AZTEKAS: a hydrodynamic GPL code

Version1.0

Generated by Doxygen 1.8.13

Contents

Chapter 1

AZTEKAS: a hydrodynamic GPL code

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 4

Class Documentation

4.1 der_gauge_ Struct Reference

Public Attributes

- double **dlapse** [3]
- double **dbeta** [3][3]
- double **dgam** [3][3][3]

The documentation for this struct was generated from the following file:

- Headers/mesh.h

4.2 eos_ Struct Reference

Public Attributes

- double **e**
- double **cs**
- double **h**

The documentation for this struct was generated from the following file:

- Headers/physics.h

4.3 flx_ Struct Reference

Public Attributes

- double **up** [eq+1]
- double **um** [eq+1]
- double **qp** [eq+1]
- double **qm** [eq+1]
- double **fp** [eq+1]
- double **fm** [eq+1]
- double **lp**
- double **lm**

The documentation for this struct was generated from the following file:

- Headers/physics.h

4.4 gauge_ Struct Reference

Public Attributes

- double **x** [4]
- double **lapse**
- double **beta_con** [3]
- double **gamma_con** [3][3]
- double **dety**

The documentation for this struct was generated from the following file:

- Headers/mesh.h

4.5 grid_ Struct Reference

Public Attributes

- double **time**
- double * **X1**
- double * **X1p**
- double * **X1m**
- double * **X2**
- double * **X2p**
- double * **X2m**
- double * **X3**
- double * **X3p**
- double * **X3m**
- double * **S1p**
- double * **S1m**
- double * **S2p**
- double * **S2m**
- double * **S3p**
- double * **S3m**

The documentation for this struct was generated from the following file:

- Headers/mesh.h

4.6 lim_ Struct Reference

```
#include <limiters.h>
```

Public Attributes

- double **ux1p** [2 ∗eq]
- double **ux1m** [2 ∗eq]
- double **sx1** [2 ∗eq]
- double **ux2p** [2 ∗eq]
- double **ux2m** [2 ∗eq]
- double **sx2** [2 ∗eq]
- double **ux3p** [2 ∗eq]
- double **ux3m** [2 ∗eq]
- double **sx3** [2 ∗eq]
- double **ux** [2 ∗eq]

4.6.1 Detailed Description

The structure **lim_** contains vectors in which the reconstructed variables of U in each cell are stored.

The documentation for this struct was generated from the following file:

- Headers/limiters.h

4.7 vec_ Struct Reference

Public Attributes

- double **A** [(eq+1) ∗(eq+1)]
- double **S** [eq+1]
- double **Fp** [eq+1]
- double **Fm** [eq+1]
- double **Gp** [eq+1]
- double **Gm** [eq+1]
- double **Hp** [eq+1]
- double **Hm** [eq+1]

The documentation for this struct was generated from the following file:

- Headers/physics.h

Chapter 5

File Documentation

5.1 alloc.c File Reference

Essential allocation functions for *aztekas*.

```
#include "main.h"
```

Functions

- void Allocate_Array ()
- void **New_Size** ()

5.1.1 Detailed Description

Author

Alejandro Aguayo-Ortiz

5.1.2 Function Documentation

5.1.2.1 Allocate_Array()

```
void Allocate_Array ( )
```

This function allocates the space in memory for all the vectors used in *aztekas*.

5.2 auxfunc.c File Reference

Helpful functions for *aztekas*.

```
#include "main.h"
```

Functions

- int **MxV** (double ∗M, double ∗V, double ∗L)
- void **RoundGen** (double ∗num)
- void **Scalar_Contraction_Range1** (double ∗scalar, double ∗cov, double ∗con)
- void **Raise_Index_Range1** (double ∗con, double ∗cov, gauge_ ∗local_grid)
- void **Low_Index_Range1** (double ∗cov, double ∗con, gauge_ ∗local_grid)
- void **Low_Index_Range2** (double ∗∗diag, double ∗∗con, gauge_ ∗local_grid)
- void **CheckSimParameters** ()

5.2.1 Detailed Description

Author

Alejandro Aguayo-Ortiz

5.3 bound_cond.c File Reference

Standard boundary conditions. Outflow, Periodic and Reflection.

```
#include "main.h"
```

Functions

- void Outflow (double ∗B)
- void Reflection (double ∗B)
- void Periodic (double ∗B)

5.3.1 Detailed Description

Author

Alejandro Aguayo-Ortiz

5.3.2 Function Documentation

5.3.2.1 Outflow()

```
void Outflow (
             double * B )
```

The function **Outflow()**, receives the vector solution as an parameter **B**. It fills the value of the ghost cells in the specified direction using the value of the last computed cell of the domain.

5.3.2.2 Periodic()

```
void Periodic (
             double * B )
```

The function **Periodic()**, receives the vector solution as an parameter **B**. It fills the value of the ghost cells with the values of the correspondent other side of the domain.

5.3.2.3 Reflection()

```
void Reflection (
             double * B )
```

The function **Reflection()**, receives the vector solution as an parameter **B**. It fills the value of the ghost cells in the specified direction using the value of the mirrored cells, and for the velocity it changes sign.

5.4 EOS/eos.c File Reference

Equation of state.

```
#include "main.h"
```

Functions

- void **EoS** (eos_ *eos, double *u, gauge_ local_grid)

5.4.1 Detailed Description

Author

Alejandro Aguayo-Ortiz

5.5 flux.c File Reference

Numerical flux computing and implementation.

```
#include "main.h"
```

Functions

- int **Flux1D** (vec_ *v, lim_ *l, int *l)
- int **Flux2D** (vec_ *v, lim_ *l, int *l)
- int **Flux3D** (vec_ *v, lim_ *l, int *l)
- int **Hll** (double *F, flx_ *f, int x)
- int **Hllc** (double *F, flx_ *f, int x)

5.5.1 Detailed Description

Author

Alejandro Aguayo-Ortiz

The functions receives as an argument the structures **vec_** and **lim_** and the integer vector **l**. The structures carries the values of the Numerical Fluxes, e.g. **Fp** = $\mathcal{F}_{i+1/2}$.

5.6 HD/fvector.c File Reference

```
#include "main.h"
```

Functions

- void **Prim2FluxF** (double *f, double *v, double *u, gauge_ local_grid)

5.6.1 Detailed Description

Author

Alejandro Aguayo-Ortiz

5.7 HD/gvector.c File Reference

```
#include "main.h"
```

Functions

- void **Prim2FluxG** (double *f, double *v, double *u, gauge_ local_grid)

5.7.1 Detailed Description

Author

Alejandro Aguayo-Ortiz

5.8 HD/hvector.c File Reference

```
#include "main.h"
```

Functions

- void **Prim2FluxH** (double *f, double *v, double *u, gauge_ local_grid)

5.8.1 Detailed Description

Author

Alejandro Aguayo-Ortiz

5.9 RHD/q2uvector.c File Reference

Function that convert from Conservative to Primitives (RHD).

```
#include "main.h"
```

Functions

- int **Cons2Prim** (double *u, double *q)

5.9.1 Detailed Description

Author

Alejandro Aguayo-Ortiz

5.10 HD/qvector.c File Reference

Function that converts Primitives to Conservative variables (HD).

```
#include "main.h"
```

Functions

- void **Prim2Cons** (double *q, double *u, gauge_ local_grid)

5.10.1 Detailed Description

Author

Alejandro Aguayo-Ortiz

5.11 Headers/boundaries.h File Reference

Boundary condition functions definitions.

Functions

- void Outflow (double *B)
- void Periodic (double *B)
- void Reflection (double *B)

5.11.1 Detailed Description

Author

Alejandro Aguayo-Ortiz

5.11.2 Function Documentation

5.11.2.1 Outflow()

```
void Outflow (
            double * B )
```

The function **Outflow()**, receives the vector solution as an parameter **B**. It fills the value of the ghost cells in the specified direction using the value of the last computed cell of the domain.

5.11.2.2 Periodic()

```
void Periodic (
            double * B )
```

The function **Periodic()**, receives the vector solution as an parameter **B**. It fills the value of the ghost cells with the values of the correspondent other side of the domain.

5.11.2.3 Reflection()

```
void Reflection (
            double * B )
```

The function **Reflection()**, receives the vector solution as an parameter **B**. It fills the value of the ghost cells in the specified direction using the value of the mirrored cells, and for the velocity it changes sign.

5.12 Headers/const.h File Reference

Physical and numerical constants.

Macros

- #define **G_cgs** 6.67408e-08
- #define **M_sol_cgs** 1.98855e-33
- #define **R_sol_cgs** 6.957e-10
- #define **K_B_cgs** 1.3806488e-16
- #define **mH_cgs** 1.66e-24
- #define **c_cgs** 3e+10
- #define **G_cgs_earth** 4∗(3.14159265358979323846)∗(3.14159265358979323846)
- #define **yr_cgs** 3.14e+07

5.12.1 Detailed Description

Author

Alejandro Aguayo-Ortiz

5.13 Headers/io.h File Reference

Input and output function and variable definitions.

Functions

- int **PrintValues** (double ∗tprint, double ∗dtprint, int ∗itprint)
- int **Output1** (int ∗itprint)
- int **Output2** (int ∗itprint)
- int **Output3** (int ∗itprint)
- int **Output1_bin** (int ∗itprint)
- int **Output2_bin** (int ∗itprint)
- int **Output3_bin** (int ∗itprint)
- void **Restart** ()
- void **Restart_Bin** ()
- int **Read_Parameters_File** (char const ∗paramfile_name)
- int **User_Parameters** (char const ∗paramfile_name)

Variables

- int **binary**
- int **check_param**
- int **restart_simulation**
- int **restart_filecount**
- char **paramfile_name** [50]
- char **outputdirectory** [50]
- char **outputfile** [50]
- char **restartfile** [50]

5.13.1 Detailed Description

Author

Alejandro Aguayo-Ortiz

5.14 Headers/limiters.h File Reference

Reconstruction variables and functions definitions.

Classes

- struct lim_

Functions

- double **Limiter** (double A, double B, int r)
- double **Godunov** (double A, double B)
- double **Maxmod** (double A, double B)
- double **Minmod** (double A, double B)
- double **Mc** (double A, double B)
- double **Superbee** (double A, double B)
- double **Weno5** (double v1, double v2, double v3, double v4, double v5)
- int **Reconst1D** (double *u, lim_ *l, int *l)
- int **Reconst2D** (double *u, lim_ *l, int *l)
- int **Reconst3D** (double *u, lim_ *l, int *l)

5.14.1 Detailed Description

Author

Alejandro Aguayo-Ortiz

5.15 Headers/macros.h File Reference

Macros definitios for *aztekas*.

Macros

- #define **MIN**(a, b) (((a)<(b))?(a):(b))
- #define **MAX**(a, b) (((a)>(b))?(a):(b))
- #define **TRUE** 1
- #define **FALSE** 0
- #define **HD** 0
- #define **RHD** 1
- #define **IDEAL** 0
- #define **DUST** 1
- #define **STIFF** 2
- #define **CARTESIAN** 0
- #define **CYLINDRICAL** 1
- #define **SPHERICAL** 2
- #define **UNIFORM** 0
- #define **LOGMESH** 1
- #define **HLL** 0
- #define **HLLC** 1
- #define **STANDARD** 0
- #define **PVRS** 1
- #define **GODUNOV** 0
- #define **MINMOD** 1
- #define **MC** 2
- #define **SUPERBEE** 3
- #define **WENO5** 4
- #define **RHO** 0
- #define **PRE** 1
- #define **VX1** 2
- #define **VX2** 3
- #define **VX3** 4

5.15.1 Detailed Description

Author

Alejandro Aguayo-Ortiz

5.16 Headers/main.h File Reference

Main function, headers and variable declaration.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <omp.h>
#include "physics.h"
#include "boundaries.h"
#include "limiters.h"
#include "const.h"
#include "macros.h"
#include "io.h"
#include "user_param.h"
```

Functions

- void Allocate_Array ()
- void **New_Size** ()
- void **Initial** ()
- int **Boundaries** (double *B)
- void **Integration** ()
- int **RK1D** (double *u, double *q, double *q1, double *q2, int order)
- int **RK2D** (double *u, double *q, double *q1, double *q2, int order)
- int **RK3D** (double *u, double *q, double *q1, double *q2, int order)
- int **Flux1D** (vec_ *v, lim_ *l, int *l)
- int **Flux2D** (vec_ *v, lim_ *l, int *l)
- int **Flux3D** (vec_ *v, lim_ *l, int *l)
- int **Hll** (double *F, flx_ *f, int x)
- int **Hllc** (double *F, flx_ *f, int x)
- int **AMATRIX1D** (double *u, vec_ *v, int *l)
- int **AMATRIX2D** (double *u, vec_ *v, int *l)
- int **AMATRIX3D** (double *u, vec_ *v, int *l)
- int **VECTOR** (int pm, char flux, lim_ *l, flx_ *f, int *l)
- int **MxV** (double *M, double *V, double *L)
- void **RoundGen** (double *num)
- void **CheckSimParameters** ()

Variables

- int **CHECK_NAN**
- double * **U**
- double * **U0**
- double * **U1**
- double * **U2**
- double * **U3**
- double * **Q**
- double * **Q1**
- double * **Q2**
- double * **Q3**
- double **start**
- double **delta**
- double **K**
- int **Nx1**
- int **Nx2**
- int **Nx3**
- double **x1max**
- double **x2max**
- double **x3max**
- double **x1min**
- double **x2min**
- double **x3min**

5.16.1 Detailed Description

Author

Alejandro Aguayo-Ortiz

5.16.2 Function Documentation

5.16.2.1 Allocate_Array()

```
void Allocate_Array ( )
```

This function allocates the space in memory for all the vectors used in *aztekas*.

5.17 input.c File Reference

Important input parameters for *aztekas*.

```
#include "main.h"
```

Functions

- int **Read_Parameters_File** (char const ∗paramfile_name)

Variables

- FILE ∗ **paramfile**

5.17.1 Detailed Description

Author

Emilio Tejeda

5.18 integration.c File Reference

Main function for the time integration in the conservative variables $\mathbf{Q}$.

```
#include "main.h"
```

Functions

- void **Integration** ()

5.18.1 Detailed Description

Author

Alejandro Aguayo-Ortiz

5.19 main.c File Reference

Main file of aztekas.

```
#include "main.h"
```

Functions

- int **main** (int argc, char ∗argv[ ])

5.19.1 Detailed Description

Author

Alejandro Aguayo-Ortiz.

5.20 output.c File Reference

Output functions: ASCII and Binary.

```
#include "main.h"
```

Functions

- int **PrintValues** (double ∗tprint, double ∗dtprint, int ∗itprint)

5.20.1 Detailed Description

Authors

Alejandro Aguayo-Oritz and Emilio Tejeda

5.21 restart.c File Reference

Functions to restart from a given file.

```
#include "main.h"
```

Functions

- void **Restart** ()
- void **Restart_Bin** ()

5.21.1 Detailed Description

Author

Emilio Tejeda

5.22 timestep.c File Reference

Time-step calculation.

```
#include "main.h"
```

Functions

- double **TimeStep** ()

5.22.1 Detailed Description

Author

Alejandro Aguayo-Ortiz

Index