

LA-UR-12-24260

Approved for public release; distribution is unlimited.

Title: Verification of Unstructured Mesh Capabilities in MCNP6 for Reactor Physics Problems

Author(s): Burke, Timothy P.  
Martz, Roger L.  
Kiedrowski, Brian C.  
Martin, William R.

Intended for: MCNP website



Disclaimer:

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

# Verification of Unstructured Mesh Capabilities in MCNP6 for Reactor Physics Problems

<sup>1</sup>Timothy P. Burke, <sup>2</sup>Roger L. Martz, <sup>2</sup>Brian C. Kiedrowski, <sup>1</sup>William R. Martin

<sup>1</sup>Department of Nuclear Engineering and Radiological Sciences  
University of Michigan, Ann Arbor, MI

<sup>2</sup>Los Alamos National Laboratory  
XCP-3: Monte Carlo Codes

## 1 Abstract

New unstructured mesh capabilities in MCNP6 (developmental version during summer 2012) show potential for conducting multi-physics analyses by coupling MCNP to a finite element solver such as Abaqus/CAE[2]. Before these new capabilities can be utilized, the ability of MCNP to accurately estimate eigenvalues and pin powers using an unstructured mesh must first be verified. Previous work to verify the unstructured mesh capabilities in MCNP was accomplished using the Godiva sphere [1], and this work attempts to build on that. To accomplish this, a criticality benchmark and a fuel assembly benchmark were used for calculations in MCNP using both the Constructive Solid Geometry (CSG) native to MCNP and the unstructured mesh geometry generated using Abaqus/CAE.

The Big Ten criticality benchmark [3] was modeled due to its geometry being similar to that of a reactor fuel pin. The C5G7 3-D Mixed Oxide (MOX) Fuel Assembly Benchmark [4] was modeled to test the unstructured mesh capabilities on a reactor-type problem.

## 2 Big Ten

### *Model*

The Big Ten criticality benchmark was chosen due to its geometry being similar to that of a reactor fuel pin. Big Ten is comprised of a cylinder of low-enriched uranium inside of a cylinder of depleted uranium. The unstructured mesh models were created using Abaqus/CAE. In Abaqus/CAE, users have a variety of methods available to create their models. Users can create separate parts for various sections of their model or they can define a collection of elements, called elsets, within their part to define different material regions in their model. For the Big Ten model, three parts were created: a cylinder for the fuel, an annulus for the reflector axially surrounding the fuel, and one part for the top and bottom of the cylindrical reflector. These parts, shown in Figure 1a, are then assembled in the Assembly section of Abaqus/CAE to create the final representation of the model. Abaqus/CAE also allows various sections in an assembly to be merged to create a new part. This capability to merge parts into one larger part was tested in the study: the three parts defining the reflector and fuel regions were merged into one part composed of three elsets defining the various regions of the model. The resulting merged-part model is shown in Figure 1b.

Thus, two model types were tested: a model composed of multiple parts and a model composed of a single merged part. The difference between these two modeling methods arises when the parts are meshed. The merged part has a contiguous mesh; the mesh from one material region directly lines up with the mesh of the adjacent regions. This isn't a necessary condition in the model composed of multiple parts, and thus there can be gaps and overlaps in the geometry. The difference between these methods is best visualized when a coarse mesh is imposed on the models. Figure 2a shows a radial cross section of the multi-part and merged-part Big Ten models meshed with first-order hexahedron elements with an 18 cm global seed size.

As seen in Figures 2a and 2b, there are no gaps or overlaps in the merged-part model while there are a significant amount of gaps and overlaps in the multi-part model even though both models use the same mesh seed size. Thus, it was appropriate to test the accuracy of the unstructured mesh by running simulations using both multi-part and merged-part models. A total of 16 different unstructured mesh models were created using Abaqus/CAE with each model comprised of either first- or second-order tetrahedrons or hexahedrons. The cases considered for the simple cylindrical model are described in Table 1. Four different global seed sizes were used per element type: 1, 3, 6, and 18 cm. The seed size gives Abaqus/CAE an element size that it attempts to match, but the elements will usually not be this size exactly because of geometric conformity

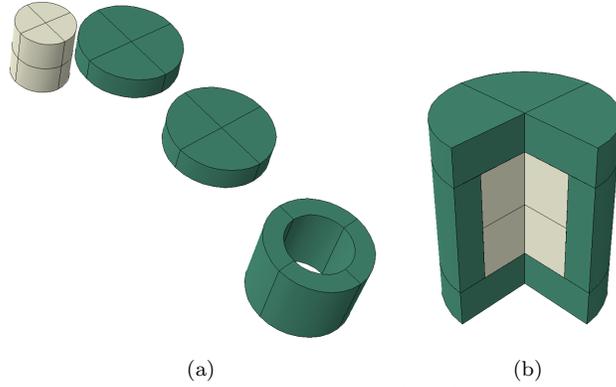


Figure 1: Various parts used in the multi-part Big Ten model with two instances of the Top/Bottom reflector (left) and one-quarter cut-away of the merged-part Big Ten model consisting of one part with two material regions (right).

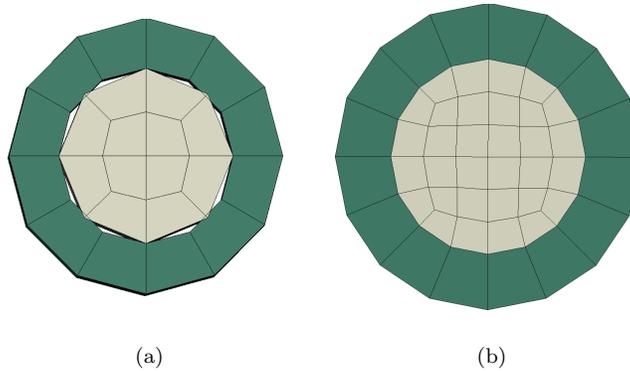


Figure 2: Top view of meshed multi-part (left) and merged-part (right) Big Ten model with a global seed size of 18 cm. Gaps and overlaps are visible in the multi-part model.

issues. All parts of the model have an axial seed size of 10 cm. This was done to prevent prohibitive runtimes with smaller seed sizes. Without the 10 cm axial seed size, element counts are on the order of hundreds of thousands for global seed sizes of interest. This reduction in element count allowed for reasonable computer times (less than 24 hours) and comparison of computing statistics for the seed sizes chosen. An example of the 3 cm meshed merged-part Big Ten model is shown in Figure 3.

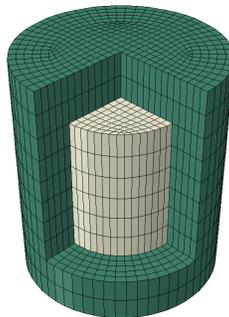


Figure 3: Cut-away view of meshed merged-part Big Ten model with a global seed size of 3 cm.

## Results

The Kcode calculations for the Big Ten model were done using 10 inactive cycles followed by 150 active cycles with 20,000 histories per cycle. The resulting eigenvalues of the various models and one standard deviation of their corresponding statistical uncertainties,  $\sigma k_{\text{eff}}$ , are shown in Table 1.

Several trends can be seen from the Big Ten results shown in Table 1. Eigenvalue calculations are highly dependent upon the mass of the fissile material in the problem, and as such the percent error in the volumes is a good predictor of how well the unstructured mesh represents the actual problem. As such, the general trend in the first-order element models is that as the number of elements increases, or the error in the representation of the volume decreases, k-effective converges on the value of the CSG model. This trend is less applicable for models meshed with second-order elements due to the element's higher degree of curvature, enabling them to accurately reproduce volumes even with a very coarse mesh. Even so, the coarse mesh with an 18 cm seed does not produce the correct eigenvalue. The larger second-order elements may preserve volume, however they do not preserve the shape of the problem as seen in Figure 4. As such, the neutron leakage in the coarse mesh differs from the CSG model and thus produces a different eigenvalue.

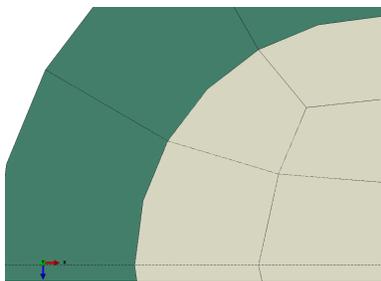


Figure 4: Close-up view of coarse second-order mesh, 18 cm seed.

One glaring result is that there is a clear difference between the resulting k-effective from the merged-part model and the multi-part model. The merged-part model converges to the same eigenvalue as the CSG model, within one standard deviation of uncertainty, while the multi-part model does not. Even though gaps and overlaps are effectively handled by the routines in the element-to-element particle tracking, the two models produce varying results due to a difference in the resulting geometries. Differences greater than 200 pcm are seen even when using fine meshes with over 30,000 elements. This is due to the merged-part model better representing the true geometry than the multi-part model, even though they use the same mesh seeding. The contiguous mesh resulting from merging the various parts into a single part optimizes the mesh alignment between the part interfaces, producing the best representation of the true model geometry, as seen in Figures 2a and 2b. Thus, it is best to merge parts that have curved interfaces with one another.

Even so, there are problems with the second-order elements in the multi-part problem. The second-order elements do not approach the correct eigenvalue with increasing element counts, and actually diverge from the correct value. This issue does not exist in the merged-part model, and indicates that there could be a tracking issue between curved surfaces of parts when using second-order elements. Furthermore, the runtimes for second-order hexahedrons shows erratic behavior for both the merged-part and the multi-part model. Every other element type shows increased runtimes corresponding to an increase in element count. While this erratic behavior in runtime does not necessarily indicate an issue with the second-order hexahedrons, it does merit further investigation.

Table 1 also shows that the merged-part model has faster runtimes than the multi-part model for similarly-meshed problems. This is due to the particle tracking routines in the unstructured mesh. When the particle is inside an element, it tracks on the mesh, calling an element-to-element tracking routine. When the particle hits the boundary of a part it has to call other routines in order to find out what part it is traveling into and what element of that part it is traveling into. The transition from one part to another takes longer than the element-to-element tracking routines, and as such accounts for the longer runtimes in the multi-part model.

One of the benefits of using an unstructured mesh model is the availability of advanced visualization of results. Figure 5 shows the one energy-bin volume-average flux results obtained from the elemental edit in

Table 1: Comparison of Big Ten CSG results to unstructured mesh results.

	Element Type	Mesh Seed (cm)	Number of Elements	Fuel Volume % Error	Reflector Volume % Error	$k_{\text{eff}}$	$\sigma k_{\text{eff}}$ (pcm)	$k_{\text{eff}}$ % Error	runtime (min)
Multi-Part	1 <sup>st</sup> -Order Hex	1	36912	0.024	0.225	0.99276	29	0.215	139.00
		3	4848	0.210	0.277	0.99327	24	0.164	82.79
		6	1944	0.837	0.456	0.99187	30	0.305	68.35
		18	984	9.969	2.330	0.97400	29	2.101	59.52
	1 <sup>st</sup> -Order Tet	1	139526	0.024	0.179	0.99327	28	0.164	178.07
		3	24233	0.210	0.237	0.99227	29	0.264	107.34
		6	10492	0.837	0.409	0.99243	30	0.248	84.67
		18	4993	9.969	2.284	0.97397	28	2.104	68.35
	2 <sup>nd</sup> -Order Hex	1	36912	0.000	-0.001	0.96533	29	2.972	1057.20
		3	4848	0.000	-0.001	0.96297	29	3.209	586.99
		6	1944	0.000	0.000	0.97364	31	2.137	549.78
		18	984	0.005	0.007	0.98499	29	0.996	698.45
2 <sup>nd</sup> -Order Tet	1	139526	0.000	-0.001	0.98038	27	1.459	916.91	
	3	24233	0.000	-0.001	0.99086	28	0.406	606.54	
	6	10492	0.000	0.000	0.99230	29	0.261	474.02	
	8	4993	0.078	0.007	0.98639	29	0.855	450.69	
Merged Part	1 <sup>st</sup> -Order Hex	1	47520	0.032	0.031	0.99499	30	-0.009	120.89
		3	5840	0.183	0.182	0.99504	29	-0.014	71.22
		6	1920	0.642	0.641	0.99404	28	0.086	61.69
		18	480	2.550	2.550	0.99101	32	0.391	55.95
	1 <sup>st</sup> -Order Tet	1	329718	0.034	0.033	0.99531	27	-0.041	243.32
		3	42395	0.160	0.160	0.99521	29	-0.031	83.98
		6	12038	0.642	0.641	0.99456	29	0.034	67.09
		18	3501	2.550	2.550	0.99078	33	0.414	57.50
	2 <sup>nd</sup> -Order Hex	1	47520	0.000	-0.001	0.99570	31	-0.080	837.71
		3	5840	0.000	-0.001	0.99507	30	-0.017	436.88
		6	1920	0.000	0.000	0.99463	29	0.027	392.89
		18	480	0.005	0.007	0.99433	29	0.057	424.63
2 <sup>nd</sup> -Order Tet	1	329718	0.000	0.000	0.99535	32	-0.045	978.93	
	3	42395	0.000	0.000	0.99505	29	-0.015	562.02	
	6	12038	0.000	0.000	0.99456	31	0.034	440.71	
	18	3501	0.078	0.005	0.99500	31	-0.010	387.98	
	CSG	-	-	-	-	0.99490	29	-	-



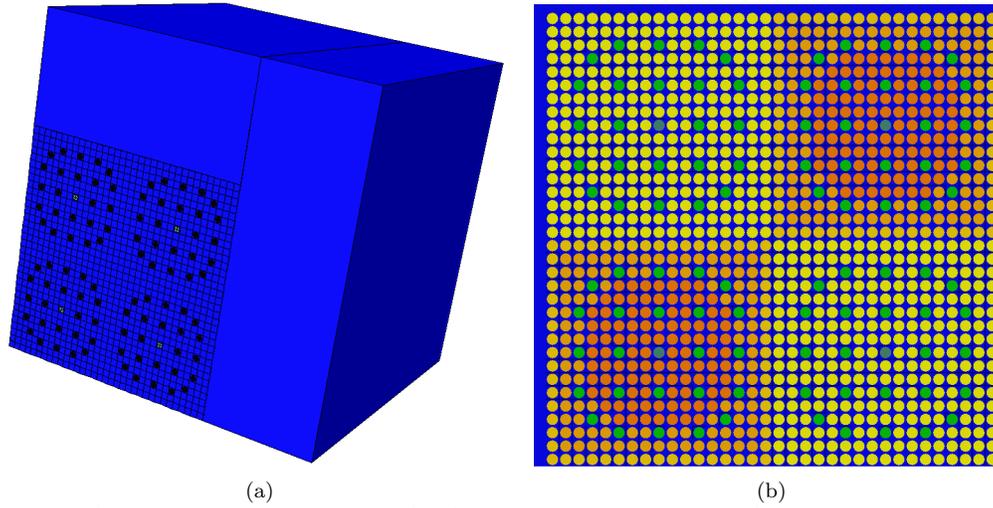


Figure 6: Abaqus/CAE model of C5G7 core (left). Planar view of Abaqus/CAE model of C5G7 assemblies (right).

Table 2: Number of elements per part and volume percent errors for the three C5G7 unrodded configuration unstructured mesh models.

Model	Part	Number of Elements	Volume % Error
Multi-Part Pincell	Water around Fuel/Guide	1540	0.17
	Fuel	1968	-0.36
	Water Around Guide/Control Rod - Above Core	380	0.70
	NormalGuide	1392	-0.51
	Central Guide	1972	-0.51
Merged-Pincell 1	Control Rod	820	-0.31
	Water around Fuel	1104	0.45
	Fuel	1968	-0.36
	Central/Normal Guide	1392	-0.34
Merged-Pincell 2	Water around Central/Normal Guide	2788	0.46
	Water around Fuel	1680	0.17
	Fuel	4272	-0.15
	Central/Normal Guide	6052	-0.15
	Water around Central/Normal Guide	2380	0.21

## Results

Kcode calculations were performed using 50,000 particles per cycle with 50 inactive cycles and 2500 active cycles. All simulations were conducted on Intel Xenon E5-2600 processors with 64 MPI processes with two threads each. A visualization of the results from the C5G7 merged-pincell 2 volume-averaged flux elemental edit is shown in Figure 7. Results of the eigenvalue calculations for the three unstructured mesh models and the CSG model are presented in Table 3. All reference CSG values were obtained from previously calculated benchmarks [4]. Performance parameters for the various simulations are presented in Table 4.

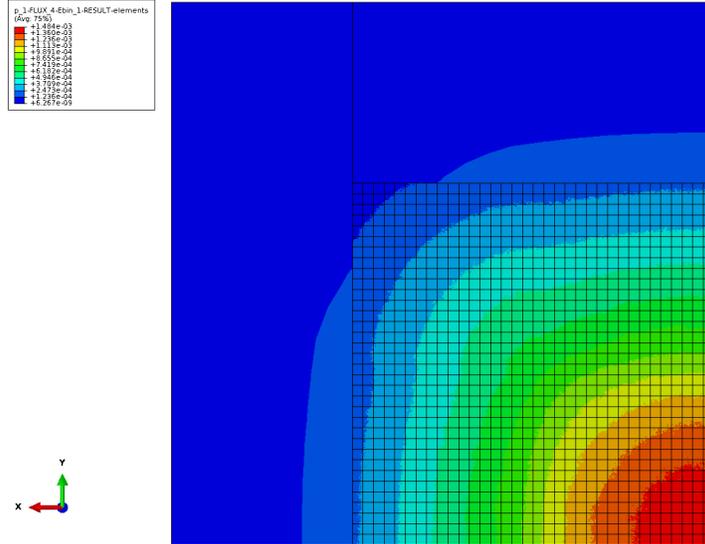


Figure 7: Contoured one energy-bin volume-averaged flux results for the bottom (left) and side (right) of the C5G7 core.

Table 3: Comparison of eigenvalues between unstructured mesh models and reported CSG value with their corresponding statistical uncertainties.

Model	Elements per Pincell	Fuel Volume % Error	Total Number of Elements	$k_{\text{eff}}$	$\sigma k_{\text{eff}}$ (pcm)	$k_{\text{eff}}$ % Change
Multi-Part Pincell	3594	-0.357	4.12E+06	1.13033	18	1.107
Merged-Pincell 1	3072	-0.357	3.69E+06	1.14432	8	0.117
Merged-Pincell 2	5952	-0.153	7.14E+06	1.14350	8	0.037
CSG	-	-	-	1.14308	3	0.000

The results in Table 3 show that none of the unstructured mesh models agree within  $2\sigma$  of the CSG model results. As expected from the results from the Big Ten unstructured mesh simulations, the multi-part pincell model produces inaccurate results with k-effective being approximately 1,000 pcm lower than the CSG model. When the pincell is merged into a single part, the results improve drastically with the merged-pincell 2 model producing results that disagree by only 42 pcm. With a finer mesh, this difference could be reduced.

The performance results in Table 4 provide some encouraging results. For one, merging the pincell into one part has a significant runtime reduction which agrees with the results from the Big Ten simulations. This could be expanded to merging multiple pincells into one part to further improve performance. Furthermore, runtime does not significantly increase with the number of elements; a doubling of the number of elements in the model only produces a modest increase in runtimes. As such, the runtimes required to increase the number of elements to reduce the discrepancy in k-effective should not be prohibitive. However, unstructured

Table 4: Computer statistics for the C5G7 problem.

Model	Number of Instances	Number of Elements	Runtime (minutes per cycle)	Memory on a Single Core (GB)
Multi-Part	3594	4.12E+06	3.6	3.1
Merged-Pincell 1	2214	3.69E+06	1.06	2.8
Merged-Pincell 2	2214	7.14E+06	1.18	5.2
CSG	-	-	-	0.03

mesh representations of assemblies require hundreds of megabytes to gigabytes of memory, and increasing the number of elements corresponds to a roughly linear increase in memory requirements. Thus, the memory requirements pose a limitation on the number of elements capable of being modeled in the unstructured mesh.

Pin powers were obtained using an F4 (volume-averaged flux) mesh tally in MCNP with an FM (tally multiplier) card to tally the absorption reaction rate in each axial pincell region. The reaction rates were then normalized to obtain pin powers where the total power is equal to the number of fuel pins using analysis tools provided in the reference benchmark [4]. Comparisons of maximum pin powers and assembly powers for the the various axial sections of the model are shown in Tables 5 and 6, respectively.

Table 5: Maximum pin power percent errors for various axial sections of the core.

Section of Pin	Reference MCNP		Unstructured Mesh	
	Max. Pin Power	Uncertainty	Max. Pin Power	Percent Error
Whole Pin	2.481	0.14	2.480	0.12
Slice 1	1.108	0.21	1.108	0.02
Slice 2	0.882	0.23	0.882	-0.02
Slice 3	0.491	0.30	0.490	-0.27

Table 6: Assembly power percent error for the various axial sections of the core.

Section	Code	Inner UO2	Percent Error	MOX	Percent Error	Outer UO2	Percent Error
Whole Assembly	Ref.	491.2 $\pm$ 0.29%	-	212.7 $\pm$ 0.21%	-	139.4 $\pm$ 0.15%	-
	UM	491.2 $\pm$ 0.01%	0.00	212.7 $\pm$ 0.03%	0.02	139.3 $\pm$ 0.02%	-0.08
Slice 1	Ref.	219.0 $\pm$ 0.19%	-	94.5 $\pm$ 0.14%	-	62.1 $\pm$ 0.10%	-
	UM	219.0 $\pm$ 0.02%	-0.01	94.5 $\pm$ 0.04%	-0.05	62.1 $\pm$ 0.3%	0.04
Slice 2	Ref.	174.2 $\pm$ 0.17%	-	75.2 $\pm$ 0.13%	-	49.5 $\pm$ 0.09%	-
	UM	174.2 $\pm$ 0.02%	0.02	75.3 $\pm$ 0.05%	0.10	49.3 $\pm$ 0.03%	-0.32
Slice 3	Ref.	97.9 $\pm$ 0.13%	-	42.9 $\pm$ 0.10%	-	27.8 $\pm$ 0.07%	-
	UM	98.0 $\pm$ 0.03%	0.10	43.0 $\pm$ 0.06%	0.28	27.8 $\pm$ 0.05%	0.08

As seen in Table 5, the pin powers for all of the axial sections are well with 1% of the reference values. None of the errors in the maximum pin powers fall outside of the reported statistical error. Furthermore, Table 6 shows good agreement (well with 1%) between the assembly powers for all axial sections; only two assembly powers have errors larger than the statistical error reported by the reference model. Even so, the errors are not significant due to their magnitude and the fact they occur in the lower-power outer UO2

assemblies.

In order to compare the results obtained from the tracking on the unstructured mesh, termed an edit, a second calculation was performed without the FM card in MCNP. These results were normalized using the same analysis tools, so the total flux is equal to the number of pins. The comparison between the edit and the MCNP tally is shown in Table 7. Results show that there is no difference between the assembly powers obtained from the unstructured mesh edit and the F4 mesh tally. Further investigation concludes that all of the pin powers match to within 4 digits of precision; there is no significant difference between the two calculations of the volume-average flux in each pincell.

Table 7: Assembly power percent errors between the F4 mesh tally and the unstructured mesh edit.

Tally/Edit	Inner UO2	Percent Error	MOX	Percent Error	Outer UO2	Percent Error
F4 Mesh Tally	490.5 ± 0.01%	-	216.2 ± 0.03%	-	133.1 ± 0.02%	-
UM Edit	490.5	0.00	216.2	0.00	133.1	0.00

#### 4 Thermal-Hydraulics Coupling

With the unstructured mesh capabilities in MCNP6 verified for reactor physics problems, the next step is to couple MCNP to a thermal-hydraulics code in order to capture the effects of temperature feedback. Abaqus/CAE was chosen as the thermal-hydraulics code because the unstructured mesh capabilities in MCNP are most compatible with Abaqus/CAE. The procedure to couple MCNP to a thermal-hydraulics code like Abaqus is as follows:

- Obtain a power profile from MCNP.
- Normalize the power profile to a user-specified pin power.
- Insert power profile into Abaqus/CAE and run thermal hydraulics analysis.
- Read resulting element-based temperatures and densities into MCNP.
- Repeat until  $k_{\text{eff}} < \epsilon$

The test problem presented here is a simple PWR pincell arranged in an infinite lattice. The pincell consists of a 3.3% enriched UO<sub>2</sub> fuel pin with zirconium cladding surrounded by light water with the coolant inlet being at the bottom of the pincell. The test problem was adapted from a single pin BWR benchmark problem for an existing coupled Monte Carlo - thermal hydraulics analysis [5]. The main parameters of the problem are listed in Table 8. Temperature dependent thermal conductivities and isobaric specific heats were used for all materials except for the helium gap, however temperature dependent densities were only used for water [6].

The pincell model was meshed with first-order hexahedrons using a global seed of 0.1 cm, an axial seed of 4 cm, and an edge seed of 0.2 cm. The current method of importing the element-wise power density into Abaqus using Abaqus' \*DFLUX routine requires that the same mesh be used for neutronics and thermal hydraulics calculations. As such, the fuel, gap, clad, and water are all explicitly modeled in the MCNP calculation. This requires a finer mesh to be used in order to accurately capture the volume representation of the gap. A future capability allowing the power densities obtained from MCNP to be translated to a different thermal-hydraulics mesh would eliminate this necessity. A radial slice of the pincell model at  $z = 2$  m is shown in Figure 8.

Power densities were obtained by first obtaining the fission energy deposition in each element using an F7 elemental edit during a Kcode calculation, and then normalizing the distribution to a pre-defined total pin power. This capability was added to the unstructured mesh post processing tools, *um\_post\_op*. The first iteration of the power density was obtained using a Kcode calculation with 20,000 histories per cycle with 50 inactive cycles followed by 200 active cycles. The power density profile along the axial centerline obtained from MCNP, without normalization, is shown in Figure 9a.

Table 8: Geometry and boundary conditions of the PWR pincell problem.

Fuel Pellet radius	0.5225 cm
Fuel Pin Radius	0.6125 cm
Fuel Cell Size	1.823 cm
Height of Pin	3.8 m
Bottom and Top Reflectors	0.2 m
Gap Thickness	85 $\mu\text{m}$
Inlet Flow Rate	1120 kg/m <sup>2</sup> s
Inlet Temperature	280 °C
Gap Conductance	10 kW/m <sup>2</sup> K
Pin Total Power	70 kW
Pressure	15 MPa
Fuel Density	10.8 g/cm <sup>3</sup>
Cladding Density	6.55 g/cm <sup>3</sup>

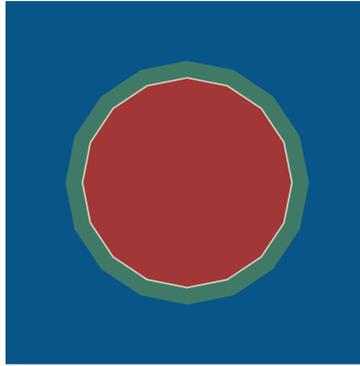


Figure 8: Radial slice at  $z = 2$  m of PWR pincell model with the gap explicitly modeled.

The power density profile is as expected, with most of the fissions occurring near the surface of the fuel pin, with the peak power density occurring at the center of the pin. After obtaining the element-wise power densities, thermal-hydraulic calculations were conducted. In order to simulate an infinite lattice of fuel pins, periodic boundary conditions were assigned to each side of the fuel pin. This was done using Abaqus's \*TIE routine and linking opposite sides of the fuel pin together. Forced convection was modeled using the \*MASS FLOW RATE routine in Abaqus, specifying a mass flow rate of 1200 kg/m<sup>2</sup>-s in the positive axial direction at the nodes of every water element and no mass flow rate in the radial or azimuthal directions. The resulting axial temperature profiles at the pin centerline, clad-gap interface, and clad-water interface are shown in Figures 9b-9d. The radial temperature profile near the fuel peak temperature is shown in Figure 10b.

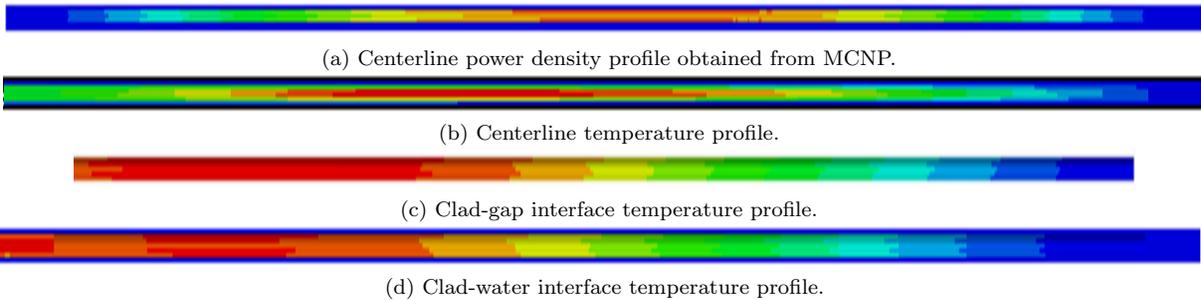


Figure 9: Axial power density profile and temperature profiles at the pincell-centerline and material interfaces. The coolant inlet channel is located at the right of this figure.

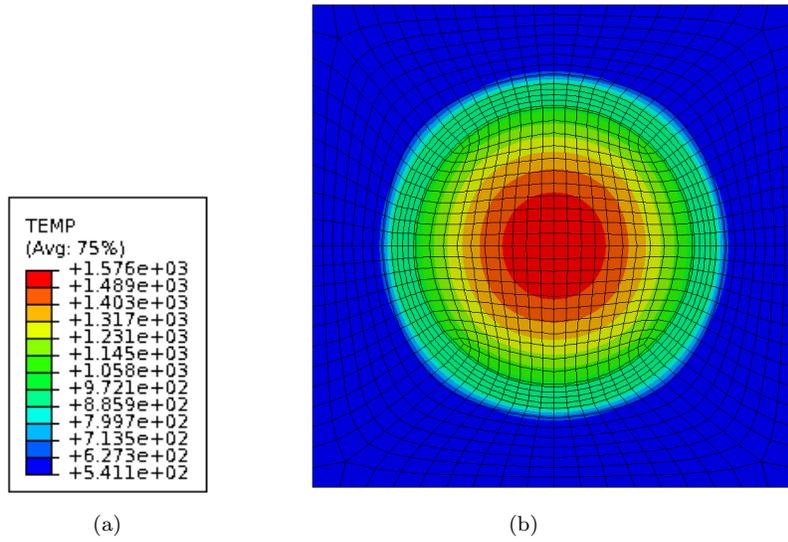


Figure 10: Radial temperature profile at  $z = 2.6$  m.

All of the axial temperature profiles exhibit features one would normally expect for a pincell thermal-hydraulics calculation. The fuel peak temperature is located above the center plane of the fuel pin, the clad temperature peaks near the top of the fuel pin, and the water temperature increases monotonically with the direction of coolant flow. Furthermore, at first glance the radial temperature profile also appears as expected. The temperature peaks at the fuel pin centerline at 1576 K and decreases to 541 K at the edges of the pincell. However, further investigation of the temperatures in the clad and near the clad-water interface reveals inaccuracies in the model (discussed below). Figures 11b and 12b show the node-based temperatures of the clad and water, respectively, 0.1 m below the top of the fuel pin.

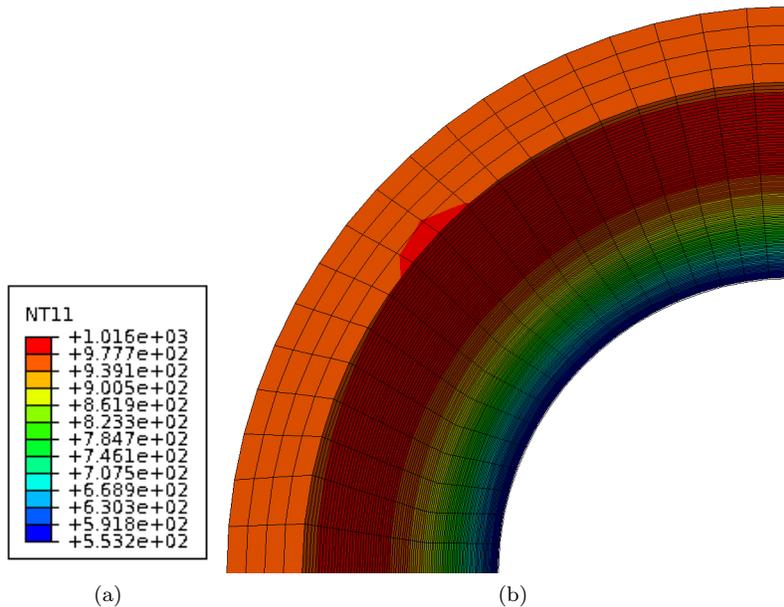


Figure 11: Clad node-based temperature at  $z = 3.9$  m (0.1m below top of fuel pin).

Figure 11b reveals that the temperature of the clad is too high at the top of the model, in excess of 1000 K. Further investigation of the cladding temperature profile reveals asymmetries. These asymmetries are likely due to the mesh becoming finer in the fuel pin adjacent to the region of local temperature increase

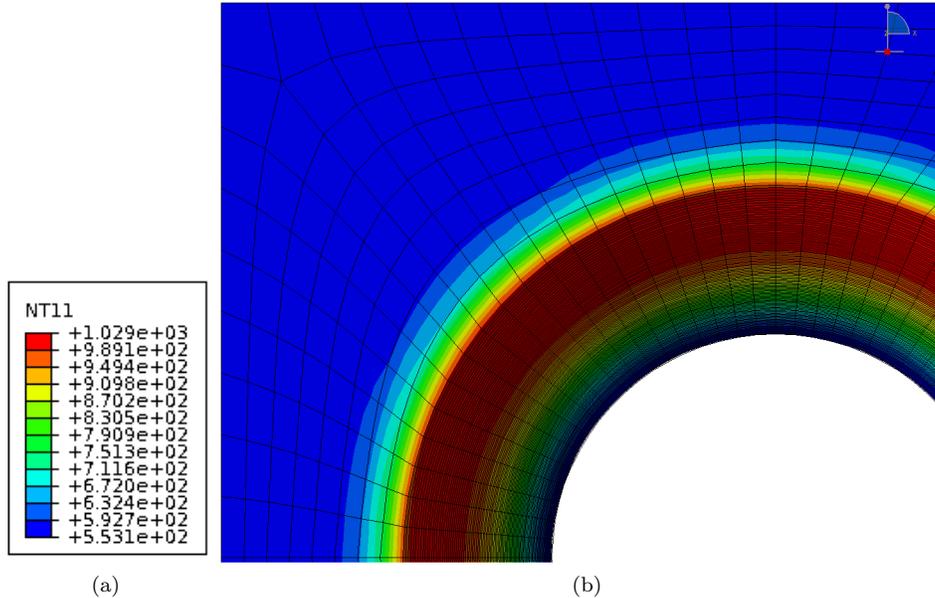


Figure 12: Water node-based temperature at  $z = 3.9$  m (0.1 m below top of fuel pin).

in the cladding. This mesh is shown in Figure 10b. Figure 12b shows a similar result, with the water in the vicinity of the clad reaching temperatures in excess of 1000 K. Furthermore, the temperature profile in the water is incorrect; the bulk temperature of the water should be higher than the inlet temperature of 553 K. This indicates that Abaqus is modeling the forced convection of the water as laminar flow, not turbulent flow. There is no mixing of the water in the coolant channel; the change in the radial temperature is caused by heat conduction only. In order to fix this issue turbulent flow must be modeled. As such, the use of turbulent flow modeling capabilities in Abaqus/CAE or Abaqus/CFD will be looked into for future implementation.

## 5 Conclusions

### *Verification*

The ability of unstructured mesh routines in MCNP to accurately model reactor physics problems has been discussed. Models constructed with an unstructured mesh were able to obtain accurate results as long as a suitable mesh was chosen for the application. While issues exist with interfaces between parts with curved surfaces, merging the parts into a single part alleviates this issue and produces accurate results. Currently, using second-order elements increases runtimes significantly, however using second order elements could reduce memory burdens if the number of first-order elements required to accurately represent a given volume is prohibitive. Additionally, the computing costs associated with using an unstructured mesh representation of a reflected four-assembly core using first-order hexahedrons are not prohibitive, and increasing the number of elements does not lead to significantly increased runtimes.

### *Thermal-Hydraulics Coupling*

Current work on coupling the unstructured mesh capabilities in MCNP6 to the thermal-hydraulic solvers of Abaqus/CAE has shown that it is possible to obtain element-based power density and temperature information from MCNP and Abaqus, respectively. However, higher fidelity forced convection modeling is required in Abaqus in order to obtain accurate temperature profiles.

## 6 Future Work

The unstructured mesh library for neutronics problems in MCNP has been verified for reactor physics problems, however the best approach for implementing an unstructured mesh is still being explored. Merging several pincells into a single part could lead to reduced runtimes, however this could cause issues if the number of elements per part grows too large. In the future, turbulent flow models will be implemented into Abaqus's thermal hydraulic solvers in order to obtain accurate temperature profiles. Meanwhile, element-based temperature and density input for MCNP6 will be developed in order to facilitate multi-physics coupling between MCNP6 and Abaqus.

## References

- [1] Karen C. Kelley, Roger L. Martz, David L. Crane, "Riding Bare-Back on Unstructured Mesh for 21st Century Criticality Calculations (U)," Los Alamos National Laboratory, LA-UR-09-07320 (2010).
- [2] Desselault Systemes Simulia, Inc., "ABAQUS USER MANUALS, Version 6.11," Providence, RI (2011).
- [3] *International Handbook of Evaluated Critical Safety Benchmark Experiments*, Nuclear Energy Agency, NEA/NSC/DOC(95) 03 Vol III, September 2010 Edition
- [4] *Benchmark on Deterministic Transport Calculations Without Spatial Homogenization, MOX Fuel Assembly 3-D Extension Case*, Nuclear Energy Agency, NEA/NSC/DOC(2005)16
- [5] A. Ivanov, V. Sanches, J.E. Hoogenboom, "Single Pin BWR Benchmark Problem for Coupled Monte Carlo - Thermal Hydraulics Analysis", *PHYSOR 2012*, Knoxville, Tennessee, April 15-20, 2012, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2012).
- [6] *Thermophysical properties database of materials for light water reactors and heavy water reactors*, International Atomic Energy Agency, IAEA-TECDOC-1496, Vienna, Austria, June 2006