Procedure decompse_whole_grid

Input: 1) the whole grid *G* after preprocessing; 2) the set of active computing resource units *C*

Output: 1) kernel subgrid domains, each of which has been assigned to an active computing resource in *C*; 2) search tree of subgrid domains

(1) If *G* is a spherical grid and covers the south pole, generate a circular kernel subdomain corresponding to the south pole, assign it to an active computing resource *c1*, and insert it into the search tree; if *c1* does not have free computational capacity for new kernel subgrid domains, remove *c1* from *C*.
(2) If *G* is a spherical grid and covers the north pole, generate a circular kernel subdomain corresponding to the north pole, assign it to an active computing resource *c2*, and insert it into the search tree; if *c2* does not have free computational capacity for new kernel subgrid domains, remove *c2* from *C*.
(3) For the remaining subgrid domain *D*, call *decompose_subgrid*(*D*, *G*).

Procedure decompose_subgrid

Input: 1) a subgrid domain *D*; 2) a set of active computing resource units *C*

Output: 1) kernel subgrid domains of *D*, each of which has been assigned to an active computing resource in *C*; 2) update of the search tree of subgrid domains

(1) If *D* is a cyclic domain and *C* contains only one computing resource unit *c1*, cut *D* into two acyclic sub domains with the same area, assign them to *c1*, insert them into the search tree as the children of *D*, and then return
(2) If *C* contains only one computing resource unit *c1*, assign *D* to *c1* and then return
(3) Divide *C* into two subsets (*C1* and *C2*), which have as equal as possible numbers of computing resource units
(4) Cut *D* into two sub domains (*D1* and *D2*) at the long edge of *D*, according to the total free computational capacity of *C1* and *C2*
(5) Insert *D1* and *D2* into the search tree as the children of *D*
(6) If the current MPI process has common computing resource units with *C1*, call *decompose_subgrid*(*D1*, *C1*)
(7) If the current MPI process has common computing resource units with *C2*, call *decompose_subgrid*(*D2*, *C2*)

Procedure expand_sub_grid_domain

Input: 1) a kernel subgrid domain *D*; 2) a given expansion rate

Output: 1) expanded subgrid domain of *D*; 2) update of the search tree of subgrid domains

(1) Estimate a halo region based on the expansion rate
(2) Search the kernel subgrid domains that overlap with the estimated halo region, generate new kernel subgrid domains and insert them into the search tree if required
(3) If the estimated halo region has more points than expected, shrink the halo region gradually
(4) After the halo region is determined, generate the expanded subgrid domain of *D*, and record the neighborhoods corresponding to *D* in the search tree