

APC_524

Generated by Doxygen 1.8.12

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	Class Documentation	5
3.1	Boris Class Reference	5
3.2	Domain Class Reference	6
3.3	Field_part Struct Reference	6
3.4	Grid Class Reference	6
3.4.1	Detailed Description	8
3.4.2	Member Function Documentation	8
3.4.2.1	evolveFields()	8
3.4.2.2	getCellID()	8
3.4.2.3	getFieldInterpolatorVec()	9
3.4.2.4	getStepSize()	9
3.5	Input_Info_t Struct Reference	9
3.6	Particle Struct Reference	10
3.7	Particle_Compare Class Reference	10
3.8	Particle_Field_List Class Reference	10
3.9	Pusher Class Reference	11

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Domain	6
Field_part	6
Grid	6
Input_Info_t	9
Particle	10
Particle_Compare	10
Particle_Field_List	10
Pusher	11
Boris	5

Chapter 2

Class Index

2.1 Class List

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

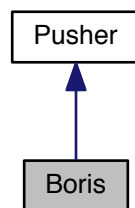
Boris	5
Domain	6
Field_part	6
Grid	
Class representing grid on which E and B fields and currents are defined	6
Input_Info_t	9
Particle	10
Particle_Compare	10
Particle_Field_List	10
Pusher	11

Chapter 3

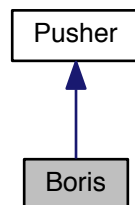
Class Documentation

3.1 Boris Class Reference

Inheritance diagram for Boris:



Collaboration diagram for Boris:



Public Member Functions

- int **Step** ([Particle](#) *part, [Field_part](#) *field, double dt)

The documentation for this class was generated from the following files:

- src/pusher/boris.hpp
- src/pusher/boris.cpp

3.2 Domain Class Reference

Public Member Functions

- **Domain** (int size, int rank, [Input_Info_t](#) *input_info)
- int **getnGhosts** (void)
- int * **getnxyz** (void)
- double * **getxyz0** (void)
- double * **getLxyz** (void)

The documentation for this class was generated from the following files:

- src/domain/domain.hpp
- src/domain/domain.cpp

3.3 Field_part Struct Reference

Public Attributes

- double **e1**
- double **e2**
- double **e3**
- double **b1**
- double **b2**
- double **b3**

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

- src/particles/particle.hpp

3.4 Grid Class Reference

Class representing grid on which E and B fields and currents are defined.

```
#include <grid.hpp>
```

Public Member Functions

- **Grid** (int *nxyz, int nGhosts, double *xyz0, double *Lxyz)
- int **evolveFields** (double dt)
Evolve Electric and Magnetic fields in time.
- int **getFieldInterpolatorVec** (int cellID, double *InterpolatorVec)
Return vector for field interpolation.
- int **getCellID** (double x, double y, double z)
Get cell ID based on particle position.
- int **getNumberOfCells** ()
Get total number of cells.
- double **getStepSize** (int dimension)
Get number of cells along dimension in grid.
- void **updateGhostCells** ()
- int **getGhostVecSize** ()
- void **getGhostVec** (const int side, double *ghostVec)
- void **getGhostVecAlt** (const int side, double *ghostVec)
- void **setGhostVec** (const int side, const double *ghostVec)
- void **setGhostVecAlt** (const int side, const double *ghostVec)
- double **getx0** ()
- double **gety0** ()
- double **getz0** ()
- double **getdx** ()
- double **getdy** ()
- double **getdz** ()
- double **getLx** ()
- double **getLy** ()
- double **getLz** ()

Protected Member Functions

- double *** **newField_** ()
- void **deleteField_** (double ***fieldPt)
- int **sideToIndex_** (const int side)
- void **sliceMatToVec_** (double ***const mat, const int side)
- void **unsliceMatToVec_** (double ***mat, const int side)

Protected Attributes

- const int **nx_**
- const int **ny_**
- const int **nz_**
- const int **nGhosts_**
- const double **x0_**
- const double **y0_**
- const double **z0_**
- const double **Lx_**
- const double **Ly_**
- const double **Lz_**
- const int **iBeg_**
- const int **jBeg_**
- const int **kBeg_**

- const int **iEnd_**
- const int **jEnd_**
- const int **kEnd_**
- const double **dx_**
- const double **dy_**
- const double **dz_**
- const int **nRealPtsYZPlane_**
- const int **nFields_**
- const int **ghostVecSize_**
- double *** **Ex_**
- double *** **Ey_**
- double *** **Ez_**
- double *** **Bx_**
- double *** **By_**
- double *** **Bz_**
- double *** **Bx_tm1_**
- double *** **By_tm1_**
- double *** **Bz_tm1_**
- double *** **Jx_**
- double *** **Jy_**
- double *** **Jz_**
- double * **sliceTmp_**

3.4.1 Detailed Description

Class representing grid on which E and B fields and currents are defined.

[Grid](#) has ghost cells on each face. The ghost cell updating in y and z arises from periodic boundary conditions. x-direction ghost cells allow communication between MPI domains.

Following Yee (1966), electric fields and currents reside on edges, and magnetic fields on faces. Fields are updated using a set of finite-difference equations approximating Ampere's and Faraday's Laws.

A set of getters are available to allow particles to interpolate electric fields based on their position.

3.4.2 Member Function Documentation

3.4.2.1 evolveFields()

```
int Grid::evolveFields (
    double dt )
```

Evolve Electric and Magnetic fields in time.

Uses Yee algorithm to advance E and B fields.

3.4.2.2 getCellID()

```
int Grid::getCellID (
    double x,
    double y,
    double z )
```

Get cell ID based on particle position.

Cell ID is uniquely given by $(ny_nz_)*ix + nz_iy + iz$. Returns -1 if particle is not on grid.

3.4.2.3 getFieldInterpolatorVec()

```
int Grid::getFieldInterpolatorVec (
    int cellID,
    double * InterpolatorVec )
```

Return vector for field interpolation.

Based on cellID, return relevant edge E and face B fields and cell origin, in format [x, y, z, ... Ex(ix, iy, iz), Ex(ix, iy+1, iz), Ex(ix, iy+1, iz+1), Ex(ix, iy, iz+1), ... Ey(ix, iy, iz), Ey(ix, iy, iz+1), Ey(ix+1, iy, iz+1), Ey(ix+1, iy, iz), ... Ez(ix, iy, iz), Ez(ix+1, iy, iz), Ez(ix+1, iy+1, iz), Ez(ix, iy+1, iz), ... Bx(ix, iy, iz), Bx(ix+1, iy, iz), ... By(ix, iy, iz), By(ix, iy+1, iz), ... Bz(ix, iy, iz), Bz(ix, iy, iz+1), ...] where ix, iy, and iz are the row indices for each of the three dimensions (calculated from the cellID)

3.4.2.4 getStepSize()

```
double Grid::getStepSize (
    int dimension )
```

Get number of cells along dimension in grid.

Returns number of cells along dimension according to; dimension = 0: x dimension = 1: y dimension = 2: z Returns -1 if invalid dimension.

The documentation for this class was generated from the following files:

- src/grid/grid.hpp
- src/grid/grid.cpp
- src/grid/oGrid.cpp
- src/grid/spookyGrid.cpp

3.5 Input_Info_t Struct Reference

Public Attributes

- int **nx**
- int **np**
- int **nt**
- int **restart**
- double **dens**
- double **temp**

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

- src/IO/IO.hpp

3.6 Particle Struct Reference

Public Attributes

- double **x1**
- double **x2**
- double **x3**
- double **v1**
- double **v2**
- double **v3**
- double **q**
- double **m**
- int **my_id**
- short **isGhost**

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

- src/particles/particle.hpp

3.7 Particle_Compare Class Reference

Public Member Functions

- **Particle_Compare** ([Grid](#) *grid)
- bool **operator()** ([Particle](#) const *a, [Particle](#) const *b) const

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

- src/particles/particle_utils.hpp

3.8 Particle_Field_List Class Reference

Public Member Functions

- **Particle_Field_List** (long np)
- void **Load** ()
- void **Push** (double dt)
- void **Pass** ()
- long **nParticles** ()
- void **SortParticles** ([Particle_Compare](#) comp)
- void **setPusher** ([Pusher](#) *pusher)

Public Attributes

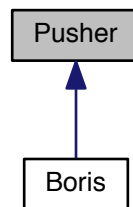
- `std::vector< Particle * > parts_`

The documentation for this class was generated from the following files:

- `src/particles/particle_list.hpp`
- `src/particles/particle_list.cpp`

3.9 Pusher Class Reference

Inheritance diagram for Pusher:



Public Member Functions

- `virtual int Step (Particle *part, Field_part *field, double dt)=0`

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

- `src/pusher/pusher.hpp`

