

# Acceleration of Thermochemical Calculations in Bison



Approved for public release.  
Distribution is unlimited.

Max Poschmann  
Markus H.A. Piro  
Srdjan Simunovic

January 2020

#### 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:** [report@osti.gov](mailto:report@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.

Computational Sciences & Engineering Division

## **Acceleration of Thermochemica Calculations in Bison**

Max Poschmann<sup>1</sup>, Markus H.A. Piro<sup>1</sup>, and Srdjan Simunovic<sup>2</sup>

<sup>1</sup> University of Ontario Institute of Technology, Oshawa, ON, Canada

<sup>2</sup> Oak Ridge National Laboratory, Oak Ridge, TN

Date Published: January 2020

Prepared by  
OAK RIDGE NATIONAL LABORATORY  
Oak Ridge, TN 37831-6283  
managed by  
UT-Battelle, LLC  
for the  
US DEPARTMENT OF ENERGY  
under contract DE-AC05-00OR22725



## CONTENTS

ABSTRACT . . . . .	1
1. INTRODUCTION . . . . .	1
2. THERMOCHEMICA OVERVIEW . . . . .	2
3. RE-INITIALIZATION METHOD . . . . .	3
4. BISON IMPLEMENTATION . . . . .	4
4.1 ThermochemicaNodalData (MOOSE UserObject) . . . . .	4
4.2 ThermochemicaNodalUO2X (MOOSE UserObject) . . . . .	4
4.3 UO2XChemistryTransportSimplified (MOOSE AuxKernel) . . . . .	6
5. DEMONSTRATION PROBLEMS . . . . .	6
5.1 Stand-Alone Thermochemica Calculations . . . . .	6
5.1.1 Carbon-Oxygen System . . . . .	6
5.1.2 Irradiated Nuclear Fuel . . . . .	8
5.2 Bison Example Calculations . . . . .	8
5.2.1 1D Bison Calculation . . . . .	8
5.2.2 3D Bison Calculation . . . . .	9
6. CONCLUSION . . . . .	10
7. ACKNOWLEDGEMENTS . . . . .	10
8. REFERENCES . . . . .	11

## ABSTRACT

The performance of the open-source thermochemical software library Thermochemica has been enhanced by developing a re-initialization algorithm to make use of data from previous calculations to reduce the number of convergence steps in the Gibbs energy minimization procedure. This algorithm has been tested in the context of stand-alone Thermochemica calculations, and speedups in the range of 2x-3x achieved for cases with chemistries resembling those of irradiated nuclear fuels. Routines to make use of this re-initialization procedure have been implemented in Bison, in which each node uses the results of the previous calculation at that node as the initial conditions for the following Thermochemica call. Examples based on calculating the diffusion of oxygen in UO<sub>2</sub> Light Water Reactor (LWR) fuel in 1D and 3D have been tested, with speedups up to 7.38x demonstrated.

## 1. INTRODUCTION

The open-source thermochemical library Thermochemica [7, 11] has previously been coupled with the Multi-physics Object-Oriented Simulation Environment (MOOSE) [3] Finite Element Method (FEM) app Bison [5] for the purpose of providing thermodynamic information to simulations of nuclear fuels, such as phase fractions, chemical potentials, concentrations of species, vacancies and interstitials. In the initial implementation, each call to Thermochemica by Bison was independent of all previous such calls, and each thermochemical calculation begins anew. However, in typical Bison simulations, the change in system parameters (i.e., chemical element concentrations, temperature, and pressure) between calls to Thermochemica is very small, and thus the calculation results are typically very similar. Here, we implement a method for initializing Thermochemica calculations by making use of the results from previous calculations, and demonstrate that considerable speed-up is achieved when Bison simulations make use of this new initialization method. This speedup is crucial for simulations of nuclear fuels, which require trillions of thermochemical calculations due to the large physical dimensions and long timescales involved in these applications. Further, nuclear fuel systems tend to be chemically complex (with 20 or more chemical elements often involved), which leads to each call to Thermochemica being highly computationally expensive.

In a stand-alone Thermochemica calculation, the leveling algorithm [2] is used to provide an initial estimation of the phase assemblage and the element potentials. The leveling method is a first approximation to equilibrium, which does not include mixing between species in a solution phase – all species and phases are temporarily treated as stoichiometric phases. Once this initial estimate of phase assemblage has been obtained, the Gibbs Energy Minimization (GEM) method [16] is used to determine the combination of phases and phase compositions that yields the lowest integral Gibbs energy. This is done by iteratively manipulating the quantities of species to minimize the integral Gibbs energy, adding to the assemblage those phases with a negative driving force, and removing phases from the assemblage that tend to form negative (i.e., non-physical) quantities.

In the new method implemented here, the first Thermochemica calculation at each point of interest in the Bison system (i.e., nodes or integration points) proceeds as above. However, immediately after each calculation, key chemical information provided by Thermochemica is stored by Bison on a mesh. Then, on following calls to Thermochemica, that information is reused to provide an enhanced estimation of the initial phase assemblage and species concentrations. This has two positive effects: the leveling solver may be bypassed, and (more significantly) the resulting assemblage and concentrations are typically much

closer to the equilibrium state than those provided by the leveling solver. When the GEM solver begins with a better initial estimate, it takes many fewer steps to converge, and the calculation time is greatly decreased. Of course, storage of this initialization data at each node or integration point results in increased total memory usage as well as additional memory operations; however, our benchmarks demonstrate that these costs are greatly outweighed by the reduced number of calculation cycles.

In this report, we will provide a brief overview of the standard Thermochemica calculation method (i.e. leveling procedure and GEM solver), describe the new initialization method and its implementation in Bison, and finally discuss the example problems used to benchmark the speedup provided by the improvements described here.

## 2. THERMOCHEMICA OVERVIEW

Thermochemica is a software library that determines the equilibrium combination of phases and compositions of those phases given input temperature, pressure, and concentrations of chemical elements. Thermochemica and the development of Thermochemica as well as its integration into the MOOSE/Bison framework have been well-documented in previous technical reports [7, 14, 13, 12], and journal articles [11, 8, 10, 9], as well as in the Thermochemica documentation (available on GitHub).

From its genesis, Thermochemica has been designed to perform efficiently when coupled to multi-physics codes. This is reflected in the two-step algorithm design consisting of a leveling procedure followed by GEM. The leveling solver efficiently provides an accurate initial estimate of the phase assemblage by ignoring the composition dependence of the chemical potential of all solution species. Quickly obtaining this estimate allows the GEM routine to begin much closer to the correct equilibrium phase assemblage and phase compositions than if it were initialized randomly; thus, a large number of unnecessary GEM iterations are reduced.

The approach of the GEM method is to optimize the system such that the residuals in mass balance equations and the minimization of total Gibbs energy are simultaneously minimized, subject to the constraints that the Gibbs Phase Rule is respected and conservation of mass [16, 11]. This leads to the Lagrangian function [15]:

$$L = \frac{G}{RT} - \sum_{j=1}^C \tilde{\Gamma}_j (b_j - b_j^m) \quad (1)$$

where  $G$  is the total Gibbs energy,  $R$  is the ideal gas constant,  $T$  is the temperature,  $C$  is the number of components\* in the system,  $\tilde{\Gamma}_j$  is the dimensionless chemical potential of component  $j$ ,  $b_j$  is the number of gram-atoms of component  $j$ , and  $m$  is the GEM iteration index. Each GEM solver iteration requires the calculation of the Hessian (second partial derivative matrix) of  $L$ , which forms an  $N \times N$  system where  $N = C + \Phi$ , where  $\Phi$  is the total number of stable phases. This elucidates why the GEM solver is the most expensive component of Thermochemica as well as why calculation expense scales rapidly with the number of system components in the system.

As an equilibrium thermodynamics code, Thermochemica explicitly does not consider chemical kinetics, but rather assumes instantaneous local equilibrium. While this assumption is not strictly accurate, it is not a poor approximation for nuclear fuel applications because in these systems the temperature is typically high

---

\*The system components are typically taken as the chemical elements, but can be interpreted as integer combinations of the chemical elements.

and chemical elements are well mixed due to the stochastic nature of nuclear fission. In multiphysics simulations involving Thermochemica and Bison, all kinetics are handled within Bison, and Thermochemica calculations consider only the instantaneous state of the system when called. However, the kinetics are driven by variables provided by Thermochemica, such as the chemical potentials of diffusing species. This approach follows previous work in coupling Thermochemica with Bison for calculations of oxygen diffusion in nuclear fuels [14].

### 3. RE-INITIALIZATION METHOD

While the leveling procedure has proven to be a major performance enhancer for Thermochemica, neglecting the compositional dependence of chemical potential leads to discrepancies in the estimated phase assemblage, compositions, and chemical potentials that are in many cases greater than the changes in those variables between two consecutive evaluations in time within a multi-physics framework. Even small improvements to the accuracy of the initial values of these variables can lead to a considerable reduction in the number of trial assemblages and compositions calculated within the GEM subroutine, which is by far the most computationally expensive section of Thermochemica. When the savings in GEM iterations (plus the relatively small cost of the leveling solver) outweigh the overhead of storing and loading chemical data, computational efficiency can be improved by re-initializing Thermochemica with data from the previous calculation rather than making use of the leveling solver.

To bypass the leveling solver in Thermochemica, the arrays `iAssemblage`, `dMolesPhase`, `dElementPotential`, `dMolFraction`, and `dChemicalPotential` must be stored in memory and supplied in a subsequent call. The physical meanings, types, and sizes of these variables are described in Table 1. Additionally, an integer array created just for restarts, `iElementsUsed`, must be created and stored. `iElementsUsed` lists which elements were present in any non-zero amount in the calculation for which data was saved. If this differs from the current collection of elements, the restart attempt is aborted. The suffix `_Old` is added to each parameter when they are saved, and the variables with this prefix are held in the module `ModuleRestart`. Additional flags to track the status of the restart procedure are also available. These include `iRestartRequested`, which the user must set to `.TRUE.` if restarting is desired, `iRestartAvailable`, which is set to `.TRUE.` after restart data is successfully loaded, and `iRestartLoaded`, which is set to `.TRUE.` after a restart is successfully loaded.



**Table 1. Summary of variables required to re-initialize a Thermochemica calculation**

Variable Name	Array Dimension	Description
nElements	1	An integer scalar representing the number of elements in the system.
nSpecies	1	An integer scalar representing the number of species in the system.
iAssemblage	1:nElements	An integer vector representing the indices of the phases predicted to coexist at equilibrium.
dMolesPhase	1:nElements	A double real vector representing the number of moles of coexisting phases at equilibrium (corresponds directly to iAssemblage).
dElementPotential	1:nElements	A double real vector representing the element potentials.
dMolFraction	1:nSpecies	A double real vector representing the mole fraction of each species in the system data-file.
dChemicalPotential	1:nSpecies	A double real vector representing the chemical potentials of all species in the system.

## 4. BISON IMPLEMENTATION

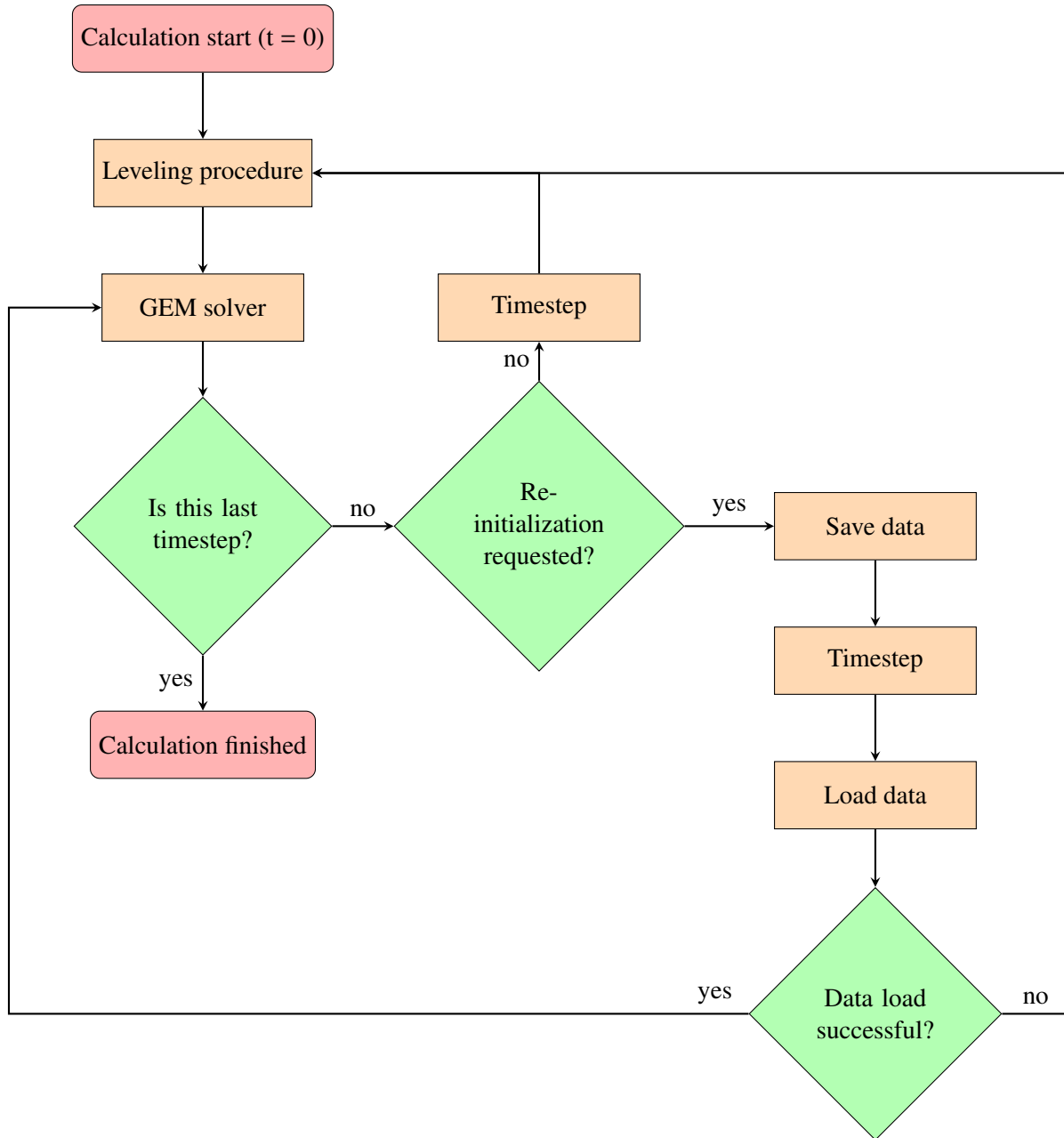
The implementation of Thermochemica re-initialization in Bison builds upon previous efforts to couple Thermochemica calculations into the MOOSE framework [14, 13, 12]. In addition to the creation of new interfaces to accommodate re-initialization, the existing Thermochemica implementations were generalized and recast into the form of a UserObject during this work. This will enable Bison users to make use of Thermochemica for any future simulations with minimal software development effort. The base UserObject, as well as an example derived UserObject and corresponding AuxKernel used for the demonstration problems discussed in Section 5.2 are described below.

### 4.1 ThermochemicaNodalData (MOOSE UserObject)

This class provides the base interface with Thermochemica for Bison. The input parameters are a vector containing the amount of each element, scalars for pressure and temperature, and optionally a switch for turning off the re-initialization algorithm (this algorithm is used by default). On each execute (which occurs when specified by the user on input), this UserObject inputs the temperature, pressure, and element concentrations to Thermochemica as well as any available re-initialization data, calls Thermochemica, then stores re-initialization data for the next time-step at each node in the FEM mesh.

### 4.2 ThermochemicaNodalUO2X (MOOSE UserObject)

The ThermochemicaNodalUO2X UserObject is an example specialization of the ThermochemicaNodalData UserObject described previously, designed to perform calculations for the



**Figure 1. Flowchart outlining re-initialization procedure.**

$\text{UO}_{2\pm x}$  system. This UserObject makes use of the `PresetElementMass` function in the Thermochemica interface to overwrite the mass of oxygen from the elements `AuxVariable` used in the base `ThermochemicaNodalData` with the value in the oxygen variable. The Thermochemica calculation is performed by the base class, and then Thermochemica outputs of interest are extracted via various accessor functions provided by the Thermochemica interface. These are then saved in member variables and made available to other objects via `get-` accessor functions.

Future MOOSE/Bison objects that will interface with Thermochemica can be written following the example provided by this class. In particular, these should make use of the program structure used in `execute()` here:

1. preset masses for any elements for which this is necessary,
2. call base class `execute()`,
3. extract any required Thermochemica outputs and make them available to other objects.

### 4.3 UO2XChemistryTransportSimplified (MOOSE AuxKernel)

UO2XChemistryTransportSimplified is an example of how results from the Thermochemica calculations performed by UserObjects that are specializations of ThermochemicaNodalData can be used to calculate values of AuxVariables. In this case, ThermochemicaNodalUO2X is a UserObject that extracts data such as oxygen interstitial and vacancy site fractions, and chemical potential from a Thermochemica calculation. This AuxKernel copies that data to the relevant AuxVariables, which are later used to calculate the chemical potential and mobility of oxygen within  $\text{UO}_{2\pm x}$  fuel, and this in turn is used to calculate the diffusion of oxygen within the fuel.

## 5. DEMONSTRATION PROBLEMS

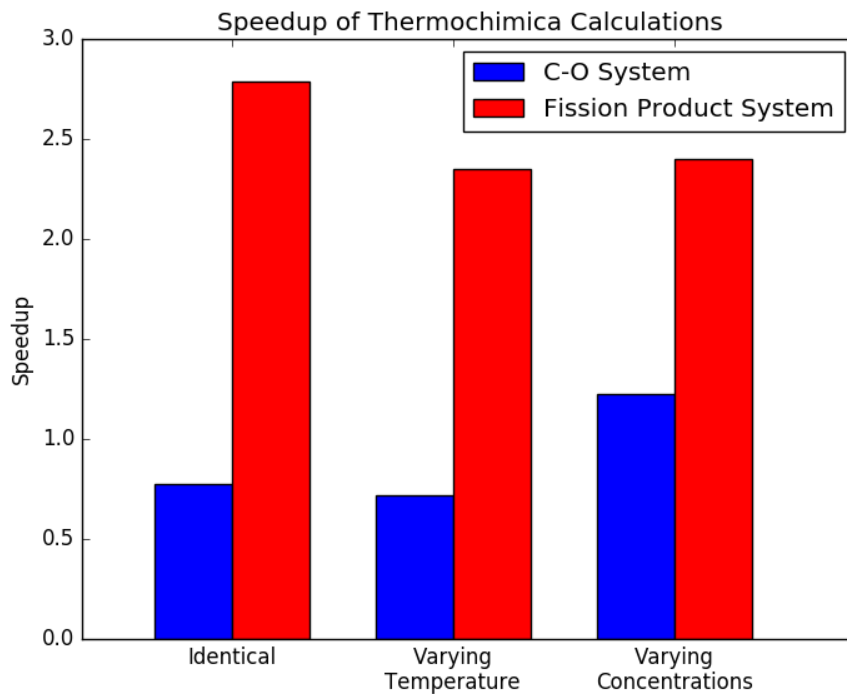
A number of demonstration problems were developed to determine the degree of speedup provided by the re-initialization algorithm, as well as to ensure that use of this algorithm does not alter the calculated results.

### 5.1 Stand-Alone Thermochemica Calculations

The first test cases were for Thermochemica in stand-alone execution (i.e., no coupling). Two example systems were chosen for testing: a very simple carbon-oxygen system containing only C and O, and a much more complex nuclear fission product system including Pu, Np, U, Nd, Pr, Ce, La, Ba, Cs, Xe, I, Te, Pd, Rh, Ru, Tc, Mo, Zr, Y, Sr, Rb, O, and H. For each system timing was performed with and without enabling the re-initialization algorithm for three cases: repetition of an identical calculation; calculations with varying temperature; and calculations with varying element concentrations. The resulting speedups determined from each of these calculations are shown in Figure 2.

#### 5.1.1 Carbon-Oxygen System

The carbon-oxygen example system contains 2 elements, 1 solution phase, and 4 pure phases. This is an extremely simple system, and therefore all Thermochemica calculations on this system are very brief. To produce a test approximately 1 minute in length, 2 million calls to Thermochemica must be made. This is naturally a highly unfavorable test case for the re-initialization algorithm, as a relatively large amount of calculation time is spent saving and loading data, and relatively little in the GEM solver loop. Therefore it is not surprising that small speedups or even slowdowns are provided by re-initialization in this case. This illustrates why re-initialization has been implemented as a setting available to the user rather than imposed



**Figure 2. Speedup due to use of the re-initialization method for stand-alone Thermochemical calculations of the carbon-oxygen and fission product example systems.**

automatically on all calculations: experienced users should be able to identify whether their combinations of system and available hardware makes re-initialization likely to be favorable, and use or omit it accordingly.

The base conditions for the carbon-oxygen example system were a temperature of 900 K, pressure of 1 atm, 1 mole of oxygen, and 2.4 moles of carbon. For the first test these conditions are unchanged between calculations. For the second case, the temperature was varied smoothly from 600 K to 1600 K over the 2 million calculations. In the third case, the amount of oxygen was varied smoothly from 1 mole to 6 moles over the calculation series. Surprisingly, only this last test shows speedup. Further testing revealed that this is because positive speedup is achieved for higher oxygen to carbon ratios. When there is more oxygen than carbon in the system then no pure condensed carbon phase is stable at 900 K, and the GEM solver takes a larger number of steps to converge. When the first test was run with 5 moles of carbon and all other conditions unchanged, the speedup was 1.16 $\times$  rather than the 0.78 $\times$  for the original test.

The lack of speedup – or even reduction in performance – in these cases can be understood when the number of GEM solver calls is considered. In the case of the identical calculation test, re-initialization results in exactly the same number of calls to the GEM solver as were made without re-initialization. Thus no speedup is provided by the algorithm, and the overhead of saving and loading the required re-initialization data results in an overall slowdown of the calculation.

### 5.1.2 Irradiated Nuclear Fuel

The second example system for stand-alone Thermochemica calculations made use of a database of uranium dioxide fuel with fission and activation products [1]. This database contains 23 elements, 15 solution phases, and 240 pure phases. Thermochemica calls for this system are much more expensive than for the carbon-oxygen system, and to keep the tests approximately the same length only 800 calculations were run per test for this example.

The base conditions for this example were a temperature of 900°C, pressure of 1 atm, and element concentrations as summarized in Table 2. These were again kept constant for the first test. For the second, the temperature was varied linearly from 600°C to 1400°C over the 800 calculations. In the third, the concentrations of all elements except U and O were varied linearly from 0 moles to 8× their base amounts over the calculation series. This very loosely approximates the evolution of fission products during burnup of uranium dioxide nuclear fuel. All three of these tests showed significant speedups, ranging from 2.62 to 2.97. For this system, the highest speedup was achieved for repetition of a calculation under constant conditions. This is the behaviour anticipated for the vast majority of cases: re-initialization ought to work best in cases where the conditions (and thus usually results) of a calculation are very similar to that of the preceding calculation. In contrast to the C-O system calculations, the number of calls to the GEM solver is greatly reduced when re-initialization is employed for this system. For the identical calculation example, the number of GEM solver calls is reduced by 76.2%, which implies that the GEM portion of the calculation is sped up approximately 4.2×, and the save/load overhead as well as non-accelerated portions of the calculations reduce the overall speedup to the 2.97× quoted above.

## 5.2 Bison Example Calculations

Previous coupling of Thermochemica to Bison has focused on calculations of the distribution and chemical potential of oxygen within the fluorite phase of uranium dioxide ( $\text{UO}_{2\pm x}$ ) nuclear fuel [14, 13] for Light Water Reactor (LWR) applications. These example Bison calculations extend this work and demonstrate the reduction in simulation time for Bison objects that call Thermochemica when the re-initialization algorithm is employed.

These examples made use of a limited dataset containing only 3 chemical elements (O, U, Pu), 8 solution phases, and 18 stoichiometric phases, rather than that used in the above stand-alone example. Therefore based on the results of the stand-alone examples, we anticipate that greater speedup is possible for Bison simulations involving more chemically complex systems. The mobility model for non-stoichiometric uranium oxide from Moore *et al.* [6] and the thermodynamic treatment for the U-O system from Guéneau *et al.* [4] were used in these examples.

### 5.2.1 1D Bison Calculation

In the first Bison example, the diffusion of oxygen within uranium dioxide fuel with small amounts of plutonium is calculated on a 1-dimensional grid with 42 nodes. This system uses the `UO2XChemistryTransportSimplifiedAuxKernel` and `ThermochemicaNodeUO2XUserObject` described previously in Section 4. to call Thermochemica and extract important calculation results such as the concentration of interstitial oxygen and the chemical potential of oxygen. The calculation was run with

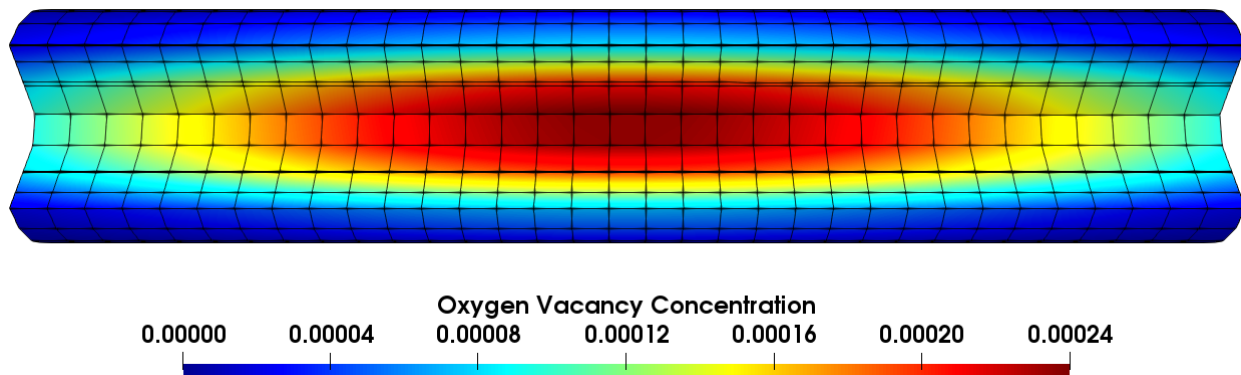
**Table 2. Base concentrations of elements in fission product example system**

Element	Moles
Pu	$1 \times 10^{-9}$
Np	$1 \times 10^{-9}$
U	$5.6849 \times 10^{-3}$
Nd	$1.8974 \times 10^{-7}$
Pr	$1 \times 10^{-9}$
Ce	$7.5956 \times 10^{-8}$
La	$3.831 \times 10^{-8}$
Ba	$9.9634 \times 10^{-8}$
Cs	$1.2584 \times 10^{-9}$
Xe	$1 \times 10^{-9}$
I	$1 \times 10^{-9}$
Te	$1 \times 10^{-9}$
Pd	$1.0003 \times 10^{-5}$
Rh	$7.3875 \times 10^{-7}$
Ru	$4.2121 \times 10^{-6}$
Tc	$1 \times 10^{-9}$
Mo	$1.4263 \times 10^{-5}$
Zr	$1.1667 \times 10^{-7}$
Y	$1 \times 10^{-9}$
Sr	$8.6762 \times 10^{-7}$
Rb	$1 \times 10^{-9}$
O	$1.1446707 \times 10^{-2}$
H	$1 \times 10^{-9}$

and without re-initialization for 2.5 million seconds of simulation time (approximately one month) with time-steps of 10,000 seconds. The speedup achieved for this example was  $2.72\times$ .

### 5.2.2 3D Bison Calculation

A similar Bison simulation was performed on a 3-dimensional cylindrical mesh containing 2176 nodes. A parabolic, non-evolving temperature gradient was applied such that the cylinder was hottest in the centre (up to 2100 K), and cooler at the ends and edges (as low as 900 K) to induce redistribution of oxygen and gradients in the other Thermochemica output variables. This calculation was performed for 10 million seconds of simulation time with time-steps of 10,000 seconds on 16 cores on a shared memory workstation, with and without re-initialization enabled. The re-initialization algorithm achieved speedup of  $7.38\times$  for this example, shortening run-time from approximately 3 hours to as little as 25 minutes. This speedup is much greater than that measured for any of the previous examples, and was possible because the number of calls to the GEM solver routine was reduced by 96.1% when the re-initialization algorithm was employed. The speedup achieved for the entire calculation was much less than the implied  $25\times$  speedup of the GEM solver due to the overhead of saving and loading re-initialization data, as well as time spent in the remaining un-accelerated portions of the calculation.



**Figure 3. Cutaway view of oxygen vacancy concentration in uranium dioxide light water reactor fuel from a 3D Bison example calculation after 10 million seconds.**

## 6. CONCLUSION

A method of re-initializing Thermochemica calculations within Bison based on the results of previous calculations has been developed. Subroutines have been provided to make use of this functionality both in the context of Thermochemica as a stand-alone software package and when Thermochemica is employed as a module in the MOOSE/Bison framework. We have developed a series of tests based on realistic nuclear fuel applications that demonstrate significant speedup of calculations due to the re-initialization algorithm, ranging from approximately  $2.5\times$  for most cases to as much as  $7.38\times$  for the largest test case: a 3-dimensional simulation in Bison of a uranium dioxide fuel element.

Along with these performance improvements, the generalization of the Thermochemica interface in Bison developed here will enable Bison developers to make use of Thermochemica in their calculations more efficiently and with less development overhead. `ThermochemicaNodalData` provides a common base class for all future development of MOOSE/Bison objects requiring Thermochemica calculations at nodal points in the FEM mesh. `ThermochemicaNodalUO2X` is an example derived `UserObject` that customizes the call to Thermochemica and stores Thermochemica output variables in a manner that allows access by `Kernels` and `AuxKernels`. `UO2XChemistryTransportSimplified` is an example of such an `AuxKernel`, and uses Thermochemica-calculated values such as the chemical potential of oxygen and the oxygen interstitial fraction to determine the rate of oxygen diffusion in uranium dioxide.

## 7. ACKNOWLEDGEMENTS

Research was sponsored by the U.S. Department of Energy, Office of Nuclear Energy, Nuclear Energy Advanced Modeling and Simulation Program and Fuel Cycle R&D Program, under contract DE-AC05-00OR22725 with UT-Battelle, LLC.

## 8. REFERENCES

1. E. C. Corcoran, M. Kaye, and M. H. A. Piro. An overview of thermochemical modelling of CANDU fuel and applications to the nuclear industry. *Calphad*, 55:52–62, 2016.
2. G. Eriksson and W. T. Thompson. A procedure to estimate equilibrium concentrations in multicomponent systems and related applications. *Calphad*, 13(4):389–400, 1989.
3. D. Gaston, C. Newman, G. Hansen, and D. Lebrun-Grandie. MOOSE: A parallel computational framework for coupled systems of nonlinear equations. *Nuclear Engineering and Design*, 239(10):1768–1778, 2009.
4. C. Guéneau, N. Dupin, B. Sundman, C. Martial, J. C. Dumas, S. Gossé, S. Chatain, F. De Bruycker, D. Manara, and R. J. M. Konings. Thermodynamic modelling of advanced oxide and carbide nuclear fuels: Description of the U-Pu-O-C systems. *Journal of Nuclear Materials*, 419(1-3):145–167, 2011. ISSN 0022-3115. doi: 10.1016/j.jnucmat.2011.07.033.
5. J. D. Hales, S. R. Novascone, R. L. Williamson, D. R. Gaston, and M. Tonks. Solving nonlinear solid mechanics problems with the Jacobian-free Newton Krylov method. *Computer Modeling in Engineering and Sciences*, 84(2):123, 2012.
6. E. Moore, C. Guéneau, and J.-P. Crocombette. Diffusion model of the non-stoichiometric uranium dioxide. *Journal of Solid State Chemistry*, 203:145–153, 2013. ISSN 0022-4596. doi: 10.1016/j.jssc.2013.04.006.
7. M. H. Piro, S. Simunovic, and T. M. Besmann. Thermochemica user manual v1.0. Technical Report ORNL/TM-2012/576, Oak Ridge National Laboratory (ORNL), Oak Ridge, TN (United States), 2012.
8. M. H. A. Piro and S. Simunovic. Performance enhancing algorithms for computing thermodynamic equilibria. *Calphad: Computer Coupling of Phase Diagrams and Thermochemistry*, 39:104–110, 2012. ISSN 03645916. doi: 10.1016/j.calphad.2012.09.005.
9. M. H. A. Piro and S. Simunovic. Global optimization algorithms to compute thermodynamic equilibria in large complex systems with performance considerations. *Computational Materials Science*, 118:87–96, 2016. ISSN 09270256. doi: 10.1016/j.commatsci.2016.02.043.
10. M. H. A. Piro, J. Banfield, K. T. Clarno, S. Simunovic, T. M. Besmann, B. J. Lewis, and W. T. Thompson. Coupled thermochemical, isotopic evolution and heat transfer simulations in highly irradiated UO<sub>2</sub> nuclear fuel. *Journal of Nuclear Materials*, 441(1-3):240–251, 2013. ISSN 00223115. doi: 10.1016/j.jnucmat.2013.05.060.
11. M. H. A. Piro, S. Simunovic, T. M. Besmann, B. J. Lewis, and W. Thompson. The thermochemistry library Thermochemica. *Computational Materials Science*, 67:266–272, 2013. ISSN 09270256.
12. S. Simunovic and T. M. Besmann. Coupling of thermochemistry solver Thermochemica with MOOSE/Bison. Technical report, Oak Ridge National Laboratory (ORNL), Oak Ridge, TN (United States), 2015.
13. S. Simunovic, T. M. Besmann, and S. L. Voit. Benchmark problem for calculating oxygen potential in high burnup LWR fuel using the Thermochemica module in MOOSE/Bison. Technical report, Oak Ridge National Laboratory (ORNL), Oak Ridge, TN (United States), 2014.



14. S. Simunovic, S. L. Voit, and T. M. Besmann. Oxygen diffusion model using the Thermochemica module in MOOSE/Bison. Technical report, Oak Ridge National Laboratory (ORNL), Oak Ridge, TN (United States), 2014.
15. W. R. Smith and R. W. Missen. *Chemical reaction equilibrium analysis: Theory and algorithms*. Wiley, 1982.
16. W. B. White, S. M. Johnson, and G. B. Dantzig. Chemical equilibrium in complex mixtures. *The Journal of Chemical Physics*, 28(5):751–755, 1958.