

The information of Test Cases and Bugs

1. Programs

a) ANT-MOC

Input:

Geometry: Geometric input files.

Global Primitives: Supplemental geometry data file.

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.

Polar Spacing: Track spacing of the polar plane.

Global Refines: Type of refines.

Z Mesh: Type of axial network.

Quadrature: Quadrature type.

Solver: Solver type.

Keff Type: Calculating k-eff using neutron balance or not.

Max Iterations: Maximum number of iterations.

Tolerance: Tolerance of convergence.

Segmentation: Type of track segmentation.

Axial Zones: Overlapping plane axial layer.

Output:

keff : Effective value-added factors.

b) MISA-MD:

Input:

[simulation]

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.

[simulation.createphase]

create_phase: Reading atoms from a file or creating atoms.

create_t_set: the initial temperature.

create_seed: random seed.

[simulation.alloy]

create_seed: random seek for creating atoms in Fe-Cu-Ni alloy material

fe: the percentage of Fe.

cu: the percentage of Cu.

ni: the percentage of Ni.

[simulation.collission]

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: y-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.

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-rodde-A.zip
ANT-MOC-DT'02	OpenMOC	test/input/ANT-MOC/DT/c5g7-rodde-B.zip
ANT-MOC-DT'03	OpenMOC	test/input/ANT-MOC/DT/c5g7-unrodde.zip
ANT-MOC-DT'04	OpenMOC	test/input/ANT-MOC/DT/takeda-unrodde.zip
ANT-MOC-DT'05	OpenMOC	test/input/ANT-MOC/CT/takeda-rodde.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/MTGroup-depth/c5g7-depth1.zip
	ANT-MOC-MR2-G1-T02	derived from ANT-MOC-CT02, setting the depth of assembly1 at "level2"	/test/input/ANT-MOC/MT/MTGroup-depth/c5g7-depth2.zip
	ANT-MOC-MR2-G1-T03	derived from ANT-MOC-CT02, setting the depth of assembly1 at "level3"	/test/input/ANT-MOC/MT/MTGroup-depth/c5g7-depth3.zip
ANT-MOC-	ANT-MOC-MR3-	derived from ANT-MOC-CT02, changing the variable of "Num Polar "	test/input/ANT-MOC/MT/MTGroup-polar-c5g7-rodde-

MR3-G1	G1-T01~T08	with "2, 4, 6, ..., 16"	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-rodde-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.toml
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.toml, (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>.toml
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.toml
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.toml,
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.toml

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>/lammmps.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>/lammmps.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>/lammmps.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>/lammmps.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>/lammmps.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>/lammmps.in

Property-based Testing

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

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.toml
	MISA-MD-MR6-G1-T02		test/input/MISA-MD/MT/MTGroup-Ni/test1.toml
	MISA-MD-MR6-G1-T03		test/input/MISA-MD/MT/MTGroup-Ni/test2.toml
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.toml
	MISA-MD-MR6-G1-T01		test/input/MISA-MD/MT/MTGroup-Cu/test1.toml
	MISA-MD-MR6-G1-T01		test/input/MISA-MD/MT/MTGroup-Cu/test2.toml
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-based 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 $\epsilon = 20$	Domain knowledge
P13	$\neg(C_u < 0 \wedge V < 0 \wedge SIA_m < 0 \wedge SIA_im < 0)$	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 i_0 , if follow-up test cases i_1, i_2, \dots are derived by gradually increasing the amount of polar angles, i.e., by densifying the grids, then $keff(p(i_k)) - keff(p(i_{k+1})) < 2 \times 10^{-6}$, when $k \rightarrow +\infty$	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 T_0 , $p(i) = p(i')$	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:

```

627      - /**
628      -  * @brief transform from str to int
629      -  * @details This method can only handle strings in format 'M-N',
630      -  *          e.g. '1-1', '90-289'.
631      -  */
632      - int ReadFromXML::getMatIdFromStr(const char *str) {
633      -      std::ostringstream ss;
634      -
635      -      for (auto c = str; *c; c++) {
636      -          if (*c == '-') {
637      -              ss << 0;          // Substitute '-' with '0', e.g. '1-1' will be '101'
638      -              continue;
639      -          }
640      -          if (!std::isdigit(*c)) // Handle non-digit chars
641      -              return -1;
642      -          ss << *c;
643      -      }
644      -
645      -      return stoi(ss.str());
646      - }

```

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

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

```

2893 2897      long TrackGenerator3D::getMyNum3DTracks() {
2894      -
2895      -      int a = _num_azim/2 - 1;
2896      -      long xy = getMyNum2DTracks(a) - 1;
2897      -      int p = _num_polar - 1;
2898      -      return _my_cum_tracks_per_stack[a][xy][p] + _my_tracks_per_stack[a][xy][p];
2899      +      return _my_num_3D_tracks;
2899 2899      }

```

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,
1293 1293      double y2 = track->getStart()->getY();
1294 1294      double z2 = track->getStart()->getZ();
1295 1295
1296      +      /* Get the first 2D track associated with this 3D track */
1297      +      Track* track_2D = _tracks_2D_chains[tci->_azim][tci->_x][link];
1298      +
1299      +      /* Set the start and end point for each 3D track,
1300      +      * or just increment arrays */
1301      -      while (!end_of_chain) {
1302      -          /* Get the 2D track associated with this 3D track */
1303      -          Track* track_2D = _tracks_2D_chains[tci->_azim][tci->_x][link];
1304      +      while (true) {
1305      +          double phi = track_2D->getPhi();
1306      +          int azim = track_2D->getAzimIndex();
1307      +          @@ -1398,15 +1398,20 @@ void TrackGenerator3D::set3DTrackData(TrackChainIndexes* tci,
1398 1398      /* Increment the link index */
1399 1399      link++;
1400 1400

```

```

1401 -      /* Move the starting x-coord to account for periodic BCs for chains
1402 -      * In the case of reflective BCs, the starting point is just the end
1403 -      * point of the previous link */
1404 -      if (!end_of_chain) {
1401 +      if (end_of_chain) {
1402 +          break;
1403 +      }
1404 +      else {
1405 +          // Move to the next 2D track
1406 +          track_2D = _tracks_2D_chains[tci->_azim][tci->_x][link];
1407 +
1408 +          /* Move the starting x-coord to account for periodic BCs for chains
1409 +          * In the case of reflective BCs, the starting point is just the end
1410 +          * point of the previous link */
1405 1411      if (!isReflectiveCyclic()) {
1406 -          if (tci->_azim < _num_azim / 4)
1407 -              x2 = _x_min;
1408 -          else
1409 -              x2 = _x_max;
1412 +          Point *next_start = track_2D->getStart();
1413 +          x2 = next_start->getX();
1414 +          y2 = next_start->getY();
1410 1415      }
1411 1416      }
1412 1417      }

```

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

B4: The generation of FSR ID is not unique

Related Code:

```

1012 1012      /* Try to get a clean copy of the fsr_id, adding the FSR data
1013 1013      if necessary where -1 indicates the key was already added */
1014 -      fsr_id = _FSR_keys_map.insert_and_get_count(fsr_key, NULL);
1014 +      if (FSRonly) {
1015 +          #pragma omp ordered
1016 +          fsr_id = _FSR_keys_map.insert_and_get_count(fsr_key, NULL);
1017 +      }
1018 +      else {
1019 +          fsr_id = _FSR_keys_map.insert_and_get_count(fsr_key, NULL);
1020 +      }
1021 +
1015 1022      if (fsr_id == -1) {
1016 1023          fsr_data volatile* fsr;
1017 1024          do {
1018 1025              ...
1019 1026              ... @@ -1730,7 +1737,7 @@ void Geometry::initializeFlatSourceRegions() {
1020 1027                  * @param z_coord the axial height at which the 2D plane of the geometry is
1021 1028                  * formed
1022 1029                  */
1023 1030          } while (fsr_id == -1);
1024 1031          fsr_data = fsr;
1025 1032          fsr_id = fsr->id;
1026 1033          fsr->id = fsr_id;
1027 1034          fsr->id = fsr_id;
1028 1035          fsr->id = fsr_id;
1029 1036          fsr->id = fsr_id;
1030 1037          fsr->id = fsr_id;
1031 1038          fsr->id = fsr_id;
1032 1039          fsr->id = fsr_id;
1033 1040          fsr->id = fsr_id;
1034 1041          fsr->id = fsr_id;
1035 1042          fsr->id = fsr_id;
1036 1043          fsr->id = fsr_id;
1037 1044          fsr->id = fsr_id;
1038 1045          fsr->id = fsr_id;
1039 1046          fsr->id = fsr_id;
1040 1047          fsr->id = fsr_id;
1041 1048          fsr->id = fsr_id;
1042 1049          fsr->id = fsr_id;
1043 1050          fsr->id = fsr_id;
1044 1051          fsr->id = fsr_id;
1045 1052          fsr->id = fsr_id;
1046 1053          fsr->id = fsr_id;
1047 1054          fsr->id = fsr_id;
1048 1055          fsr->id = fsr_id;
1049 1056          fsr->id = fsr_id;
1050 1057          fsr->id = fsr_id;
1051 1058          fsr->id = fsr_id;
1052 1059          fsr->id = fsr_id;
1053 1060          fsr->id = fsr_id;
1054 1061          fsr->id = fsr_id;
1055 1062          fsr->id = fsr_id;
1056 1063          fsr->id = fsr_id;
1057 1064          fsr->id = fsr_id;
1058 1065          fsr->id = fsr_id;
1059 1066          fsr->id = fsr_id;
1060 1067          fsr->id = fsr_id;
1061 1068          fsr->id = fsr_id;
1062 1069          fsr->id = fsr_id;
1063 1070          fsr->id = fsr_id;
1064 1071          fsr->id = fsr_id;
1065 1072          fsr->id = fsr_id;
1066 1073          fsr->id = fsr_id;
1067 1074          fsr->id = fsr_id;
1068 1075          fsr->id = fsr_id;
1069 1076          fsr->id = fsr_id;
1070 1077          fsr->id = fsr_id;
1071 1078          fsr->id = fsr_id;
1072 1079          fsr->id = fsr_id;
1073 1080          fsr->id = fsr_id;
1074 1081          fsr->id = fsr_id;
1075 1082          fsr->id = fsr_id;
1076 1083          fsr->id = fsr_id;
1077 1084          fsr->id = fsr_id;
1078 1085          fsr->id = fsr_id;
1079 1086          fsr->id = fsr_id;
1080 1087          fsr->id = fsr_id;
1081 1088          fsr->id = fsr_id;
1082 1089          fsr->id = fsr_id;
1083 1090          fsr->id = fsr_id;
1084 1091          fsr->id = fsr_id;
1085 1092          fsr->id = fsr_id;
1086 1093          fsr->id = fsr_id;
1087 1094          fsr->id = fsr_id;
1088 1095          fsr->id = fsr_id;
1089 1096          fsr->id = fsr_id;
1090 1097          fsr->id = fsr_id;
1091 1098          fsr->id = fsr_id;
1092 1099          fsr->id = fsr_id;
1093 1100          fsr->id = fsr_id;
1094 1101          fsr->id = fsr_id;
1095 1102          fsr->id = fsr_id;
1096 1103          fsr->id = fsr_id;
1097 1104          fsr->id = fsr_id;
1098 1105          fsr->id = fsr_id;
1099 1106          fsr->id = fsr_id;
1100 1107          fsr->id = fsr_id;
1101 1108          fsr->id = fsr_id;
1102 1109          fsr->id = fsr_id;
1103 1110          fsr->id = fsr_id;
1104 1111          fsr->id = fsr_id;
1105 1112          fsr->id = fsr_id;
1106 1113          fsr->id = fsr_id;
1107 1114          fsr->id = fsr_id;
1108 1115          fsr->id = fsr_id;
1109 1116          fsr->id = fsr_id;
1110 1117          fsr->id = fsr_id;
1111 1118          fsr->id = fsr_id;
1112 1119          fsr->id = fsr_id;
1113 1120          fsr->id = fsr_id;
1114 1121          fsr->id = fsr_id;
1115 1122          fsr->id = fsr_id;
1116 1123          fsr->id = fsr_id;
1117 1124          fsr->id = fsr_id;
1118 1125          fsr->id = fsr_id;
1119 1126          fsr->id = fsr_id;
1120 1127          fsr->id = fsr_id;
1121 1128          fsr->id = fsr_id;
1122 1129          fsr->id = fsr_id;
1123 1130          fsr->id = fsr_id;
1124 1131          fsr->id = fsr_id;
1125 1132          fsr->id = fsr_id;
1126 1133          fsr->id = fsr_id;
1127 1134          fsr->id = fsr_id;
1128 1135          fsr->id = fsr_id;
1129 1136          fsr->id = fsr_id;
1130 1137          fsr->id = fsr_id;
1131 1138          fsr->id = fsr_id;
1132 1139          fsr->id = fsr_id;
1133 1140          fsr->id = fsr_id;
1134 1141          fsr->id = fsr_id;
1135 1142          fsr->id = fsr_id;
1136 1143          fsr->id = fsr_id;
1137 1144          fsr->id = fsr_id;
1138 1145          fsr->id = fsr_id;
1139 1146          fsr->id = fsr_id;
1140 1147          fsr->id = fsr_id;
1141 1148          fsr->id = fsr_id;
1142 1149          fsr->id = fsr_id;
1143 1150          fsr->id = fsr_id;
1144 1151          fsr->id = fsr_id;
1145 1152          fsr->id = fsr_id;
1146 1153          fsr->id = fsr_id;
1147 1154          fsr->id = fsr_id;
1148 1155          fsr->id = fsr_id;
1149 1156          fsr->id = fsr_id;
1150 1157          fsr->id = fsr_id;
1151 1158          fsr->id = fsr_id;
1152 1159          fsr->id = fsr_id;
1153 1160          fsr->id = fsr_id;
1154 1161          fsr->id = fsr_id;
1155 1162          fsr->id = fsr_id;
1156 1163          fsr->id = fsr_id;
1157 1164          fsr->id = fsr_id;
1158 1165          fsr->id = fsr_id;
1159 1166          fsr->id = fsr_id;
1160 1167          fsr->id = fsr_id;
1161 1168          fsr->id = fsr_id;
1162 1169          fsr->id = fsr_id;
1163 1170          fsr->id = fsr_id;
1164 1171          fsr->id = fsr_id;
1165 1172          fsr->id = fsr_id;
1166 1173          fsr->id = fsr_id;
1167 1174          fsr->id = fsr_id;
1168 1175          fsr->id = fsr_id;
1169 1176          fsr->id = fsr_id;
1170 1177          fsr->id = fsr_id;
1171 1178          fsr->id = fsr_id;
1172 1179          fsr->id = fsr_id;
1173 1180          fsr->id = fsr_id;
1174 1181          fsr->id = fsr_id;
1175 1182          fsr->id = fsr_id;
1176 1183          fsr->id = fsr_id;
1177 1184          fsr->id = fsr_id;
1178 1185          fsr->id = fsr_id;
1179 1186          fsr->id = fsr_id;
1180 1187          fsr->id = fsr_id;
1181 1188          fsr->id = fsr_id;
1182 1189          fsr->id = fsr_id;
1183 1190          fsr->id = fsr_id;
1184 1191          fsr->id = fsr_id;
1185 1192          fsr->id = fsr_id;
1186 1193          fsr->id = fsr_id;
1187 1194          fsr->id = fsr_id;
1188 1195          fsr->id = fsr_id;
1189 1196          fsr->id = fsr_id;
1190 1197          fsr->id = fsr_id;
1191 1198          fsr->id = fsr_id;
1192 1199          fsr->id = fsr_id;
1193 1200          fsr->id = fsr_id;
1194 1201          fsr->id = fsr_id;
1195 1202          fsr->id = fsr_id;
1196 1203          fsr->id = fsr_id;
1197 1204          fsr->id = fsr_id;
1198 1205          fsr->id = fsr_id;
1199 1206          fsr->id = fsr_id;
1200 1207          fsr->id = fsr_id;
1201 1208          fsr->id = fsr_id;
1202 1209          fsr->id = fsr_id;
1203 1210          fsr->id = fsr_id;
1204 1211          fsr->id = fsr_id;
1205 1212          fsr->id = fsr_id;
1206 1213          fsr->id = fsr_id;
1207 1214          fsr->id = fsr_id;
1208 1215          fsr->id = fsr_id;
1209 1216          fsr->id = fsr_id;
1210 1217          fsr->id = fsr_id;
1211 1218          fsr->id = fsr_id;
1212 1219          fsr->id = fsr_id;
1213 1220          fsr->id = fsr_id;
1214 1221          fsr->id = fsr_id;
1215 1222          fsr->id = fsr_id;
1216 1223          fsr->id = fsr_id;
1217 1224          fsr->id = fsr_id;
1218 1225          fsr->id = fsr_id;
1219 1226          fsr->id = fsr_id;
1220 1227          fsr->id = fsr_id;
1221 1228          fsr->id = fsr_id;
1222 1229          fsr->id = fsr_id;
1223 1230          fsr->id = fsr_id;
1224 1231          fsr->id = fsr_id;
1225 1232          fsr->id = fsr_id;
1226 1233          fsr->id = fsr_id;
1227 1234          fsr->id = fsr_id;
1228 1235          fsr->id = fsr_id;
1229 1236          fsr->id = fsr_id;
1230 1237          fsr->id = fsr_id;
1231 1238          fsr->id = fsr_id;
1232 1239          fsr->id = fsr_id;
1233 1240          fsr->id = fsr_id;
1234 1241          fsr->id = fsr_id;
1235 1242          fsr->id = fsr_id;
1236 1243          fsr->id = fsr_id;
1237 1244          fsr->id = fsr_id;
1238 1245          fsr->id = fsr_id;
1239 1246          fsr->id = fsr_id;
1240 1247          fsr->id = fsr_id;
1241 1248          fsr->id = fsr_id;
1242 1249          fsr->id = fsr_id;
1243 1250          fsr->id = fsr_id;
1244 1251          fsr->id = fsr_id;
1245 1252          fsr->id = fsr_id;
1246 1253          fsr->id = fsr_id;
1247 1254          fsr->id = fsr_id;
1248 1255          fsr->id = fsr_id;
1249 1256          fsr->id = fsr_id;
1250 1257          fsr->id = fsr_id;
1251 1258          fsr->id = fsr_id;
1252 1259          fsr->id = fsr_id;
1253 1260          fsr->id = fsr_id;
1254 1261          fsr->id = fsr_id;
1255 1262          fsr->id = fsr_id;
1256 1263          fsr->id = fsr_id;
1257 1264          fsr->id = fsr_id;
1258 1265          fsr->id = fsr_id;
1259 1266          fsr->id = fsr_id;
1260 1267          fsr->id = fsr_id;
1261 1268          fsr->id = fsr_id;
1262 1269          fsr->id = fsr_id;
1263 1270          fsr->id = fsr_id;
1264 1271          fsr->id = fsr_id;
1265 1272          fsr->id = fsr_id;
1266 1273          fsr->id = fsr_id;
1267 1274          fsr->id = fsr_id;
1268 1275          fsr->id = fsr_id;
1269 1276          fsr->id = fsr_id;
1270 1277          fsr->id = fsr_id;
1271 1278          fsr->id = fsr_id;
1272 1279          fsr->id = fsr_id;
1273 1280          fsr->id = fsr_id;
1274 1281          fsr->id = fsr_id;
1275 1282          fsr->id = fsr_id;
1276 1283          fsr->id = fsr_id;
1277 1284          fsr->id = fsr_id;
1278 1285          fsr->id = fsr_id;
1279 1286          fsr->id = fsr_id;
1280 1287          fsr->id = fsr_id;
1281 1288          fsr->id = fsr_id;
1282 1289          fsr->id = fsr_id;
1283 1290          fsr->id = fsr_id;
1284 1291          fsr->id = fsr_id;
1285 1292          fsr->id = fsr_id;
1286 1293          fsr->id = fsr_id;
1287 1294          fsr->id = fsr_id;
1288 1295          fsr->id = fsr_id;
1289 1296          fsr->id = fsr_id;
1290 1297          fsr->id = fsr_id;
1291 1298          fsr->id = fsr_id;
1292 1299          fsr->id = fsr_id;
1293 1300          fsr->id = fsr_id;
1294 1301          fsr->id = fsr_id;
1295 1302          fsr->id = fsr_id;
1296 1303          fsr->id = fsr_id;
1297 1304          fsr->id = fsr_id;
1298 1305          fsr->id = fsr_id;
1299 1306          fsr->id = fsr_id;
1300 1307          fsr->id = fsr_id;
1301 1308          fsr->id = fsr_id;
1302 1309          fsr->id = fsr_id;
1303 1310          fsr->id = fsr_id;
1304 1311          fsr->id = fsr_id;
1305 1312          fsr->id = fsr_id;
1306 1313          fsr->id = fsr_id;
1307 1314          fsr->id = fsr_id;
1308 1315          fsr->id = fsr_id;
1309 1316          fsr->id = fsr_id;
1310 1317          fsr->id = fsr_id;
1311 1318          fsr->id = fsr_id;
1312 1319          fsr->id = fsr_id;
1313 1320          fsr->id = fsr_id;
1314 1321          fsr->id = fsr_id;
1315 1322          fsr->id = fsr_id;
1316 1323          fsr->id = fsr_id;
1317 1324          fsr->id = fsr_id;
1318 1325          fsr->id = fsr_id;
1319 1326          fsr->id = fsr_id;
1320 1327          fsr->id = fsr_id;
1321 1328          fsr->id = fsr_id;
1322 1329          fsr->id = fsr_id;
1323 1330          fsr->id = fsr_id;
1324 1331          fsr->id = fsr_id;
1325 1332          fsr->id = fsr_id;
1326 1333          fsr->id = fsr_id;
1327 1334          fsr->id = fsr_id;
1328 1335          fsr->id = fsr_id;
1329 1336          fsr->id = fsr_id;
1330 1337          fsr->id = fsr_id;
1331 1338          fsr->id = fsr_id;
1332 1339          fsr->id = fsr_id;
1333 1340          fsr->id = fsr_id;
1334 1341          fsr->id = fsr_id;
1335 1342          fsr->id = fsr_id;
1336 1343          fsr->id = fsr_id;
1337 1344          fsr->id = fsr_id;
1338 1345          fsr->id = fsr_id;
1339 1346          fsr->id = fsr_id;
1340 1347          fsr->id = fsr_id;
1341 1348          fsr->id = fsr_id;
1342 1349          fsr->id = fsr_id;
1343 1350          fsr->id = fsr_id;
1344 1351          fsr->id = fsr_id;
1345 1352          fsr->id = fsr_id;
1346 1353          fsr->id = fsr_id;
1347 1354          fsr->id = fsr_id;
1348 1355          fsr->id = fsr_id;
1349 1356          fsr->id = fsr_id;
1350 1357          fsr->id = fsr_id;
1351 1358          fsr->id = fsr_id;
1352 1359          fsr->id = fsr_id;
1353 1360          fsr->id = fsr_id;
1354 1361          fsr->id = fsr_id;
1355 1362          fsr->id = fsr_id;
1356 1363          fsr->id = fsr_id;
1357 1364          fsr->id = fsr_id;
1358 1365          fsr->id = fsr_id;
1359 1366          fsr->id = fsr_id;
1360 1367          fsr->id = fsr_id;
1361 1368          fsr->id = fsr_id;
1362 1369          fsr->id = fsr_id;
1363 1370          fsr->id = fsr_id;
1364 1371          fsr->id = fsr_id;
1365 1372          fsr->id = fsr_id;
1366 1373          fsr->id = fsr_id;
1367 1374          fsr->id = fsr_id;
1368 1375          fsr->id = fsr_id;
1369 1376          fsr->id = fsr_id;
1370 1377          fsr->id = fsr_id;
1371 1378          fsr->id = fsr_id;
1372 1379          fsr->id = fsr_id;
1373 1380          fsr->id = fsr_id;
1374 1381          fsr->id = fsr_id;
1375 1382          fsr->id = fsr_id;
1376 1383          fsr->id = fsr_id;
1377 1384          fsr->id = fsr_id;
1378 1385          fsr->id = fsr_id;
1379 1386          fsr->id = fsr_id;
1380 1387          fsr->id = fsr_id;
1381 1388          fsr->id = fsr_id;
1382 1389          fsr->id = fsr_id;
1383 1390          fsr->id = fsr_id;
1384 1391          fsr->id = fsr_id;
1385 1392          fsr->id = fsr_id;
1386 1393          fsr->id = fsr_id;
1387 1394          fsr->id = fsr_id;
1388 1395          fsr->id = fsr_id;
1389 1396          fsr->id = fsr_id;
1390 1397          fsr->id = fsr_id;
1391 1398          fsr->id = fsr_id;
1392 1399          fsr->id = fsr_id;
1393 1400          fsr->id = fsr_id;
1394 1401          fsr->id = fsr_id;
1395 1402          fsr->id = fsr_id;
1396 1403          fsr->id = fsr_id;
1397 1404          fsr->id = fsr_id;
1398 1405          fsr->id = fsr_id;
1399 1406          fsr->id = fsr_id;
1400 1407          fsr->id = fsr_id;
1401 1408          fsr->id = fsr_id;
1402 1409          fsr->id = fsr_id;
1403 1410          fsr->id = fsr_id;
1404 1411          fsr->id = fsr_id;
1405 1412          fsr->id = fsr_id;
1406 1413          fsr->id = fsr_id;
1407 1414          fsr->id = fsr_id;
1408 1415          fsr->id = fsr_id;
1409 1416          fsr->id = fsr_id;
1410 1417          fsr->id = fsr_id;
1411 1418          fsr->id = fsr_id;
1412 1419          fsr->id = fsr_id;
1413 1420          fsr->id = fsr_id;
1414 1421          fsr->id = fsr_id;
1415 1422          fsr->id = fsr_id;
1416 1423          fsr->id = fsr_id;
1417 1424          fsr->id = fsr_id;
1418 1425          fsr->id = fsr_id;
1419 1426          fsr->id = fsr_id;
1420 1427          fsr->id = fsr_id;
1421 1428          fsr->id = fsr_id;
1422 1429          fsr->id = fsr_id;
1423 1430          fsr->id = fsr_id;
1424 1431          fsr->id = fsr_id;
1425 1432          fsr->id = fsr_id;
1426 1433          fsr->id = fsr_id;
1427 1434          fsr->id = fsr_id;
1428 1435          fsr->id = fsr_id;
1429 1436          fsr->id = fsr_id;
1430 1437          fsr->id = fsr_id;
1431 1438          fsr->id = fsr_id;
1432 1439          fsr->id = fsr_id;
1433 1440          fsr->id = fsr_id;
1434 1441          fsr->id = fsr_id;
1435 1442          fsr->id = fsr_id;
1436 1443          fsr->id = fsr_id;
1437 1444          fsr->id = fsr_id;
1438 1445          fsr->id = fsr_id;
1439 1446          fsr->id = fsr_id;
1440 1447          fsr->id = fsr_id;
1441 1448          fsr->id = fsr_id;
1442 1449          fsr->id = fsr_id;
1443 1450          fsr->id = fsr_id;
1444 1451          fsr->id = fsr_id;
1445 1452          fsr->id = fsr_id;
1446 1453          fsr->id = fsr_id;
1447 1454          fsr->id = fsr_id;
1448 1455          fsr->id = fsr_id;
1449 1456          fsr->id = fsr_id;
1450 1457          fsr->id = fsr_id;
1451 1458          fsr->id = fsr_id;
1452 1459          fsr->id = fsr_id;
1453 1460          fsr->id = fsr_id;
1454 1461          fsr->id = fsr_id;
1455 1462          fsr->id = fsr_id;
1456 1463          fsr->id = fsr_id;
1457 1464          fsr->id = fsr_id;
1458 1465          fsr->id = fsr_id;
1459 1466          fsr->id = fsr_id;
1460 1467          fsr->id = fsr_id;
1461 1468          fsr->id = fsr_id;
1462 1469          fsr->id = fsr_id;
1463 1470          fsr->id = fsr_id;
1464 1471          fsr->id = fsr_id;
1465 1472          fsr->id = fsr_id;
1466 1473          fsr->id = fsr_id;
1467 1474          fsr->id = fsr_id;
1468 1475          fsr->id = fsr_id;
1469 1476          fsr->id = fsr_id;
1470 1477          fsr->id = fsr_id;
1471 1478          fsr->id = fsr_id;
1472 1479          fsr->id = fsr_id;
1473 1480          fsr->id = fsr_id;
1474 1481          fsr->id = fsr_id;
1475 1482          fsr->id = fsr_id;
1476 1483          fsr->id = fsr_id;
1477 1484          fsr->id = fsr_id;
1478 1485          fsr->id = fsr_id;
1479 1486          fsr->id = fsr_id;
1480 1487          fsr->id = fsr_id;
1481 1488          fsr->id = fsr_id;
1482 1489          fsr->id = fsr_id;
1483 1490          fsr->id = fsr_id;
1484 1491          fsr->id = fsr_id;
1485 1492          fsr->id = fsr_id;
1486 1493          fsr->id = fsr_id;
1487 1494          fsr->id = fsr_id;
1488 1495          fsr->id = fsr_id;
1489 1496          fsr->id = fsr_id;
1490 1497          fsr->id = fsr_id;
1491 1498          fsr->id = fsr_id;
1492 1499          fsr->id = fsr_id;
1493 1500          fsr->id = fsr_id;
1494 1501          fsr->id = fsr_id;
1495 1502          fsr->id = fsr_id;
1496 1503          fsr->id = fsr_id;
1497 1504          fsr->id = fsr_id;
1498 1505          fsr->id = fsr_id;
1499 1506          fsr->id = fsr_id;
1500 1507          fsr->id = fsr_id;
1501 1508          fsr->id = fsr_id;
1502 1509          fsr->id = fsr_id;
1503 1510          fsr->id = fsr_id;
1504 1511          fsr->id = fsr_id;
1505 1512          fsr->id = fsr_id;
1506 1513          fsr->id = fsr_id;
1507 1514          fsr->id = fsr_id;
1508 1515          fsr->id = fsr_id;
1509 1516          fsr->id = fsr_id;
1510 1517          fsr->id = fsr_id;
1511 1518          fsr->id = fsr_id;
1512 1519          fsr->id = fsr_id;
1513 1520          fsr->id = fsr_id;
1514 1521          fsr->id = fsr_id;
1515 1522          fsr->id = fsr_id;
1516 1523          fsr->id = fsr_id;
1517 1524          fsr->id = fsr_id;
1518 1525          fsr->id = fsr_id;
1519 1526          fsr->id = fsr_id;
1520 1527          fsr->id = fsr_id;
1521 1528          fsr->id = fsr_id;
1522 1529          fsr->id = fsr_id;
1523 1530          fsr->id = fsr_id;
1524 1531          fsr->id = fsr_id;
1525 1532          fsr->id = fsr_id;
1526 1533          fsr->id = fsr_id;
1527 1534          fsr->id = fsr_id;
1528 1535          fsr->id = fsr_id;
1529 1536          fsr->id = fsr_id;
1530 1537          fsr->id = fsr_id;
1531 1538          fsr->id = fsr_id;
1532 1539          fsr->id = fsr_id;
1533 1540          fsr->id = fsr_id;
1534 1541          fsr->id = fsr_id;
1535 1542          fsr->id = fsr_id;
1536 1543          fsr->id = fsr_id;
1537 1544          fsr->id = fsr_id;
1538 1545          fsr->id = fsr_id;
1539 1546          fsr->id = fsr_id;
1540 1547          fsr->id = fsr_id;
1541 1548          fsr->id =
```

2348	-	/* SURFACE_Y_MIN */
2353	+	/* SURFACE_Y_MIN (the leftmost surface in l-z plane) */
2349	2354	else if (tci->_link == 0 && lz >= nl && !outgoing) {
2350	2355	
2351	-	tci_prdc._x = tci->_x;
2352	2356	tci_prdc._lz = lz - nl;
2353	2357	tci_prdc._link = getNum3DTrackChainLinks(&tci_prdc) - 1;
2354	2358	
2355	-	tci_refl._x = tci->_x;
2356	-	//tci_refl._azim = ac;
2357	-	tci_refl._lz = lz - nl;
2359	+	tci_refl._lz = tci_prdc._lz;
2360	+	tci_refl._link = tci_prdc._link;
2358	2361	
2359	-	tci_next._lz = lz - nl;
2362	+	tci_next._lz = tci_prdc._lz;
2363	+	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:

2426	-	int start_out;
2426	+	int start_out = _fluxes_per_track * !direction;
2427	2427	
2428	2428	/* For the "forward" direction */
2429	2429	if (direction) {
2430	2430	bc_out = track->getBCFwd();
2431	2431	track_out_id = track->getTrackNextFwd();
2432	2432	//start_out = _fluxes_per_track * (!track->getNextFwdFwd());
2433	-	start_out = 0;
2434	2433	}
2435	2434	
2436	2435	/* For the "reverse" direction */
...	...	@@ -2438,7 +2437,6 @@ void CPUSolver::transferBoundaryFlux(Track* track,
2438	2437	bc_out = track->getBCBwd();
2439	2438	track_out_id = track->getTrackNextBwd();
2440	2439	//start_out = _fluxes_per_track * (!track->getNextBwdFwd());
2441	-	start_out = _fluxes_per_track;
2442	2440	}

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:

27	27	if (length < 16) {
28	28	
29	-	#ifdef INTEL
30	-	#pragma simd reduction(+:sum)
31	-	#endif
29	+	#pragma omp simd reduction(+:sum)
32	30	for (L i=0; i < length; i++)
33	31	sum += vector[i];
34	32	}

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

B9: Error in load balance algorithm

Related Code:

```
4648 - // Re-construct all of the ExtrudedFSRs
4649 - if (rank) {
4650 -     for (long i = 0; i < num_points; ++i)
4651 -         merge_extruded_fsr(sendrecv_buf, i);
4652 - }
4648 + // Clear ExtrudedFSRs and re-construct them to make them consistent
4649 + // among all of the processes
4650 + fsr_map.clear();
4651 + MPI_Wait(&handle, &status);
4652 + for (long i = 0; i < num_points; ++i)
4653 +     merge_extruded_fsr(sendrecv_buf, i);
4653 4654
```

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 :

```
103 103 void WorldBuilder::createPhaseSpace() {
104 -     unsigned long id_pre = (unsigned long) box_x * box_y * _p_domain-
>getGlobalSubBoxLatticeCoordLower(2)
104 +     unsigned long id_pre = (unsigned long) box_x * box_y * _p_domain-
>getGlobalSubBoxLatticeCoordLower(2) * 2
105 105         + (unsigned long) _p_domain->getGlobalSubBoxLatticeCoordLower(1) *
106 -         box_x * _p_domain->getSubBoxLatticeSize(2)
106 +         box_x * _p_domain->getSubBoxLatticeSize(2) * 2
107 107         + (unsigned long) _p_domain->getGlobalSubBoxLatticeCoordLower(0) *
108 -         _p_domain->getSubBoxLatticeSize(1) * _p_domain->getSubBoxLatticeSize(2);
108 +         _p_domain->getSubBoxLatticeSize(1) *
109 +         _p_domain->getSubBoxLatticeSize(2);
```

Reason: Error in Computation-Parallelization.

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

Related Code:

146	-	<code>void InterAtomList::unpackExInterRecv(int d, int n, const double *lower, const double *upper, particledata *buf) {</code>
147	-	<code>AtomElement atom;</code>
145	+	<code>void</code>
146	+	<code>InterAtomList::unpackExInterRecv(Domain *p_domain, particledata *buf, std::list<AtomElement></code>
147	+	<code>&delay_buffer, int n) {</code>
148	+	<code>AtomElement atom{};</code>
148	+	<code>for (int i = 0; i < n; i++) {</code>
149	+	<code>atom.id = buf[i].id;</code>
150	+	<code>atom.type = buf[i].type;</code>
...	...	<code>@@ -154,11 +154,12 @@ void InterAtomList::unpackExInterRecv(int d, int n, const double *lower, const d</code>
154	+	<code>atom.v[0] = buf[i].v[0];</code>
155	+	<code>atom.v[1] = buf[i].v[1];</code>
156	+	<code>atom.v[2] = buf[i].v[2];</code>
157	-	<code>if (atom.x[d] >= lower[d] && atom.x[d] < upper[d]) {</code>
158	-	<code>inter_list.push_back(atom);</code>
159	-	<code>nlocalinter++;</code>
157	+	<code>// todo condition: we can judge only one direction the inter atom comes from.</code>
158	+	<code>if (ws::isOutBox(atom, p_domain) == box::IN_BOX) {</code>
159	+	<code>addInterAtom(atom);</code>
160	+	<code>} else {</code>
161	-	<code>// todo waring</code>
161	+	<code>kiwi::logs::w("unpack", "unexpected atom, id: {}\n", atom.id);</code>
162	+	<code>delay_buffer.push_back(atom);</code>
162	163	<code>}</code>
163	164	<code>}</code>

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:

17	20	<code>const unsigned long InterBorderPacker::sendLength(const int dimension, const int direction) {</code>
18	21	<code>const int index = 2 * dimension + direction;</code>
19	-	<code>getInterToSend(&domain, dimension, direction,</code>
20	-	<code>domain.meas_ghost_length[dimension],</code>
21	-	<code>inter_atom_list.intersendlist[index]);</code>
22	-	<code>return inter_atom_list.intersendlist[index].size();</code>
22	+	<code>std::vector<AtomElement *> &sendlist = inter_atom_list.intersendlist[index];</code>
23	+	<code>// before x dimension communication, ghost list is empty.</code>
24	+	<code>comm::Region<comm::type_lattice_size> region = comm::fwCommLocalRegion(&domain, dimension,</code>
25	+	<code>direction);</code>
26	+	<code>_type_atom_index coords[DIMENSION] = {0, 0, 0};</code>
27	+	<code>for (AtomElement &inter_ref : inter_atom_list.inter_list) {</code>
28	+	<code>// get the lattice coordinate the inter atom belongs to.</code>
29	+	<code>ws::getNearLatCoord(inter_ref, &domain, coords);</code>
30	+	<code>if (region.isIn(coords[0], coords[1], coords[2])) {</code>
31	+	<code>sendlist.push_back(&inter_ref);</code>
32	+	<code>}</code>
33	+	<code>}</code>
34	+	<code>for (AtomElement &ghost_ref : inter_atom_list.inter_ghost_list) {</code>
35	+	<code>ws::getNearLatCoord(ghost_ref, &domain, coords);</code>
36	+	<code>if (region.isIn(coords[0], coords[1], coords[2])) {</code>
37	+	<code>sendlist.push_back(&ghost_ref);</code>
38	+	<code>}</code>
39	+	<code>}</code>
40	+	<code>return sendlist.size();</code>
23	41	<code>}</code>

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

B5: Incorrect communication algorithm in ghost regions

Related Code:

Adding new GhostAtom operation.

99	99		// remove this atom from inter list.
100		-	inter_it = inter_atom_list->inter_list.erase(inter_it);
101		-	inter_atom_list->nlocalinter--;
	100	+	inter_it = inter_atom_list->removeInter(inter_it);

Reason: Incorrect communication algorithm in ghost regions.

To fix this bug, we should remove ghost inter atoms before each simulation time step and set atom's property 0 when receiving a new inter atom.

B6: Error in indexing computation when updating molecule velocity

Related Code:

74	76		void ws::getNearLatCoord(const AtomElement &src_atom, const comm::Domain *p_domain,
75	77		_type_atom_index coords[DIMENSION]) {
76	78		auto j = static_cast <_type_atom_index>(lround(src_atom.x[0] * 2 / p_domain->lattice_const));
77		-	auto k = static_cast <_type_atom_index>(lround(src_atom.x[1] / p_domain->lattice_const));
78		-	auto l = static_cast <_type_atom_index>(lround(src_atom.x[2] / p_domain->lattice_const));
	79	+	auto k = static_cast <_type_atom_index>(lround(
	80	+	(src_atom.x[1] - (j % 2) * p_domain->lattice_const / 2) / p_domain->lattice_const));
	81	+	auto l = static_cast <_type_atom_index>(lround(
	82	+	(src_atom.x[2] - (j % 2) * p_domain->lattice_const / 2) / p_domain->lattice_const));
	83	+	
79	84		coords[0] = j - 2 * p_domain->lattice_coord_ghost_region.x_low;
80	85		coords[1] = k - p_domain->lattice_coord_ghost_region.y_low;
81	86		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
84	89		void ws::getNearLatSubBoxCoord(const AtomElement &src_atom, const comm::Domain *p_domain,
85	90		_type_atom_index coords[DIMENSION]) {
86	91		auto j = static_cast <_type_atom_index>(lround(src_atom.x[0] * 2 / p_domain->lattice_const));
87		-	auto k = static_cast <_type_atom_index>(lround(src_atom.x[1] / p_domain->lattice_const));
88		-	auto l = static_cast <_type_atom_index>(lround(src_atom.x[2] / p_domain->lattice_const));
	92	+	auto k = static_cast <_type_atom_index>(lround(
	93	+	(src_atom.x[1] - (j % 2) * p_domain->lattice_const / 2) / p_domain->lattice_const));
	94	+	auto l = static_cast <_type_atom_index>(lround(
	95	+	(src_atom.x[2] - (j % 2) * p_domain->lattice_const / 2) / p_domain->lattice_const));
	96	+	
89	97		coords[0] = j - 2 * p_domain->lattice_coord_sub_box_region.x_low;
90	98		coords[1] = k - p_domain->lattice_coord_sub_box_region.y_low;
91	99		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:

481	480	for (AtomNei::iterator nei_itl = neighbours->begin(false, x, y, z);
482	481	nei_itl != nei_full_itl_end; ++nei_itl) {
483	-	AtomElement &atom_neighbour_up = *nei_itl; // this is a lattice atom.
484	-	delx = (*inter_it).x[0] - atom_neighbour_up.x[0];
485	-	dely = (*inter_it).x[1] - atom_neighbour_up.x[1];
486	-	delz = (*inter_it).x[2] - atom_neighbour_up.x[2];
	482 +	AtomElement &lattice_neighbour = *nei_itl; // this is a lattice atom.
	483 +	delx = (*inter_it).x[0] - lattice_neighbour.x[0];
	484 +	dely = (*inter_it).x[1] - lattice_neighbour.x[1];
	485 +	delz = (*inter_it).x[2] - lattice_neighbour.x[2];
487	486	dist2 = delx * delx + dely * dely + delz * delz;
488	-	if (dist2 < (_cutoffRadius * _cutoffRadius) && !atom_neighbour_up.isInterElement()) {
	487 +	if (dist2 < (_cutoffRadius * _cutoffRadius) && !lattice_neighbour.isInterElement()) {
489	488	// fixme
490	489	fpair = pot->toForce(
491	490	atom_type::getTypeIdByType(*inter_it).type,
492	-	atom_type::getTypeIdByType(atom_neighbour_up.type),
493	-	dist2, (*inter_it).df + atom_neighbour_up.df);
	491 +	atom_type::getTypeIdByType(lattice_neighbour.type),
	492 +	dist2, (*inter_it).df + lattice_neighbour.df);
494	493	
495	494	(*inter_it).f[0] += delx * fpair;
496	495	(*inter_it).f[1] += dely * fpair;
497	496	(*inter_it).f[2] += delz * fpair;
498	497	
499	-	atom_neighbour_up.f[0] -= delx * fpair;
500	-	atom_neighbour_up.f[1] -= dely * fpair;
501	-	atom_neighbour_up.f[2] -= delz * fpair;
	498 +	lattice_neighbour.f[0] -= delx * fpair;
	499 +	lattice_neighbour.f[1] -= dely * fpair;
	500 +	lattice_neighbour.f[2] -= delz * fpair;

B8: Incorrect coding in speed calculation during cascade collision**

Related Code:

589	-	double v_ = sqrt(energy / atom_type::getAtomMass(atom_.type) / mvv2e); // the unit of v is
		100m/s
590	-	double d_ = sqrt(direction[0] * direction[0] + direction[1] * direction[1] + direction[2] * direction[2]);
591	-	atom_.v[0] += v_ * direction[0] / sqrt(d_);
592	-	atom_.v[1] += v_ * direction[1] / sqrt(d_);
593	-	atom_.v[2] += v_ * direction[2] / sqrt(d_);
	589 +	const double v_ = sqrt(2 * energy / atom_type::getAtomMass(atom_.type) / mvv2e); // the unit of
		v is A/ps (or 100m/s)
	590 +	const double d_ = sqrt(direction[0] * direction[0] + direction[1] * direction[1] + direction[2] * direction[2]);
	591 +	atom_.v[0] += v_ * direction[0] / d_;
	592 +	atom_.v[1] += v_ * direction[1] / d_;
	593 +	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:

257	257	configValues.alloyRatio[atom_type::Fe] = ratios["Fe"].as<int>(1);
258	-	configValues.alloyRatio[atom_type::Cu] = ratios["Cu"].as<int>(1);
259	-	configValues.alloyRatio[atom_type::Ni] = ratios["Nii"].as<int>(1);
	258 +	configValues.alloyRatio[atom_type::Cu] = ratios["Cu"].as<int>(0);
	259 +	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:

```
199 199 do i=1,numImplantReac(matNum)
200 - deallocate(ImplantReactions(matNum,i)%reactants)
201 - deallocate(ImplantReactions(matNum,i)%products)
202 - deallocate(ImplantReactions(matNum,i)%min)
203 - deallocate(ImplantReactions(matNum,i)%max)
+ if(allocated(ImplantReactions(matNum,i)%reactants))
+ deallocate(ImplantReactions(matNum,i)%reactants)
201 + if(allocated(ImplantReactions(matNum,i)%products))
+ deallocate(ImplantReactions(matNum,i)%products)
202 + if(allocated(ImplantReactions(matNum,i)%min)) deallocate(ImplantReactions(matNum,i)%min)
203 + if(allocated(ImplantReactions(matNum,i)%max)) deallocate(ImplantReactions(matNum,i)%max)
204 204 end do
```

B2: Incorrect merge of two defect species

Related Code:

```
4309 - !SIA+CuV
4310 - if(products(1)/=0 .AND. products(3) >= products(2)) then
4311 - product2(3)=products(3)-products(2)
4312 - products(2)=0
4313 - products(3)=0
4314 - else if(products(1)/=0 .AND. products(4) >= products(2)) then
4315 - product2(4)=products(4)-products(2)
4316 - products(2)=0
4317 - products(4)=0
4318 - if(product2(4)/=0 .AND. product2(4) <= max3DInt) then
4319 - product2(3)=product2(4)
4320 - product2(4)=0
4321 - end if
4322 - end if
+ !SIA+Cu or Cu+SIA, not combine
4301 + if(products(1)/=0 .AND. products(2)==0 .AND. &
4302 + (defectTemp%defectType(3)/=0 .OR. defectTemp%defectType(4)/=0)) then !SIA+Cu
4303 + isCombined=.FALSE.
4323 4304
4305 + else if(defectTemp%defectType(1)/=0 .AND. defectTemp%defectType(2)==0 .AND. &
4306 + (products(3)/=0 .OR. products(4)/=0)) then !Cu+SIA
4307 + isCombined=.FALSE.
4324 4308
```

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%
```