Information of Test Cases and Bugs

1. Programs

a) ANT-MOC

Input：

Geometry: Geometric input files.

Materials: Material data files.

XS File Layout: Material data structure file.

Tally Mesh: A grid about reaction rate output.

Num Azims: The number of azimuths.

Azim Spacing: Track spacing of the azimuth plane.

Num Polars: The number of polars.

Global Refines: Type of refines.

Segmentation: Type of track segmentation.

Z Mesh: Type of axial network.

Axial Zones: Overlapping plane axial layer.

Quadrature: Quadrature type.

Solver: Solver type.

Max Iterations: Maximum number of iterations.

Tolerance: Tolerance of convergence.

Output:

keff : Effective value-added factors

b) MISA-MD:

Input:

phasespace: the size of the simulate box, describing the number of lattices on each dimension.

cutoff_radius_factor: Truncation radius, calculating the maximum distance of the index when the force is affected between two atoms.

lattice_const: Related to atomic type.

timesteps: number of time steps during the whole simulation.

timesteps_length: time step.

create_phase: Reading atoms from a file or creating atoms.

create_t_set: the initial temperature.

create_seed: random seed.

fe: the percentage of Fe.

cu: the percentage of Cu.

ni: the percentage of Ni.

collision_step: a parameter of cascading collisions.

lat: a parameter of cascading collisions.

energy: the energy of Primary Knock-on Atom.

direction: the direction of Primary Knock-on Atom.

Output:

T： temperature.

E： energy.

lattice_const： lattice constant.

id： atom id

step： time step

type： atom type

inter_type： inter-atom type

locate.x： x coordinates

locate.y： y coordinates

locate.z： z coordinates

v.x： x-directional speed

v.y： x-directional speed

v.z： z-directional speed

c) MISA-SCD:

Input:

---

meshFile: mesh input file

meshType: a uniform or non-uniform mesh file.

strainField: Toggle for calculating defect diffusion interaction with strain field.

debugRestart: toggle restart from data file.

debugRestartFile: name of file to restart from

numMaterials: number of materials

materialFile: the material input file(s)

implantType: Type of implantation ('Cascade' for cascades, 'FrenkelPair' for Frenkel pairs).

implantScheme: Toggle Monte Carlo cascade introduction vs explicit cascade introduction (for better weak scaling).

cascadeFile: the cascade input file.

meshingType: whether using adaptive meshing protocol or not

implantDist: whether implanting defects uniformly.

implantFile: the data file containing non-uniform implantation profile.

temperature: Temperature, in K

CuContent: Initial content of Cu in iron

annealTemp: Annealing temperature, in K.

dpaRate: the rate of dpa.

atomSize: the size of atom.

firr: Radiation enhanced factor.

burgers: dislocation loop burgers vector.

totalDPA: total DPA in simulation

annealTime: total anneal time.

grainBoundaries: Toggle whether we are going to include the effect of grain boundaries (Removing defects that travel too far).

pointDefect: Toggle whether point defects are allowed to move only.

grainSize: Mean free path for interstitial clusters to travel before removal.

dislocDensity: dislocation density.

impurityConc: carbon impurity concentration (atomic fraction).

max3DInt: maximum size for SIA defect to diffuse in 3D.

cascadeVolume: volume of cascade, used for cascade-defect interactions.

numSims: number of times to repeat simulatio.

fineLength: Adaptive meshing parameters, the length of one cascade implantation element

numxFine: number of cascade elements in x-direction (fine mesh).

numyFine: number of cascade elements in y-direction (fine mesh).

numzFine: number of cascade elements in z-direction (fine mesh).

Output:

CuCluster: the number of Cu Cluster.

d) MISA-ETD

Input:

---

maxClusterSize : the max cluster size during simulation. **totalTime** : simulation time. **timeStep** : time step. **Cinit** : the initial concentration of cluster.

Output:

[n, c]: cluster size and cluster concentration.

e) ATHENA

Input :

Too many parameters, you can see it in "*test/input/ATHENA/Input information of ATHENA.docx*".

Output:

Rod Internal Pressure： including Initial Cold Fuel Rod Plenum Volume, Maximum Fuel Rod Internal Pressure, Peak nodal burnup, Rod average burnup, Fuel rod void volume, Fission gas release, etc.

Centerline Temperature: including Maximum Fuel Centerline Temperature, Axial node, Max Rad. Ave Fuel Enthalpy, Nodal burnup, Rod average burnup,etc.

Strain Increment: including Maximum Strain Increment, Axial node, Nodal burnup, Rod average burnup,etc.

2. Test Input

a) ANT-MOC

Custom testing:

Test ID	Reference	Location
ANT-MOC-CT01	Literature	test/input/ANT-MOC/CT/beavrs-3d.zip
ANT-MOC-CT02	Literature	test/input/ANT-MOC/CT/c5g7-3d-rodded-A.zip
ANT-MOC-CT03	Literature	test/input/ANT-MOC/CT/c5g7-3d-rodded-B.zip
ANT-MOC-T04	Literature	test/input/ANT-MOC/CT/c5g7-3d-unrodded.zip
ANT-MOC-CT05	Literature	test/input/ANT-MOC/CT/takeda-unrodded.zip
ANT-MOC-CT06	Literature	test/input/ANT-MOC/CT/takeda-rodded.zip

Differential testing:

Test ID	Reference	Location
ANT-MOC-DT01	ANT-MOC-CT02	
ANT-MOC-DT02	ANT-MOC-CT03	
ANT-MOC-DT03	ANT-MOC-CT04	
ANT-MOC-DT04	ANT-MOC-CT05	
ANT-MOC-DT05	ANT-MOC-CT06	
ANT-MOC-DT06~DT11	Derived from ANT-MOC-CT05 by changing the variable of "Num Polars" with "4, 6, ... , 14"	
ANT-MOC-DT12~DT17	Derived from ANT-MOC-CT06 by changing the variable of "Num Polars" with "4, 6, ... , 14"	

Test ID	Reference	Location
ANT-MOC-DT'01	OpenMOC	test/input/ANT-MOC/DT/c5g7-rodded-A.zip
ANT-MOC-DT'02	OpenMOC	test/input/ANT-MOC/DT/c5g7-rodded-B.zip
ANT-MOC-DT'03	OpenMOC	test/input/ANT-MOC/DT/c5g7-unrodded.zip
ANT-MOC-DT'04	OpenMOC	test/input/ANT-MOC/DT/takeda-unrodded.zip
ANT-MOC-DT'05	OpenMOC	test/input/ANT-MOC/CT/takeda-rodded.zip
ANT-MOC-DT'06~DT'11	Derived from ANT-MOC-DT'04 by changing the variable of "Num Polars" with "4, 6, ... , 14"	
ANT-MOC-DT'12~DT'17	Derived from ANT-MOC-DT'05 by changing the variable of "Num Polars" with "4, 6, ... , 14"	

Property-based Testing:

Test ID	Reference	Location
ANT-MOC-PT01	ANT-MOC-CT01	
ANT-MOC-PT02	ANT-MOC-CT02	
ANT-MOC-PT03	ANT-MOC-CT03	
ANT-MOC-CT04	ANT-MOC-CT04	
ANT-MOC-PT05	ANT-MOC-CT05	
ANT-MOC-PT06	ANT-MOC-CT06	
ANT-MOC-PT07	derived by expert experience	test/input/ANT-MOC/PT/beavrs-single-assembly.zip
ANT-MOC-PT08	derived by expert experience	test/input/ANT-MOC/PT/c5g7-single-assembly.zip

Metamorphic Testing:

MTGroup ID	Test ID	Reference	Location
ANT-MOC-MR2-G1	ANT-MOC-MR2-1-T01	ANT-MOC-CT02	/test/input/ANT-MOC/MT/MTGrou-deepth/c5g7-deepth1.zip
	ANT-MOC-MR2-G1-T02	derived from ANT-MOC-CT02, setting the deepth of assembly1 at "level2"	/test/input/ANT-MOC/MT/MTGrou-deepth/c5g7-deepth2.zip
	ANT-MOC-MR2-G1-T03	derived from ANT-MOC-CT02, setting the deepth of assembly1 at "level3"	/test/input/ANT-MOC/MT/MTGrou-deepth/c5g7-deepth3.zip
ANT-MOC-MR3-G1	ANT-MOC-MR3-G1-T01~T08	derived from ANT-MOC-CT02, changing the variable of "Num Polar " with "2, 4, 6, ..., 16"	test/input/ANT-MOC/MT/MTGroup-polar-c5g7-rodded-A/polar = ${i} (i=2, 4, 6, ..., 16)

ANT-MOC-MR3-G2	ANT-MOC-MR3-G2-T01~T08	derived from ANT-MOC-CT03, changing the variable of "Num Polar " with "2, 4, 6, ..., 16"	test/input/ANT-MOC/MT/MTGroup-polar-c5g7-rodded-B/polar = ${i} (i=2, 4, 6, ..., 16)
ANT-MOC-MR3-G3	ANT-MOC-MR3-G3-T01~T08	derived from ANT-MOC-CT04, changing the variable of "Num Polar " with "2, 4, 6, ..., 16"	test/input/ANT-MOC/MT/MTGroup-polar-c5g7-unrodded/polar = ${i} (i=2, 4, 6, ..., 16)
ANT-MOC-MR3-G4	ANT-MOC-MR3-G4-T01~T08	derived from ANT-MOC-CT05, changing the variable of "Num Polar " with "2, 4, 6, ..., 40"	test/input/ANT-MOC/MT/MTGroup-polar-c5g7-unrodded/polar = ${i} (i=2, 4, 6, ..., 40)
ANT-MOC-MR3-G5	ANT-MOC-MR3-G5-T01~T08	derived from ANT-MOC-CT05, setting the variable of "Num Modules" at "1,1,1", changing the variable of "Num Polar " with "2, 4, 6, ..., 40"	
ANT-MOC-MR3-G6	ANT-MOC-MR3-G6-T01~T08	derived from ANT-MOC-CT05, setting the variable of "Num Modules" at "5,5,1", changing the variable of "Num Polar " with "2, 4, 6, ..., 40"	

b) MISA-SCD

Custom Testing:

ID	Reference	Location
MISA-SCD-CT01	electron irradiation	test/input/MISA-SCD/CT/test-electron.zip
MISA-SCD-CT02	neutron irradiation	test/input/MISA-SCD/CT/test-neutron.zip
MISA-SCD-CT03	electron irradiation using adaptive grid	test/input/MISA-SCD/CT/test-electron-adaptive.zip
MISA-SCD-CT04	neutron irradiation using adaptive grid	test/input/MISA-SCD/CT/test-neutron-adaptive.zip

Property-based Testing:

ID	Reference	Location
MISA-SCD-PT01	MISA-SCD-CT01	
MISA-SCD-PT02	MISA-SCD-T02	
MISA-SCD-PT03	MISA-SCD-CT03	
MISA-SCD-T04	MISA-SCD-CT04	

c) MISA-MD

Test ID	Reference	Location
MISA-MD-T01	Expert experience	test/input/MISA-MD/DT/1kev-<135>/MISA-MD.yaml
MISA-MD-T02~T07	derived from MISA-MD-T01, changing the variable of "energy" with "5000, 10000, 15000, ..., 30000"	test/input/MISA-MD/DT/${i}kev-<135>/MISA-MD.yaml, (i=5, 10, 15,..., 30)
MISA-MD-T08	derived from MISA-MD-T02, changing the variable of "direction" with [1,2,2]	test/input/MISA-MD/DT/5kev-<122>/MISA-MD>.yaml
MISA-MD-T09	derived from MISA-MD-T02, changing the variable of "direction" with [2,3,5]	test/input/MISA-MD/DT/5kev-<235>/MISA-MD.yaml
MISA-MD-T10	derived from MISA-MD-T02, changing the variable of "direction" with [2,4,5]	test/input/MISA-MD/DT/5kev-<245>/MISA-MD.yaml,
MISA-MD-T11	derived from MISA-MD-T02, changing the variable of "direction" with [3,4,5]	test/input/MISA-MD/DT/5kev-<345>/MISA-MD.yaml

Differential testing

Test ID	Reference	Location
MISA-MD-DT01~DT11	MISA-MD-T01~T11	
MISA-MD-DT'01	Expert experience	test/input/MISA-MD/DT/1kev-<135>/lammps.in
MISA-MD-DT'02~DT'07	derived from MISA-MD-DT'01 by changing the line of 3, 4, 5 of the input file	test/input/MISA-MD/DT/${i}kev-<135>/lammps.in, (i=5, 10, 15,..., 30)
MISA-MD-DT'08	derived from MISA-MD-DT'01 by changing the line of 3, 4, 5 of the input file	test/input/MISA-MD/DT/1kev-<122>/lammps.in
MISA-MD-DT'09	derived from MISA-MD-DT'01 by changing the line of 3, 4, 5 of the input file	test/input/MISA-MD/DT/1kev-<235>/lammps.in
MISA-MD-DT'010	derived from MISA-MD-DT'01 by changing the line of 3, 4, 5 of the input file	test/input/MISA-MD/DT/1kev-<245>/lammps.in
MISA-MD-DT'011	derived from MISA-MD-DT'01 by changing the line of 3, 4, 5 of the input file	test/input/MISA-MD/DT/1kev-<345>/lammps.in

Property-based Testing

Test ID	Reference	Location
MISA-MD-PT01	MISA-MD-T01	test/input/MISA-MD/PT/5kev-<135>-80/MISA-MD.yaml
MISA-MD-PT02	MISA-MD-T02	test/input/MISA-MD/PT/5kev-<122>-80/MISA-MD.yaml
MISA-MD-PT03	Expert experience	test/input/MISA-MD/PT/5kev-<135>-50/MISA-MD.yaml
MISA-MD-PT04	Expert experience	test/input/MISA-MD/PT/5kev-<135>-80-Cu-Ni/MISA-MD.yaml

Metamorphic testing:

MTGroup ID	Test ID	Reference	Location
MISA-MD-MR4-G1	MISA-MD-MR4-G1-T01~T07	MISA-MD-T01~T07	
MISA-MD-MR5-G1	MISA-MD-MR5-G1-T01~T04	derived from MISA-MD-T01, increasing the variable of "phasespace" for "[50,50,50], [60,60,60] ,... [80,80,80]	
MISA-MD-MR6-G1	MISA-MD-MR6-G1-T01	derived from MISA-MD-T01, increasing the variable of "Ni"	test/input/MISA-MD/MT/MTGroup-Ni/test0.yaml
	MISA-MD-MR6-G1-T02		test/input/MISA-MD/MT/MTGroup-Ni/test1.yaml
	MISA-MD-MR6-G1-T03		test/input/MISA-MD/MT/MTGroup-Ni/test2.yaml
MISA-MD-MR7-G1	MISA-MD-MR6-G1-T01	derived from MISA-MD-T01, increasing the variable of "Cu"	test/input/MISA-MD/MT/MTGroup-Cu/test0.yaml
	MISA-MD-MR6-G1-T01		test/input/MISA-MD/MT/MTGroup-Cu/test1.yaml
	MISA-MD-MR6-G1-T01		test/input/MISA-MD/MT/MTGroup-Cu/test2.yaml
MISA-MD-MR8-G1	MISA-MD-MR8-G1-T01~T07	MISA-MD-T01~T07	

MTGroup ID	Test ID	Reference	Location
MISA-MD-MR9-G1	MISA-MD-MR9-G1-T01~T07	MISA-MD-T01~T07	

d) MISA-ETD

custom testing:

Test ID	Reference	Location
MISA-ETD-CT01	literature	test/input/MISA-ETD/CT/test0.in

Differential Testing:

Test ID	Reference	Location
MISA-ETD-DT01	MISA-ETD-CT01	
MISA-ETD-DT02	derived from MISA-ETD-CT01, changing the variable of totalTime from "100000" to "150000"	test/input/MISA-ETD/DT/totalTime=150000.in
MISA-ETD-DT03	derived from MISA-ETD-CT01, changing the variable of totaltime from "150000" to "200000"	test/input/MISA-ETD/DT/totalTime=200000.in
MISA-ETD-DT'01	Implementing another version of the program（matlab）	Console input
MISA-ETD-DT'02	derived from MISA-ETD-DT'01, changing the variable of totaltime from "100000" to "150000"	
MISA-ETD-DT'03	derived from MISA-ETD-DT'01, changing the variable of totaltime from "100000" to "200000"	

Metamorphic testing:

MTGroup ID	Test ID	Reference	Location
MISA-ETD-MR10-G1	MISA-ETD-MR10-G1-T0001~T1000	Derived from MISA-ETD-CT01, changing the variable of totalTime from "1000, 2000, 3000, ..., 1000000"	

Richardson’s Extrapolation:

Test ID	Reference	Location
MISA-ETD-RE01	MISA-ETD-T01	
MISA-ETD-RE02	derived from MISA-ETD-T01, replace the variable of timeStep with "timeStep / 2"	test/input/MISA-ETD/RE/dt=0.05.in
MISA-ETD-RE03	derived from MISA-ETD-RE02, replace the variable of timeStep with "timeStep / 2"	test/input/MISA-ETD/RE/dt=0.025
MISA-ETD-RE04	derived from MISA-ETD-RE03, replace the variable of timeStep with "timeStep / 2"	test/input/MISA-ETD/RE/dt=0.0125
MISA-ETD-RE05	derived from MISA-ETD-RE04, replace the variable of timeStep with "timeStep / 2"	test/input/MISA-ETD/RE/dt=0.00625
MISA-ETD-RE06	derived from MISA-ETD-RE05, replace the variable of timeStep with "timeStep / 2"	test/input/MISA-ETD/RE/dt=0.003125

e) ATHENA

Custom Testing:

ID	Reference	Location
ATHENA-T01	expert experience	test/input/ATHENA/CT/test0.inp

Differential Testing:

ID	Reference	Location
ATHENA-DT01	ATHENA-T01	
ATHENA-DT02	expert experience	test/input/ATHENA/DT/test1.inp
ATHENA-DT03	expert experience	test/input/ATHENA/DT/test2.inp
ATHENA-DT04	expert experience	test/input/ATHENA/DT/test3.inp

Property-baseed Testing:

ID	Reference	Location
ATHENA-PT01	ATHENA-DT01	
ATHENA-PT02	ATHENA-DT02	
ATHENA-PT03	ATHENA-DT03	
ATHENA-PT04	ATHENA-DT04	

Richardson’s Extrapolation:

ID	Reference	Location
ATHENA-RE01	derived from ATHENA-RE01, changing the variable of "na" with "5"	test/input/ATHENA/RE/na=5.inp
ATHENA-RE02	derived from ATHENA-RE01, replacing the variable of "na" with "na * 2"	test/input/ATHENA/RE/na=10.inp
ATHENA-RE03	derived from ATHENA-RE02, replacing the variable of "na" with "na * 2"	test/input/ATHENA/RE/na=20.inp
ATHENA-RE04	derived from ATHENA-Re03, replacing the variable of "na" with "na * 2"	test/input/ATHENA/RE/na=40.inp
ATHENA-RE05	derived from ATHENA-Re04, replacing the variable of "na" with "na * 2"	test/input/ATHENA/RE/na=80.inp

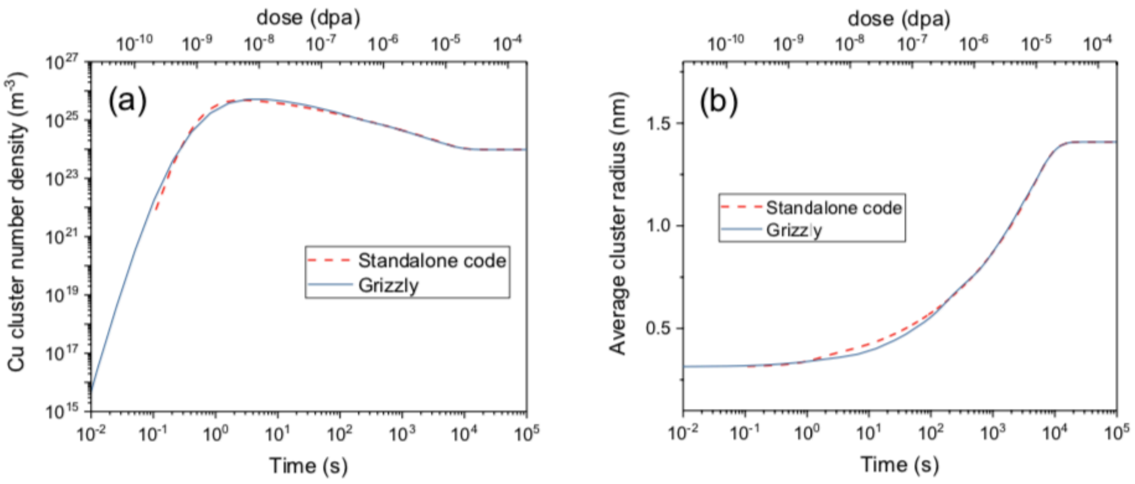
3. Oracles

a) Expected results

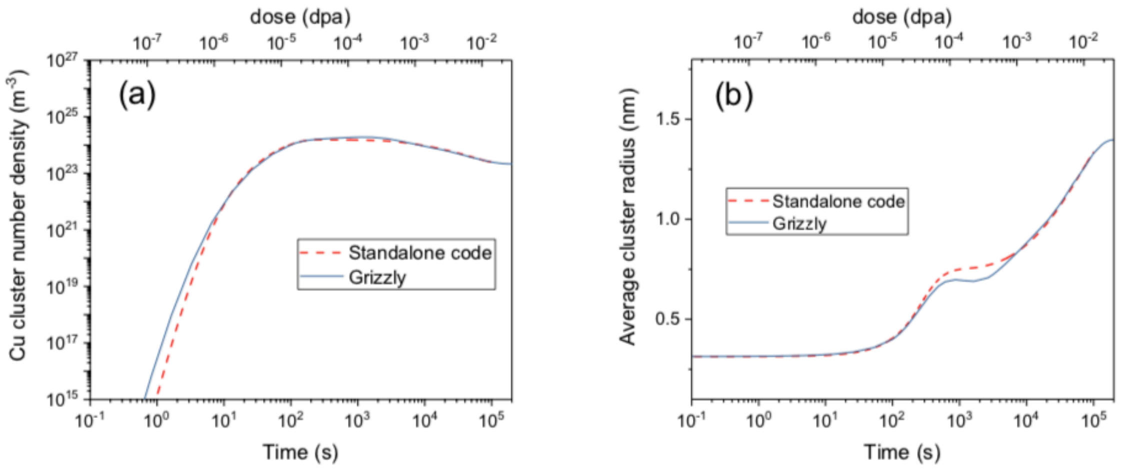
Test Id	Description	xpected	Expected type
ANT-MOC-CT01	beavrs-3d	0.99927	Value + Deviation
ANT-MOC-CT02	C5g7-rodded-A	1.12806	Value + Deviation
ANT-MOC-CT03	C5g7-rodded-B	1.07777	Value + Deviation
ANT-MOC-CT04	C5g7-unrodded	1.14308	Value + Deviation
ANT-MOC-CT05	Takeda-unrodded	0.97732	Value + Deviation
ANT-MOC-CT06	Takeda-rodded	0.9623	Value + Deviation

Test Id	Description	Expected	Expected type
MISA-SCD-CT01	Electron irradiation	Evolutionary Trends	Picture from literature
MISA-SCD-CT02	Neutron irradiation	Evolutionary Trends	Picture from literature
MISA-SCD-CT03	Electron irradiation	Evolutionary Trends	Picture from literature
MISA-SCD-CT04	Neutron irradiation	Evolutionary Trends	Picture from literature

Electron irradiation：

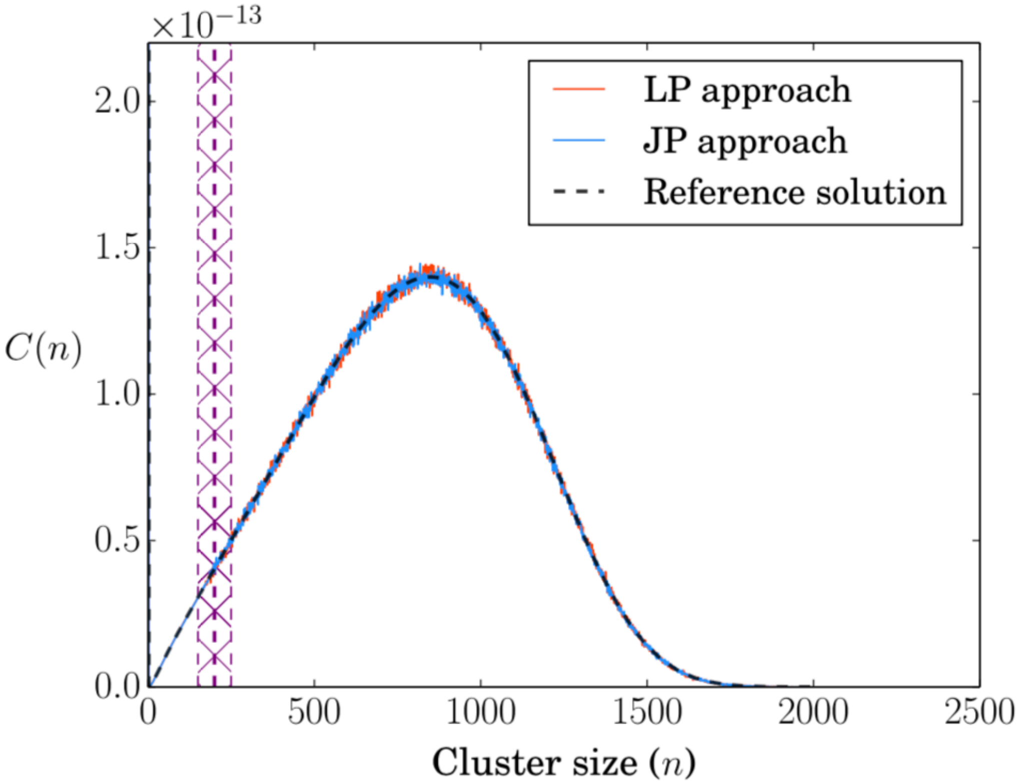


Neutron irradiation:



Test Id	Description	Expected	Expected type
MISA-ETD-CT01	test	Matrix	Picture from literature

Reference result：



b) Propertis

Property id	Definition	Reason
P1	The ID of any characteristic line must be unique	Implementation
P2	The ID of any characteristic line must be a positive integer	Implementation
P3	The length of any characteristic line must be a positive integer	Implementation
P4	The ID of FSR must be unique	Implementation
P5	The flux distribution must be geometrically symmetric	Physics & Computation
P6	The ID of any molecule must be unique	Implementation
P7	The position and velocity of a molecule should not be NAN	Implementation
P8	The position of a molecule must be within the simulation space	Implementation
P9	The action and the reaction force between any two molecules must be identical	Physics

Property id	Definition	Reason
P10	The kinetic energy of the entire system must be conserved (equal to the input PKA energy)	Physics
P11	The number of molecules should not change	Physics
P12	Within two consecutive time steps, the change of the force exerted on a molecule should not be greater than ε, where $\varepsilon = 20$	Domain knowledge
P13	$\neg(Cu < 0 \wedge V < 0 \wedge SIA\_m < 0 \wedge SIA\_im < 0)$	Computation
P14	$\neg(SIA\_m > 0 \wedge SIA\_im > 0)$	Computation
P15	$\neg((SIA\_m > 0 \vee SIA\_im > 0) \wedge V > 0)$	Computation
P16	The pressure of the system should not change within the inner loop	Numeric algorithm
P17	The result of the inner loop must converge	Numeric algorithm

c) MRs

MR id	Definition	Reason
MR1	For any two test case i and i', if i is identical to i' except that i is performed on a single physical core and i' is performed on multiple physical cores, then $keff(p(i)) = keff(p(i'))$	Execution
MR2	For a source test case i, if the follow-up test case i' is derived from i by increasing the depth of a control pin, then $keff(p(i))>keff(p(i'))$	Physics
MR3	For a source test case i0, if follow-up test cases i1, i2,... are derived by gradually increasing the amount of polar angles, i.e., by densifying the grids, then $keff(p(ik)) − keff(p(ik+1))<2×10$, when $k\longrightarrow+\infty$	Computation
MR4	For a source test case i, if the follow-up test case i' is derived from i by increasing the initial PKA, then $T(p(i))<T(p(i'))$, where T extracts the system temperature from the program output	Physics
MR5	For a source test case i, if the follow-up test case i' is derived from i by increasing the simulation space, then $T(p(i))>T(p(i'))$	Physics
MR6	For a source test case i, if the follow-up test case i' is derived from i by increasing Ni content, then $T(p(i))=T(p(i'))$	Physics
MR7	For a source test case i, if the follow-up test case i' is derived from i by increasing Cu content, then $T(p(i))=T(p(i'))$	Physics
MR8	For a source test case i, if the follow-up test case i' is derived from i by increasing the initial PKA, then $Fs(p(i))<Fs(p(i'))$, where Fs extracts the amount of Frankel defects from the program output.	Physics
MR9	For a source test case i, if the follow-up test case i' is derived from i by increasing the initial PKA, then $Fp(p(i))<Fp(p(i'))$, where Fp extracts the peak value of Frankel defects from the program output	Physics

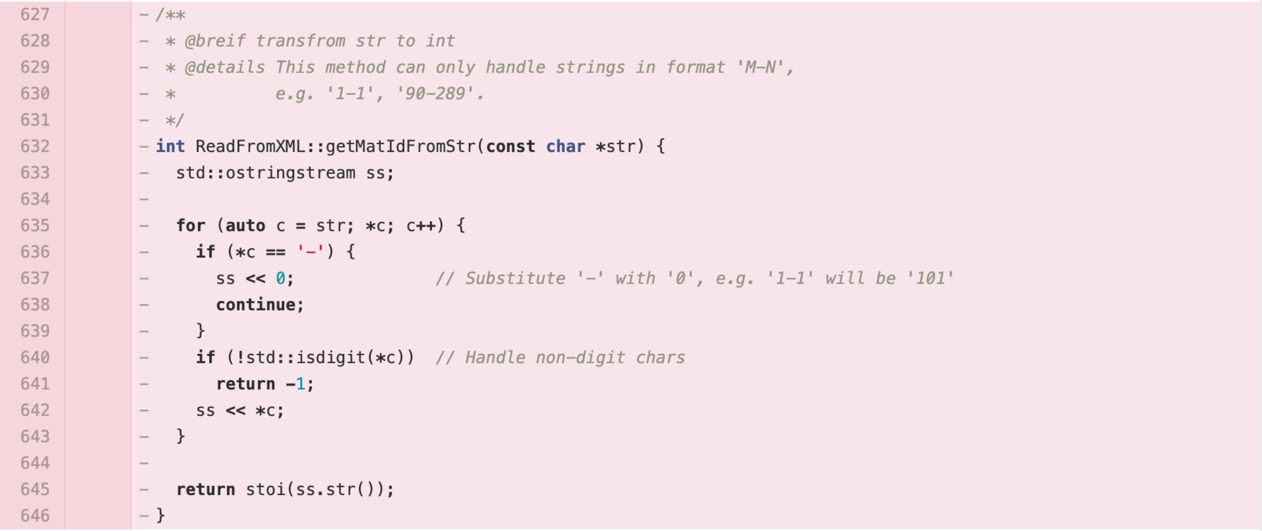
MR id	Definition	Reason
MR10	If the simulation times of the source and the follow-up test cases i and i' are larger than a certain T0, $p(i) = p(i')$	Physics

4. Bugs

a) ANT-MOC

B1: When deserializing from an input file, “-” is converted into “0”. Hence, “10-1” and “1-01” will both become “1001”

Related Code:



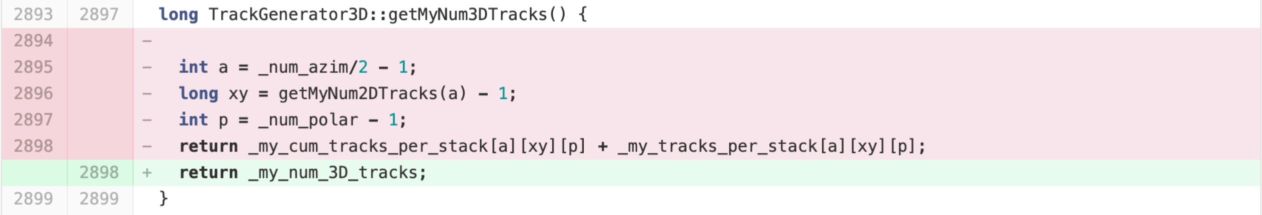
```
- /**
-  * @breif transfrom str to int
-  * @details This method can only handle strings in format 'M-N',
-  *          e.g. '1-1', '90-289'.
-  */
- int ReadFromXML::getMatIdFromStr(const char *str) {
-   std::ostringstream ss;
-
-   for (auto c = str; *c; c++) {
-     if (*c == '-') {
-       ss << 0;              // Substitute '-' with '0', e.g. '1-1' will be '101'
-       continue;
-     }
-     if (!std::isdigit(*c))  // Handle non-digit chars
-       return -1;
-     ss << *c;
-   }
-
-   return stoi(ss.str());
- }
```

Reason: Substitute '-' with '0' may has a bug.

B2: A negative number may be generated for ID of characteristic line

B2: A negative number may be generated for ID of characteristic line

Related code:

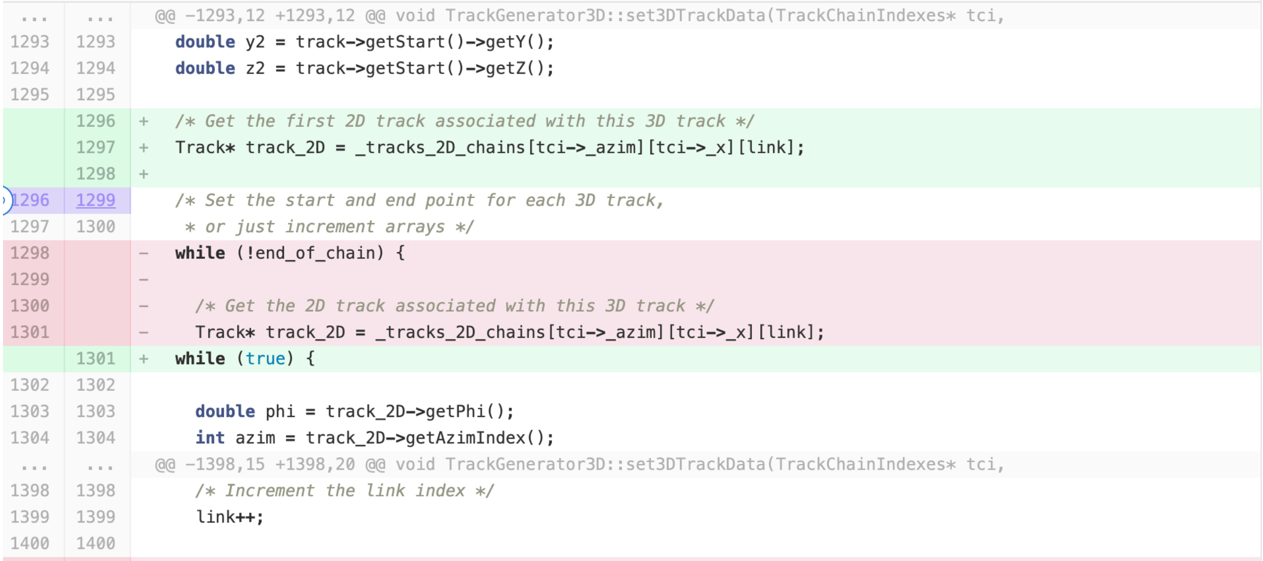


```
  long TrackGenerator3D::getMyNum3DTracks() {
-
-   int a = _num_azim/2 - 1;
-   long xy = getMyNum2DTracks(a) - 1;
-   int p = _num_polar - 1;
-   return _my_cum_tracks_per_stack[a][xy][p] + _my_tracks_per_stack[a][xy][p];
+   return _my_num_3D_tracks;
  }
```

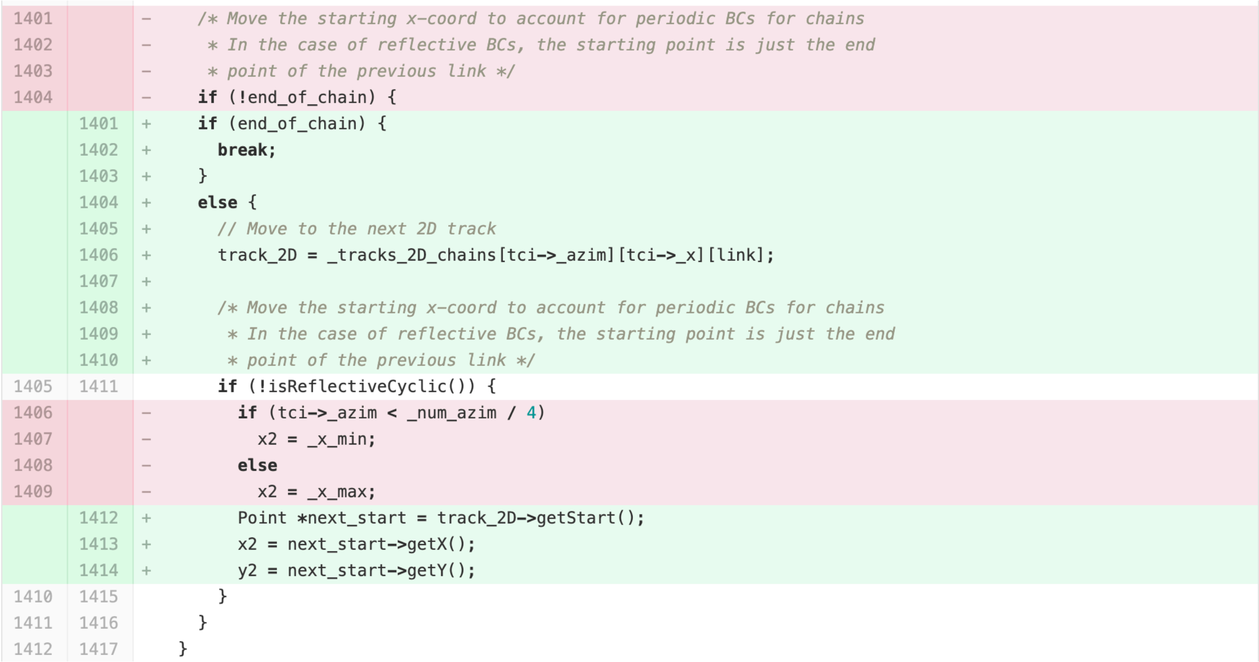
Reason: There may be no tracks under a specified azimuthal angle for a certain process, which would lead to a `0` when `getMyNum2DTracks()` was invoked. If it happens, `xy` evaluates to `-1`, cause a negative number.

B3: Around the corner of the simulation space, the length of a characteristic line may be 0

Related Code:



```
@@ -1293,12 +1293,12 @@ void TrackGenerator3D::set3DTrackData(TrackChainIndexes* tci,
   double y2 = track->getStart()->getY();
   double z2 = track->getStart()->getZ();

+  /* Get the first 2D track associated with this 3D track */
+  Track* track_2D = _tracks_2D_chains[tci->_azim][tci->_x][link];
+
   /* Set the start and end point for each 3D track,
    * or just increment arrays */
-  while (!end_of_chain) {
-
-    /* Get the 2D track associated with this 3D track */
-    Track* track_2D = _tracks_2D_chains[tci->_azim][tci->_x][link];
+  while (true) {

     double phi = track_2D->getPhi();
     int azim = track_2D->getAzimIndex();
@@ -1398,15 +1398,20 @@ void TrackGenerator3D::set3DTrackData(TrackChainIndexes* tci,
     /* Increment the link index */
     link++;

```

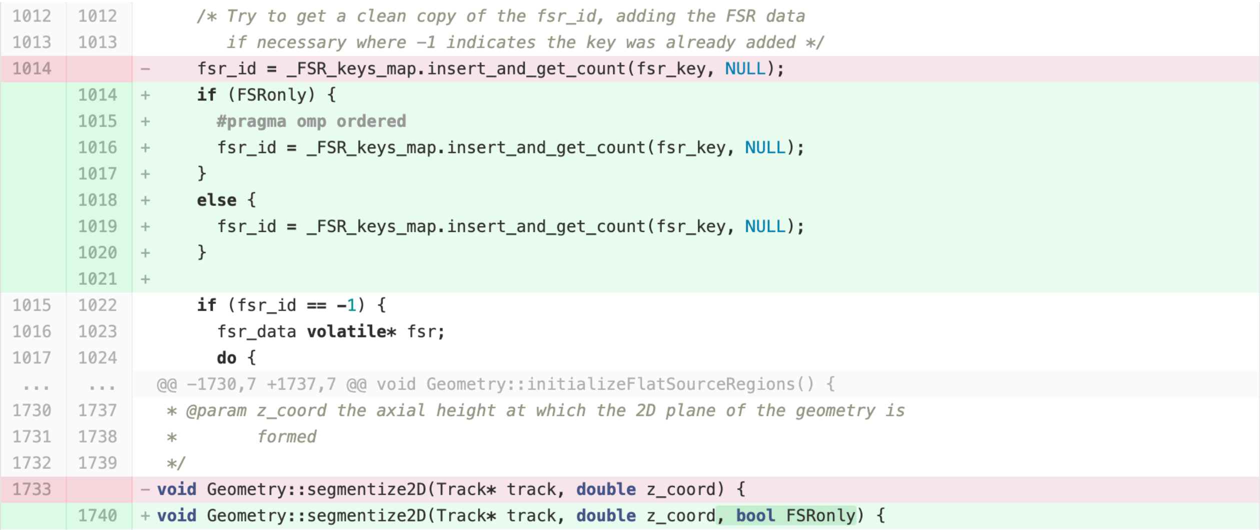


```
-    /* Move the starting x-coord to account for periodic BCs for chains
-     * In the case of reflective BCs, the starting point is just the end
-     * point of the previous link */
-    if (!end_of_chain) {
+    if (end_of_chain) {
+      break;
+    }
+    else {
+      // Move to the next 2D track
+      track_2D = _tracks_2D_chains[tci->_azim][tci->_x][link];
+
+      /* Move the starting x-coord to account for periodic BCs for chains
+       * In the case of reflective BCs, the starting point is just the end
+       * point of the previous link */
       if (!isReflectiveCyclic()) {
-        if (tci->_azim < _num_azim / 4)
-          x2 = _x_min;
-        else
-          x2 = _x_max;
+        Point *next_start = track_2D->getStart();
+        x2 = next_start->getX();
+        y2 = next_start->getY();
       }
     }
   }
```

Reason: tracks shoot out the surface YMAX will always re-enter the geometry at corners, which probably produces invalid tracks or zero-lengh tracks or something

B4: The generation of FSR ID is not unique

Related Code:

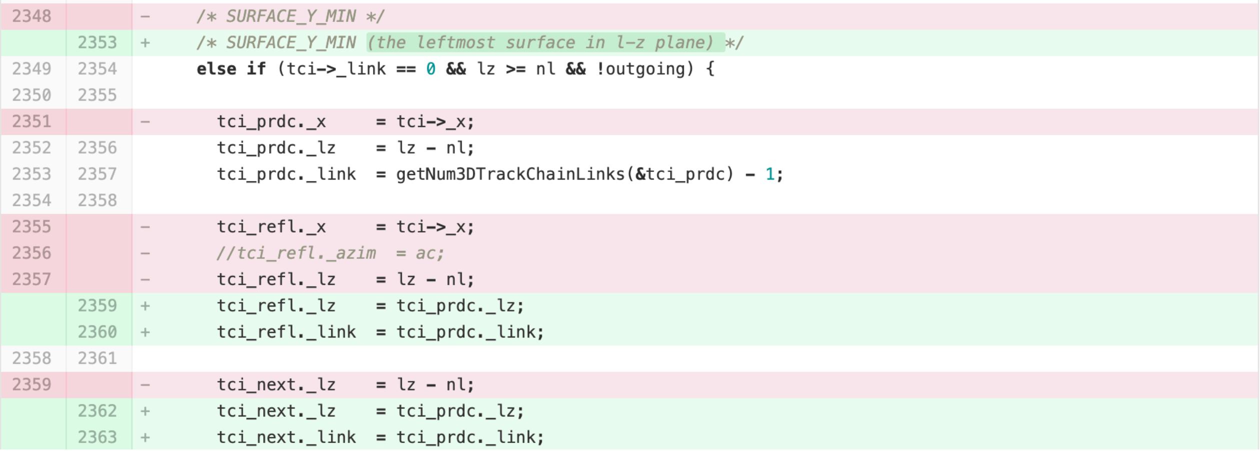


```diff
     /* Try to get a clean copy of the fsr_id, adding the FSR data
        if necessary where -1 indicates the key was already added */
-    fsr_id = _FSR_keys_map.insert_and_get_count(fsr_key, NULL);
+    if (FSRonly) {
+      #pragma omp ordered
+      fsr_id = _FSR_keys_map.insert_and_get_count(fsr_key, NULL);
+    }
+    else {
+      fsr_id = _FSR_keys_map.insert_and_get_count(fsr_key, NULL);
+    }
+
     if (fsr_id == -1) {
       fsr_data volatile* fsr;
       do {
 ...  ...
@@ -1730,7 +1737,7 @@ void Geometry::initializeFlatSourceRegions() {
  * @param z_coord the axial height at which the 2D plane of the geometry is
  *        formed
  */
-void Geometry::segmentize2D(Track* track, double z_coord) {
+void Geometry::segmentize2D(Track* track, double z_coord, bool FSRonly) {
```

Reason: Because of the cyclic track decomposition algorithm, FSRs are supposed to be synchronized across processes. However, this means that we have to break the thread-parallelized FSR generation algorithm to ensure that FSRs are numbered in the same order across all processes..

B5: Related Code: Due to the incorrect offset computation, the length of a characteristic line may be 0

Related Code:

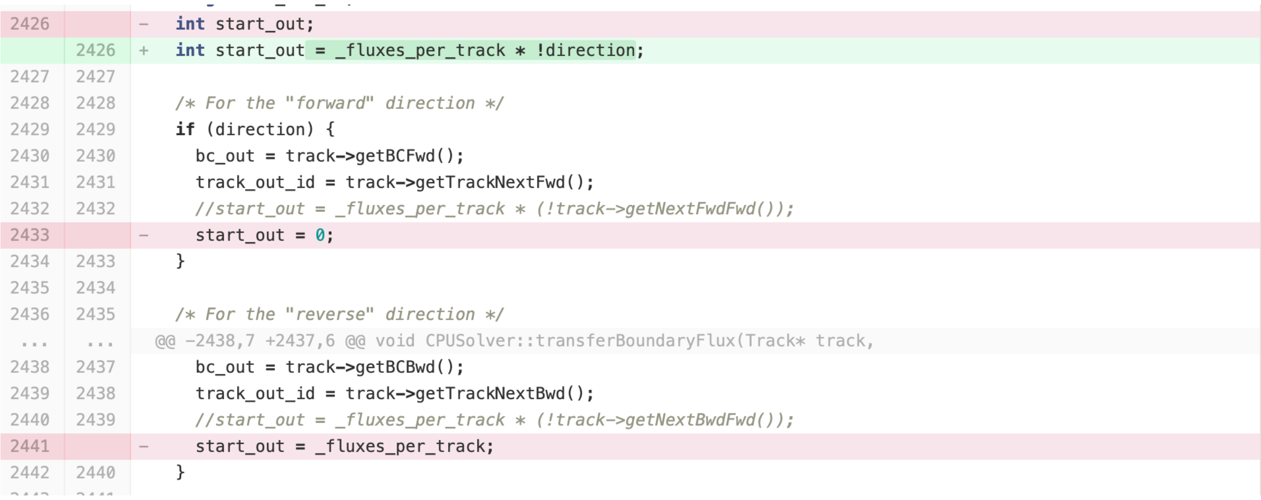


```diff
-    /* SURFACE_Y_MIN */
+    /* SURFACE_Y_MIN (the leftmost surface in l-z plane) */
     else if (tci->_link == 0 && lz >= nl && !outgoing) {
 
-      tci_prdc._x     = tci->_x;
       tci_prdc._lz    = lz - nl;
       tci_prdc._link  = getNum3DTrackChainLinks(&tci_prdc) - 1;
 
-      tci_refl._x     = tci->_x;
-      //tci_refl._azim  = ac;
-      tci_refl._lz    = lz - nl;
+      tci_refl._lz    = tci_prdc._lz;
+      tci_refl._link  = tci_prdc._link;
 
-      tci_next._lz    = lz - nl;
+      tci_next._lz    = tci_prdc._lz;
+      tci_next._link  = tci_prdc._link;
```

Reason: mismatched links on YMIN and YMAX surfaces.

B6: Due to mismatch of the endpoints of a characteristic line, the length of the line may be 0

Related Code:

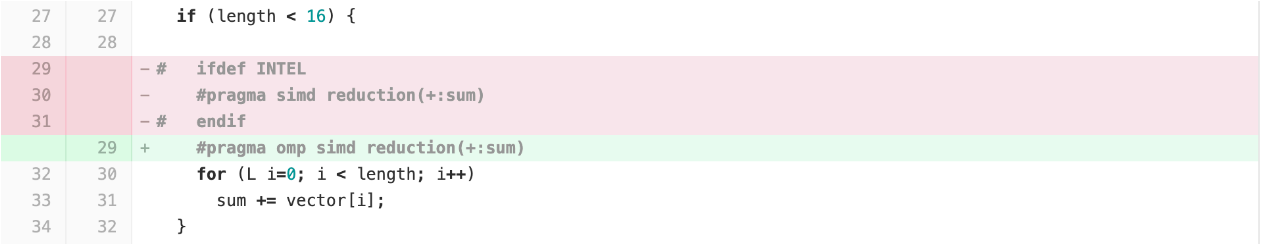


```
-   int start_out;
+   int start_out = _fluxes_per_track * !direction;

    /* For the "forward" direction */
    if (direction) {
      bc_out = track->getBCFwd();
      track_out_id = track->getTrackNextFwd();
      //start_out = _fluxes_per_track * (!track->getNextFwdFwd());
-     start_out = 0;
    }

    /* For the "reverse" direction */
 @@ -2438,7 +2437,6 @@ void CPUSolver::transferBoundaryFlux(Track* track,
      bc_out = track->getBCBwd();
      track_out_id = track->getTrackNextBwd();
      //start_out = _fluxes_per_track * (!track->getNextBwdFwd());
-     start_out = _fluxes_per_track;
    }
```

Reason: zero-length caused by wrongly nudged endpoints of tracks.

B8: The ICPC compiler of Tianhe-2 supercomputer is incompatible with our program.

Related Code:

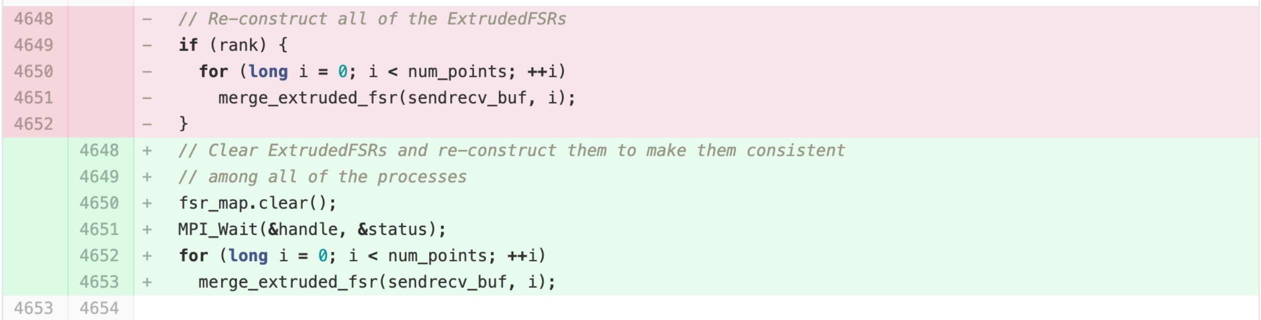


```
    if (length < 16) {

-#    ifdef INTEL
-     #pragma simd reduction(+:sum)
-#    endif
+     #pragma omp simd reduction(+:sum)
      for (L i=0; i < length; i++)
        sum += vector[i];
    }
```

Reason: According to the manual, `simd` can be replaced by `omp simd` construct.

B9: Error in load balance algorithm

Related Code:



```
-   // Re-construct all of the ExtrudedFSRs
-   if (rank) {
-     for (long i = 0; i < num_points; ++i)
-       merge_extruded_fsr(sendrecv_buf, i);
-   }
+   // Clear ExtrudedFSRs and re-construct them to make them consistent
+   // among all of the processes
+   fsr_map.clear();
+   MPI_Wait(&handle, &status);
+   for (long i = 0; i < num_points; ++i)
+     merge_extruded_fsr(sendrecv_buf, i);
```

Reason: Load balancing for angular decomposition has a confirm issue that the fluxes may be wrongly computed when a process has zero angle. This is not fixed yet and should be avoided by set the number of azimuthal angles properly.

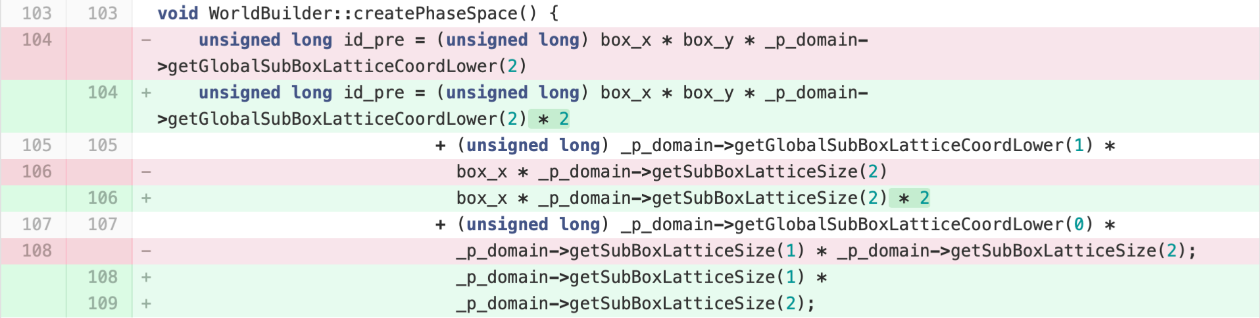
b) MISA-MD

B1： Error in communication algorithm causes the collision of molecule IDs

A part of error output showing below :

ID	Locate.x	...	ID	Locate.x
62501	0.0151414	...	62501	71.3981
31251	0.0151414	...	31251	71.3981
93751	0.0151414		93751	71.3981

Related code :

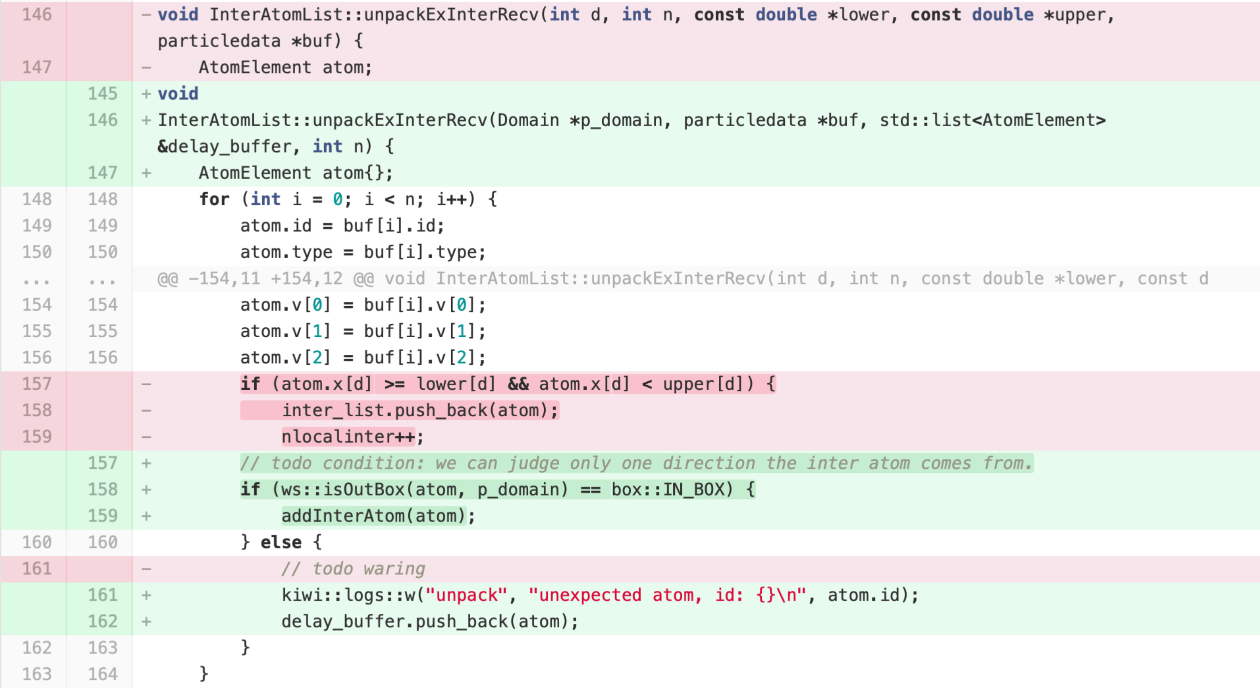


```
  void WorldBuilder::createPhaseSpace() {
-     unsigned long id_pre = (unsigned long) box_x * box_y * _p_domain->getGlobalSubBoxLatticeCoordLower(2)
+     unsigned long id_pre = (unsigned long) box_x * box_y * _p_domain->getGlobalSubBoxLatticeCoordLower(2) * 2
                             + (unsigned long) _p_domain->getGlobalSubBoxLatticeCoordLower(1) *
-                              box_x * _p_domain->getSubBoxLatticeSize(2)
+                              box_x * _p_domain->getSubBoxLatticeSize(2) * 2
                             + (unsigned long) _p_domain->getGlobalSubBoxLatticeCoordLower(0) *
-                              _p_domain->getSubBoxLatticeSize(1) * _p_domain->getSubBoxLatticeSize(2);
+                              _p_domain->getSubBoxLatticeSize(1) *
+                              _p_domain->getSubBoxLatticeSize(2);
```

Reason: Error in Computation-Parallelization.

B3: Error in indexing computation causes the loss of molecules in communication

Related Code:

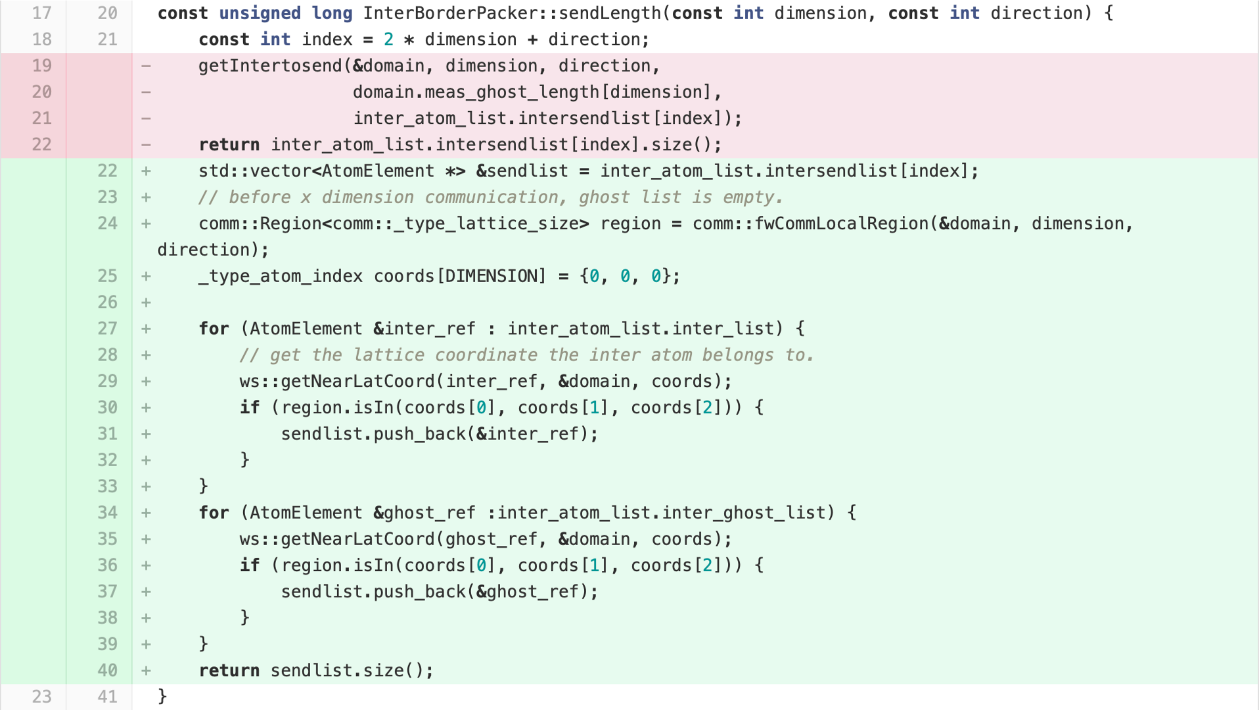


```
- void InterAtomList::unpackExInterRecv(int d, int n, const double *lower, const double *upper, particledata *buf) {
-     AtomElement atom;
+ void
+ InterAtomList::unpackExInterRecv(Domain *p_domain, particledata *buf, std::list<AtomElement> &delay_buffer, int n) {
+     AtomElement atom{};
      for (int i = 0; i < n; i++) {
          atom.id = buf[i].id;
          atom.type = buf[i].type;
...  @@ -154,11 +154,12 @@ void InterAtomList::unpackExInterRecv(int d, int n, const double *lower, const d
          atom.v[0] = buf[i].v[0];
          atom.v[1] = buf[i].v[1];
          atom.v[2] = buf[i].v[2];
-         if (atom.x[d] >= lower[d] && atom.x[d] < upper[d]) {
-             inter_list.push_back(atom);
-             nlocalinter++;
+         // todo condition: we can judge only one direction the inter atom comes from.
+         if (ws::isOutBox(atom, p_domain) == box::IN_BOX) {
+             addInterAtom(atom);
          } else {
-             // todo waring
+             kiwi::logs::w("unpack", "unexpected atom, id: {}\n", atom.id);
+             delay_buffer.push_back(atom);
          }
      }
```

Reason: Error in indexing computation causes the loss of molecules in communication, inter atoms are filtered in *InterAtomList::unpackExInterRecv()*.

B4: Incorrect communication strategy

Related code:



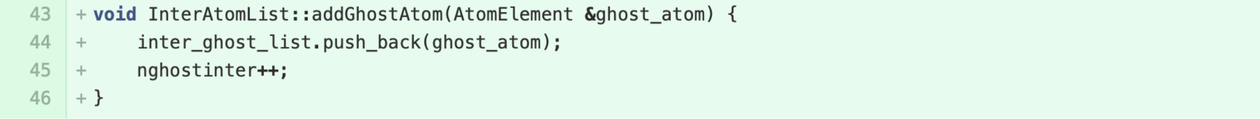
```
 const unsigned long InterBorderPacker::sendLength(const int dimension, const int direction) {
     const int index = 2 * dimension + direction;
-    getIntertosend(&domain, dimension, direction,
-                   domain.meas_ghost_length[dimension],
-                   inter_atom_list.intersendlist[index]);
-    return inter_atom_list.intersendlist[index].size();
+    std::vector<AtomElement *> &sendlist = inter_atom_list.intersendlist[index];
+    // before x dimension communication, ghost list is empty.
+    comm::Region<comm::_type_lattice_size> region = comm::fwCommLocalRegion(&domain, dimension, direction);
+    _type_atom_index coords[DIMENSION] = {0, 0, 0};
+
+    for (AtomElement &inter_ref : inter_atom_list.inter_list) {
+        // get the lattice coordinate the inter atom belongs to.
+        ws::getNearLatCoord(inter_ref, &domain, coords);
+        if (region.isIn(coords[0], coords[1], coords[2])) {
+            sendlist.push_back(&inter_ref);
+        }
+    }
+    for (AtomElement &ghost_ref :inter_atom_list.inter_ghost_list) {
+        ws::getNearLatCoord(ghost_ref, &domain, coords);
+        if (region.isIn(coords[0], coords[1], coords[2])) {
+            sendlist.push_back(&ghost_ref);
+        }
+    }
+    return sendlist.size();
 }
```

Reason: Incorrect communication strategy in *InterBorderPacker::sendLength()*.

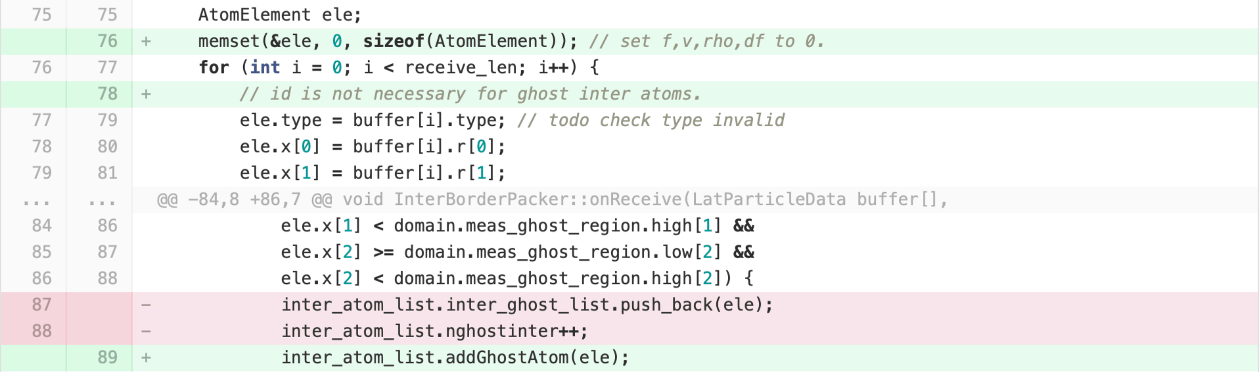
B5: Incorrect communication algorithm in ghost regions

Related Code:

Adding new GhostAtom operation.



```
+void InterAtomList::addGhostAtom(AtomElement &ghost_atom) {
+    inter_ghost_list.push_back(ghost_atom);
+    nghostinter++;
+}
```

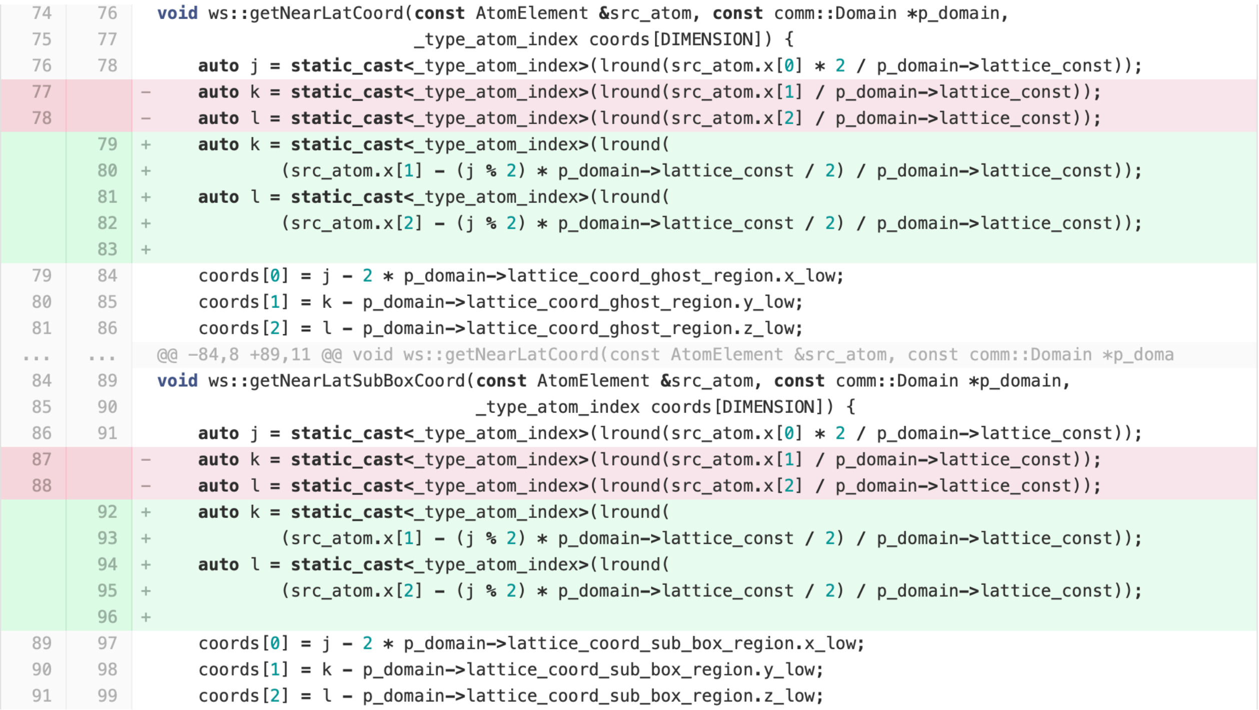


```
     AtomElement ele;
+    memset(&ele, 0, sizeof(AtomElement)); // set f,v,rho,df to 0.
     for (int i = 0; i < receive_len; i++) {
+        // id is not necessary for ghost inter atoms.
         ele.type = buffer[i].type; // todo check type invalid
         ele.x[0] = buffer[i].r[0];
         ele.x[1] = buffer[i].r[1];
@@ -84,8 +86,7 @@ void InterBorderPacker::onReceive(LatParticleData buffer[],
             ele.x[1] < domain.meas_ghost_region.high[1] &&
             ele.x[2] >= domain.meas_ghost_region.low[2] &&
             ele.x[2] < domain.meas_ghost_region.high[2]) {
-            inter_atom_list.inter_ghost_list.push_back(ele);
-            inter_atom_list.nghostinter++;
+            inter_atom_list.addGhostAtom(ele);
```

Reason: Incorrect communication algorithm in ghost regions.

B6: Error in indexing computation when updating molecule velocity

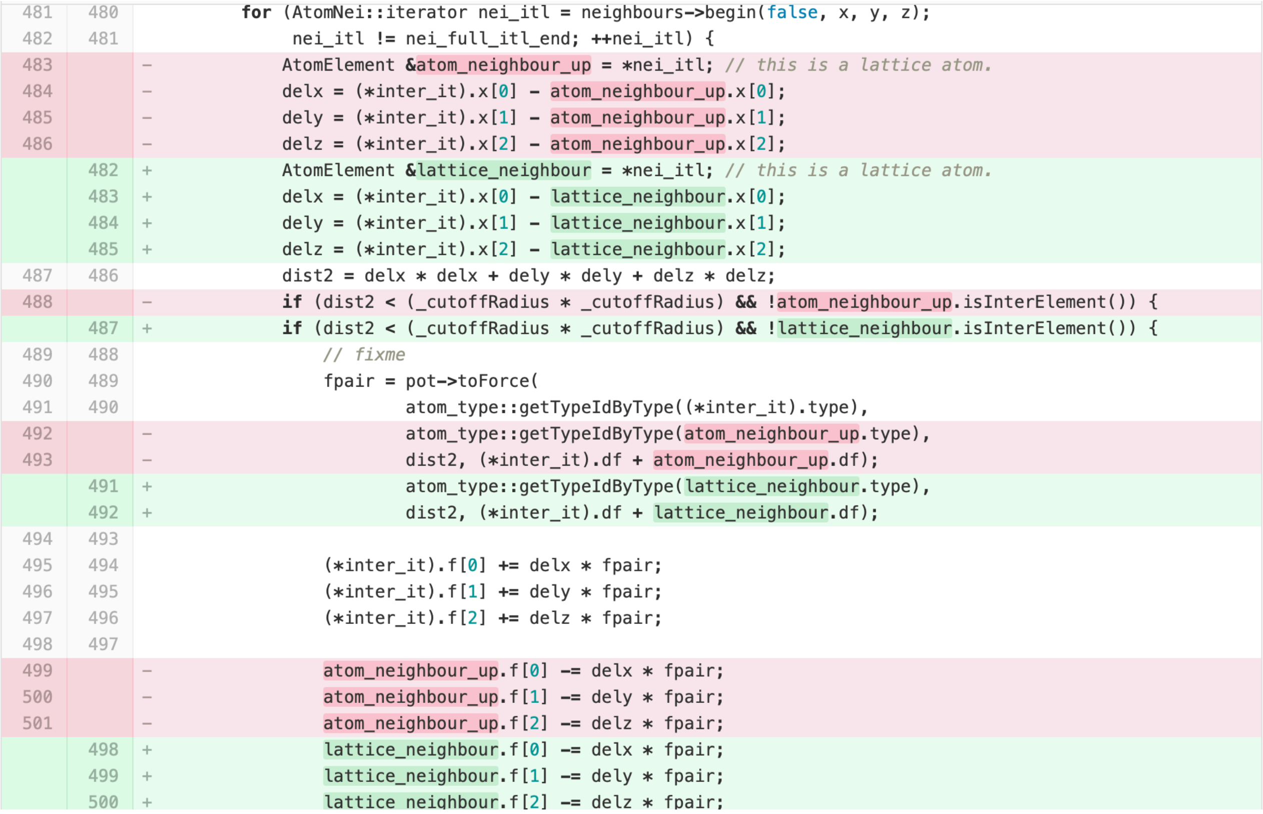
Related Code:



```diff
 void ws::getNearLatCoord(const AtomElement &src_atom, const comm::Domain *p_domain,
                          _type_atom_index coords[DIMENSION]) {
     auto j = static_cast<_type_atom_index>(lround(src_atom.x[0] * 2 / p_domain->lattice_const));
-    auto k = static_cast<_type_atom_index>(lround(src_atom.x[1] / p_domain->lattice_const));
-    auto l = static_cast<_type_atom_index>(lround(src_atom.x[2] / p_domain->lattice_const));
+    auto k = static_cast<_type_atom_index>(lround(
+            (src_atom.x[1] - (j % 2) * p_domain->lattice_const / 2) / p_domain->lattice_const));
+    auto l = static_cast<_type_atom_index>(lround(
+            (src_atom.x[2] - (j % 2) * p_domain->lattice_const / 2) / p_domain->lattice_const));
+
     coords[0] = j - 2 * p_domain->lattice_coord_ghost_region.x_low;
     coords[1] = k - p_domain->lattice_coord_ghost_region.y_low;
     coords[2] = l - p_domain->lattice_coord_ghost_region.z_low;
@@ -84,8 +89,11 @@ void ws::getNearLatCoord(const AtomElement &src_atom, const comm::Domain *p_doma
 void ws::getNearLatSubBoxCoord(const AtomElement &src_atom, const comm::Domain *p_domain,
                                _type_atom_index coords[DIMENSION]) {
     auto j = static_cast<_type_atom_index>(lround(src_atom.x[0] * 2 / p_domain->lattice_const));
-    auto k = static_cast<_type_atom_index>(lround(src_atom.x[1] / p_domain->lattice_const));
-    auto l = static_cast<_type_atom_index>(lround(src_atom.x[2] / p_domain->lattice_const));
+    auto k = static_cast<_type_atom_index>(lround(
+            (src_atom.x[1] - (j % 2) * p_domain->lattice_const / 2) / p_domain->lattice_const));
+    auto l = static_cast<_type_atom_index>(lround(
+            (src_atom.x[2] - (j % 2) * p_domain->lattice_const / 2) / p_domain->lattice_const));
+
     coords[0] = j - 2 * p_domain->lattice_coord_sub_box_region.x_low;
     coords[1] = k - p_domain->lattice_coord_sub_box_region.y_low;
     coords[2] = l - p_domain->lattice_coord_sub_box_region.z_low;
```

B7: Error in computing molecule distance causes the incorrect indexing computation

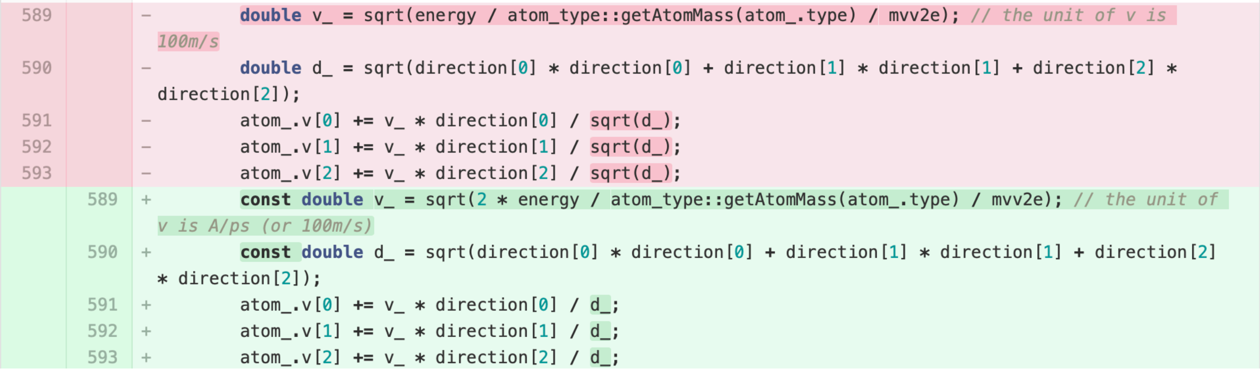
Related code:



```diff
         for (AtomNei::iterator nei_itl = neighbours->begin(false, x, y, z);
              nei_itl != nei_full_itl_end; ++nei_itl) {
-            AtomElement &atom_neighbour_up = *nei_itl; // this is a lattice atom.
-            delx = (*inter_it).x[0] - atom_neighbour_up.x[0];
-            dely = (*inter_it).x[1] - atom_neighbour_up.x[1];
-            delz = (*inter_it).x[2] - atom_neighbour_up.x[2];
+            AtomElement &lattice_neighbour = *nei_itl; // this is a lattice atom.
+            delx = (*inter_it).x[0] - lattice_neighbour.x[0];
+            dely = (*inter_it).x[1] - lattice_neighbour.x[1];
+            delz = (*inter_it).x[2] - lattice_neighbour.x[2];
             dist2 = delx * delx + dely * dely + delz * delz;
-            if (dist2 < (_cutoffRadius * _cutoffRadius) && !atom_neighbour_up.isInterElement()) {
+            if (dist2 < (_cutoffRadius * _cutoffRadius) && !lattice_neighbour.isInterElement()) {
                 // fixme
                 fpair = pot->toForce(
                         atom_type::getTypeIdByType((*inter_it).type),
-                        atom_type::getTypeIdByType(atom_neighbour_up.type),
-                        dist2, (*inter_it).df + atom_neighbour_up.df);
+                        atom_type::getTypeIdByType(lattice_neighbour.type),
+                        dist2, (*inter_it).df + lattice_neighbour.df);

                 (*inter_it).f[0] += delx * fpair;
                 (*inter_it).f[1] += dely * fpair;
                 (*inter_it).f[2] += delz * fpair;

-                atom_neighbour_up.f[0] -= delx * fpair;
-                atom_neighbour_up.f[1] -= dely * fpair;
-                atom_neighbour_up.f[2] -= delz * fpair;
+                lattice_neighbour.f[0] -= delx * fpair;
+                lattice_neighbour.f[1] -= dely * fpair;
+                lattice_neighbour.f[2] -= delz * fpair;
```

B8：Incorrect coding in speed calculation during cascade collision

Related Code:

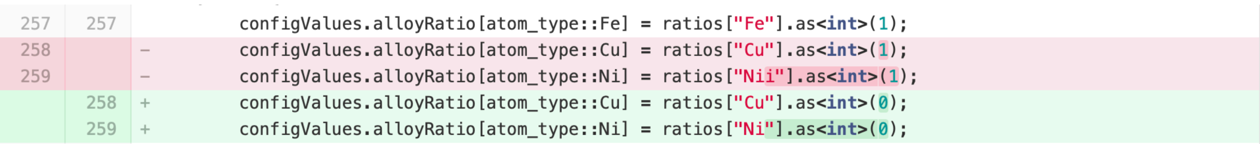


```
-        double v_ = sqrt(energy / atom_type::getAtomMass(atom_.type) / mvv2e); // the unit of v is 100m/s
-        double d_ = sqrt(direction[0] * direction[0] + direction[1] * direction[1] + direction[2] * direction[2]);
-        atom_.v[0] += v_ * direction[0] / sqrt(d_);
-        atom_.v[1] += v_ * direction[1] / sqrt(d_);
-        atom_.v[2] += v_ * direction[2] / sqrt(d_);
+        const double v_ = sqrt(2 * energy / atom_type::getAtomMass(atom_.type) / mvv2e); // the unit of v is A/ps (or 100m/s)
+        const double d_ = sqrt(direction[0] * direction[0] + direction[1] * direction[1] + direction[2] * direction[2]);
+        atom_.v[0] += v_ * direction[0] / d_;
+        atom_.v[1] += v_ * direction[1] / d_;
+        atom_.v[2] += v_ * direction[2] / d_;
```

Reason： Incorrect coding in speed calculation during cascade collision.

B9： Incorrect string constant, where “Ni” was written as “Nii”

Related Code:



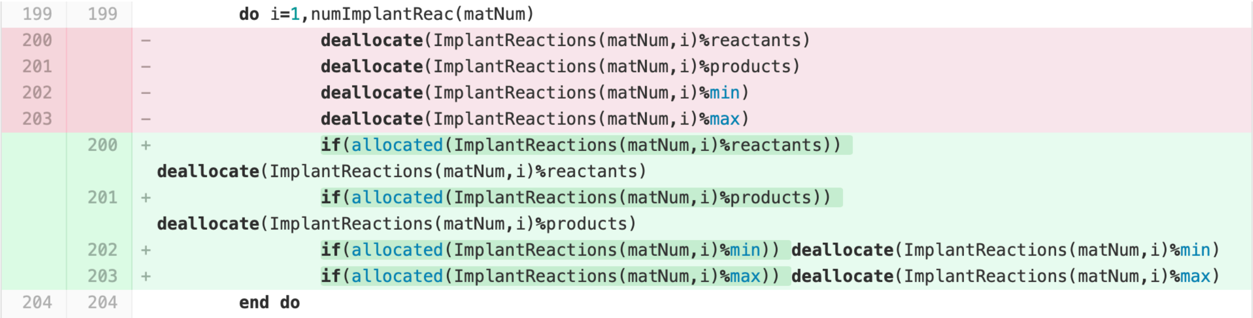
```
         configValues.alloyRatio[atom_type::Fe] = ratios["Fe"].as<int>(1);
-        configValues.alloyRatio[atom_type::Cu] = ratios["Cu"].as<int>(1);
-        configValues.alloyRatio[atom_type::Ni] = ratios["Nii"].as<int>(1);
+        configValues.alloyRatio[atom_type::Cu] = ratios["Cu"].as<int>(0);
+        configValues.alloyRatio[atom_type::Ni] = ratios["Ni"].as<int>(0);
```

Reason： Incorrect string constant, where “Ni” was written as “Nii”.

c) MISA-SCD:

B1： Duplicate insertion of the same data

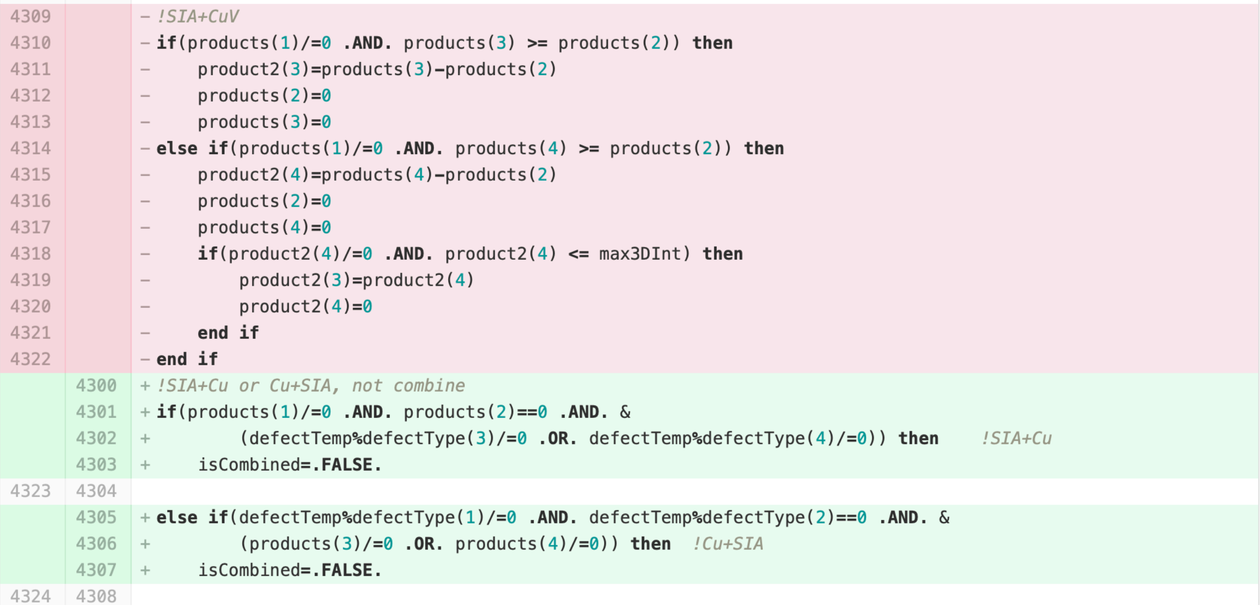
Related Code:



```
         do i=1,numImplantReac(matNum)
-                deallocate(ImplantReactions(matNum,i)%reactants)
-                deallocate(ImplantReactions(matNum,i)%products)
-                deallocate(ImplantReactions(matNum,i)%min)
-                deallocate(ImplantReactions(matNum,i)%max)
+                if(allocated(ImplantReactions(matNum,i)%reactants)) deallocate(ImplantReactions(matNum,i)%reactants)
+                if(allocated(ImplantReactions(matNum,i)%products)) deallocate(ImplantReactions(matNum,i)%products)
+                if(allocated(ImplantReactions(matNum,i)%min)) deallocate(ImplantReactions(matNum,i)%min)
+                if(allocated(ImplantReactions(matNum,i)%max)) deallocate(ImplantReactions(matNum,i)%max)
         end do
```

B2: Incorrect merge of two defect species

Related Code:



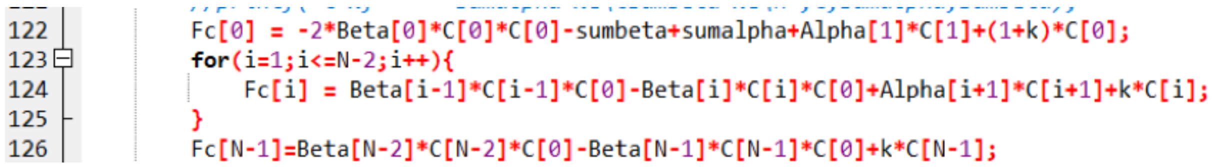
```
- !SIA+CuV
- if(products(1)/=0 .AND. products(3) >= products(2)) then
-     product2(3)=products(3)-products(2)
-     products(2)=0
-     products(3)=0
- else if(products(1)/=0 .AND. products(4) >= products(2)) then
-     product2(4)=products(4)-products(2)
-     products(2)=0
-     products(4)=0
-     if(product2(4)/=0 .AND. product2(4) <= max3DInt) then
-         product2(3)=product2(4)
-         product2(4)=0
-     end if
- end if
+ !SIA+Cu or Cu+SIA, not combine
+ if(products(1)/=0 .AND. products(2)==0 .AND. &
+         (defectTemp%defectType(3)/=0 .OR. defectTemp%defectType(4)/=0)) then    !SIA+Cu
+     isCombined=.FALSE.

+ else if(defectTemp%defectType(1)/=0 .AND. defectTemp%defectType(2)==0 .AND. &
+         (products(3)/=0 .OR. products(4)/=0)) then  !Cu+SIA
+     isCombined=.FALSE.

```

Reason: SIA+Cu should not combine.

d) MISA-ETD

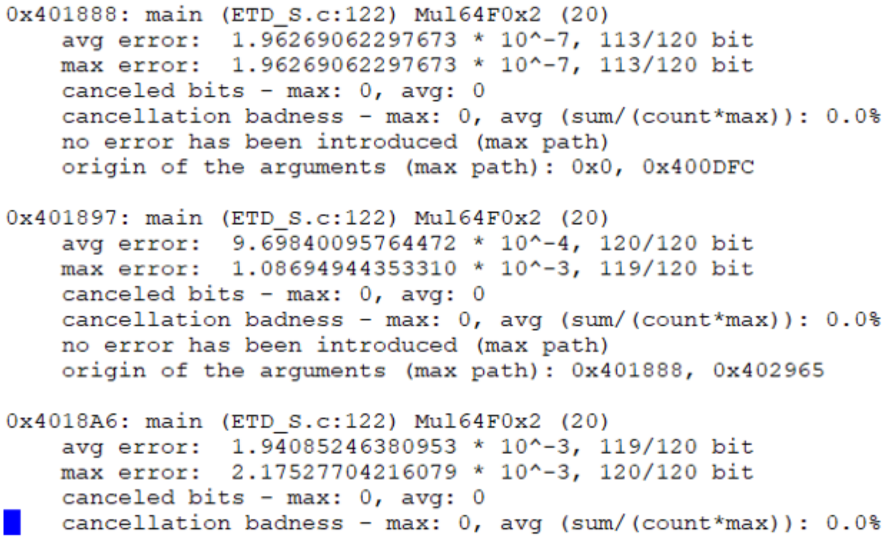
B1: Cancellation

Related Code:



```
122    Fc[0] = -2*Beta[0]*C[0]*C[0]-sumbeta+sumalpha+Alpha[1]*C[1]+(1+k)*C[0];
123    for(i=1;i<=N-2;i++){
124        Fc[i] = Beta[i-1]*C[i-1]*C[0]-Beta[i]*C[i]*C[0]+Alpha[i+1]*C[i+1]+k*C[i];
125    }
126    Fc[N-1]=Beta[N-2]*C[N-2]*C[0]-Beta[N-1]*C[N-1]*C[0]+k*C[N-1];
```

Reason: line 122 may has large cancellation error.



```
0x401888: main (ETD_S.c:122) Mul64F0x2 (20)
    avg error:  1.96269062297673 * 10^-7, 113/120 bit
    max error:  1.96269062297673 * 10^-7, 113/120 bit
    canceled bits - max: 0, avg: 0
    cancellation badness - max: 0, avg (sum/(count*max)): 0.0%
    no error has been introduced (max path)
    origin of the arguments (max path): 0x0, 0x400DFC

0x401897: main (ETD_S.c:122) Mul64F0x2 (20)
    avg error:  9.69840095764472 * 10^-4, 120/120 bit
    max error:  1.08694944353310 * 10^-3, 119/120 bit
    canceled bits - max: 0, avg: 0
    cancellation badness - max: 0, avg (sum/(count*max)): 0.0%
    no error has been introduced (max path)
    origin of the arguments (max path): 0x401888, 0x402965

0x4018A6: main (ETD_S.c:122) Mul64F0x2 (20)
    avg error:  1.94085246380953 * 10^-3, 119/120 bit
    max error:  2.17527704216079 * 10^-3, 120/120 bit
    canceled bits - max: 0, avg: 0
    cancellation badness - max: 0, avg (sum/(count*max)): 0.0%
```