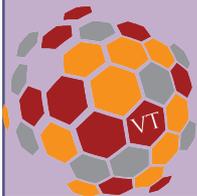


# Demonstration: Real-time Analysis for spent fuel Pool *In-situ* Detection (RAPID)



**VIRGINIA TECH TRANSPORT THEORY GROUP (VT<sup>3</sup>G)  
NUCLEAR SCIENCE AND ENGINEERING LABORATORY  
VIRGINIA TECH**



**NSEL**

Nuclear Science and Engineering Laboratory  
Virginia Tech Research Center, Arlington, VA



**VirginiaTech**  
*Invent the Future<sup>®</sup>*

*Joint International Conference on Mathematics and Computation, Supercomputing in Nuclear Applications and the Monte Carlo Method  
Nashville, TN, April 19-23, 2015*



## RAPID Structure

### Pre-Calculation (one time):

1. Burnup Calculation – to obtain material composition
2. Fission Matrix Coefficient Generation

### Real-time Analysis:

1. Run Fission Matrix Code
2. Process Results

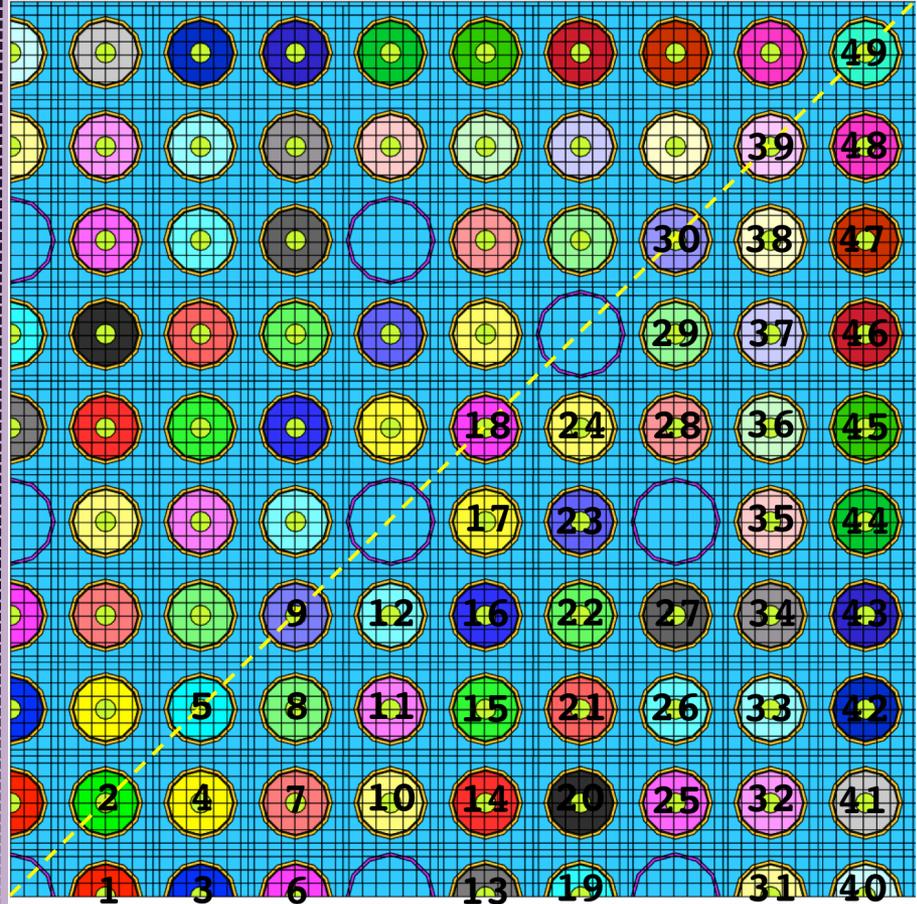
# Pre-calculation



- 1. SCALE BURNUP CALCULATION**
- 2. SCALE OUTPUT PROCESSING**
- 3. MCNP INPUT GENERATION & CALCULATION**
- 4. MCNP TALLY PROCESSING**

# Pre-calculation – Step 1

- At each desired burnup, run a quarter assembly SCALE (t-dep1 module) model
  - Reflected on  $-x$  and  $-y$
  - Octal symmetry
  - 49 fuel materials (each pin within octant is unique)



# Pre-calculation – Step 2



- **SCALE outputs:**
  - neutron/gamma spectra
  - actinide/fission product concentrations
  - Fission Spectrum ( $\chi$ )
    - ✦ Fit to Watt fission spectrum (nonlinear regression)
- **Process SCALE outputs**
  - `getdat.sh` - prepares material and source information for MCNP input file
  - `fitChi.R` - prepare continuous energy fission spectrum (Watt's spectrum format) from multigroup SCALE generated spectrum for MCNP input file

# Pre-calculation –Step 3



- Automatic input file generation for MCNP
  - `calMat.f90` – generates necessary input block segments as a function of burnup and cooling time (source definition and material composition)
  - `makeMCNP.sh` – concocts input block segments to generate a full MCNP input file
  - `mkzmcnp.sh` – generate 55 unique input files for each  $a$  &  $b$  calculation
- Run MCNP for each coefficient as a function of burnup and cooling times

# Pre-calculation – Step 4



- Processing MCNP output files to generate database
  - `getFMco.sh` - extract fission density tally from each MCNP output file
  - `rdmc.f90` – generates FM coefficient database file

# Pre-processing– Estimated Time Requirements



- For a single coefficient calculation and processing.

| Step  |                  | Time (serial)   |
|---|------------------|-----------------|
| 1.1) SCALE Run                              |                  | ~158 min        |
| 1.2) SCALE Output Processing                |                  | n/a             |
| 3) MCNP<br>Fixed-<br>source<br>Calculation: | Input Generation | n/a             |
|   | Calculation      | ~28 min         |
| 4) Tally Processing/Consolidation           |                  | ~1 min          |
| <b>Total</b>                                |                  | <b>~187 min</b> |

# Real-time Analysis



- 1. INPUTS FOR POOL SETUP**
- 2. EXAMPLES**

# Input Files

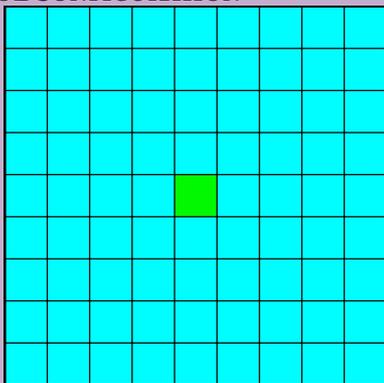


- `pool.inp` – defines the pool structure and range of burnups and cooling times (driver file)
- `runName.burn` – defines assembly-wise axial burnup distribution
- `runName.cool` – defines assembly-wise cooling times

# Examples

## Case 1

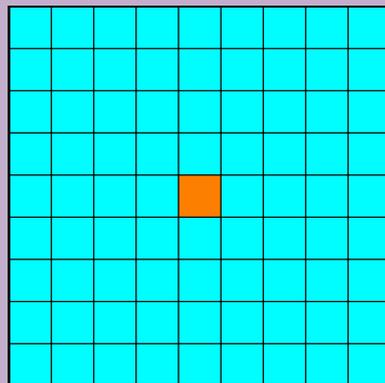
POOL CONFIGURATION



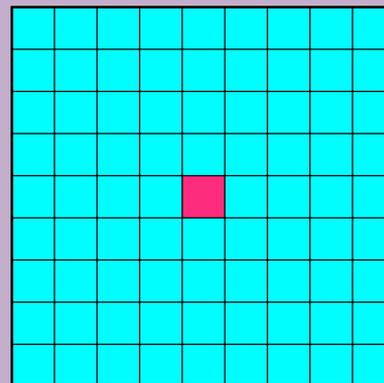
BURNUP DISTRIBUTION  
[GW<sub>D</sub>/MTHM]



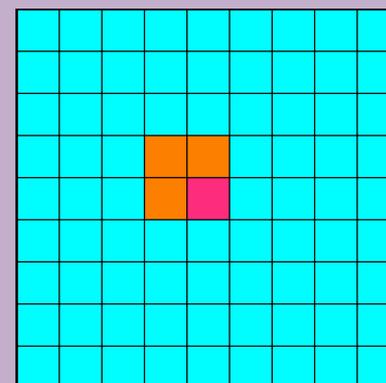
## Case 2



## Case 3



## Case 4



# Post Processing



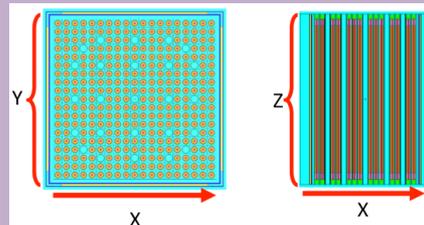
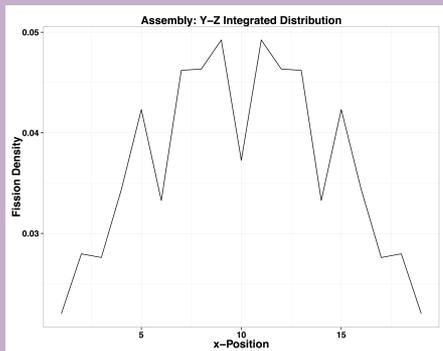
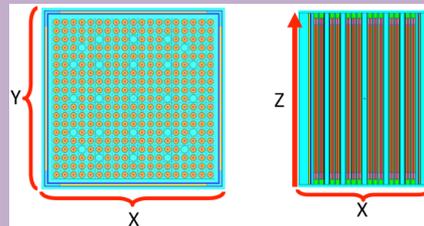
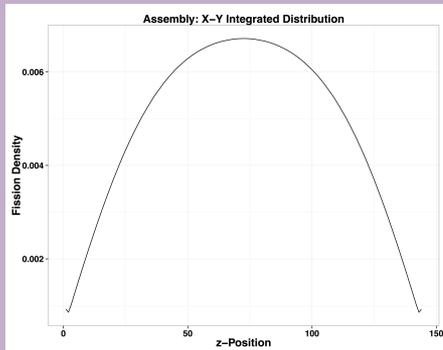
- 1. FISSION DENSITY – 1D PROFILES**
- 2. FISSION DENSITY – 3D DISTRIBUTIONS**
- 3. FISSION DENSITY – 2D HEAT MAPS**

# Post Processing: 1x1 Pool Layout

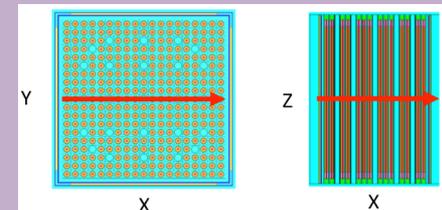
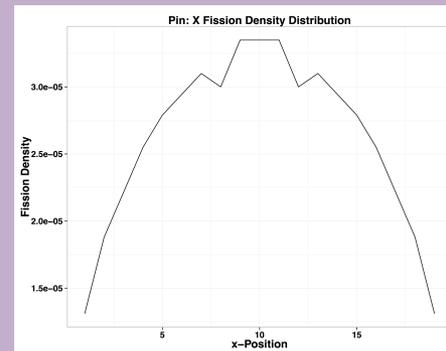
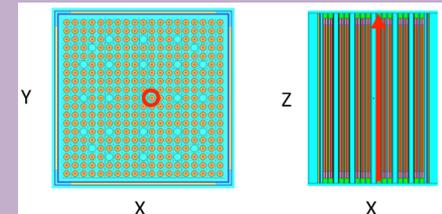
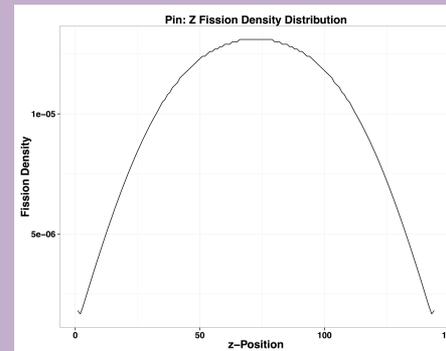


- 1D Fission Density Profiles

## ASSEMBLY INTEGRATED



## PIN-WISE

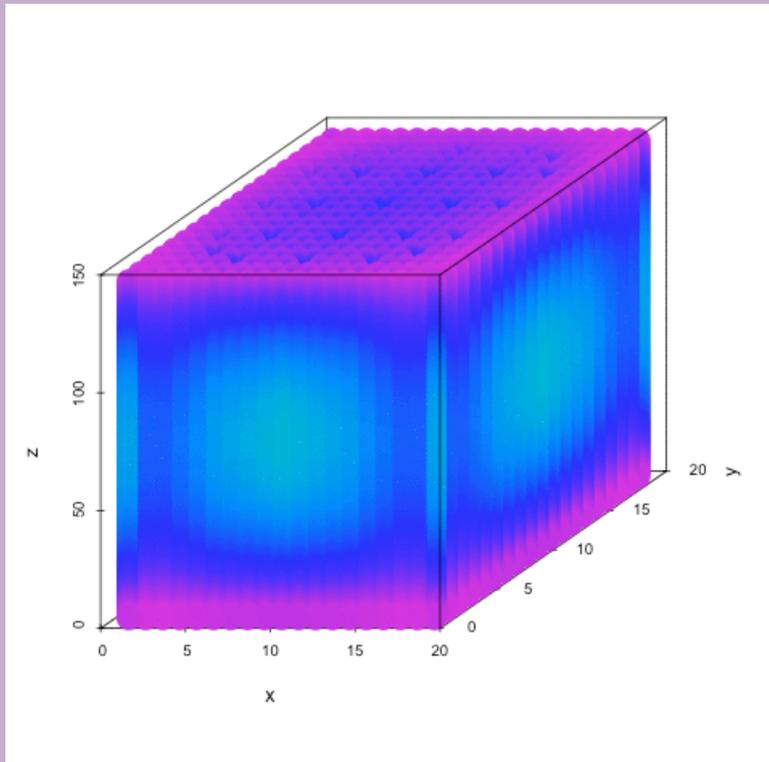


# Post Processing: 1x1 Pool Layout

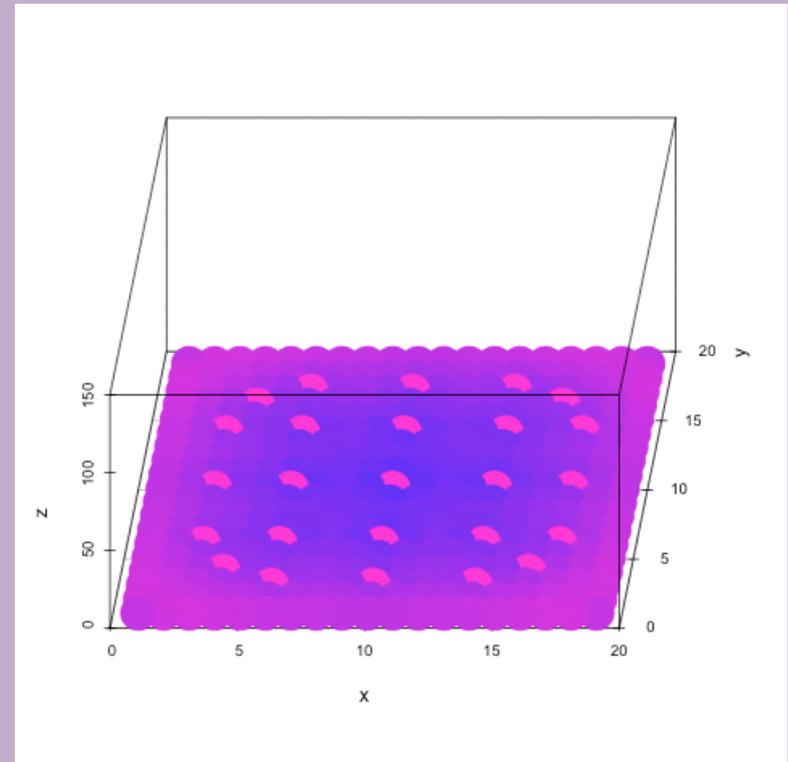


- 3-D Fission Density

Y-LEVEL ANIMATION



Z-LEVEL ANIMATION



# Post Processing: 2x2 Pool Layout



- 2-D Fission Density Heat Map

