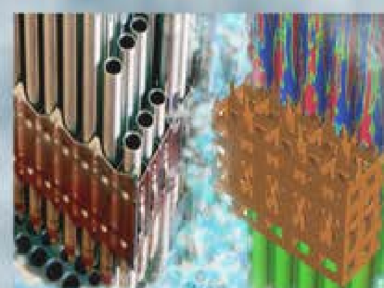
CASL-U-2015-0021-000

Contact Memory Report
CASL.FY14



Rich Williamson
Idaho National Laboratory

January 31, 2015

U.S. DEPARTMENT OF ENERGY | Nuclear Energy

CASL-U-2015-0021-000

Contact Memory Reduction

CASL FY15 Letter Report

January 30, 2015

Introduction

As documented in a prior CASL milestone report [1], work has been underway to transition to a new system of contact enforcement in MOOSE, and by extension, the fuel performance codes BISON and BISON-CASL, which are based on MOOSE. This new system, known as the Constraint system, achieves significantly better solution convergence than the prior system because it allows for the full set of terms related to contact enforcement to be included in the Jacobian matrix that is used for preconditioning.

While this system provides a significantly more accurate preconditioning matrix, that matrix is also more dense than the matrix produced by the prior system, and it has been found that the algebraic multigrid preconditioning methods that are commonly used for other MOOSE problems do not work well with that matrix because important terms are coarsened. Good convergence can be obtained using the SuperLU parallel direct solver or an additive Schwartz preconditioner. Unfortunately, the size of the models that can be run with these strategies has been limited because their implementation has resulted in excessive memory usage.

Automatic patch update

One of the major problems leading to excessive memory usage is the way that contact searches have been handled in MOOSE. The goal of the contact search is to find the element face that each node on the slave contact surface should be constrained to. To locate those faces, the contact search finds the nearest n nodes on the master face to a given slave node, where n is a user-supplied parameter that defines the contact patch size. The contact algorithm then loops through the faces connected to those nodes and finds the face that is the strongest candidate. As the bodies on either side of the contact interface move relative to each other, the slave node can come in contact with other faces in the patch of nodes.

Until recently, MOOSE only provided the option of forming the contact patch once at the beginning of the analysis. For problems involving sliding across many faces (such as a full-length fuel rod), this meant that a large patch had to be used to ensure that the node would not slide outside the patch. If a node slid outside the patch, contact would no longer be enforced.

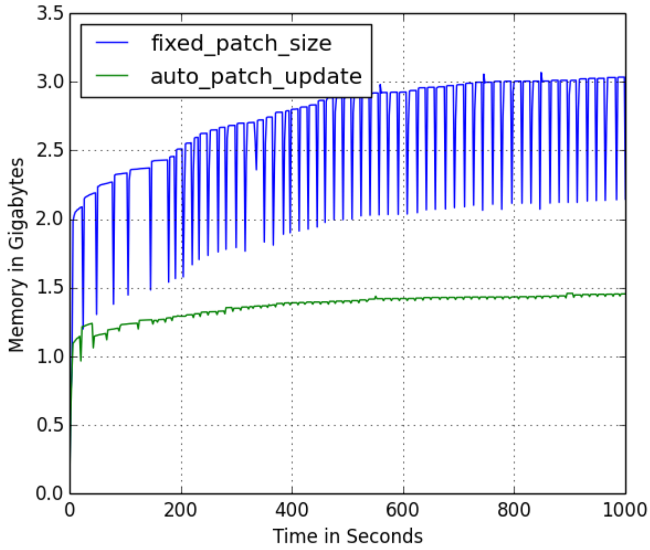


Figure 1: Comparison of memory usage on full-length 2D fuel rod model, with a fixed contact node locator patch with a size of 40 nodes, and the automatic patch update strategy with 3-node patch.

This is not a problem with a sufficiently large patch, but using a large patch has a side effect that it results in a large set of off-diagonal entries in the preconditioning matrix with the Constraint system. This is because allowance needed to be made for every possible node/face interaction that might occur during the analysis.

A new option has been added to MOOSE to permit updating of the patch of nearest nodes during the analysis. This permits the use of a much smaller patch than was previously required, especially for models of full-length fuel rods with refined meshes, where a node on one side of the contact interface is likely to slide past a large number of faces during the analysis. When any slave node nears the edge of the current patch, an update of the contact search is triggered, and the preconditioner is reinitialized. This results in a significant memory reduction.

To demonstrate this capability, a full-length 2D axisymmetric fuel rod model was run using both a fixed patch and a dynamically update patch. The fixed patch had a size of 40 nodes, which is necessary to prevent nodes from sliding outside the patch. The dynamically updated patch had a size of 3 nodes. This model was run for 70 solution steps in both cases, and the patch was updated 6 times for the dynamic patch case. Figure 1 shows the time history of the memory usage in the two cases. As can be seen in this plot, there is roughly a 50% reduction in memory usage with the smaller patch.

In addition to reducing memory usage, this patch update strategy also results in decreased run time. This model was run on 8 cores in both cases. The total run time for the fixed patch case was 1230 s, while the total run time for the patch updating case was 1030 s. The two models had essentially identical iteration counts.

Remaining Issues

This automatic patch updating strategy has been used to successfully run some models to completion after performing many patch updates. There is, however, still a remaining bug that is occasionally causing segmentation faults to occur within the preconditioner on steps following a patch update. This issue is being actively pursued and will be resolved shortly. A relatively small test case that reproduces this problem has been developed, and is being used for debugging.

References

[1] B. W. Spencer, J. D. Hales, D. R. Gaston, D. A. Karpeev, R. L. Williamson, S. R. Novascone, D. M. Perez, R. J. Gardner, and K. A. Gamble. BISON contact improvements CASL FY14 report. Technical Report INL/EXT-14-33285, Idaho National Laboratory, Idaho Falls, ID, September 2014.