

## **DISCLAIMER**

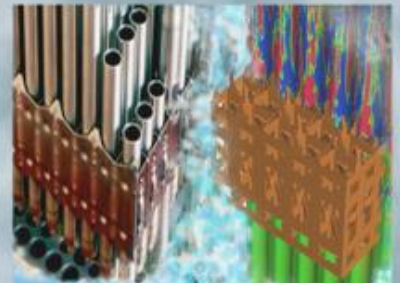
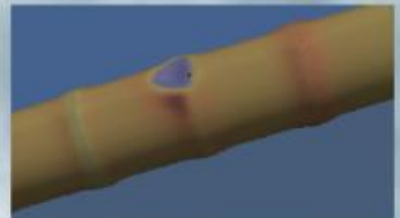
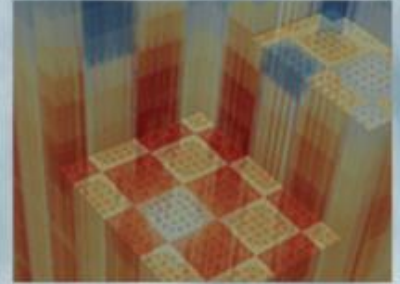
**This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. Reference herein to any social initiative (including but not limited to Diversity, Equity, and Inclusion (DEI); Community Benefits Plans (CBP); Justice 40; etc.) is made by the Author independent of any current requirement by the United States Government and does not constitute or imply endorsement, recommendation, or support by the United States Government or any agency thereof.**

# Restart File Improvements for MPACT

Brendan Kochunas, University of Michigan  
Daniel Jabaay, University of Michigan

Robert Salko, Oak Ridge National Lab

**January, 2017**



## DOCUMENT AVAILABILITY

Reports produced after January 1, 1996, are generally available free via US Department of Energy (DOE) SciTech Connect.

**Website** [www.osti.gov](http://www.osti.gov)

Reports produced before January 1, 1996, may be purchased by members of the public from the following source:

National Technical Information Service  
5285 Port Royal Road  
Springfield, VA 22161  
**Telephone** 703-605-6000 (1-800-553-6847)  
**TDD** 703-487-4639  
**Fax** 703-605-6900  
**E-mail** [info@ntis.gov](mailto:info@ntis.gov)  
**Website** <http://classic.ntis.gov/>

Reports are available to DOE employees, DOE contractors, Energy Technology Data Exchange representatives, and International Nuclear Information System representatives from the following source:

Office of Scientific and Technical Information  
PO Box 62  
Oak Ridge, TN 37831  
**Telephone** 865-576-8401  
**Fax** 865-576-5728  
**E-mail** [reports@osti.gov](mailto:reports@osti.gov)  
**Website** <http://www.osti.gov/contact.html>

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

## REVISION LOG

Revision	Date	Affected Pages	Revision Description
0	1/16/2017	All	Initial Release

**Document pages that are:**

Export Controlled \_\_\_\_\_

IP/Proprietary/NDA Controlled \_\_\_\_\_

Sensitive Controlled \_\_\_\_\_

**Requested Distribution:**

To:

Copy:

## EXECUTIVE SUMMARY

This report documents the work performed to address several defects and features request related to the restart capability in MPACT. The following defects were fixed as a part of this milestone:

- #3616 (Defect) Error in MPACT Restart Files with BPRs
- #3637 (Defect) Restart with Semi-PC Yields Inconsistent Results
- #3652 (Defect) Fix Incorrect Exposure Value on HDF File After Restart
- #3777 (Defect) Fix State Point Number when Using Restart File
- #3986 (Defect) Pin Exposures are Incorrect
- #4106 (Defect) Pin Exposures Equal to Zero After Restart
- #4288 (Defect) Core Average Exposure Wrong when Reading Restart Files

The new features that were added are:

- Axial Re-meshing capability
- Incorporation of MAMBA-1D data necessary to restart coupled MPACT/CTF/MAMBA calculations.
- Addition of state variables to the restart file.

Numerous tests were also implemented to verify and protect these features and this report was prepared to document the work performed.

For consistent restarts of MPACT/CTF/MAMBA-1D calculations The following input options are required otherwise there will be small differences:

```
[MPACT]
  shielder_th 1000 1e-6 1e-6
[MAMBA1D]
  onlyBoronPrecip 1
```

The MPACT option forces the shielding calculation to be performed every outer iteration. This is necessary to obtain agreement within convergence for any restart case. The MAMBA1D option means that the only boron mass that will be counted in the CRUD will be the boron that has precipitated, it excludes the boron in the CRUD layer that is still in solution.



## CONTENTS

EXECUTIVE SUMMARY .....	iii
CONTENTS.....	v
FIGURES.....	vi
TABLES .....	vii
ACRONYMS.....	viii
1. INTRODUCTION .....	1
1.1 Scope and Objectives .....	1
2. WORK PERFORMED .....	1
2.1 Task 1 – Fix Defects.....	1
2.2 Task 2 – State Variables.....	3
2.3 Task 3 – Axial Re-mesh .....	4
2.4 Task 4 – MAMBA Restart .....	8
2.5 Task 5 – Testing .....	13
3. CONCLUSIONS AND FUTURE WORK.....	14
3.1 Future Work.....	14
REFERENCES .....	<b>Error! Bookmark not defined.</b>
There are no sources in the current document. ....	<b>Error! Bookmark not defined.</b>

## FIGURES

Figure 1 StateType Code Design .....	3
Figure 2 Illustration of Axial Mesh Mapping Procedure .....	5
Figure 3 Single Pin Axial Mesh Mapped Power .....	6
Figure 4 Problem 6 Axial Power from Restart and Mapped Restart .....	7
Figure 5 Restart File Structure incorporating Feedback Operators .....	11

## TABLES

Table 1 - Results for Various Single Pin Axial Remeshed Restarts .....	6
Table 2 - Comparisons of Solution of Problem 6 with Restart and Remeshed Restart .....	7

## ACRONYMS

2-D	Two-Dimensional
3-D	Three-Dimensional
BWR	Boiling Water Reactor
CASL	Consortium for Advanced Simulation of Light Water Reactors
CIPS	CRUD Induced Power Shift
CTF	COBRA-TF subchannel thermal-hydraulics code
CZP	Cold Zero Power
HFP	Hot Full Power
HZP	Hot Zero Power
ITC	Isothermal Temperature Coefficient
LWR	Light Water Reactor
MPACT	Michigan Parallel Characteristics Transport code
PC	Predictor-Corrector
PCM	Percent Mille ( $10^{-5}$ )
PPM	Parts per Million (usually boron concentration)
PHI	Physics Integration
PWR	Pressurized Water Reactor
T/H	Thermal-Hydraulics
V&V	Verification and Validation
VERA	Virtual Environment for Reactor Applications
XML	Extensible Markup Language
XS	Cross section(s)

## 1. INTRODUCTION

The purpose of this milestone was to add improvements to the MPACT restart/shuffle file. The previous work from 2015 implemented the initial capability to write, read, and shuffle a restart file in MPACT [1]. That work was focused on adding the initial capability to be used in the analysis of several PWR cores. The work done in this milestone builds upon that initial capability. Over the past year, several user suggested improvements have been requested and defects discovered. The foremost of these improvements, writing MAMBA CRUD data, is required for coupled MPACT-CTF-MAMBA CIPS analysis. The other high priority request was the capability to allow MPACT to restart with a different axial mesh than the one used when the restart file was written. This feature is important multi-cycle depletion analysis since the design of all or even several cycles is known not *a priori*. If MPACT can only run with a single axial mesh for all cycles, that could potentially introduce significant modeling errors should the axial heterogeneity change. The follow on work includes sufficient testing to ensure that these features are implemented properly, as well as to document the features thoroughly.

### 1.1 Scope and Objectives

The list of tasks required for completion of this milestone is:

- a. Fix all known defects in restart/shuffle file
- b. Allow for axial re-meshing of restart file data
- c. Complete implementation of MAMBA restart data
- d. Implement shuffle by assembly serial ID
- e. Add appropriate testing to A-D as evidence of completion
- f. Prepare milestone report documenting efforts

The implementation of shuffling by serial ID was replaced with a refactor of how states are handled within MPACT.

## 2. WORK PERFORMED

### 2.1 Task 1 – Fix Defects

Since the initial restart capability was introduced, its thorough use has uncovered several bugs that have since been fixed. A full list of the bugs fixed as reported in the PHI Kanban system are:

- #3616 (Defect) Error in MPACT Restart Files with Burnable Poison Rods (BPRs)
- #3637 (Defect) Restart with Semi-PC Yields Inconsistent Results
- #3652 (Defect) Fix Incorrect Exposure Value on HDF File After Restart
- #3777 (Defect) Fix State Point Number when Using Restart File
- #3986 (Defect) Pin Exposures are Incorrect
- #4106 (Defect) Pin Exposures Equal to Zero After Restart
- #4288 (Defect) Core Average Exposure Wrong when Reading Restart Files

There were several defects that involved processing data that was missed during the first iteration. There was an error when running a case with BPRs not having their isotopes written to the restart file because they were not specified in the insert block nor did they have any fuel isotopes. The issue discussed was how to determine non-fuel depletable rods. Ultimately, a 2-D lattice map is created by the code and the data in those pin locations is exported to the restart file.

Restarting with the semi-predictor corrector (Semi-PC) depletion option produced different results than using the predictor-corrector (PC) option. The reason for the discrepancy was the difference in the methods, which are highlighted below.

- Predictor/Corrector solution at step n
  - Perform predictor depletion
  - Perform predictor flux calculation
  - Perform corrector depletion
  - Perform corrector flux calculation
- Predictor/Corrector solution from restart:
  - Load corrected isotopics
  - Perform flux calculation (re-solves for corrector flux)
  - Begin depletion at step n+1
- Semi-Predictor/Corrector Solution at step n
  - Perform predictor depletion
  - Perform predictor flux calculation
  - Perform corrector depletion
- Semi-Predictor/Corrector solution from restart:
  - Load corrected isotopics
  - Perform flux calculation (solves for corrector flux)
  - Begin depletion at step n+1

If a restart is performed, then there is no difference in predictor/corrector because upon loading the isotopics, the corrector flux is re-solved. However, in the semi-predictor/corrector after the restart is now the solving for a new flux, which would be a corrector flux. This new step is not present in the no restart semi-predictor/corrector case. So by restarting with semi-predictor/corrector at present, an extra transport solve is performed, presumably leading to a better answer ultimately. This step is the reason for the difference on restart with Semi-PC.

The incorrect exposure value being written to the restart was another issue where the depletion object was not initialized, so there was not data to write. The solution to the defect was to move the calculation of the state exposure from the depletion object to the multistate solver. This change allows the exposure to always be calculated whether or not the depletion object is initialized.

Fixing the state point numbering when using a restart file created a discussion about how MPACT uses the restart file and specifically under what circumstances a solve is performed for a given state point. The underlying issue was MPACT solved two states that were identical, and thus appeared to be numbered incorrectly in the output file. The conclusions from the discussion were:

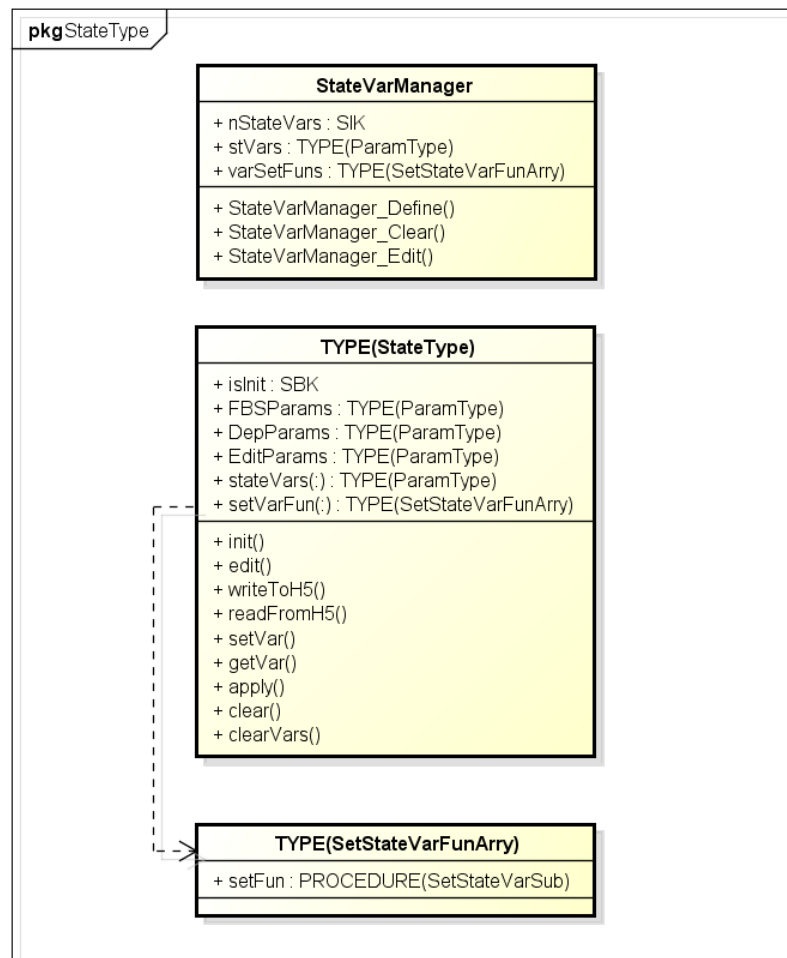
- If there is no `deplete` card, a `[STATE]` block will trigger a single calculation at the current exposure.
- A `deplete` card will run a calculation at each exposure point listed.
- A `restart_read` card will reset the current exposure to what was on the file.
- If the `deplete` card is used in the first `[STATE]` block, and there is no `restart_read` in this block, then the first exposure point on the `deplete` card must be zero, or there is an error.
- If the `deplete` card appears in the same `[STATE]` block as a `restart_read` card, then the first entry on the `deplete` card must match the cycle exposure of the restart state that was loaded, or there is an error.

The pin exposures were not consistent with the total core exposure, and this ticket also exposed an issue with the heavy metal masses not being written correctly on each pin. The pin exposures were zero after a restart due to no depletion being done. The mass and burnup calculations were not performed because they were controlled by the depletion object. The fix was to move the calculation to the base depletion class so it can be called regardless of whether the restarted state includes depletion or not.

For the last defect, the core average exposure was only updated once during the reading of the first restart point. Any subsequent state blocks where a restart point was also read would not update the core exposure. The solution was to be able to update the exposure values correctly for multiple states where restarts are read. The total time required to fix these defects is estimated at around 98 man hours.

## 2.2 Task 2 – State Variables

The other improvement was to read and write the state data to the restart file. This processing would save the user from having to specify the exact state definition for both the restart write and restart read case. The ticket prompted a refactor of how MPACT defines and tracks a single state. The current design is shown in Figure 1 below:



powered by astah

Figure 1 StateType Code Design

The new state type consists of parameter lists for other components including the feedback solver, depletion object, and edits. The primary data structure is the `statevars` array of parameter lists. This array is where each individual state variable is set. The `setvarfun` is an array of procedure pointers that corresponds to the state variable at a given index in the arrays. This array stores the procedures that determine how the variable is applied to the cross section mesh, if at all.

All of the state variables are now defined upfront. The current list of variables is:

- State Information
  - `iState, title, op_date`
- Core Description
  - `rated_power, power, rated_flow, flow, rated_flow_units, bank_pos, bank_labels, boron, b10`
- T/H
  - `tinlet, bypass, pressure, void, modden, coolden, tfuel`
- Depletion
  - `exposure, exposure_efpd, exposure_hours`
- Chemistry
  - `cool_chem, uniform_crud_thickness, uniform_crud_mass, uniform_crud_boron_mass`
- Solution Options
  - `reset_sol, feedback, search, search_bank, kcrit, xenopt, samopt, rlx_xesm`

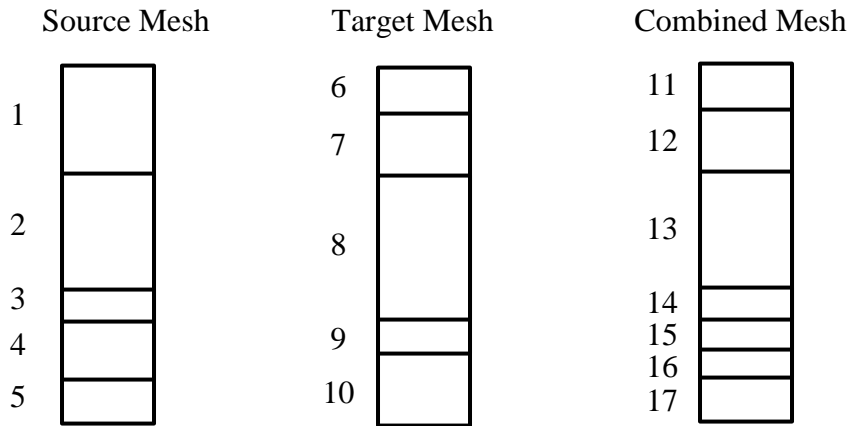
All of these variables can be defined with a range of acceptable values. In this way, the range is used to check that the user input variables are properly specified.

The process for adding a new state variable is to first define it on the state variable manager. Then, add the necessary input processing to parse the XML input file and assign the data to the specified state. The rest of the processing will be handled by the code to initialize the state type with all of the variables defined in the manager with their given defaults. The defaults will be overridden if the value is given by the user. The state variable object has its data written to both the restart file for a state with a restart write and to the output file for all states.

### 2.3 Task 3 – Axial Re-mesh

The axial remesh was another top feature priority for this milestone. The initial work using MPACT to model the Watts-Bar Unit 1 reactor was fortunate in that the assembly models used for the cycle depletion did not vary significantly in the axial direction. Also, since the reactor had years of operational history, it was straightforward to find an axial mesh that would work for modeling the 12 cycles. However, in the future that information may not be knowable *a priori*, thus restarting to another cycle may require a different axial mesh than the previous cycle. That possibility is the motivation for this task. Otherwise previous cycles would need to be rerun with the new axial mesh, thus wasting all the previous time spent running cases and performing analyses.

In order to allow MPACT to restart with a different axial mesh, more information needs to be written to the restart file and more processing needs to be done when reading a restart file. The restart file needs to have the axial mesh corresponding to the data at the time it was written. The data can then be read into MPACT and given the old axial mesh, the restart isotopics and volumes, and the new axial mesh, the restart isotope number densities can be volume averaged into the new axial mesh description.

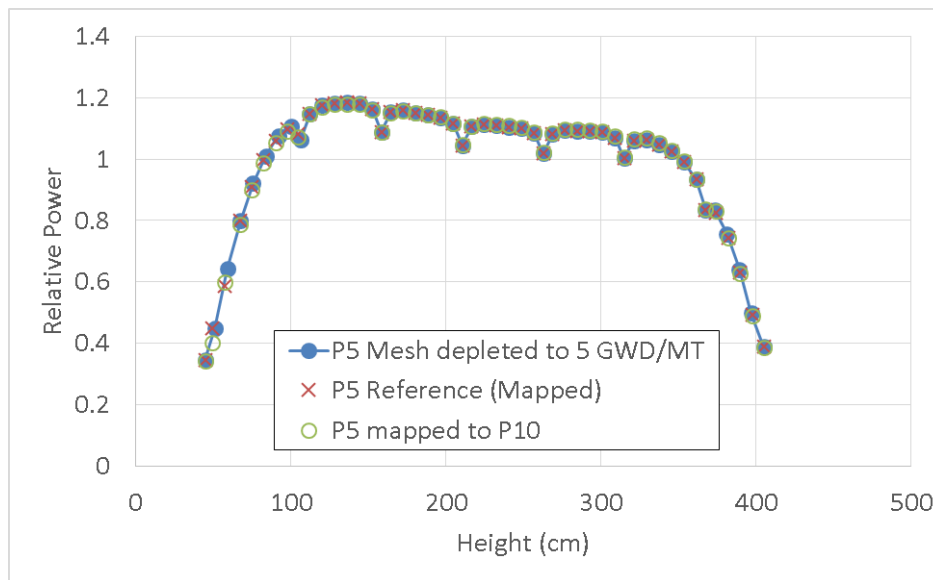


**Figure 2 Illustration of Axial Mesh Mapping Procedure**

The old axial mesh shown on the left will be used to apply the isotope number densities to the new mesh in the middle. The mesh on the right is the union of both the old and new. A few mapping operators were initially investigated, however it was determined that volume-weighted homogenization was the best way to map the number densities and exposures. In the mapping procedure, the code first checks to see if the axial mesh are different. If they are, then the mesh mapping is performed. A union of the axial mesh is first constructed and index maps for the source and target mesh to the unionized mesh are then determined. The relative axial heights in each plane from the unionized mesh of the values of the combined mesh are calculated and then used with the old mesh to scale the values onto the new mesh.

Every region on the new axial mesh can be expressed in terms of the old and combined mesh regions, and therefore the number densities can be volume/height weighted into the new axial mesh regions.

To verify this capability, a single rod calculation was tested depleting the rod to 5 GWd/MT. These results are shown in Figure 3. The mapping was originally going to be applied for all cases, but logic was later added that to first verify that the axial mesh were different before performing the mapping.



**Figure 3 Single Pin Axial Mesh Mapped Power**

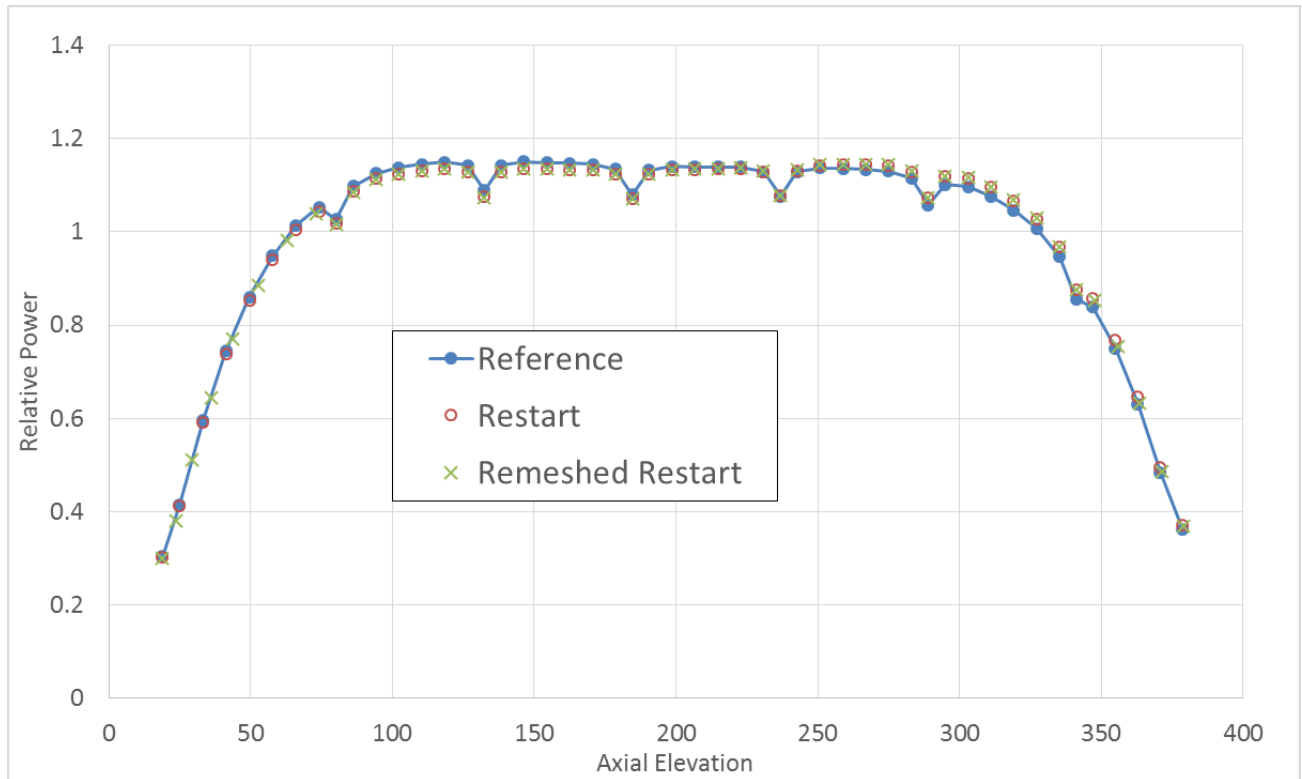
A comparison of the solutions for the single pin case is also give:

**Table 1 - Results for Various Single Pin Axial Remeshed Restarts**

CASE	dk	Max %	RMS %
Restart Write	1.1576106	---	---
Restart Read	6.89	0.790%	0.427%
Restart Map	7.66	4.760%	0.760%
Restart +1 plane	6.88	0.790%	0.427%
Restart -1 plane	6.94	0.780%	0.419%

In the mapped case, the values are not interpolated in the comparison by `MPACTdiff` therefore the 4.76% max difference is quite misleading. This evidenced when looking at the previous figure. This difference occurs for the bottom most plane.

Additional cases were evaluated. The next logical step was to simulate an assembly with feedback. For this case, the readily available VERA progression problem 6 was used. For this case, depletion to 5GWd/MT was also performed.



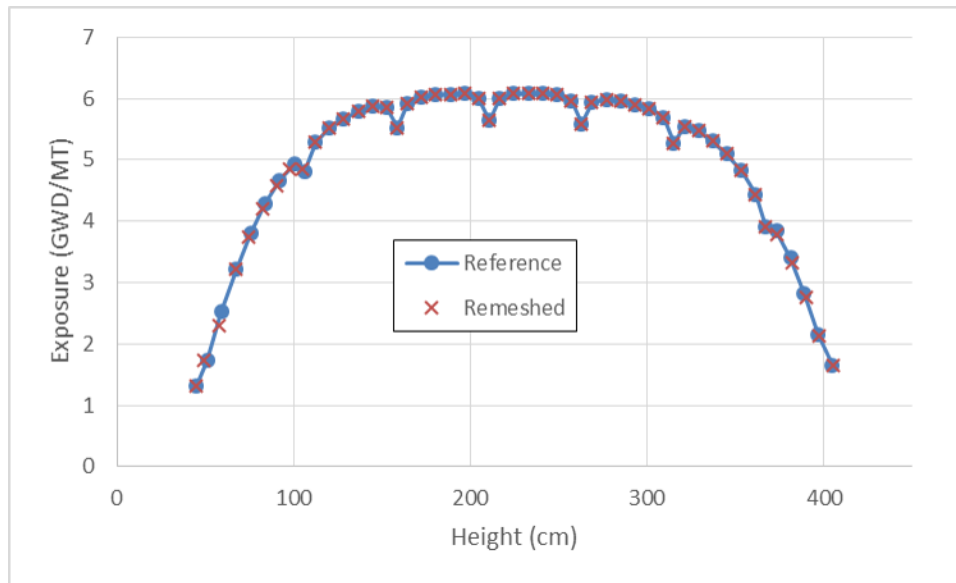
**Figure 4 Problem 6 Axial Power from Restart and Mapped Restart**

Comparisons of the solutions for this case are given in Table 2 below. As can be seen from this result the differences from remeshing the WB1C1 axial mesh to the WB1C2 mesh introduces nearly negligible differences. The reason that there are differences in the basic restart case are due to differences in when the resonance self-shielding calculation is performed.

**Table 2 - Comparisons of Solution of Problem 6 with Restart and Remeshed Restart**

Case	Max Diff %	RMS %
Original Mesh	2.110	1.25
Mapped to P10 Mesh	2.19	1.32

In addition to the mapped isotopes, the burnup must also be mapped. To verify the burnup is mapped correctly the edit for the pinwise exposure is shown below for a single rod.



## 2.4 Task 4 – MAMBA Restart

The MAMBA restart capability, built on previous work done in CTF to provide interfaces to set and get the MAMBA work arrays by rod. These interfaces look like:

```

subroutine CTF_set_crud_work_arrays(pin_number,Rwork_ne,Iwork_ne,&
                                   Rwork_se,Iwork_se,&
                                   Rwork_sw,Iwork_sw,&
                                   Rwork_nw,Iwork_nw)

integer,intent(in) :: pin_number
integer,allocatable,intent(in) :: Iwork_ne(:),Iwork_se(:),Iwork_sw(:),Iwork_nw(:)
real,allocatable,intent(in) :: Rwork_ne(:),Rwork_se(:),Rwork_sw(:),Rwork_nw(:)

```

```

subroutine CTF_get_crud_work_arrays(pin_number,Rwork_ne,Iwork_ne,&
                                   Rwork_se,Iwork_se,&
                                   Rwork_sw,Iwork_sw,&
                                   Rwork_nw,Iwork_nw)

integer,intent(in) :: pin_number
integer,allocatable,intent(in) :: Iwork_ne(:),Iwork_se(:),Iwork_sw(:),Iwork_nw(:)
real,allocatable,intent(in) :: Rwork_ne(:),Rwork_se(:),Rwork_sw(:),Rwork_nw(:)

```

Where the `pin_number` is the global CTF rod index, and the work arrays are for each quadrant around the rod and for the full length of the rod. The routines must be called by all CTF processors to insure the data is transferred and set properly since different quadrants of the same rod may be on different processors. The work arrays for each quadrant are allocated only if that quadrant of the pin exists in the model. This has implications for symmetry.

The original interfaces in MPACT implemented last year to facilitate this feature relied on the infrastructure already built up for the restart file. In this infrastructure the data in the file is mapped to MPACT's fine, cross section mesh and handles symmetry and data unfolding, etc. The expectation was that MPACT would need to be storing values for each cross section mesh region labeled as CRUD. However, given the nature of the CTF interfaces, this approach was not amenable.

Thus, a new set of interfaces needed to be implemented for the restart file. These original interfaces are defined as:

```
PURE FUNCTION hasRstDat(thisFBOp) RESULT(bool)
  CLASS(FeedbackOpType), INTENT(IN) :: thisFBOp
  LOGICAL(SBK) :: bool
ENDFUNCTION
```

```
SUBROUTINE getRstCorDat(thisFBOp, idat, rdat, v)
  CLASS(FeedbackOpType), INTENT(IN) :: thisFBOp
  INTEGER(SIK), ALLOCATABLE, INTENT(INOUT) :: idat(:)
  REAL(SRK), ALLOCATABLE, INTENT(INOUT) :: rdat(:)
  INTEGER(SIK), INTENT(OUT) :: v
ENDSUBROUTINE
```

```
SUBROUTINE getRstAsyXSDat(thisFBOp, iasy, jasy, irot, sym, nxsreg_pin, idat, rdat, v)
  CLASS(FeedbackOpType), INTENT(IN) :: thisFBOp
  INTEGER(SIK), INTENT(IN) :: iasy
  INTEGER(SIK), INTENT(IN) :: jasy
  INTEGER(SIK), INTENT(IN) :: irot
  CHARACTER(LEN=*), INTENT(IN) :: sym
  INTEGER(SIK), INTENT(IN) :: nxsreg_pin(:, :, :)
  INTEGER(SIK), ALLOCATABLE, INTENT(INOUT) :: idat(:, :)
  REAL(SRK), ALLOCATABLE, INTENT(INOUT) :: rdat(:, :)
  INTEGER(SIK), INTENT(OUT) :: v
ENDSUBROUTINE
```

Here the first interface is used by the restart file object to determine if the physics implemented for a given feedback operation has data to add put in the restart file. There were effectively two types of data treated: core level data and cross-section-mesh-wise data for each assembly. So the “has” function existed for each type of data. The get routines (and corresponding set routines which are not shown) then include basic data containers and for intrinsic data types and version flag, so that the version of the feedback data may vary independently of the restart file version. In the XS interface the x and y assembly locations along with the rotation and symmetry of the assembly are required information to perform the mapping. The `nxsreg_pin` array is a 3D array corresponding to the pin mesh that indicates the number of cross section mesh regions that have data in each pin cell mesh location. To identify this another routine was needed to “filter” the cross section mesh to those that have data relevant to the feedback physics and those that do not. This was done by defining the following function so the restart file can query the feedback operation for the function to perform the filtering.

```
PURE FUNCTION XSMeshFilterFunc(thisXSMesh) RESULT(bool)
  TYPE(XSMeshType), INTENT(IN) :: thisXSMesh
  LOGICAL(SBK) :: bool
ENDFUNCTION XSMeshFilterFunc

PURE SUBROUTINE getXSMFilter_null_FBOp(filterFunPtr)
  PROCEDURE(XSMeshFilterFunc), POINTER, INTENT(OUT) :: filterFunPtr
ENDSUBROUTINE getXSMFilter_null_FBOp
```

Again, these existing interfaces facilitated placing core level, and cross-section-mesh-wise data in the restart file. To be able to handle the case of rod-wise data for MAMBA the following interfaces needed to be defined for a feedback operation. This effectively used the same pattern as the previous interfaces, but instead defined data for the pin-wise mesh.

```

FUNCTION PinMapFun(thisFBOp, sym, iasy, ipinx, ipiny, ipinz) RESULT(ipin)
  CLASS(FeedbackOpType), INTENT(IN) :: thisFBOp
  CHARACTER(LEN=*), INTENT(IN) :: sym
  INTEGER(SIK), INTENT(IN) :: iasy
  INTEGER(SIK), INTENT(IN) :: ipinx
  INTEGER(SIK), INTENT(IN) :: ipiny
  INTEGER(SIK), INTENT(IN) :: ipinz
  INTEGER(SIK) :: ipin
  ipin=-1
ENDFUNCTION PinMapFun

```

```

SUBROUTINE getMaxRstDatPin(thisFBOp, maxIdatPin, maxRdatPin)
  CLASS(FeedbackOpType), INTENT(IN) :: thisFBOp
  INTEGER(SIK), INTENT(OUT) :: maxIdatPin
  INTEGER(SIK), INTENT(OUT) :: maxRdatPin
  maxIdatPin=0
  maxRdatPin=0
ENDSUBROUTINE getMaxRstDatPin_null_FBOp

```

```

SUBROUTINE getRstAsyPinDat(thisFBOp, ipin, ipinsym, irot, sym, idat, rdat, v)
  CLASS(FeedbackOpType), INTENT(IN) :: thisFBOp
  INTEGER(SIK), INTENT(IN) :: ipin
  INTEGER(SIK), INTENT(IN) :: ipinsym
  INTEGER(SIK), INTENT(IN) :: irot
  CHARACTER(LEN=*), INTENT(IN) :: sym
  INTEGER(SIK), ALLOCATABLE, INTENT(INOUT) :: idat(:)
  REAL(SRK), ALLOCATABLE, INTENT(INOUT) :: rdat(:)
  INTEGER(SIK), INTENT(OUT) :: v
ENDSUBROUTINE

```

The first interface here is analogous to the cross section mesh filter function. The purpose of this routine is to convert the  $i, j, k$  indexing of the pin mesh in the restart file to a global 1-D index corresponding to how the feedback solver indexes the given location. The valid return arguments should conform to 1-based indexing and therefore be larger than 0. Arguments are needed for the assembly index in this case as well as the symmetry to use. The fuel shuffle file is able to then call this routine to perform symmetry unfolding type operations. Using this interface the fuel shuffle file will identify all pin mesh locations having data on the feedback operation that needs to be written to the restart file. The fuel shuffle file create a reduced list of all pin indices that get mapped and remove duplicates before proceeding to the next step which is to retrieve data for these pins.

This brings us to the second interface which is provided so that the fuel shuffle file can pre-allocate arrays for each unique feedback pin and all of the data it will receive for that pin. The last interface is for the same for the set procedure, but with modified intent on the arguments.

In the case of MAMBA the *idat* and *rdat* arrays are the quadrant data flattened and placed into the order of NE, SE, SW, NW for the quadrants. For the MAMBA feedback operator these implementations are trivial adaptor functions. The actual and complex implementations for these

routines are implemented in MPACT's CTF coupler. The CTF coupler, the interfaces by CTF are called directly, then data is unfolded for a whole pin and rotated as required by the model. These routines are also responsible for flattening and decomposing the *idat* and *rdat* arrays into the respective quadrant arrays.

The final restart file structure for the MAMBA data looks something like:

```

CORE_<label> -> RESTART_<label>
-> version
-> cycle label
-> unit number
-> op_date
-> nxasy
-> nyasy
-> xlabel
-> ylabel
-> Assembly_<IAF>_<JAF>
-> ... (unchanged)
-> INSERT_<IAF>_<JAF>
-> ... (unchanged)
-> <FEEDBACKOP_NAME>
-> version
-> core_rdat(:) (if feedback has core level restart real dat)
-> core_idat(:) (if feedback has core level restart integer dat)
-> Assembly_<IAF>_<JAF> (if feedback has assembly level dat)
-> nxpin (link to Assembly_<IAF>_<JAF>)
-> nypin (link to Assembly_<IAF>_<JAF>)
-> nz (link to Assembly_<IAF>_<JAF>)
-> symmetry (link to Assembly_<IAF>_<JAF>)
-> rotation (link to Assembly_<IAF>_<JAF>)
-> axial mesh (link to Assembly_<IAF>_<JAF>)
-> version
-> xs region volume (link to Assembly_<IAF>_<JAF>)
-> nxsreg_pin
-> index_map (converts ith feedback region to ixsmesh)
-> rdat(:, :)
-> idat(:, :)
-> pin data version
-> nFBPins
-> nIDatPin
-> nRDatPin
-> PinIdxMap(:, :, :)
-> IDatPin(:, :)
-> RDatPin(:, :)

```

**Figure 5 Restart File Structure incorporating Feedback Operators**

In this figure the bolded entries are what was added and is used for the MAMBA restart data. The italicized entries refer to those created in the initial implementation last year. The rest of the entries are standard to the restart file and were unchanged in this work.

To verify this capability three tests were examined. A single rod case, a single assembly case, and a multi-assembly case. The single rod case input is in Appendix A. The case effectively depletes to 180 EFPD in 45 day increments and writes a restart file. Then this restart file is read and the solution is recalculated and compared. The CRUD edits are given in Table 3.

Plane	pin_crud_boron_dens			pin_crud_mass_dens			pin_crud_thickness		
	Original	Restart	diff	Original	Restart	diff	Original	Restart	diff
Bottom	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0
2	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0
3	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0
4	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0
5	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0
6	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0
7	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0
8	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0
9	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0
10	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0
11	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0
12	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0
13	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0
14	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0
15	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0
16	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0
17	0.00E+0	0.00E+0	0.00E+0	1.15E+0	1.15E+0	8.92E-8	7.20E+0	7.20E+0	5.58E-7
18	4.45E-2	4.45E-2	0.00E+0	1.51E+0	1.51E+0	1.18E-7	9.43E+0	9.43E+0	7.38E-7
19	4.45E-2	4.45E-2	0.00E+0	1.57E+0	1.57E+0	0.00E+0	9.84E+0	9.84E+0	0.00E+0
20	4.45E-2	4.45E-2	0.00E+0	1.60E+0	1.60E+0	9.15E-8	1.00E+1	1.00E+1	5.72E-7
21	4.45E-2	4.45E-2	0.00E+0	1.63E+0	1.63E+0	1.32E-7	1.02E+1	1.02E+1	8.24E-7
22	4.45E-2	4.45E-2	0.00E+0	1.63E+0	1.63E+0	1.39E-7	1.02E+1	1.02E+1	8.70E-7
23	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0
24	0.00E+0	0.00E+0	0.00E+0	6.45E-1	6.45E-1	1.35E-7	4.04E+0	4.04E+0	8.42E-7
25	0.00E+0	0.00E+0	0.00E+0	1.09E+0	1.09E+0	1.40E-7	6.83E+0	6.83E+0	8.78E-7
26	0.00E+0	0.00E+0	0.00E+0	1.26E+0	1.26E+0	2.00E-7	7.89E+0	7.89E+0	1.25E-6
27	4.45E-2	4.45E-2	0.00E+0	1.47E+0	1.47E+0	1.57E-7	9.21E+0	9.21E+0	9.84E-7
Top	4.45E-2	4.45E-2	0.00E+0	1.56E+0	1.56E+0	1.42E-7	9.77E+0	9.77E+0	8.90E-7

The CRUD work arrays were verified to be passed exactly as they are retrieved from MAMBA (through CTF).

For the `mass_dens` and `thickness` edits, the results are effectively within single precision. It is suspected that somewhere in MAMBA there is a conversion of the work array from double precision down to single precision, and perhaps back (for the edit). The boron mass density within the CRUD will be different on the restarted problem unless the following option is set in the `[MAMBA]` block of the VERA input:

```
onlyBoronPrecip 1
```

Another change in MAMBA1D that was required to make this work was to eliminate precipitation for “restart time steps” that are  $< 2$  seconds.

For the single assembly case, the problem was run in quarter symmetry to generate the restart file. Following this the two cases were run that read the restart file, one that was in full symmetry, and one that was in quarter symmetry. The full and quarter symmetry cases differed by 0.05 pcm, well below the convergence tolerance of 0.1 pcm. Indicating the symmetry unfolding for a quarter symmetric assembly is correct. A cursory examination of the data showed the same trends for the CRUD edits, as in the single rod case. Finally for the 4-mini like problem the again there are trends similar to the previous two cases.

## 2.5 Task 5 – Testing

Each of the aforementioned features included the addition of new tests for the features.

When fixing a defect, a new regression test was added or a test was updated. The list of modified tests for each defect ticket are:

- #3616 - added regression test `verify_1lv11assem3x3pin_full`
- #3637 - No specific modifications were made for this as a bug in the Semi-PC depletion time stepping scheme was determined to have a bug.
- #3652 - `MPACT_libs_testMultiStateSolver` was updated to verify this defect was fixed.
- #3777 - No tests updated, however, this fixed the `MPACT_exe_testProgression_Problems_10-mini` regression test.
- #4106 - After fixing this defect, the `evenpin_unfold*_read` regression tests needed to be rebaselined. There are at least two tests protecting this feature.
- #4288 - The `verify_1lv11assem1pin_full` regression test was updated to perform multiple restart reads to protect this feature.

In the case of the state variables, new unit tests were added for these objects. The multi-state solver was also updated to insure that the state variable object was being used correctly. Then several regression tests were added to verify the capability to set the appropriate state variables from state to state. These tests are:

- `MPACT_libs_testStateTypes` (unit test)
- `MPACT_Drivers_testStateVars` (unit test)
- `MPACT_exe_testVerify_stateCount_nominal` (regression test)
- `MPACT_exe_testValid_basic` (regression test)
- `MPACT_exe_testValid_bwr` (regression test)
- `MPACT_exe_testValid_crit` (regression test)
- `MPACT_exe_testValid_depl` (regression test)
- `MPACT_exe_testValid_edits` (regression test)
- `MPACT_exe_testValid_feedback` (regression test)
- `MPACT_exe_testValid_restartstate` (regression test)

The axial re-mesh capability when performing restarts added a new regression test for a single rod titled `MPACT_exe_testValid_axialRemesh_restart` to verify this feature. This model is a single rod, and uses a restart file generated for the axial mesh described for Progression Problem 5 based on the Watts Bar Unit 1 cycle 1. The mesh that this data is mapped to includes the axial mesh that was developed for Watts Bar Unit 1 cycle 2. Additionally, in the implementation of the feature manual, testing was performed to verify that adding or subtracting planes from the model did not cause issues. This evidence is available in the ticket for this feature on the PHI Kanban (#3562).

For the development of the MAMBA restart capability, a single assembly model was used. Much to our surprise there were no existing regression test inputs for MPACT that incorporate MAMBA and CTF feedback. Thus one was developed from existing uncoupled regression test models, using operation similar to the one developed for CTF to verify the CTF interfaces are correct in setting and getting then data to and from MAMBA. In development of the tests a single rod and multi-assembly case were also added to better cover the feature. The new regression tests for this feature are:

- MPACT\_exe\_testMambaRestart\_singlerod
  - Single rod case (CONTINUOUS)
- MPACT\_exe\_testMambaRestart\_6-mini
  - Single assembly case (HEAVY)
- MPACT\_exe\_testMambaRestart\_4-mini
  - Multi-assembly case with quarter symmetry (HEAVY)

The aforementioned tests for the MAMBA restart capability have been added to the repository as regression tests. This constitutes three tests for a single rod, single assembly, and multi-assembly case.

### 3. CONCLUSIONS AND FUTURE WORK

#### 3.1 Conclusions

The restart file capability was modified to include three important features:

- Support for MAMBA for restart of CRUD calculations.
- Support for axial re-meshing during shuffle.
- Reading/Writing of state variables to the restart file.

In addition to that numerous bugs were fixed.

#### 3.2 Future Work

There are still outstanding features to be added to the restart file capability. The two primary requests are shuffling by assembly serial ID and capability to store the geometry data in some form on the restart file.

In addition to that with the incorporation of other features, there are new requirements on the restart capability. These include support for changing the thermal expansion between restarts and shuffles, adding in capability to flake off CRUD during a shuffle.