Hubbard Model Exact Diagonalization

Generated by Doxygen 1.8.11

Contents

Chapter 1

Hubbard model exact diagonalization

References:

1. H. Q. Lin, J. E. Gubernatis, Harvey Gould, and Tobochnik, computers in physics, 7,400(1993)
2. S. A. Jafari, IJPR, 8,113,(2008)

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 basis Class Reference

Public Member Functions

- **basis** (long, long, long)
- const basis & **operator=** (const basis &)
- long **hopping_up** (long, long, long)
- long **hopping_down** (long, long, long)
- long **potential** (long, long, long)
- long **onsite_up** (long, long)
- long **onsite_down** (long, long)
- long **factorial** (long, long)
- void **init** ()
- void **generate_up** (long)
- void **generate_down** (long)
- long **creation** (long, long)
- long **annihilation** (long, long)
- void **prlong** ()

Public Attributes

- long **nsite**
- long **nel_up**
- long **nel_down**
- map< long, long > **basis_up**
- map< long, long > **basis_down**
- long **nbasis_up**
- long **nbasis_down**
- vector< long > **id_up**
- vector< long > **id_down**

Friends

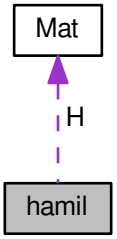
- ostream & **operator**<< (ostream &os, const basis &)

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

- basis.h
- basis.cpp

3.2 hamil Class Reference

Collaboration diagram for hamil:



Public Member Functions

- hamil (basis &sector, double t, double U)
- double ground_state_energy ()
- void diag ()
- complex< double > **Greens_function** (double, double)
- void print_hamil ()
- void print_eigen ()

Public Attributes

- long nHilbert
- unsigned seed
- Mat H
- std::vector< double > eigenvalues
- std::vector< double > psi_0
- std::vector< double > psi_n0

3.2.1 Constructor & Destructor Documentation

3.2.1.1 hamil::hamil (basis & *sector,* double *t,* double *U*)

Initialize the hamiltonian matrix elements.

Parameters

sector	basis sector,
t	hopping strength,
U	onsite replusive interaction strength

3.2.2 Member Function Documentation

3.2.2.1 void hamil::diag ( )

Diagonalize the full hamiltonian

3.2.2.2 double hamil::ground_state_energy ( )

Return the ground state energy of the system

3.2.2.3 void hamil::print_eigen ( )

Print the eigenvalues of the system

3.2.2.4 void hamil::print_hamil ( )

Print the hamiltonian matrix

3.2.3 Member Data Documentation

3.2.3.1 std::vector<double> hamil::eigenvalues

Eigenvalues of the hamiltonian

3.2.3.2 Mat hamil::H

Hamiltonian matrix in CSR format

3.2.3.3 long hamil::nHilbert

Size of the Hilbert space

3.2.3.4 std::vector<double> hamil::psi_0

Ground state wave function

3.2.3.5 std::vector<double> hamil::psi_n0

First element of all wave functions

3.2.3.6 unsigned hamil::seed

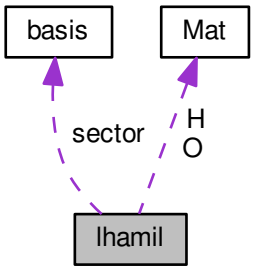
seed for the RNGs

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

- hamiltonian.h
- hamiltonian.cpp

3.3 lhamil Class Reference

Collaboration diagram for lhamil:



Public Member Functions

- **lhamil** (const Mat &, long, long, unsigned)
- **lhamil** (basis &, double, double, long, unsigned)
- void **set_hamil** (basis &, double, double)
- void **psir0_creation_el_up** (basis &, basis &, vector< double > &, long)
- void **psir0_creation_el_down** (basis &, basis &, vector< double > &, long)
- void **psir0_annihilation_el_up** (basis &, basis &, vector< double > &, long)
- void **psir0_annihilation_el_down** (basis &, basis &, vector< double > &, long)
- void **set_onsite_optc** (int r, int alpha, int annil)
- void **coeff_update** ()
- void **coeff_explicit_update** ()
- void **coeff_update_wopt** (vector< double >)
- void **diag** ()
- void **diag** (int)
- void **eigenstates_reconstruction** ()
- double **ground_state_energy** ()
- double **spectral_function** (double omega, double eta)
- double **spectral_function** (double omega, double eta, int annil)
- complex< double > **Greens_function** (double omega, double eta, int annil)
- complex< double > **Greens_function_ij_ab** (int i, int j, int alpha, int beta, double E, double eta)
- complex< double > **Greens_function_k** (int k, int alpha, int beta, double E, double eta)
- void **print_hamil** ()
- void **print_lhamil** (int)
- void **print_eigen** (int)
- void **save_to_file** (const char *)
- void **read_from_file** (const char *)

Public Attributes

- unsigned seed

  Seed for RNGs.
- long nHilbert

  Hilbert space size.
- long lambda

  Lanczos update steps.
- double E0

  Ground state eigen energy.
- basis sector

  Basis.
- Mat H

  Hamiltonian matrix in CSR format.
- Mat O

  Operator matrix in CSR format.
- std::vector< double > norm

  Normalization coefficients vector in Lanczos update.
- std::vector< double > overlap

  Overlap coefficients vector in Lanczos update.
- std::vector< double > psir_0

  Ground state wave function in real-space.
- std::vector< double > psi_0

  Ground state eigenvector in Krylov subspace.
- std::vector< double > psi_n0

  First element of eigenvectors in Krylov subspace.
- std::vector< double > eigenvalues

  Eigenvalues.

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

- lanczos_hamiltonian.h
- lanczos_hamiltonian.cpp

3.4 Mat Class Reference

Public Member Functions

- **Mat** (const Mat &rhs)
- Mat & **operator=** (const Mat &rhs)
- Vec **operator*** (const Vec &) const
- vector< double > **operator*** (const vector< double > &) const
- void **init** (const vector< long > &, const vector< long > &, const vector< double > &)
- void **clear** ()
- void **print** ()

Public Attributes

- std::vector< long > **outer_starts**
- std::vector< long > **inner_indices**
- std::vector< double > **value**

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

- matrix.h
- matrix.cpp

3.5 Timer Class Reference

Public Member Functions

- double **elapsed** ()
- unsigned long **nanoseconds** ()
- void **reset** ()

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

- init.h

3.6 Vec Class Reference

Public Member Functions

- **Vec** (long _size)
- **Vec** (long _size, const double _init)
- **Vec** (const Vec &rhs)
- void **assign** (long _size, const double _init)
- void **init_random** (unsigned)
- void **init_random** (long, unsigned)
- void **clear** ()
- double **normalize** ()
- Vec & **operator=** (const Vec &rhs)
- Vec & **operator-=** (const Vec &rhs)
- Vec & **operator+=** (const Vec &rhs)
- Vec & **operator*=** (const double &rhs)
- Vec & **operator/=** (const double &rhs)
- Vec **operator+** (const Vec &)
- Vec **operator-** (const Vec &)
- Vec **operator*** (const double &)
- Vec **operator/** (const double &)
- double **operator*** (const Vec &)

Public Attributes

- std::vector< double > **value**
- long **size**

Friends

- ostream & **operator**<< (ostream &os, const Vec &)

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

- matrix.h
- matrix.cpp

Index