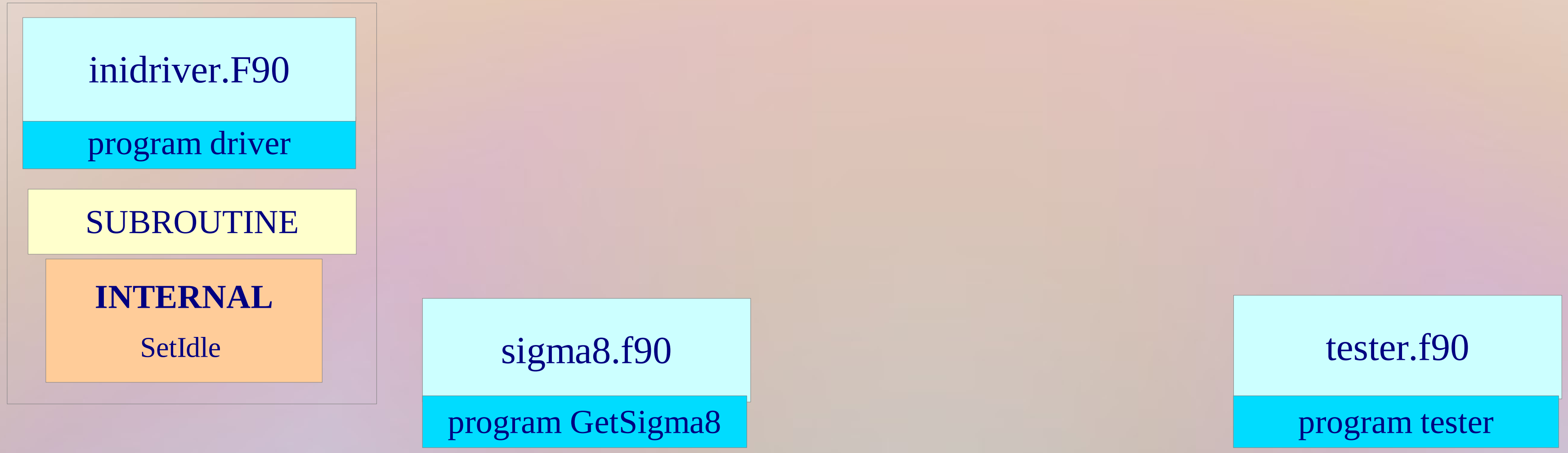
- 22 fortran-program-files

main-program-files of CAMB:
drivers: can be used to run different tests

- bessels.f90
- camb.f90
- cmbmain.f90
- constants.f90
- cosmorec.F90
- equations.f90
- halofit.f90
- hyrec.F90
- inidriver.F90
- inifile.f90
- lensing.f90
- Matrix_utils.F90
- modules.f90
- power_tilt.f90
- recfast.f90
- reionization.f90
- SeparableBispectrum.F90
- sigma8.f90
- subroutines.f90
- tester.f90
- utils.F90
- writefits.f90

- 1 testdriver, 2 examples for calling CAMB

main-program-files of CAMB: drivers: can be used to run different tests



inidriver.F90

program driver

SUBROUTINE

INTERNAL

SetIdle

sigma8.f90

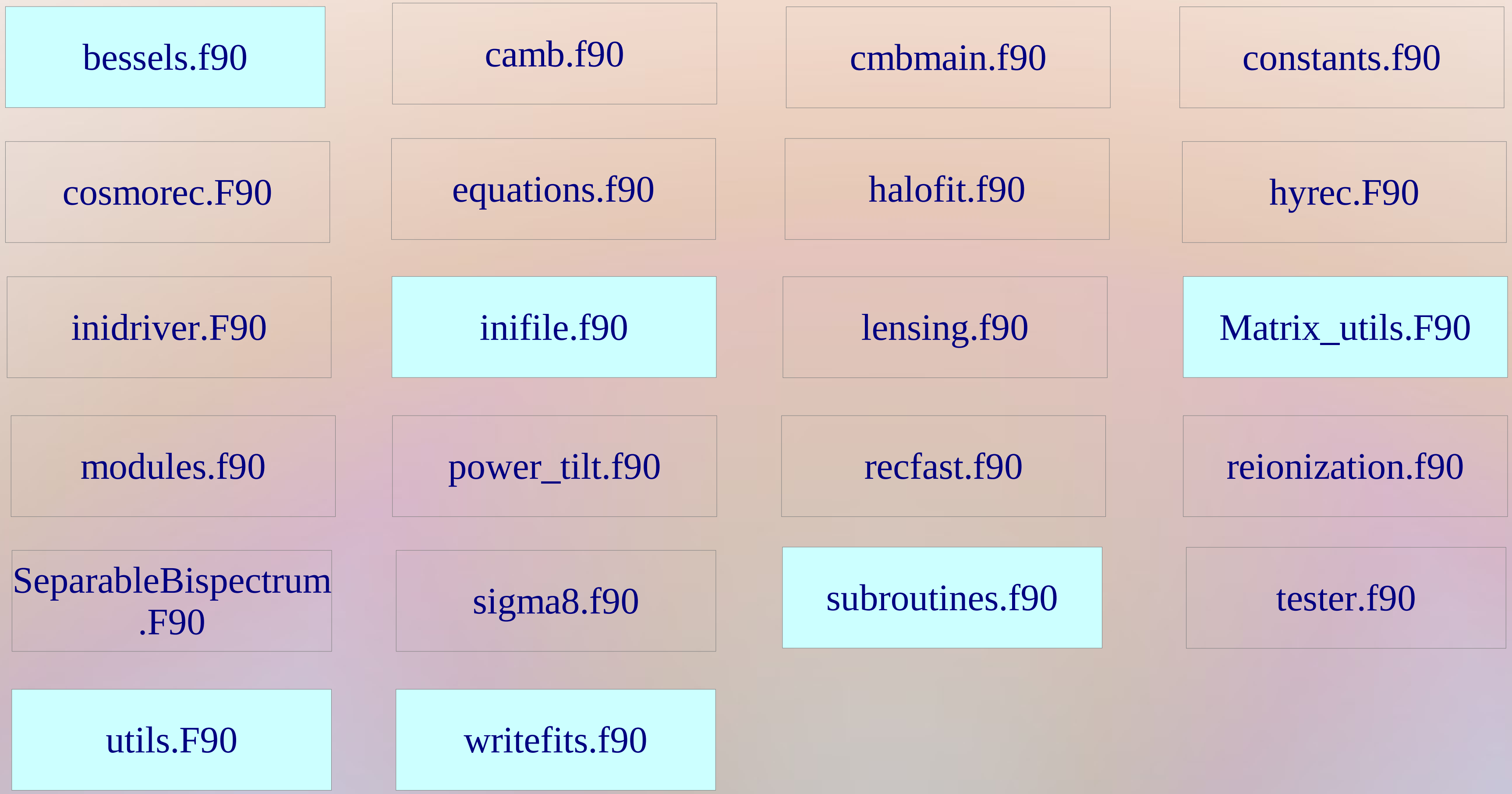
program GetSigma8

tester.f90

program tester

- **Inidriver**: “Reads in parameters from a file of name/value pairs and calls CAMB. Modify this file to generate grids of models, change the parameterization, etc.”
- **sigma8, tester**: “Sample programs … are supplied showing how to use CAMB from your own programs.”

“Useful” program-files of CAMB “**without cosmological logic**”



bessels.f90	camb.f90	cmbmain.f90	constants.f90
cosmorec.F90	equations.f90	halofit.f90	hyrec.F90
inidriver.F90	inifile.f90	lensing.f90	Matrix_utils.F90
modules.f90	power_tilt.f90	recfast.f90	reionization.f90
SeparableBispectrum .F90	sigma8.f90	subroutines.f90	tester.f90
utils.F90	writefits.f90		

- 6 program-files with useful routines: only use them - never change them!

“Useful” program-files of CAMB
“**without cosmological logic**”

bessels.f90

module SpherBessels

bessels.f90: “Module to calculate spherical and hyper-spherical Bessel functions. Hyper-spherical functions generated by use of either the recursion relation or Kosowsky's WKB approximation. Based on Arthur Kosowsky's "hyperjl.c".”

constants.f90

module Precision
module constants
module Errors

inifile.f90

module IniFile

“Module to read in name/value pairs from a file, with each line of the form line 'name = value' Should correctly interpret FITS headers”

Matrix_utils.F90

module MatrixUtils

subroutines.f90

“General numerical routines and global accuracy. Includes modified dverk for CAMB.”

subroutines.f90: “Various subroutines for interpolation, and modified Runge-Kutta dverk for parallelized evolution.”

utils.F90

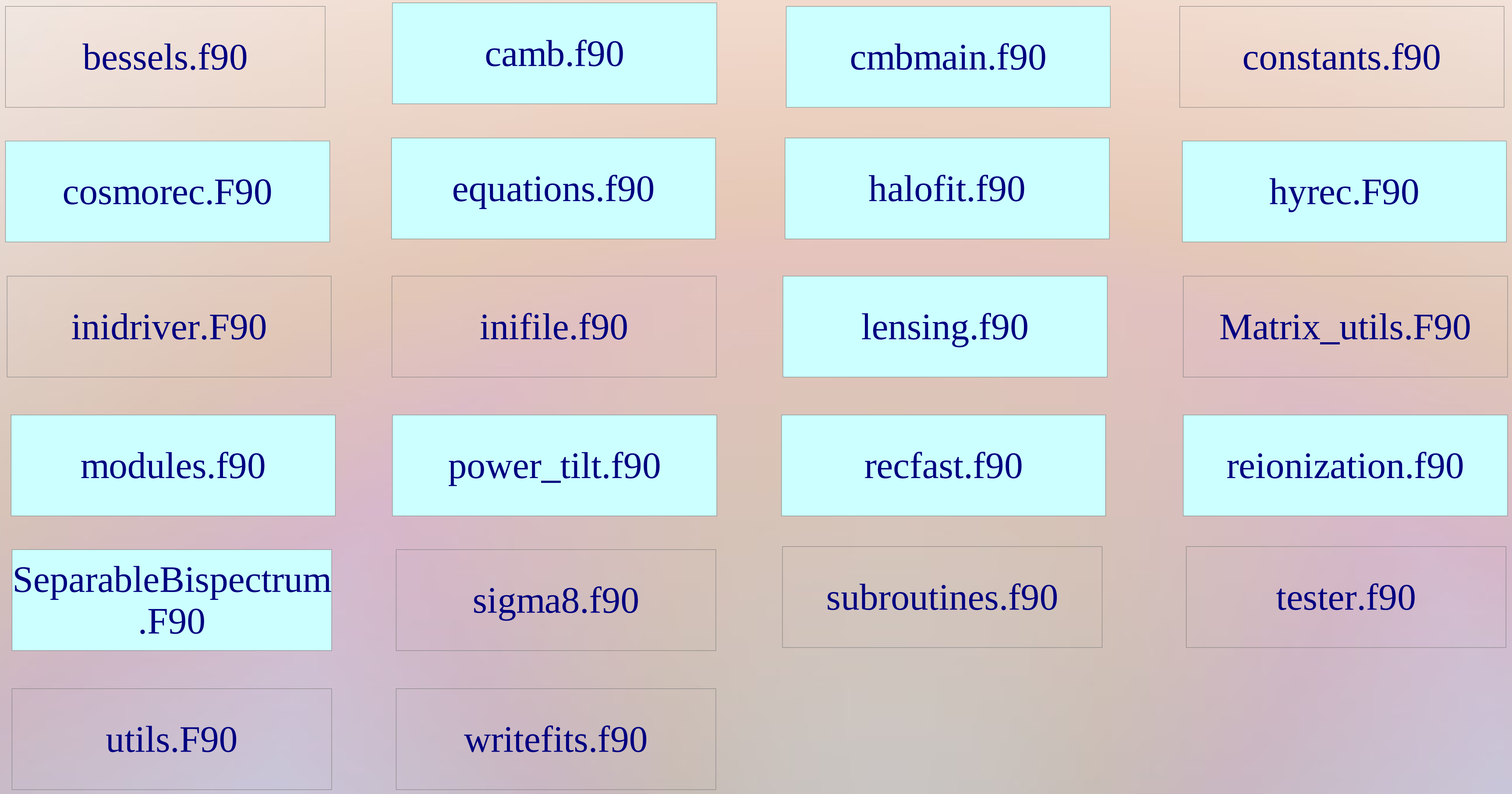
module Ranges
module Lists
module AMLutils
MODULE Ziggurat
module Random

writefits.f90

“subroutine to export Cls in FITS format for HEALPix 1.2”

writefits.f90: “Subroutine WriteFitsCls that uses HEALPIX routines to output power spectrum in FITS format.”

program-files of CAMB **with cosmology inside**

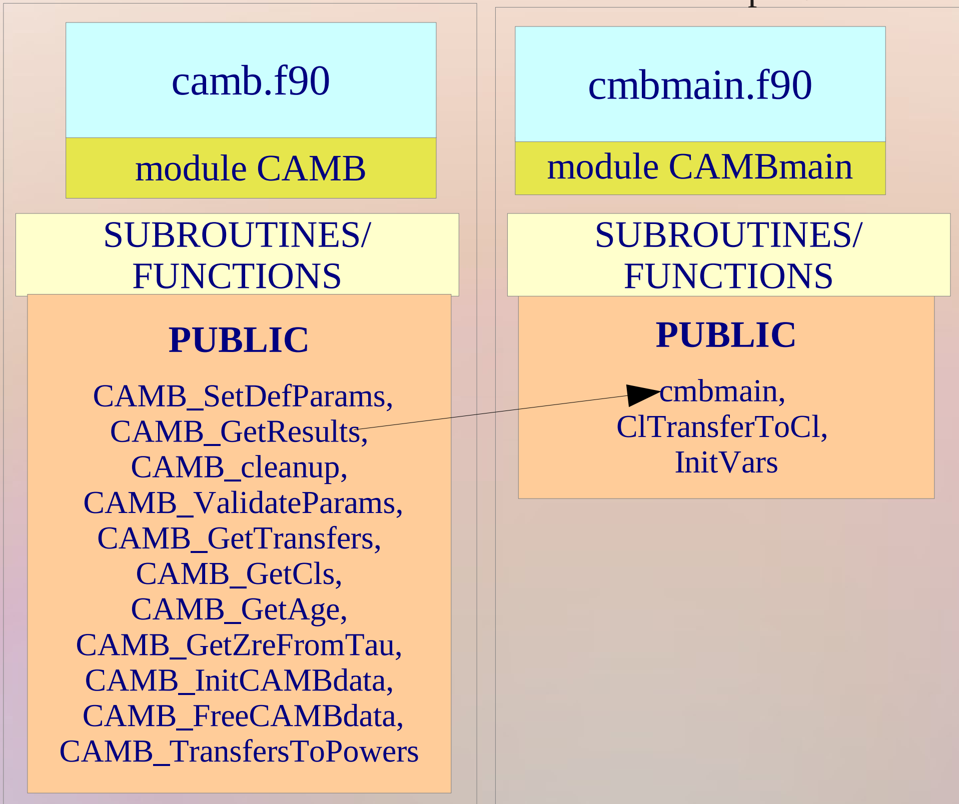


bessels.f90	camb.f90	cmbmain.f90	constants.f90
cosmorec.F90	equations.f90	halofit.f90	hyrec.F90
inidriver.F90	inifile.f90	lensing.f90	Matrix_utils.F90
modules.f90	power_tilt.f90	recfast.f90	reionization.f90
SeparableBispectrum .F90	sigma8.f90	subroutines.f90	tester.f90
utils.F90	writefits.f90		

- 12 program-files with cosmology inside

program-files of CAMB:
cosmology inside: **steering modules**

- **cmbmain.f90**:”The main subroutine that does integrations, etc. Encompasses CMBFAST's cmbflat and cmbopen.”



camb.f90

module CAMB

SUBROUTINES/ FUNCTIONS

PUBLIC

CAMB_SetDefParams,
CAMB_GetResults,
CAMB_cleanup,
CAMB_ValidateParams,
CAMB_GetTransfers,
CAMB_GetCls,
CAMB_GetAge,
CAMB_GetZreFromTau,
CAMB_InitCAMBdata,
CAMB_FreeCAMBdata,
CAMB_TransfersToPowers

cmbmain.f90

module CAMBmain

SUBROUTINES/ FUNCTIONS

PUBLIC

cmbmain,
ClTransferToCl,
InitVars

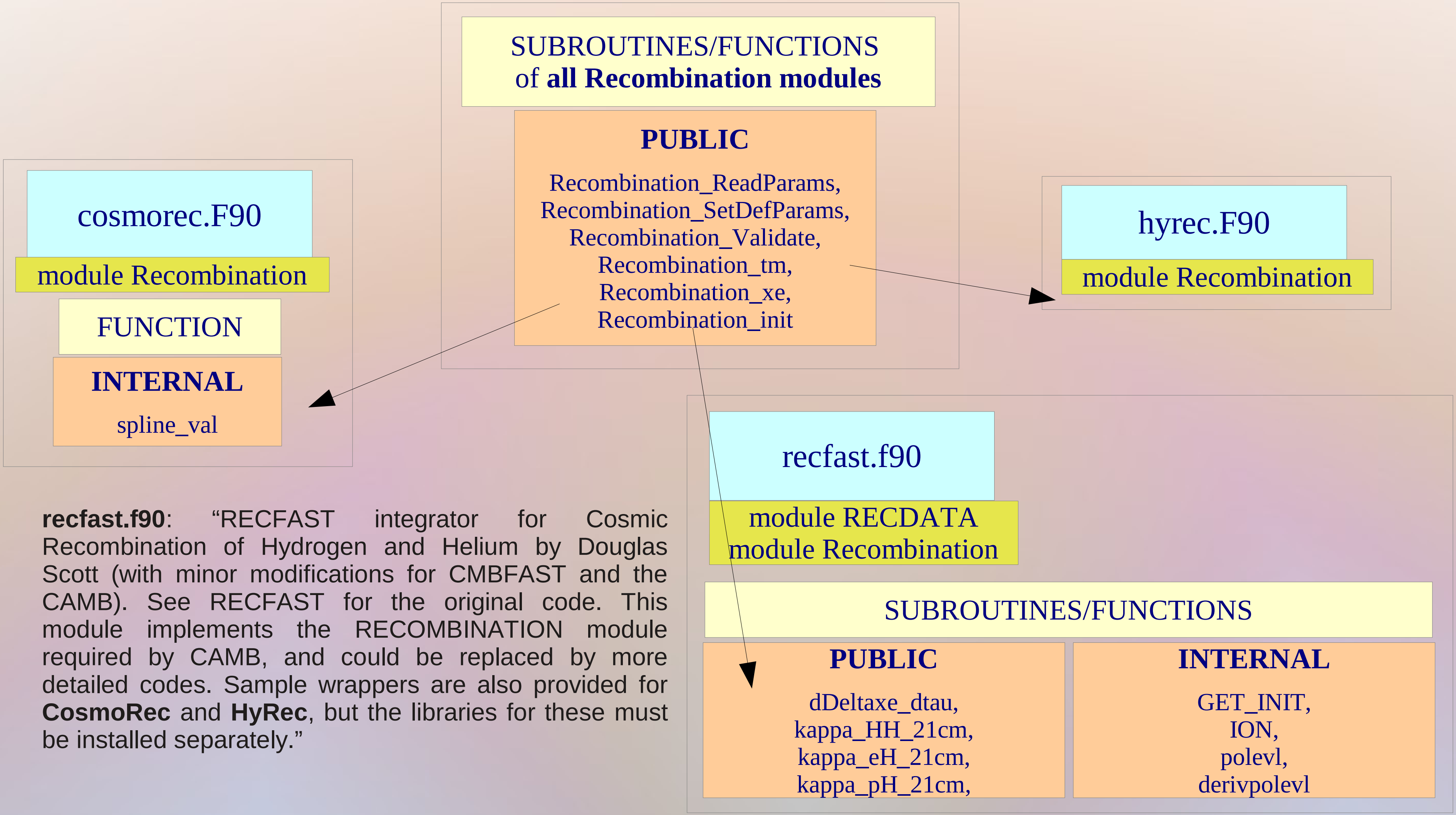
INTERNAL

CalcLimberScalCls,
GetLimberTransfers,
SourceToTransfers,
InitTransfer,
DoSourcek,
GetSourceMem,
FreeSourceMem,
SetkValuesForSources,
SetClosedkValuesFromArr,
CalcScalarSources,
CalcTensorSources,
CalcVectorSources,
TransferOut,
GetTransfer,
MakeNonlinearSources,
InitSourceInterpolation,
SetkValuesForInt,
InterpolateSources,
IntegrationVars_Init,
DoSourceIntegration,
DoFlatIntegration,
IntegrateSourcesBessels,
DoRangeInt,
DoRangeIntTensor,
GetInitPowerArrayVec,
GetInitPowerArrayTens,
CalcScalCls,
(CalcScalCls2?),
CalcTensCls,
CalcVecCls,
InterpolateCls

camb.f90: “Main wrapper routines for running CAMB in your programs.

- Add "**use camb**" to your programs and **call CAMB_GetResults** to generate output from a set of model parameters (specified in the **CAMBparams type** - defined at the top of modules.f90).
- You can **call CAMB_ValidateParams(P)** to check that the parameter set is valid.
- use **CAMB_GetAge** to compute the age of a model in gigayears, and **CAMB_GetCls** to retrieve the computed Cls.
- The results can also be accessed directly using the arrays in the ModelData module (defined in modules.f90).”

program-files of CAMB:
cosmology inside: **Recombination**



SUBROUTINES/FUNCTIONS
of **all Recombination modules**

PUBLIC

Recombination_ReadParams,
Recombination_SetDefParams,
Recombination_Validate,
Recombination_tm,
Recombination_xe,
Recombination_init

cosmorec.F90

module Recombination

FUNCTION

INTERNAL

spline_val

hyrec.F90

module Recombination

recfast.f90

module RECDATA
module Recombination

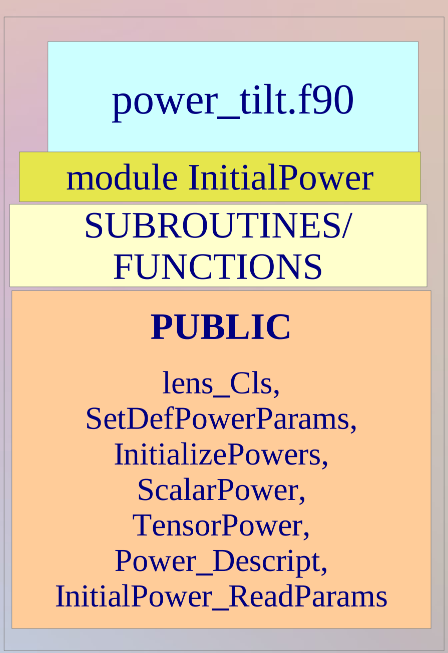
SUBROUTINES/FUNCTIONS

PUBLIC	**INTERNAL**
dDeltaxe_dtau, kappa_HH_21cm, kappa_eH_21cm, kappa_pH_21cm,	GET_INIT, ION, polevl, derivpolevl

recfast.f90: “RECFAST integrator for Cosmic Recombination of Hydrogen and Helium by Douglas Scott (with minor modifications for CMBFAST and the CAMB). See RECFAST for the original code. This module implements the RECOMBINATION module required by CAMB, and could be replaced by more detailed codes. Sample wrappers are also provided for **CosmoRec** and **HyRec**, but the libraries for these must be installed separately.”

program-files of CAMB cosmology inside: **InitialPower, Reionization**

power_tilt.f90: “This file defines a module called InitialPower that returns the initial power spectra. Change this file to use your own initial power spectrum, change how the spectra are parameterized, or to change how the Cls are normalized. Comments in the code explain this further.”



power_tilt.f90

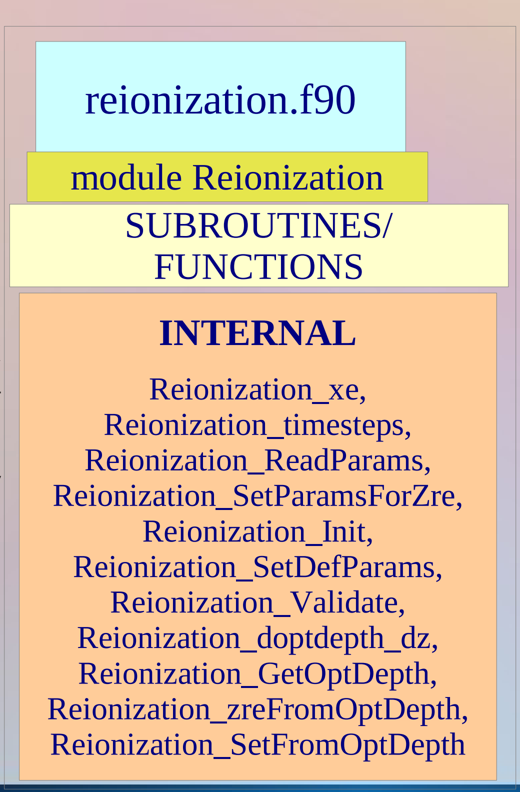
module InitialPower

SUBROUTINES/ FUNCTIONS

PUBLIC

lens_Cls, SetDefPowerParams, InitializePowers, ScalarPower, TensorPower, Power_Descript, InitialPower_ReadParams

reionization.f90: “This file defines a module called Reionization that parameterizes the reionization history and supplies a function Reionization_xe that gives xe as a function of redshift. Optical depth input parameters are mapped into zre (defined as where xe is half its maximum (ex second He reionization)) using a binary search. See the CAMB notes for discussion. This module should be easily modifiable for alternative reionization models.”



reionization.f90

module Reionization

SUBROUTINES/ FUNCTIONS

INTERNAL

Reionization_xe, Reionization_timesteps, Reionization_ReadParams, Reionization_SetParamsForZre, Reionization_Init, Reionization_SetDefParams, Reionization_Validate, Reionization_doptdepth_dz, Reionization_GetOptDepth, Reionization_zreFromOptDepth, Reionization_SetFromOptDepth

program-files of CAMB cosmology inside: **NonLinear, Bispectrum, lensing**

halofit.f90

module NonLinear

SUBROUTINES/FUNCTIONS

INTERNAL

NonLinear_GetNonLinRatios,
halofit,
wint,
omega_m,
omega_v

Not within the module

NonLinear_GetRatios,
NonLinear_GetRatios_all

halofit.f90: “Implements the NonLinear module, to calculate non linear scalings of the matter power spectrum as a function of redshift. This module can be replaced to use a different non-linear fitting method if desired.”

lensing.f90

module lensing

SUBROUTINES/ FUNCTIONS

INTERNAL

CorrFuncFullSky,
CorrFuncFullSkyImpl,
CorrFuncFlatSky,
BadHarmonic,
GetBessels,
bessj,
BESSI0,
BESSI1

PUBLIC

lens_Cls,
BESSI

lensing.f90: “Lensing module for computing the lensed CMB power spectra from the unlensed spectra and a lensing power spectrum.”

SeparableBispectrum .F90

module Bispectrum

SUBROUTINES/ FUNCTIONS

INTERNAL

InitBesselDerivs,
NonGauss_l_r_localOpt,
NonGauss_l_r,
GetBispectrum,
NonGauss_deriv_l_r,
Bispectrum_SetDefParams,
Bispectrum_ReadParams

SeparableBispectrum.f90: “Implements calculation of simple separable primordial bispectra, specifically the local constant fNL model, and the CMB lensing bispectrum due to the linear temperature and polarization cross-correlation with the lensing potential.”

program-files of CAMB
cosmology inside: equations.f90

module LambdaGeneral

SUBROUTINES/FUNCTIONS

INTERNAL

subroutine DarkEnergy_ReadParams

INTERNAL

subroutine init_background,
function dtauda(a)

equations.f90

module LambdaGeneral
module GaugeInterface

module LambdaGeneral
is **used** in
camb.f90
cmbmain.f90
halofit.f90
inidriver.f90

module GaugeInterface
is **used** in
camb.f90
cmbmain.f90

equations.f90: “Files containing background and perturbation evolution equations. The perturbations equations used are derived in the covariant approach, fixing to the CDM (zero acceleration) frame, which are essentially equivalent to the synchronous gauge equations.

The file defines a module called "**GaugeInterface**" which provides the necessary perturbation calculation routines for "cmbmain".

The subroutine **dtauda**(a) returns dt/da and is used wherever the background evolution is needed. It can be modified for different backgrounds. You may also need to change the GetOmegak routine if you add additional components, and can edit the init_background routine to do additional initialization.

outtransf writes out the matter transfer functions.

The "**output**" subroutine computes the scalar sources at a given time for a given wavenumber. These are the temperature, E polarization and (if doing lensing) the lensing source. By editing the equation for the lensing source it should be straightforward to compute power spectra for other matter tracers, e.g. for cross-correlation with the CMB. The lensing power spectrum is automatically computed if DoLensing=T.”

module GaugeInterface

SUBROUTINES/FUNCTIONS

INTERNAL

(cm) subroutine GetNumEqns,

(cm) subroutine GaugeInterface_Init,
subroutine GaugeInterface_ScalEv,

(cm) subroutine initial,
(cm) subroutine GaugeInterface_EvolveScal,
(cm) subroutine derivs,
(cm) subroutine output,
(cm) subroutine outtransf

(cm) subroutine initialv,
(cm) subroutine derivsv (with dverk),
(cm) subroutine outputv,

(cm) subroutine initialt,
(cm) subroutine GaugeInterface_EvolveTens,
subroutine derivst,
(cm) subroutine outputt,

(cm) subroutine initial,
(cm) subroutine GaugeInterface_EvolveScal,
(cm) subroutine outtransf

function DeltaTimeMaxed,
function next_nu_nq,

subroutine SetupScalarArrayIndices,
subroutine CopyScalarVariableArray,
subroutine SetupTensorArrayIndices,
subroutine CopyTensorVariableArray,

subroutine SwitchToMassiveNuApprox,
subroutine MassiveNuVars,,
subroutine MassiveNuVarsOut,
subroutine Nu_Integrate_L012,
subroutine Nu_pinudot,
function Nu_pi,
subroutine Nu_Intvsq

program-files of CAMB
cosmology inside: equations_quint.f90

module LambdaGeneral

SUBROUTINES/FUNCTIONS

INTERNAL

subroutine DarkEnergy_ReadParams

INTERNAL

subroutine init_background,
function dtauda(a)

equations_quint.f90

module LambdaGeneral
module GaugeInterface
additional: module Quint

module Quint

SUBROUTINES/FUNCTIONS

INTERNAL

function Vofphi,
subroutine EvolveBackground,
function Quint_GetOmegaFromInitial,
function Quint_phidot_start,
subroutine Quint_init_background,
subroutine Quint_ValsAta

equations_quint.f90: “There is also a more general quintessence module that lets you specific a single scalar field potential. It will need customization for different potentials, and changes to the way initial conditions are set if trying to use a tracker model. You can download the modified equations module for CAMB. It is not very well tested. “

module Quint
is **used** only in
equations.f90

module LambdaGeneral
is **used** in
camb.f90
cmbmain.f90
halofit.f90
inidriver.f90

module GaugeInterface
is **used** in
camb.f90
cmbmain.f90

module GaugeInterface

SUBROUTINES/FUNCTIONS

INTERNAL

(cm) subroutine GetNumEqns,

(cm) subroutine GaugeInterface_Init

(cm) subroutine initial,
(cm) subroutine GaugeInterface_EvolveScal,
(cm) subroutine derivs,
(cm) subroutine output,
(cm) subroutine outtransf

(cm) subroutine initialv,
(cm) subroutine derivsv (with dverk),
(cm) subroutine outputv,

(cm) subroutine initialt,
(cm) subroutine GaugeInterface_EvolveTens,
subroutine derivst,
(cm) subroutine outputt,

(cm) subroutine initial,
(cm) subroutine GaugeInterface_EvolveScal,
(cm) subroutine outtransf

subroutine GaugeInterface_ScalEv,
function DeltaTimeMaxed,
function next_nu_nq,

subroutine SetupScalarArrayIndices,
subroutine CopyScalarVariableArray,
subroutine SetupTensorArrayIndices,
subroutine CopyTensorVariableArray,

subroutine SwitchToMassiveNuApprox,
subroutine MassiveNuVars,,
subroutine MassiveNuVarsOut,
subroutine Nu_Integrate_L012,
subroutine Nu_pinudot,
function Nu_pi,
subroutine Nu_Intvsq

program-files of CAMB
cosmology inside: part 3

module LambdaGeneral

SUBROUTINES/FUNCTIONS

INTERNAL

subroutine DarkEnergy_ReadParams

INTERNAL

subroutine init_background,
function dtauda(a)

equations.f90

module LambdaGeneral
module GaugeInterface

equations_ppf.f90: " is an alternative module that allows evolving dark energy crossing w=-1."

equations_quint.f90: "There is also a more general quintessence module that lets you specific a single scalar field potential. It will need customization for different potentials, and changes to the way initial conditions are set if trying to use a tracker model. You can download the modified equations module for CAMB. It is not very well tested. "

equations.f90: "Files containing background and perturbation evolution equations. The perturbations equations used are derived in the covariant approach, fixing to the CDM (zero acceleration) frame, which are essentially equivalent to the synchronous gauge equations.

The file defines a module called "**GaugeInterface**" which provides the necessary perturbation calculation routines for "cmbmain".

The subroutine **dtauda**(a) returns dt/da and is used wherever the background evolution is needed. It can be modified for different backgrounds. You may also need to change the GetOmegak routine if you add additional components, and can edit the init_background routine to do additional initialization.

outtransf writes out the matter transfer functions.

The "**output**" subroutine computes the scalar sources at a given time for a given wavenumber. These are the temperature, E polarization and (if doing lensing) the lensing source. By editing the equation for the lensing source it should be straightforward to compute power spectra for other matter tracers, e.g. for cross-correlation with the CMB. The lensing power spectrum is automatically computed if DoLensing=T."

module GaugeInterface

SUBROUTINES/FUNCTIONS

INTERNAL

subroutine GaugeInterface_ScalEv,
function next_nu_nq,
subroutine GaugeInterface_EvolveTens,
function DeltaTimeMaxed,
subroutine GaugeInterface_Init,

subroutine SetupScalarArrayIndices,
subroutine CopyScalarVariableArray,
subroutine SetupTensorArrayIndices,
subroutine CopyTensorVariableArray,

subroutine GetNumEqns,

subroutine SwitchToMassiveNuApprox,
subroutine MassiveNuVars,,
subroutine MassiveNuVarsOut,
subroutine Nu_Integrate_L012,
subroutine Nu_pinudot,
function Nu_pi,
subroutine Nu_Intvsq

subroutine initial,
subroutine initialv,
subroutine initialt,

subroutine derivs,
subroutine derivsv,
subroutine derivst,

subroutine output,
subroutine outputv,
subroutine outputt,

subroutine outtransf

program-files of CAMB
cosmology inside: part 3

equations_ppf.f90: " is an alternative module that allows evolving dark energy crossing w=-1."

equations_quint.f90: "There is also a more general quintessence module that lets you specific a single scalar field potential. It will need customization for different potentials, and changes to the way initial conditions are set if trying to use a tracker model. You can download the modified equations module for CAMB. It is not very well tested. "

equations.f90

module LambdaGeneral
module GaugeInterface

equations.f90: "Files containing background and perturbation evolution equations. The perturbations equations used are derived in the covariant approach, fixing to the CDM (zero acceleration) frame, which are essentially equivalent to the synchronous gauge equations.

The file defines a module called "**GaugeInterface**" which provides the necessary perturbation calculation routines for "cmbmain".

The subroutine **dtauda**(a) returns dt/da and is used wherever the background evolution is needed. It can be modified for different backgrounds. You may also need to change the GetOmegak routine if you add additional components, and can edit the init_background routine to do additional initialization.

outtransf writes out the matter transfer functions.

The "**output**" subroutine computes the scalar sources at a given time for a given wavenumber. These are the temperature, E polarization and (if doing lensing) the lensing source. By editing the equation for the lensing source it should be straightforward to compute power spectra for other matter tracers, e.g. for cross-correlation with the CMB. The lensing power spectrum is automatically computed if DoLensing=T."

modules.f90

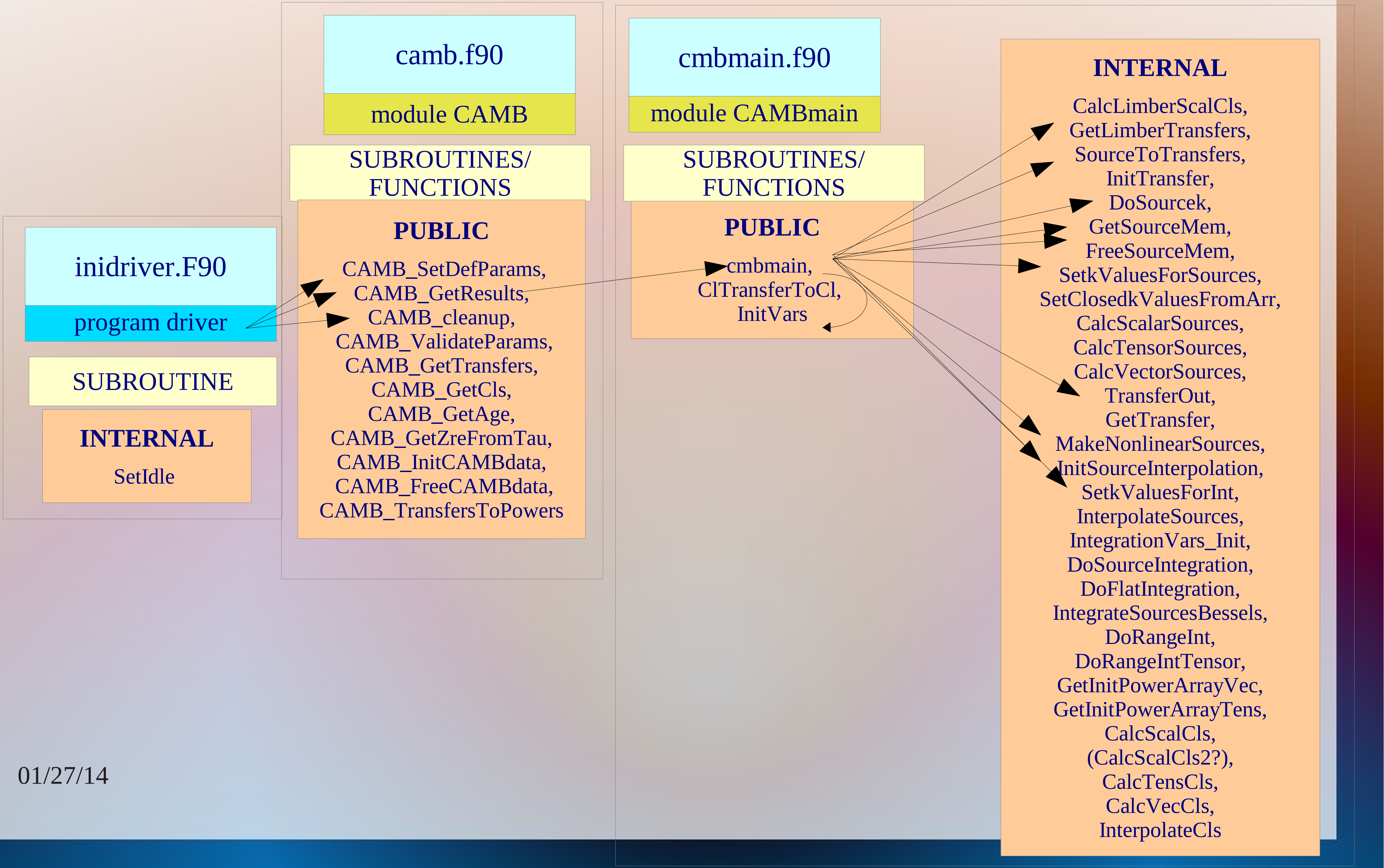
module ModelParams
module lvalues
module ModelData
module MassiveNu
module Transfer
module ThermoData

modules.f90: "Various modules used by the other parts of the program, Module "ModelParams" contains most of the model parameters. Boolean vars flat, open and closed determine the model type."

SUBROUTINES/FUNCTIONS

PUBLIC	**INTERNAL**
d, k, k, ,	G, I, p, dl

EXAMPLE: **inidriver.f90**: program driver



inidriver.F90

program driver

SUBROUTINE

INTERNAL

SetIdle

camb.f90

module CAMB

SUBROUTINES/ FUNCTIONS

PUBLIC

CAMB_SetDefParams, CAMB_GetResults, CAMB_cleanup, CAMB_ValidateParams, CAMB_GetTransfers, CAMB_GetCls, CAMB_GetAge, CAMB_GetZreFromTau, CAMB_InitCAMBdata, CAMB_FreeCAMBdata, CAMB_TransfersToPowers

cmbmain.f90

module CAMBmain

SUBROUTINES/ FUNCTIONS

PUBLIC

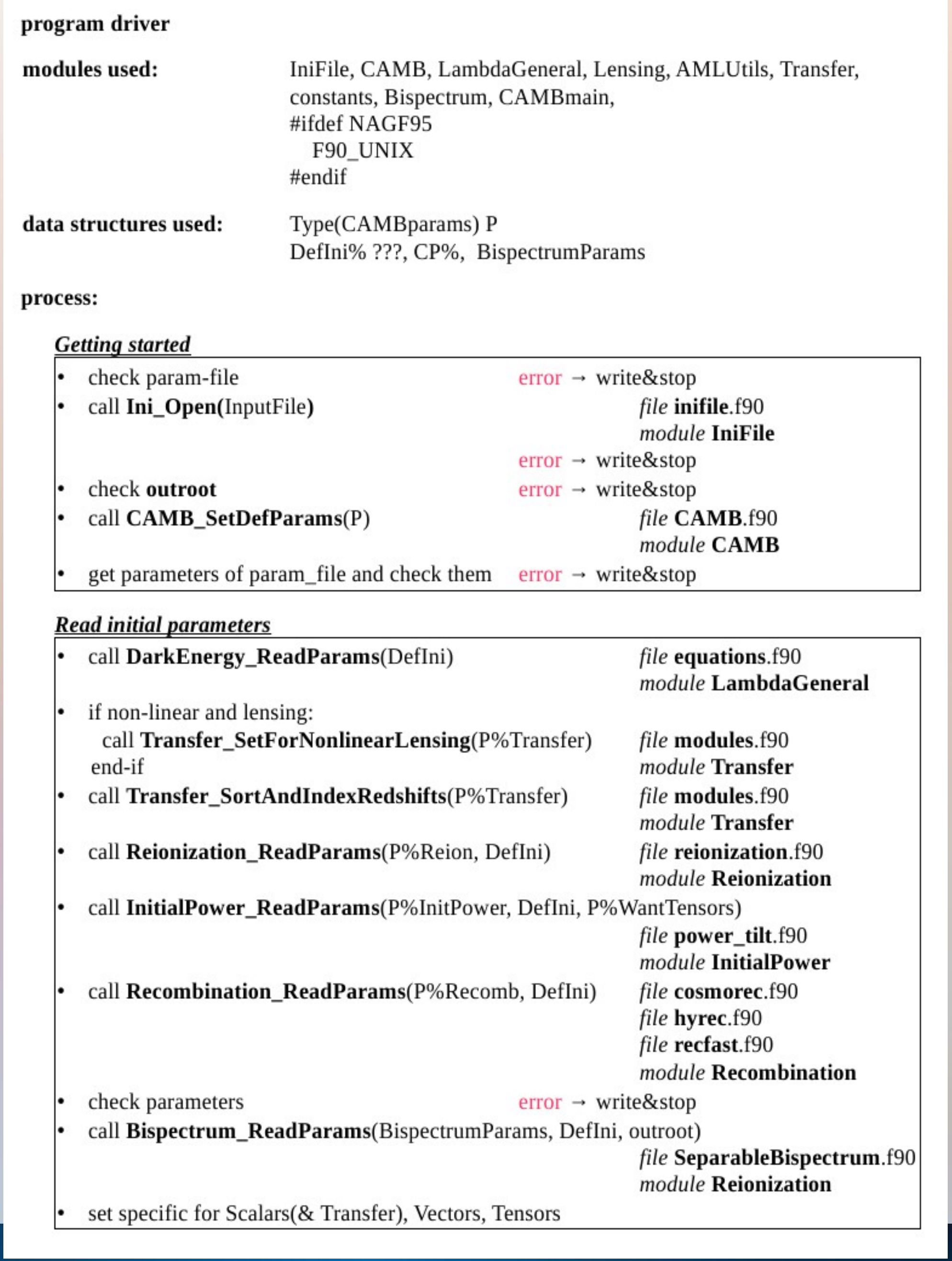
cmbmain, ClTransferToCl, InitVars

INTERNAL

CalcLimberScalCls, GetLimberTransfers, SourceToTransfers, InitTransfer, DoSourcek, GetSourceMem, FreeSourceMem, SetkValuesForSources, SetClosedkValuesFromArr, CalcScalarSources, CalcTensorSources, CalcVectorSources, TransferOut, GetTransfer, MakeNonlinearSources, InitSourceInterpolation, SetkValuesForInt, InterpolateSources, IntegrationVars_Init, DoSourceIntegration, DoFlatIntegration, IntegrateSourcesBessels, DoRangeInt, DoRangeIntTensor, GetInitPowerArrayVec, GetInitPowerArrayTens, CalcScalCls, (CalcScalCls2?), CalcTensCls, CalcVecCls, InterpolateCls

01/27/14

EXAMPLE: **inidriver.f90**: program driver



program driver

modules used: IniFile, CAMB, LambdaGeneral, Lensing, AMLUtils, Transfer, constants, Bispectrum, CAMBmain,
#ifdef NAGF95
F90_UNIX
#endif

data structures used: Type(CAMBparams) P
DefIni% ???, CP%, BispectrumParams

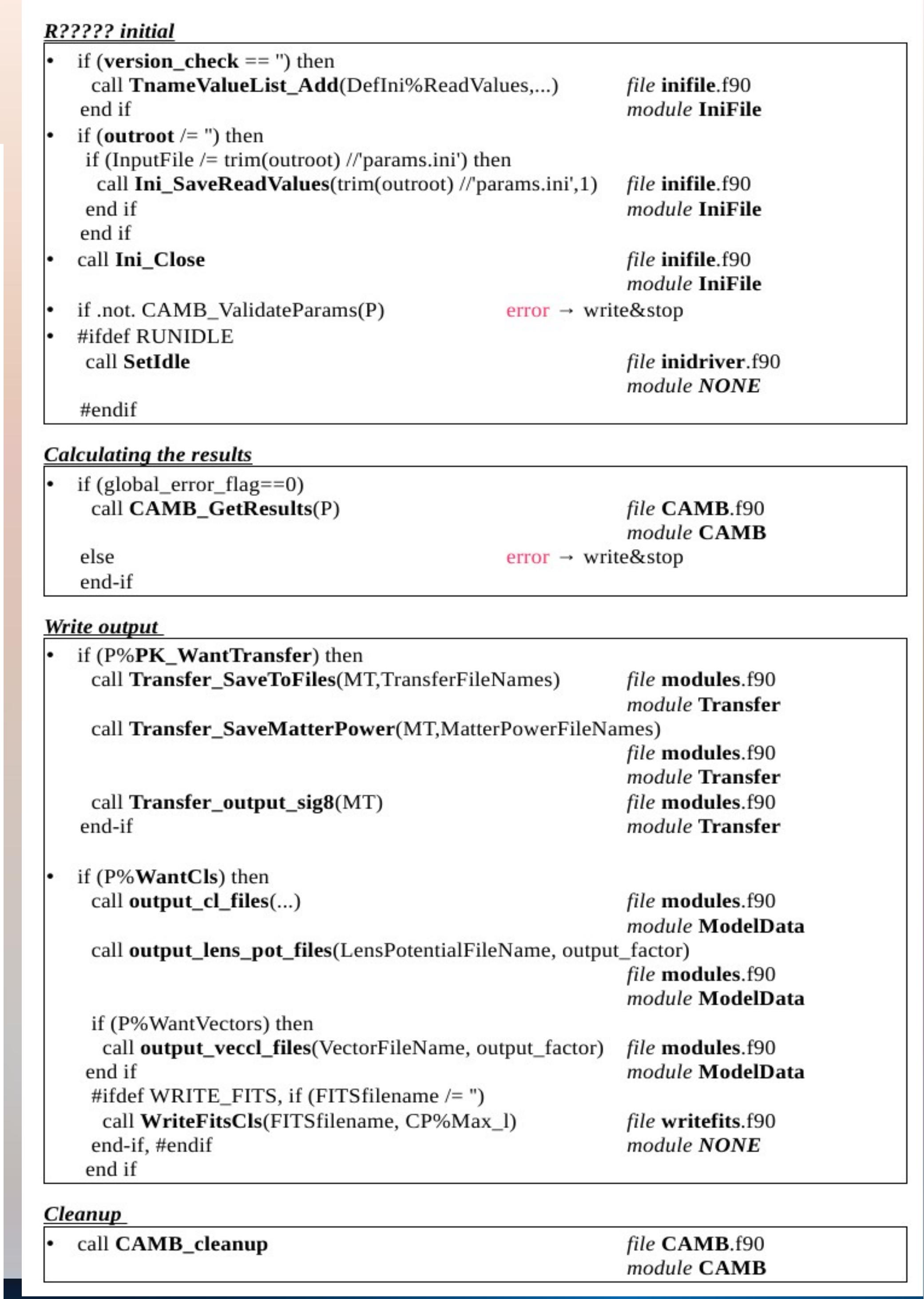
process:

Getting started

```
•  check param-file                              error → write&stop
•  call Ini_Open(InputFile)                               file inifile.f90
                                                          module IniFile
                                                 error → write&stop
•  check outroot                                 error → write&stop
•  call CAMB_SetDefParams(P)                              file CAMB.f90
                                                          module CAMB
•  get parameters of param_file and check them   error → write&stop
```

Read initial parameters

```
•  call DarkEnergy_ReadParams(DefIni)                     file equations.f90
                                                          module LambdaGeneral
•  if non-linear and lensing:
     call Transfer_SetForNonlinearLensing(P%Transfer)     file modules.f90
   end-if                                                 module Transfer
•  call Transfer_SortAndIndexRedshifts(P%Transfer)        file modules.f90
                                                          module Transfer
•  call Reionization_ReadParams(P%Reion, DefIni)          file reionization.f90
                                                          module Reionization
•  call InitialPower_ReadParams(P%InitPower, DefIni, P%WantTensors)
                                                          file power_tilt.f90
                                                          module InitialPower
•  call Recombination_ReadParams(P%Recomb, DefIni)        file cosmorec.f90
                                                          file hyrec.f90
                                                          file recfast.f90
                                                          module Recombination
•  check parameters                              error → write&stop
•  call Bispectrum_ReadParams(BispectrumParams, DefIni, outroot)
                                                          file SeparableBispectrum.f90
                                                          module Reionization
•  set specific for Scalars(& Transfer), Vectors, Tensors
```



R????? initial

```
•  if (version_check == '') then
     call TnameValueList_Add(DefIni%ReadValues,...)        file inifile.f90
   end if                                                  module IniFile
•  if (outroot /= '') then
    if (InputFile /= trim(outroot) //'params.ini') then
      call Ini_SaveReadValues(trim(outroot) //'params.ini',1)   file inifile.f90
    end if                                                 module IniFile
   end if
•  call Ini_Close                                          file inifile.f90
                                                           module IniFile
•  if .not. CAMB_ValidateParams(P)              error → write&stop
•  #ifdef RUNIDLE
    call SetIdle                                           file inidriver.f90
                                                           module NONE
   #endif
```

Calculating the results

```
•  if (global_error_flag==0)
     call CAMB_GetResults(P)                               file CAMB.f90
                                                           module CAMB
   else                                         error → write&stop
   end-if
```

Write output

```
•  if (P%PK_WantTransfer) then
     call Transfer_SaveToFiles(MT,TransferFileNames)       file modules.f90
                                                           module Transfer
     call Transfer_SaveMatterPower(MT,MatterPowerFileNames)
                                                           file modules.f90
                                                           module Transfer
     call Transfer_output_sig8(MT)                         file modules.f90
   end-if                                                  module Transfer

•  if (P%WantCls) then
     call output_cl_files(...)                             file modules.f90
                                                           module ModelData
     call output_lens_pot_files(LensPotentialFileName, output_factor)
                                                           file modules.f90
                                                           module ModelData
     if (P%WantVectors) then
       call output_veccl_files(VectorFileName, output_factor)   file modules.f90
    end if                                                 module ModelData
     #ifdef WRITE_FITS, if (FITSfilename /= '')
       call WriteFitsCls(FITSfilename, CP%Max_l)           file writefits.f90
     end-if, #endif                                        module NONE
    end if
```

Cleanup

```
•  call CAMB_cleanup                                       file CAMB.f90
                                                           module CAMB
```