MCNP6 Code Developments and CGMF/FREYA Integration and Validation

Michael E. Rising
Monte Carlo Algorithms, Codes and Applications
Los Alamos National Laboratory

NA-22 Project Review
Los Alamos, NM

July 31, 2018
Outline

• Background on MCNP6

• Release of MCNP6.2
  • With CGMF & FREYA
  • With New Utilities

• Ongoing Validation Efforts

• Code Modernization

• Conclusions & Upcoming Work
Background of MCNP6

- The history of MCNP can be traced back to the early days of LANL

- MCNP 6.1.0 released by RSICC in July 2013
  MCNP6.1 + MCNP5-1.60 + MCNPX-2.70
  Nuclear Data Libraries + MCNP Reference Collection

- MCNP 6.1.1 Update in July 2014

- MCNP 6.2.0 release to RSICC in April 2018

- MCNP5 & MCNPX are frozen – no future releases
Release of MCNP6.2
(see mcnp.lanl.gov)

**2017 NA-22 Collaboration Meeting**

**MCNP6.2 Release Notes and Manual**

**New Utilities**

**Release Information for MCNP6.2 Release**


**Data Changes for the MCNP6.2 Release**, LA-UR-17-21486, LA-UR-17-20703


2.1 PHYSICS

2.1.1 Correlated prompt fission neutron and gamma-ray emission models (CGMF & FREYA):

Two new correlated fission event generators, CGMF and FREYA, have been integrated into the code to address needs within the nuclear nonproliferation and safeguards communities for high-fidelity models of the neutron and gamma-ray emissions from both spontaneous and neutron-induced fission processes [2]. The ultimate use of these models, currently under active development at LANL and Lawrence Livermore National Lab/Lawrence Berkeley National Lab, respectively, is to provide a predictive capability in simulating the unique signatures of special nuclear materials in situations where multiple detectors may be used in time-coincidence resulting in correlated counts from fission events. The new fission models can only be used for fixed-source calculations and are turned on with the FMULT card using the METHOD keyword.
Ongoing Validation Efforts
(MCNP developer maintained, automated suites)

**Verification Suites**

- **REGRESSION**
  - Run by developers for QA checking

- **VERIFICATION KEFF**
  - Analytic benchmarks, exact solutions for $k_{\text{eff}}$
  - Continuous-energy & multigroup

- **VERIFICATION GENTIME**
  - 10 benchmarks for reactor kinetics parameters

- **KOBAYASHI**
  - 6 void & duct streaming problems, with point detectors, exact solutions

- **Ganapol Benchmarks**
  - Exact, semi-analytic benchmark problems
  - Fixed source, not criticality

- **Gonzales Benchmark**
  - Exact analytic benchmark with elastic scatter, including free-gas scatter

**Validation Suites**

- **VALIDATION CRITICALITY**
  - 31 ICSBEP Cases, too small for serious V&V
  - Today, used for
    - Code-to-code verification, with real NCS problems & data
    - Compiler-to-compiler verification, with real NCS problems & data
    - Timing tests for optimizing MCNP coding & threading
  - Run at least weekly, to check MCNP6 for NCS

- **VALIDATION CRIT EXPANDED**
  - 119 ICSBEP Cases
  - Broad-range validation, for developers

- **VALIDATION CRIT WHISPER**
  - 1101 ICSBEP Cases
  - Used with Whisper methodology for serious validation
Ongoing Validation Efforts

- MCNP has historically had extensive verification and validation
  - Need to extend and leverage this work for correlated fission multiplicity applications

- Over the past 1-2 years, this important validation work for the correlated fission models has been getting started

- MCNP6.2 and MCNPX/PoliMi code-to-code comparison
  - Not really verification or validation
  - Help in understanding differences in codes

- Creating an automated subcritical benchmark test suite
  - Validation with evaluated benchmark quantities

- Using the criticality benchmark test suites
Ongoing Validation Efforts
MCNP6.2 – MCNPX/PoliMi Comparisons

• Follow-up of 2014 NSE paper by S.A. Pozzi et al.
• Presented at 2017 IRRMA X (LANL-UM collaboration)

• MCNP6.2 Simulations
  • Binary PTRAC file written and processed by DRiFT using MCNPtools
  • Convert PTRAC using MCNPtools for MPPost code detector response

• MCNPX-PoliMi Simulations
  • Collision file used for MPPost code detector response processing
  • ASCII PTRAC file written and processed by DRiFT using MCNPtools

• Count rates/pulse height spectra between transport codes and multiplicity models are consistent

<table>
<thead>
<tr>
<th></th>
<th>Total Counts (#/fission)</th>
<th>Correlated Counts (#/fission)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experimental rate</td>
<td>0.142</td>
<td>0.00439</td>
</tr>
<tr>
<td>Exp. light output threshold</td>
<td>64 keVee</td>
<td>100 keVee</td>
</tr>
<tr>
<td>DRiFT rates</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCNP6.2, FMULT</td>
<td>0.211</td>
<td>0.0116</td>
</tr>
<tr>
<td>PoliMi, FMULT</td>
<td>0.212</td>
<td>0.0116</td>
</tr>
<tr>
<td>MCNP6.2, FREYA</td>
<td>0.209</td>
<td>0.0119</td>
</tr>
<tr>
<td>MCNP6.2, CGMF</td>
<td>0.212</td>
<td>0.0122</td>
</tr>
<tr>
<td>PoliMi, IPOL(1)=1</td>
<td>0.211</td>
<td>0.0121</td>
</tr>
<tr>
<td>PoliMi, IPOL(1)=10</td>
<td>0.211</td>
<td>0.0119</td>
</tr>
<tr>
<td>MPPost rates</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCNP6.2, FMULT</td>
<td>0.138</td>
<td>0.00398</td>
</tr>
<tr>
<td>PoliMi, FMULT</td>
<td>0.138</td>
<td>0.00398</td>
</tr>
<tr>
<td>MCNP6.2, FREYA</td>
<td>0.141</td>
<td>0.00468</td>
</tr>
<tr>
<td>MCNP6.2, CGMF</td>
<td>0.143</td>
<td>0.00490</td>
</tr>
<tr>
<td>PoliMi, IPOL(1)=1</td>
<td>0.143</td>
<td>0.00461</td>
</tr>
<tr>
<td>PoliMi, IPOL(1)=10</td>
<td>0.141</td>
<td>0.00452</td>
</tr>
<tr>
<td>Sim. light output threshold</td>
<td>53 keVee</td>
<td>100 keVee</td>
</tr>
</tbody>
</table>
Ongoing Validation Efforts
Subcritical Benchmark Validation

- Leveraging benchmark quality subcritical experiments
- Two papers with lead author J.A. Arthur (CNEC Student)
  - “Validating the performance of correlated fission multiplicity implementation in radiation transport codes with subcritical neutron multiplication benchmark experiments”, Annals of Nuclear Energy
  - “Validation of MCNP6 Using Subcritical Benchmark Experiments”, submitted to 2018 ANS ANTPC
- These applications represent “integral” quantities that can be sensitive to the correlated multiplicity models
Ongoing Validation Efforts
Criticality Benchmark Validation

- Leveraging benchmark quality criticality experiments
- One paper with lead author D. Timmons (UNM Student)
  - “Evaluating the MCNP6.2 Correlated Fission Multiplicity Models for Criticality Calculations”, submitted to 2018 ANS Winter Meeting
- The default MCNP5/6 have been extensively validated for criticality safety applications
- MCNP6.2 disallows the use of the correlated fission multiplicity models in criticality safety because a serious lack of validation
- Using these models in criticality calculations is currently being investigated

<table>
<thead>
<tr>
<th>Model</th>
<th>GODIVA $k_{\text{eff}}$</th>
<th>JEZEBEL $k_{\text{eff}}$</th>
<th>FLAT23 $k_{\text{eff}}$</th>
<th>FLAT25 $k_{\text{eff}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>0.99987(19)</td>
<td>0.99987(19)</td>
<td>0.99915(30)</td>
<td>1.00331(30)</td>
</tr>
<tr>
<td>LLNL</td>
<td>0.99897(34)</td>
<td>0.99515(31)</td>
<td>0.99673(37)</td>
<td>1.00005(37)</td>
</tr>
<tr>
<td>FREYA</td>
<td>1.00053(34)</td>
<td>1.00097(31)</td>
<td>1.00121(39)</td>
<td>1.00568(37)</td>
</tr>
<tr>
<td>CGMF</td>
<td>0.99589(32)</td>
<td>0.99500(30)</td>
<td>0.99730(55)</td>
<td>0.99928(55)</td>
</tr>
</tbody>
</table>
Ongoing Validation Efforts

• Part of the reason that MCNP has a reputation as the “gold standard” for many applications is because of the validation testing
  • Constant and automated testing done by code developers
  • Regular reports issued on the validation test suite results
  • Changes to the code and nuclear data can be immediately tested

• Critical and subcritical benchmark applications are being leveraged to test the correlated fission models
  • These applications are relevant to the models
  • However, they are not highly sensitive to CGMF/FREYA

• A new test suite dedicated to correlated fission model experiments would be a great addition to the MCNP testing repertoire
  • Immediate feedback to the user community on application validity
  • Immediate feedback to the CGMF/FREYA developers on model improvements
Code Modernization

- The state of MCNP6.2
  - The long (and occasionally windy) road to today

Currently, the code base includes:
- 431K lines of code + 88K lines of comments ~ 500K total
- ~ 9-10 FTE total primarily funded through ASC / NCSP / SC / Institutional

This is a big job for all of us
- Seeking the best and brightest new staff members, of course
- Young staff are used to and expect more modern code practices!
Code Modernization

• At LANL, there exists a strong will to see MCNP succeed

• How do we (curators of the code) setup for long term success?
  • Create a code modernization plan (under development)

• Improve code development practices as a team
  • Code design documentation and prototyping practices
  • Peer-review and testing of all integrated code
  • Complete documentation of all code work
  • With all of these improved team processes documented and in place leads to improved SQA

• Adopt modern software development tools
  ✓ Version control system – CVS ➔ git
  ✓ Build, test and package software – GNU Make ➔ CMake
  ✓ Repository management / code reviews – TeamForge / Gerrit ➔ Bitbucket
  ✓ Artifact / issue tracking – TeamForge ➔ Jira
  ✓ Team communication / wiki – TeamForge ➔ Confluence
  • Continuous build and testing system – CBTS ➔ ???
Code Modernization

• With all of the history of MCNP comes…

... many benefits

• Very feature-rich, many applications possible
• Extensive testing (90% coverage of code)
• Validation for important applications (i.e. NCS)
• Expert experience and guidance available

... many challenges

• A tangled mess of source code, dependencies, etc. (see dependency graph ➔)
  • No modularity – need to break dependencies
  • Unintended side-effects when calling a function
• Diminished knowledge of some features
• Structural upgrades sorely needed
  • Remove cryptic variable naming (2-3 letter variables)
  • Organize data structures in a logical way
Code Modernization

- How does the MCNP code modernization efforts impact this project?

- Examples from previous venture integration efforts
  - CGMF now contains API to interact with MCNP
  - FREYA is now Fortran 2003 compliant

- During current project (CGMF-specific)
  - Using newer tools (git, Bitbucket, etc.)
  - Currently implementing CMake to replace build system
  - Will address automated testing soon
  - Include C/Fortran/Python interface

- With these kinds of changes, the burden on MCNP developers to maintain external dependencies is reduced
Conclusions & Upcoming Work

• **MCNP6.2 includes CGMF & FREYA!**
  • Documented in many places, with many links on website
  • Release notes and manual updated with correlated model info
  • New tools available to support users

• **Validation is essential**
  • Historically, MCNP focus on criticality and shielding applications
  • Need to expand validation suites to correlated fission multiplicity applications – ongoing & future work

• **MCNP6 is undergoing serious code modernization**
  • Maintain performance and validation
  • Need modern code for long-term sustainability

• **Need to improve parallelism and performance**
Questions?