FSP Advanced Physics Component: mission, work plan, and schedule

Xianzhu Tang (LANL)

September 17, 2009

On behalf of FSP Physics Component team

Outline

- Mission and tasks
- Work plan (WBS)
- Work schedule

http://fspcomp.web.lehigh.edu/index.php/Main_Page

Mission & Tasks

- Mission:
  - develop a plan for the identification, improvement, and creation of advanced software components to be used as modules in the integrated FSP framework.
- Tasks:
  - Assembling a team of experts in computational physics, applied math and computer science to carry out the planning exercise.
  - Performing an analysis of the science drivers to determine the needed physics capabilities and advanced code features.
  - Analyzing existing codes and libraries for their ability to meet the needs of science drivers and their readiness for incorporation into FSP.
  - Assessing and developing adequate verification methods.
  - Developing an effective management plan to address the gaps and produce a “living-scientific-road-map” that identifies viable deliverables.
  - Producing an implementation plan with initial technical approaches and milestones, estimate of manpower, computing resource, and funding.

Key characteristics of a FSP component

- Provide the simulation of key physical processes in the same or different physical domains.
- Have well-defined inputs and outputs that are clearly documented.
- Clearly documented to have been verified and validated for the regimes of physical parameters intended and open to retrospective verification review as needed.
- Conform to software development and management practices defined and accepted by FSP team.
- FSP component spans a wide range of fidelity and resolution requirements.

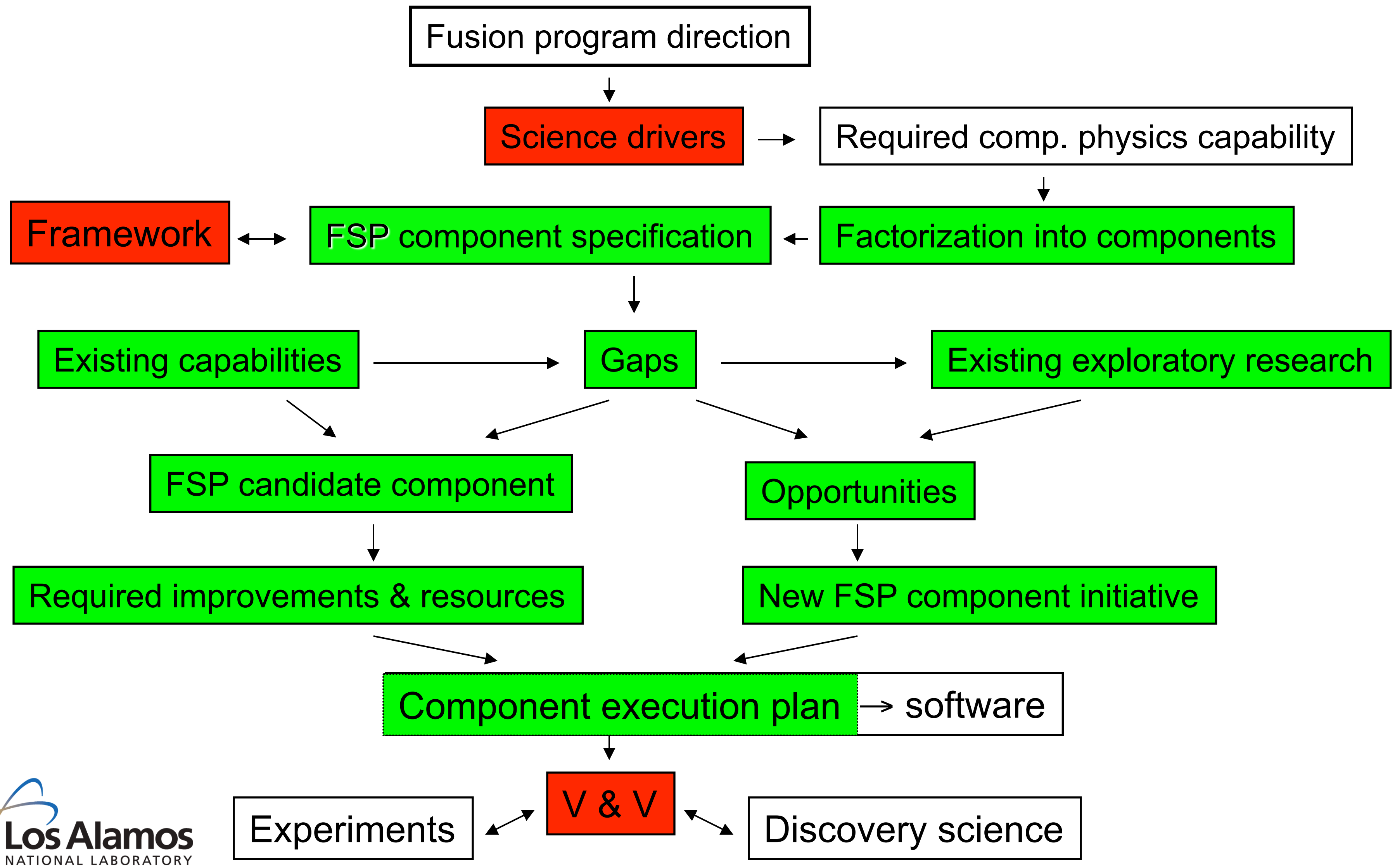
Workplan: (1) organization & approaches

- Team consists of experts in fusion science topical areas:
  - Paul Bonoli (MIT): auxiliary heating and wave-particle interaction.
  - Jeff Brooks (Purdue): Plasma/wall interaction.
  - Jeff Candy (GA): Transport & Turbulence.
  - Luis Chacon (ORNL): MHD & two-fluids.
  - C.S. Chang (NYU): Edge Physics.
  - Lori Diachin (LLNL): Applied Mathematics & Computer Science.
  - Nikolai Gorelenkov (PPPL): Energetic Particles.
  - Weixing Wang (PPPL): Transport & Turbulence.
- Team has identified and will engage a large pool of expert advisors
  - Include both code developers and science leaders.

Workplan: (1) organization & approaches

- Four example thrusts, emphasizing cross-integration.
  - (1) Plasma/materials interaction & edge plasmas
  - (2) Transport (+ heating + CD + EP + macro)
  - (3) Macrodynamics (+ heating + CD + EP + transport)
  - (4) Math and computer science
- Approaches: focused workshop, interviews, and solicitation of reports in response to questionnaires
  - Engaging the broader FES and ASCR communities.
  - Involving FES SciDAC centers: GPS-TTBP (P. Diamond); CSPM (W. Nevins); CEMM (S. Jardin); CSWPI (P. Bonoli); GSEP (Z. Lin).

Workplan: (2) determining component specification from science drivers



Workplan: (3) Assessing current capability and performing gaps analysis

- (3.a) Assessing existing physics component capabilities and their readiness for FSP integration
- (3.b) Assessing mathematical and computer science infrastructure needs for FSP components
- (3.c) Gaps analysis to provide prioritization for FSP component program directions

Workplan: (3.a) Readiness of current physics component capabilities

- Objectives:
  - Determine the initial set of physics component codes to be integrated into FSP framework
  - Determine the additional FSP work scope and hence cost estimate in terms of necessary software engineering and physics/algorithm upgrade beyond their SciDAC and base program support.
- Topical areas and candidate codes
  - Turbulence & transport
    - PIC: GTC/GTS, GEM,XGC; continuum: GYRO, GS2, Tempest; etc
  - MHD & two-fluids
    - NIMROD, M3D, BOUT, and various Newton-Krylov codes.
  - Auxiliary heating & RF
    - AORSA, TORIC. CQL3D, NUBEAM, ORBIT-RF, XGC, etc
  - Energetic particles
    - MHD-particle hybrid codes and gyrokinetic codes
  - Plasma/materials interaction
    - SOL codes, 6D sheath codes, and materials response codes (e.g. MD).

Workplan: (3.b) Mathematical and computer science needs for physics components

- Objectives:
  - Define the requirements for applied math and computer science infrastructure required for physics components.
  - Plan to meet these requirements
    - New development specifically for FSP and/or adaptation of existing tools from SC, NOAA, NSF, and ASC programs.
- Focus areas:
  - Verification and uncertainty quantification (separated out for their importance).
  - (3.b.1) Use of high-performance libraries to improve algorithmic performance.
  - (3.b.2) Tools for understanding code performance.
  - (3.b.3) Development of new algorithmic capabilities.

Workplan: (3.c) Gaps and opportunities analysis for FSP investment prioritization

- Objectives:
  - Provide the basis to identify additional resource requirements for new initiatives in FSP component development.
- Tasks:
  - Gaps and opportunities analysis
    - Fidelity of physics and mathematical models in relation to science driver requirements
    - Stability, accuracy, efficiency, and fidelity of coupling technique for multiphysics and multiscale integration within a component.
    - Accuracy and adaptivity of numerical discretization
    - Scalability of numerical algorithms to petascale and exascale cmputing
  - Develop criteria and process for prioritization in FSP investment
    - Balance the need for short term deliverables and strategic necessity of high risk/ high reward exploratory research.
    - Transparent mechanism for resource allocation and re-allocation.
  - Develop process and strategies for risk mitigation
    - Programmatic changes in fusion development path
    - Task failure in component development initiatives
    - Computer architectural and software tools evolution/revolution.

Workplan: (4) Robust verification and uncertainty quantification strategies for FSP components

- Integration into a whole device modeling framework requires that the components are validated and verified.
  - Code verification: determining if the component correctly implement the mathematical algorithm as specified.
  - Uncertainty quantification: determining the errors associated with the mathematical model, parameterizations, input data, and numerical solution, etc.
- Tasks:
  - Assess existing verification and uncertainty quantification methodologies to determine best practices and lessons learned in other large projects.
  - Design common processes for verification in component development.
  - Define process coordinating component development and experimental validation, and facilitating discovery science activities to guide V&V design.

Workplan: (5) Develop the FSP component program execution plan

- Objectives:
  - Ensure the successful execution of a committed component project.
- Tasks:
  - Assess and define the life cycle of component development
    - Risk mitigation requires accountability, responsibility, and a large degree of transparency.
    - Provide a reference map for tracking project progression, updating milestones, and planning contingencies.
  - Define software engineering standards for components
    - For both new development and re-engineering of existing components.
  - Develop the FSP component deliverables and schedule
    - 5, 10, 15 years perspective from the three prior FSP reports.
  - Ensure community assimilation and distribution
    - Acceptance standards and user support for scientific discovery.
  - Determining the resource requirements.

Work schedule (October 2009 – December 2009)

- Finalize and send out the first questionnaire to the commmunity soliciting component candidate. Questionaire will emphasize
  - physics objectives;
  - mathematical models, discretizations, and numerical methods;
  - software engineering and portability issues;
  - collecting benchmark and verification tests performed;
  - developmental goals.
- Determine the initial list of required components
  - develop the process of component specification (emphasizing physics functionality) from science drivers: A-list
    - carry out one or a few example studies using the initial list of science drivers.
  - develop the process of component specification (physics functionality, math models and discretization, software engineering, and verification) from framework design
    - by fusion science topical areas (equation-oriented: B-list);
    - by major off-normal events (event-oriented: C-list);
    - by physical domain decomposition (domain-oriented: D-list).
- Develop the process for selection and prioritization of the initial FSP component codes.

Work schedule (January 2010 – March 2010)

- Assess from community feedback the initial list of candidate codes for components in response to
  - initial list of science drivers (A-list);
  - fusion science topical areas (B-list);
  - major fusion plasma off-normal events (C-list);
  - physical domain decomposition, e.g. core-pedestal-sol-boundary. (D-list).
- Assess the numerical methods and use of state-of-the-art HPC practices.
- Assess the resource requirement to bring these candidate codes into FSP.
  - Analyze status of verification and benchmark information performed to assess code readiness

Work schedule (April 2010 – June 2010)

- Perform the initial gaps analysis between the existing capability from the community input and required component functionality from A, B, C, and D-lists.
- Identify candidate components that will benefit from advanced numerical algorithms and HPC tools.
- Propose to the community a set of key verification and benchmark tests for component readiness in each topical area.
- Finalize and send out the second questionnaire to the community soliciting input on ideas to address such gaps.
- Develop the process for selection and prioritization in new component development.

Work schedule (July 2010 – September 2010)

- Using the community feedback, carry out a combined gaps and opportunities analysis.
- Work with the development teams of key candidate components, assess the resources required to incorporate advanced numerical methods and HPC tools.
- Using the community feedback, finalize the required verification and benchmark tests.
- Develop the initial list of new component development for FSP investment and specify the resource requirement.

Work schedule (October 2010 – December 2010)

- Develop the FSP component program execution plan.
  - Assess and define the life cycle of component development.
  - Define software engineering standards for component.
  - Document the required verification and benchmark tests in each topical area.
  - Develop the FSP component deliverables and schedule.

Everything is on the wiki page

http://fspcomp.web.lehigh.edu/index.php/Main_Page

First questionnaire

- **Physics focus and programmatic scope of the component candidate (CC)**
  - Give a brief, high-level description of CC's functionality.
  - Describe CC's user base and application scope (who uses CC and for what).
- **Physical and mathematical models**
  - What are the equations solved in CC?
  - What are the limitations of CC imposed by orderings or by neglected terms?
- **Numerical approaches**
  - What are the discretization approaches for time and space?
  - What are the linear and nonlinear solvers involved?
  - Describe algorithmic scalability.
  - List other specialized performance-enhancing tools.
- **Software engineering issues**
  - Give a compete list of CC's inputs, e.g. the set of input parameters, the range of valid values for each, and their dependence on each other
  - Give a compete list of CC's outputs.
  - Give a list of CC's software dependencies.
  - Give a list of smaller components contained in CC; for example, CC1 and CC2.
  - List supported platforms and describe portability

First questionnaire

- **Verification**
  - Give a list of verification tests; highlight disagreements to identify problem areas.
  - Give the appropriate/valid subsets of the equations/models/parameters that can be used in an independent way.
  - Illustrate convergence to analytic or asymptotic solutions in special cases.
  - Rate of convergence studies to show the numerical methods are behaving as expected.
  - Can CC be instrumented to provide RHS source terms? This is to facilitate the use of the Method of Manufactured solutions to demonstrate convergence for a sufficiently rich test problem to showcase the physics of interest.
- **Performance**
  - Document processor scaling of time-to-solution on topical verification or other physically-relevant problems.
  - Describe performance variation with complexity of physics.
  - List the major serial and parallel bottlenecks (e.g., I/O, message-passing).
- **Developmental issues**
  - What problems would you like to solve with more development? How would this change the equations, discretization, or numerical methods that you use?
  - What tools do you wish you had available to you in your code development processes?