

# The information of Test Cases and Bugs

---

## 1. Programs

### a) ANT-MOC

Input:

Name	Meaning
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:

**k<sub>eff</sub>** : Effective value-added factors.

### b) MISA-MD:

Input:

Module	Name	Meaning
[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
[simulation.createphase]	timesteps_length	time step
	create_phase	Reading atoms from a file or creating atoms
	create_t_set	the initial temperature
[simulation.alloy]	create_seed	random seed
	fe	random seek for creating atoms in Fe-Cu-Ni alloy material
	cu	the percentage of Cu in Fe-Cu-Ni alloy material
	ni	the percentage of Ni in Fe-Cu-Ni alloy material
[simulation.collision]	collision_step	a parameter of cascading collisions
	lat	
	energy	the energy of Primary Knock-on Atom
	direction	the direction of Primary Knock-on Atom

Output:

Name	Meaning
T	System temperature
E	System energy
lattice_const	lattice constant
id	atom id
step	time step
type	atom type
inter_type	inter-atom type
locate.x	x coordinates of the atom
locate.y	y coordinates of the atom
locate.z	z coordinates of the atom
v.x	x-directional speed of the atom
v.y	y-directional speed of the atom
v.z	z-directional speed of the atom

### c) MISA-SCD:

Input:

Name	Meaning
defectFile	The defect attributes file
meshFile	The mesh file
irradiationType	Type of irradiation ('Cascade' for neutron irradiation, 'FrenkelPair' for electron irradiation, 'None' for no irradiation)
implantScheme	Toggle between Monte Carlo defect implantation and explicit defect implantation
cascadeFile	The cascade defects file
implantType	Where uniformly implanting defects.
implantFile	The data file containing non-uniform implantation profile.
grainBoundaries	Toggle whether we are going to include the effect of grain boundaries.
pointDefect	Toggle whether point defects are allowed to move only.
temperature	Temperature (K)
soluteConcentration	Initial content of solute atoms (Cu) in iron.
numVac	Initial number of vacancies in the simulation system.
dpaRate	The rate of DPA.
totalDPA	Radiation enhanced factor for Cu.
annealTemperature	Annealing temperature (K).
annealTime	Total anneal time.
lattice	The lattice constant (nm).
burgers	Dislocation loop burgers vector (nm).
reactionRadius	Reaction distances (nm).
grainSize	Grain size (nm).
dislocDensity	Dislocation density ( $nm^{-2}$ ).
cascadeVolume	Volume of cascade ( $nm^3$ ).
max3D	Maximum size for SIA defect to diffuse in 3D.
numSims	Number of times to repeat simulation.
totdatToggle	Toggle whether the total defects file is output . (The file contains number of ever defect species, cluster number densities, average cluster size, average radius of clusters
minSolute	The minimum size of solute clusters included in the statistics.
minVoid	The minimum size of vacancy clusters included in the statistics.
minLoop	The minimum size of SIA clusters included in the statistics.
minVS	The minimum size of S_Vac clusters included in the statistics.

Output:

**CuCluster:** the number of Cu Cluster.

#### d) MISA-ETD

Input:

Name	Meaning
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 refer to the file "./test/input/ATHENA/Input information of ATHENA.docx".

Output:

Name	Meaning
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

Method	GID	TID	Reference	Location
Custom Testing		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		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"	
		ANT-MOC-DT01	OpenMOC	test/input/ANT-MOC/DT/c5g7-rodded-A.zip
		ANT-MOC-DT02	OpenMOC	test/input/ANT-MOC/DT/c5g7-rodded-B.zip
		ANT-MOC-DT03	OpenMOC	test/input/ANT-MOC/DT/c5g7-unrodded.zip
		ANT-MOC-DT04	OpenMOC	test/input/ANT-MOC/DT/takeda-unrodded.zip
		ANT-MOC-DT05	OpenMOC	test/input/ANT-MOC/CT/takeda-rodded.zip
		ANT-MOC-DT06-DT11	Derived from ANT-MOC-DT04 by changing the variable of "Num Polars" with "4, 6, ..., 14"	
		ANT-MOC-DT12-DT17	Derived from ANT-MOC-DT05 by changing the variable of "Num Polars" with "4, 6, ..., 14"	
Property-based Testing		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	ANT-MOC-MR2-G1	ANT-MOC-MR2-1-T01	ANT-MOC-CT02	/test/input/ANT-MOC/MT/MTGroup-deepth/c5g7-deepth1.zip
		ANT-MOC-MR2-G1-T02	derived from ANT-MOC-CT02, setting the depth of assembly1 at "level2"	/test/input/ANT-MOC/MT/MTGroup-deepth/c5g7-deepth2.zip
		ANT-MOC-MR2-G1-T03	derived from ANT-MOC-CT02, setting the depth of assembly1 at "level3"	/test/input/ANT-MOC/MT/MTGroup-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

Method	TID	Reference	Location
Custom Testing	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	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

Method	GID	TID	Reference	Location
Differential Testing		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
		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		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
	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
	MISA-MD-MR9-G1	MISA-MD-MR9-G1-T01~T07	MISA-MD-T01~T07	

#### d) MISA-ETD

Method	GID	TID	Reference	Location
Custom Testing		MISA-ETD-CT01	literature	test/input/MISA-ETD/CT/test0.in
Differential Testing		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	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		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

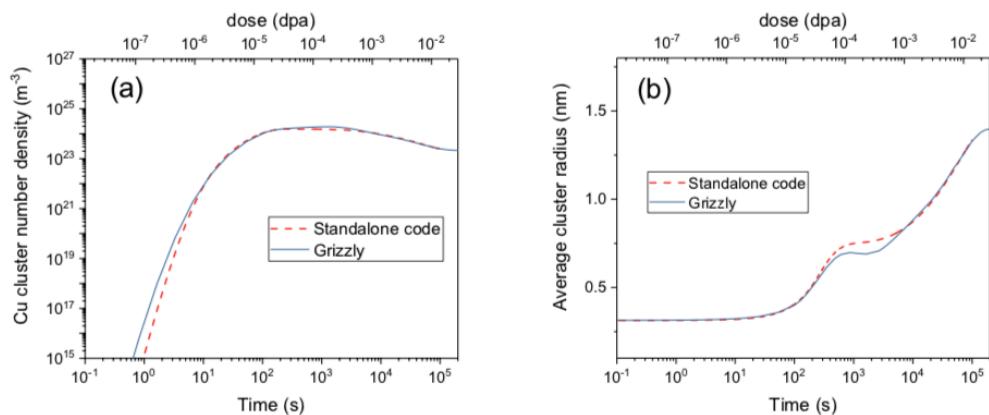
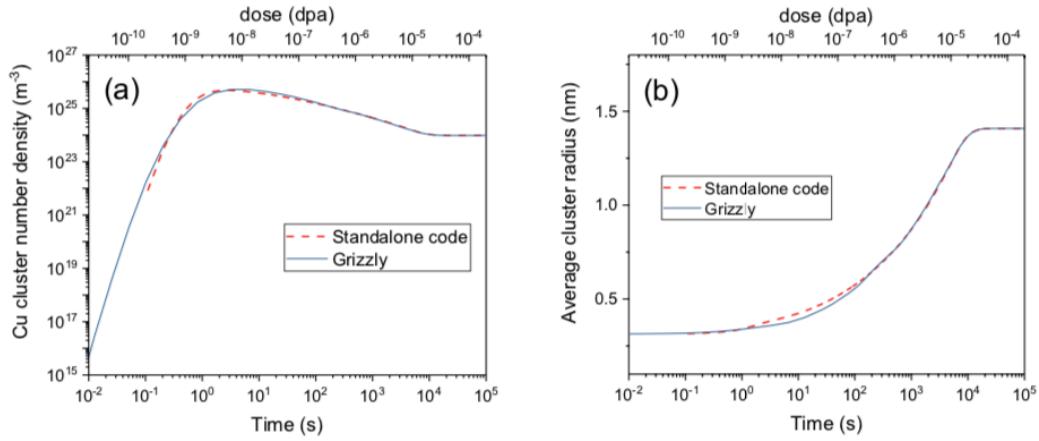
Method	TID	Reference	Location
Custom Testing	ATHENA-T01	expert experience	test/input/ATHENA/CT/test0.inp
Differential Testing	ATHENA-DT01	ATHENA-T01	
	ATHENA-DT02	expert experience	test/input/ATHENA/DT/test1.inp
	ATHENA-DT03	expert experience	test/input/ATHENA/DT/test2.inp
	ATHENA-DT04	expert experience	test/input/ATHENA/DT/test3.inp
Property-baseed Testing	ATHENA-PT01	ATHENA-DT01	
	ATHENA-PT02	ATHENA-DT02	
	ATHENA-PT03	ATHENA-DT03	
	ATHENA-PT04	ATHENA-DT04	
Richardson's Extrapolation	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

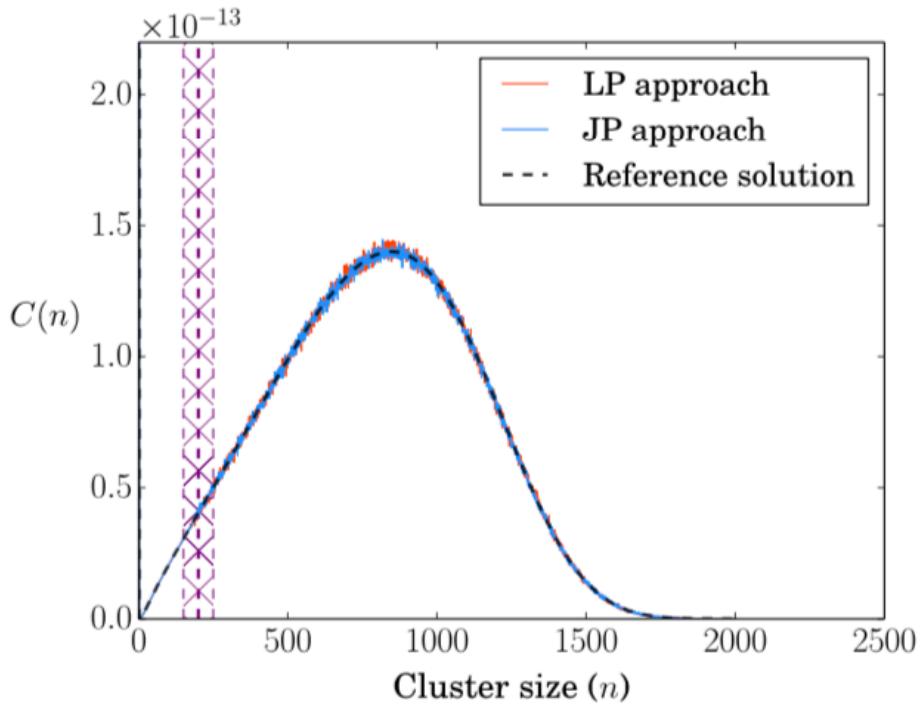
TId	Description	xpected	Expected type
ANT-MOC-CT01	beavrs-3d	0.99927	Value + Deviation
ANT-MOC-CT02	C5g7-rodded-A	1.12806	Value + Deviation
ANT-MOC-CT03	C5g7-rodded-B	1.07777	Value + Deviation
ANT-MOC-CT04	C5g7-unrodded	1.14308	Value + Deviation
ANT-MOC-CT05	Takeda-unrodded	0.97732	Value + Deviation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

TId	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



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

**Reference result:**



### b) Properties

Program	PID	Description	Reason
ANT-MOC	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
MISA-MD	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
	P10	The kinetic energy of the entire system must be conserved (equal to the input PKA energy)	Physics
MISA-SCD	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 $\epsilon$ , where $\epsilon = 20$	Domain knowledge
ATHENA	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
ATHENA	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

Program	MRID	Description	Reason
ANT-MOC	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 $\text{keff}(p(i)) = \text{keff}(p(i'))$	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 $\text{keff}(p(i)) > \text{keff}(p(i'))$	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 $\text{keff}(p(ik)) - \text{keff}(p(ik+1)) < 2 \times 10$ , when $k \rightarrow +\infty$	Computation
MISA-MD	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 $T(p(i)) \neq T(p(i'))$	Physics
	MR7	For a source test case i, if the follow-up test case i' is derived from i by increasing Cu content, then $T(p(i)) \neq T(p(i'))$	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
	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

#### d) Richardson

Program	Expected
MISA-ETD	$2^4$
ATHENA	Unchanged

## 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  - * @breif transfrom 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 2898    -  
2895 2899    - int a = _num_azim/2 - 1;  
2896 2900    - long xy = getMyNum2DTracks(a) - 1;  
2897 2901    - int p = _num_polar - 1;  
2898 2902    - return _my_cum_tracks_per_stack[a][xy][p] + _my_tracks_per_stack[a][xy][p];  
2899 2903    + return _my_num_3D_tracks;  
2900 }
```

**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 1296     /* Get the first 2D track associated with this 3D track */  
1297 1297     Track* track_2D = _tracks_2D_chains[tci->_azim][tci->_x][link];  
1298 1298  
1299 1299     /* Set the start and end point for each 3D track,  
1300 1300     * or just increment arrays */  
1301 1301     while (!end_of_chain) {  
1302 1302         /* Get the 2D track associated with this 3D track */  
1303 1303         Track* track_2D = _tracks_2D_chains[tci->_azim][tci->_x][link];  
1304 1304         while (true) {  
... ... @@ -1398,15 +1398,20 @@ void TrackGenerator3D::set3DTrackData(TrackChainIndexes* tci,  
1398 1398     /* Increment the link index */  
1399 1399     link++;  
1400 1400  
1401 1401     /* Move the starting x-coord to account for periodic BCs for chains  
1402 1402     * In the case of reflective BCs, the starting point is just the end  
1403 1403     * point of the previous link */  
1404 1404     if (!end_of_chain) {  
1405 1405         if (end_of_chain) {  
1406 1406             break;  
1407 1407         }  
1408 1408         else {  
1409 1409             // Move to the next 2D track  
1410 1410             track_2D = _tracks_2D_chains[tci->_azim][tci->_x][link];  
1411 1411             /* Move the starting x-coord to account for periodic BCs for chains  
1412 1412             * In the case of reflective BCs, the starting point is just the end  
1413 1413             * point of the previous link */  
1414 1414             if (!isReflectiveCyclic()) {  
1415 1415                 if (tci->_azim < _num_azim / 4)  
1416 1416                 x2 = _x_min;  
1417 1417                 else  
1418 1418                     x2 = _x_max;  
1419 1419                 Point *next_start = track_2D->getStart();  
1420 1420                 x2 = next_start->getX();  
1421 1421                 y2 = next_start->getY();  
1422 1422             }  
1423 1423         }  
1424 1424     }  
1425 1425 }
```

**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 {
...
...           @@ -1730,7 +1737,7 @@ void Geometry::initializeFlatSourceRegions() {
1730 1737             * @param z_coord the axial height at which the 2D plane of the geometry is
1731 1738             *          formed
1732 1739             */
1733 -     void Geometry::segmentize2D(Track* track, double z_coord) {
1740 +     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:

```

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 */
2436 ...
2437 @@ -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
32 + #pragma omp simd reduction(+:sum)
33   for (L i=0; i < length; i++)
34   sum += vector[i];

```

**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);
4654 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      - void InterAtomList::unpackExInterRecv(int d, int n, const double *lower, const double *upper,
147          particledata *buf) {
148          - AtomElement atom;
149
150      + void
151      + InterAtomList::unpackExInterRecv(Domain *p_domain, particledata *buf, std::list<AtomElement>
152          &delay_buffer, int n) {
153          + AtomElement atom{};
154
155          for (int i = 0; i < n; i++) {
156              atom.id = buf[i].id;
157              atom.type = buf[i].type;
158
159              @@ -154,11 +154,12 @@ void InterAtomList::unpackExInterRecv(int d, int n, const double *lower, const d
160                  atom.v[0] = buf[i].v[0];
161                  atom.v[1] = buf[i].v[1];
162                  atom.v[2] = buf[i].v[2];
163
164          - if (atom.x[d] >= lower[d] && atom.x[d] < upper[d]) {
165              -     inter_list.push_back(atom);
166              -     nlocalinter++;
167
168          + // todo condition: we can judge only one direction the inter atom comes from.
169          + if (ws::isOutBox(atom, p_domain) == box::IN_BOX) {
170              +     addInterAtom(atom);
171
172          } else {
173
174              // todo waring
175              +     kiwi::logs::w("unpack", "unexpected atom, id: {}\\n", atom.id);
176
177              +     delay_buffer.push_back(atom);
178
179          }
180
181      }

```

**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  const unsigned long InterBorderPacker::sendLength(const int dimension, const int direction) {
18  21      const int index = 2 * dimension + direction;
19
20      -     getIntertosend(&domain, dimension, direction,
21                      domain.meas_ghost_length[dimension],
22                      inter_atom_list.intersendlist[index]);
23
24      -     return inter_atom_list.intersendlist[index].size();
25
26      +     std::vector<AtomElement *> &sendlist = inter_atom_list.intersendlist[index];
27
28      + // before x dimension communication, ghost list is empty.
29
30      + comm::Region<comm::_type_lattice_size> region = comm::fwCommLocalRegion(&domain, dimension,
31          direction);
32
33      +     _type_atom_index coords[DIMENSION] = {0, 0, 0};
34
35      +     for (AtomElement &inter_ref : inter_atom_list.inter_list) {
36          // get the lattice coordinate the inter atom belongs to.
37          ws::getNearLatCoord(inter_ref, &domain, coords);
38
39          if (region.isIn(coords[0], coords[1], coords[2])) {
40              sendlist.push_back(&inter_ref);
41
42          }
43
44      }
45
46      +     for (AtomElement &ghost_ref : inter_atom_list.inter_ghost_list) {
47          ws::getNearLatCoord(ghost_ref, &domain, coords);
48
49          if (region.isIn(coords[0], coords[1], coords[2])) {
50              sendlist.push_back(&ghost_ref);
51
52          }
53
54      }
55
56      +     return sendlist.size();
57
58  }

```

**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
101      -     inter_it = inter_atom_list->inter_list.erase(inter_it);
102
103      -     inter_atom_list->nlocalinter--;
104
105      +     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 79         - auto k = static_cast<_type_atom_index>(lround(src_atom.x[1] / p_domain->lattice_const));
78 80         - auto l = static_cast<_type_atom_index>(lround(src_atom.x[2] / p_domain->lattice_const));
79 81         + auto k = static_cast<_type_atom_index>(lround(
80 82             (src_atom.x[1] - (j % 2) * p_domain->lattice_const / 2) / p_domain->lattice_const));
81 83         + auto l = static_cast<_type_atom_index>(lround(
82 84             (src_atom.x[2] - (j % 2) * p_domain->lattice_const / 2) / p_domain->lattice_const));
83 85
84 86         coords[0] = j - 2 * p_domain->lattice_coord_ghost_region.x_low;
85 87         coords[1] = k - p_domain->lattice_coord_ghost_region.y_low;
86 88         coords[2] = l - p_domain->lattice_coord_ghost_region.z_low;
*** ***
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 92         - auto k = static_cast<_type_atom_index>(lround(src_atom.x[1] / p_domain->lattice_const));
88 93         - auto l = static_cast<_type_atom_index>(lround(src_atom.x[2] / p_domain->lattice_const));
89 94         + auto k = static_cast<_type_atom_index>(lround(
90 95             (src_atom.x[1] - (j % 2) * p_domain->lattice_const / 2) / p_domain->lattice_const));
91 96         + auto l = static_cast<_type_atom_index>(lround(
92 97             (src_atom.x[2] - (j % 2) * p_domain->lattice_const / 2) / p_domain->lattice_const));
93 98
94 99         coords[0] = j - 2 * p_domain->lattice_coord_sub_box_region.x_low;
95 100        coords[1] = k - p_domain->lattice_coord_sub_box_region.y_low;
96 101        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) {
483 482             - AtomElement &atom_neighbour_up = *nei_itl; // this is a lattice atom.
484 483             delx = (*inter_it).x[0] - atom_neighbour_up.x[0];
485 484             dely = (*inter_it).x[1] - atom_neighbour_up.x[1];
486 485             delz = (*inter_it).x[2] - atom_neighbour_up.x[2];
487 486             + AtomElement &lattice_neighbour = *nei_itl; // this is a lattice atom.
488 487             + delx = (*inter_it).x[0] - lattice_neighbour.x[0];
489 488             + dely = (*inter_it).x[1] - lattice_neighbour.x[1];
490 489             + delz = (*inter_it).x[2] - lattice_neighbour.x[2];
491 490             dist2 = delx * delx + dely * dely + delz * delz;
492 491             if (dist2 < (_cutoffRadius * _cutoffRadius) && !atom_neighbour_up.isInterElement()) {
493 492             + if (dist2 < (_cutoffRadius * _cutoffRadius) && !lattice_neighbour.isInterElement()) {
494 493                 // fixme
495 494                 fpair = pot->toForce(
496 495                     atom_type::getTypeIdByType((*inter_it).type),
497 496                     atom_type::getTypeIdByType(atom_neighbour_up.type),
498 497                     dist2, (*inter_it).df + atom_neighbour_up.df);
499 498             + atom_type::getTypeIdByType(lattice_neighbour.type),
500 500             + dist2, (*inter_it).df + lattice_neighbour.df);
501
502
503             (*inter_it).f[0] += delx * fpair;
504             (*inter_it).f[1] += dely * fpair;
505             (*inter_it).f[2] += delz * fpair;
506
507             atom_neighbour_up.f[0] -= delx * fpair;
508             atom_neighbour_up.f[1] -= dely * fpair;
509             atom_neighbour_up.f[2] -= delz * fpair;
510
511             lattice_neighbour.f[0] -= delx * fpair;
512             lattice_neighbour.f[1] -= dely * fpair;
513             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
590      -     100m/s
591      -     double d_ = sqrt(direction[0] * direction[0] + direction[1] * direction[1] + direction[2] *
592      -     direction[2]);
593      -     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
590      +     v is A/ps (or 100m/s)
590      +     const double d_ = sqrt(direction[0] * direction[0] + direction[1] * direction[1] + direction[2] *
591      +     * 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 "Ni"

**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["Ni"].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 "Ni".

#### 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)
200  +         if(allocated(ImplantReactions(matNum,i)%reactants))
201  +             deallocate(ImplantReactions(matNum,i)%reactants)
201  +             if(allocated(ImplantReactions(matNum,i)%products))
202  +                 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
4300  + !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%
```