Uncertainty Quantification in Two-dimensional Simulations of Spent Nuclear Fuel Assemblies

Dr. Imane Khalil, Quinn Pratt, Harry Schmachtenberger
University of San Diego

Dr. Roger Ghanem
University of Southern California
Outline

• Nuclear Energy Background
• 9x9 Fuel Assembly FLUENT Model
• Uncertainty Quantification (UQ) Introduction and Background
• Applying UQ to FLUENT Model
• Results
• Future Work
Nuclear Energy

Spent Fuel Pool to Shield Radiation and Cool Fuel Rods
Spent Fuel Storage Casks

- **Cask Lid**: Diameter: 98 in.
- **Steel Canister**: 0.5 in.
- **Concrete Wall**: 26.75 in.

**Dimensions**: Overall Length: 197 to 225 in.
Loaded Weight: 360,000 lbs.
Typical Payload: 24 PWR Bundles

*Storage and Transportation*
Fluent Geometry

- **Stainless Steel Wall**
- **Helium Gas Filled Regions**
- **Uranium Dioxide Fuel Rod (Ø =1 cm)**
- **Zircaloy**

47,944 nodes
47,702 elements
Fluent Model

- Steady State 2D simulations using Pressure-based solver in Fluent
- Gravity in Y direction
- Energy Equation ON to model:
  - Conduction heat transfer in solid region
  - Buoyancy induced gas motion and natural convection heat transfer within the gas filled regions
  - Radiation heat transfer across all gas filled regions
## Fluent Model Input and B.C.

<table>
<thead>
<tr>
<th>Material</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zircaloy</td>
<td>Thermal Conductivity, Specific Heat, Emissivity</td>
</tr>
<tr>
<td>Helium</td>
<td>Thermal Conductivity, Specific Heat</td>
</tr>
<tr>
<td>Uranium Dioxide</td>
<td>Thermal Conductivity, Specific Heat, Emissivity, Power</td>
</tr>
<tr>
<td>Stainless Steel</td>
<td>Wall Temperature</td>
</tr>
</tbody>
</table>
Uncertainty Quantification

Mathematical models of heat transfer are often challenged by random/uncertain properties. Uncertainty quantification is needed in order to get a predictive fidelity of the simulation.
Introduction to PCE

PCE is Polynomial Chaos Expansion

**Univariate Case:** For one input parameter, we model the propagation of uncertainty through our model

1. Assume our input variable can be expressed as a polynomial expansion. Where $\xi$ represents a uniform random variable, and $\psi$ represents Legendre Polynomials
2. After propagating $X$ through our model we can model the output of the model as a similar expansion
3. We can then perform non-intrusive spectral projection (NISP) to extract the coefficients of this expansion

\[
X = \sum_{i=0}^{p} x_i \psi_i(\xi)
\]

\[
T = \sum_{i=0}^{p} T_i \psi_i(\xi)
\]

\[
K = \frac{\langle x_i, \psi_i \rangle}{\langle \psi_i, \psi_i \rangle} \quad T_i = K \ast T
\]

$p = N$-ord (Highest order Legendre Present).

$n = N$-dim (Number of uncertain dimensions).
**Introduction to PCE**

**Multivariate Case:** For $n$-input parameters, we model the propagation of uncertainty through our model.

1. When modeling the propagation of multiple uncertain parameters we introduce a Multi-index ($\mathbf{M}$) showing the ordering of polynomials in the expansion. The number of terms in our expansion is $N_{pc}$.

2. Using identical methods from the univariate case we are able to extract an expansion for the output parameter as a function of our multiple input variables.

\[
p = N_{ord} \quad \text{(Highest order Legendre Present)}.
\]

\[
n = N_{dim} \quad \text{(Number of uncertain dimensions)}.
\]

\[
N_{pc} = \frac{(n + p)!}{n!p!}
\]

\[
T = \sum_{i=0}^{N_{pc}} T_i \Psi_i(\Xi)
\]

\[
= \sum_{i=0}^{N_{pc}} \prod_{j=1}^{n} \psi_i^j(\xi^j)
\]

where \( l = M_i^j \)
We will now consider the case where Nord = 2, and Ndim = 2.

In this case we will use the nPCETerms equation to see that we expect 6 terms in our output expansion.

We will then look at our MultiIndex matrix which shows how the terms are paired.

This yields the following general expression for the center temperature expansion:

\[
T_c = T_0 + T_1 x_1 + T_2 x_2 + T_3 \left(\frac{1}{2}\right)(3(x_1)^2 - 1) + T_4 x_1 x_2 + T_5 \left(\frac{1}{2}\right)(3(x_2)^2 - 1)
\]

Now we refer to the table of Legendre polynomials to see what these basis Functions actually are.
Matlab-Fluent Interface

- Using user-input, MATLAB can create arbitrarily complex journal files based on a custom dictionary of FLUENT text user-interface commands.
- These journal files are sent to the ANSYS - FLUENT environment where they are used to modify model parameters based on their uncertainty.
- Solution data is exported after each simulation, these files are then collected by MATLAB and used for UQ data analysis.
Sensitivity Analysis

- 10 Dimensions & 1st Order Legendre Polynomials
- This allows us to see the relative contribution of each variable to the overall output parameter, be that Temperature, Velocity, Pressure, etc.
Sensitivity Analysis Results

- 1,024 FLUENT Simulations
- Time-to-Run on local machine: Intel Xeon E3 - 4 (8 thread) 3.50GHz cores Approximately 16 hours.
- Coefficients of Significance for Center Temperature

<table>
<thead>
<tr>
<th>(Mean)</th>
<th>HeK</th>
<th>HeCp</th>
<th>ZrK</th>
<th>ZrCp</th>
<th>ZrEm</th>
<th>UOK</th>
<th>UOCp</th>
<th>WallT</th>
<th>Power</th>
<th>FuelEm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>379.6246</td>
<td>-0.11803</td>
<td>9.08E-03</td>
<td>-0.04534</td>
<td>-1.44E-15</td>
<td>-0.09561</td>
<td>-5.42435</td>
<td>-5.30E-14</td>
<td>48.18958</td>
<td>1.596366</td>
</tr>
</tbody>
</table>

- ZrCp and UOCp can be dropped (no convection heat transfer in solid)
- HeCp can also be dropped. This could be explained by the dominant conduction/radiation heat transfer modes in gas filled regions (ref. Araya and Greiner)
PDF of the Center Temperature Distribution
Mean Temperature Distribution
Variance and Coefficient of Variance

COV = mean / standard deviation
Average Velocity Distribution

Max velocity = 2 mm/sec
2\textsuperscript{nd} Order PCE Analysis Results

- Reduced Dimensions to 7 after Cp concluded to be insignificant, based on sensitivity analysis.
- 2,187 FLUENT Runs
- Time-to-Run on local machine: less than 2 Days
PDF of the Center Temperature Distribution
Mean Temperature Distribution
Future Work

- Convergence Analysis with 3rd order polynomial
- HPC - run larger, more complex models faster
- Building 3D models of the assemblies