Indication on the kMC code

As indicated in lecture slide the code build in four level, 00, 01, 02 and user level wich permit to implement a system you want. All the variable are accessible in kmc_type structue in which you find also another structure event_type. The definition of this structure can be found in file lev00_new_types.f90. Moreover the memories alllocation of the array is occurs by the call of two subroutines, builder_kmc_type() and builder_event_type() which can be found in lev00_module.f90. The first is call in the source part of the code and the second is call in the read_event() subroutine which is in user part of the code.

The allocation procede in two part:

1. The allocation of array with the prefix "h_" following by the name of the array. You can find these arrays in hidden_table module.

2. Link between the "h_" array and c pointer which has the same but with the prefix "ptr_".

If you use the c version of the exercice you can use directly the c pointer with prefix "ptr_". In fortran version is different because the definition of a fortran pointer is different. A fortran pointer is define with address of the first element of array and the size of array. To do that I write some subroutine, leak_int1_ptr for instance, which can be found in lev00_module.f90. These routines link a fortran pointer with a c pointer. The routine used depend of the type of your array and its dimension. The fortran pointer are declare in "variables" module in user part of the code. In fortran version of the exercice I let write some call to these subroutines to let you see how you can use it.

On the storage in the array

Now I will explain the way choose to store the information in arrays. In use the c notation but in fortran is the same. I begin with the neighbor list. The storage of the neighbors list is occurs in lev02_neig_list.f90 and we use two arrays, ptr_nneig and ptr_neig with "tot_sites" is total number of site in my system.

We allocation tot_sites integer to ptr_nneig and nvois*tot_sites integer to ptr_neig.The variable "nvois" is declared as a parameter in lev00_new_types.f90. Is initialize at 10 but is not accessible in user part of the code. For all the exercice I gave you this value is enougth.

I store in ptr_neig the number of the neighbor and the list of the neighbors of each sites. So to know where begin each list in ptr_neig we use ptr_nneig. So for a site "is" ptr_nneig[is] = pos_list and ptr_neig[pos_list] = number of neighbors in the list. Then the list of neighbors begin at ptr_neig[pos_list + 1].

In the exercie I ask you to store the rate of the site in ptr_rate and in ptr_event_rate. ptr_rate[i] is the sum of the rate of all event possible in the site "i". The ptr_event_rate is the rate list of the event possible in site "i". These information can be stored in the same way as the neighbors list and you can take the value of ptr_nneig[is] = pos_list to store the number of event and the rate list of the event.

So same logical can be used for ptr_event_site wich you store the list of the event index possible for site "i".