Bo Feng – Valuable lessons from fuel cycle code comparisons

- Suggestions for benchmark organization
  - Avoid ambiguous language
  - Avoid borderline scenarios
  - Specify timing sequence
    - This also means to give sufficiently small timestep
  - Specify data reporting
    - How data is reported.
  - Use output data template
  - Set schedule - have a hard time limit.
    - Have set calls
    - Fixed deadlines
    - 'Agree when close enough and move on'
- 'For more complex scenarios, all code may need to make approximations at one point or another'

Brent Dixon – Simulation tool benchmarking and verification

- What to test?
  - Test basic features and essential functionalities
  - Integrated features
    - Does this work?
    - Full cycle tests
  - Exemplary features
    - May not be present in all codes
    - Usually specific to a simulator - like economics
- General Approaches
  - Manual verification
  - Spreadsheets
  - Benchmarking with a set of codes
  - Auto-verification of code functions/methods
- Scenario-Specific approach
- Parts test
  - Enrichment etc.
- Full Code test
  - One reactor steady state
  - Startup / shutdown
  - Homogenous fleet - growth / no growth
- Code to code testing
  - Goal - produce exactly the same result
  - Differences result (hopefully) in code improvements
- MIT benchmark
  - Good matching - showed differences in the simulators for one test.
- NEA/OECD Benchmark
- IAEA Program
  - Included codes that did not benchmark in the previous
  - Newer codes - didn't have the same set of skills as the older codes.
- Take Aways
  - Less involved scenarios - easiest to understand what's going on.
  - Use intersection of code capabilities - not the union
  - Expect results to be close but not exact

    - Allow time for iterations
    - Strong similarities not exact results.

Paul Wilson – Recipe vs model

- Using EG30
- Recipes provided by Argonne national lab. Models in Cyclus
- 3 Calculations
    - One with recipes
    - One using models without decay
    - One using neural networking models with decay
- Thermal power
    - Discrepancies before 20 years into the simulation but after that there is good agreement between the cases
    - This makes sense.
- The transition from blanket to no blanket can happen much faster with the model than with a recipe reactor.
- Conclusion - recipe might be best in some cases, but models might be better in others.

Steve Skutnik – Integrating physics-based depletion into fuel cycle simulation: When, how, and why

- Thermal - spectrum one group cross sections are highly sensitive to initial enrichment, and burnup.
- Recipes have a hard time with thermal spectrum reactors and MOX reactors.
- CyBorg

Joshua Peterson-Droogh – Cross section versus recipes for fuel cycle transition analysis using ORION

- Uses Orion to doing the modeling
- Example - Switch from uox to mox cycle
- 5 reactor stages to get to a MOX - driver - blanket steady state
- Modeling with cross sections makes the simulation simple in this case. As the recipes are not required for each stage. The model calculates the stages.
- Looking at the full domain space, and the sensitivities then cross sections make more sense than recipes.
- Orion has live streaming of results.
- Orion now has origen 6.2
- Orion uses cross sections from MC^2, origen and orion
- Forced steady state with recipes can cause some issues with reactor types that take a long time to reach a steady state.
- ‘We have to be careful when it comes to letting fuel cycle codes dictate reactor design’

Discussion 1 (Fri @ 11:30)

- Do equilibrium cases even exist? Has there ever been a equilibrium case? Fleets evolve over time. Few reactors are ever really in steady state.
- But how far away is ‘steady state’ from the actual state of things
- Where do we get data to check this? Actual spent fuel data from industry, but that is very difficult to get.

- In a well defined system, we may not even need a simulator, however in systems without good definitions, even if we can define all of the parts well, we may not define the system well.
- Models can save time if they store their recipe history, therefore saving time in the future.
- Do gain confidence in these tools, we need to be able to really accurate describe the limitations of bounds of each of the tools that we have. Until we can do that we can't really prove we are inside or outside of a specific domain.
- SHAMELESS CYCLUS PLUG

Yarden Post Workshop Discussion

- Is the maximum always the best? Sometimes one maximum is not good because of its stability. For example a very sharp maximum is unstable.
- Break up sections of data into groupings that are based on monotonic behavior from either a shared minimum or shared maximum
- This technique relies on breaking down n-dimensional structures into smaller dimensions for visualization.
- Surfaces are actually built using the data points in the sampling space.
- Along enough 'group' or 'patch' you can see how parameters behavior in respect to the objective value.
- This becomes difficult to read if we have too many dimensions, or dimensions that have values that change in a minor way.
- This might give a way of visualizing which parameters really matter in advanced fuel cycle cases by showing which parameters have a sensitivity.

Summary Statement

Fuel cycle simulation technologies connect a variety of spaces and domains, and their associated challenges:

- There are at least three distinct meanings for "uncertainty" - physical parameter uncertainty, model uncertainty, and unknowable future uncertainty. There is curiosity and a lack of consensus about which of these uncertainties is an important place to focus improvements.
- Benchmarking and Code-to-code comparison activities could be more community-driven. We recommend that the next workshop have a tie in with a communal (open) benchmarking activity.
- Fuel Cycle Game Design could benefit from discussion of simplification, integration with other energy systems, and inclusion of an economic model.
- Visualization - visualization as an exploratory tool could be leveraged in this field, particularly for the comparison of large numbers of simulations. There are three types of visualizations:
    - **Presentation**
        I know my data,
        I know what I want to show,
        I have a story to tell
        Specific answers to known questions, where the location of the necessary data is known. This can enable powerful inter-model comparisons.
    - **Search**
        I know what's in the data,
        I don't know where it is

    - **Exploration**
        I don't know what's in the data,
        I don't know what I'm looking for
        No set question to answer, looking through the data and exploring it to capture emergent features (to see if anything 'jumps out at you').
    - Possible combination of 2 or more of these.
- It is important to carefully justify choices related to reactor modeling fidelity, specifically with relation to the comparative strengths and weaknesses among recipe-type, cross-section driven, or other modeling choices (e.g., machine learning). These choices can be influenced by the objectives of the simulation (e.g., calculating economics, material flows, or fuel cycle metrics such as repository impacts) as well as the required computational scale (1 simulation to millions).
- This community could benefit from a glossary. What do we mean by:
    - Simulation
    - Scenario
    - Ensemble
    - Benchmark
    - Code-to-code comparison
    - Sensitivity study
    - Transition
    - etc.

  Defining these terms could be an activity for the next workshop.
- Use case open questions:
    - Who is the end user / consumer of these simulations? What do these stakeholders need/want out of a fuel cycle simulator? What kinds of questions do they want answered?
    - Does the demand/need for a perfect fuel cycle simulator exist?
    - Meta question: what questions are we (or should we be) asking?
    - Need more demand to drive improvements/priorities (use cases drive development; i.e., gap analysis)
    - Decisions about FC technology not necessarily driven by the fuel cycle itself, but by larger factors (e.g., economics); do fuel cycle simulations "matter" at the macro-level?
    - Is there a need to keep an up-to-date list of use cases (i.e. FCS requirements)?
    - What are the appropriate "boundaries" for FCS codes where its capabilities end and another's begin? (Mission creep / duplication of effort). Should we consider when to demarcate boundaries and "blend in" / hand off to complementary codes (e.g., for economics, other metrics)?

There is a need for this community of practice to have meetings and workshops such as this one. The discussion-led format is a highly effective way in which to engage this community. It is also desirable to have cross-pollination between this community of practice and other communities (e.g. NEA Expert Groups, IAEA collaborative projects, etc.).

Next steps include having a third workshop next year. Coupled with this workshop, well in advance, should be a call to participate in an open benchmarking activity. We recommend talks in groups of 3-4 talks, rather than 2-3. Additional flexibility in the schedule of discussion sessions would be desirable. Having a more tightly bound schedule, with maybe a couple of

longer talks to start each session may have advantages. The breakout sessions were engaging and productive, but the topics should ideally be conceived of earlier in the day if possible.