On the separation between physics-oriented research and ITER-driven research and the role of software performance.

Francesca M. Poli, Princeton Plasma Physics Laboratory

The workshop-level question: "Are we building tools to model ITER, or are we doing basic fusion research in a general sense?" can probably be reformulated in a "what do we need to address in the short and medium term for what we need to achieve in the long-term?"
While ITER should not be the ultimate goal, the simulation needs of ITER can probably be regarded as general.
When I think about the tools to model ITER (or to model any existing tokamak device), they should satisfy at a minimum level:

1) Accurate physics
2) Computationally stable
3) Fast

Accurate physics does not necessarily mean that the modeling tools should include all the physics, but that they have to include the relevant physics. This means that first principle physics codes should be used to develop reduced models for use iun time-dependent simulations. It is expected that the codes that are installed in IMAS (the ITER Integrated Modeling and Analysis Suite) have been verified and validated by the time they start to get used for ITER analysis and predictions, on existing devices in a wide range of operational regimes.

Let me focus on the second and third point, which is where collaboration between SciDAC and ASCR is most needed.
Having 'fast' modeling tools is driven by the need of being able to analyze discharges in-between shots (quick assessment of transport), predicting the plasma response in the upcoming shot for variations of the actuators input, analyzing the whole run day over night. These desires set tight constraints on the duration of an in-between shot simulation, which should be typically below one hour in the case of ITER, even lower in the case of existing devices.

Tests have been run at PPPL with TRANSP for NSTX-U in-between shot analysis along this direction. By tailoring the number of times the Neutral Beam calculations are performed, reducing the number of outputs to the minimum required, reducing the number of MonteCarlo particles, and running on a dedicated server, an analysis run has been taken down to a few minutes. This is an example of input optimization.

In collaboration with the Whole Device Modeling panel, this panel could perhaps work on a list of codes that are typically used in time-dependent simulations and start a discussion on how these codes can be optimized. It comes a time when parallelization is not sufficient any longer and it might be required to re-write the code or part of it. The discussion should extend to short-term goals and long-term goals. Short-term goals might focus on existing codes, medium to long-term goals on codes that are not routinely used in time-dependent simulations for scenario prediction, but that will be likely needed to be includes, like edge physics codes and reduced MHD stability codes.

At a minimum level, necessary requirements for a discharge simulation are the Heating and Current drive sources. Codes to calculate wave interactions in the regimes of the Ion Cyclotron harmonics or in the range of Lower Hybrid waves are complex and computationally expensive, and they require a coupling to a Fokker Planck solver. Is working on more efficient Fokker-Planck solvers an area where RF SciDAC and ASCR can work together? Another example is Neutral Beam calculations, which involve MonteCarlo statistics. In some regimes, like on ITER, it is required that the number of particles is large, to reduce statistical errors in the beam deposition.
What are the steps that can be undertaken to make these codes more efficient? If rewriting the NUBEAM code on GPU (which is what is being done at the moment at PPPL) is a short-term solution, what are longer term solutions? Would a reduction of a factor 10 in the computational time of Heating and Current Drive sources a realistic target? Another area where collaboration between SciDAC and ASCR is envisaged is integration of MHD codes. These are computationally expensive calculations for which the development of reduced models for use in time-dependent simulations (like sawtooth and Neoclassical Tearing Modes) would be required.

What can be done in the short and in medium term to facilitate the achievement of these performance goals? Some of these questions are probably not pertinent to this panel in the specific, but they offer points for discussion with other panels.

1) Identify critical questions: what physics problem is critical for ITER to be solved and how are we going to address it using the existing capabilities we have?
2) Do we have the right codes to answer the question or do we need new physics to be included in specific codes?
3) Understand that no individual code can do everything and that different physics problems can be addressed by different codes.
4) Codes are becoming more and more complex. Does this mean we need increasingly complex architecture and increasingly powerful computing capabilities or perhaps some of these codes should be re-written and optimized by a computer scientist? What is the balance between the two approaches and what are the benefits from a combined approach?
5) How can these codes be used to derive reduced models?
6) How can these reduced models be effectively integrated in a ‘production’ code?