fix atom/swap command¶

Syntax¶

```
fix ID group-ID atom/swap N X seed T keyword values ...
```

- ID, group-ID are documented in fix command
- atom/swap = style name of this fix command
- N = invoke this fix every N steps
- X = number of swaps to attempt every N steps
- seed = random # seed (positive integer)
- T = scaling temperature of the MC swaps (temperature units)
- one or more keyword/value pairs may be appended to args
- keyword = *types* or *mu* or *ke* or *semi-grand* or *region* or *dist_sep/type_sep* and *distribution* or *poly*

```
• types values = two or more atom types

• mu values = chemical potential of swap types (energy units)

• ke value = no or yes

•   no = no conservation of kinetic energy after atom swaps

•   yes = kinetic energy is conserved after atom swaps

• semi-grand value = no or yes

•   no = particle type counts and fractions conserved

•   yes = semi-grand canonical ensemble, particle fractions not conserved

• region value = region-ID

•    region-ID = ID of region to use as an exchange/move volume

• dist_sep (optional x, y, z, xy, xz, or yz)  value >= 1.0

• type_sep value >=1

• distribution values = max_separation, separation number (<=20)
```

- must be used with either dist_sep or type_sep; the distribution recorded is type or distance as is appropriate

- *poly* allows swap of any number of types in non-semi-grand mode

Examples¶

```
fix 2 all atom/swap 1 1 29494 300.0 ke no types 1 2
fix myFix all atom/swap 100 1 12345 298.0 region my_swap_region types 5 6
fix SGMC all atom/swap 1 100 345 1.0 semi-grand yes types 1 2 3 mu 0.0 4.3 -5.0
fix EQUIL swap_rgn atom/swap 1000 10 506 <temp> dist_sep 9 pair region equil poly
distribution 5 5 types 3 4 5 6 7 8 9 10 11 12 13 14
```

Description¶

This fix performs Monte Carlo swaps of atoms of one given atom type with atoms of the other given atom types. The specified T is used in the Metropolis criterion dictating swap probabilities.

Perform X swaps of atoms of one type with atoms of another type according to a Monte Carlo probability. Swap candidates must be in the fix group, must be in the region (if specified), and must be of one of the listed types. Swaps are attempted between candidates that are chosen randomly with equal probability among the candidate atoms. Swaps are not attempted between atoms of the same type since nothing would happen.

All atoms in the simulation domain can be moved using regular time integration displacements, e.g. via fix nvt, resulting in a hybrid MC+MD simulation. A smaller-than-usual timestep size may be needed when running such a hybrid simulation, especially if the swapped atoms are not well equilibrated.

The *types* keyword is required. At least two atom types must be specified.

The *ke* keyword can be set to *no* to turn off kinetic energy conservation for swaps. The default is *yes*, which means that swapped atoms have their velocities scaled by the ratio of the masses of the swapped atom types. This ensures that the kinetic energy of each atom is the same after the swap as it was before the swap, even though the atom masses have changed.

The *semi-grand* keyword can be set to *yes* to switch to the semi-grand canonical ensemble as discussed in (Sadigh). This means that the total number of each particle type does not need to be conserved. The default is *no*, which means that the only kind of swap allowed exchanges an atom of one type with an atom of a different given type. In other words, the relative mole fractions of the swapped atoms remains constant. Whereas in the semi-grand canonical ensemble, the composition of the system can change. Note that when using *semi-grand*, atoms in the fix group whose type is not listed in the *types* keyword are ineligible for attempted conversion. An attempt is made to switch the selected atom (if eligible) to one of the other listed types with equal probability. Acceptance of each attempt depends upon the Metropolis criterion.

The *mu* keyword allows users to specify chemical potentials. This is required and allowed only when using *semi-grand*. All chemical potentials are absolute, so there is one for each swap type listed following the *types* keyword. In semi-grand canonical ensemble simulations the chemical composition of the system is controlled by the difference in these values. So shifting all values by a constant amount will have no effect on the simulation.

The *dist_sep* and *type_sep* keywords define the maximum difference in either distance or type allowed for selected swap pairs. Optionally the distance separation restriction is applied to the named axes (*x* or *y* or *z* or *xy* or *xz* or *yz*). The keyword *distribution* can be used in conjunction with either one to give the accumulated distribution (for successful and unsuccessful swaps separately) of the selected types. The associated values <max_sep> and <sep_no> set the range and divisions of the array returned in a global vector as described below. The distance recorded is projected within the restricted dimensions if that option is used.

The *poly* keyword allows pairwise swapping among many types in the non-semi-grand mode.

This command may optionally use the *region* keyword to define swap volume. The specified region must have been previously defined with a region command. It must be defined with side = *in*. Swap attempts occur only between atoms that are both within the specified region. Swaps are not otherwise attempted.

You should ensure you do not swap atoms belonging to a molecule, or LAMMPS will eventually generate an error when it tries to find those atoms. LAMMPS will warn you if any of the atoms eligible for swapping have a non-zero molecule ID, but does not check for this at the time of swapping.

If not using *semi-grand* this fix checks to ensure all atoms of the given types have the same atomic charge. LAMMPS does not enforce this in general, but it is needed for this fix to simplify the swapping procedure. Successful swaps will swap the atom type and charge of the swapped atoms. Conversely, when using *semi-grand*, it is assumed that all the atom types involved in switches have the same charge. Otherwise, charge would not be conserved. As a consequence, no checks on atomic charges are performed, and successful switches update the atom type but not the atom charge. While it is possible to use *semi-grand* with groups of atoms that have different charges, these charges will not be changed when the atom types change.

Since this fix computes total potential energies before and after proposed swaps, so even complicated potential energy calculations are OK, including the following:

- long-range electrostatics (kspace)
- many body pair styles
- hybrid pair styles
- eam pair styles
- triclinic systems
- need to include potential energy contributions from other fixes

Some fixes have an associated potential energy. Examples of such fixes include: efield, gravity, addforce, langevin, restrain, temp/berendsen, temp/rescale, and wall fixes. For that energy to be included in the total potential energy of the system (the quantity used when performing GCMC moves), you MUST enable the fix_modify *energy* option for that fix. The doc pages for individual fix commands specify if this should be done.

Restart, fix_modify, output, run start/stop, minimize info¶

This fix writes the state of the fix to binary restart files. This includes information about the random number generator seed, the next timestep for MC exchanges, the number of exchange attempts and successes etc. See the read_restart command for info on how to re-specify a fix in an input script that reads a restart file, so that the operation of the fix continues in an uninterrupted fashion.

Note

For this to work correctly, the timestep must **not** be changed after reading the restart with reset_timestep. The fix will try to detect it and stop with an error.

None of the fix_modify options are relevant to this fix.

This fix computes a global vector of length 2, which can be accessed by various output commands. The vector values are the following global cumulative quantities:

- 1 = swap attempts
- 2 = swap accepts

The vector values calculated by this fix are “extensive”.

With ‘distribution’ key word and the associated <max_sep> and <sep_no> values the vector is expanded by 2*<sep_no> + 1 (maximum of 43 – therefore <sep_no> <= 20) to include information about the distribution of swaps with respect to either separation or atom type. These added values are:

- 0 to n-1 – separation for unsuccessful swaps
- n – swaps for which no pair was established
- n+1 to 2n+1 – separation for successful swaps

No parameter of this fix can be used with the *start/stop* keywords of the run command. This fix is not invoked during energy minimization.

Restrictions¶

This fix is part of the MC package. It is only enabled if LAMMPS was built with that package. See the Build package doc page for more info.

Related commands¶

fix nvt, neighbor, fix deposit, fix evaporate, delete_atoms, fix gcmc, fix mol/swap

Default¶

The option defaults are ke = yes, semi-grand = no, mu = 0.0 for all atom types.

(Sadigh) B Sadigh, P Erhart, A Stukowski, A Caro, E Martinez, and L Zepeda-Ruiz, Phys. Rev. B, 85, 184203 (2012).