NEA and MIT Systems Code Benchmarks

Brent Dixon

Extract of slides given at
Systems Analysis Working Group Meeting
July 8, 2009
Background

- Systems codes are complex and difficult to verify
- Benchmarking provides a means for code validation
- VISION has now been benchmarked in two separate studies
  - NEA benchmark
    - 3 scenarios in a progressive series
    - 5 codes (COSI, FAMILY, DESAE, EVOLCODE, VISION)
  - MIT benchmark
    - 5 scenarios varying in growth rate and fuel cycle
    - 4 codes (CAFCA, COSI, DANESS, VISION)
NEA Benchmark Series

- Three benchmarks based on a constant level of nuclear energy
  - Open cycle
  - Monorecycling of the Plutonium in the PWRs.
  - Monorecycling of the Plutonium in the PWRs and then deployment of the Gen IV fast reactors recycling Plutonium and minor actinides.

- Benchmark specification includes numerous parameters defining the scenarios
  - Reactor properties
  - Core properties
  - Fuel properties and isotopic contents
  - Reprocessing schedules, capacities, priorities, efficiencies
  - Electricity output by reactor type by year
NEA Benchmark Series

- Specified outputs in a spreadsheet format, to include:
  - Natural Uranium consumption,
  - SWU needs,
  - Fuel fabrication flows
  - Interim storage inventories
    - spent fuel
    - depleted Uranium
    - Plutonium
    - Etc.
  - Processed spent fuel
  - Pu and MA mass flows
  - Plutonium and minor actinides losses from reprocessing
NEA Benchmark #1 – Open Cycle

- A constant energy level with a single reactor type
  - Confirms initial conditions modeled consistently
  - Confirms fuel cycle front-end flows
  - Simple case easily verified

Scenario 1

![Graph showing Scenario 1 with PWR UOX power over time.](image-url)
NEA Benchmark #2 – Adds MOX

- Designed for equilibrium behavior
  - Confirms separations initialization
  - Confirms fleet fuel mix transition
    - Rate of introduction
    - Level sustained
  - Storage inventory decay impacts results
NEA Benchmark #3 – Adds FR Transition

- **Much more complex**
  - Adds two more transitions
    - *Ending MOX*
    - *Starting FRs (convertors)*
  - Augments separations strategy
    - *UOX, MOX, FR core, FR blanket all specified separately*
  - Adds reactor retirement

- **Tests TRU mass management**
  - TRU for FR startup schedule barely sufficient
Discussion

- Benchmarking is hard to do
  - Even a simple case requires specifying pages of input
  - Differences in interpretation require iteration of the specification

- In general, all the codes demonstrated similar behavior
  - Especially true for general trends, which is purpose of these codes
  - Specific differences usually traceable back to how each code modeled features (more stages/details gives more time step delays, etc.)

- Benchmarks generally did not test advanced features of codes
  - Many intelligent capabilities were overridden (code dumbed down) to get best match with other cases
  - Many advanced extensions appear only in a single code