Direct Sparse Matrix Linear Solver for the Nodal Kinetics in RELAP5-3D

2019 International RELAP5 Users Group Meeting

Idaho Falls, Idaho

April 18 - 19, 2019

Doug Barber
Information Systems Laboratories, Inc.

Presentation Outline

- Direct Sparse Matrix Linear Solution Implementation
- Input Manual Changes
- Verification Testing
- CPU Performance
- Direct Solver Summary
- Project Summary

Direct Sparse Matrix Linear Solution

- Purpose: Provide a direct linear solver option for when convergence of the iterative solver breaks down.
  - 3 new direct solvers added:
    - PARDISO
    - MA41
    - SuperLU_MT
  - 3 new compressed matrix storage formats added:
    - Compressed Column Storage (CCS)
    - Coordinate Matrix Storage (CMS)
    - Compressed Row Storage (CRS)
  - Solver refactorization performed
  - Switching logic added, including time step backup
- Status: Completed; Transmittal to INL pending

Direct Sparse Matrix Linear Solution

- PARDISO Solver
  - Implementation utilizes the Intel® Math Kernel Library (MKL)
  - Solution includes 4 steps:
    - Reordering and symbolic factorization (for memory allocation)
    - Actual factorization
    - Backward substitution with optional iterative refinement
    - Release memory and terminate solution

Direct Sparse Matrix Linear Solution

- MA41 Solver
  - Library provided by Science and Technology Facilities Council (STFC) in the United Kingdom
  - Written in the early 1980s with only static memory allocation.
    - Wrapper added which includes dynamic allocation
    - Resizing of scratch arrays is performed:
      - During initialization
      - Following the analysis of the sparsity pattern
      - During factorization
  - Solution includes 3 steps
    - Analysis of the sparsity pattern using minimum degree ordering
    - Matrix factorization
    - Solution and error analysis

Direct Sparse Matrix Linear Solution

- SuperLU_MT Solver
  - Multi-threaded variant of the SuperLU solver developed at Lawrence Berkeley National Lab
  - Single interface for both the factorization and the solution

Compressed Matrix Storage Formats

- Banded Matrix Storage (BMS) used for Krylov solver
- 3 new compressed storage formats:
  - Compressed Column Storage (CCS)
    - Used by SuperLU_MT
  - Coordinate Matrix Storage (CMS)
    - Used by MA41
  - Compressed Row Storage (CRS)
    - Used by PARDISO
- Mapping is performed between BMS and each of the 3 new formats

Compressed Matrix Storage Formats

- CCS stores the following arrays:
  - Real non-zero matrix values
  - Integer row indices
  - Integer non-zero count at start of each column
- CMS stores the following arrays:
  - Real non-zero matrix values
  - Integer row indices
  - Integer column indices
- CRS stores the following arrays:
  - Real non-zero matrix values
  - Integer column indices
  - Integer non-zero count at start of each row

Linear Solver Switching Logic

- Purpose: Provide user control of which solver to use during the nodal kinetics calculation
  - Utilizes the kinetics time step cards (2200-2299)
  - Allow the solver to switch between LSOR, Krylov, and direct solvers
    - Steady-state
    - Transient
  - Allow switching for both hexagonal and Cartesian
  - Preserve CMFD / NEM / TPEN selection for all solvers
  - Provide for exact restarts, even when restarting during a solver switch
  - Solver selection included in minor edits
- Status: Completed

Linear Solver Switching Logic

- Manual Switching
  - Made available through Word 4 on Cards 2201-2299

Word 4	Nodal Kinetics Solver
0	LSOR
1	Krylov (BiCGStab/GMRES)
10	SuperLU_MT
11	MA41
12	PARDISO

Linear Solver Switching Logic

- Automatic Switching
  - Available on the following two words:
    - Word 4 on Cards 2201-2299
    - Word 17 on Card 30000003
  - Selection uses format 1MMNN:
    - NN is ID of the base solver
    - MM is the ID of the alternate solver
  - Examples:
    - 10001 – Krylov is the base; LSOR is the alternate
    - 11201 – Krylov is the base; PARDISO is the alternate

Other Changes

- Solver Refactorization
  - Linear Solver
  - Non-linear Solver
  - Iteration Control
  - Convergence Checking
- Time Step Backup Logic
- Migrating Albedo Boundary Condition

Input Manual Changes

- Card 199 (The Debug Card):
  - New option "BACKKIN" added for Word 2
  - Forces nodal kinetics backup (Krylov/Direct only)
  - New automatic solver switching logic requires a time step backup
  - This debug option verifies that the nodal kinetics time step backup is being performed accurately

Input Manual Changes

- Cards 2201 – 2299 (Kinetics Time Step Data)
- Add Word 4:

W4(I) Nodal kinetics solver selection. The following options are available to the user:

0 - LSOR (Line Successive Over-Relaxation)

1 - Krylov Subspace Method (BiCGStab or GMRES can be specified via Word 13 on Card 30000010)

10 - SuperLU_MT Direct Solver

11 - MA41 Direct Solver

12 - PARDISO Direct Solver

The value input here will overwrite the solver selection from Card 30000003, Word 17.

The user has the option to select a base solver and an alternate solver using the format 1MMNN, where NN is the ID of the base solver, and MM is the ID of the alternate solver. Note that the "1" in front of "MMNN" is needed in order to provide the flexibility to use LSOR as the alternate solver, which would otherwise result in a value of "00" for "MM".

Inputting the solver selection using the 1MMNN format will activate the automatic solver switching logic. This logic will detect if the base solver (NN) failed to converge. If this is the case, the code will repeat the time step using the alternate solver (MM). At the beginning of the next time step, the solution will revert back to the base solver.

Input Manual Changes

- Card 30000003 (Nodal Kinetics Control Information):
  - Word 17 same as Cards 2201 - 2209, Word 4

W17(I) Nodal kinetics solver selection. The following options are available to the user:

0 - LSOR (Line Successive Over-Relaxation)

1 - Krylov Subspace Method (BiCGStab or GMRES can be specified via Word 13 on Card 30000010)

10 - SuperLU_MT Direct Solver

11 - MA41 Direct Solver

12 - PARDISO Direct Solver

The default value is 0. The value input here can be overwritten by the solver selection from Cards 2201 - 2209, Word 4.

The user has the option to select a base solver and an alternate solver using the format 1MMNN, where NN is the ID of the base solver, and MM is the ID of the alternate solver. Note that the "1" in front of "MMNN" is needed in order to provide the flexibility to use LSOR as the alternate solver, which would otherwise result in a value of "00" for "MM".

Inputting the solver selection using the 1MMNN format will activate the automatic solver switching logic. This logic will detect if the base solver (NN) failed to converge. If this is the case, the code will repeat the time step using the alternate solver (MM). At the beginning of the next time step, the solution will revert back to the base solver.

Input Manual Changes

- Card 30000010 (Krylov/Direct Solution Algorithm Information):
  - This card can now be input if the LSOR solver is selected.
  - The functionality made active by W16 through W19 on this card is now also available to the LSOR solver.
  - It is recommended that this card also be specified if performing nodal kinetics solver switching (see W4 of Cards 2201 - 2299).
    - This allows the Krylov/Direct parameters to be input in the event that the LSOR solver is switched to the Krylov/Direct solver during the steady-state or transient.

Input Manual Changes

- The following solution features are now available for both LSOR and Krylov/Direct solvers:
  – The TPEN nonlinear nodal solver for hexagonal geometry, including corner point discontinuity factors
  – The adjoint-based reactivity calculation, including feedback reactivity
  – The control rod decusping correction
  – Group-wise albedos (Card 30000021) and face-dependent albedos (Card 37IIJJKKF)

Input Manual Changes

- Minor edit / plot variable added: rkosol

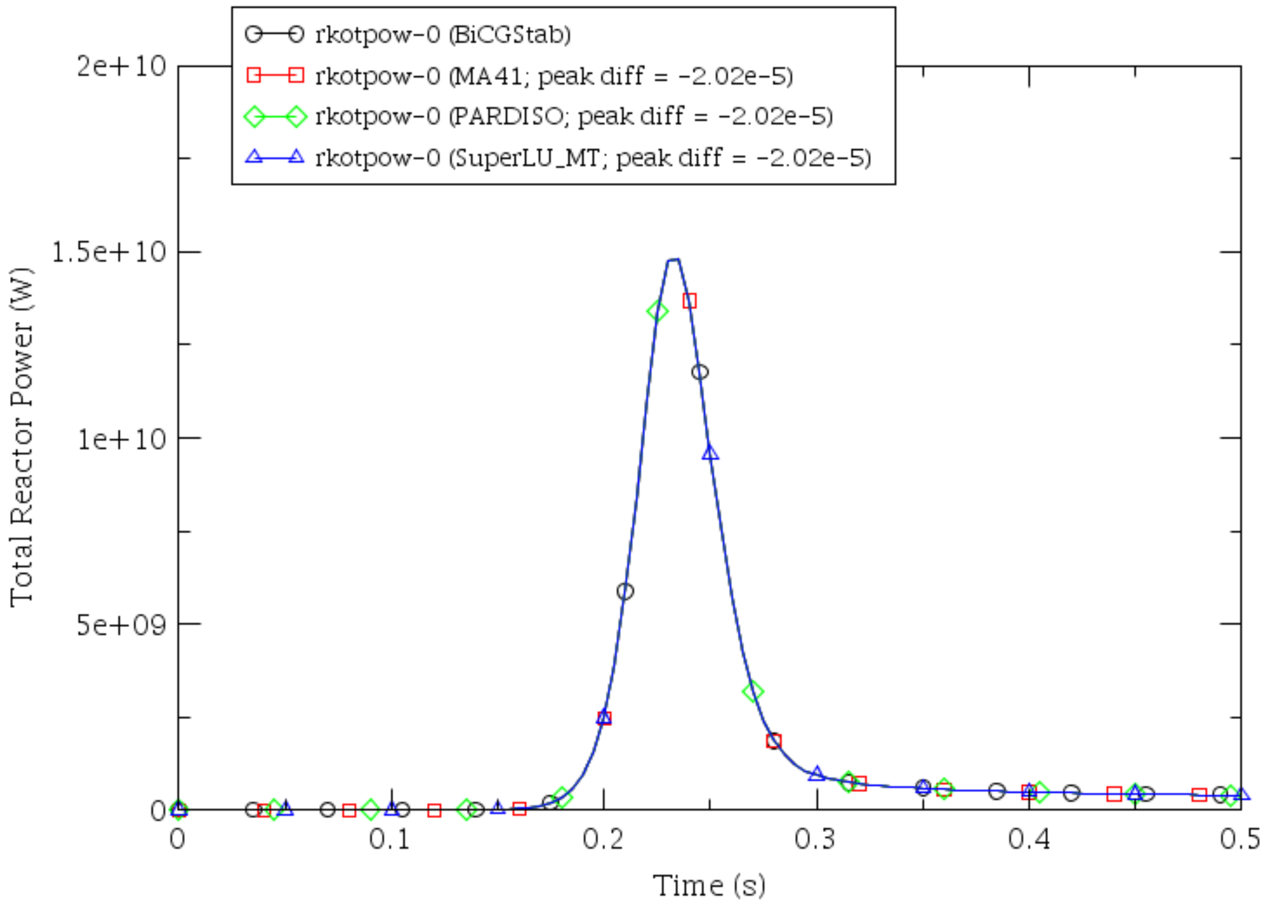
rkosol	Nodal Kinetics Solver
0	LSOR
1	Krylov (BiCGStab/GMRES)
10	SuperLU_MT
11	MA41
12	PARDISO

Verification Testing

- Unit Tests Implement and Performed
  - Matrix storage conversion for both Cartesian and hexagonal
  - Direct solve for both symmetric and nonsymmetric
- Functional Verification Tests
  - NEACRP-C1 Rod Eject (Cartesian, 2 group)
    - With and without NEM
    - Manual switch between BiCGStab and each direct solver
  - SMART-330 C1G4 (Hexagonal, 4 group)
    - Compare BiCGStab, GMRES, MA41, PARDISO, and SuperLU_MT
  - VVER440 HZP Rod Eject (Hexagonal, 2 group)
    - Compare BiCGStab, GMRES, MA41, PARDISO, and SuperLU_MT

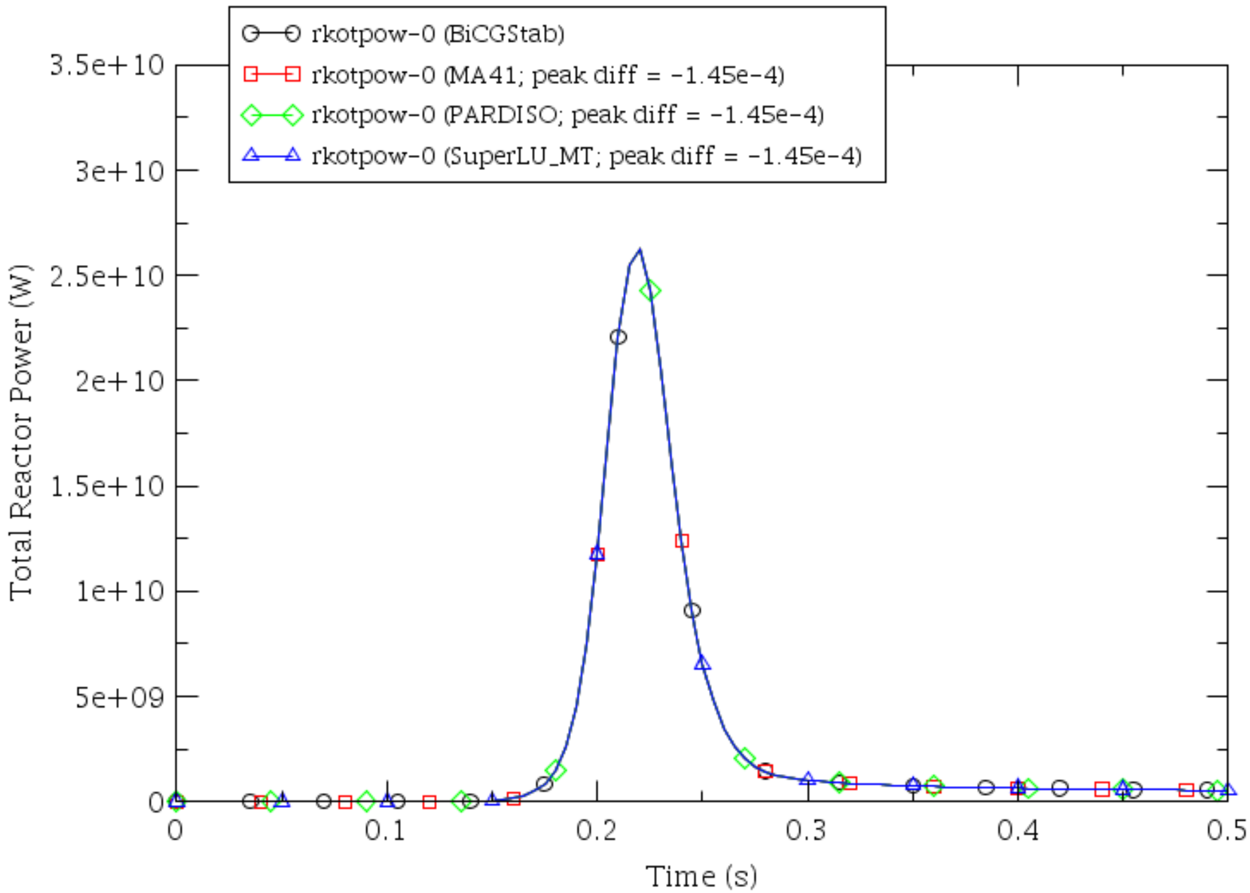
Verification Testing

- NEACRP-C1 Rod Eject (CMFD)



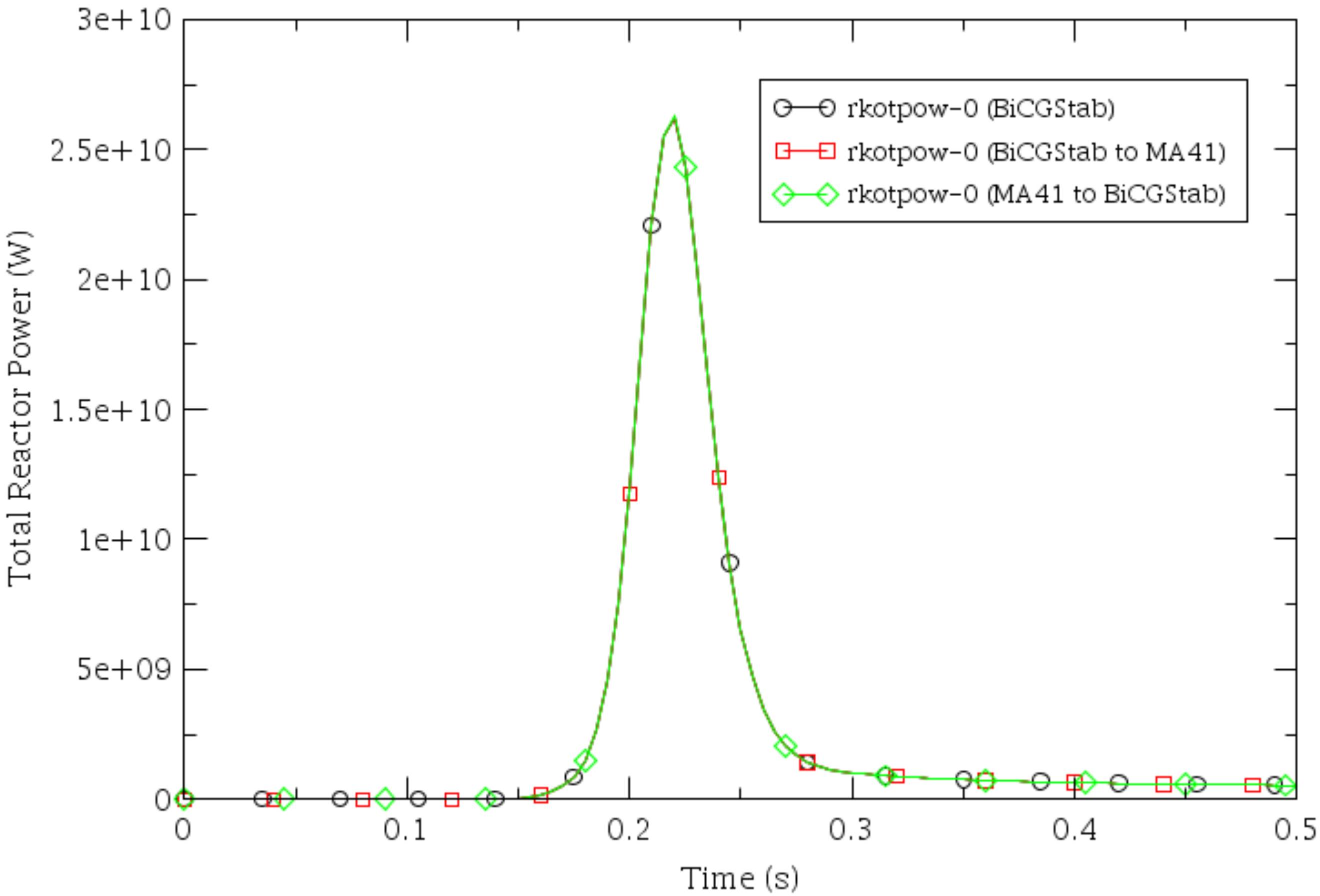
Verification Testing

- NEACRP-C1 Rod Eject (NEM)



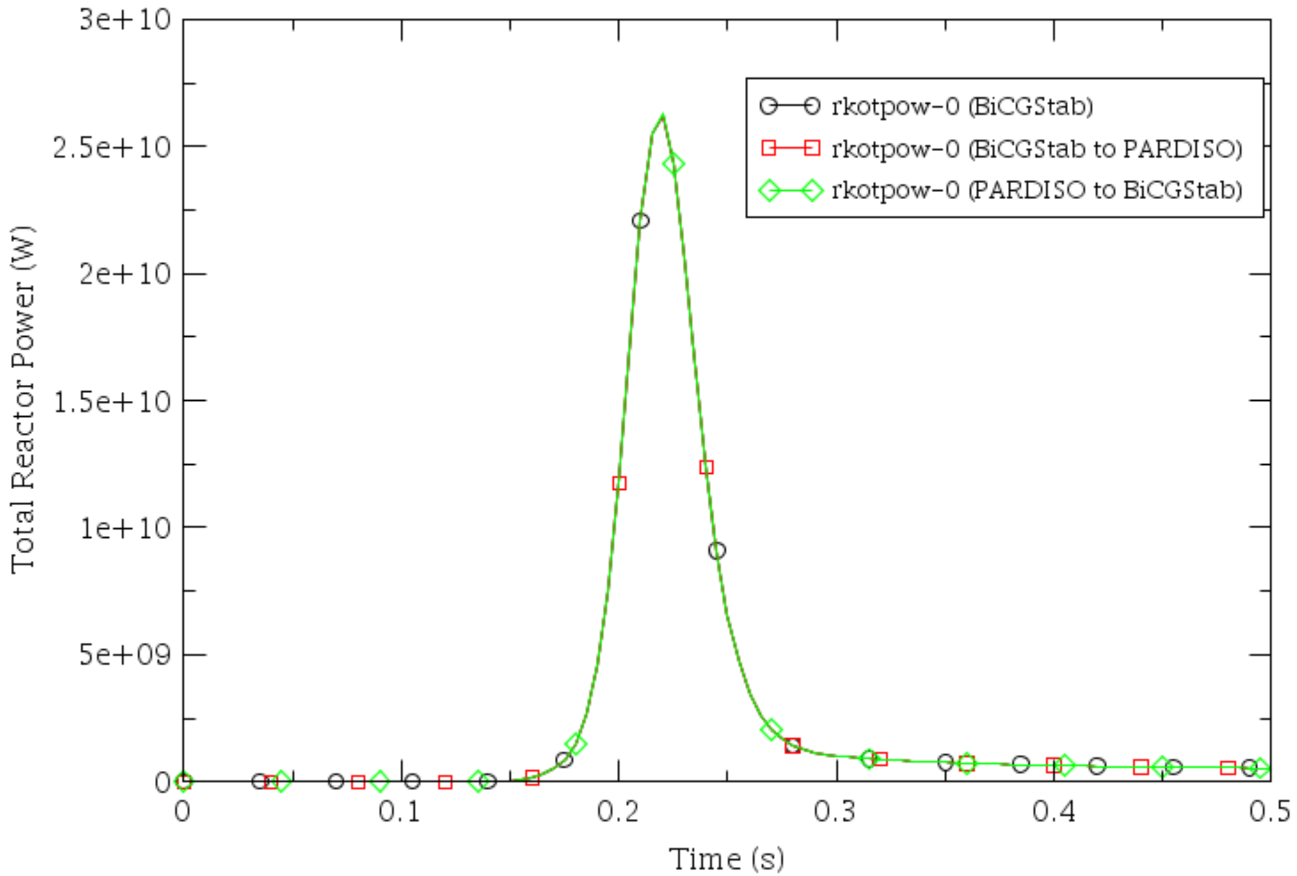
Verification Testing

- NEACRP-C1 Rod Eject (BiCGStab to MA41)



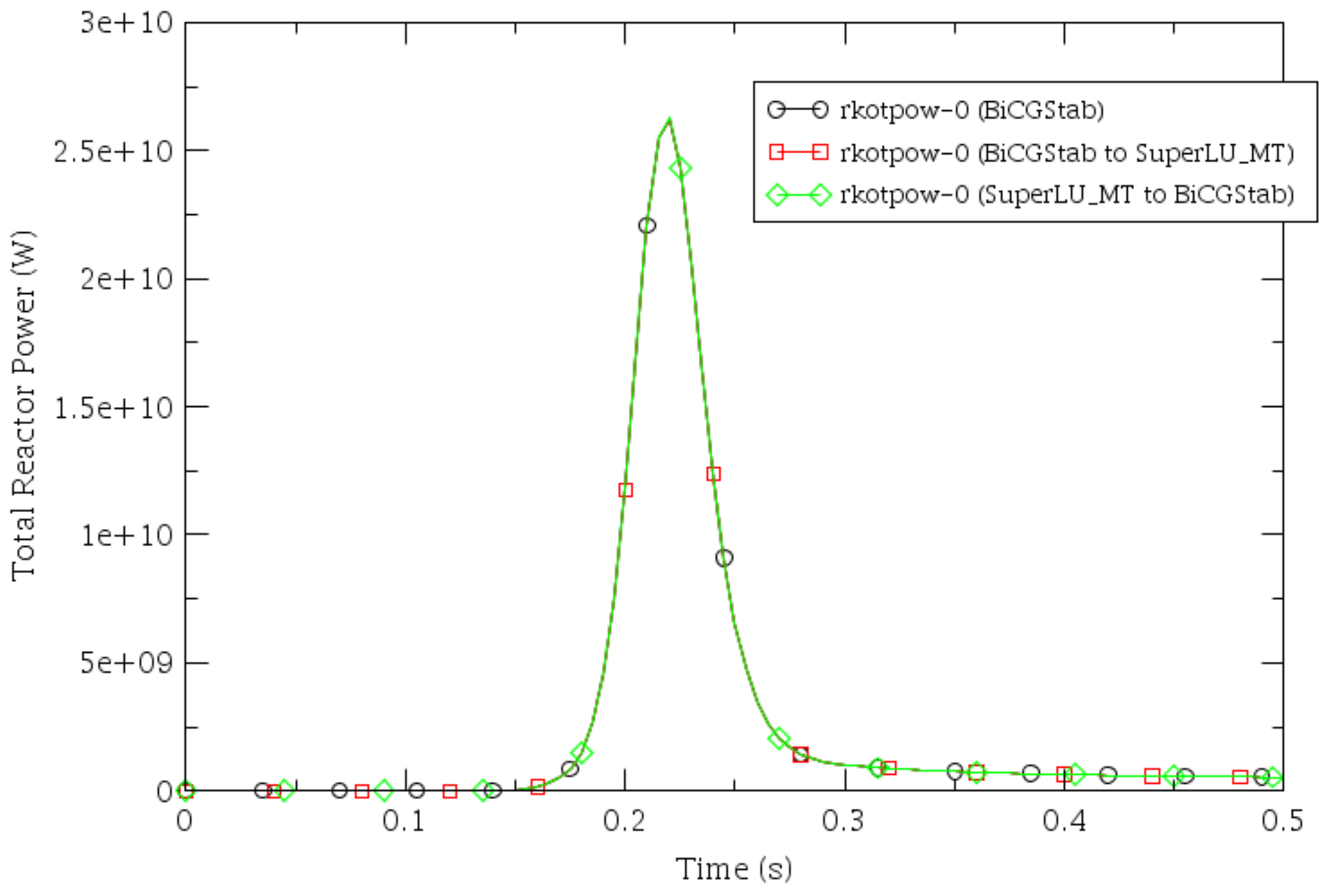
Verification Testing

- NEACRP-C1 Rod Eject (BiCGStab to PARDISO)



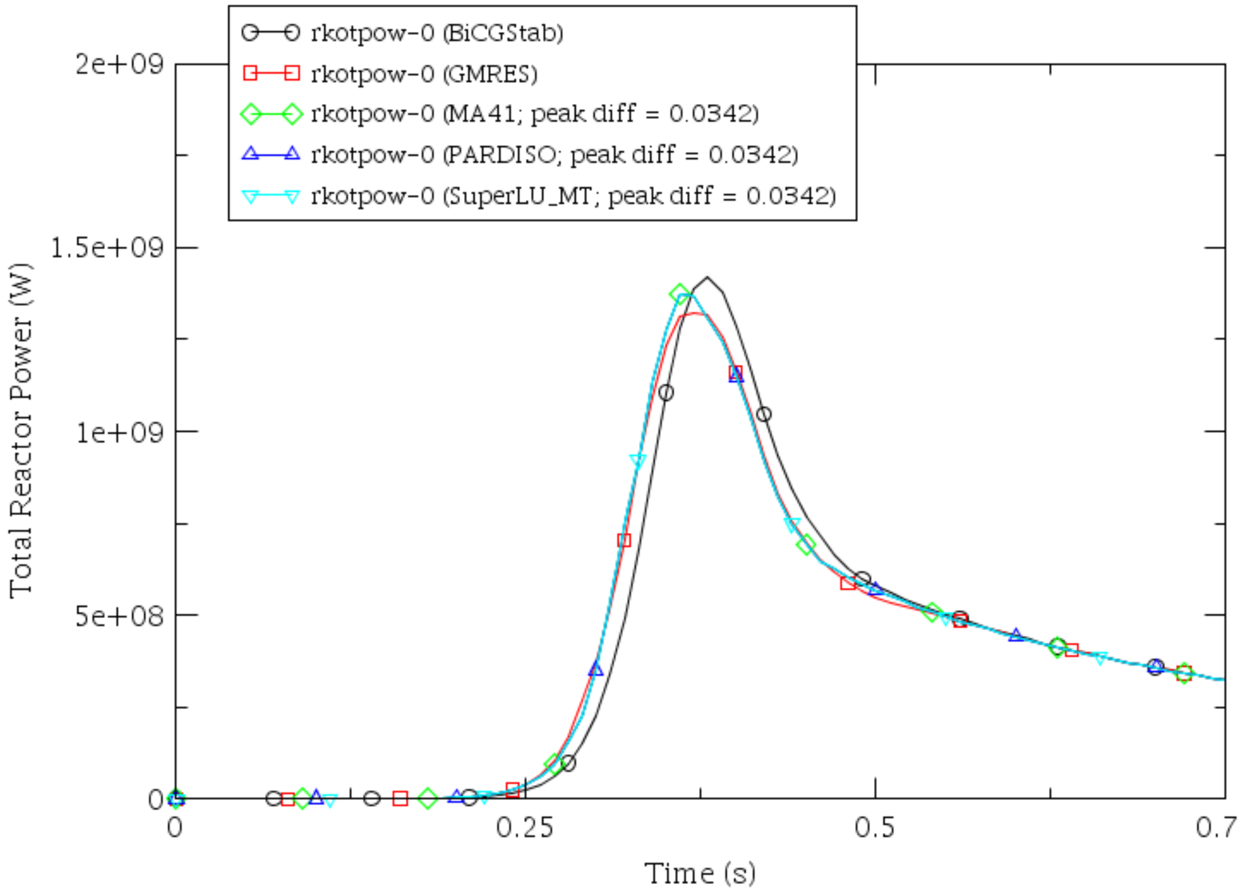
Verification Testing

- NEACRP-C1 Rod Eject (BiCGStab to SuperLU_MT)



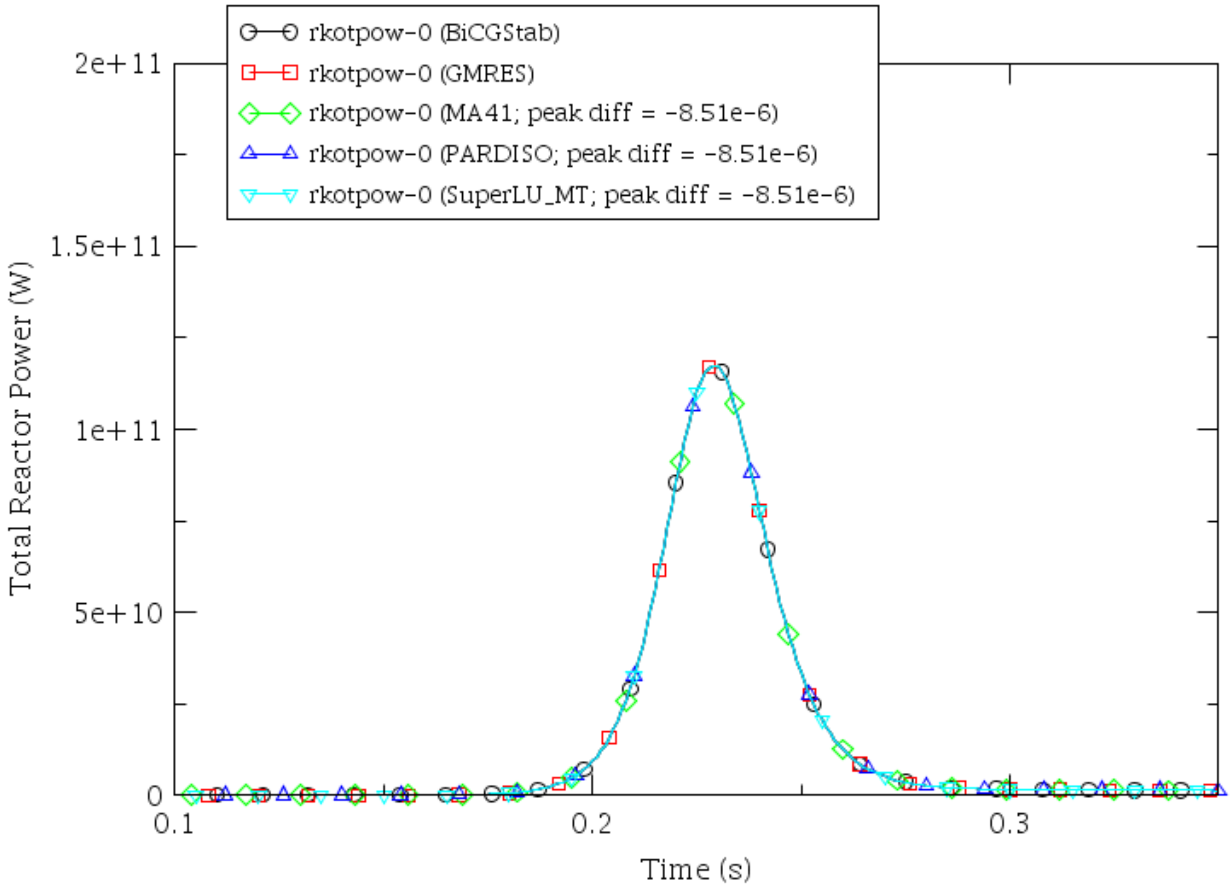
Verification Testing

- SMART-330 C1G4 Benchmark (TPEN)



Verification Testing

- VVER440 HZP Rod Eject Benchmark (TPEN)



CPU Performance

Input Cases	# of Outers	# of Nodal	LS CPU (s)	Trans. CPU (s)
neacrp-c1-krlv-cmfd.i	508	0	1.952	7.680
neacrp-c1-ma41-cmfd.i	100	0	21.426	27.541
neacrp-c1-pardiso-cmfd.i	100	0	21.588	27.851
neacrp-c1-superlu-cmfd.i	100	0	186.819	193.445
neacrp-c1-krlv-nem.i	805	568	3.207	91.080
neacrp-c1-ma41-nem.i	441	341	105.005	163.364
neacrp-c1-pardiso-nem.i	441	341	103.586	161.849
neacrp-c1-superlu-nem.i	441	341	820.827	872.850
smart330-c1g4-tr-krlv-tpen.i	7229	3255	20.820	144.185
smart330-c1g4-tr-gmres-tpen.i	9369	7989	135.314	466.402
smart330-c1g4-tr-ma41-tpen.i	9413	9390	1473.258	1858.952
smart330-c1g4-tr-pardiso-tpen.i	9413	9390	1792.285	2202.898
smart330-c1g4-tr-superlu-tpen.i	9413	9390	5367.504	5602.829
vver440-hzp-krlv-tpen_trsa.i	3398	2859	27.168	301.209
vver440-hzp-gmres-tpen_trsa.i	3227	2727	64.848	322.631
vver440-hzp-ma41-tpen_trsa.i	3225	2725	2165.288	2374.581
vver440-hzp-pardiso-tpen_trsa.i	3225	2725	1248.373	1488.011
vver440-hzp-superlu-tpen_trsa.i	3225	2725	8513.742	8640.289

Direct Solver Summary

- 3 new direct sparse matrix solvers implemented
- Verification testing showed nearly identical results with the iterative solvers
- Differences in SMART-330 C1G4 case were due to loose convergence for the iterative cases
- PARDISO generally had the best CPU performance among the 3 direct solvers
  - Generally 2 orders of magnitude longer than Krylov
- MA41 was comparable to PARDISO, but license restrictions make this less attractive
- SuperLU_MT had significantly worse performance
  - Some improvement might be had from further tuning

Project Summary

- Merged Data Structures for LSOR and Krylov
  - Code modifications completed August 2017
  - Testing completed March 2018
- Merged Non-linear Solvers
  Migrated Functionality from Krylov to LSOR
  LSOR / Krylov Solver Switching Logic
  - Code modifications (Rev. 0) completed December 2017
  - Code modifications (Rev. 1) completed April 2018
  - Testing completed April 2018
- Sparse Direct Linear Solvers
  - Code modifications / testing completed December 2018
- Final Integrated Testing
  - Completed April 2019