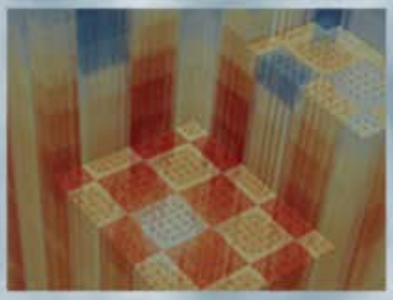
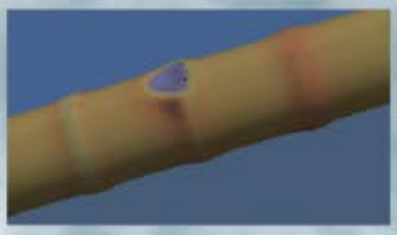
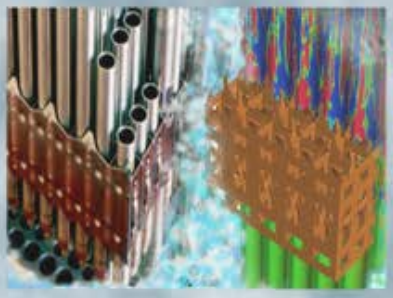
CASL-U-2017-1305-000







Review of Bison Usage in VERA

Russell Gardner, Idaho National Laboratory
Shane Stimpson, Oak Ridge National Laboratory
Kevin Clarno, Oak Ridge National Laboratory
Roger Pawlowski, Sandia National Laboratories

03/31/2017

Approved for Public Release

REVISION LOG

Revision	Date	Affected Pages	Revision Description
0	3/31/2017	All	Initial Release

Document pages that are:

Export Controlled None

IP/Proprietary/NDA Controlled None

Sensitive Controlled None

Unlimited All

Requested Distribution:

To:

Copy:

The following is a narrative of CASL-PHI related activities within BISON from October 2016 to March 2017 for milestone reporting purposes. The CASL milestone #1766 stated the following:

"Bison is being used in VERA in a variety of ways; this milestone will document an independent review of the current usage of Bison and provide guidance that will improve the accuracy, performance, and consistency in the ways that Bison is used. This task will entail running a suite of small, single and multi-cycle problems with VERA-CS, followed by Bison, and Tiamat (inline) and evaluating the usage. It will also entail performing several detailed ramp to full power solutions to compare the one-way coupled solver with the fully-coupled Tiamat. This will include at least one iteration with the PHI team to incorporate some of the feedback and improve the usage. This work will also be completed in conjunction with an FMC task to evaluate the ability of Bison to model load-follow in a PWR."

Multi-App Output Bug

It was reported by CASL that there was an issue with the outputted CSV files when the MOOSE multi-app system was used. The MOOSE multi-app system is being used for the purpose of spawning multiple BISON runs (known as sub-apps) from an originating (master) input file. The issue was that the CSV files produced from the master were complete while the sub-app's CSV files were not. In the case of the input files that were sent to INL to demonstrate the issue, the master CSV file had data for 31 timesteps and the sub-app's CSV file only had data for 6 timesteps. The output CSV files should have been identical in this case. The complete CSV files are necessary for data transfers within Tiamat.

The input file was analyzed and nothing that was an obvious fault showed. Through a model-by-model dissection of the input file it was found that the Damper block was the culprit. A simple MOOSE test that reproduced the results was made and a MOOSE issue was submitted for review (github issue #8245). The purpose of the Damper block is to give control over how a simulation advances. For example, if one needs the temperature to increment in steps no larger than 50 Kelvin, a maximum increment damper may be used to watch the temperature variable and assure that it does not. In this case, BISON was using a maximum increment damper to attempt to clamp down on solutions that were numerically correct, but not physically (i.e. negative temperatures in Kelvin). There is another option in BISON called BoundingValueNodalDamper that will allow for direct minimum and maximum control of a variable on the non-linear iterations. This option was tested with the original input file and the results for the master and sub-app came back matching. This method and the plotted results were reported to CASL. The original bug with the max increment damper has been since corrected inside MOOSE.

Volumetric Locking

It was reported by CASL that after a version update the iteration counts for BISON runs increased by about a factor of three. There was a gap in-between CASL synchronizes at this point, but a possible git commit was reported to BISON as the start of the problem. Input files were sent to BISON and the problem was verified. Over the course of about a week a git bisect was completed on the BISON git repository to pin point the exact commit that changed the behavior.

For information sake, git bisect is a tool built into git that allows for locating problems by marking a start point and an end point (usually a point when the problem did not exist and the current date) in the repository. Git then picks a commit in the middle which one would then build against and test the problem. The problem is either seen or not seen and once the results are communicated to git a new comment is chosen in a bisecting style. This process is repeated until the problem commit is located.

The results of the bisect pointed to a commit that added a volumetric locking correction in to BISON; this option was set default as true. Volumetric locking at its simplest is the artificial hardening of a material due to the inability of the meshed element to shear. It is also a problem that would probably only be seen in situations of high plasticity (i.e. LOCA, RIA) and these were normal operation simulations. The initial advice was to turn off volumetric locking, but this problem touched off a good discussion with CASL and multiple BISON developers. Options were found and tested that used the volumetric locking correction, a numerical pre-conditioner and mechanical large strain deformation that had no speed loss from pre-volumetric locking timeframe.

Access to ORNL Resources

Russell Gardner was granted access to CASL computer resources at ORNL for the purpose of troubleshooting and input file assistance. During a visit to INL Kevin Clarno assisted in setting up VERA and a nightly build for Russell's account.

Watts Bar Load Follow

BISON was asked to investigate the 16 rods from the Watts Bar load-follow quarter core simulation that were failing. On receiving the input files they were all run on INL's Falcon computer to verify the failure. All rods failed on BISON defined minimum dt at different times in the simulation. That particular failure does not give any easy answers or places to start looking for the cause of the failure. It was also noted that there are rods with higher and lower power histories in the core that run to completion. After checking some of the more likely models for the fault without answer, we moved to dismantling the input file. Removing the models bit-by-bit, as we could, it was found that two sub-models within MechUO2 seemed to be causing the fail. When these models, model_creep and isotropic_cracking, are set to

false the simulations run to completion. There was another Watts Bar rod that did not fail that was sent to BISON at the same time. Using this rod a comparison was done with the models running and not running. Model_creep and isotropic_cracking are responsible for calculating more realistic strains and material deformation in the fuel. The comparison of the, above mentioned, simulations showed that the thermal properties were pretty close but the mechanical properties, such as stress and gap width evolution were not. This was not unexpected though. A better answer as to what is truly causing the convergence issues will require a much deeper look in to the material models. It was been suggested that these are "edge cases", as in they are running at the edge of a model's range or at a place where the correlations change for example. This is just speculation at this point but further investigation will continue.

1.5D BISON

A CASL milestone was submitted for a 1.5D version of BISON. The details of this request were reported in the milestone report for 1.5D, the purpose for this report is to report on BISON support. When 1.5D was implemented, BISON started testing the feature and originally found issues, such is expected in major new features. On a side note this request came at a time where a transition from solid mechanics to tensor mechanics was underway. Due to this the 1.5D was built in tensor mechanics only. This matters because not all of the current BISON has been converted, meaning that some material model are not available yet, and the new input file for tensor mechanics is quite different looking.
As for the testing of 1.5D, BISON set up a couple simple test cases, a short rod and a full rod, and were seeing substantial speed increases (roughly 83x). The examples were sent to CASL where Shane Stimpson made a representative input file to test. BISON assisted in getting the input file functional and the final results showed around a 3x difference in runtime. The 2D non-tensor mechanics simulation took 420 seconds on 8 processors and the 1.5D tensor mechanics simulation took 137 seconds on 8 processors, both were run on a 2014 iMac Pro desktop. Comparison work for the 1.5D is continuing and BISON intends on validating 1.5D against our their current 2D-RZ assessment simulation bank in the future.

Miscellaneous Activities

Over the past 6 months INL-BISON and CASL have worked closely on a myriad of problems and issues. One such service that BISON has provided is continuous feedback on simulation input files and the use of material models. Whenever an input was sent to BISON it was checked for usage consistent with the current INL-BISON standards, and whenever syntax or a model of interest to CASL was changed, CASL was notified. A more through, line-by-line, input check was initiated in March at INL and the results and questions were reported back to CASL. In looking at input files that were sent, some questions about what values were being used for fuel relocation and maximum material iterations came up and were relayed to CASL for discussion. In another instance questions were asked about if BISON could calculate

decay heat and how it does this. The question was answered and the user was directed to an example problem for input syntax. Another collaboration example is that BISON members were asked if they would like to co-author a journal article with CASL-ORNL. A couple of the BISON team accepted and co-author reviews have been submitted.

CASL personnel requested the capability for Bison to read fission rate from MPACT and use it as input to the BurnupFunction class instead of the user-defined functions for rod average linear power and axial profile. A new function is in development that meets these requirements. We plan to merge this new function into MOOSE sometime in mid-March, and then modify BurnupFunction to utilize this new feature. A testable version should be ready in late March.