

MP2 Calc

Generated by Doxygen 1.9.8

1 Computation of the MP2 energy	1
1.1 Theory	1
1.2 Dependencies	1
1.2.1 HDF5	1
1.2.2 TREXIO	1
1.3 Installation and usage	2
1.4 Example	2
2 Installation	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Class Documentation	9
5.1 Mol Struct Reference	9
5.1.1 Detailed Description	10
6 File Documentation	11
6.1 src/hf.c File Reference	11
6.2 src/hf.h File Reference	11
6.2.1 Detailed Description	12
6.2.2 Function Documentation	12
6.2.2.1 get_eri_sum()	12
6.2.2.2 get_hcore_sum()	12
6.2.2.3 hf_kernel()	12
6.2.2.4 match_eri()	13
6.3 hf.h	13
6.4 src/main.c File Reference	14
6.4.1 Detailed Description	14
6.5 src/mol.c File Reference	14
6.5.1 Detailed Description	15
6.5.2 Function Documentation	15
6.5.2.1 gen_eri_index()	15
6.5.2.2 get_eri_energy()	16
6.5.2.3 get_nth_eri()	16
6.5.2.4 mol_destruct()	16
6.5.2.5 mol_init()	17
6.6 src/mol.h File Reference	17
6.6.1 Detailed Description	18
6.6.2 Function Documentation	18
6.6.2.1 gen_eri_index()	18

6.6.2.2 get_eri_energy()	18
6.6.2.3 get_nth_eri()	19
6.6.2.4 mol_destruct()	19
6.6.2.5 mol_init()	19
6.7 mol.h	20
6.8 src/mp2.c File Reference	20
6.8.1 Detailed Description	20
6.8.2 Function Documentation	21
6.8.2.1 cal_mp2_correlation()	21
6.9 src/mp2.h File Reference	21
6.9.1 Detailed Description	21
6.9.2 Function Documentation	22
6.9.2.1 cal_mp2_correlation()	22
6.10 mp2.h	22
6.11 src/reader.c File Reference	22
6.11.1 Detailed Description	23
6.11.2 Function Documentation	23
6.11.2.1 read_electron_up_num()	23
6.11.2.2 read_mo_1e_int_h_core()	24
6.11.2.3 read_mo_2e_int_eri()	24
6.11.2.4 read_mo_2e_int_eri_size()	24
6.11.2.5 read_mo_energy()	25
6.11.2.6 read_mo_num()	25
6.11.2.7 read_nucleus_repulsion()	25
6.12 src/reader.h File Reference	25
6.12.1 Detailed Description	26
6.12.2 Function Documentation	26
6.12.2.1 read_electron_up_num()	26
6.12.2.2 read_mo_1e_int_h_core()	27
6.12.2.3 read_mo_2e_int_eri()	27
6.12.2.4 read_mo_2e_int_eri_size()	27
6.12.2.5 read_mo_energy()	28
6.12.2.6 read_mo_num()	28
6.12.2.7 read_nucleus_repulsion()	28
6.13 reader.h	28
6.14 src/util.c File Reference	29
6.14.1 Detailed Description	29
6.14.2 Function Documentation	30
6.14.2.1 index_2e()	30
6.14.2.2 print_matrix()	30
6.14.2.3 sum_diag()	30
6.15 src/util.h File Reference	31

6.15.1 Detailed Description	31
6.15.2 Function Documentation	32
6.15.2.1 index_2e()	32
6.15.2.2 print_matrix()	32
6.15.2.3 sum_diag()	32
6.16 util.h	33
Index	35

Chapter 1

Computation of the MP2 energy

This program reads the .h5 file and computes the MP2 energy.

1.1 Theory

The MP2 energy is given by the formula:

$$E_{\text{MP2}} = E_{\text{HF}} + \sum_{(i,j) \in \text{occupied}} \sum_{(a,b) \in \text{virtual}} \langle ij || ab \rangle \frac{2 \langle ij || ab \rangle - \langle ij || ab \rangle - \langle ij || ba \rangle}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b}$$

where the E_{HF} is the Hartree-Fock energy calculated by the following formula:

$$E_{\text{HF}} = E_{\text{NN}} + 2 \sum_{i=1}^{N_{\text{occ}}} \langle i | h | i \rangle + \sum_{i=1}^{N_{\text{occ}}} \sum_{j=1}^{N_{\text{occ}}} \left(2 \langle ij || ij \rangle - \langle ij || ji \rangle \right)$$

1.2 Dependencies

The program depends on the HDF5 library and TREXIO library.

1.2.1 HDF5

On Ubuntu, the HDF5 library can be installed using the following command:

```
$ sudo apt-get install libhdf5-dev
```

1.2.2 TREXIO

The TREXIO library can be installed using the following command:

```
$ wget https://github.com/TREX-CoE/trexio/releases/download/v2.5.0/trexio-2.5.0.tar.gz
$ tar -zxvf trexio-2.5.0.tar.gz
$ cd trexio-2.5.0
$ ./configure
$ make
$ sudo make install
```

Note that when configuring the TREXIO library, you may need to specify the path to the HDF5 library if it is not installed in the default location. See more information [here](#).

1.3 Installation and usage

To install the program, run the following commands:

```
$ git clone git@github.com:vfornero/ACT-homeworks.git
$ cd project1
$ ./compile.sh
```

The program can be run using the following command:

```
./main.o [path/to/input.h5]
```

The program will read the `input.h5` file and write the MP2 energy to the standard output.

1.4 Example

```
$ ./main.o ./data/h2o.h5
```

The output will be:

```
Reading file: ./data/h2o.h5
Nucleus repulsion energy: 9.194966
Number of up electrons: 5
Number of MOs: 24
Number of 2e integrals: 13458
h_core_sum: -123.151186
eri_sum: 37.929422
Electronic energy: -85.221764
HF energy: -76.026799
MP2 energy: -76.230759
```


Chapter 2

Installation

To install the program, run the following commands:

```
$ git clone git@github.com:vfornero/ACT-homeworks.git  
$ cd project1  
$ ./compile.sh
```

The program will be available as `main.o` in the `project1` directory.

Chapter 3

Class Index

3.1 Class List

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

Mol	Mol struct contains all the information of the molecule	9
---------------------	---	-------------------

Chapter 4

File Index

4.1 File List

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

src/hf.c	This module contains the functions for HF energy calculation	11
src/hf.h	This module contains the functions for HF energy calculation	11
src/main.c	This program reads the .h5 file and computes the MP2 energy	14
src/mol.c	This module contains the functions for Mol struct	14
src/mol.h	This module contains the Mol struct and functions for Mol struct	17
src/mp2.c	This module contains the functions for MP2 energy calculation	20
src/mp2.h	This module contains the functions for MP2 energy calculation	21
src/reader.c	This module contains the functions for reading the trexio file	22
src/reader.h	This module contains the functions for reading the trexio file	25
src/util.c	This module contains the utility functions	29
src/util.h	This module contains the utility functions	31

Chapter 5

Class Documentation

5.1 Mol Struct Reference

[Mol](#) struct contains all the information of the molecule.

```
#include <mol.h>
```

Public Attributes

- `int32_t mo_num`
number of MOs
- `int32_t n_up`
number of up electrons (or down electrons for RHF)
- `int64_t n_2e_int`
number of 2e integrals
- energy `e_nuc`
nucleus repulsion energy
- `int32_t * eri_index`
2e integrals index
- energy * `eri_value`
2e integrals values
- energy * `eri_seq`
2e integrals in unique index sequence
- energy * `h_core`
1e integrals
- energy * `e_mo`
MO energies.
- energy `h_core_sum`
sum of diagonal elements of h_core
- energy `eri_sum`
sum of eri values
- energy `e_elec`
electronic energy
- energy `e_hf`
HF energy.
- energy `e_mp2`
MP2 energy.

5.1.1 Detailed Description

[Mol](#) struct contains all the information of the molecule.

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

- [src/mol.h](#)

Chapter 6

File Documentation

6.1 src/hf.c File Reference

This module contains the functions for HF energy calculation.

```
#include <stdio.h>
#include <stdlib.h>
#include <trexio.h>
#include "reader.h"
#include "mol.h"
#include "util.h"
Include dependency graph for hf.c:
```

6.2 src/hf.h File Reference

This module contains the functions for HF energy calculation.

```
#include <stdio.h>
#include <stdlib.h>
#include "mol.h"
#include "util.h"
```

Include dependency graph for hf.h: This graph shows which files directly or indirectly include this file:

Functions

- void [get_hcore_sum](#) ([Mol](#) *const m)
Get the core Hamiltonian sum.
- void [hf_kernel](#) ([Mol](#) *const m)
Calculate the HF energy.
- int [match_eri](#) (int i, int j, int k, int l, [Mol](#) *const m)
Match the ERI index HF ERI should be $\langle ij|ij \rangle$ or $\langle ij|ji \rangle$, for $\langle ij|kl \rangle$, check if $i == k$ and $j == l$ or $i == l$ and $j == k$.
- void [get_eri_sum](#) ([Mol](#) *const m)
Get sum of ERI values.

6.2.1 Detailed Description

This module contains the functions for HF energy calculation.

Author

Tianyi Gao

Version

1.0

Date

2024-11-24

Copyright

GNU Public License V3.0

6.2.2 Function Documentation

6.2.2.1 get_eri_sum()

```
void get_eri_sum (
    Mol *const m )
```

Get sum of ERI values.

Parameters

<i>m</i>	Mol struct
----------	------------

6.2.2.2 get_hcore_sum()

```
void get_hcore_sum (
    Mol *const m )
```

Get the core Hamiltonian sum.

Parameters

<i>m</i>	Mol struct
----------	------------

6.2.2.3 hf_kernel()

```
void hf_kernel (
```

```
Mol *const m )
```

Calculate the HF energy.

Parameters

<i>m</i>	Mol struct
----------	------------

6.2.2.4 match_eri()

```
int match_eri (
    int i,
    int j,
    int k,
    int l,
    Mol *const m )
```

Match the ERI index HF ERI should be $\langle ij|ij \rangle$ or $\langle ij|ji \rangle$, for $\langle ij|kl \rangle$, check if $i == k$ and $j == l$ or $i == l$ and $j == k$.

Parameters

<i>i</i>	
<i>j</i>	
<i>k</i>	
<i>l</i>	
<i>m</i>	

Returns

- 1 if $i = j = k = l$
- 2 if $i = k$ and $j = l$
- 3 if $i = l$ and $j = k$
- 0 if none of the above

6.3 hf.h

[Go to the documentation of this file.](#)

```
00001
00011 #ifndef HF_H
00012 #define HF_H
00013
00014 #include <stdio.h>
00015 #include <stdlib.h>
00016 #include "mol.h"
00017 #include "util.h"
00018
00024 void get_hcore_sum(Mol* const m);
00025
00031 void hf_kernel(Mol* const m);
00032
00047 int match_eri(int i, int j, int k, int l, Mol* const m);
00048
00054 void get_eri_sum(Mol* const m);
00055
00056 #endif
```

6.4 src/main.c File Reference

This program reads the .h5 file and computes the MP2 energy.

```
#include <stdio.h>
#include <stdlib.h>
#include <trexio.h>
#include "reader.h"
#include "mol.h"
#include "util.h"
#include "hf.h"
#include "mp2.h"
```

Include dependency graph for main.c:

Functions

- int **main** (int argc, char *argv[])
Application entry.

6.4.1 Detailed Description

This program reads the .h5 file and computes the MP2 energy.

Author

Yifan Jiang, Tianyi Gao

Version

1.0

Date

2024-11-25

Copyright

GNU Public License V3.0

6.5 src/mol.c File Reference

This module contains the functions for [Mol](#) struct.

```
#include <stdio.h>
#include <stdlib.h>
#include "mol.h"
#include "reader.h"
#include "util.h"
```

Include dependency graph for mol.c:

Functions

- void `mol_init` (`Mol *const m`, `char *filename`)
initialize the `Mol` struct from the .h5 file
- void `mol_destruct` (`Mol *const m`)
destruct the `Mol` struct
- energy `get_nth_eri` (`int64_t n`, `Mol *const m`)
*get the *nth* 2e integral value*
- void `gen_eri_index` (`Mol *const m`)
extract the 2e integral value into a 1D array, with unique index sequence
- energy `get_eri_energy` (`Mol *const m`, `int32_t i`, `int32_t j`, `int32_t a`, `int32_t b`)
Given (ijab), return the 2e integral energy of (ia|jb)

6.5.1 Detailed Description

This module contains the functions for `Mol` struct.

Author

Yifan Jiang, Tianyi Gao

Version

1.0

Date

2024-11-25

Copyright

GNU Public License V3.0

6.5.2 Function Documentation

6.5.2.1 `gen_eri_index()`

```
void gen_eri_index (  
    Mol *const m )
```

extract the 2e integral value into a 1D array, with unique index sequence

Parameters

<code><i>m</i></code>	
-----------------------	--

6.5.2.2 get_eri_energy()

```
energy get_eri_energy (
    Mol *const m,
    int32_t i,
    int32_t j,
    int32_t a,
    int32_t b ) [inline]
```

Given (ijab), return the 2e integral energy of (ia|jb)

Parameters

<i>m</i>	Mol struct
<i>i,j,a,b</i>	indices of the 2e integral

Returns

2e integral of (ia|jb)

6.5.2.3 get_nth_eri()

```
energy get_nth_eri (
    int64_t n,
    Mol *const m )
```

get the nth 2e integral value

Parameters

<i>n</i>	index of the 2e integral
<i>m</i>	Mol struct

Returns

nth 2e integral value

6.5.2.4 mol_destruct()

```
void mol_destruct (
    Mol *const m )
```

destruct the Mol struct

Parameters

<i>m</i>	Mol struct
----------	------------

6.5.2.5 mol_init()

```
void mol_init (
    Mol *const m,
    char * filename )
```

initialize the [Mol](#) struct from the .h5 file

Parameters

<i>m</i>	Mol struct
<i>filename</i>	path to the .h5 file

6.6 src/mol.h File Reference

This module contains the [Mol](#) struct and functions for [Mol](#) struct.

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
```

Include dependency graph for mol.h: This graph shows which files directly or indirectly include this file:

Classes

- struct [Mol](#)
[Mol](#) struct contains all the information of the molecule.

Typedefs

- typedef double **energy**

Functions

- void [mol_init](#) ([Mol](#) *const m, char *filename)
initialize the [Mol](#) struct from the .h5 file
- void [mol_destruct](#) ([Mol](#) *const m)
destruct the [Mol](#) struct
- energy [get_nth_eri](#) (int64_t n, [Mol](#) *const m)
get the nth 2e integral value
- void [gen_eri_index](#) ([Mol](#) *const m)
extract the 2e integral value into a 1D array, with unique index sequence
- energy [get_eri_energy](#) ([Mol](#) *const m, int32_t i, int32_t j, int32_t a, int32_t b)
Given (ijab), return the 2e integral energy of (ia|jb)

6.6.1 Detailed Description

This module contains the [Mol](#) struct and functions for [Mol](#) struct.

Author

Yifan Jiang, Tianyi Gao

Version

1.0

Date

2024-11-23

Copyright

GNU Public License V3.0

6.6.2 Function Documentation

6.6.2.1 `gen_eri_index()`

```
void gen_eri_index (  
    Mol *const m )
```

extract the 2e integral value into a 1D array, with unique index sequence

Parameters

<i>m</i>	
----------	--

6.6.2.2 `get_eri_energy()`

```
energy get_eri_energy (  
    Mol *const m,  
    int32_t i,  
    int32_t j,  
    int32_t a,  
    int32_t b ) [inline]
```

Given (ijab), return the 2e integral energy of (ia|jb)

Parameters

<i>m</i>	Mol struct
<i>i,j,a,b</i>	indices of the 2e integral

Returns

2e integral of (ia|jb)

6.6.2.3 get_nth_eri()

```
energy get_nth_eri (
    int64_t n,
    Mol *const m )
```

get the nth 2e integral value

Parameters

<i>n</i>	index of the 2e integral
<i>m</i>	Mol struct

Returns

nth 2e integral value

6.6.2.4 mol_destruct()

```
void mol_destruct (
    Mol *const m )
```

destruct the Mol struct

Parameters

<i>m</i>	Mol struct
----------	------------

6.6.2.5 mol_init()

```
void mol_init (
    Mol *const m,
    char * filename )
```

initialize the Mol struct from the .h5 file

Parameters

<i>m</i>	Mol struct
<i>filename</i>	path to the .h5 file

6.7 mol.h

[Go to the documentation of this file.](#)

```

00001
00011 #ifndef MOL_H
00012 #define MOL_H
00013
00014 #include <stdio.h>
00015 #include <stdlib.h>
00016 #include <stdint.h>
00017
00018 typedef double energy;
00019
00023 typedef struct {
00024     int32_t mo_num;
00025     int32_t n_up;
00026     int64_t n_2e_int;
00027     energy e_nuc;
00028     int32_t* eri_index;
00029     energy* eri_value;
00030     energy* eri_seq;
00031     energy* h_core;
00032     energy* e_mo;
00033     energy h_core_sum;
00034     energy eri_sum;
00035     energy e_elec;
00036     energy e_hf;
00037     energy e_mp2;
00038 } Mol;
00039
00046 void mol_init(Mol* const m, char* filename);
00047
00053 void mol_destruct(Mol* const m);
00054
00062 energy get_nth_eri(int64_t n, Mol* const m);
00063
00069 void gen_eri_index(Mol* const m);
00070
00078 energy get_eri_energy(Mol* const m, int32_t i, int32_t j, int32_t a, int32_t b);
00079
00080
00081 #endif
00082

```

6.8 src/mp2.c File Reference

This module contains the functions for MP2 energy calculation.

```
#include "mp2.h"
```

Include dependency graph for mp2.c:

Functions

- void [cal_mp2_correlation](#) (Mol *const m)
Calculate the MP2 correlation energy.

6.8.1 Detailed Description

This module contains the functions for MP2 energy calculation.

Author

Yifan Jiang

Version

1.0

Date

2024-11-25

Copyright

GNU Public License V3.0

6.8.2 Function Documentation

6.8.2.1 cal_mp2_correlation()

```
void cal_mp2_correlation (  
    Mol *const m )
```

Calculate the MP2 correlation energy.

Parameters

<i>m</i>	Mol struct
----------	------------

6.9 src/mp2.h File Reference

This module contains the functions for MP2 energy calculation.

```
#include "mol.h"
```

Include dependency graph for mp2.h: This graph shows which files directly or indirectly include this file:

Functions

- void [cal_mp2_correlation](#) (Mol *const m)
Calculate the MP2 correlation energy.

6.9.1 Detailed Description

This module contains the functions for MP2 energy calculation.

Author

Yifan Jiang

Version

1.0

Date

2024-11-25

Copyright

GNU Public License V3.0

6.9.2 Function Documentation

6.9.2.1 cal_mp2_correlation()

```
void cal_mp2_correlation (  
    Mol *const m )
```

Calculate the MP2 correlation energy.

Parameters

<i>m</i>	Mol struct
----------	------------

6.10 mp2.h

[Go to the documentation of this file.](#)

```
00001  
00011 #ifndef MP2_H  
00012 #define MP2_H  
00013  
00014 #include "mol.h"  
00015  
00021 void cal_mp2_correlation(Mol* const m);  
00022  
00023 #endif
```

6.11 src/reader.c File Reference

This module contains the functions for reading the trexio file.

```
#include <stdio.h>  
#include <stdlib.h>  
#include <trexio.h>  
#include "reader.h"  
#include "mol.h"  
#include "util.h"  
Include dependency graph for reader.c:
```

Functions

- void `read_nucleus_repulsion` (`trexio_t *const trexio_file`, `Mol *const m`)
read the nucleus repulsion energy from the trexio file
- void `read_electron_up_num` (`trexio_t *const trexio_file`, `Mol *const m`)
read the number of up electrons from the trexio file
- void `read_mo_num` (`trexio_t *const trexio_file`, `Mol *const m`)
read the number of MOs from the trexio file
- void `read_mo_1e_int_h_core` (`trexio_t *const trexio_file`, `Mol *const m`)
read the 1e integrals from the trexio file
- void `read_mo_2e_int_eri_size` (`trexio_t *const trexio_file`, `Mol *const m`)
read the size of 2e integrals from the trexio file
- void `read_mo_2e_int_eri` (`trexio_t *const trexio_file`, `Mol *const m`)
read the 2e integrals from the trexio file
- void `read_mo_energy` (`trexio_t *const trexio_file`, `Mol *const m`)
read the MO energies from the trexio file

6.11.1 Detailed Description

This module contains the functions for reading the trexio file.

Author

Tianyi Gao, Yifan Jiang

Version

1.0

Date

2024-11-25

Copyright

GNU Public License V3.0

6.11.2 Function Documentation

6.11.2.1 `read_electron_up_num()`

```
void read_electron_up_num (  
    trexio_t *const trexio_file,  
    Mol *const m )
```

read the number of up electrons from the trexio file

Parameters

<i>trexio_file</i>	trexio file handler
<i>m</i>	Mol struct

6.11.2.2 read_mo_1e_int_h_core()

```
void read_mo_1e_int_h_core (
    trexio_t *const trexio_file,
    Mol *const m )
```

read the 1e integrals from the trexio file

Parameters

<i>trexio_file</i>	trexio file handler
<i>m</i>	Mol struct

6.11.2.3 read_mo_2e_int_eri()

```
void read_mo_2e_int_eri (
    trexio_t *const trexio_file,
    Mol *const m )
```

read the 2e integrals from the trexio file

Parameters

<i>trexio_file</i>	trexio file handler
<i>m</i>	Mol struct

6.11.2.4 read_mo_2e_int_eri_size()

```
void read_mo_2e_int_eri_size (
    trexio_t *const trexio_file,
    Mol *const m )
```

read the size of 2e integrals from the trexio file

Parameters

<i>trexio_file</i>	trexio file handler
<i>m</i>	Mol struct

6.11.2.5 read_mo_energy()

```
void read_mo_energy (
    trexio_t *const trexio_file,
    Mol *const m )
```

read the MO energies from the trexio file

Parameters

<i>trexio_file</i>	trexio file handler
<i>m</i>	Mol struct

6.11.2.6 read_mo_num()

```
void read_mo_num (
    trexio_t *const trexio_file,
    Mol *const m )
```

read the number of MOs from the trexio file

Parameters

<i>trexio_file</i>	trexio file handler
<i>m</i>	Mol struct

6.11.2.7 read_nucleus_repulsion()

```
void read_nucleus_repulsion (
    trexio_t *const trexio_file,
    Mol *const m )
```

read the nucleus repulsion energy from the trexio file

Parameters

<i>trexio_file</i>	trexio file handler
<i>m</i>	Mol struct

6.12 src/reader.h File Reference

This module contains the functions for reading the trexio file.

```
#include <stdio.h>
#include <trexio.h>
#include "mol.h"
```

Include dependency graph for reader.h: This graph shows which files directly or indirectly include this file:

Functions

- void `read_nucleus_repulsion` (`trexio_t *const trexio_file`, `Mol *const m`)
read the nucleus repulsion energy from the trexio file
- void `read_electron_up_num` (`trexio_t *const trexio_file`, `Mol *const m`)
read the number of up electrons from the trexio file
- void `read_mo_num` (`trexio_t *const trexio_file`, `Mol *const m`)
read the number of MOs from the trexio file
- void `read_mo_1e_int_h_core` (`trexio_t *const trexio_file`, `Mol *const m`)
read the 1e integrals from the trexio file
- void `read_mo_2e_int_eri_size` (`trexio_t *const trexio_file`, `Mol *const m`)
read the size of 2e integrals from the trexio file
- void `read_mo_2e_int_eri` (`trexio_t *const trexio_file`, `Mol *const m`)
read the 2e integrals from the trexio file
- void `read_mo_energy` (`trexio_t *const trexio_file`, `Mol *const m`)
read the MO energies from the trexio file

6.12.1 Detailed Description

This module contains the functions for reading the trexio file.

Author

Tianyi Gao, Yifan Jiang

Version

1.0

Date

2024-11-23

Copyright

GNU Public License V3.0

6.12.2 Function Documentation

6.12.2.1 `read_electron_up_num()`

```
void read_electron_up_num (  
    trexio_t *const trexio_file,  
    Mol *const m )
```

read the number of up electrons from the trexio file

Parameters

<i>trexio_file</i>	trexio file handler
<i>m</i>	Mol struct

6.12.2.2 read_mo_1e_int_h_core()

```
void read_mo_1e_int_h_core (
    trexio_t *const trexio_file,
    Mol *const m )
```

read the 1e integrals from the trexio file

Parameters

<i>trexio_file</i>	trexio file handler
<i>m</i>	Mol struct

6.12.2.3 read_mo_2e_int_eri()

```
void read_mo_2e_int_eri (
    trexio_t *const trexio_file,
    Mol *const m )
```

read the 2e integrals from the trexio file

Parameters

<i>trexio_file</i>	trexio file handler
<i>m</i>	Mol struct

6.12.2.4 read_mo_2e_int_eri_size()

```
void read_mo_2e_int_eri_size (
    trexio_t *const trexio_file,
    Mol *const m )
```

read the size of 2e integrals from the trexio file

Parameters

<i>trexio_file</i>	trexio file handler
<i>m</i>	Mol struct

6.12.2.5 read_mo_energy()

```
void read_mo_energy (
    trexio_t *const trexio_file,
    Mol *const m )
```

read the MO energies from the trexio file

Parameters

<i>trexio_file</i>	trexio file handler
<i>m</i>	Mol struct

6.12.2.6 read_mo_num()

```
void read_mo_num (
    trexio_t *const trexio_file,
    Mol *const m )
```

read the number of MOs from the trexio file

Parameters

<i>trexio_file</i>	trexio file handler
<i>m</i>	Mol struct

6.12.2.7 read_nucleus_repulsion()

```
void read_nucleus_repulsion (
    trexio_t *const trexio_file,
    Mol *const m )
```

read the nucleus repulsion energy from the trexio file

Parameters

<i>trexio_file</i>	trexio file handler
<i>m</i>	Mol struct

6.13 reader.h

[Go to the documentation of this file.](#)

```
00001
00011 #ifndef READER_H
00012 #define READER_H
00013
00014 #include <stdio.h>
00015 #include <trexio.h>
00016 #include "mol.h"
```

```
00017
00024 void read_nucleus_repulsion(trexio_t* const trexio_file, Mol* const m);
00025
00032 void read_electron_up_num(trexio_t* const trexio_file, Mol* const m);
00033
00040 void read_mo_num(trexio_t* const trexio_file, Mol* const m);
00041
00048 void read_mo_le_int_h_core(trexio_t* const trexio_file, Mol* const m);
00049
00056 void read_mo_2e_int_eri_size(trexio_t* const trexio_file, Mol* const m);
00057
00064 void read_mo_2e_int_eri(trexio_t* const trexio_file, Mol* const m);
00065
00072 void read_mo_energy(trexio_t* const trexio_file, Mol* const m);
00073
00074 #endif
```

6.14 src/util.c File Reference

This module contains the utility functions.

```
#include <stdio.h>
#include <stdlib.h>
#include "util.h"
#include "mol.h"
Include dependency graph for util.c:
```

Functions

- void `print_matrix` (double *const m, int row, int col)
print a double array in matrix format
- double `sum_diag` (double *const m, int n, int order)
sum to nth diagonal elements of a square matrix
- int32_t `index_2e` (int i, int j, int k, int l, int n)
calculates unique index of given (ijkl) for 2e integrals

6.14.1 Detailed Description

This module contains the utility functions.

Author

Tianyi Gao

Version

1.0

Date

2024-11-23

Copyright

GNU Public License V3.0

6.14.2 Function Documentation

6.14.2.1 index_2e()

```
int32_t index_2e (
    int i,
    int j,
    int k,
    int l,
    int n ) [inline]
```

calculates unique index of given (ijkl) for 2e integrals

Note

$0 \leq i, j, k, l < n$
 $i \leq j, k \leq l, i \leq k$
 if $i == k$, then $j \leq l$

Parameters

<i>i,j,k,l</i>	indices of the 2e integrals
<i>n</i>	number of MOs

Returns

unique index

6.14.2.2 print_matrix()

```
void print_matrix (
    double *const m,
    int row,
    int col )
```

print a double array in matrix format

Parameters

<i>m</i>	array to be printed
<i>row</i>	number of rows
<i>col</i>	number of columns

6.14.2.3 sum_diag()

```
double sum_diag (
    double *const m,
```

```
int n,  
int order )
```

sum to nth diagonal elements of a square matrix

Parameters

<i>m</i>	1D matrix data array
<i>n</i>	sum up to nth diagonal elements
<i>order</i>	order of the matrix (row = col)

Returns

sum of the diagonal elements

6.15 src/util.h File Reference

This module contains the utility functions.

```
#include <stdio.h>  
#include <stdlib.h>  
#include "mol.h"
```

Include dependency graph for util.h: This graph shows which files directly or indirectly include this file:

Functions

- void [print_matrix](#) (double *const m, int row, int col)
print a double array in matrix format
- double [sum_diag](#) (double *const m, int n, int order)
sum to nth diagonal elements of a square matrix
- int32_t [index_2e](#) (int i, int j, int k, int l, int n)
calculates unique index of given (ijkl) for 2e integrals

6.15.1 Detailed Description

This module contains the utility functions.

Author

Tianyi Gao

Version

1.0

Date

2024-11-23

Copyright

GNU Public License V3.0

6.15.2 Function Documentation

6.15.2.1 index_2e()

```
int32_t index_2e (
    int i,
    int j,
    int k,
    int l,
    int n ) [inline]
```

calculates unique index of given (ijkl) for 2e integrals

Note

$0 \leq i, j, k, l < n$
 $i \leq j, k \leq l, i \leq k$
 if $i == k$, then $j \leq l$

Parameters

<i>i,j,k,l</i>	indices of the 2e integrals
<i>n</i>	number of MOs

Returns

unique index

6.15.2.2 print_matrix()

```
void print_matrix (
    double *const m,
    int row,
    int col )
```

print a double array in matrix format

Parameters

<i>m</i>	array to be printed
<i>row</i>	number of rows
<i>col</i>	number of columns

6.15.2.3 sum_diag()

```
double sum_diag (
    double *const m,
```

```
int n,  
int order )
```

sum to nth diagonal elements of a square matrix

Parameters

<i>m</i>	1D matrix data array
<i>n</i>	sum up to nth diagonal elements
<i>order</i>	order of the matrix (row = col)

Returns

sum of the diagonal elements

6.16 util.h

[Go to the documentation of this file.](#)

```
00001  
00012 #ifndef UTIL_H  
00013 #define UTIL_H  
00014 #include <stdio.h>  
00015 #include <stdlib.h>  
00016 #include "mol.h"  
00017  
00025 void print_matrix(double* const m, int row, int col);  
00034 double sum_diag(double* const m, int n, int order);  
00035  
00045 int32_t index_2e(int i, int j, int k, int l, int n);  
00046  
00047 #endif
```


Index

cal_mp2_correlation
 mp2.c, [21](#)
 mp2.h, [22](#)
Computation of the MP2 energy, [1](#)

gen_eri_index
 mol.c, [15](#)
 mol.h, [18](#)

get_eri_energy
 mol.c, [15](#)
 mol.h, [18](#)

get_eri_sum
 hf.h, [12](#)

get_hcore_sum
 hf.h, [12](#)

get_nth_eri
 mol.c, [16](#)
 mol.h, [19](#)

hf.h
 get_eri_sum, [12](#)
 get_hcore_sum, [12](#)
 hf_kernel, [12](#)
 match_eri, [13](#)

hf_kernel
 hf.h, [12](#)

index_2e
 util.c, [30](#)
 util.h, [32](#)

Installation, [3](#)

match_eri
 hf.h, [13](#)

Mol, [9](#)

mol.c
 gen_eri_index, [15](#)
 get_eri_energy, [15](#)
 get_nth_eri, [16](#)
 mol_destruct, [16](#)
 mol_init, [16](#)

mol.h
 gen_eri_index, [18](#)
 get_eri_energy, [18](#)
 get_nth_eri, [19](#)
 mol_destruct, [19](#)
 mol_init, [19](#)

mol_destruct
 mol.c, [16](#)
 mol.h, [19](#)

mol_init
 mol.c, [16](#)
 mol.h, [19](#)

mp2.c
 cal_mp2_correlation, [21](#)

mp2.h
 cal_mp2_correlation, [22](#)

print_matrix
 util.c, [30](#)
 util.h, [32](#)

read_electron_up_num
 reader.c, [23](#)
 reader.h, [26](#)

read_mo_1e_int_h_core
 reader.c, [24](#)
 reader.h, [27](#)

read_mo_2e_int_eri
 reader.c, [24](#)
 reader.h, [27](#)

read_mo_2e_int_eri_size
 reader.c, [24](#)
 reader.h, [27](#)

read_mo_energy
 reader.c, [24](#)
 reader.h, [27](#)

read_mo_num
 reader.c, [25](#)
 reader.h, [28](#)

read_nucleus_repulsion
 reader.c, [25](#)
 reader.h, [28](#)

reader.c
 read_electron_up_num, [23](#)
 read_mo_1e_int_h_core, [24](#)
 read_mo_2e_int_eri, [24](#)
 read_mo_2e_int_eri_size, [24](#)
 read_mo_energy, [24](#)
 read_mo_num, [25](#)
 read_nucleus_repulsion, [25](#)

reader.h
 read_electron_up_num, [26](#)
 read_mo_1e_int_h_core, [27](#)
 read_mo_2e_int_eri, [27](#)
 read_mo_2e_int_eri_size, [27](#)
 read_mo_energy, [27](#)
 read_mo_num, [28](#)
 read_nucleus_repulsion, [28](#)

- src/hf.c, [11](#)
- src/hf.h, [11](#), [13](#)
- src/main.c, [14](#)
- src/mol.c, [14](#)
- src/mol.h, [17](#), [20](#)
- src/mp2.c, [20](#)
- src/mp2.h, [21](#), [22](#)
- src/reader.c, [22](#)
- src/reader.h, [25](#), [28](#)
- src/util.c, [29](#)
- src/util.h, [31](#), [33](#)
- sum_diag
 - util.c, [30](#)
 - util.h, [32](#)
- util.c
 - index_2e, [30](#)
 - print_matrix, [30](#)
 - sum_diag, [30](#)
- util.h
 - index_2e, [32](#)
 - print_matrix, [32](#)
 - sum_diag, [32](#)