My Project

Generated by Doxygen 1.9.2

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

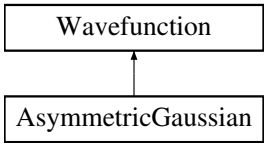
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 AsymmetricGaussian Class Reference

Inheritance diagram for AsymmetricGaussian:



Public Member Functions

- **AsymmetricGaussian** (class System *s, double alpha, double beta, double a)
- double evaluateAll ()

  Evaluates the wavefunction in the specified point.
- double evaluateSing (int part_idx)

  Evaluates the gaussian contribution relative to the part_idx-th particle.
- double analyticalAlphaDerivative ()

  evaluates the analytical derivative with respect to alpha
- double numericalSecondDerivative (int part_idx, int direction, double h)
- vector< double > DriftForce (int part_idx)

  evaluates the drift force associated to the part_idx-th particle

Additional Inherited Members

3.1.1 Member Function Documentation

3.1.1.1 evaluateSing()

```
double AsymmetricGaussian::evaluateSing (
            int part_idx )  [virtual]
```

Evaluates the gaussian contribution relative to the part_idx-th particle.

See also

evaluateAll()

Implements Wavefunction.

3.1.1.2 numericalSecondDerivative()

```
double AsymmetricGaussian::numericalSecondDerivative (
            int part_idx,
            int direction,
            double h )  [virtual]
```

Evaluates numerically the second derivative with respect to the coordinate "direction" of particle "part_idx". The derivative is evaluated in the point in which the particles are in this moment. direction can be 0 (x), 1 (y), 2 (z), accordingly to the dimension of the system chosen.

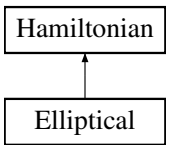
Implements Wavefunction.

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

- Wavefunctions/asymmetricGaussian.h
- Wavefunctions/asymmetricGaussian.cpp

3.2 Elliptical Class Reference

Inheritance diagram for Elliptical:



Public Member Functions

- **Elliptical** (class System *system, double omegaXY, double omegaZ)
- double **getOmegaXY** ()
- double **getOmegaZ** ()
- double **LocalEnergyAnalytic** ()
- double **LocalEnergyNumeric** (double h)

Additional Inherited Members

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

- Hamiltonians/elliptical.h
- Hamiltonians/elliptical.cpp

3.3 Functions Class Reference

Public Member Functions

- **Functions** (class System *system)
- vector< vector< double > > **solve_varying_alpha** (double alpha_min, double alpha_max, int Nalphas)
- double **gradientDescent** (double inizialAlpha, double gamma, double tolerance, int Nmax, int Nsteps)

Public Attributes

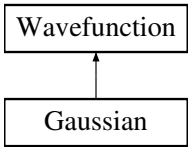
- class System * **system**

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

- Others/functions.h
- Others/functions.cpp

3.4 Gaussian Class Reference

Inheritance diagram for Gaussian:



Public Member Functions

- **Gaussian** (class System *s, double alpha)
- double evaluateAll ()
- double evaluateSing (int part_idx)

  Evaluates the gaussian contribution relative to the part_idx-th particle.
- double numericalSecondDerivative (int part_idx, int direction, double h)
- double analyticalAlphaDerivative ()

  evaluates the analytical derivative with respect to alpha
- vector< double > DriftForce (int part_idx)

  evaluates the drift force associated to the part_idx-th particle

Additional Inherited Members

3.4.1 Member Function Documentation

3.4.1.1 evaluateAll()

```
double Gaussian::evaluateAll ( )  [virtual]
```

See also

Wavefunction::evaluateAll()

Implements Wavefunction.

3.4.1.2 evaluateSing()

```
double Gaussian::evaluateSing (
            int part_idx )  [virtual]
```

Evaluates the gaussian contribution relative to the part_idx-th particle.

See also

evaluateAll()

Implements Wavefunction.

3.4.1.3 numericalSecondDerivative()

```
double Gaussian::numericalSecondDerivative (
            int part_idx,
            int direction,
            double h )  [virtual]
```

Evaluates numerically the second derivative with respect to the coordinate "direction" of particle "part_idx". The derivative is evaluated in the point in which the particles are in this moment. direction can be 0 (x), 1 (y), 2 (z), accordingly to the dimension of the system chosen.

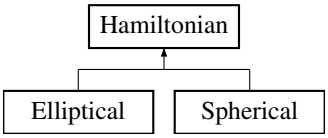
Implements Wavefunction.

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

- Wavefunctions/gaussian.h
- Wavefunctions/gaussian.cpp

3.5 Hamiltonian Class Reference

Inheritance diagram for Hamiltonian:



Public Member Functions

- **Hamiltonian** (class System *system)
- virtual double **LocalEnergyAnalytic** ()=0
- virtual double **LocalEnergyNumeric** (double h)=0

Public Attributes

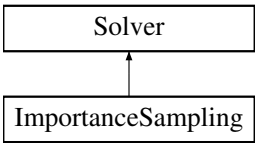
- class System * **system**

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

- Hamiltonians/hamiltonian.h
- Hamiltonians/hamiltonian.cpp

3.6 ImportanceSampling Class Reference

Inheritance diagram for ImportanceSampling:



Public Member Functions

- **ImportanceSampling** (class System *system, int Nsteps, double initialFraction, double dt, double D)
- double **getdt** ()
- double **getD** ()
- void **setdt** (double dt)
- void **setD** (double D)
- vector< double > **solve** (bool allAverages)
- vector< double > **solve** (double h)

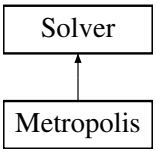
Additional Inherited Members

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

- Solvers/importanceSampling.h
- Solvers/importanceSampling.cpp

3.7 Metropolis Class Reference

Inheritance diagram for Metropolis:



Public Member Functions

- **Metropolis** (class System ∗system, int Nsteps, double initialFraction, double step)
- double **getStep** ()
- double **setStep** (double step)
- vector< double > **solve** (bool allAverages)
- vector< double > **solve** (double h)

Additional Inherited Members

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

- Solvers/metropolis.h
- Solvers/metropolis.cpp

3.8 Particle Class Reference

Public Member Functions

- **Particle** (class System ∗system, double mass, vector< double > pos)
- void **setMass** (double m)
- void **setPosition** (vector< double > new_pos)
- vector< double > **getPosition** ()
- double **getMass** ()
- void **move** (vector< double > delta_pos)

Public Attributes

- class System ∗ **system**

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

- Particles/particle.h
- Particles/particle.cpp

3.9 RandomGenerator Class Reference

Public Attributes

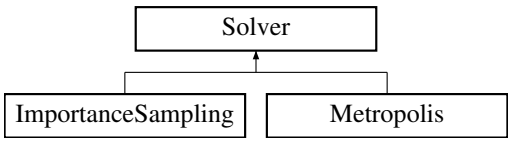
- uniform_real_distribution< double > **uniform**
- normal_distribution< double > **normal**

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

- Others/random_generator.h
- Others/random_generator.cpp

3.10 Solver Class Reference

Inheritance diagram for Solver:



Public Member Functions

- **Solver** (class System ∗system, int Nsteps, double initialFraction)
- int **getNsteps** ()
- double **getInitialFraction** ()
- void **setNsteps** (int Nsteps)
- void **setInitialFraction** (double initialFraction)
- virtual vector< double > **solve** (bool allAverages)=0
- virtual vector< double > **solve** (double h)=0

Public Attributes

- class System ∗ **system**

Protected Attributes

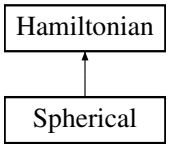
- int **Nsteps**
- double **InitialFraction**

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

- Solvers/solver.h
- Solvers/solver.cpp

3.11 Spherical Class Reference

Inheritance diagram for Spherical:



Public Member Functions

- **Spherical** (class System ∗system, double omega)
- double **getOmega** ()
- double **LocalEnergyAnalytic** ()
- double **LocalEnergyNumeric** (double h)

Additional Inherited Members

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

- Hamiltonians/spherical.h
- Hamiltonians/spherical.cpp

3.12 System Class Reference

Public Member Functions

- **System** (int dim, int Npart)
- class Hamiltonian ∗ **getHamiltonian** ()
- class Wavefunction ∗ **getWavefunction** ()
- class Solver ∗ **getSolver** ()
- class RandomGenerator ∗ **getRandomGenerator** ()
- int **getDimension** ()
- int **getNParticles** ()
- vector< class Particle ∗ > **getParticles** ()

- bool getUseMatrix ()
- void **setHamiltonian** (class Hamiltonian *hamiltonian)
- void **setSolver** (class Solver *solver)
- void **setWavefunction** (class Wavefunction *wavefunction)
- void **setRandomGenerator** (class RandomGenerator *randomgenerator)
- void setUseMatrix (bool usematrix)
- void **addParticle** (double mass, vector< double > pos)
- double r2 (double parameter)

  Evaluates the squared distance of particles and sums.
- double r2 (vector< double > vect, double parameter)

  Evaluates the sum of the square of the vector components.
- double cdot (vector< double > v1, vector< double > v2)

  Scalar product between vector 1 and vector 2.
- void EvaluateRelativeDistance ()

  Updates the relative_distance matrix.
- void EvaluateRelativePosition ()

  Updates the relative position matrix.
- void EvaluateRelativePosition (int idx)

  Updates a special row of the relative position matrix.
- void EvaluateRelativeDistance (int idx)

  Updates a special row of the relative distance matrix.

Public Attributes

- vector< vector< vector< double > > > relative_position

  NxN matrix of 3d vectors. The ij-th element is a 3D vector containing posi - posj.
- vector< vector< double > > relative_distance

  NxN matrix of doubles. The ij-th element is the distance between particle i and particle j.

3.12.1 Member Function Documentation

3.12.1.1 EvaluateRelativeDistance() [1/2]

```
void System::EvaluateRelativeDistance ( )
```

Updates the relative_distance matrix.

See also

relative_distance

3.12.1.2 EvaluateRelativeDistance() [2/2]

```
void System::EvaluateRelativeDistance (
            int idx )
```

Updates a special row of the relative distance matrix.

See also

relative_distance

3.12.1.3 EvaluateRelativePosition() [1/2]

```
void System::EvaluateRelativePosition ( )
```

Updates the relative position matrix.

See also

relative_position

3.12.1.4 EvaluateRelativePosition() [2/2]

```
void System::EvaluateRelativePosition (
            int idx )
```

Updates a special row of the relative position matrix.

See also

relative_position

3.12.1.5 getUseMatrix()

```
bool System::getUseMatrix ( )
```

See also

relative_position, relative_distance

3.12.1.6 setUseMatrix()

```
void System::setUseMatrix (
            bool usematrix )
```

See also

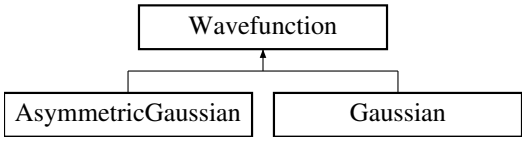
relative_position, relative_distance

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

- System/system.h
- System/system.cpp

3.13 Wavefunction Class Reference

Inheritance diagram for Wavefunction:



Public Member Functions

- **Wavefunction** (class System *system, int nparams)
- virtual double evaluateAll ()=0

  Evaluates the wavefunction in the specified point.
- virtual double evaluateSing (int part_idx)=0

  Evaluates the gaussian contribution relative to the part_idx-th particle.
- virtual double numericalSecondDerivative (int part_idx, int direction, double h)=0
- virtual vector< double > DriftForce (int part_idx)=0

  evaluates the drift force associated to the part_idx-th particle
- virtual double analyticalAlphaDerivative ()=0

  evaluates the analytical derivative with respect to alpha
- void **setParameter** (int idx, double value)
- double **getParameter** (int idx)

Public Attributes

- class System * **s**

Protected Attributes

- int **nparams**
- vector< double > **params**

3.13.1 Member Function Documentation

3.13.1.1 evaluateSing()

```
virtual double Wavefunction::evaluateSing (
            int part_idx )  [pure virtual]
```

Evaluates the gaussian contribution relative to the part_idx-th particle.

See also

evaluateAll()

Implemented in AsymmetricGaussian, and Gaussian.

3.13.1.2 numericalSecondDerivative()

```
virtual double Wavefunction::numericalSecondDerivative (
            int part_idx,
            int direction,
            double h )  [pure virtual]
```

Evaluates numerically the second derivative with respect to the coordinate "direction" of particle "part_idx". The derivative is evaluated in the point in which the particles are in this moment. direction can be 0 (x), 1 (y), 2 (z), accordingly to the dimension of the system chosen.

Implemented in AsymmetricGaussian, and Gaussian.

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

- Wavefunctions/wavefunction.h
- Wavefunctions/wavefunction.cpp

Index