ORB5 Memory optimisation
Re-using Larmor marker arrays

Huw Leggate

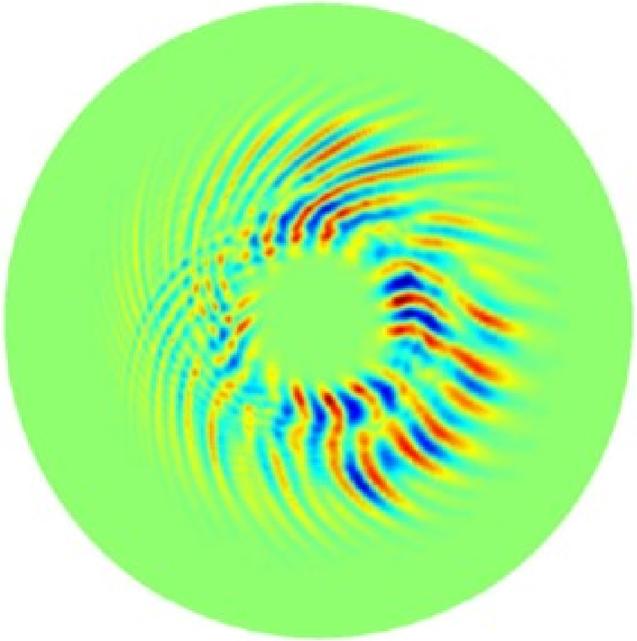
Dublin City University/Advanced Computing Hub - Garching

ACH meeting – 27th-28th Nov 2024 – Barcelona Super Computing Centre

ORB5

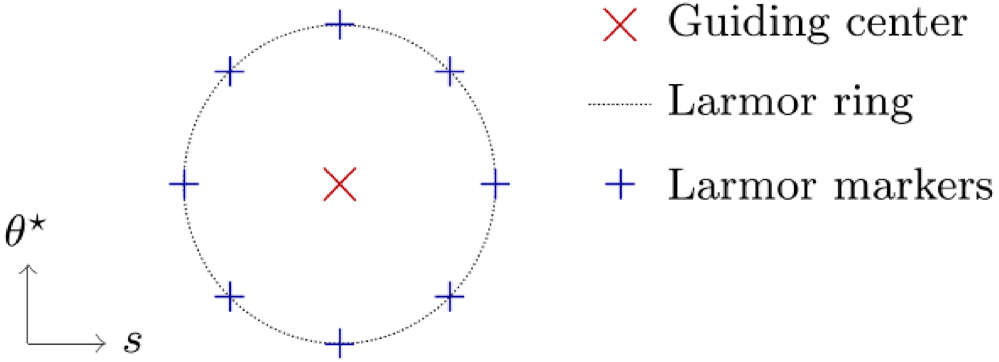
"ORB5: a global electromagnetic gyrokinetic code using the PIC approach in toroidal geometry" [for details, see Lanti 2020]

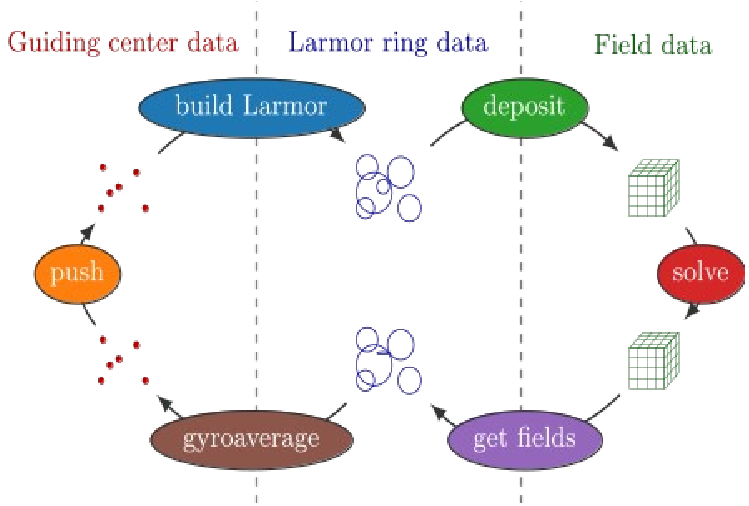
- δ-f modified distribution function discretized with PIC
- Fields solved using finite elements
- Global electromagnetic (EM) simulations
- Adiabatic, hybrid, or kinetic electrons, EM drift-kinetic electrons (or fluid)
- Previously used for turbulence studies



ORB5 memory optimisation

- Simple test case – 10^7 particles, deuterium 4 markers, electrons
  - Memory footprint ~20 GB – (Larmor points - 8 GB...)
- Optimisation reuses reduced size Larmor points arrays

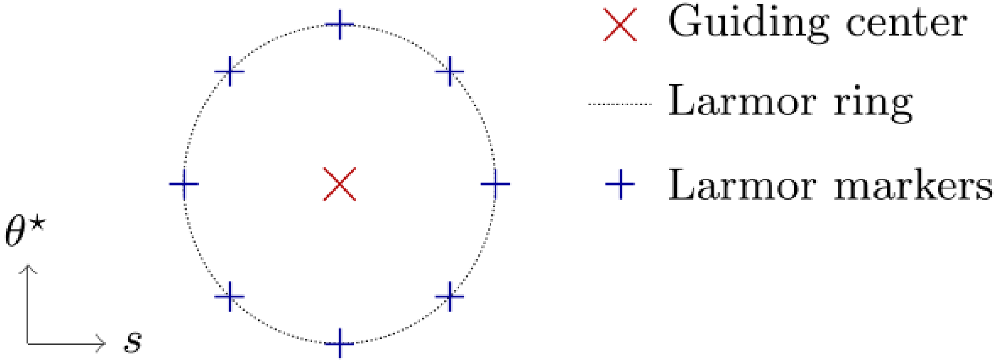


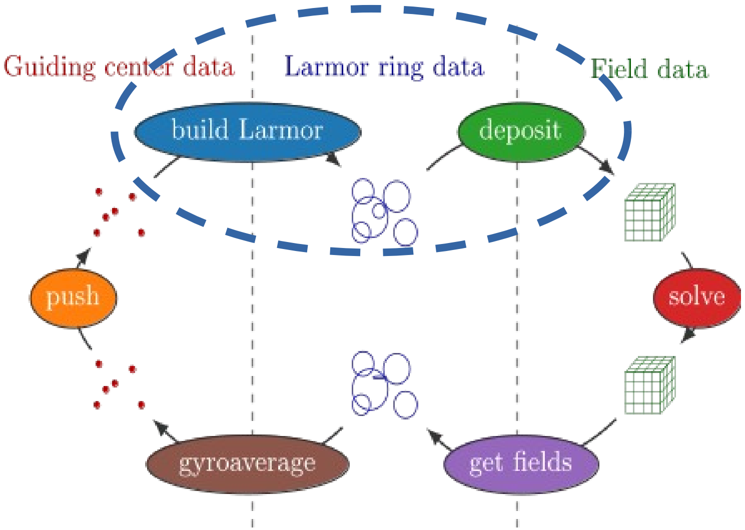


Ohana 2021 https://doi.org/10.1016/j.cpc.2020.107208

ORB5 memory optimisation

- Simple test case – 10^7 particles, deuterium 4 markers, electrons
  - Memory footprint ~20 GB – (Larmor points - 8 GB...)
- Optimisation reuses reduced size Larmor points arrays



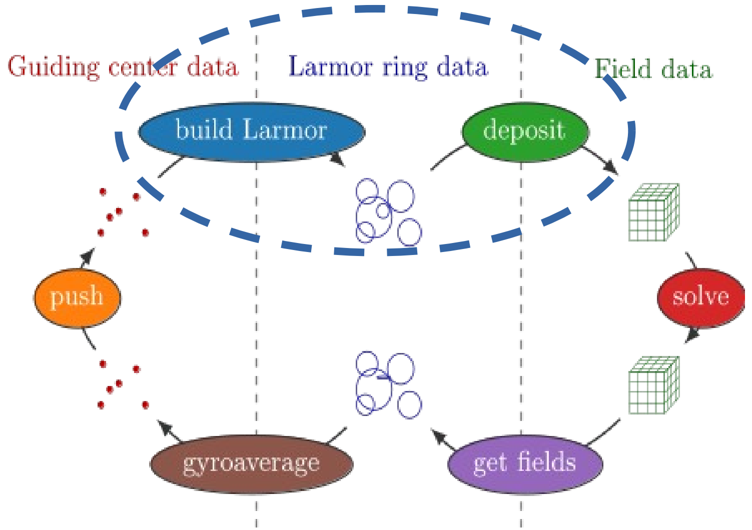


Ohana 2021 https://doi.org/10.1016/j.cpc.2020.107208

ORB5 memory optimisation

- Simple test case – 10^7 particles, deuterium 4 markers, electrons
    - Memory footprint ~20 GB – (Larmor points - 8 GB...)
- Optimisation reuses reduced size Larmor points arrays

1) Charge & current deposition

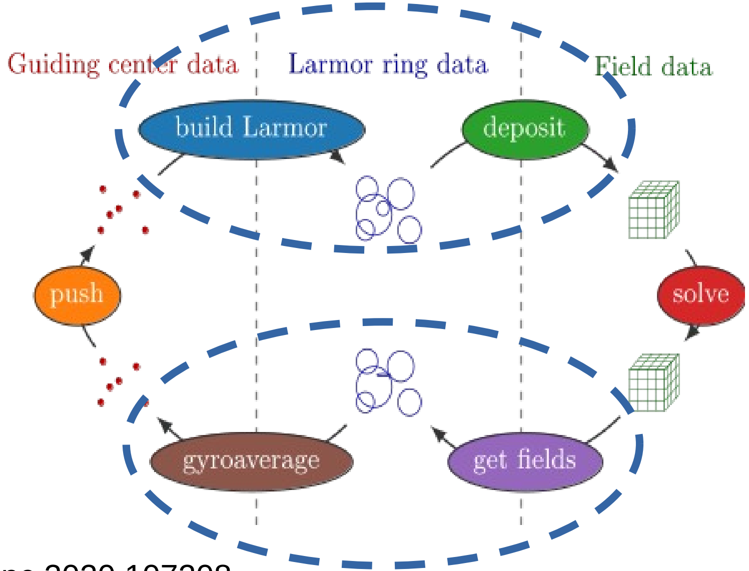


Ohana 2021 https://doi.org/10.1016/j.cpc.2020.107208

ORB5 memory optimisation

- Simple test case – 10^7 particles, deuterium 4 markers, electrons
  - Memory footprint ~20 GB – (Larmor points - 8 GB...)
- Optimisation reuses reduced size Larmor points arrays

1) Charge & current deposition



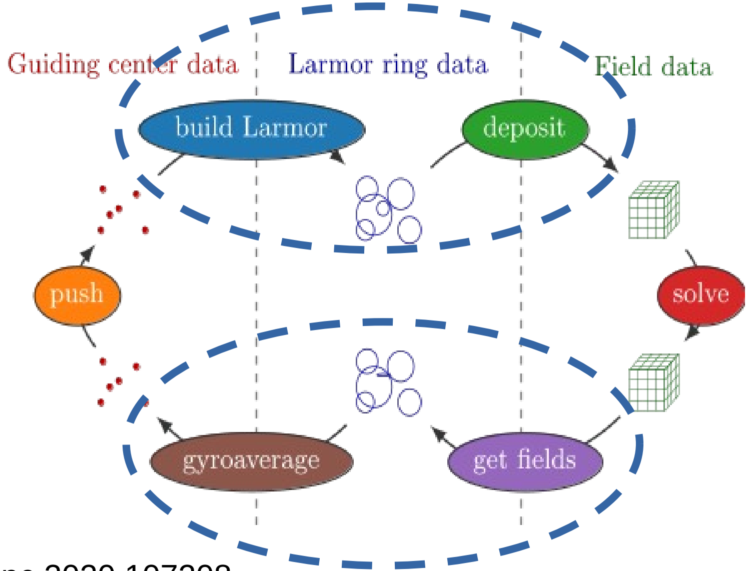
Ohana 2021 https://doi.org/10.1016/j.cpc.2020.107208

ORB5 memory optimisation

- Simple test case – 10^7 particles, deuterium 4 markers, electrons
  - Memory footprint ~20 GB – (Larmor points - 8 GB...)
- Optimisation reuses reduced size Larmor points arrays

1) Charge & current deposition

2) Field calculation & gyro-averaging



Ohana 2021 https://doi.org/10.1016/j.cpc.2020.107208

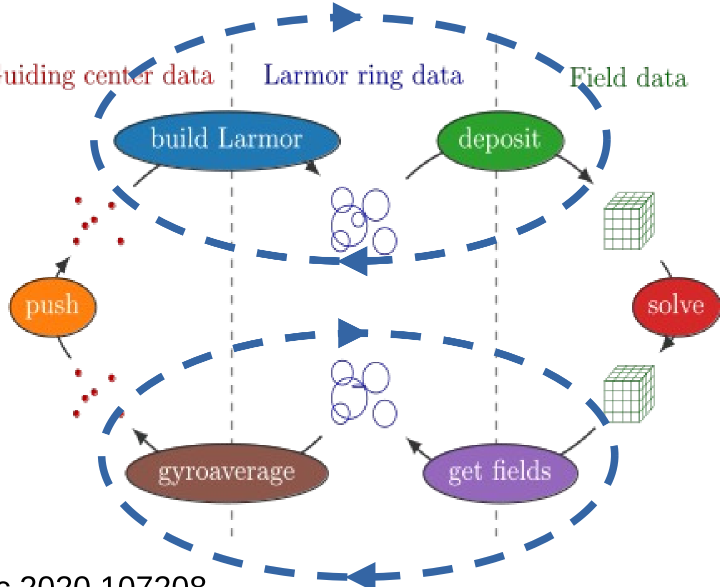
ORB5 memory optimisation

- Simple test case – 10^7 particles, deuterium 4 markers, electrons
  - Memory footprint ~20 GB – (Larmor points - 8 GB...)
- Optimisation reuses reduced size Larmor points arrays

1) Charge & current deposition

2) Field calculation & gyro-averaging

Iterate using same arrays



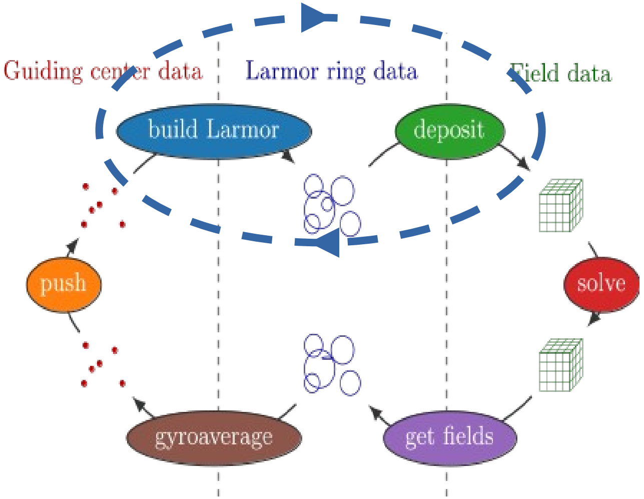
Ohana 2021 https://doi.org/10.1016/j.cpc.2020.107208

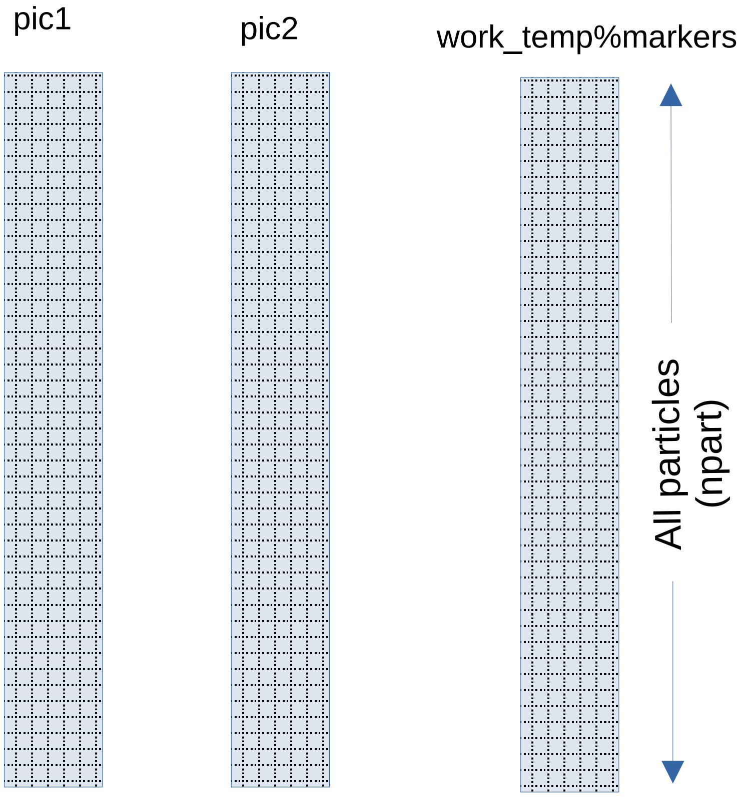
Code changes

- natts_iter added to basic namelist – default 1 - check on sorting
  - Defines number of iterations over Larmor points
- marker_type

```
type, public :: marker_type
   integer                                   :: nmarkers          !< Number of markers
   integer                                   :: nmarkers_max      !< Allocated number of markers
   integer                                   :: atts_offset       !< Offset in particles for this atts
   integer                                   :: atts_npart        !< Number of particles stored
   real,    dimension(:,:), allocatable :: atts              !< Marker attributes
   real,    dimension(:,:), allocatable :: atts_sorted       !< Sorted marker attributes
end type marker_type
```

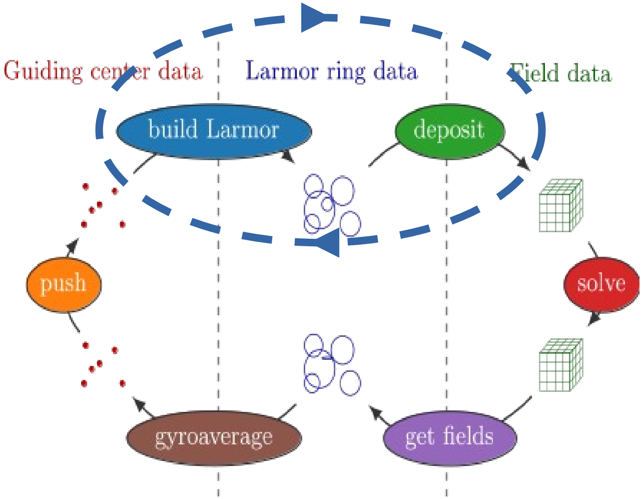
Reusing Reduced ATT arrays - Deposition

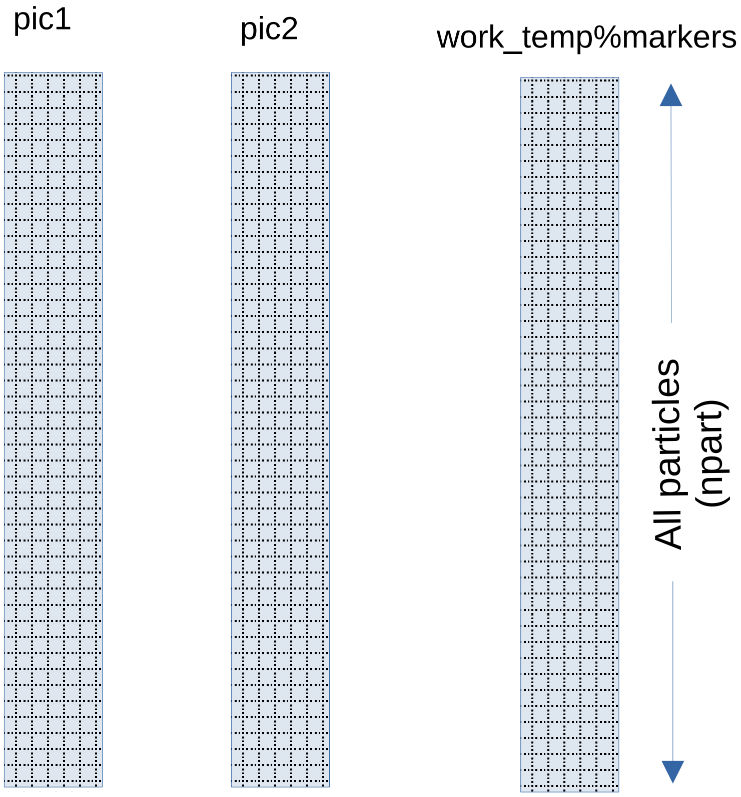




Reusing Reduced ATT arrays - Deposition

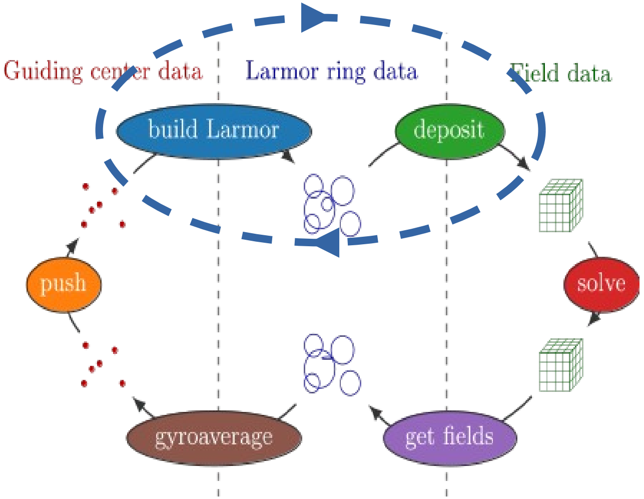
- call get_larmor_indices()

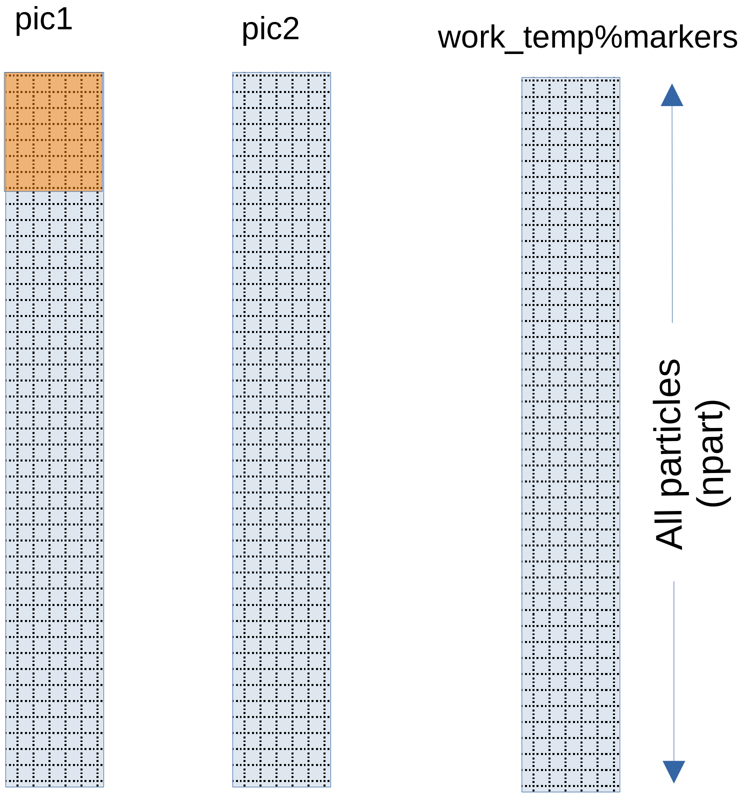




Reusing Reduced ATT arrays - Deposition

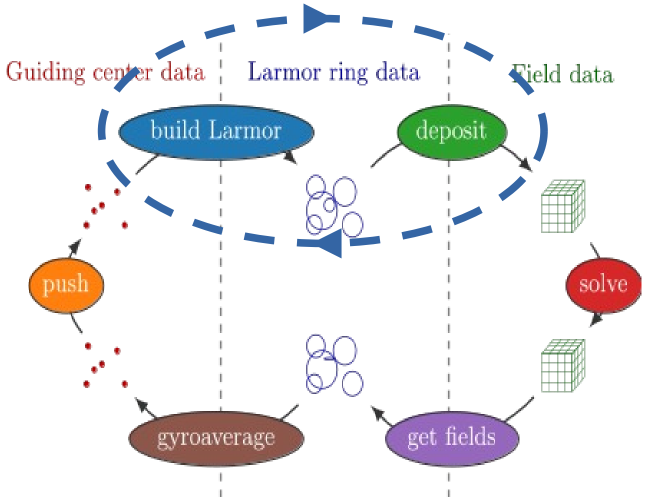
- call get_larmor_indices()

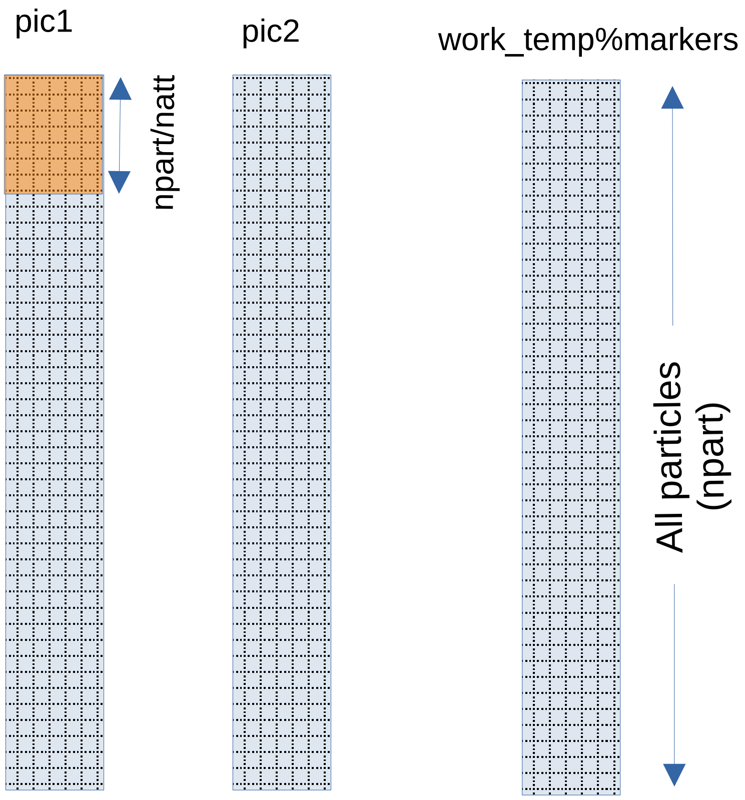




Reusing Reduced ATT arrays - Deposition

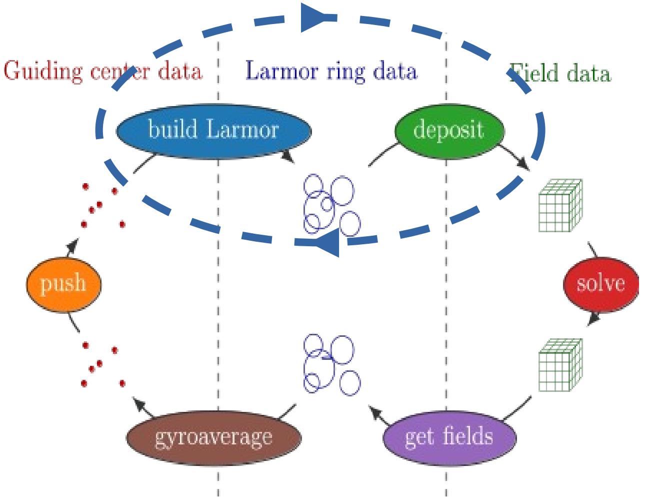
- call get_larmor_indices()

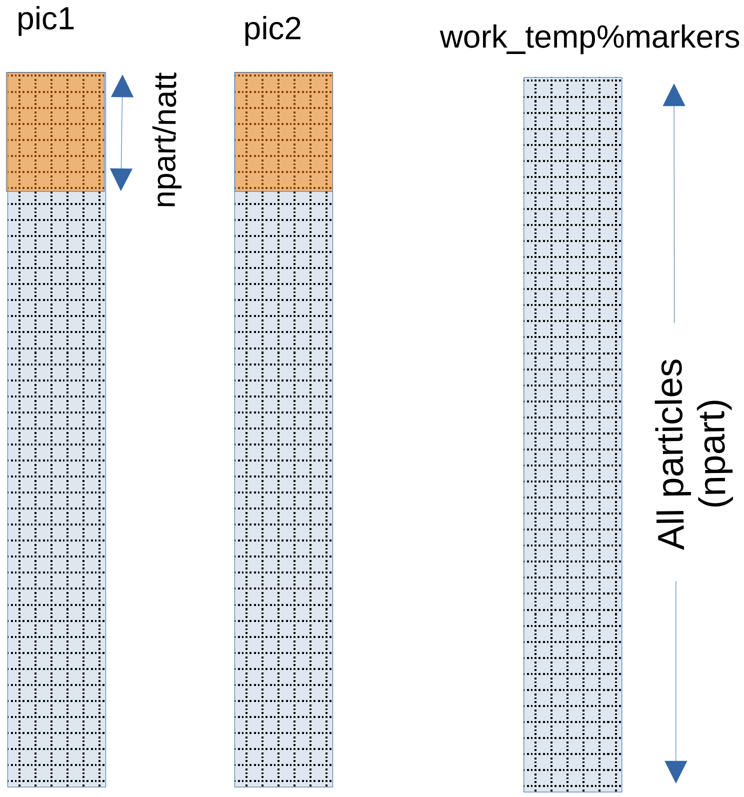




Reusing Reduced ATT arrays - Deposition

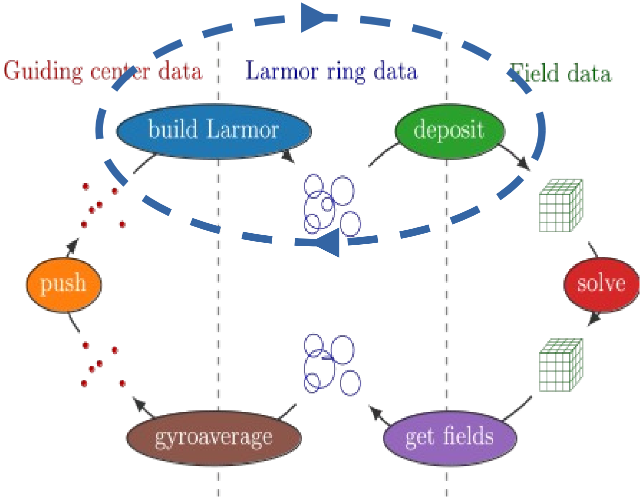
- call get_larmor_indices()

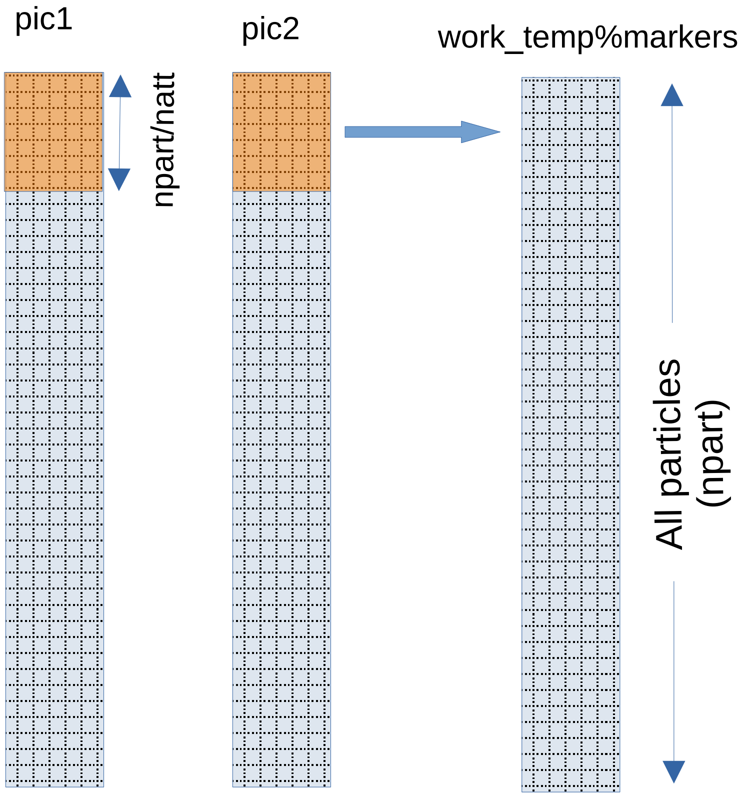




Reusing Reduced ATT arrays - Deposition

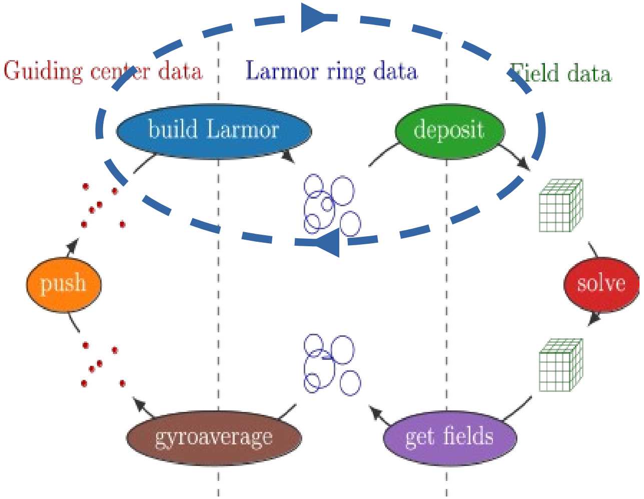
- call get_larmor_indices()

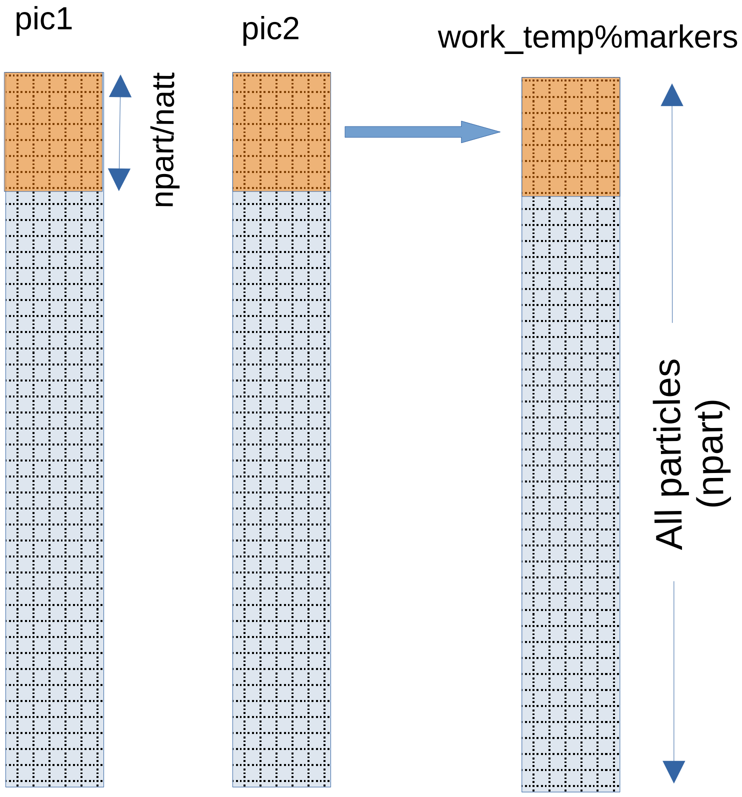




Reusing Reduced ATT arrays - Deposition

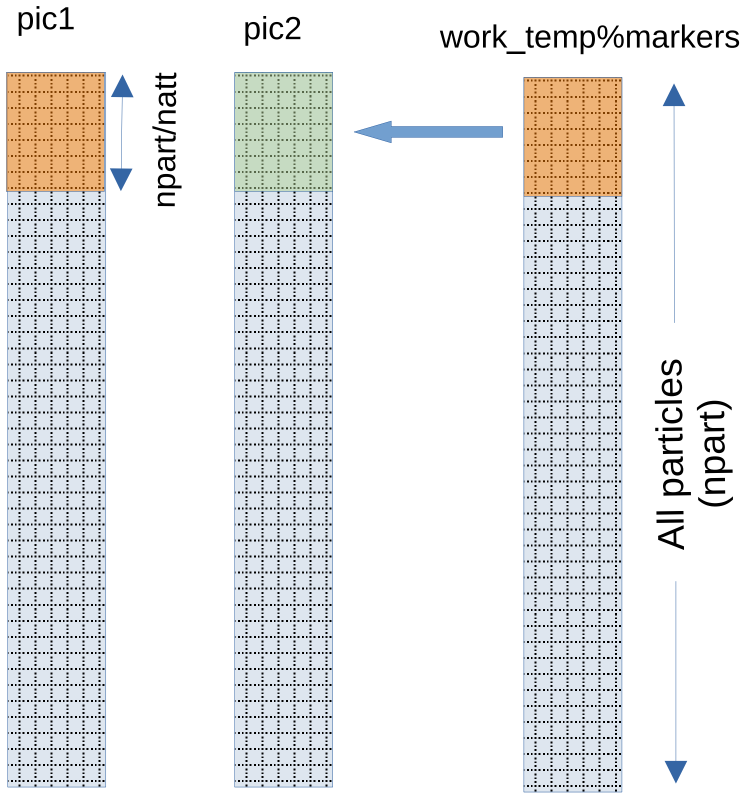
- call get_larmor_indices()

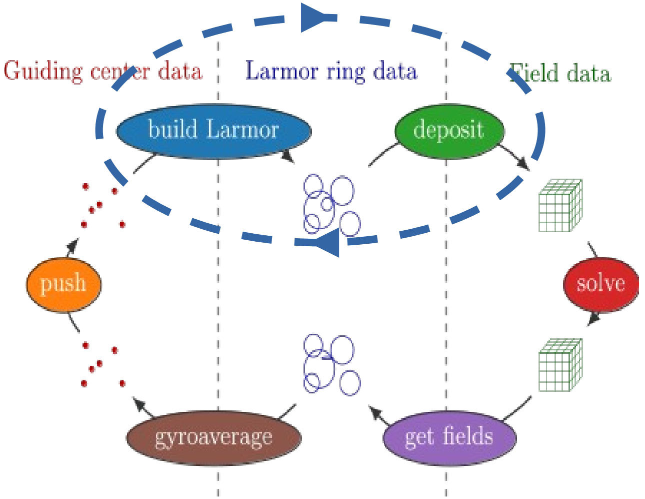




Reusing Reduced ATT arrays - Deposition

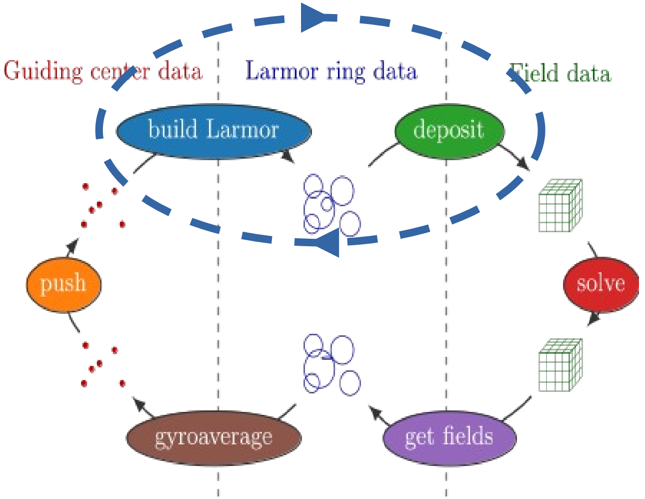
- call get_larmor_indices()

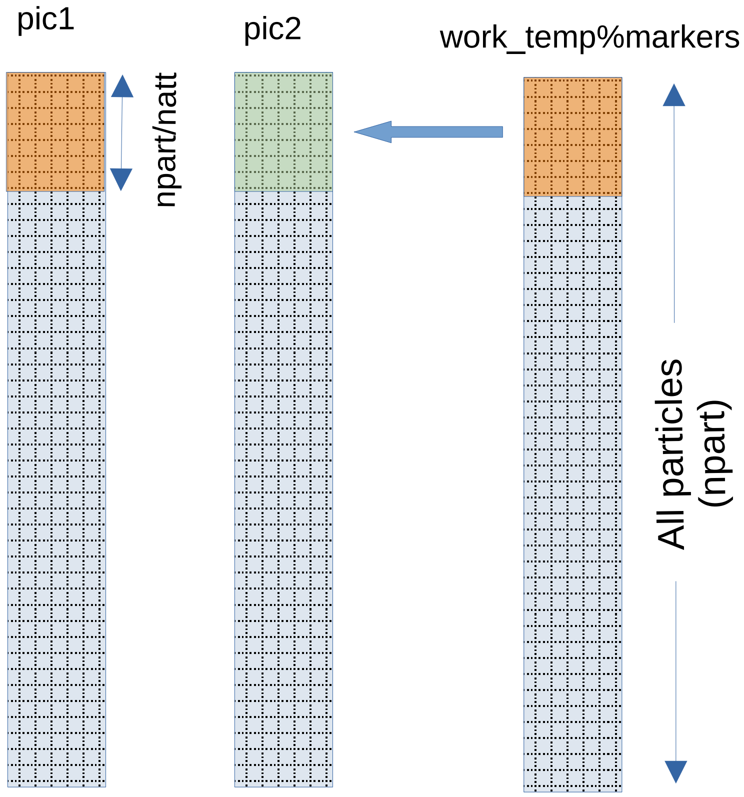




Reusing Reduced ATT arrays - Deposition

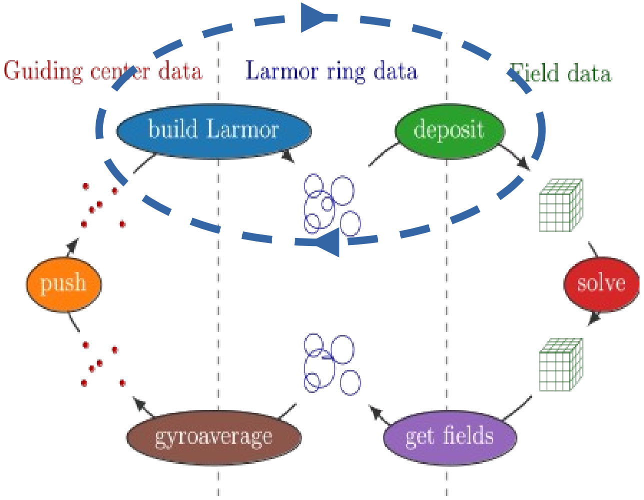
- call get_larmor_indices()
- call compute_all_larmor_positions()

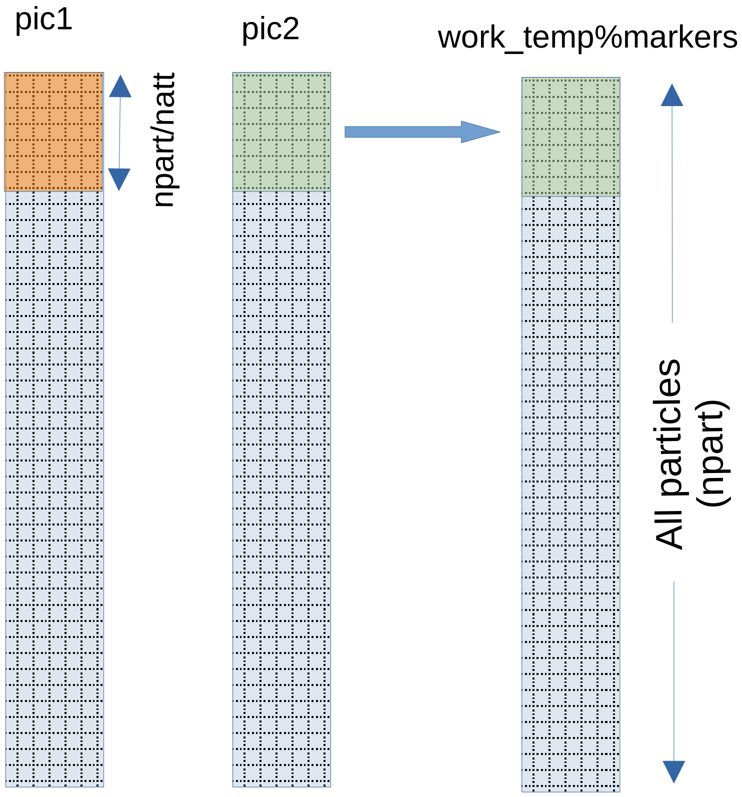




Reusing Reduced ATT arrays - Deposition

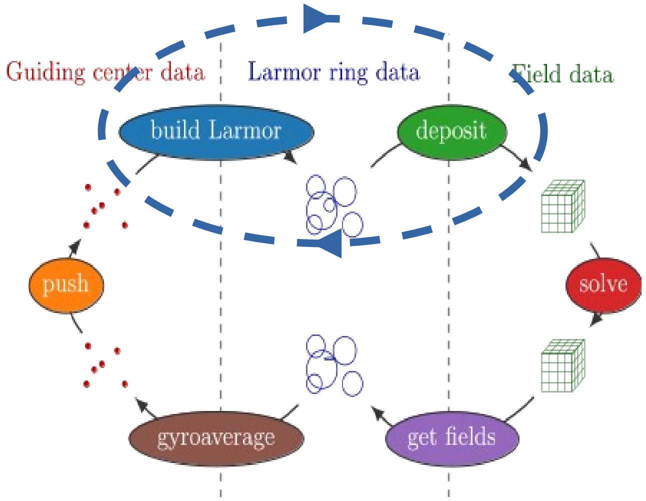
- call get_larmor_indices()
- call compute_all_larmor_positions()

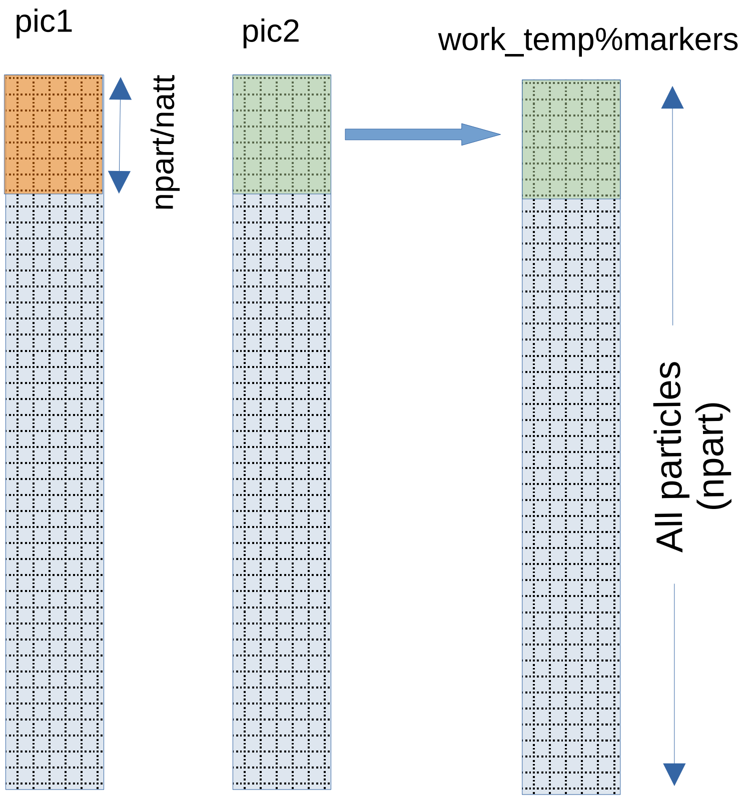




Reusing Reduced ATT arrays - Deposition

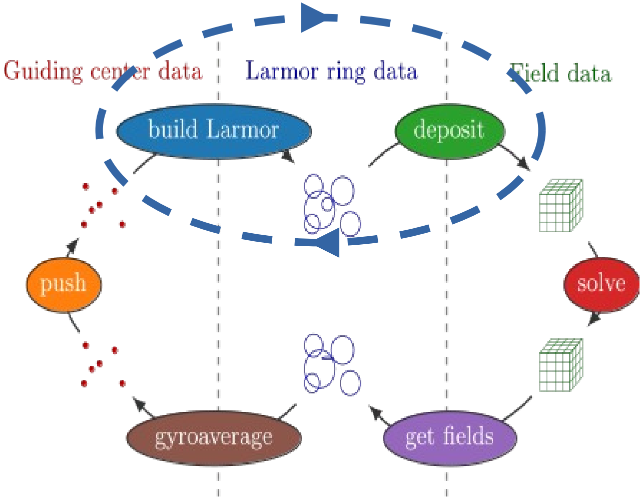
- call get_larmor_indices()
- call compute_all_larmor_positions()

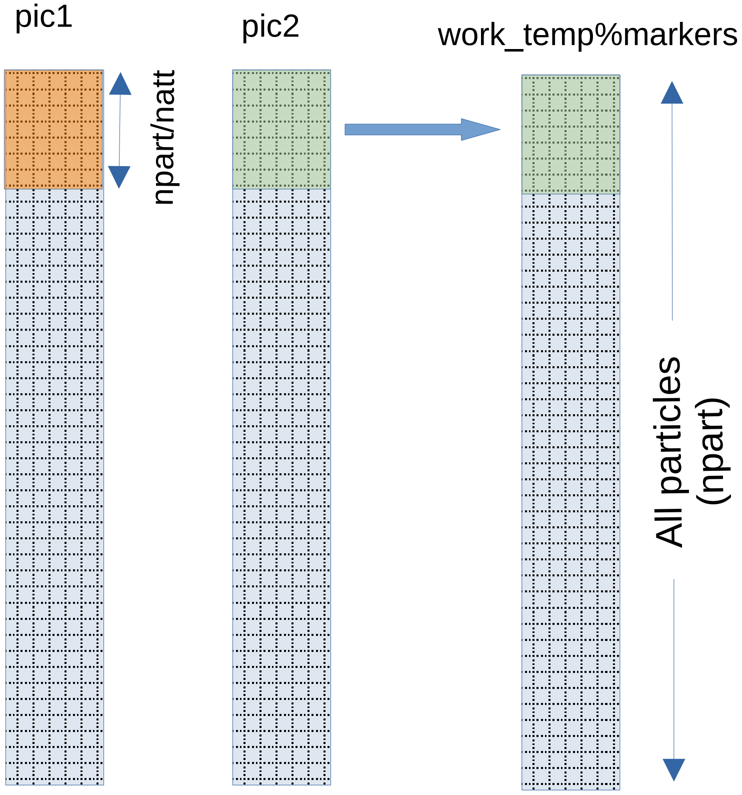




Reusing Reduced ATT arrays - Deposition

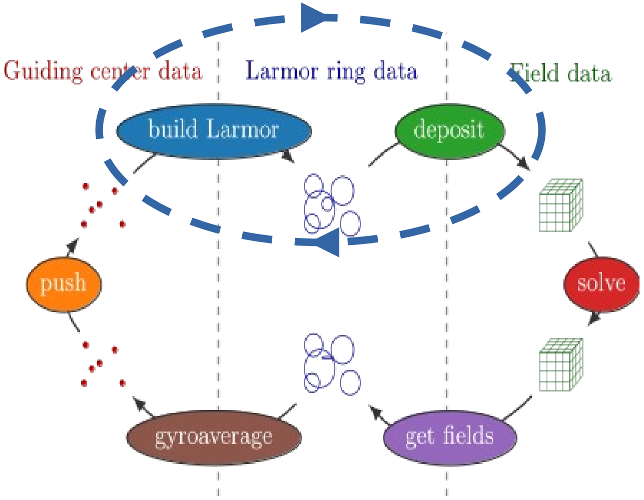
- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()

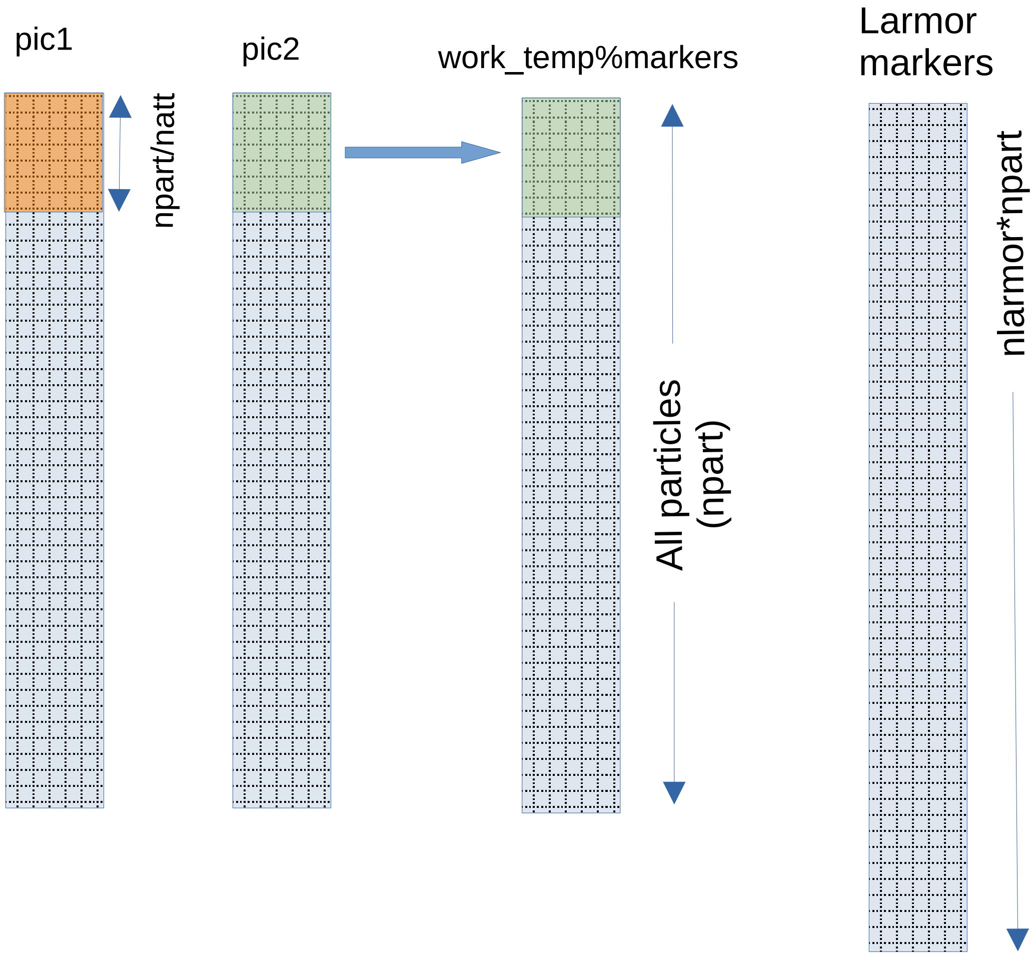




Reusing Reduced ATT arrays - Deposition

- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()





pic1

npart/natt

pic2

work_temp%markers

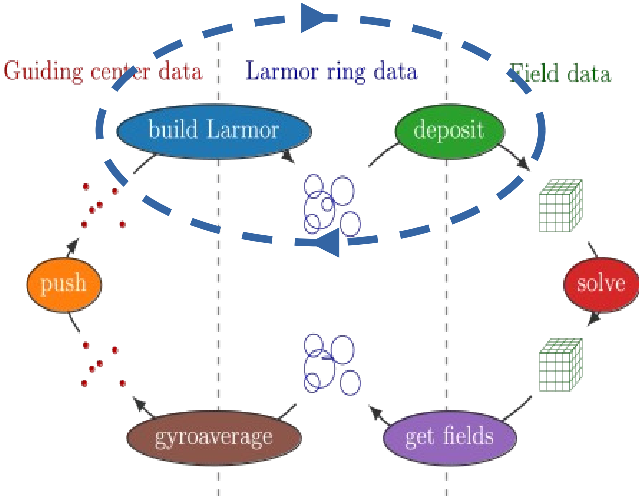
All particles (npart)

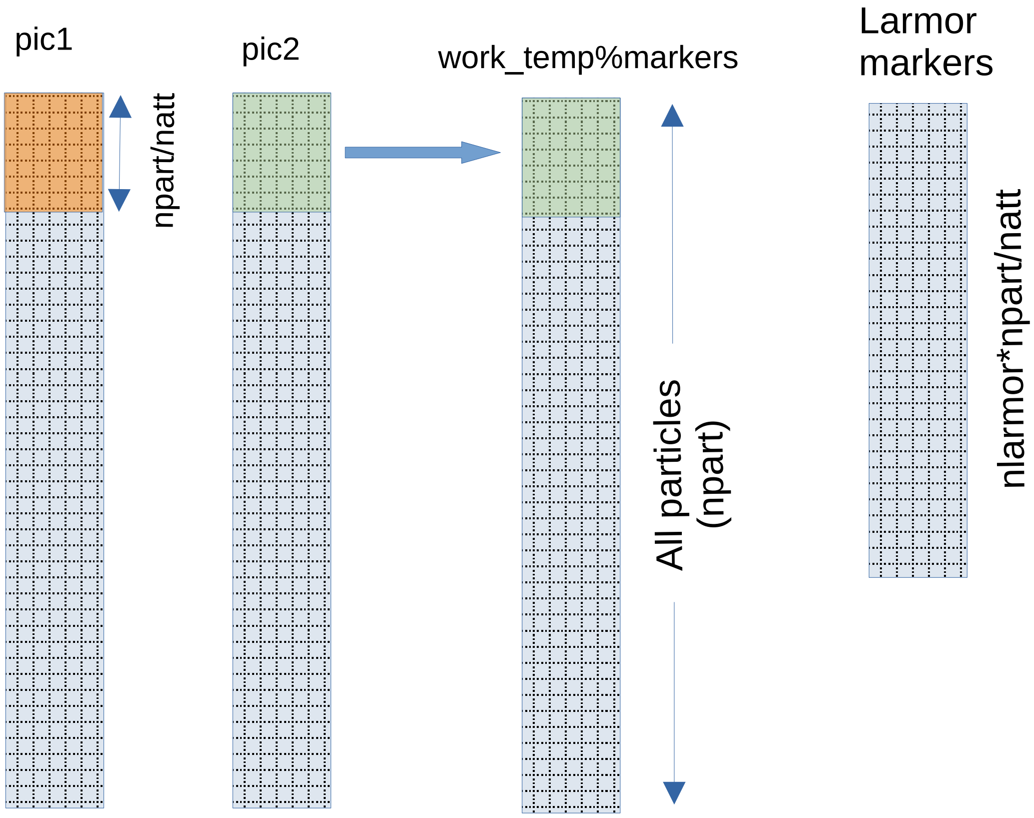
Larmor markers

nlarmor*npart

Reusing Reduced ATT arrays - Deposition

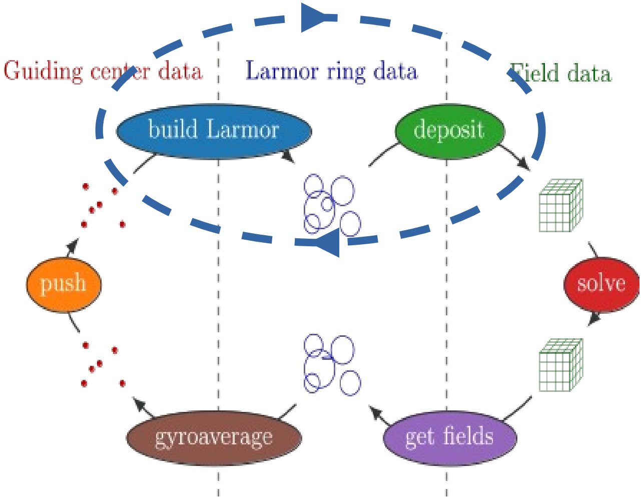
- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()

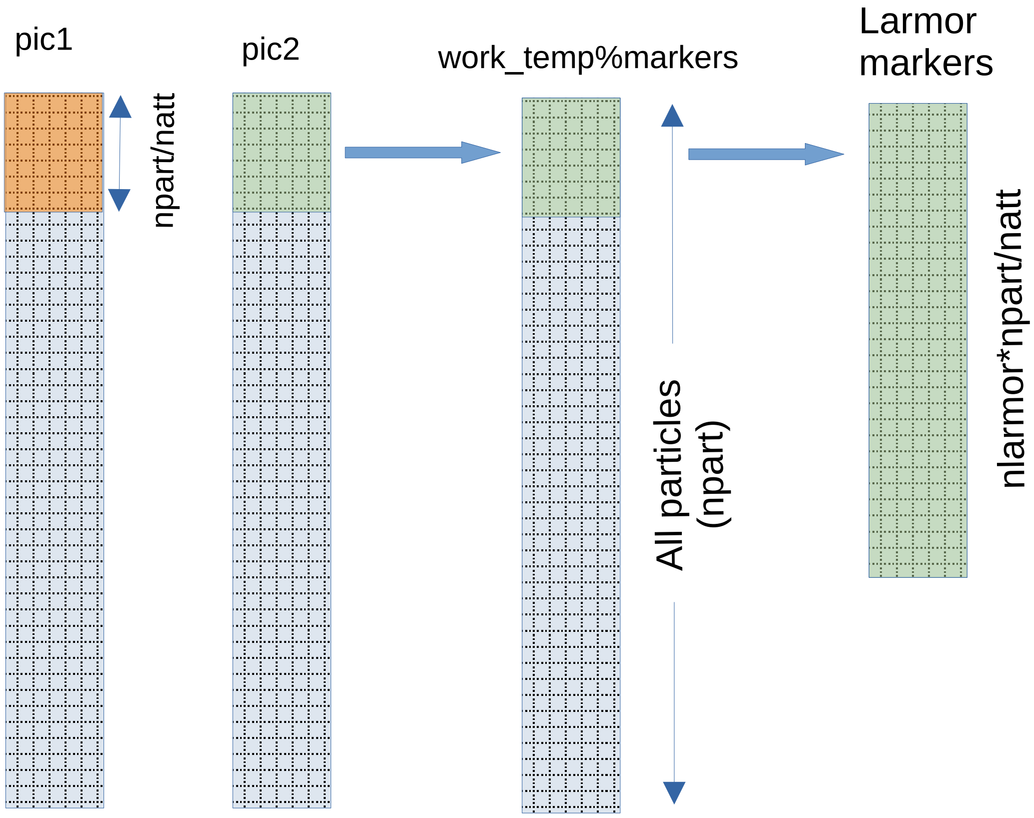




Reusing Reduced ATT arrays - Deposition

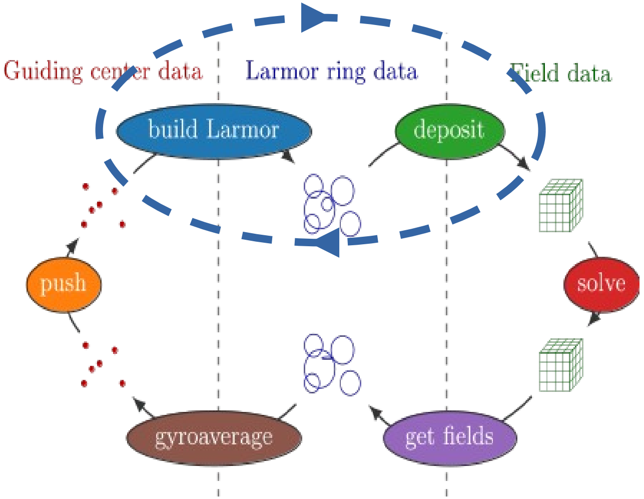
- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()

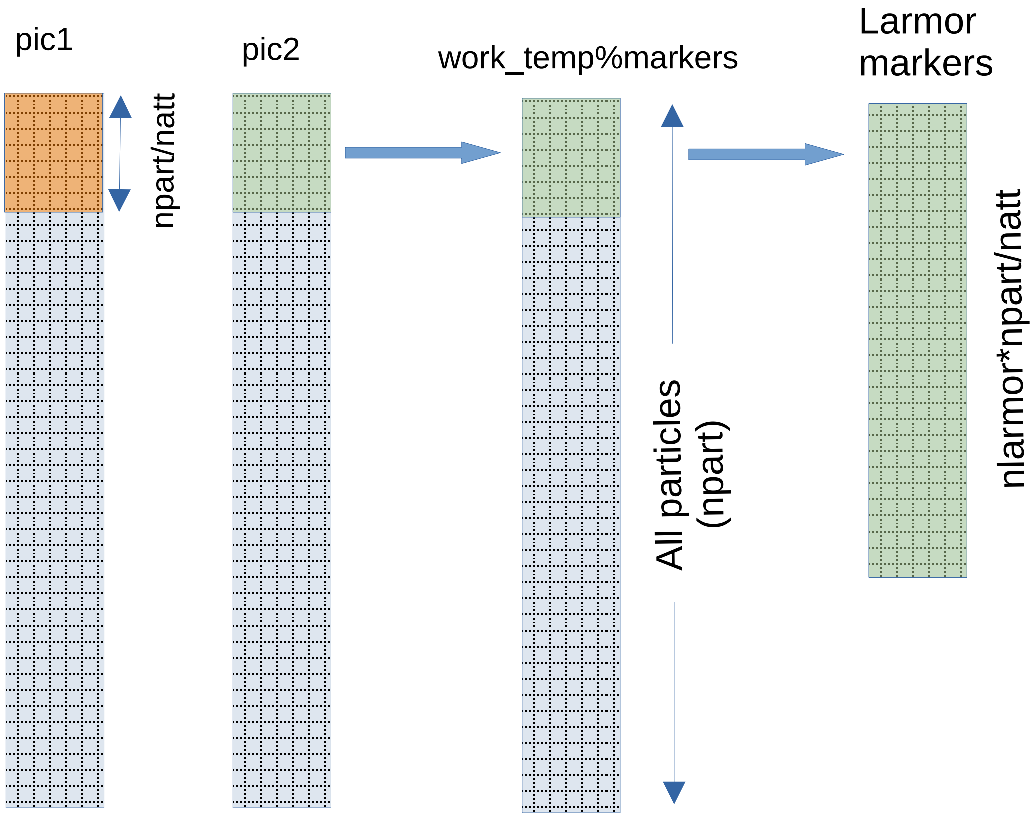




Reusing Reduced ATT arrays - Deposition

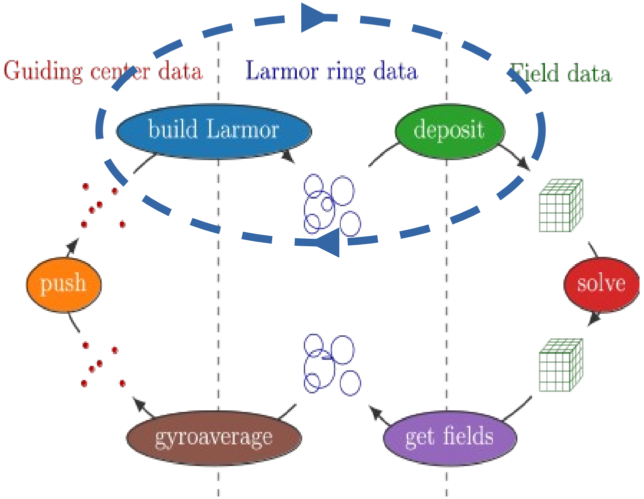
- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()
- call set_rho()

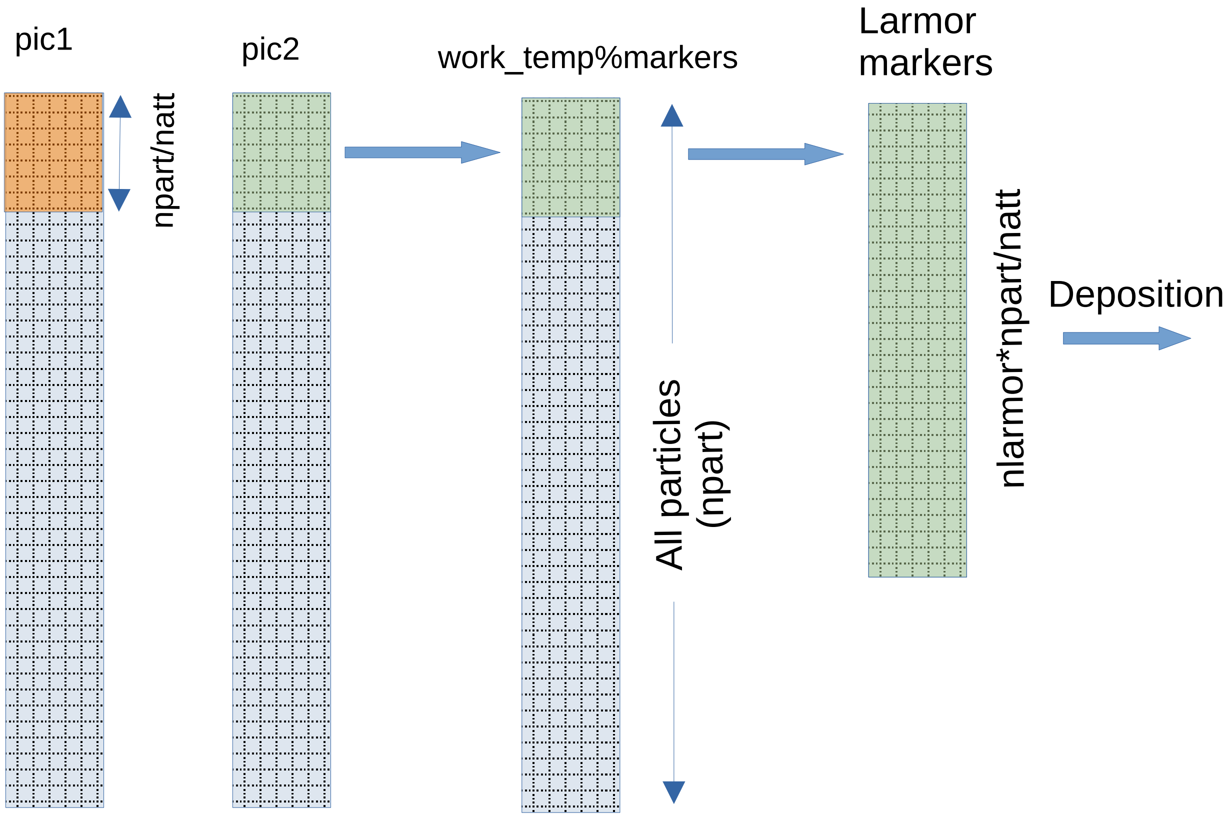




Reusing Reduced ATT arrays - Deposition

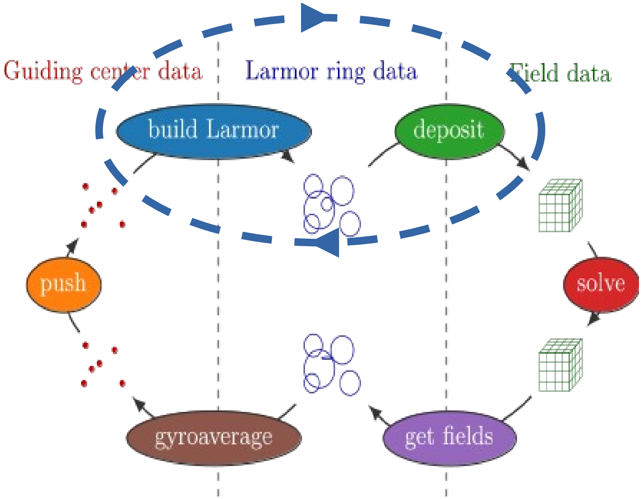
- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()
- call set_rho()

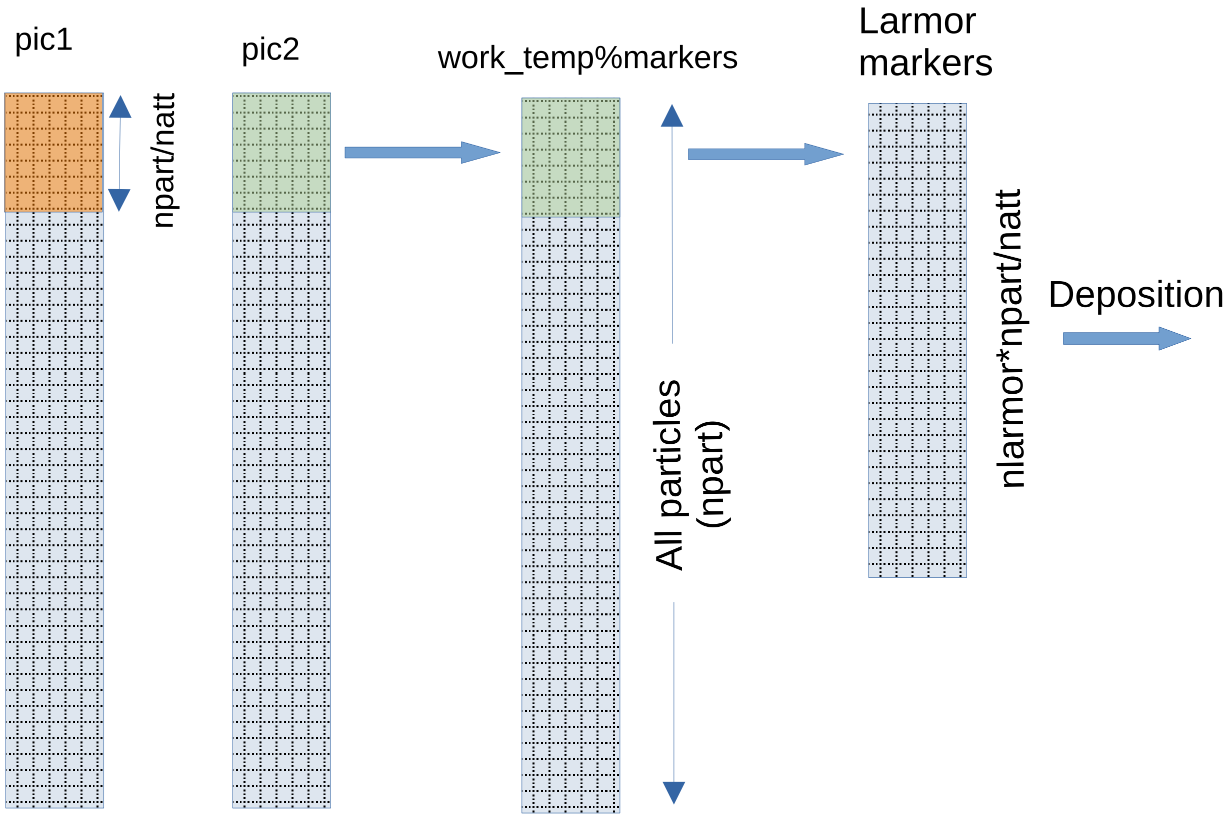




Reusing Reduced ATT arrays - Deposition

- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()
- call set_rho()




pic1

npart/natt

pic2

work_temp%markers

All particles (npart)

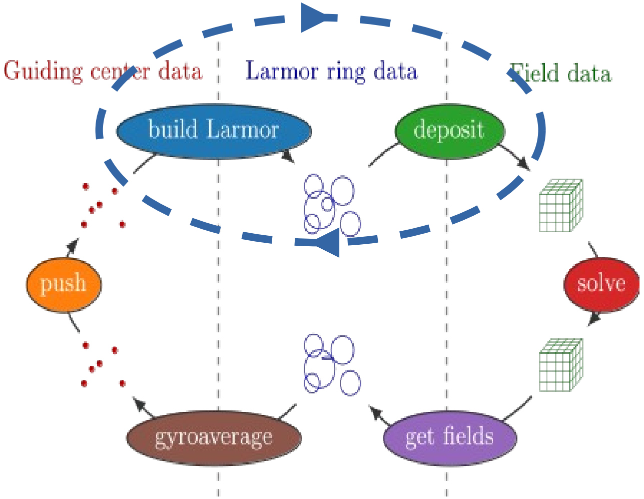
Larmor markers

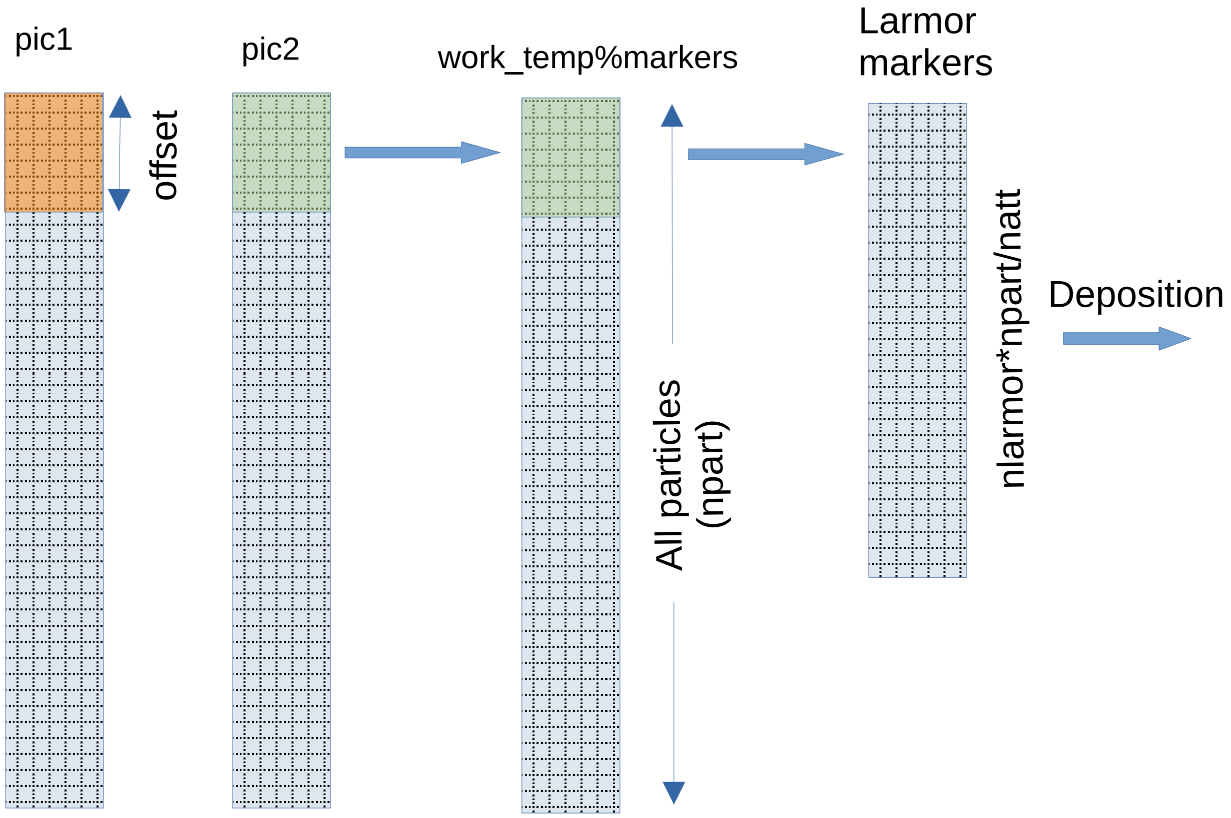
nlarmor*npart/natt

Deposition

Reusing Reduced ATT arrays - Deposition

- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()
- call set_rho()





pic1

offset

pic2

work_temp%markers

All particles (npart)

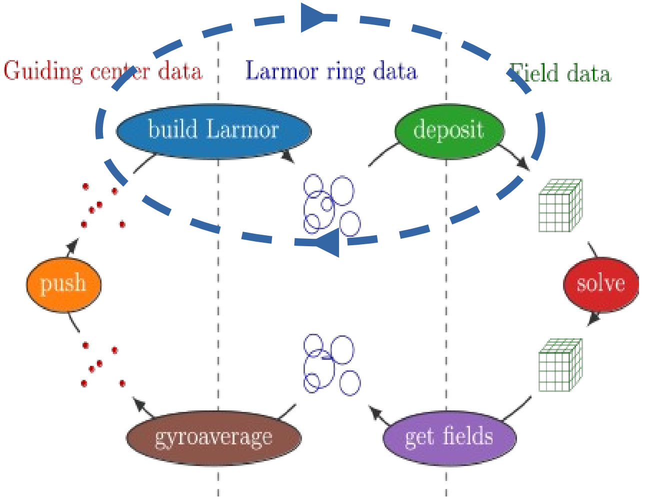
Larmor markers

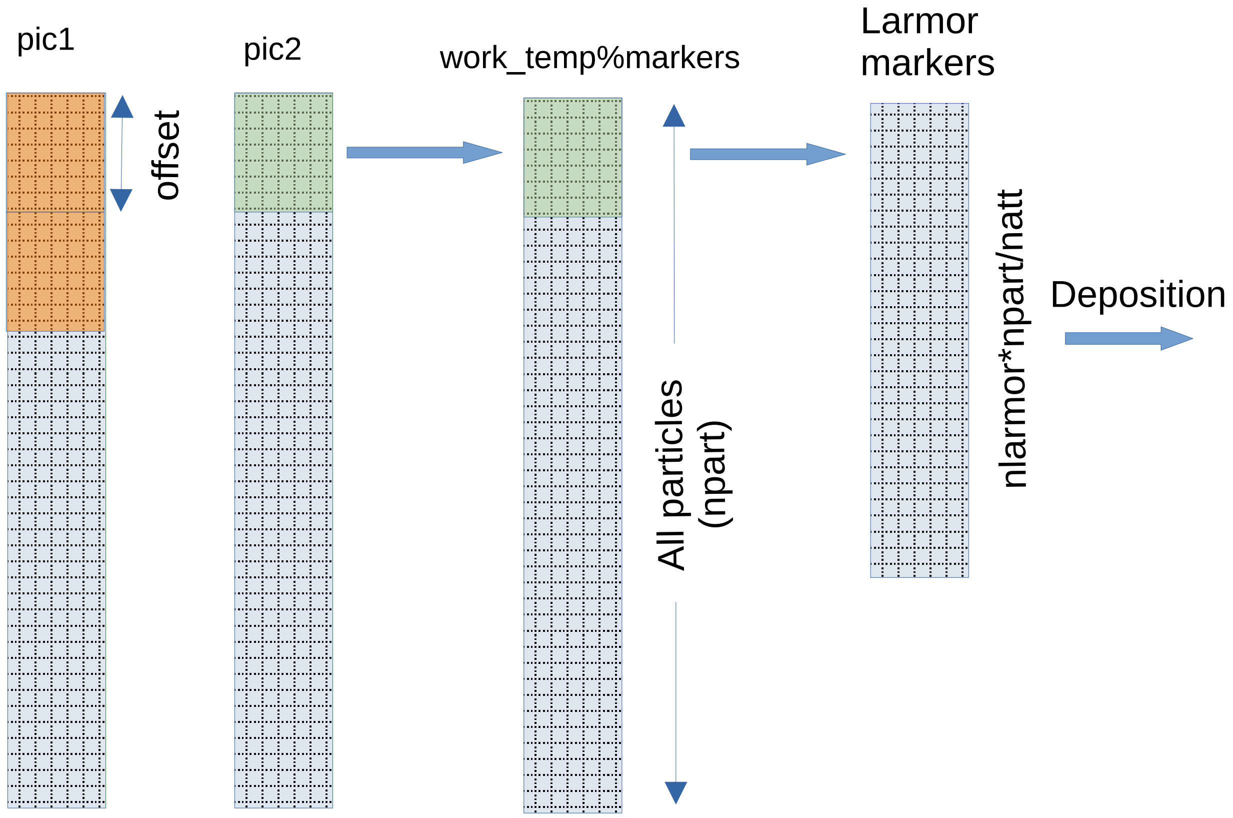
nlarmor*npart/natt

Deposition

Reusing Reduced ATT arrays - Deposition

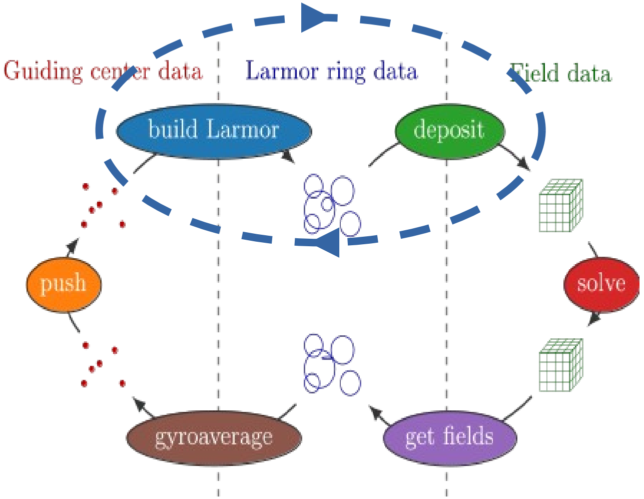
- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()
- call set_rho()

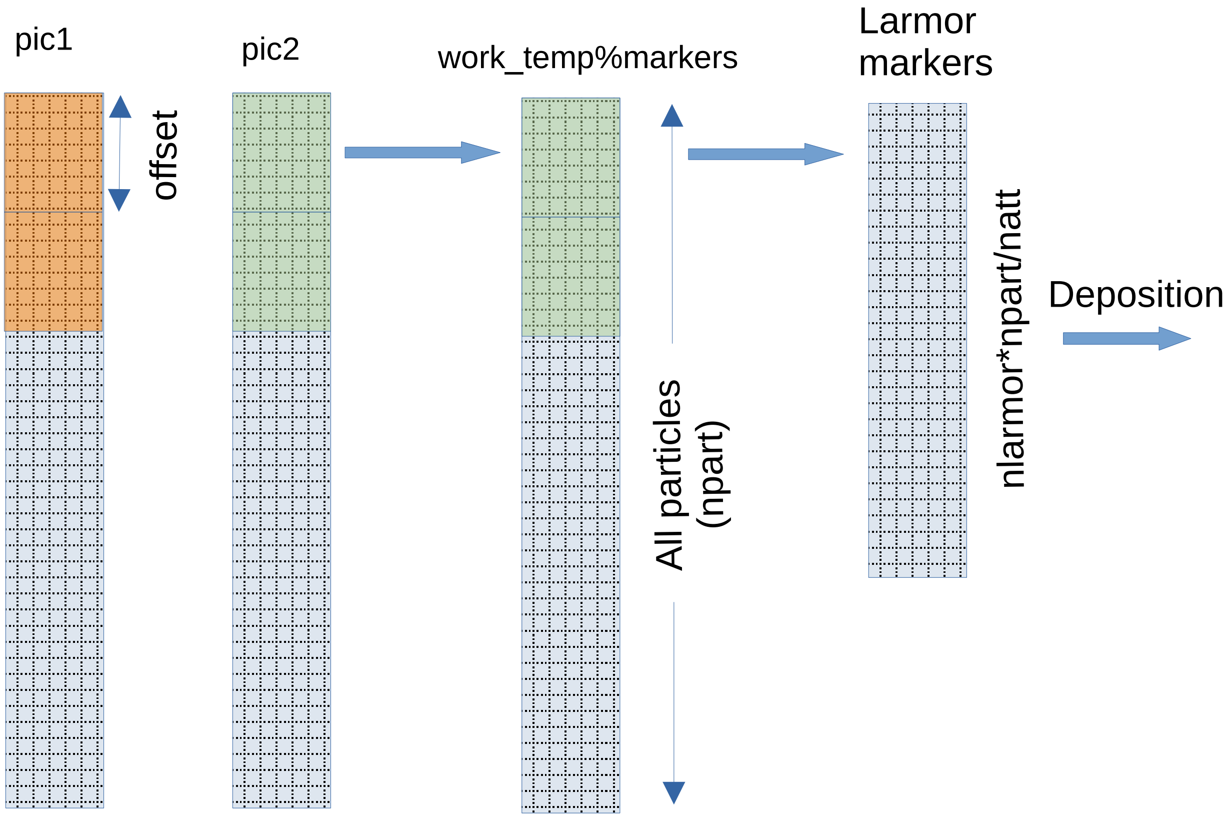




Reusing Reduced ATT arrays - Deposition

- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()
- call set_rho()




pic1

offset

pic2

work_temp%markers

All particles (npart)

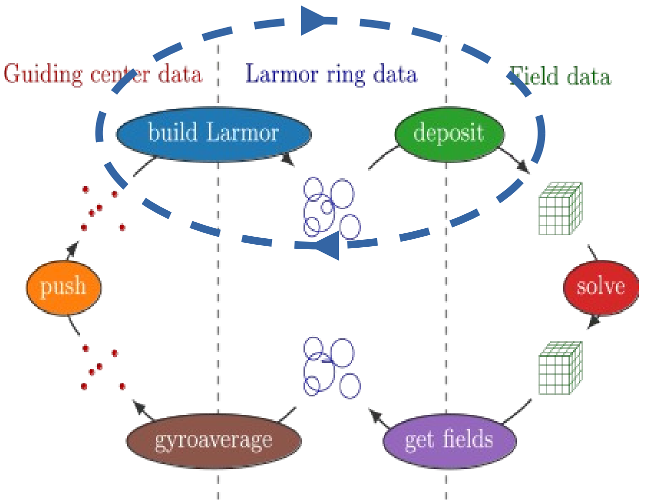
Larmor markers

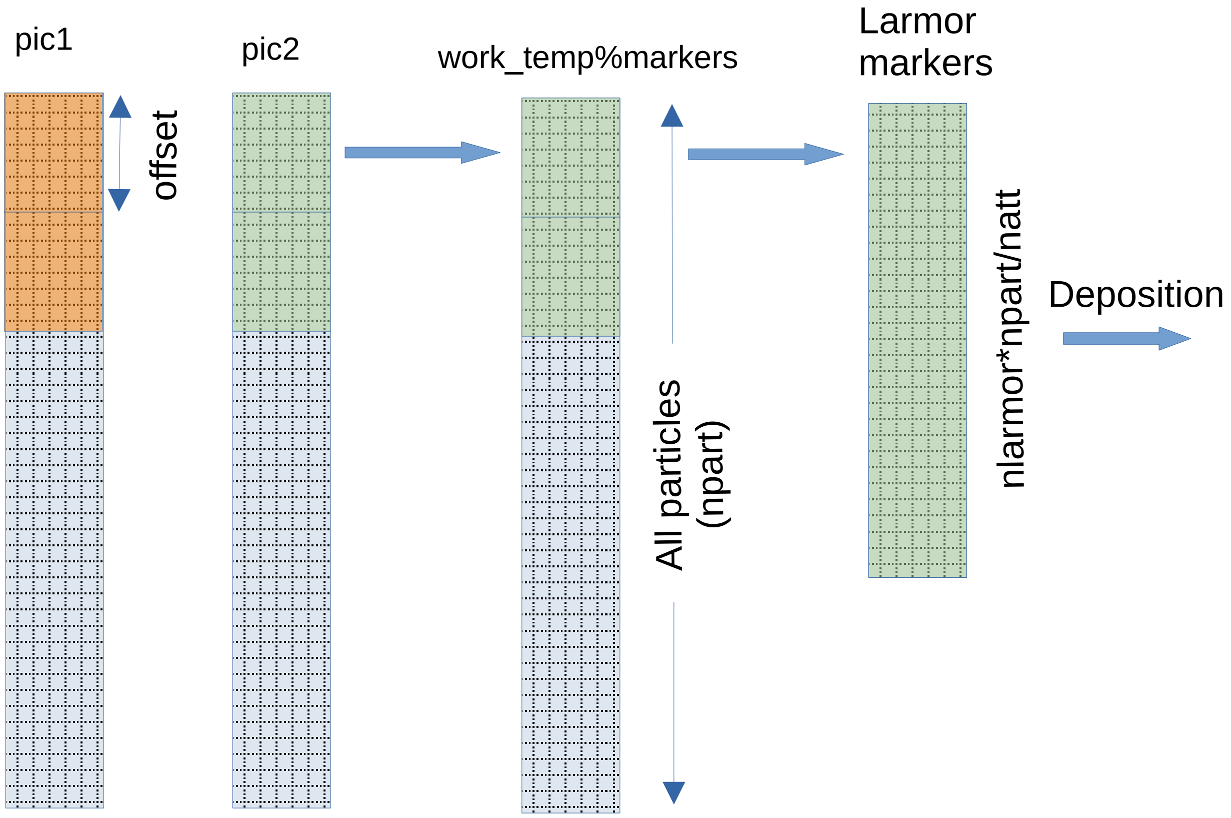
nlarmor*npart/natt

Deposition

Reusing Reduced ATT arrays - Deposition

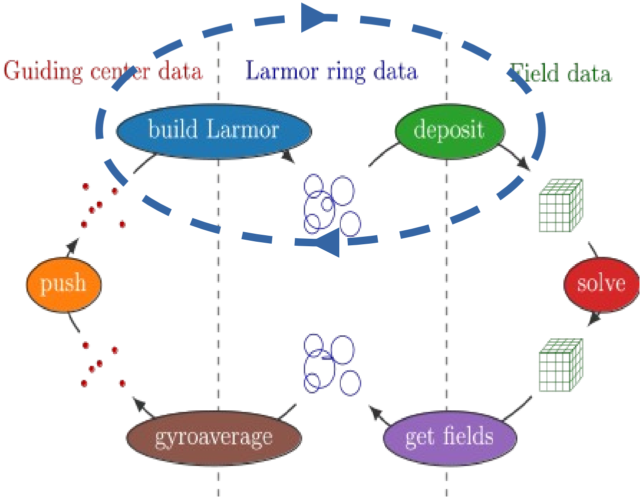
- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()
- call set_rho()

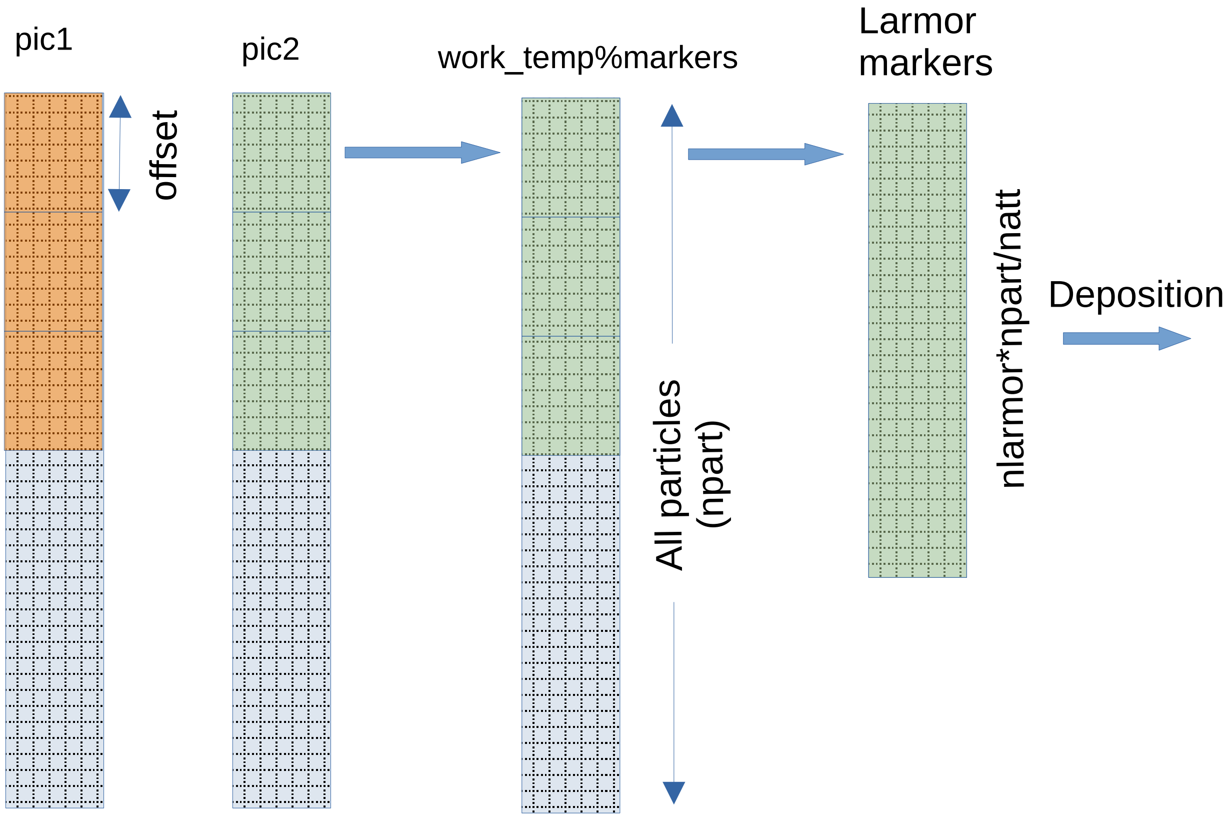




Reusing Reduced ATT arrays - Deposition

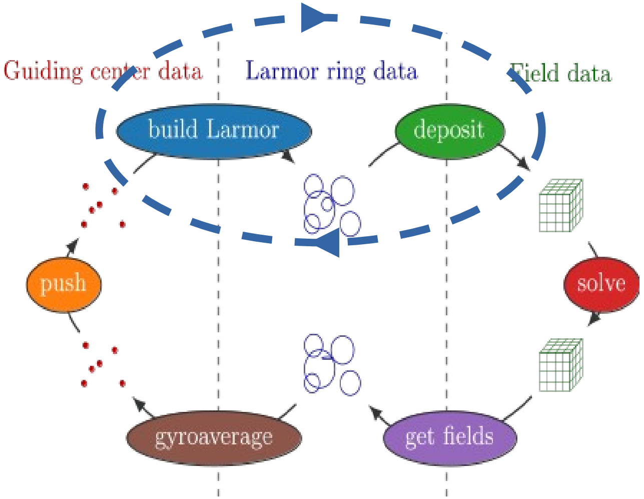
- call get_larmor_indices()
- call compute_all_larmor_positions()
    - call assign_larmor_atts()
- call set_rho()

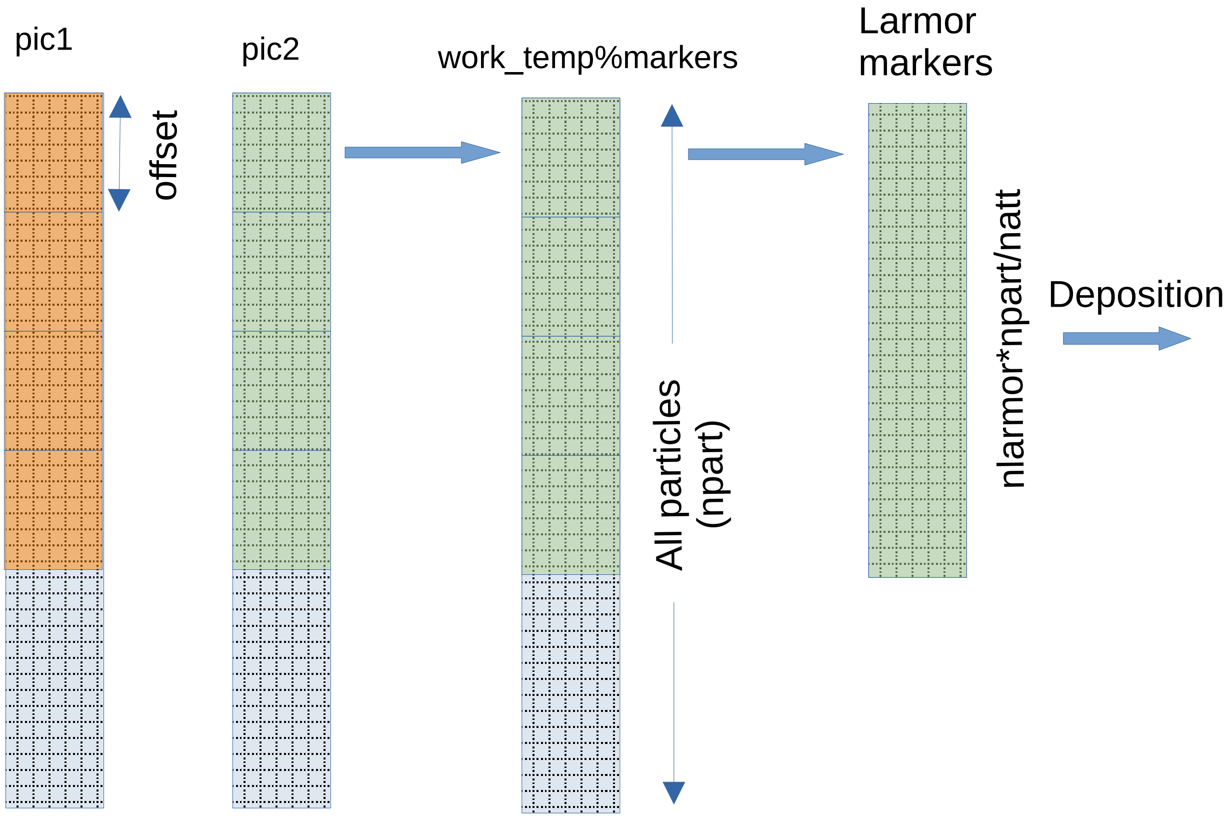




Reusing Reduced ATT arrays - Deposition

- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()
- call set_rho()




pic1

offset

pic2

work_temp%markers

All particles (npart)

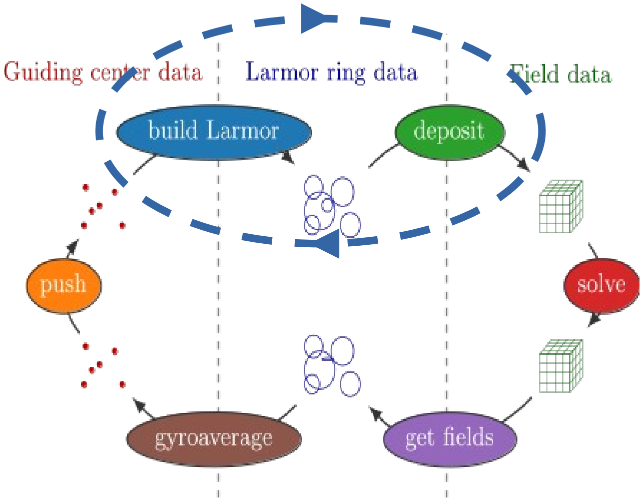
Larmor markers

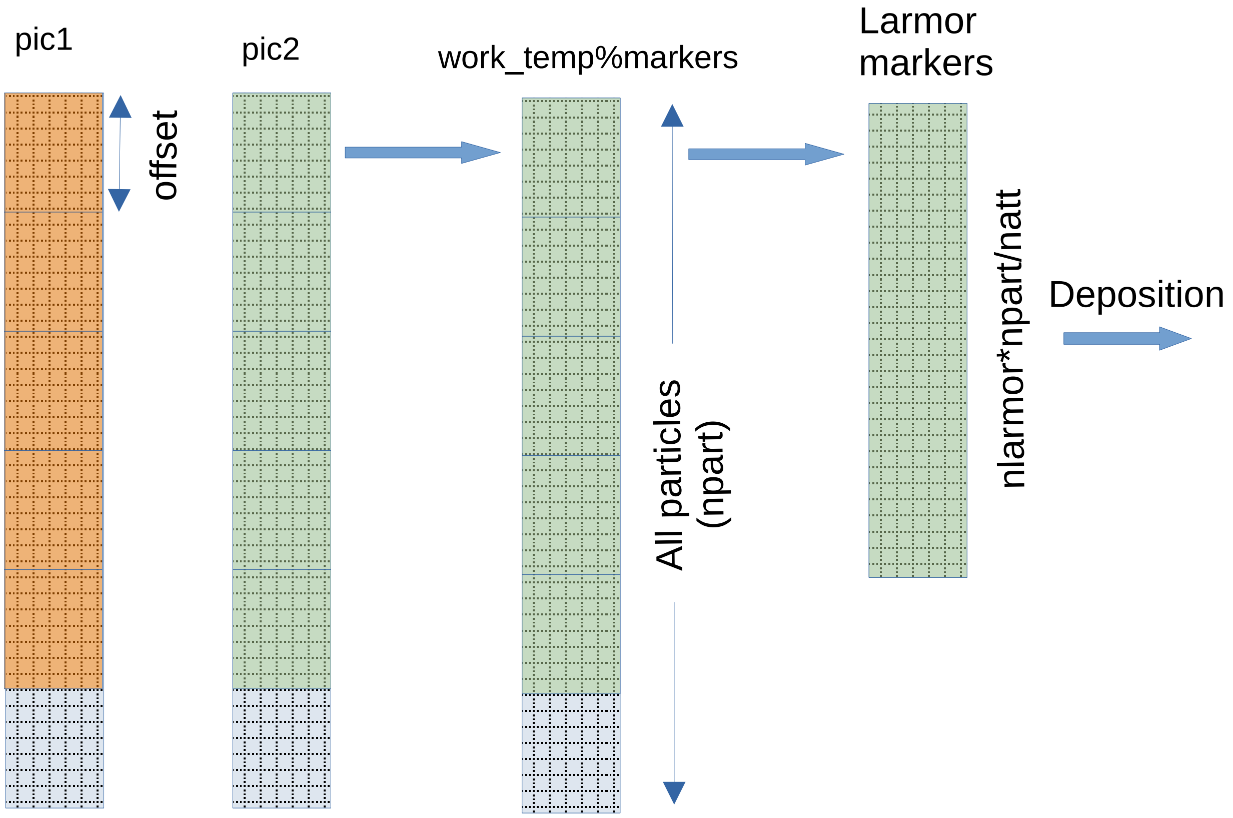
nlarmor*npart/natt

Deposition

Reusing Reduced ATT arrays - Deposition

- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()
- call set_rho()





pic1

offset

pic2

work_temp%markers

All particles (npart)

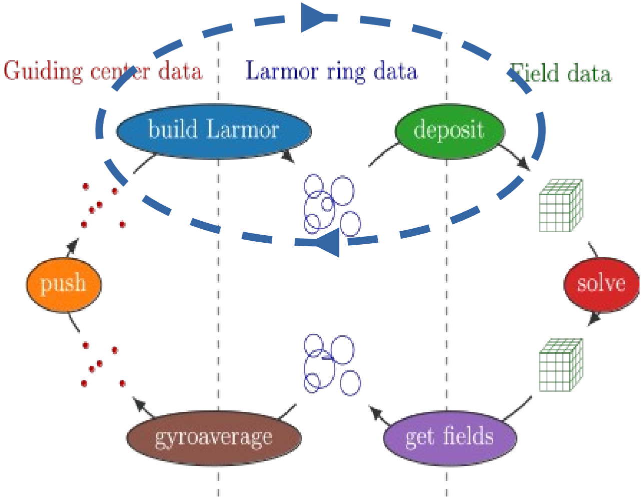
Larmor markers

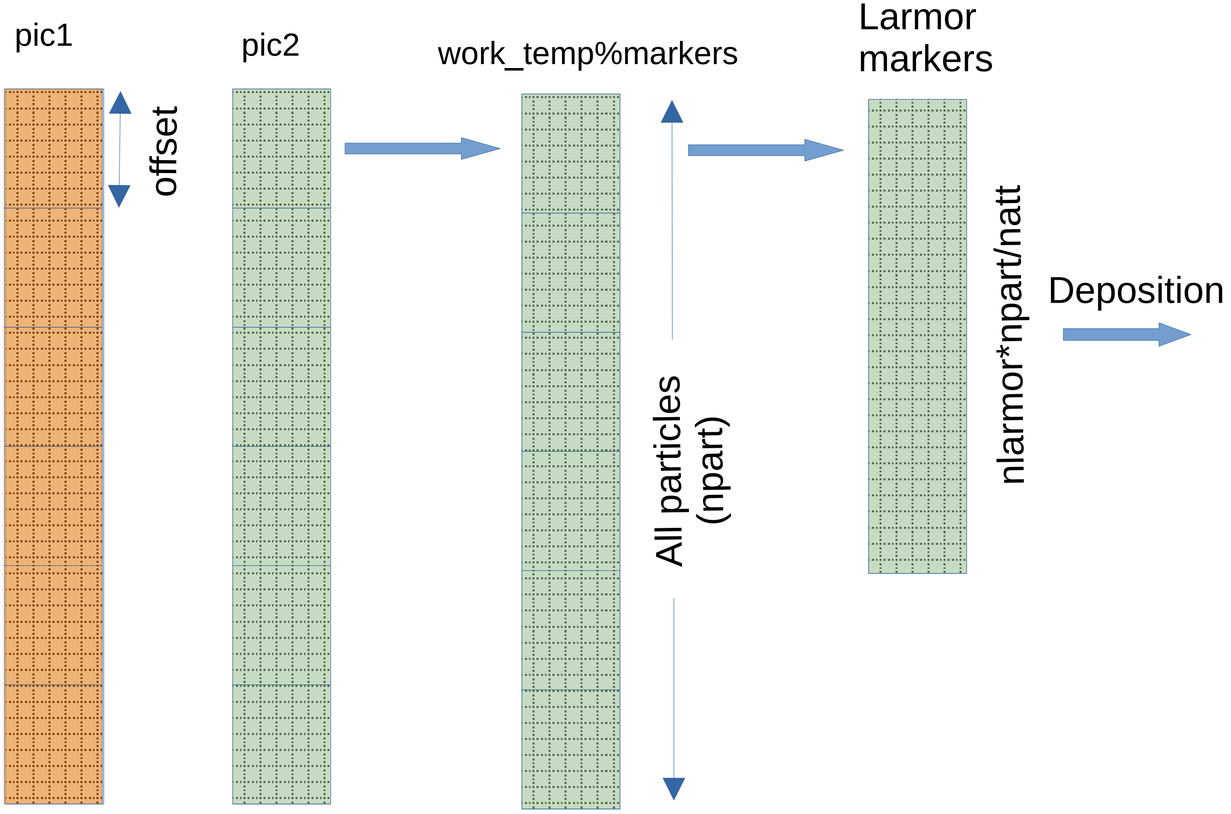
nlarmor*npart/natt

Deposition

Reusing Reduced ATT arrays - Deposition

- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()
- call set_rho()

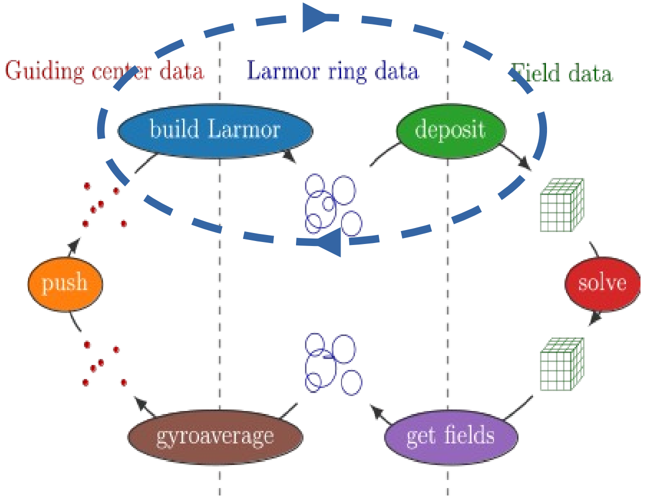




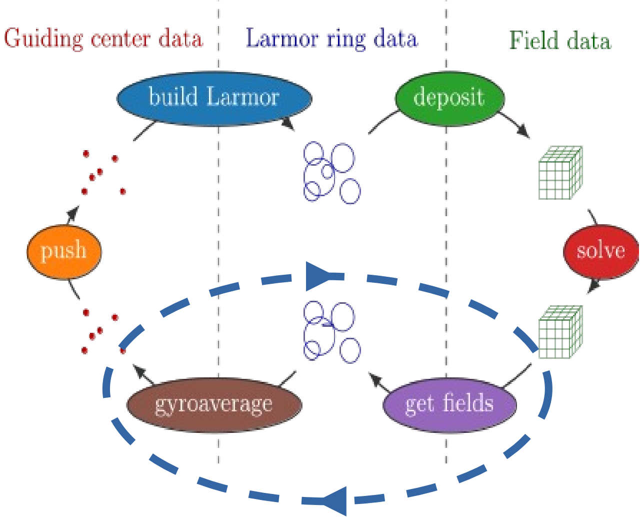
Code changes

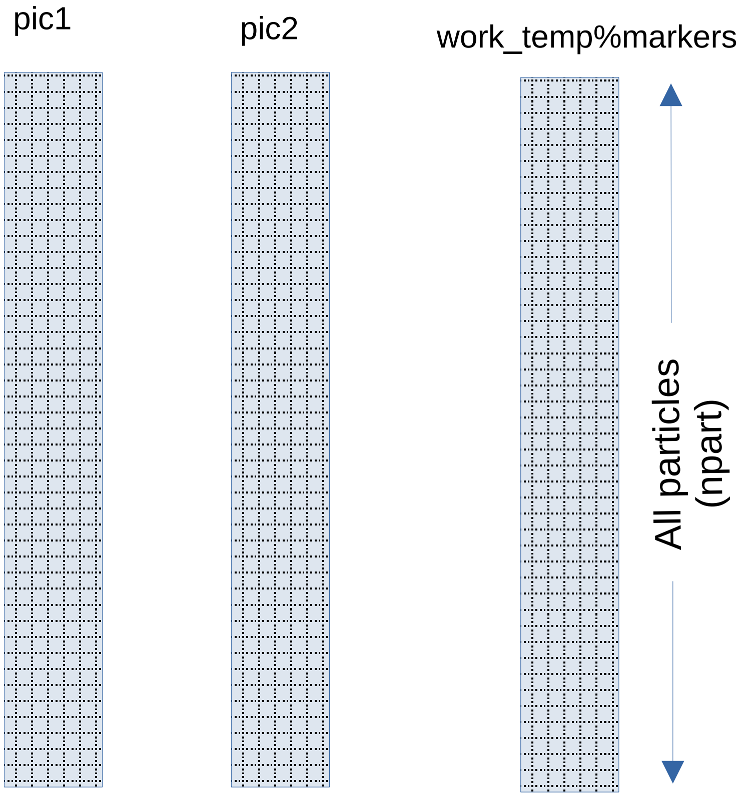
- one_step (deposition)
  - offset used to access correct element of pic arrays

```
if (.not.(fields%nl_neoclassical)) then
    ! Compute positions of gyropoints
    do i_atts = 1, basic%natts_iter
        call get_larmor_indices_disp()
        call compute_all_larmor_positions_disp()
        call setrho(i_atts)
        call advance_atts_offset_disp()
    end do
    call reset_atts_offset_disp()
end if
```



Reusing Reduced ATT arrays – Fields





pic1

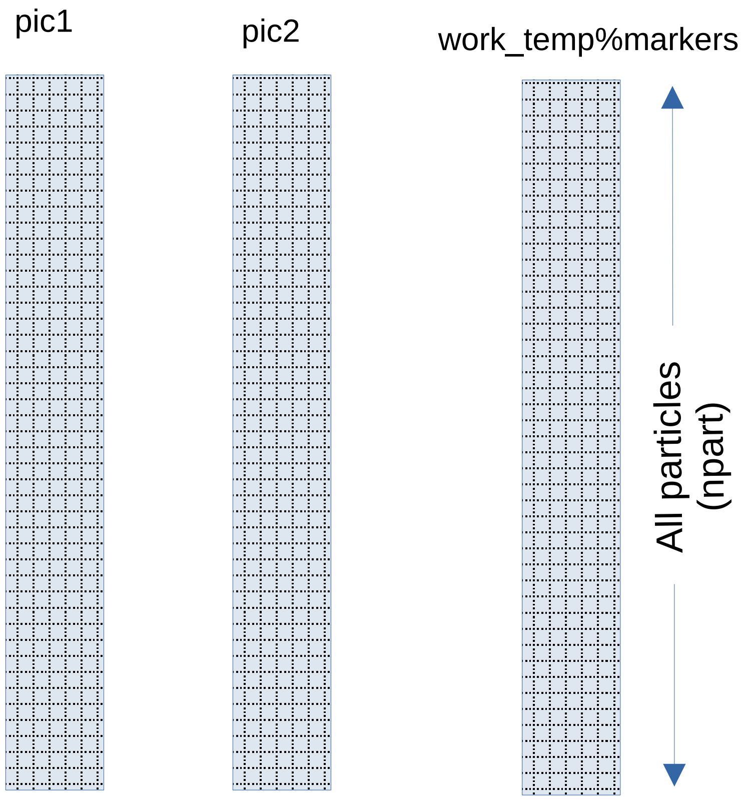
pic2

work_temp%markers

All particles
(npart)

Reusing Reduced ATT arrays – Fields

- call get_larmor_indices()

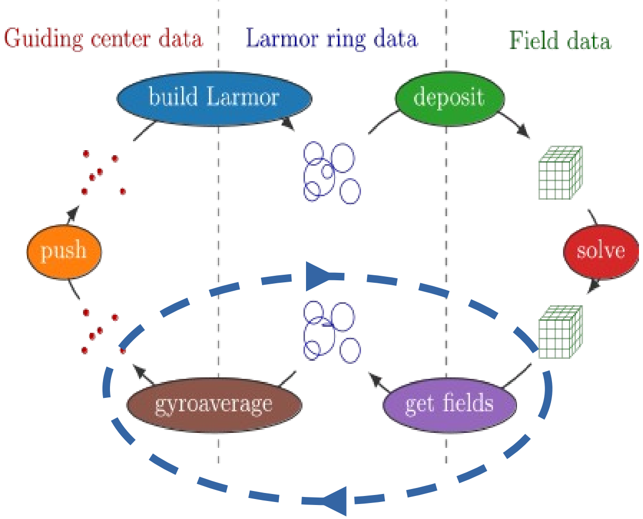


pic1

pic2

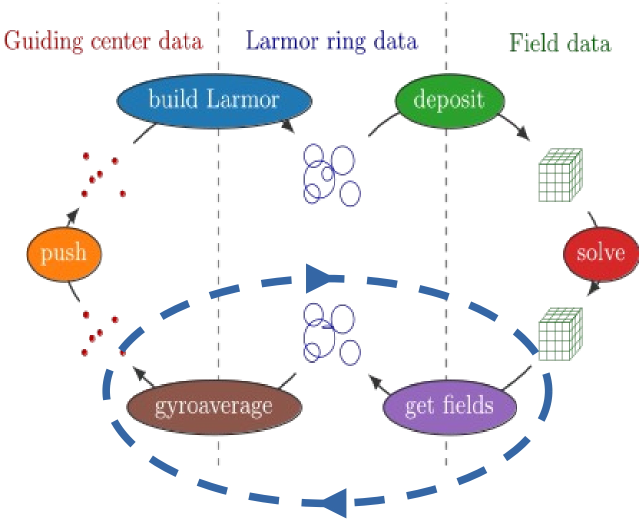
work_temp%markers

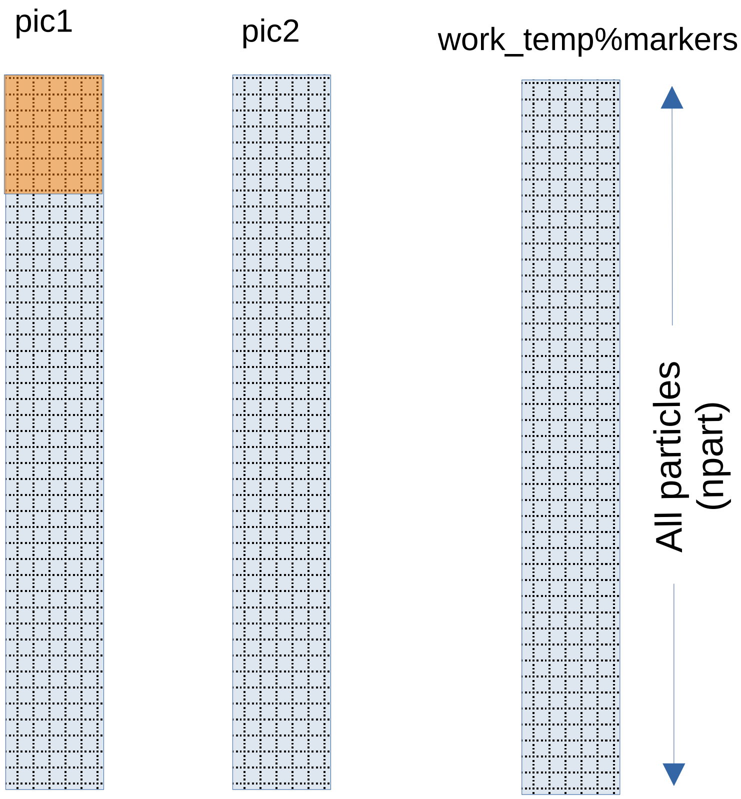
All particles (npart)



Reusing Reduced ATT arrays – Fields

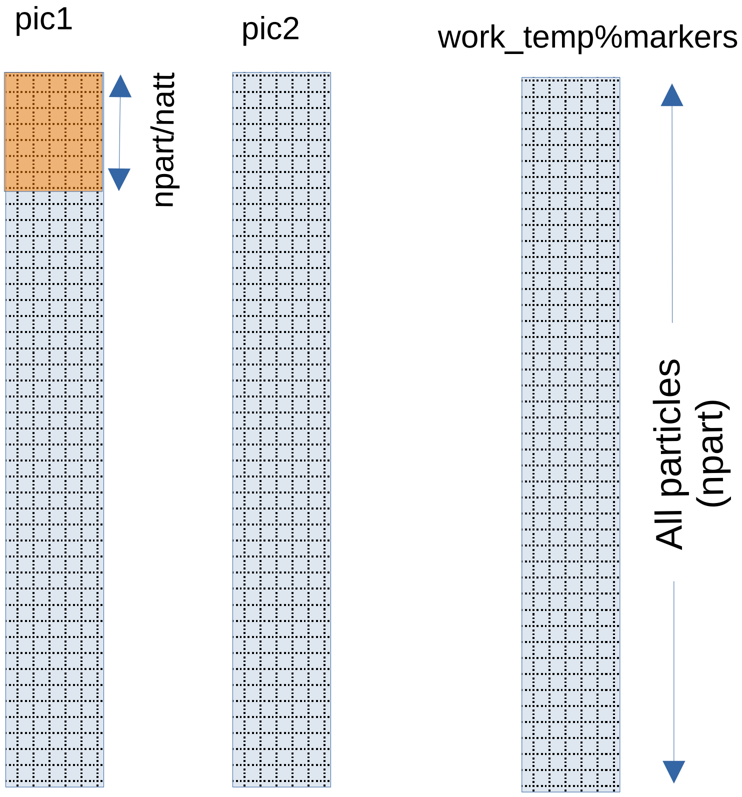
- call get_larmor_indices()





Reusing Reduced ATT arrays – Fields

- call get_larmor_indices()



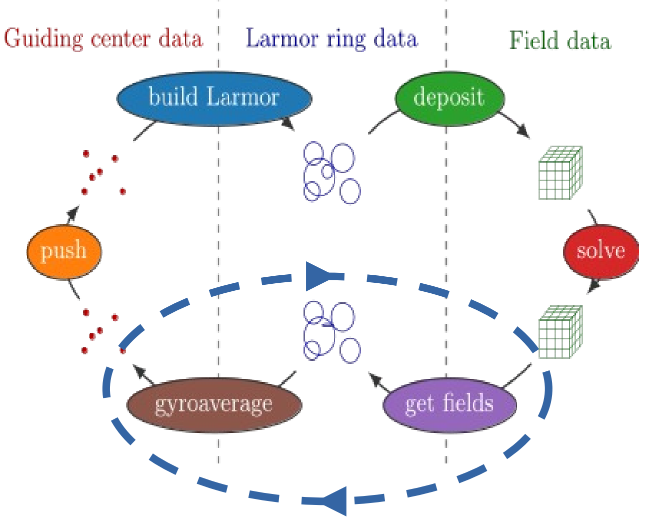
pic1

pic2

work_temp%markers

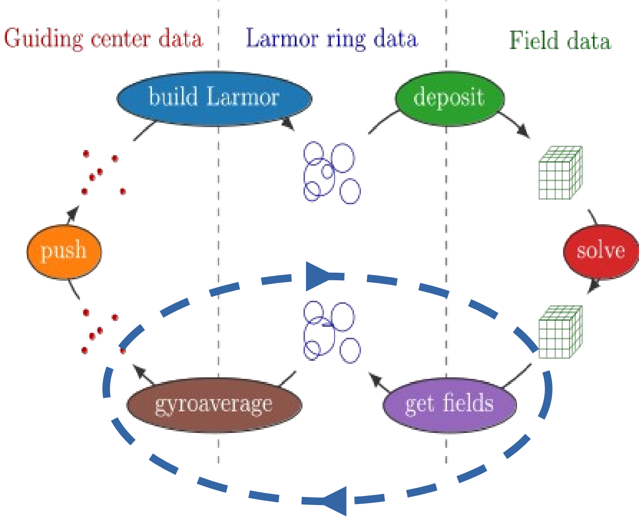
npart/natt

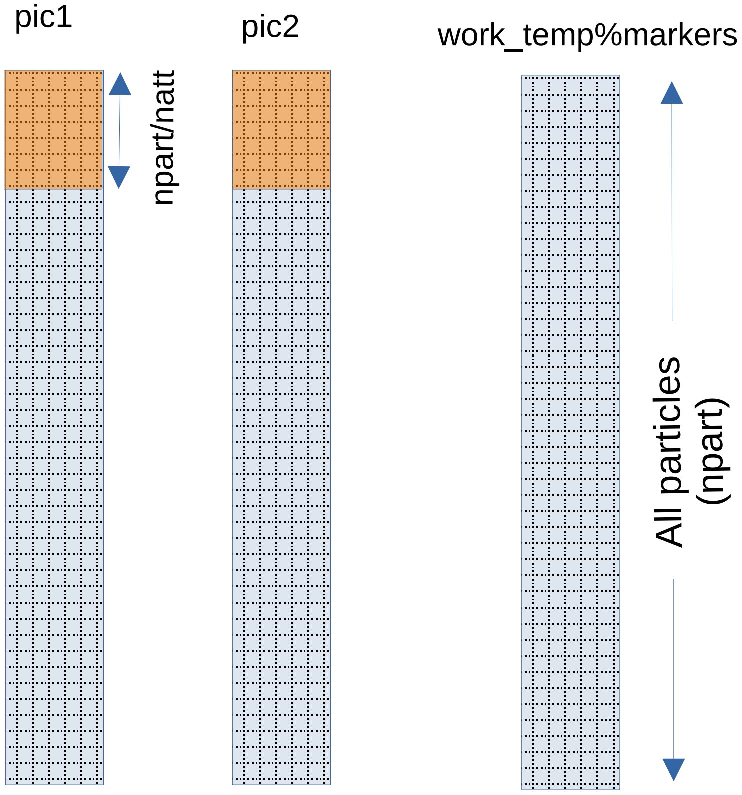
All particles
(npart)



Reusing Reduced ATT arrays – Fields

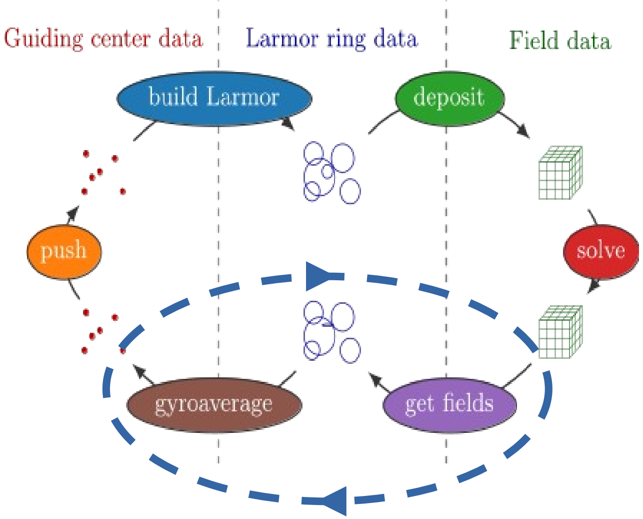
- call get_larmor_indices()

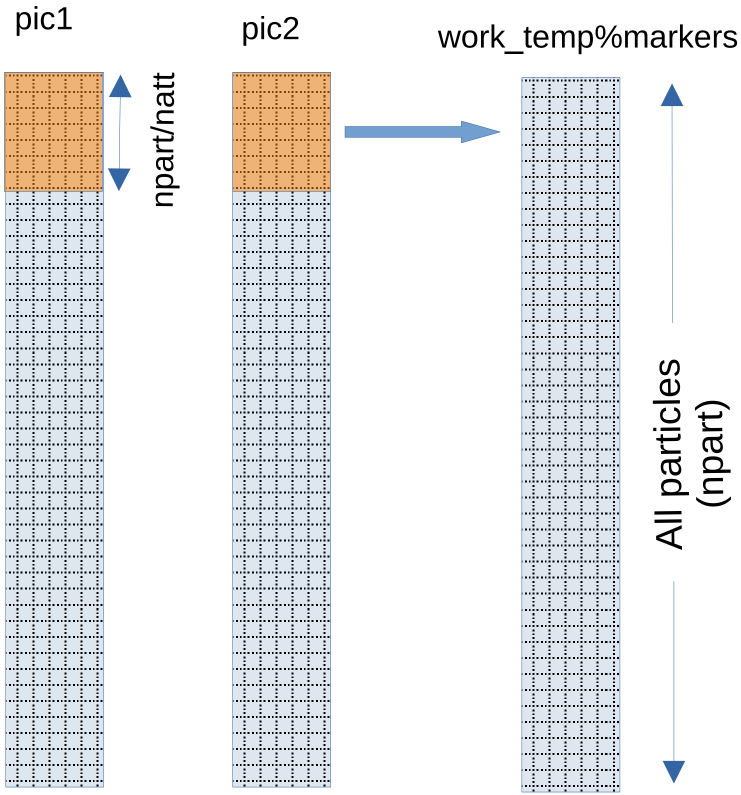




Reusing Reduced ATT arrays – Fields

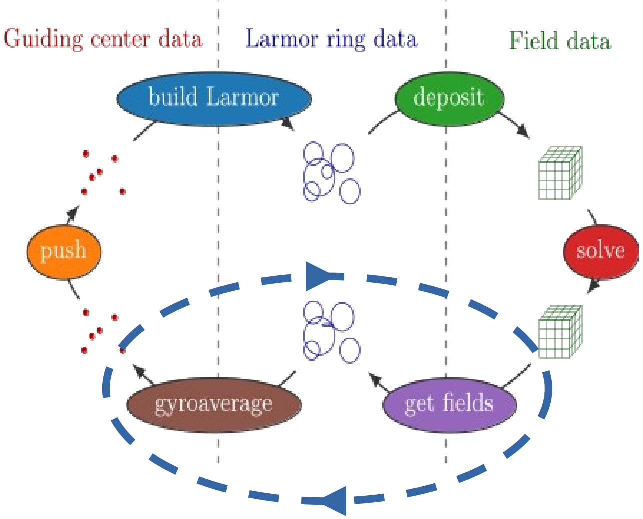
- call get_larmor_indices()

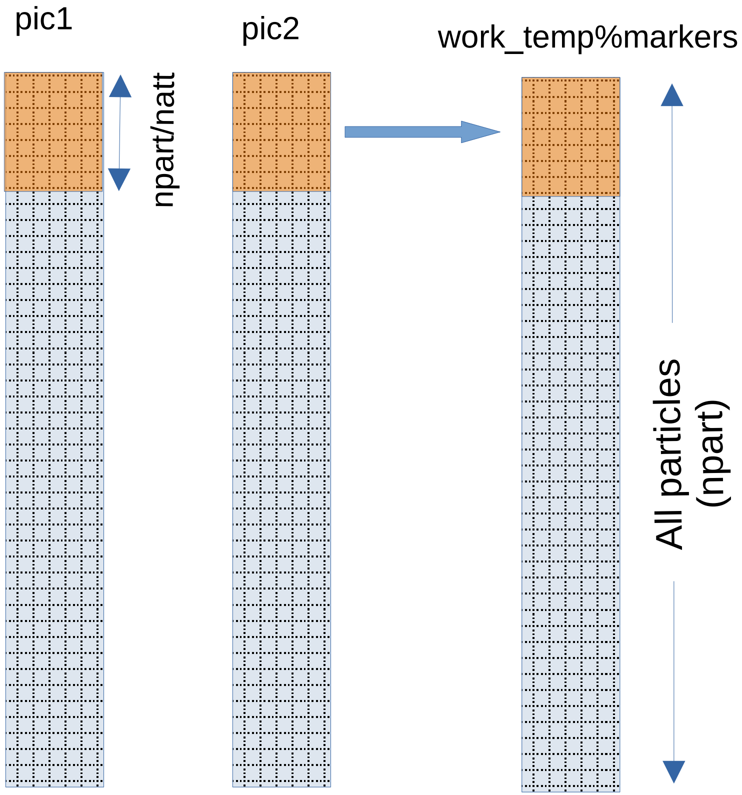




Reusing Reduced ATT arrays – Fields

- call get_larmor_indices()




pic1

pic2

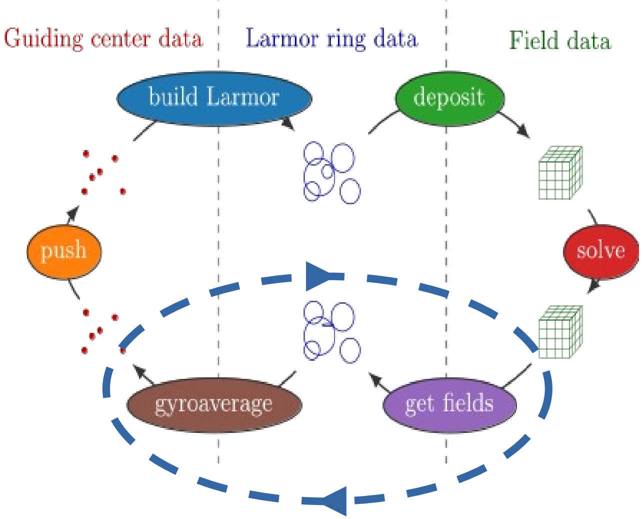
work_temp%markers

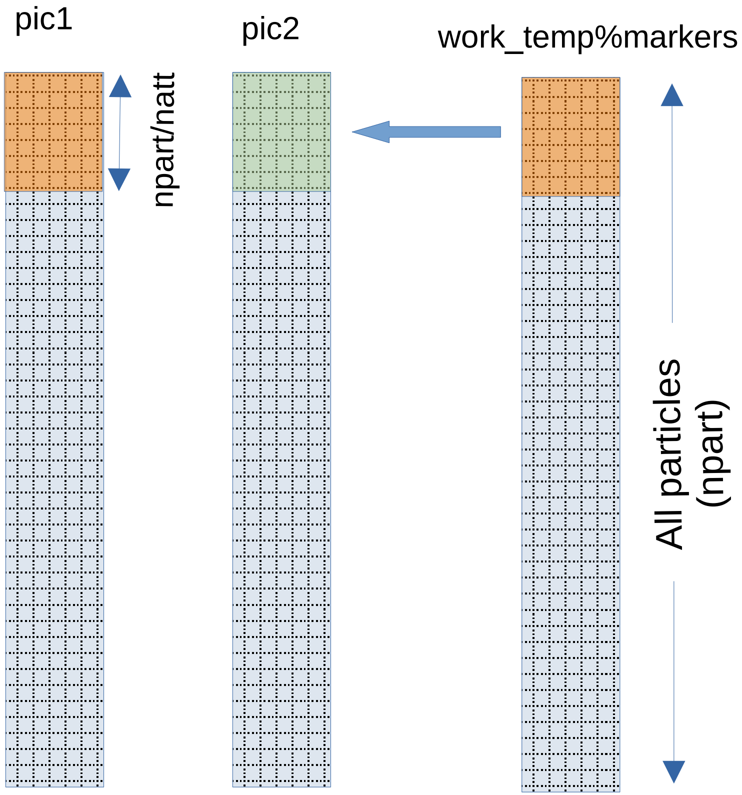
npart/natt

All particles (npart)

Reusing Reduced ATT arrays – Fields

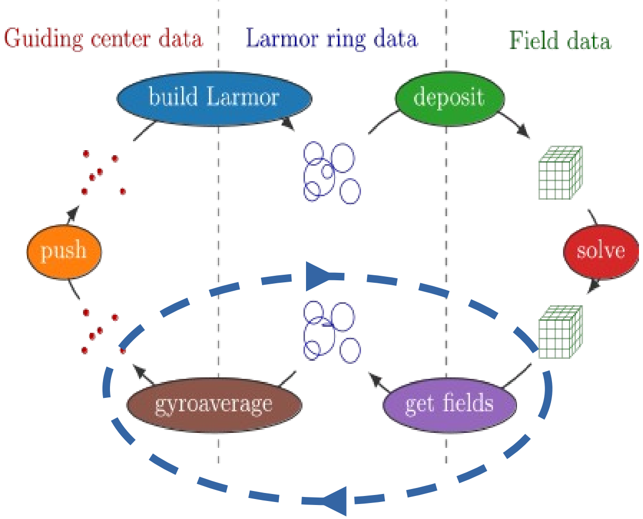
- call get_larmor_indices()

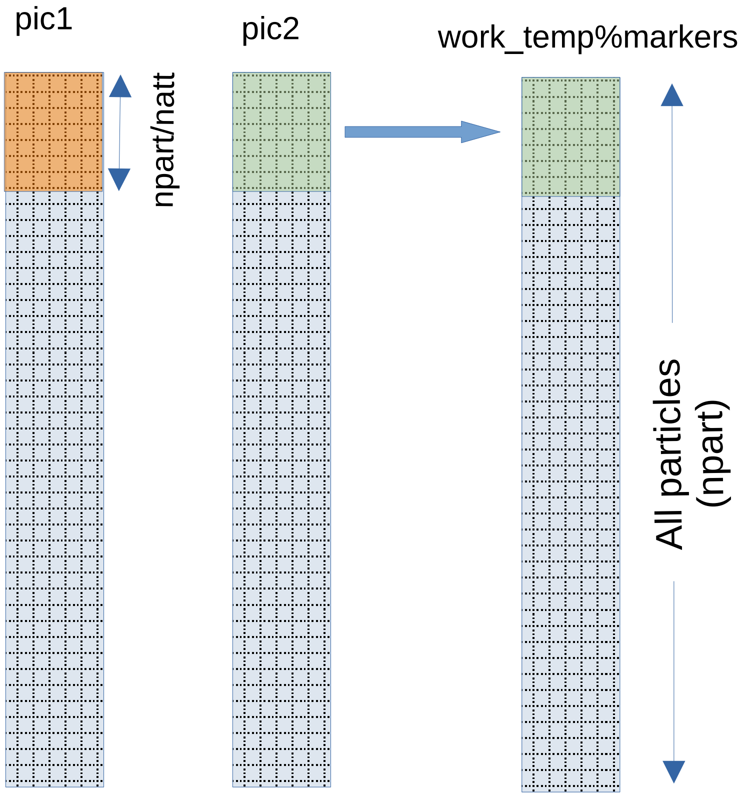




Reusing Reduced ATT arrays – Fields

- call get_larmor_indices()





pic1

npart/natt

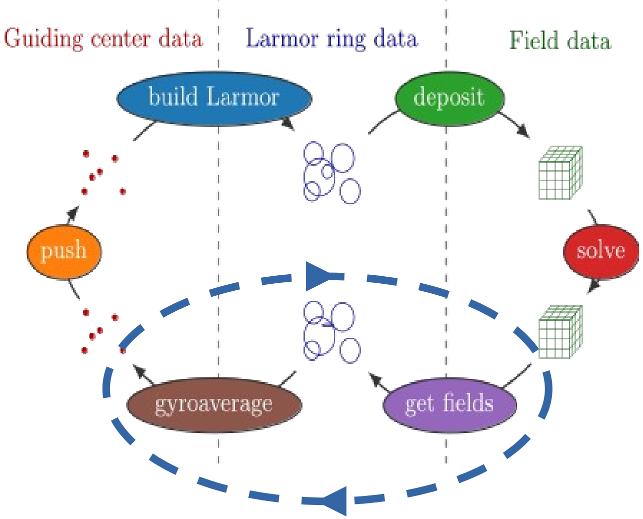
pic2

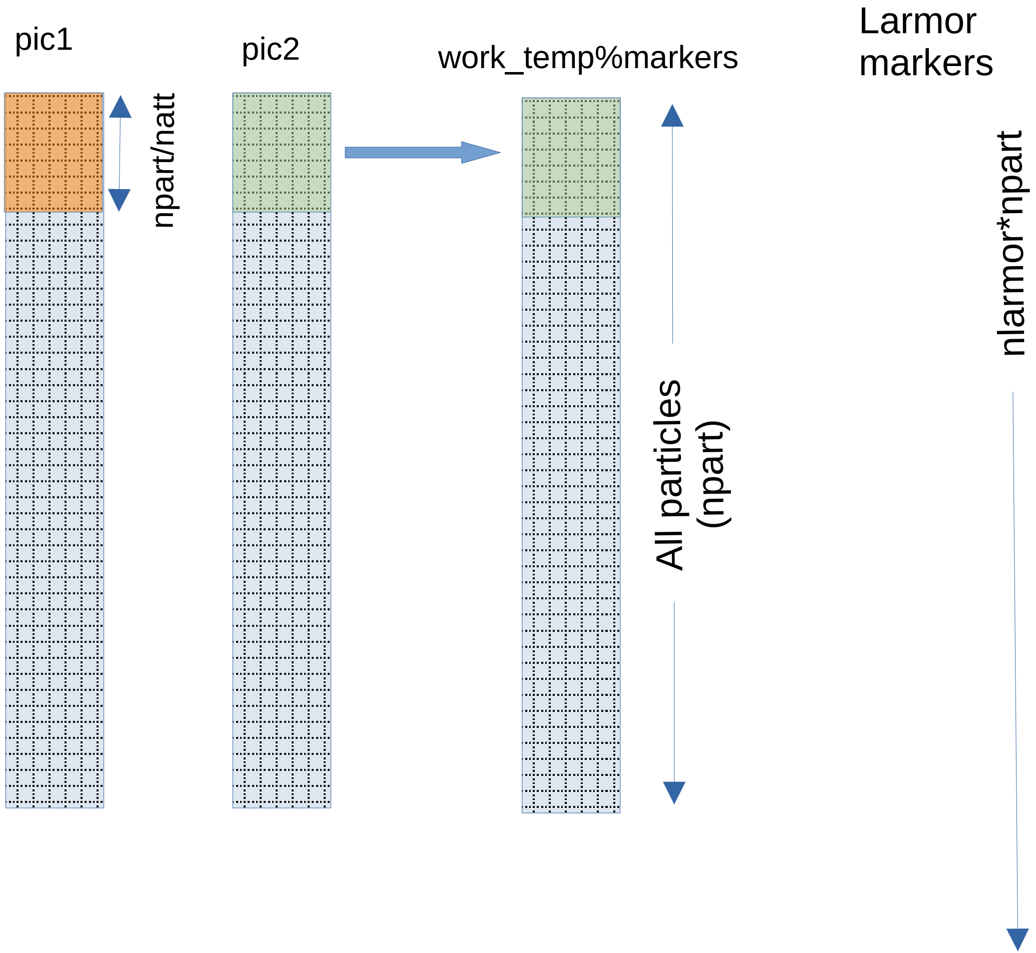
work_temp%markers

All particles (npart)

Reusing Reduced ATT arrays – Fields

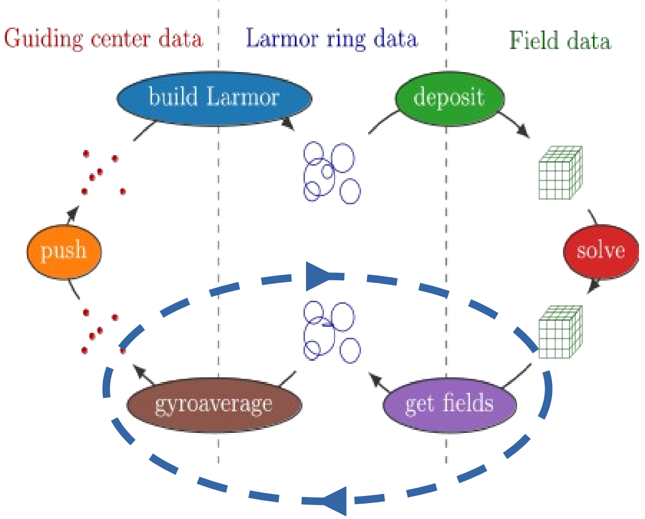
- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()
- call get_field()

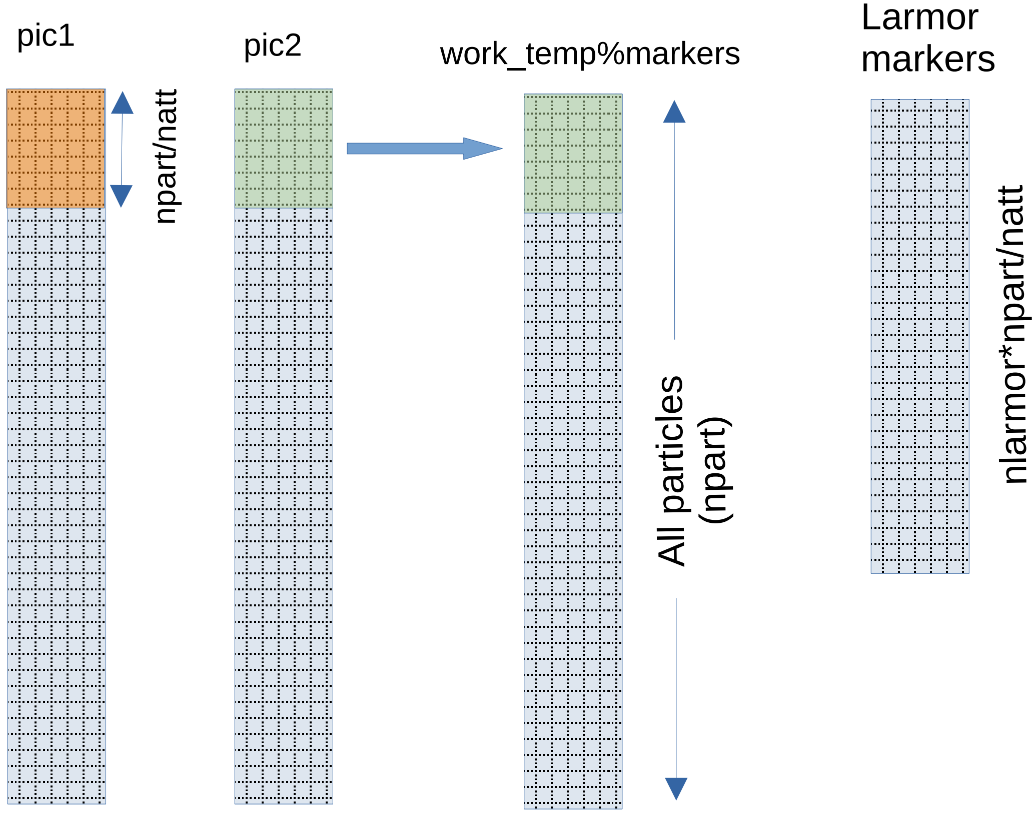




Reusing Reduced ATT arrays – Fields

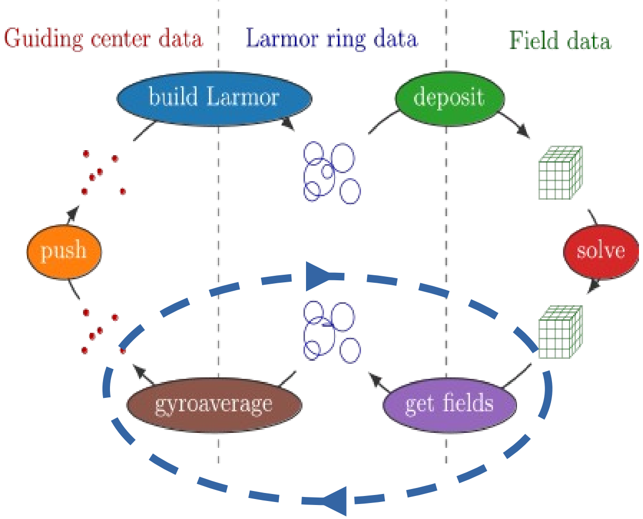
- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()
- call get_field()

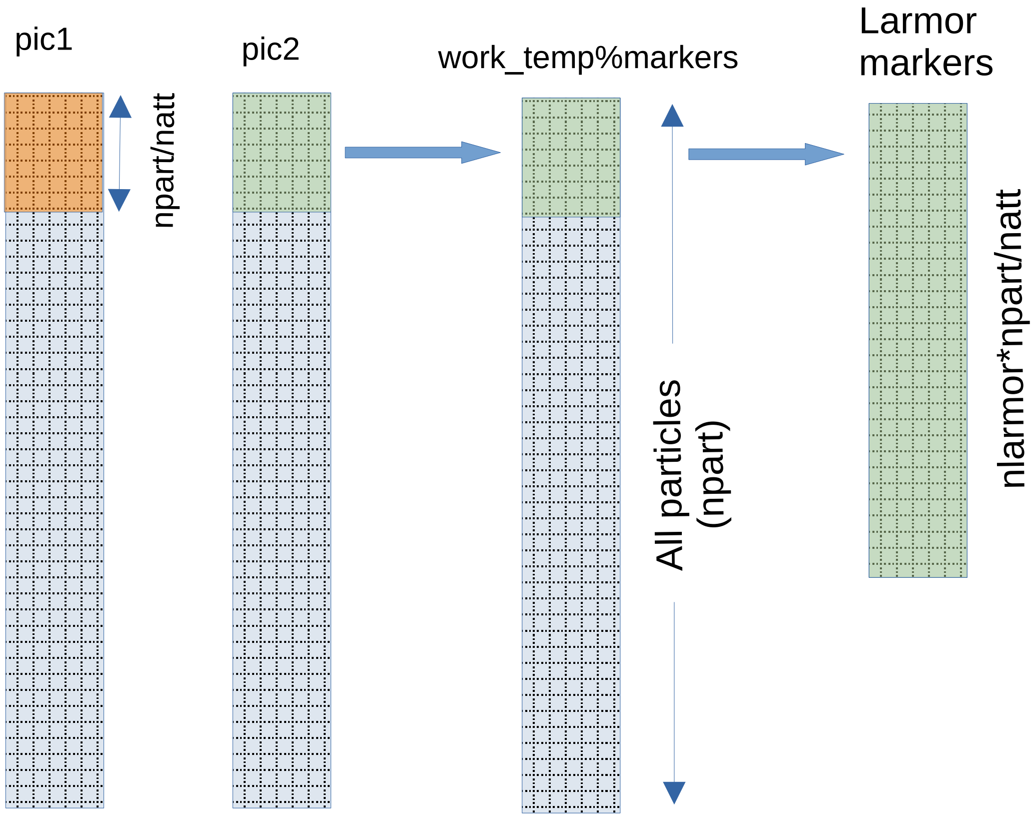




Reusing Reduced ATT arrays – Fields

- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()
- call get_field()




pic1

npart/natt

pic2

work_temp%markers

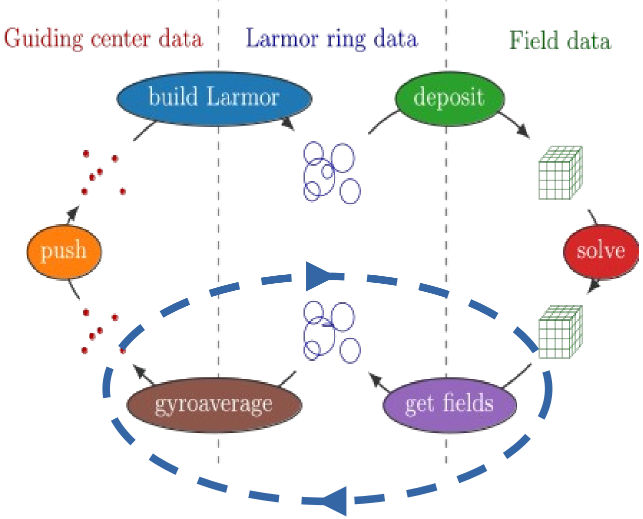
All particles (npart)

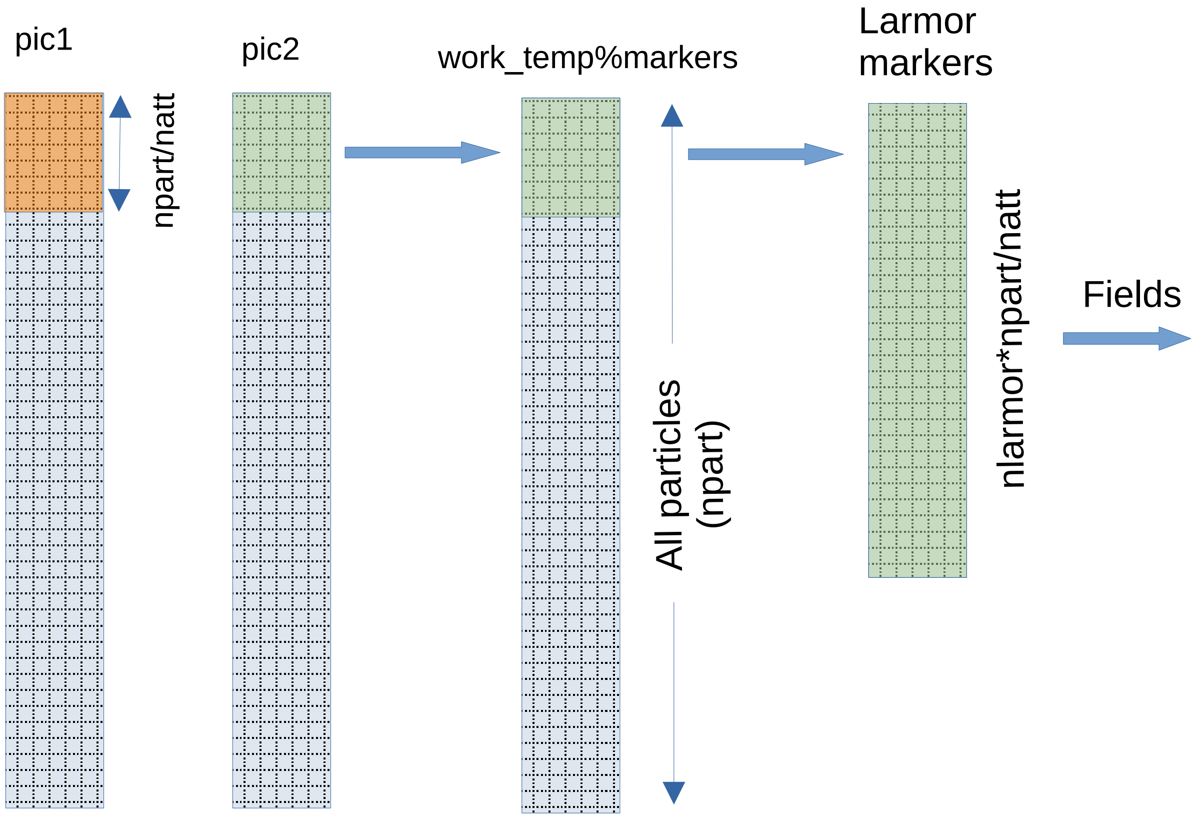
Larmor markers

nlarmor*npart/natt

Reusing Reduced ATT arrays – Fields

- call get_larmor_indices()
- call compute_all_larmor_positions()
    - call assign_larmor_atts()
- call get_field()




pic1

npart/natt

pic2

work_temp%markers

All particles (npart)

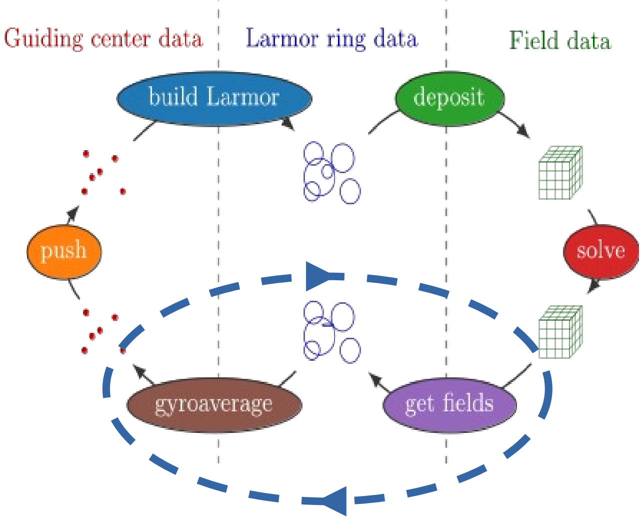
Larmor markers

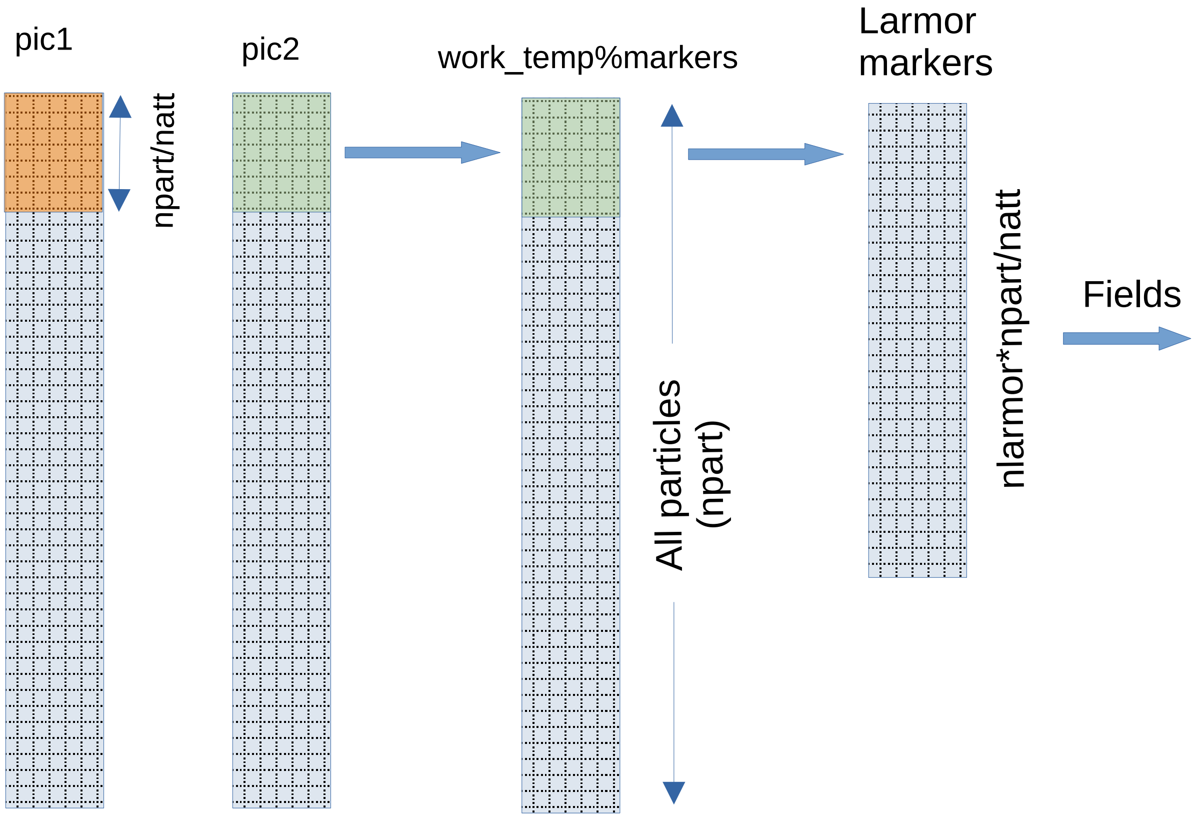
nlarmor*npart/natt

Fields

Reusing Reduced ATT arrays – Fields

- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()
- call get_field()




pic1

npart/natt

pic2

work_temp%markers

All particles (npart)

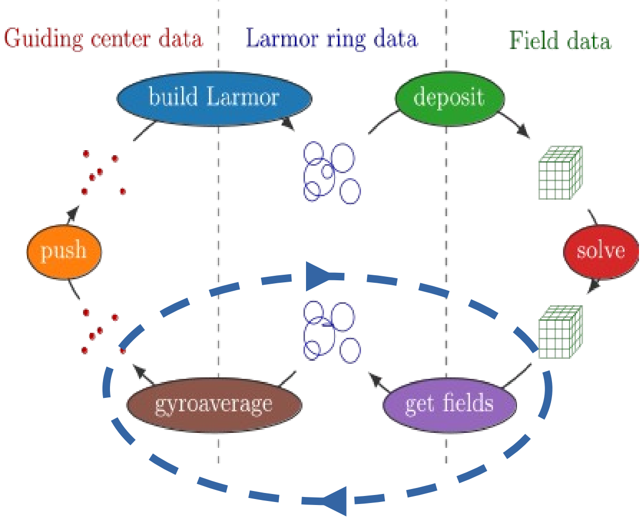
Larmor markers

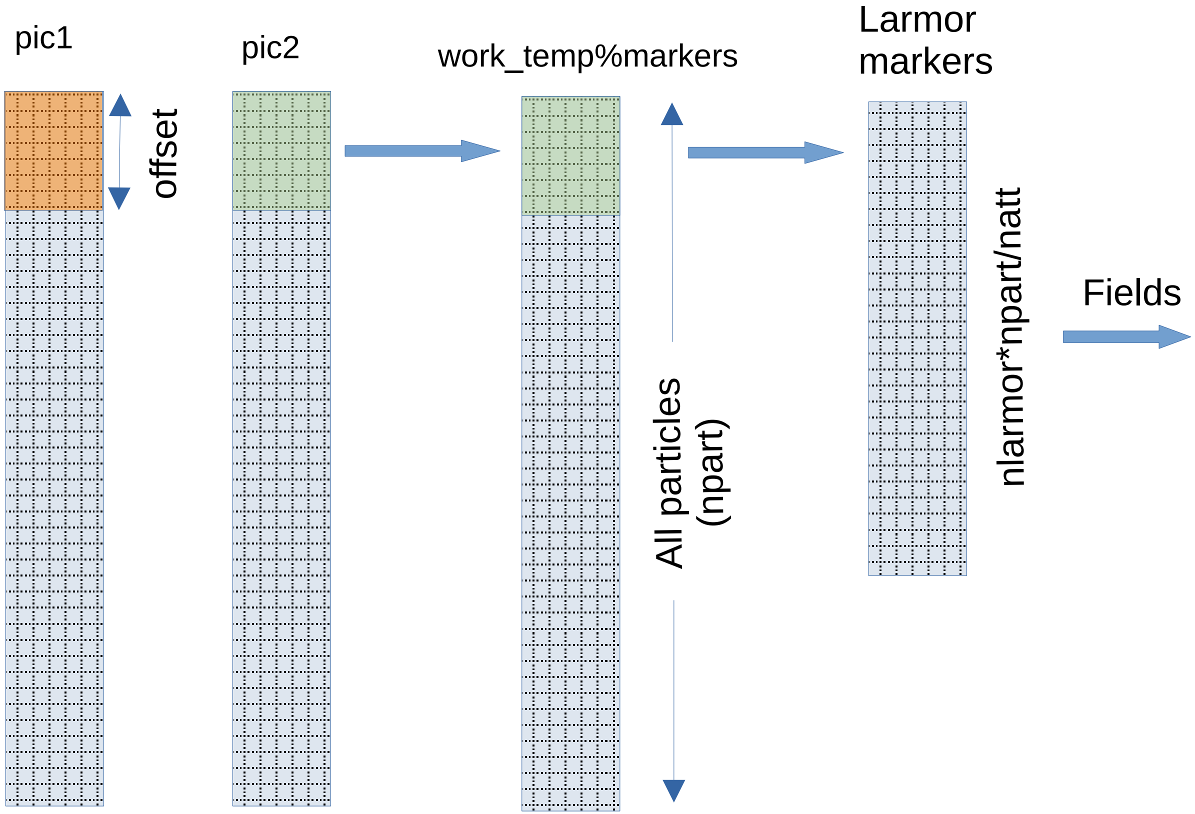
nlarmor*npart/natt

Fields

Reusing Reduced ATT arrays – Fields

- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()
- call get_field()





pic1

offset

pic2

work_temp%markers

All particles (npart)

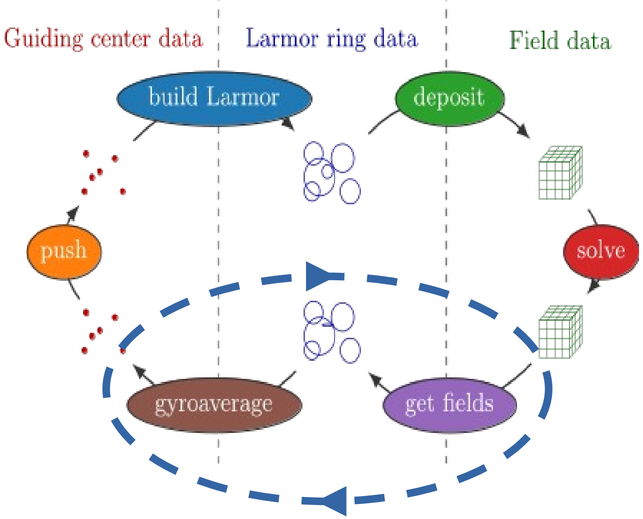
Larmor markers

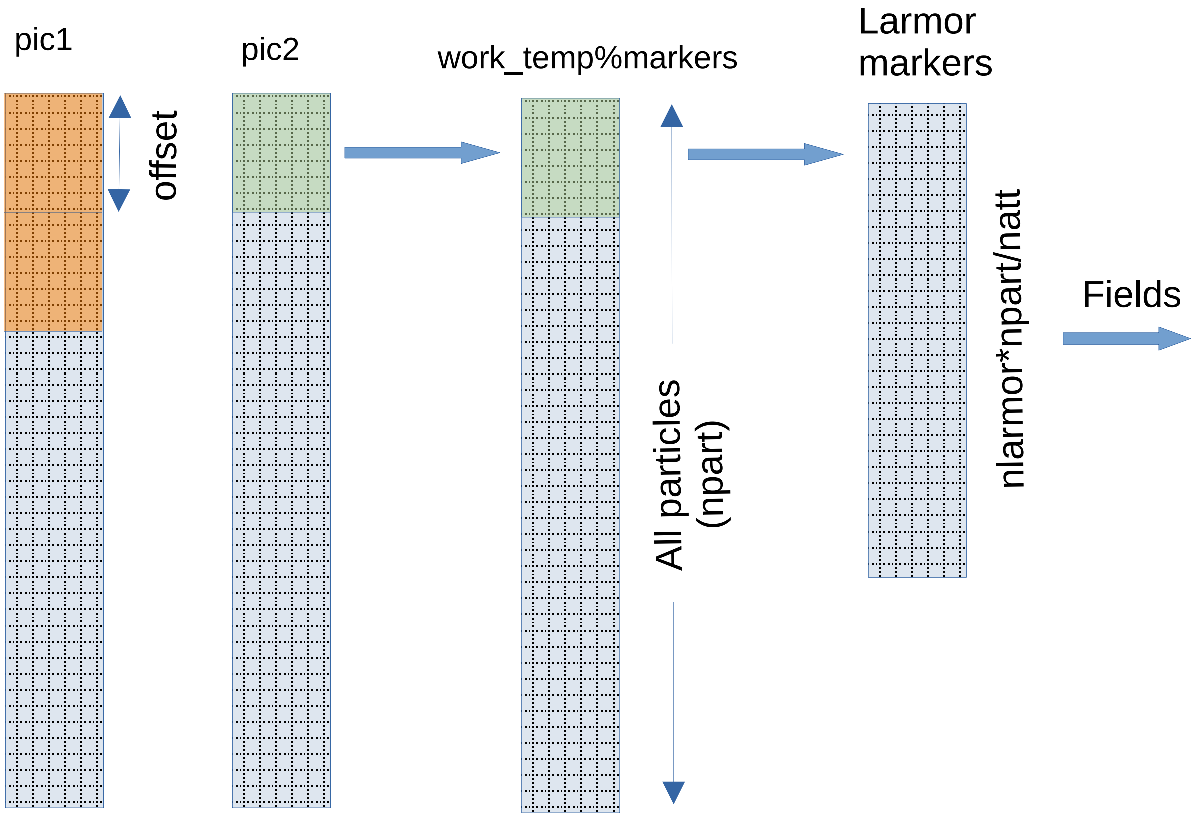
nlarmor*npart/natt

Fields

Reusing Reduced ATT arrays – Fields

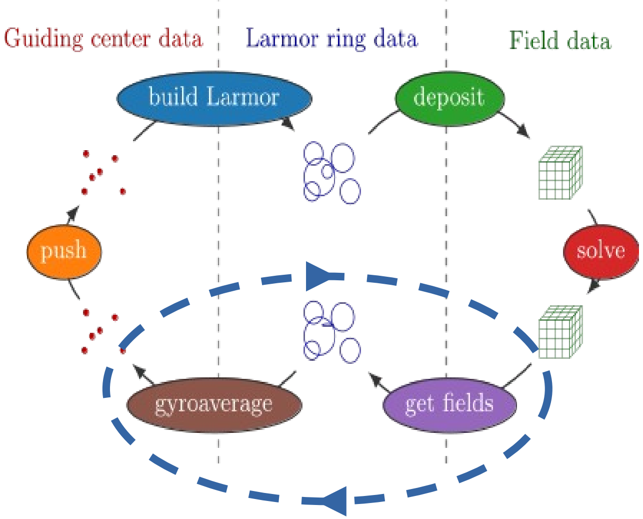
- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()
- call get_field()

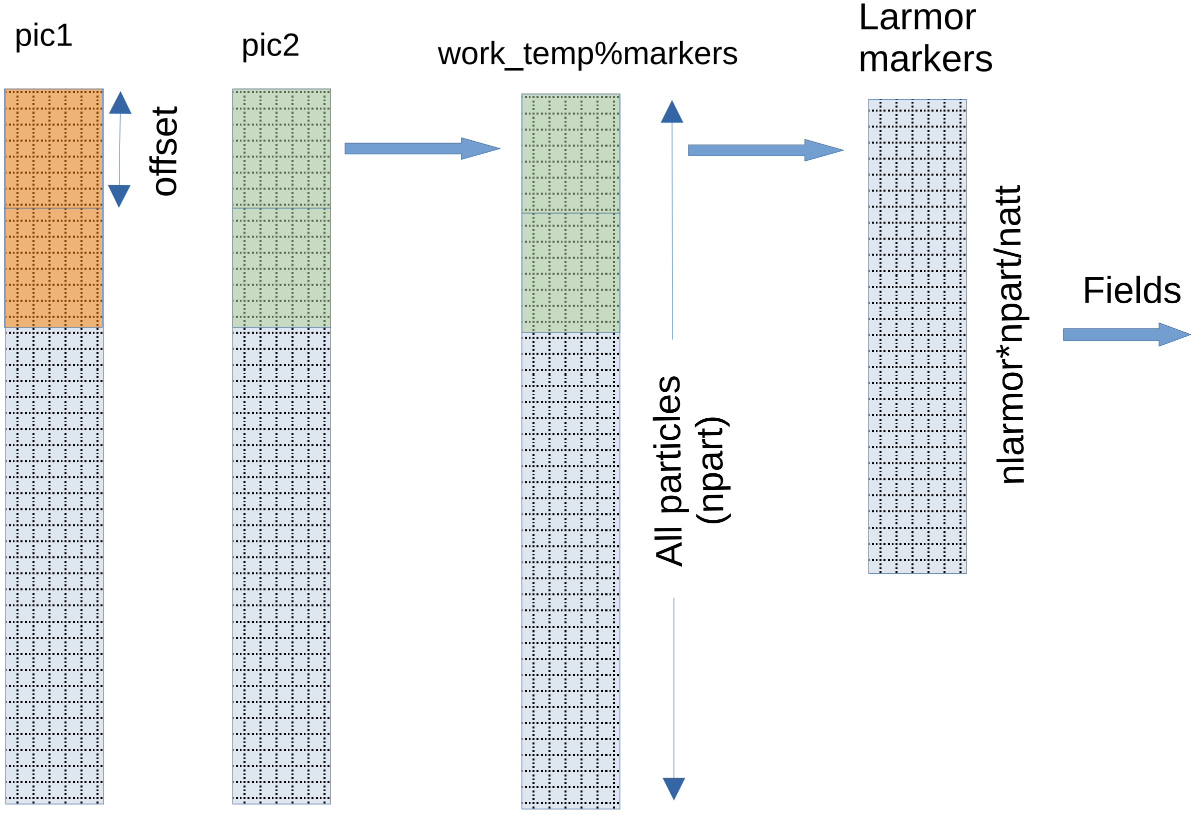




Reusing Reduced ATT arrays – Fields

- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()
- call get_field()





pic1

offset

pic2

work_temp%markers

All particles (npart)

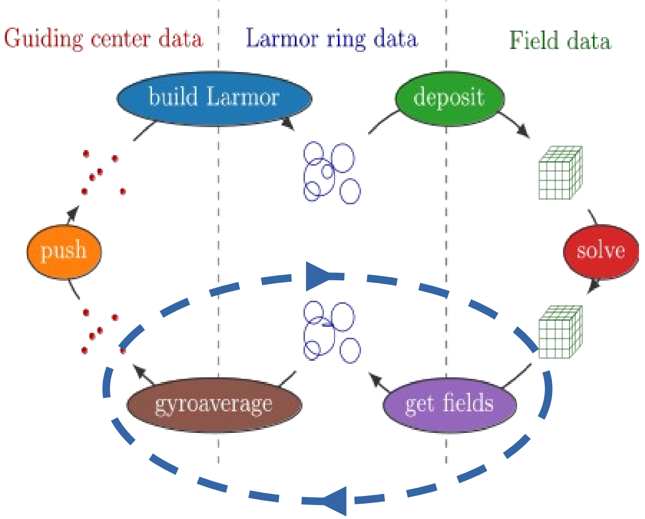
Larmor markers

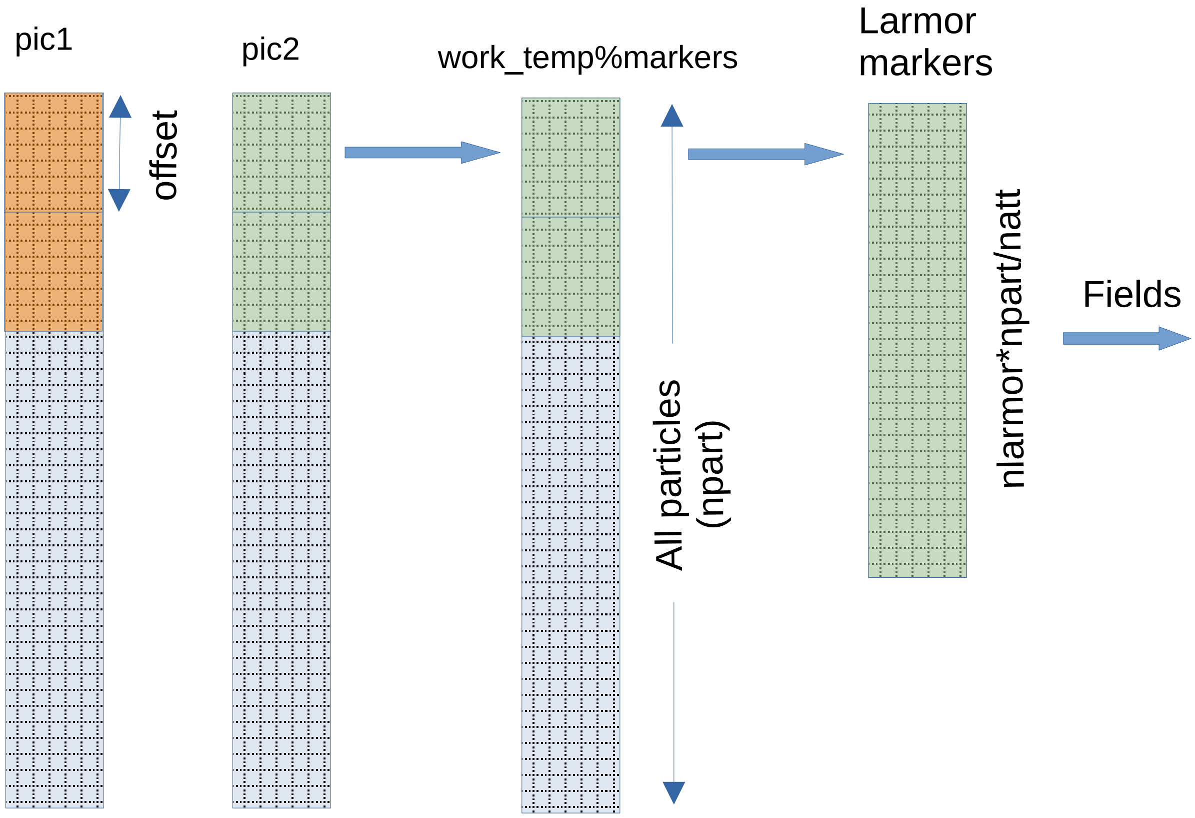
nlarmor*npart/natt

Fields

Reusing Reduced ATT arrays – Fields

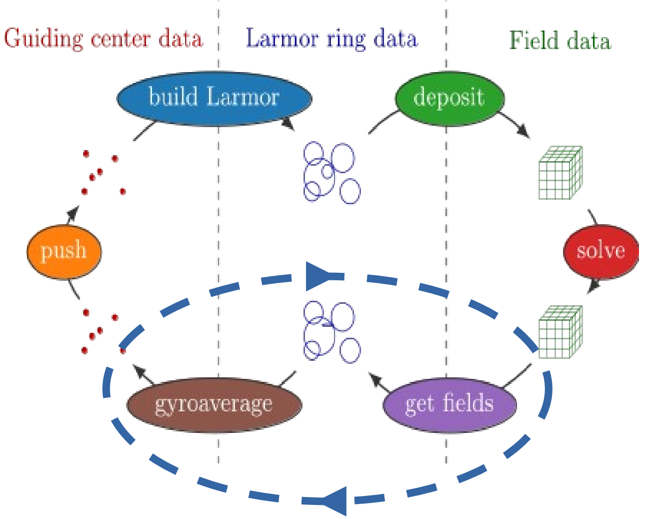
- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()
- call get_field()

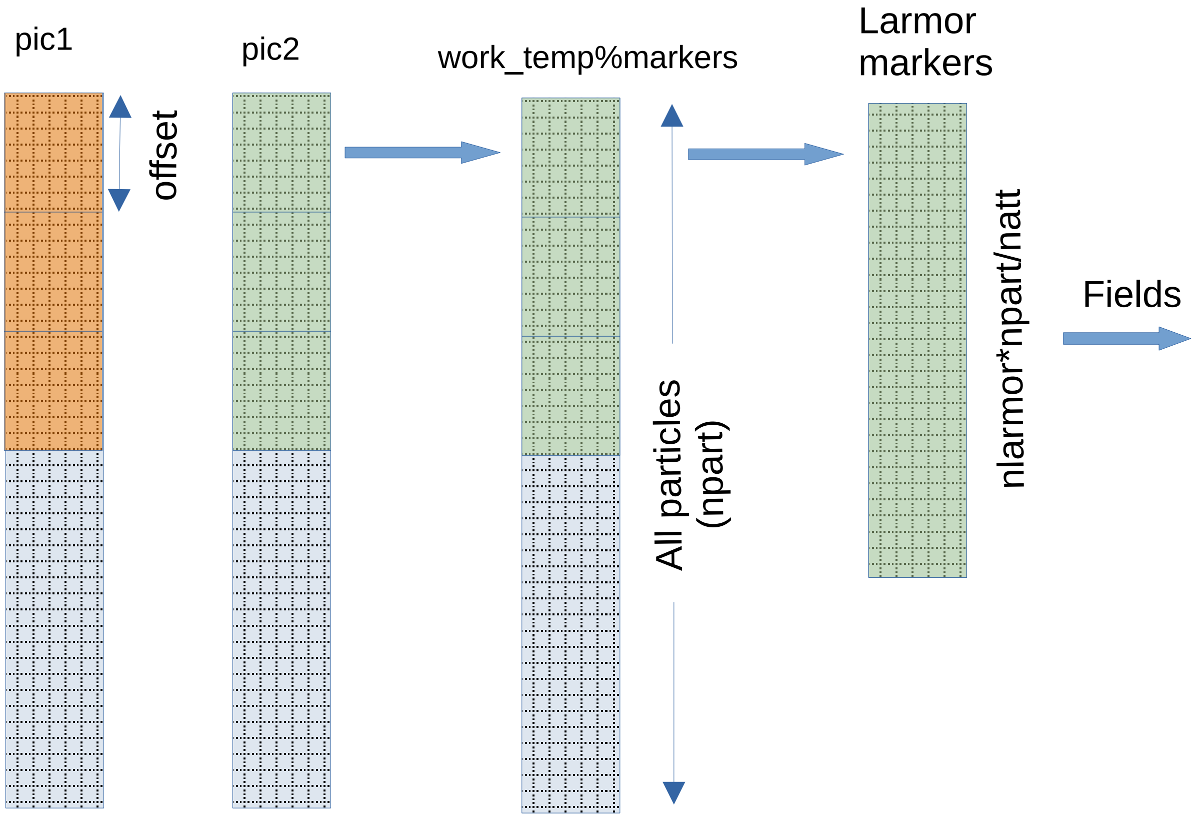




Reusing Reduced ATT arrays – Fields

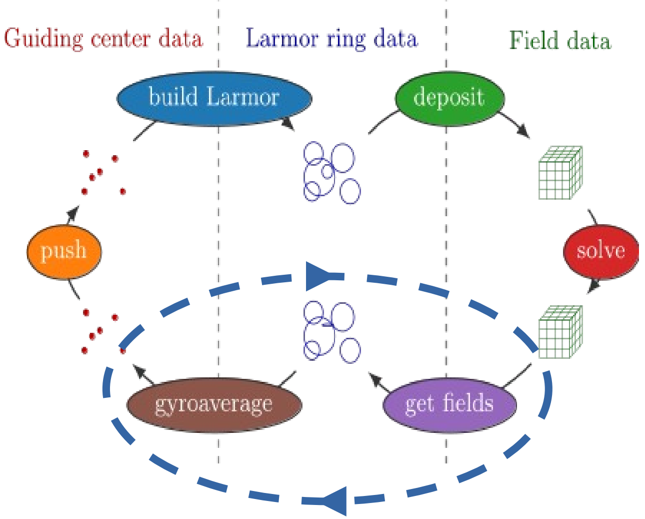
- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()
- call get_field()

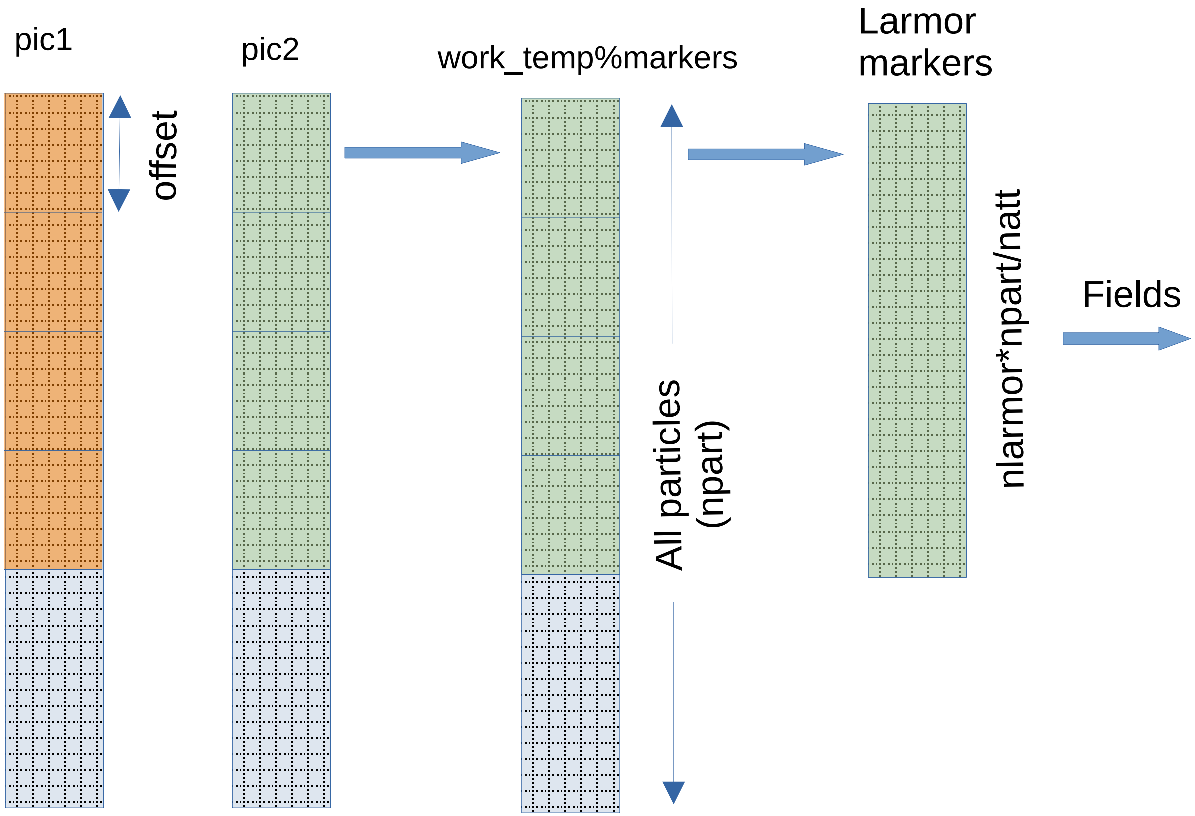




Reusing Reduced ATT arrays – Fields

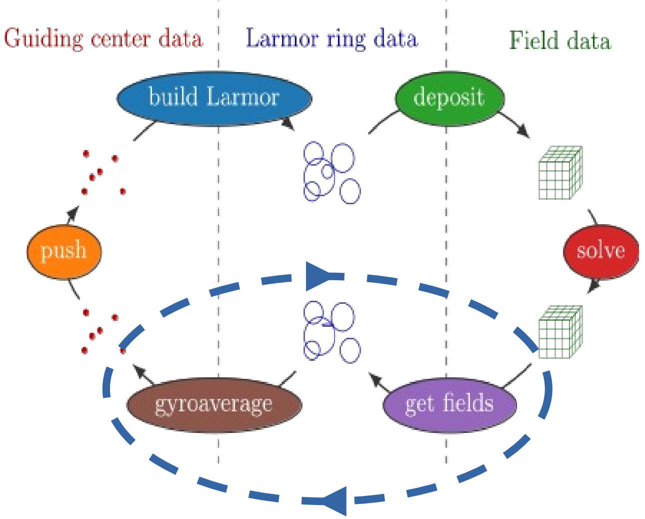
- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()
- call get_field()

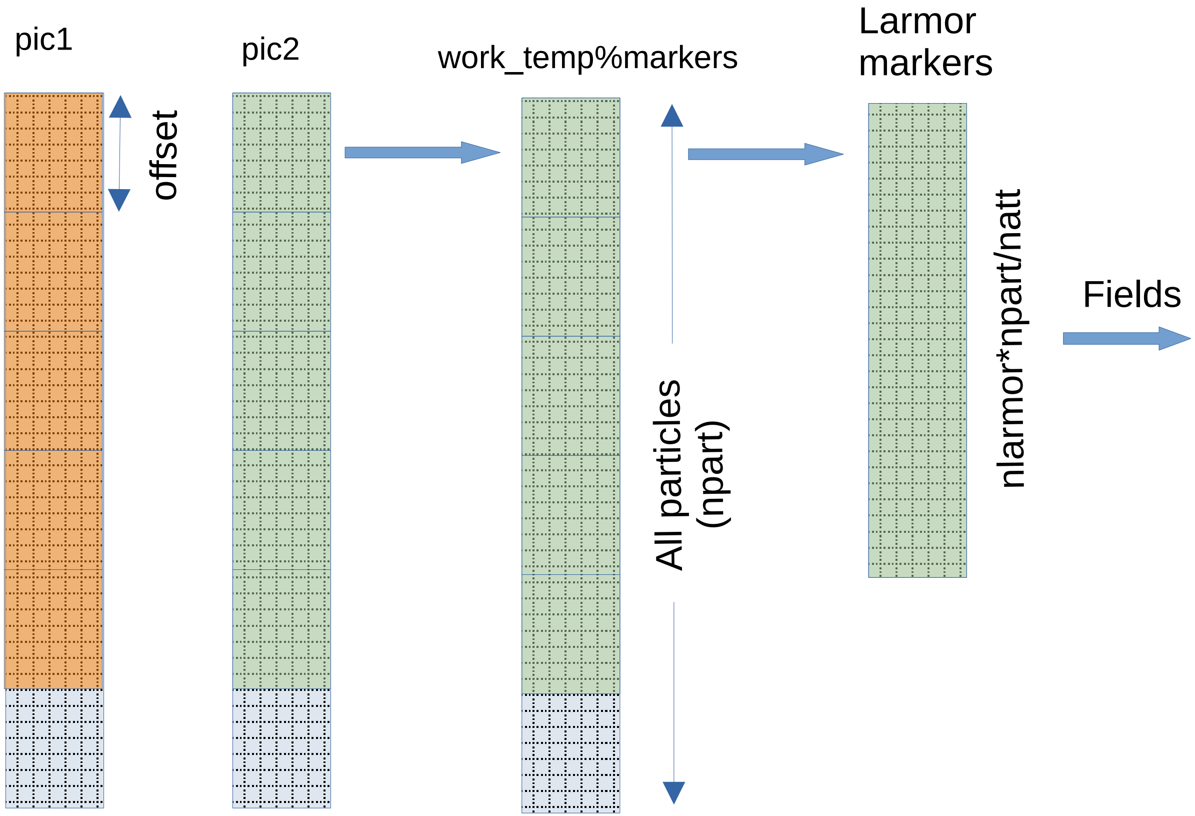




Reusing Reduced ATT arrays – Fields

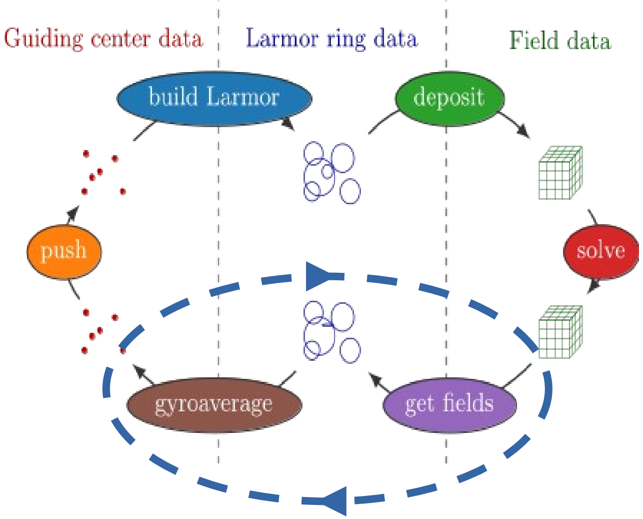
- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()
- call get_field()

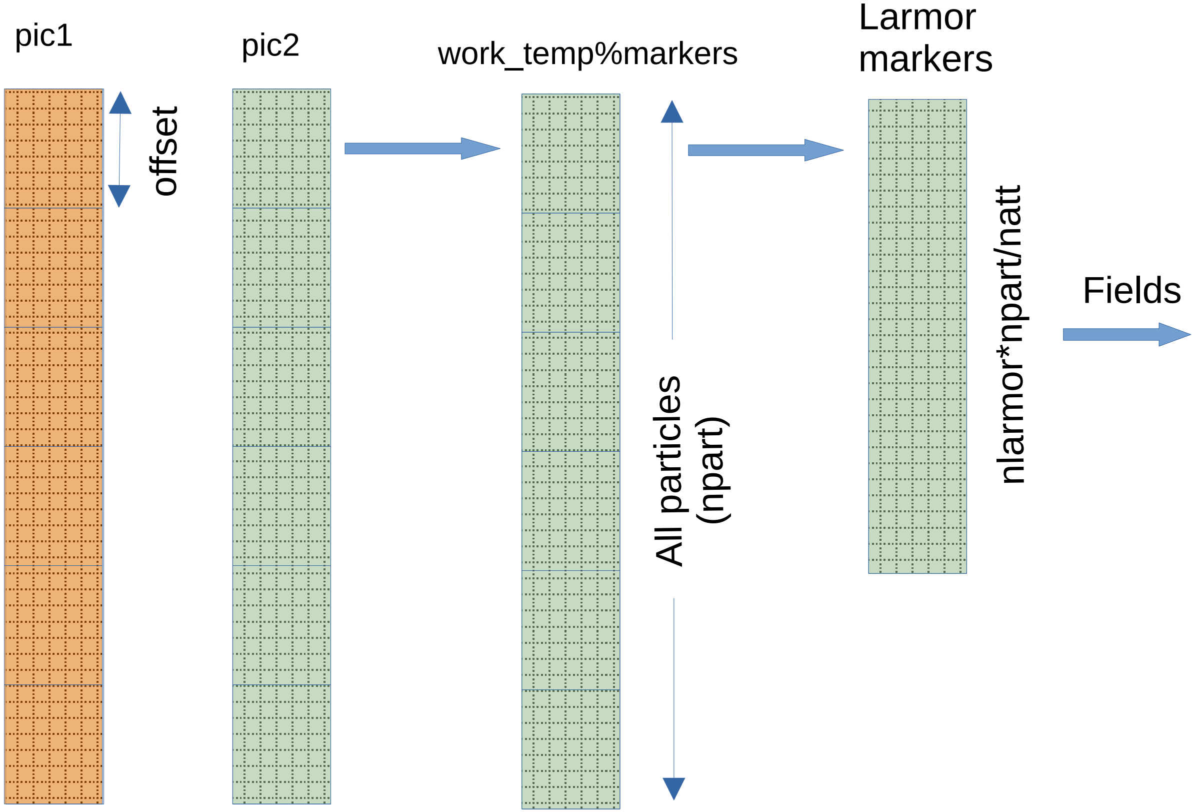




Reusing Reduced ATT arrays – Fields

- call get_larmor_indices()
- call compute_all_larmor_positions()
  - call assign_larmor_atts()
- call get_field()

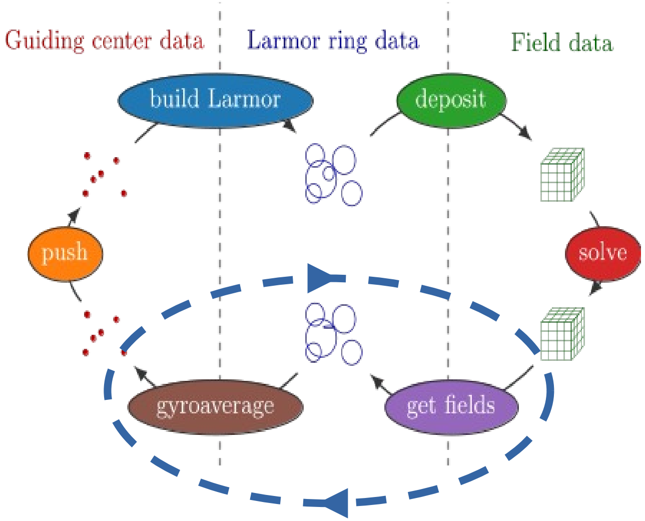




Code changes (original)

- get_field_disp (field)

```
do isp = 1, basic%nspecies
   do i_atts = 1,basic%natts_iter
      if(i_atts == 1) then
         call get_guiding_centre_field()
      end if
      if (basic%natts_iter .gt. 1) then
         np = npart(isp)/basic%natts_iter
         call compute_all_larmor_positions(np)
      end if
      call gyroaverage_field(...)
      call advance_atts_offset(isp)
   end do
   call reset_atts_offset(isp)
end do
```

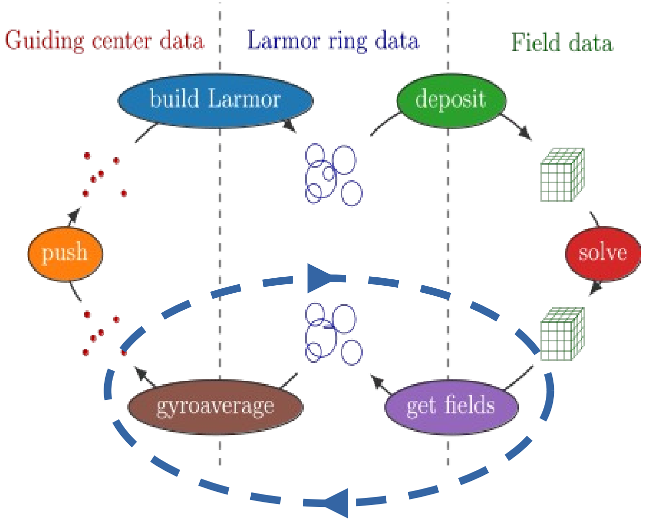


Code changes (optimised)

- get_field_disp (field)

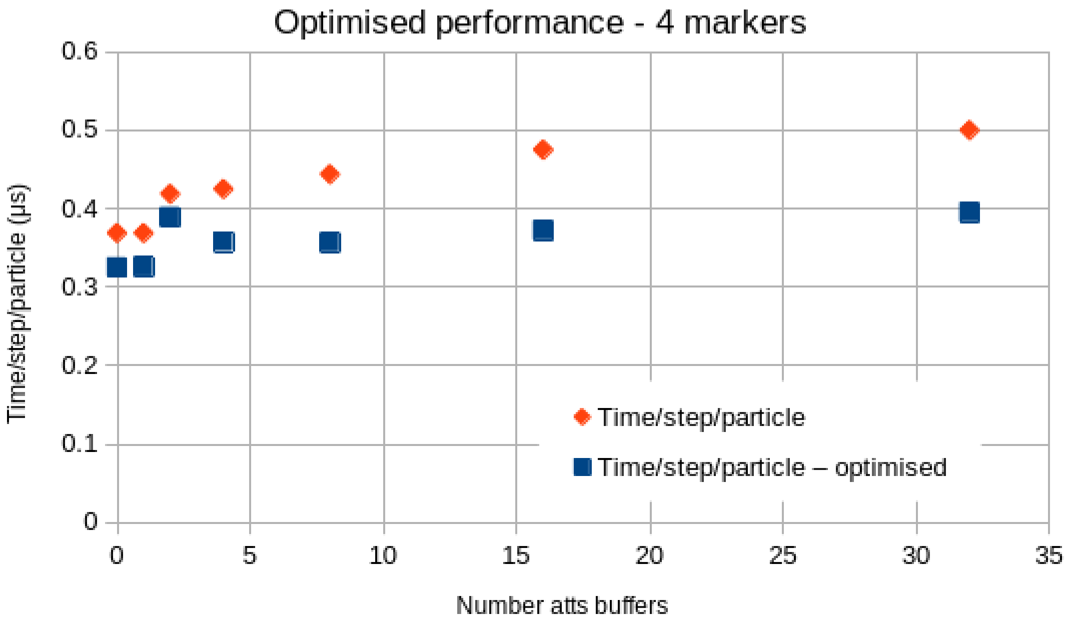
```
do i_att = 1, basic%natts_iter
   call update_larmor
   call get_field_disp(phi)
   call get_field_disp(apar)
   call get_field_disp(apar_sympl)
   call advance_atts_offset_disp
end do

call reset_atts_offset_disp
```



Optimisation

- Original - calculate Larmor positions for each species and field
  - Optimisation moved this outside get_field routines



Code changes

- Argument changes
  - fields:setrho(iteration)
  - larmor:set n larmor tot(isp,larmor_index,np)
  - larmor:set larmor max(isp,larmor_index,np)
  - part::get_larmor_indices(initflag_in, adapt_only_in)
  - part::compute_all_larmor_positions(comp_dk)
  - part:part grid(isp, nsel_att, reduce_flag, iteration, lb, ub, assign1, assign1_passing)
  - part:grid pic(isp, nsel_att, reduce_flag, lb, ub, assign1, iteration)
  - part:get field disp_lvl2(bspl,..., nl main substep, atts_offset) - *optional*
  - part:gyroaverage_field(markers,..., nl_main_substep, atts_offset) - *optional*
  - deposition(np, x, y, z, w, rhs, nl_depos, displs, first) - *optional*

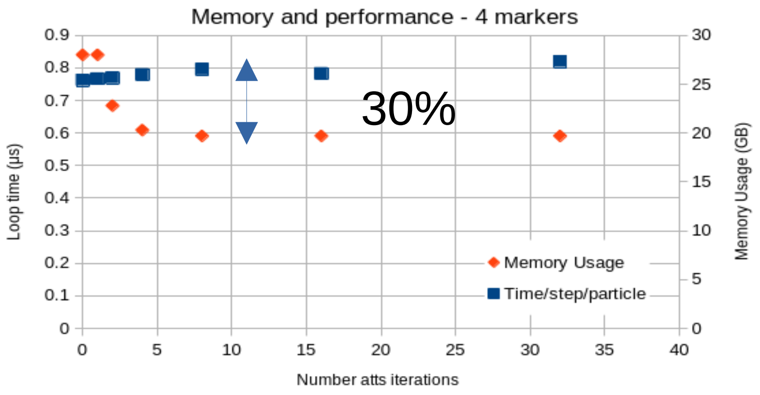
Testing

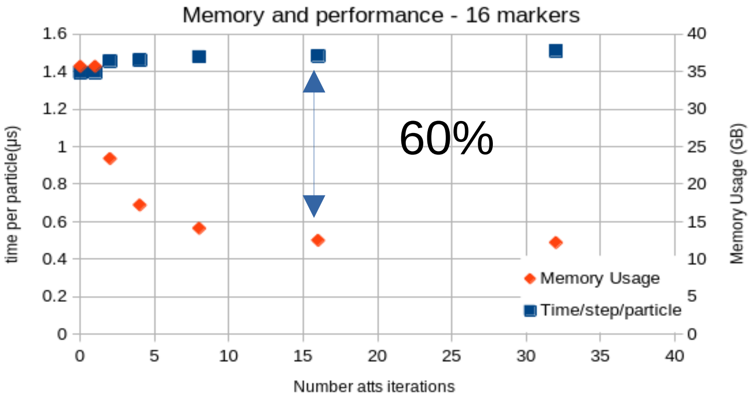
- Test cases
  - tae_pb, antenna, heating, saw
    - nsel_gyro=adapt
    - nsel_init_pert=POTENTIAL
- Restrictions
  - Only deposition by particle
  - nl_apar_den_corr
  - Sorting
- Documented at https://c4science.ch/w/orb5/technical_reports/

Performance - Raven

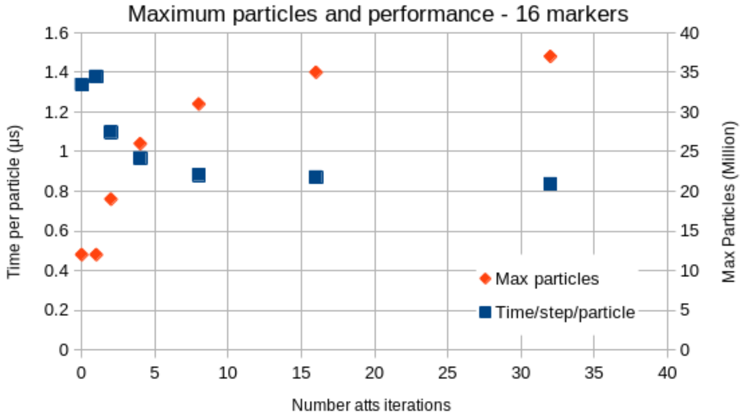
TAE_pb.in test case
12 million particles
Deuterium & electrons
Navg = 4,16 Larmor points

- 30%-60% reduction in memory < 10% slowdown



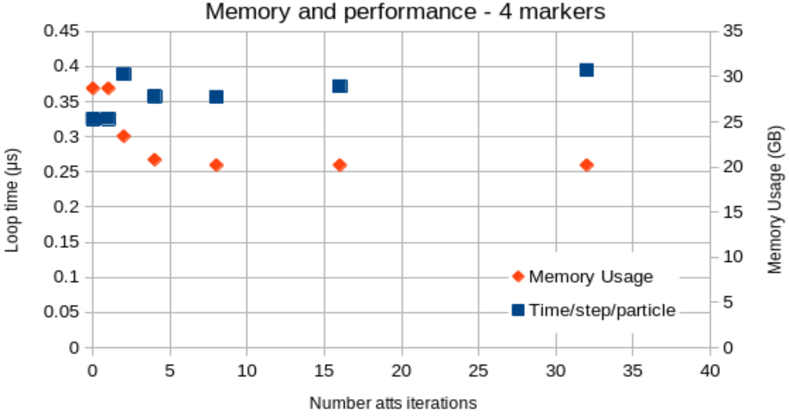


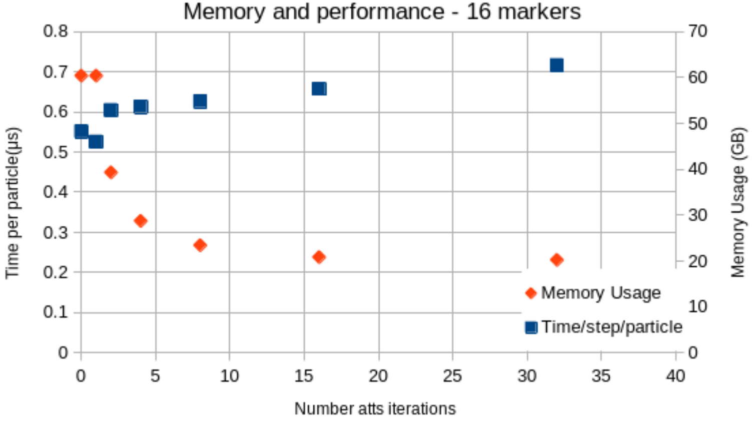
- 200% more particles/GPU
  - 30% speedup per particle



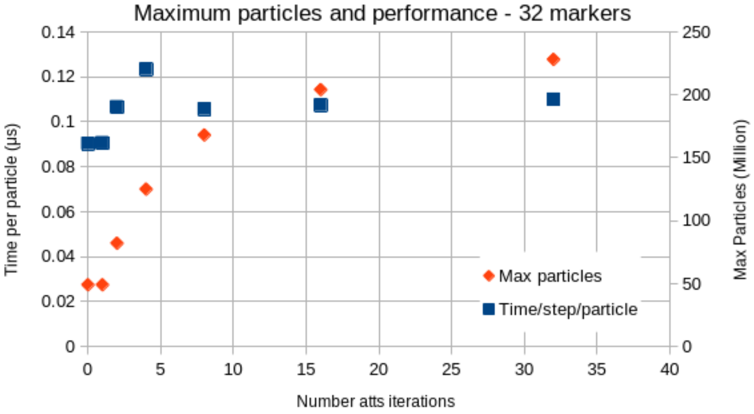
Performance - Leonardo

TAE_pb.in test case
12 million particles
Deuterium & electrons
Navg = 4-32 Larmor points





- x 5 more particles/GPU
  - Small performance reduction



Summary

- Large decrease in required memory dependant on number of markers
  - x 5 more particles for 32 Larmor markers
  - Small (<10%) decrease in performance for same number of markers
  - Small increase in performance when using maximum particles
- No impact on output
- Tested on several common options
  - Not all – documented
- Does not work with sorting, corrected density (apar_den_cor)