MASTEQ Manual (ver.1.1.1)

[Abstract]

MASTEQ is a numerical estimator of atomic diffusivity in crystals using a master equation approach. The diffusion tensor is calculated under the independent particle approximation only by specifying the initial and final sites, jump vector, and jump frequency of every atomic jump in unitcell. See the following references for the detailed theoretical background. Note that any interaction between diffusion carriers cannot be treated until ver.1.0.2. After ver.1.1.0, *site blocking* can be treated by correcting jump frequencies based on thermal equilibrium site occupancies using siteblk.py. The correlation between subsequent atomic jumps still cannot be treated in the current version. This code was written by Kazuaki Toyoura. Mr. Takeo Fujii has great contribution to the code development with fruitful discussions.

[1] K. Toyoura, T. Fujii, N. Hatada, D. Han, T. Uda, *The Journal of Physical Chemistry C* **123**, 26823-26830 (2019).

[2] K. Toyoura, T. Fujii, K. Kanamori, I. Takeuchi, *Phys. Rev. B* **101**, 184117/1-11 (2020).

[3] T. Fujii, K. Toyoura, T. Uda, S. Kasamatsu, *Phys. Chem. Chem. Phys.* **23**, 5908-5918 (2021).

[4] K. Kanamori, K. Toyoura, J Honda, K Hattori, A Seko, M Karasuyama, K. Shitara, M. Shiga, A. Kuwabara, I. Takeuchi, *Phys. Rev. B* **97**, 125124/1-6 (2018). [Optimal path search]

[Programming language]

Python 3.x (installed through anaconda, https://www.anaconda.com/products/individual)

Imported modules in this code: numpy, scipy, argparse, copy, datetime, os, sys

* mkemig.py in tools requires *pymatgen, https://pymatgen.org/*.

* optPathSearch.py in tools requires *anytree, https://anytree.readthedocs.io/en/latest/index.html*.

[Input file]

Only an input file, ***jmpdata.csv***, is required, which includes initial and final site IDs, jump vectors, and jump frequencies for every atomic jump in unitcell. Note that all atomic jumps with initial sites in the unitcell have to be specified including the atomic jumps across the periodic boundary, and that both jumps in the opposite directions have to be specified separately. New line for different atomic jumps. "#" denotes a comment line. Put the following items for every atomic jump separated by comma. (Site IDs have to be sequential numbers starting from 1.)

- Initial site ID [integer]: ID of the initial site for an atomic jump
- Final site ID [integer]: ID of the final site for an atomic jump
- *x* component of the jump vector [float]: The *x* component of jump vector in Å for an atomic jump
- *y* component of the jump vector [float]: The *y* component of jump vector in Å for an atomic jump
- *z* component of the jump vector [float]: The *z* component of jump vector in Å for an atomic jump
- Jump frequency [float]: Jump frequency in Hz for an atomic jump

(jmpdata.csv example)

```
#InitialSiteID,FinalSiteID,jmpVec_x[Ang.], jmpVec_y[Ang.], jmpVec_z[Ang.],frequency[Hz]
1,7,-1.2067005467,1.2067005467,0.0000000000,5.4961104742e+11
1,9,0.0000000000,-0.9114043880,-0.9114043880,1.3907136217e+12
```

[Options]

-h, --help　Help information. List of options in this code.

--jmp　File name of atomic jump data in a given system. If it is the default name (jmpdata.csv), the argument is not necessary.

--nonzero　Non-zero elements in diffusion tensor. Specified non-zero elements are estimated in this program. Specify element indexes separated by comma. Default is 1,2,3,4,5,6. Note that the independent elements are six in diffusion tensor, where the index is as follows:

$$\text{Diffusion tensor: } \begin{pmatrix} D_1 & D_6 & D_5 \\ D_6 & D_2 & D_4 \\ D_5 & D_4 & D_3 \end{pmatrix}$$

For example, "--nonzero 1,2,3" means estimating only the diagonal elements.

--factor　Scaling factor f for wave vector $\mathbf{Q}$. This factor have to be enough small to correspond to large scale for atomic diffusion. The length of the shortest jump vector l_{min} is used for the standardized scale of reciprocal space, given by $2\pi/l_{\mathrm{min}}$. The magnitude of wave vector $|\mathbf{Q}|$ for estimating diffusion tensor is then $(2\pi/l_{\mathrm{min}})\times f$. Default value is 1.0e-3.

[Usage]

Type the following command at the directory with jmpdata.csv. The estimated diffusion tensor is printed on the standard output. Please redirect it to a file, e.g., stdout.

```
python [MASTEQ_DIR]/masteq.py --jmp jmpdata.csv --nonzero 1,2,3 --factor 1.0e-3 > stdout
```

[Note]

This program assumes a single diffusion network without any other competing network in a given system. For example, if several parallel two-dimensional (2D) networks coexist in a given system, the estimated diffusivity reflects only the diffusivity of the fastest one. In such a case, the jump matrix has several eigenvalues close to zero, and this program prints a ***warning*** on the standard output. This warning is often output for anisotropic diffusivity at low temperatures, because a single network at high temperatures can substantially be separated into several networks at low temperatures. Generally, the differences in jump frequency between atomic jumps become larger rapidly with decreasing temperatures, which creates negligible paths for atomic jumps at low temperatures, leading to the separated networks.

The accuracy of estimated diffusivity corresponds to the accuracy of the minimum-magnitude eigenvalue in the jump matrix. The smaller scaling factor f for wave vector $\mathbf{Q}$ is better in terms of larger scale for atomic diffusion, but worse in terms of the accuracy of eigenvalues in the jump matrix. Please carefully check the temperature dependence of estimated diffusivity, which should be a smooth curve (approximately a straight line) in the Arrhenius plot. In the future, this program has a function for checking the diffusion network (site connectivity) in a given system.

[Examples]

H_BaZrO3

The proton diffusivity in the perfect crystal of $BaZrO_3$ [3] is estimated in the range of 200-1000 K under the independent-particle approximation. In the perfect crystal, a proton migrates two types of proton jumps, i.e., *rotation* around single oxide ions and *hopping* between adjacent oxide ions. The calculated migration energies of proton rotation and hopping are 0.17 and 0.25 eV, respectively. The input files (***jmpdata.csv***) at all temperatures can be generated at once by ***mkjmpdata.py***. An input file for mkjmpdata.py, ***emig.csv***, is also generated by ***mkemig.py***. See the next section for the details.

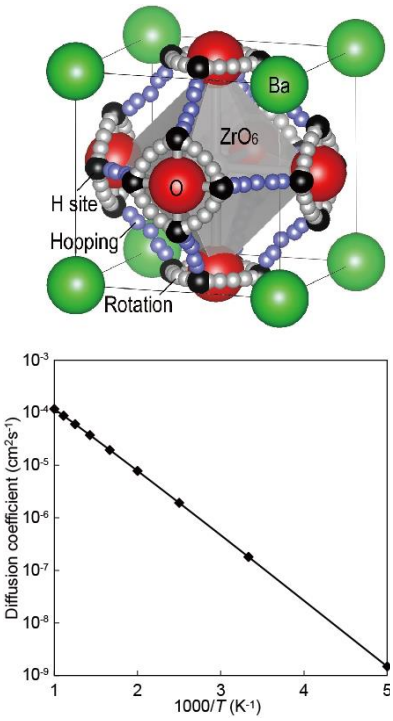

Fig. 1. Migration paths and estimated diffusivity of a proton in $BaZrO_3$.

H_Y-dopedBaZrO3

The proton diffusivity in 30%Y-doped $BaZrO_3$ at 600 K is estimated with simple proton-proton interaction by site blocking [1,3]. Specifically, a supercell of 10×10×10 $BaZrO_3$ unitcells with randomly-distributed 300 Y dopants is employed, where more than one protons are not bonded to the same O ion. The input file, jumpdata.csv, is corrected using the thermal equilibrium site occupancy, which can be performed by siteblk.py. Note that the correlation between subsequent atomic jumps is not treated in this example. See the next section for the details.

[Useful tools]

mkjmpdata.py

This program makes multiple ***jmpdata.csv*** all at once in a specified temperature range, using the migration energy ΔE^{mig} and vibrational prefactor ν_0 for atomic jump. The jump frequency ν is estimated on the basis of the classical transition state theory, $\nu = \nu_0 \exp(-\Delta E^{mig}/k_B T)$, where k_B and T are the Boltzmann constant and temperature. The input file (default name: emig.csv) have to be prepared in the csv format. The file format is almost same as jmpdata.csv, where ν [Hz], is replaced by ΔE^{mig} [eV] and ν_0 [Hz] separated by comma. Emig.csv have to contain the information on all atomic jumps in the unitcell as well as jmpdata.csv. The file example is as follows:

(emig.csv example)

```
#InitialSiteID,FinalSiteID,jmpVec_x[Ang.], jmpVec_y[Ang.], jmpVec_z[Ang.],DEmig[eV],v0[Hz]
1,7,-1.2067005467,1.2067005467,0.0000000000,0.25,1.0e+13
1,9,0.0000000000,-0.9114043880,-0.9114043880,0.17,1.0e+13
```

The temperature range [K] is specified by three parameters, i.e., lb_T (lower bound of temperature), ub_T (upper bound of temperature), and int_T (interval of temperature), which are specified as arguments.

-h, --help	Help information. List of options in this code.
--emig	File name of migration energy [eV] and vibrational prefactor [Hz] for every atomic jump. If it is the default name (emig.csv), this argument is not necessary.
--lb_T	Lower bound of the temperature range [K]. Default value is 300.
--ub_T	Upper bound of the temperature range [K]. Default value is 1000.
--int_T	Interval of the temperature step [K]. Default value is 100.

When typing the following command, jmpdata_300K.csv, jmpdata_400K.csv, and jmpdata_500K.csv are generated.

```
python [MASTEQ_DIR]/tools/mkjmpdata.py --emig emig.csv --lb_T 300 --ub_T 500 --int_T 100
```

mkemig.py (*pymatgen* is required!)

This program makes ***emig.csv*** from only the information on crystallographically non-equivalent atomic jumps. Specifically, two files are required, ***site.cif*** with site information of diffusion carriers and ***emig_noneq.csv*** with jump information including the initial and final site types, jump distance [ang.], migration energy [eV], and vibrational prefactor [Hz]. All equivalent sites in the unitcell are generated using symmetry operations in site.cif, and the equivalent atomic jumps are then searched by the jump distance and the initial and final site types specified in emig_noneq.csv.

For reading ***site.cif*** by ***pymatgen***, non-equivalent site types of diffusion carriers have to be distinguished by ***element symbols*** (H, He, Li, Be, …), whatever the diffusion carriers are. The example of site information line is shown below.

(Site information example in site.cif)

```
loop_
  _atom_site_label
  _atom_site_occupancy
  _atom_site_fract_x
  _atom_site_fract_y
  _atom_site_fract_z
  _atom_site_type_symbol
  H     1.0   0.000000  0.000000  0.000000   H
  He    1.0   0.500000  0.500000  0.500000   He
  Li    1.0   0.000000  0.500000  0.500000   Li
```

Concerning the file format of ***emig_noneq.csv***, non-equivalent atomic jumps are specified by the initial and final site types (element symbols in ***site.cif***), the jump distance [ang.], ΔE^{mig} [eV], and ν_0 [Hz], separated by comma. New line for different atomic jumps. Both jumps in the opposite directions have to be specified separately, if they are non-equivalent. "#" denotes a comment line. The file example is as follows:

(emig_noneq.csv example)

```
#InitialSiteType,FinalSiteType,jumpDistance[Ang.],DEmig[eV],v0[Hz]
H,Li,1.41421356,0.30,1.0e+13
Li,H,1.41421356,0.20,1.0e+13
He,Li,1.00000000,0.35,1.0e+13
Li,He,1.00000000,0.30,1.0e+13
```

Type the following command at the directory with ***site.cif*** and ***emig_noneq.csv***, and ***emig.csv*** is generated.

```
python [MASTEQ_DIR]/tools/mkemig.py --cif site.cif --emig_noneq emig_noneq.csv --supercell 1,1,1 --prec 1.0e-4
```

The option of this program is as follows:

-h, --help	Help information. List of options in this code.
--cif	File name of the cif file with site information. If it is the default name (site.cif), this argument is not necessary.
--emig_noneq	File name with information on non-equivalent atomic jumps. Initial and final site type, jump vector, Emig, and vibrational prefactor in the csv format. New line for non-equivalent atomic jumps. Both jumps in the opposite directions have to be specified if they are non-equivalent. Site types have to be specified by ELEMENT SYMBOL! e.g., H, He, Li, Be, B, etc. If it is the default name (emig_noneq.cif), this argument is not necessary.

--supercell Supercell size for expanding unitcell. Since the inter-site distance is defined as the 1NN distance with periodicity, longer jump vectors than the half of lattice vectors are not accepted in this program. '3,3,3' is enough even for a small cell with few sites. Default value: 1,1,1

--prec Precision used in symmetry operation and atomic jump search. Default value: 1.0e-4

siteblk.py

This program corrects ***jmpdata.csv*** taking a simple carrier-carrier interaction into account, i.e., ***site blocking***. ***Site blocking*** means that any site in a given system cannot be occupied by more than one diffusion carriers at the same time. The thermal equilibrium site occupancy can easily be estimated from Fermi distribution. In this program, the site blocking is extended by defining ***exclusive region*** around every site, which cannot be occupied by the other carriers as long as the focused site is occupied by a diffusion carrier. Thermal equilibrium site occupancies are used as site blocking probability, where the correlation between subsequent atomic jumps is not considered.

The three input files are required in this program as follows:

1. jmpdata.csv jmpdata.csv under independent-particle approximation without any carrier-carrier interaction.
2. sitePE.csv Potential energy (PE) at each site in a given system. All the PE values [eV] are written in a single row separated by comma, OR are written one by one in a single column.
   (sitePE.csv example)
   0.12,0.34,0.03,0.42, …
3. excl.csv Exclusive region (exclusive sites) around every site, which is specified by site IDs (Site IDs have to be sequential numbers starting from 1.). Each line specifies the exclusive sites around the focused site with the same ID as the line. The exclusive sites have to be separated by comma in each row, including the focused site ID.
   (excl.csv example)
   1,2,3
   2,3,4
   2,3,5,6
   …

Type the following command at the directory with jmpdata.csv, sitePE.csv, and excl.csv, and the corrected jmpdata.csv, ***corr_jmpdata.csv***, is generated. The other output file, ***corrFactors.dat*** is also generated, in which the correction factor for each atomic jump is listed.

```
python [MASTEQ_DIR]/tools/siteblk.py --jmp jmpdata.csv --sitePE sitePE.csv --excl excl.csv --T 1000 --n 300 --prec exact
```

The option of this program is as follows:

-h, --help Help information. List of options in this code.

--jmp File name of atomic jump data in a given system under the independent-particle approximation. Default name (jmpdata.csv) can be skipped.

--sitePE	File name with the information of site PEs. Default name (sitePE.csv) can be skipped.
--excl	File name with the information of exclusive regions. Default name (excl.csv) can be skipped.
--T	Temperature [K]. Default value is 1000.
--n	The number of diffusion carriers in the system [integer]. Default value is 1, meaning no correction.
--prec	Precision for estimation of site occupancies [string]. 'exact' or 'rough'. Default value is 'exact'. If the number of sites in a given system is too large (> 10000), the occupancy estimation is time consuming. In such case, try to use prec = 'rough'.

optPathSearch.py (*anytree* is required!)

This program searches the *optimal path* in a given crystal, which is defined as the lowest-energy path between two global minimum points separated by a lattice translation vector. In other words, the optimal path corresponds to the *long-range migration pathway* in the crystal. In general, there are three linearly-independent optimal paths. This program is based on dynamic programing. See Ref. [4] for the details of the algorithm. This program requires ***emig.csv*** and ***sitePE.csv***. In addition, the three lattice vectors (**a**, **b**, and **c**) are also required, which should be specified in Cartesian coordinate [Å]. Type the following command at the directory with emig.csv and sitePE.csv, and the information of optimal paths is printed on the standard output. The direction of optimal path is shown by integer ratios of the lattice vectors. Note that the direction and trajectory of the optimal path could be only an example, particularly in a highly-symmetric crystal. For example, the diffusivity is isotropic in a cubic system, where any long-range path has the same potential barrier.

```
python [MASTEQ_DIR]/tools/optPathSearch.py --emig emig.csv --sitePE sitePE.csv --a 4.23621,0.0,0.0
--b 0.0,4.23621,0.0 --c 0.0,0.0,4.23621 --n_path 3 –gmin 0 > stdout_ops
```

The option of this program is as follows:

-h, --help	Help information. List of options in this code.
--emig	File name of migration energy [eV] and vibrational prefactor [Hz] for every atomic jump. If it is the default name (emig.csv), this argument is not necessary.
--sitePE	File name with the information of site PEs. Default name (sitePE.csv) can be skipped.
--a	Lattice vector **a** in Cartesian coordinate [Å]. e.g.) 4.23621,0.00000,0.00000
--b	Lattice vector **b** in Cartesian coordinate [Å]. e.g.) 0.00000, 4.23621,0.00000
--c	Lattice vector **c** in Cartesian coordinate [Å]. e.g.) 0.00000,0.00000,4.23621
--n_path	Number of optimal paths explored in this program. Default value is 1.
--gmin	Site ID of the global minimum point. If *gmin* = 0, it is searched in sitePE.csv. Default value is 0.

[Theoretical background]

Master equation

Under the independent-particle approximation, the master equation corresponding to the balance of the existence probability p_i of a single particle at site i is given by

$$\frac{\partial p_i(t)}{\partial t} = \sum_j \left[\Gamma_{ji} p_j(t) - \Gamma_{ij} p_i(t)\right],$$

where t is the time, and Γ_{ij} is the jump frequency between sites i and j. In the case of no path for atomic jump between sites i and j, Γ_{ij} is zero. The first and second terms on the right side correspond to the inflow and outflow of the existence probability, respectively. The jump frequency matrix $\mathbf{\Gamma}$ is defined as the negative of the Laplacian matrix for a weighted directed graph, in which the off-diagonal elements are Γ_{ij} and the diagonal elements are $-\sum_{j\neq i}\Gamma_{ij}$. Using the matrix $\mathbf{\Gamma}$, the above master equation can be expressed as

$$\frac{\partial \mathbf{p}}{\partial t} = \mathbf{\Gamma}^{\mathrm{T}}\mathbf{p},$$

where $\mathbf{p}$ is the vector of the existence probabilities of the single particle at all sites, $\mathbf{p} = [p_1(t), \ldots, p_N(t)]^{\mathrm{T}}$ (N: number of sites in the system). The solution of the master equation is expressed using a given initial condition $\mathbf{p}_0$ at $t = 0$,

$$\mathbf{p} = \exp(\mathbf{\Gamma}^{\mathrm{T}} t)\mathbf{p_0}.$$

In a crystal, the dimensions of the matrix $\mathbf{\Gamma}$ and the vector $\mathbf{p}$ can be reduced by the translational symmetry. The master equation is rewritten as

$$\frac{\partial p_i(\mathbf{r},t)}{\partial t} = \sum_{(j,\alpha)\in A_i}\left[\Gamma_{ji}^{\alpha} p_j(\mathbf{r}+\mathbf{s}_{ij}^{\alpha},t) - \Gamma_{ij}^{\alpha} p_i(\mathbf{r},t)\right],$$

where i is the site index in the unitcell ($i = 1, \ldots, n$), $p_i(\mathbf{r},t)$ is the existence probability of the single particle at site i as a function of position $\mathbf{r}$ and time t, and A_i is the set of all adjacent sites to site i. The set A_i includes adjacent sites in different unitcells from the focused unitcell beyond the periodic boundaries, if any. Therefore, when site i has several adjacent sites j in different unitcells, they are distinguished by the unitcell index α. Γ_{ij}^{α} and $\mathbf{s}_{ij}^{\alpha}$ are the jump frequency and jump vector from site i in the focused unitcell to site j in unitcell α, respectively.

This master equation is generally solved in Fourier space. With the Fourier transform, $p_i(\mathbf{r},t)$ is transformed into $P_i(\mathbf{Q},t)$ $\left(= \int \exp(i\mathbf{Q}\mathbf{r})p_i(\mathbf{r},t)d\mathbf{r}\right)$, where $\mathbf{Q}$ is the Fourier variable for position $\mathbf{r}$, resulting in the following master equation in Fourier space,

$$\frac{\partial P_i(\mathbf{Q},t)}{\partial t} = \sum_{(j,\alpha)\in A_i}\left[\Gamma_{ji}^{\alpha} P_j(\mathbf{Q},t)\exp(-i\mathbf{Q}\mathbf{s}_{ij}^{\alpha}) - \Gamma_{ij}^{\alpha} P_i(\mathbf{Q},t)\right].$$

Defining the jump matrix $\mathbf{\Lambda}$ as the $n\times n$ matrix with the elements Λ_{ij},

$$\Lambda_{ij} = \sum_{\alpha}\Gamma_{ij}^{\alpha}\exp(i\mathbf{Q}\mathbf{s}_{ij}^{\alpha}) - \delta_{ij}\sum_{j',\alpha}\Gamma_{ij'}^{\alpha} \quad (\delta_{ij}\text{: Kronecker delta}),$$

the master equation in Fourier space is simply expressed as

$$\frac{\partial \mathbf{P}}{\partial t} = \mathbf{\Lambda}^{\mathrm{T}}\mathbf{P},$$

where $\mathbf{P}$ is the vector of the existence probabilities of the single particle at all n sites in Fourier space. The solution of the master equation is also expressed as $\exp(\mathbf{\Lambda}^{\mathrm{T}}t)\mathbf{P}_0$ ($\mathbf{P}_0$: existence probability vector in Fourier space at $t = 0$). When $\mathbf{\Lambda}^{\mathrm{T}}$ is eigendecomposed into $\mathbf{XYX}^{-1}$ ($\mathbf{Y}$: diagonal matrix with eigenvalues λ_i, $\mathbf{X}$: transformation matrix), the

solution can finally be transformed as follows:

$$\mathbf{P} = \exp(\mathbf{X}\mathbf{Y}\mathbf{X}^{-1}t)\mathbf{P}_0 = \mathbf{X}\exp(\mathbf{Y}t)\mathbf{X}^{-1}\mathbf{P}_0 = \mathbf{X}\begin{pmatrix} e^{\lambda_1 t} & & 0 \\ & \ddots & \\ 0 & & e^{\lambda_n t} \end{pmatrix}\mathbf{X}^{-1}\mathbf{P}_0.$$

To connect the above solution with the diffusion coefficient tensor **D**, the Fick's second law, $\frac{\partial p(\mathbf{r},t)}{\partial t} = \nabla[\mathbf{D}\nabla p(\mathbf{r},t)]$, is solved also in Fourier space, where $p(\mathbf{r},t)$ is the existence probability distribution of a particle at position **r** and time t. Under the initial condition, $p(\mathbf{r},0) = \delta(\mathbf{r})$, the solution is given by

$$P(\mathbf{Q},t) = \exp\left[\left(-\sum_{m,n=x,y,z} D_{mn}\, Q_m Q_n\right)t\right],$$

where D_{mn} and Q_m are the elements of the **D** tensor and the **Q** vector, respectively. Considering the time and spatial scales of atomic diffusion ($t \to \infty$, $|Q| \to 0$), $-\sum_{m,n=x,y,z} D_{mn}\, Q_m Q_n$ coincides with the minimum-magnitude eigenvalue of matrix $\mathbf{\Lambda}^{\mathrm{T}}$. Note that all eigenvalues are negative real numbers due to the mathematical property of matrix $\mathbf{\Lambda}^{\mathrm{T}}$ relevant to the Laplacian matrix in graph theory.