Meudon initial data for binary black holes

Contents

1 **Bin_BH — *Binary black hole configuration on a Cartesian grid.*** 4
Class Graph 19

Meudon data represents quasistationary binary black configurations, obtained by P. Grandclément, E. Gourgoulhon & S. Bonazzola, Phys. Rev. D **65**, 044021 (2002).

The exportation of this data, computed by means of LORENE on a multi-domain spectral grid, onto a Cartesian grid (e.g. for CACTUS), is performed by means of the C++ class `Bin_BH`. The class `Bin_BH` comes along with LORENE distribution. This class is very simple, with all data members being public. A typical example of use is the following one

```
 *     // Define the Cartesian grid by means of the arrays xg, yg, zg:
 *     for (int i=0; i<nb_points; i++) {
 *           xg[i] = ...
 *           yg[i] = ...
 *           zg[i] = ...
 *     }
 *
 *     // Read the file containing the spectral data and evaluate
 *     //  all the fields on the Cartesian grid :
 *
 *     Bin_BH binary_system(nb_points, xg, yg, zg, fill, datafile) ;
 *
 *     // Extract what you need :
 *
 *     double* gamma_xx = binary_system.g_xx ; // metric coefficient g_xx
 *
 *     double* shift_x = binary_system.beta_x ; // x comp. of shift vector
 *
 *     ...
 *
 *     // Save everything in an ASCII file :
 *
 *     ofstream file_ini("ini.d") ;
 *     binary_system.save_form(file_ini) ;
 *     file_ini.close() ;
 *
 *
```

1 class **Bin_BH**

Binary black hole configuration on a Cartesian grid.

Public Members

1.1	double	**omega**	*Orbital angular velocity [unit: a^{-1}]*	8
1.2	double	**dist**	*Distance between the coordinate centers of two black holes [unit: a]*	8
1.3	double	**radius2**	*Coordinate radius of the apparent horizon (throat) of black hole 2 [unit: a].*	8
1.4	int	**np**	*Total number of grid points*	8
1.5	double*	**xx**	*1-D array storing the values of coordinate x of the* `np` *grid points [unit: a]*	9
1.6	double*	**yy**	*1-D array storing the values of coordinate y of the* `np` *grid points [unit: a]*	9
1.7	double*	**zz**	*1-D array storing the values of coordinate z of the* `np` *grid points [unit: a]*	9
1.8	double*	**nnn**	*Lapse function N at the* `np` *grid points (1-D array)*	9
1.9	double*	**beta_x**	*Component β^x of the shift vector of corotating coordinates [unit: c]*	10
1.10	double*	**beta_y**	*Component β^y of the shift vector of corotating coordinates [unit: c]*	10
1.11	double*	**beta_z**	*Component β^z of the shift vector of corotating coordinates [unit: c]*	10
1.12	double*	**g_xx**	*Metric coefficient γ_{xx} at the grid points (1-D array)*	10

1.13	double*	**g_xy**	*Metric coefficient γ_{xy} at the grid points (1-D array)*	11
1.14	double*	**g_xz**	*Metric coefficient γ_{xz} at the grid points (1-D array)*	11
1.15	double*	**g_yy**	*Metric coefficient γ_{yy} at the grid points (1-D array)*	11
1.16	double*	**g_yz**	*Metric coefficient γ_{yz} at the grid points (1-D array)*	11
1.17	double*	**g_zz**	*Metric coefficient γ_{zz} at the grid points (1-D array)*	12
1.18	double*	**k_xx**	*Component K_{xx} of the extrinsic curvature at the grid points (1-D array) [unit: c/a]*	12
1.19	double*	**k_xy**	*Component K_{xy} of the extrinsic curvature at the grid points (1-D array) [unit: c/a]*	12
1.20	double*	**k_xz**	*Component K_{xz} of the extrinsic curvature at the grid points (1-D array) [unit: c/a]*	13
1.21	double*	**k_yy**	*Component K_{yy} of the extrinsic curvature at the grid points (1-D array) [unit: c/a]*	13
1.22	double*	**k_yz**	*Component K_{yz} of the extrinsic curvature at the grid points (1-D array) [unit: c/a]*	13
1.23	double*	**k_zz**	*Component K_{zz} of the extrinsic curvature at the grid points (1-D array) [unit: c/a]*	13
1.24	double*	**dpsi_x**	*First derivative $\partial/\partial x$ of the conformal factor Ψ [unit: a^{-1}]*	14
1.25	double*	**dpsi_y**	*First derivative $\partial/\partial y$ of the conformal factor Ψ [unit: a^{-1}]*	14
1.26	double*	**dpsi_z**	*First derivative $\partial/\partial z$ of the conformal factor Ψ [unit: a^{-1}]*	14
1.27	double*	**d2psi_xx**	*Second derivative $\partial^2/\partial x^2$ of the conformal factor Ψ [unit: a^{-2}]*	14
1.28	double*	**d2psi_xy**	*Second derivative $\partial^2/\partial x\partial y$ of the conformal factor Ψ [unit: a^{-2}]*	15

1.29	double*	**d2psi_xz**	*Second derivative $\partial^2/\partial x \partial z$ of the conformal factor Ψ [unit: a^{-2}]* .	15
1.30	double*	**d2psi_yy**	*Second derivative $\partial^2/\partial y^2$ of the conformal factor Ψ [unit: a^{-2}]* .	15
1.31	double*	**d2psi_yz**	*Second derivative $\partial^2/\partial y \partial z$ of the conformal factor Ψ [unit: a^{-2}]* .	15
1.32	double*	**d2psi_zz**	*Second derivative $\partial^2/\partial z^2$ of the conformal factor Ψ [unit: a^{-2}]* .	16
1.33		**Bin_BH** (int nbpoints, const double* xi, const double* yi, const double* zi, int fill, const char* filename, bool mdiff=false)	*Constructor from Lorene spectral data.*	16
1.34		**Bin_BH** (FILE*)	*Constructor from a binary file (previously created by `save_bin`)*	17
1.35		**Bin_BH** (ifstream&)	*Constructor from a formatted file (previously created by `save_form`)*	17
1.36		**~Bin_BH** ()	*Destructor*	17
1.38	void	**save_bin** (FILE*) const	*Save in a binary file.*	17
1.39	void	**save_form** (ofstream&) const	*Save in a formatted file.*	18

Private Members

1.37	void	**alloc_memory** ()	*Allocate the memory for the arrays g_ij, k_ij, etc*	18

Binary black hole configuration on a Cartesian grid.

A binary black hole system is constructed on a Cartesian grid from data stored in a file resulting from a computation by Grandclement, Gourgoulhon and Bonazzola, Phys. Rev. D 65, 044021 (2002).

All the quantities are in units derived from the length scale defined by the coordinate radius a of black hole 1 apparent horizon (throat).

Importation of Lorene data is performed by means of the constructor

`Bin_BH::Bin_BH(int, const double*, const double*, const double*, const char*)`. This constructor takes general arrays for the location of the Cartesian coordinates (x, y, z), i.e. it does not assume that the grid is a uniform one. Note also that these arrays are 1-D, as well as all the metric fields, in order to be use with any ordering of the 3-D storage.

This class is very simple, with all data members being public. A typical example of use is the following one

```
 *     // Define the Cartesian grid by means of the arrays xg, yg, zg:
 *     for (int i=0; i<nb_points; i++) {
 *           xg[i] = ...
 *           yg[i] = ...
 *           zg[i] = ...
 *     }
 *
 *     // Read the file containing the spectral data and evaluate
 *     //  all the fields on the Cartesian grid :
 *
 *     Bin_BH binary_system(nb_points, xg, yg, zg, fill, datafile) ;
 *
 *     // Extract what you need :
 *
 *     double* gamma_xx = binary_system.g_xx ; // metric coefficient g_xx
 *
 *     double* shift_x = binary_system.beta_x ; // x comp. of shift vector
 *
 *     ...
 *
 *     // Save everything in an ASCII file :
 *
 *     ofstream file_ini("ini.d") ;
 *     binary_system.save_form(file_ini) ;
 *     file_ini.close() ;
 *
 *
```

Version: `$Id: bin_bh.h,v 1.10 2010/07/14 16:47:30 e_gourgoulhon Exp $`

1.1

double **omega**

Orbital angular velocity [unit: a^{-1}]

Orbital angular velocity [unit: a^{-1}]

1.2

double **dist**

Distance between the coordinate centers of two black holes [unit: a]

Distance between the coordinate centers of two black holes [unit: a]

1.3

double **radius2**

Coordinate radius of the apparent horizon (throat) of black hole 2 [unit: a].

Coordinate radius of the apparent horizon (throat) of black hole 2 [unit: a]. NB: The coordinate radius of black hole 1 is 1 by definition of the length unit.

1.4

int **np**

Total number of grid points

Total number of grid points

1.5

double* **xx**

1-D array storing the values of coordinate x of the `np` *grid points [unit: a]*

1-D array storing the values of coordinate x of the `np` grid points [unit: a]

1.6

double* **yy**

1-D array storing the values of coordinate y of the `np` *grid points [unit: a]*

1-D array storing the values of coordinate y of the `np` grid points [unit: a]

1.7

double* **zz**

1-D array storing the values of coordinate z of the `np` *grid points [unit: a]*

1-D array storing the values of coordinate z of the `np` grid points [unit: a]

1.8

double* **nnn**

Lapse function N at the `np` *grid points (1-D array)*

Lapse function N at the `np` grid points (1-D array)

1.9

double* **beta_x**

Component β^x of the shift vector of corotating coordinates [unit: c]

Component β^x of the shift vector of corotating coordinates [unit: c]

1.10

double* **beta_y**

Component β^y of the shift vector of corotating coordinates [unit: c]

Component β^y of the shift vector of corotating coordinates [unit: c]

1.11

double* **beta_z**

Component β^z of the shift vector of corotating coordinates [unit: c]

Component β^z of the shift vector of corotating coordinates [unit: c]

1.12

double* **g_xx**

Metric coefficient γ_{xx} at the grid points (1-D array)

Metric coefficient γ_{xx} at the grid points (1-D array)

1.13

double* **g_xy**

Metric coefficient γ_{xy} at the grid points (1-D array)

Metric coefficient γ_{xy} at the grid points (1-D array)

1.14

double* **g_xz**

Metric coefficient γ_{xz} at the grid points (1-D array)

Metric coefficient γ_{xz} at the grid points (1-D array)

1.15

double* **g_yy**

Metric coefficient γ_{yy} at the grid points (1-D array)

Metric coefficient γ_{yy} at the grid points (1-D array)

1.16

double* **g_yz**

Metric coefficient γ_{yz} at the grid points (1-D array)

Metric coefficient γ_{yz} at the grid points (1-D array)

1.17

double* **g_zz**

Metric coefficient γ_{zz} at the grid points (1-D array)

Metric coefficient γ_{zz} at the grid points (1-D array)

1.18

double* **k_xx**

Component K_{xx} of the extrinsic curvature at the grid points (1-D array) [unit: c/a]

Component K_{xx} of the extrinsic curvature at the grid points (1-D array) [unit: c/a]

1.19

double* **k_xy**

Component K_{xy} of the extrinsic curvature at the grid points (1-D array) [unit: c/a]

Component K_{xy} of the extrinsic curvature at the grid points (1-D array) [unit: c/a]

1.20

double* **k_xz**

Component K_{xz} of the extrinsic curvature at the grid points (1-D array) [unit: c/a]

Component K_{xz} of the extrinsic curvature at the grid points (1-D array) [unit: c/a]

1.21

double* **k_yy**

Component K_{yy} of the extrinsic curvature at the grid points (1-D array) [unit: c/a]

Component K_{yy} of the extrinsic curvature at the grid points (1-D array) [unit: c/a]

1.22

double* **k_yz**

Component K_{yz} of the extrinsic curvature at the grid points (1-D array) [unit: c/a]

Component K_{yz} of the extrinsic curvature at the grid points (1-D array) [unit: c/a]

1.23

double* **k_zz**

Component K_{zz} of the extrinsic curvature at the grid points (1-D array) [unit: c/a]

Component K_{zz} of the extrinsic curvature at the grid points (1-D array) [unit: c/a]

1.24

double* **dpsi_x**

First derivative $\partial/\partial x$ of the conformal factor Ψ [unit: a^{-1}]

First derivative $\partial/\partial x$ of the conformal factor Ψ [unit: a^{-1}]

1.25

double* **dpsi_y**

First derivative $\partial/\partial y$ of the conformal factor Ψ [unit: a^{-1}]

First derivative $\partial/\partial y$ of the conformal factor Ψ [unit: a^{-1}]

1.26

double* **dpsi_z**

First derivative $\partial/\partial z$ of the conformal factor Ψ [unit: a^{-1}]

First derivative $\partial/\partial z$ of the conformal factor Ψ [unit: a^{-1}]

1.27

double* **d2psi_xx**

Second derivative $\partial^2/\partial x^2$ of the conformal factor Ψ [unit: a^{-2}]

Second derivative $\partial^2/\partial x^2$ of the conformal factor Ψ [unit: a^{-2}]

1.28

double* **d2psi_xy**

Second derivative $\partial^2/\partial x \partial y$ of the conformal factor Ψ [unit: a^{-2}]

Second derivative $\partial^2/\partial x \partial y$ of the conformal factor Ψ [unit: a^{-2}]

1.29

double* **d2psi_xz**

Second derivative $\partial^2/\partial x \partial z$ of the conformal factor Ψ [unit: a^{-2}]

Second derivative $\partial^2/\partial x \partial z$ of the conformal factor Ψ [unit: a^{-2}]

1.30

double* **d2psi_yy**

Second derivative $\partial^2/\partial y^2$ of the conformal factor Ψ [unit: a^{-2}]

Second derivative $\partial^2/\partial y^2$ of the conformal factor Ψ [unit: a^{-2}]

1.31

double* **d2psi_yz**

Second derivative $\partial^2/\partial y \partial z$ of the conformal factor Ψ [unit: a^{-2}]

Second derivative $\partial^2/\partial y \partial z$ of the conformal factor Ψ [unit: a^{-2}]

1.32

double* **d2psi_zz**

Second derivative $\partial^2/\partial z^2$ of the conformal factor Ψ [unit: a^{-2}]

Second derivative $\partial^2/\partial z^2$ of the conformal factor Ψ [unit: a^{-2}]

1.33

Bin_BH (int nbpoints, const double* xi, const double* yi, const double* zi, int fill, const char* filename, bool mdiff=false)

Constructor from Lorene spectral data.

Constructor from Lorene spectral data.

This constructor takes general arrays `xi, yi, zi` for the location of the Cartesian coordinates (x, y, z), i.e. it does not assume that the grid is a uniform one. These arrays are 1-D to deal with any ordering of a 3-D storage.

Parameters:

`nbpoints`
`xi`

`yi`

`zi`

`fill`

fill $= 0$: all the fields are set to zero
fill $= 1$: the fields are extrapolated from theirvalues "outside" the holes, by mea
`filename`

1.34

Bin_BH (FILE*)

Constructor from a binary file (previously created by `save_bin`*)*

Constructor from a binary file (previously created by `save_bin`)

1.35

Bin_BH (ifstream&)

Constructor from a formatted file (previously created by `save_form`*)*

Constructor from a formatted file (previously created by `save_form`)

1.36

~Bin_BH ()

Destructor

Destructor

1.38

void **save_bin** (FILE*) const

Save in a binary file.

Save in a binary file. This file can be subsenquently read by the evolution code, or by the constructor `Bin_BH::Bin_BH(FILE* )`.

1.39

void **save_form** (ofstream&) const

Save in a formatted file.

Save in a formatted file. This file can be subsenquently read by the evolution code, or by the constructor `Bin_BH::Bin_BH(ifstream& )`.

1.37

void **alloc_memory** ()

Allocate the memory for the arrays g_ij, k_ij, etc

Allocate the memory for the arrays g_ij, k_ij, etc

Class Graph

1 Bin_BH 4