Modern, High Performance Transport and Thermal-Hydraulics Codes at LLNL

Scott McKinley
for
Richard Procassini, Christopher Clouse, Jeffrey Grandy and William Henshaw
Lawrence Livermore National Laboratory

Virtual Nuclear Reactor Center Workshop
21 – 22 August 2006

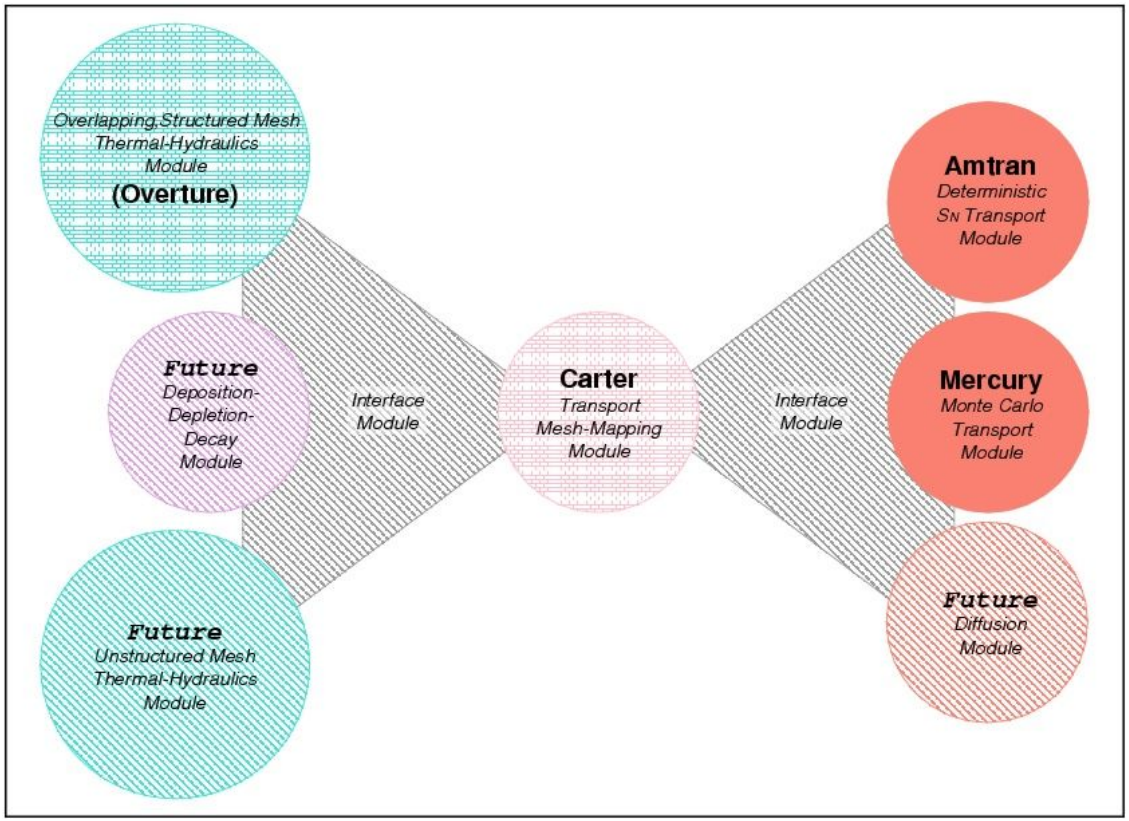
University of California
Lawrence Livermore National Laboratory

Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94551

A New Beginning for Reactor Analysis Codes at LLNL

- Development of a high-performance, coupled, multi-physics reactor analysis code is getting underway at LLNL with LDRD funding.
- This effort leverages mature code development activities which have been underway for more than a decade.

Schematic Layout of the LLNL Coupled, Multi-Physics Reactor Analysis Code



Modern Particle Transport Codes at LLNL

- Modern particle transport codes have been developed at LLNL over the past decade as part of the Advanced Simulation and Computing (ASC) program:
  - ***Amtran***: A discrete ordinates S_N (deterministic) particle transport code
    (Project Leader: *Chris Clouse* – clouse1@llnl.gov)
  - ***Mercury***: A Monte Carlo (stochastic) particle transport code
    (Project Leader: *Rich Procassini* – spike@llnl.gov)
  - ***Carter:*** A mesh-mapping code for transport mesh flexibility
    (Project Leader: *Jeffrey Grandy* – grandy1@llnl.gov)
- These codes were designed and developed for high-performance computing applications which require:
  - Very-large problem domains in space, energy, angle and time $\phi(\vec{r}, E, \hat{\Omega}, t)$
  - Solution on a wide variety of parallel computing platforms, using both message-passing and shared-memory data communication methods:
    - ➔ PCs → SMP nodes → Clusters → MPPs (such as Purple and BlueGene/L)

The *Amtran* S_N Transport Code

- `Status` → **Amtran** is a *production ready* code which:
  - Has undergone ~30 person-years of development
  - Has been V&V for a wide variety of applications
  - Runs efficiently on a *very* diverse set of serial and parallel computing platforms
  - Is an 'export controlled' code (true for *all modern* neutron transport codes)
  - Reference: Clouse, C. J., *Parallel Deterministic Neutron Transport with AMR*, in Computational Methods in Transport, edited by Graziani, F.R., Springer-Verlag, Pages 499 - 512 (2006).
- `Capabilities` → The features of the **Amtran** S_N code include:
  - Time dependent or static transport of neutrons (n) and gammas (γ)
  - Finite element treatment of angular fluxes
  - Support for 1-D spherical (radial), 2-D *r-z* cylindrical and 3-D Cartesian meshes
  - Variable spatial resolution via the Adaptive Mesh Refinement (AMR) method
  - Variable angular resolution via the Adaptive Directional Refinement (ADR) method

The *Amtran* S_N Transport Code *(continued)*

- `Capabilities` *(continued)* →
  - Static k_{eff} and α eigenvalue criticality calculations for neutrons
  - Multi-group treatment of cross sections
  - Thermal up- or down-scattering model
  - Thermally-broadened cross sections with *no* temperature-bin interpolation
  - Delayed neutron emission
  - Criticality probability method for estimating the probability of a chain reaction
  - Complete parallelization of phase space is supported:
    - Message-passing (MPI) decomposition over both spatial domains and energy groups $(\vec{r}, E)$
    - Shared-memory threading (Open-MP) over discrete ordinates (rays / angles) $(\hat{\Omega})$
    - Spatial load balancing via (static) domain overloading

The *Mercury* Monte Carlo Transport Code

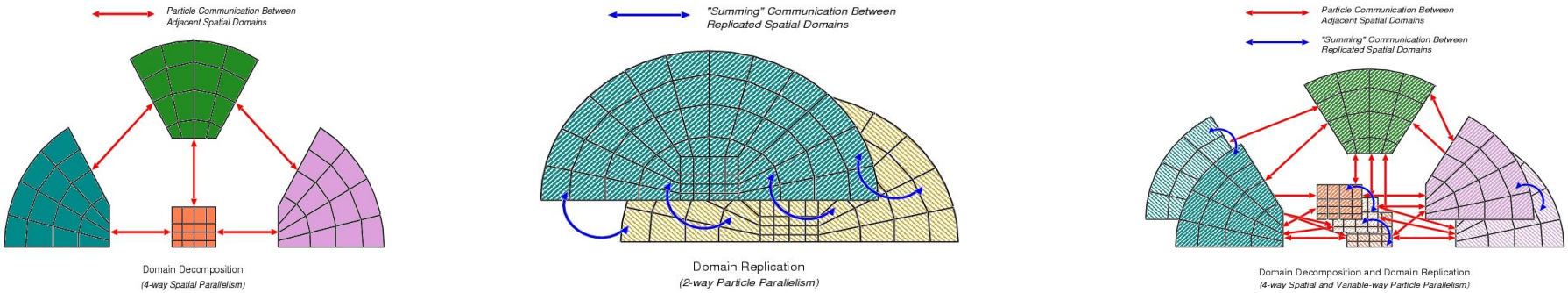
- `Status` → **Mercury** is a *production ready* code which:
  - Has undergone ~30 person-years of development
  - Has been V&V for a wide variety of applications
  - Runs efficiently on a *very* diverse set of serial and parallel computing platforms
  - Is an 'export controlled' code (true for *all modern* neutron transport codes)
  - References: See the **Mercury** web site → http://nuclear.llnl.gov/mercury
- `Capabilities` → The features of the **Mercury** Monte Carlo code include:
  - Time dependent or static transport of neutrons (n), gammas (γ) and light charged ions $({}^{1}H, {}^{2}H, {}^{3}H, {}^{3}He, {}^{4}He)$
  - Particle tracking through a wide variety of problem geometries:
    - *Mesh:* 1-D spherical (radial), 2-D *r-z* structured and quadrilateral unstructured, 3-D Cartesian structured and tetrahedral unstructured
    - *Non-Mesh:* 3-D combinatorial geometry

The *Mercury* Monte Carlo Transport Code *(continued)*

- `Capabilities` *(continued)* →
  - Static k_{eff} and α eigenvalue criticality calculations for neutrons
  - Dynamic α calculations for all types of particles
  - Multi-group and continuous energy treatment of cross sections
  - Thermal up- or down-scattering model
  - Thermally-broadened cross sections with *linear* temperature-bin interpolation
  - Deposition of energy, plus isotope depletion / production via nuclear reactions
  - Criticality probability method for estimating the probability of a chain reaction
  - Support for general sources and tallies

The *Mercury* Monte Carlo Transport Code *(continued)*

- `Capabilities` *(continued)* →
  - A flexible, hybrid approach to parallelism is supported:
    - ➔ *Domain Decomposition:* Spatial parallelism via message passing (MPI)
    - ➔ *Domain Replication:* Particle parallelism via message passing (MPI)



    - ➔ *Task Decomposition:* Particle parallelism via shared-memory threads (Open-MP)
  - These methods may be used individually or in combination
  - Two forms of dynamic load balancing are supported. Each responds to particle workload in an effort to achieve efficient parallel calculations:
    - ➔ *Dynamic Replication:* Dynamic, variable assignment of processors to domains
    - ➔ *Dynamic Decomposition:* Dynamically sized spatial domains

The *Carter* Mesh Mapping Code

- `Status` → **Carter** is a *production ready* code which:
  - Has undergone ~5 person-years of development
  - Runs efficiently on a *very* diverse set of serial and parallel computing platforms
- `Capabilities` → The features of the **Carter** mesh mapping code include:
  - Second-order accurate mapping of mesh-based fields between a *general donor* (thermal-hydraulics) mesh to a *structured target* (transport) mesh
  - Permits efficient, coupled, multi-physics calculations via different / optimal meshes for the thermal-hydraulic and transport modules:
    - Disparate donor / target mesh types (connectivities)
    - Disparate donor / target mesh resolutions
  - Parallel computations are supported via:
    - Spatial domain decomposition
    - Static load balancing

Modern Thermal-Hydraulics Codes at LLNL

- A modern thermal-hydraulics codes is being developed at LLNL within the **Overture** framework, which has been developed with support from the DOE Office of Science.
  (Project Leader: *William Henshaw* – henshaw@llnl.gov)
- `Status` → **Overture** is a *near-production quality* code framework which:
  - Has undergone >10 person-years of development
  - Has been used to model diverse physics: flow, heat transfer, electromagnetics, etc.
  - Reference: See the **Overture** web site → http://www.llnl.gov/CASC/Overture
- `Capabilities` → The features of the **Overture** code framework include:
  - Solving coupled sets of PDEs on overlapping, structured (overset) meshes.
  - Incompressible flow and thermal transfer modules are being integrated to produce a thermal-hydraulics (T-H) module.
  - High accuracy, parallel multi-grid PDE solvers are under development for use by the T-H module.
  - Flexible restart capabilities are currently limited and need to be extended.

Acknowledgments

This work was performed under the auspices of the

U. S. Department of Energy by the University of California

Lawrence Livermore National Laboratory

under Contract W-7405-Eng-48.