Nuclear Instruments and Methods in Physics Research B 376 (2016) 168–170

Contents lists available at ScienceDirect

Nuclear Instruments and Methods in Physics Research B

journal homepage: www.elsevier.com/locate/nimb

Plans for performance and model improvements in the LISE^{++} software

M.P. Kuchera *, O.B. Tarasov, D. Bazin, B.M. Sherrill, K.V. Tarasova

National Superconducting Cyclotron Laboratory, Michigan State University, 640 South Shaw Lane, East Lansing, MI 48824, USA

ARTICLE INFO

Article history:
Received 31 July 2015
Received in revised form 4 December 2015
Accepted 10 December 2015
Available online 23 December 2015

Keywords:
LISE^{++}
Fragment-separator
Simulation

ABSTRACT

The LISE^{++} software for fragment separator simulations is undergoing a major update. LISE^{++} is the standard software used at in-flight separator facilities for predicting beam intensity and purity. The code simulates nuclear physics experiments where fragments are produced and then selected with a fragment separator. A set of modifications to improve the functionality of the code is discussed in this work. These modifications include transportation to a modern graphics framework and updated compilers to aid in the performance and sustainability of the code. To accommodate the diversity of our users' computer platform preferences, we extend the software from Windows to a cross-platform application. The calculations of beam transport and isotope production are becoming more computationally intense with the new large scale facilities. Planned new features include new types of optimization, for example, optimization of ion optics, improvements in reaction models, and new event generator options. In addition, LISE^{++} interface with control systems are planned. Computational improvements as well as the schedule for updating this large package will be discussed.

1. LISE^{++} development plans

LISE^{++} is the standard software used at in-flight separator facilities for predicting beam intensity and purity. The code simulates nuclear physics experiments where fragments are produced then selected with a spectrometer. The LISE^{++} software package is being transported to a modern graphics framework with new compilers to aid in the performance and sustainability of the code [1–3]. To accommodate the various operating system preferences of the users, LISE^{++} will be adapted for cross-platform compatibility. Windows, Mac and Linux operating systems will be supported. This work aims to address the increased computational demands associated with more complicated devices at new large scale facilities. In order to perform the necessary calculations in an acceptable time, code optimization and parallel methods will be applied. New features such as optimization, for example, of ion optics, improvements in reaction models, and new event generator choices are planned. Finally creation of a LISE^{++} interface with control systems is envisioned. The LISE^{++} development plan, as summarized in Fig. 1, includes improvements in the software as well as the models.

2. Nuclear physics

Creation of reaction models within the LISE^{++} framework and implementation of modern powerful algorithms in the code remains an important priority for LISE^{++} development. Recent experiments at RIKEN and GSI showed that the LISE^{++}. Three-Excitation-Energy-Regions model [4] does not reproduce yields of high-Z neutron rich isotopes well with default parameter settings which indicates the necessity of improving in-flight fission yield calculations. A new analytical Abrasion-Fission model will be developed soon by Monte Carlo benchmarking. A dissipation step will be inserted in the Abrasion-Ablation model to improve quantitative agreement with measured projectile fragmentation cross sections at lower energies and with heavy targets.

3. Software development

The LISE^{++} software suite is undergoing a major transportation to a new graphics framework in order to support modern compilers and computing methods.

Qt *framework*. For compatibility with future operating systems, the graphics framework is being transported to Qt. Benefits include provisions for 64-bit operation, cross-platform compatibility, and the ability to take advantage of computational advances. Qt was chosen as the graphics framework based on its cross-platform capabilities, large feature set, and widespread use in cross-platform C++ applications. Qt is a package of C++ graphics libraries that has great benefits for developing applications for nearly all operating systems and devices. The code remains essentially identical for all platforms, which allows for easy compilation of executable programs for any operating system or device. We will release Windows, Mac, and Linux versions of the software.

* Corresponding author.

http://dx.doi.org/10.1016/j.nimb.2015.12.013
0168-583X/© 2015 Elsevier B.V. All rights reserved.

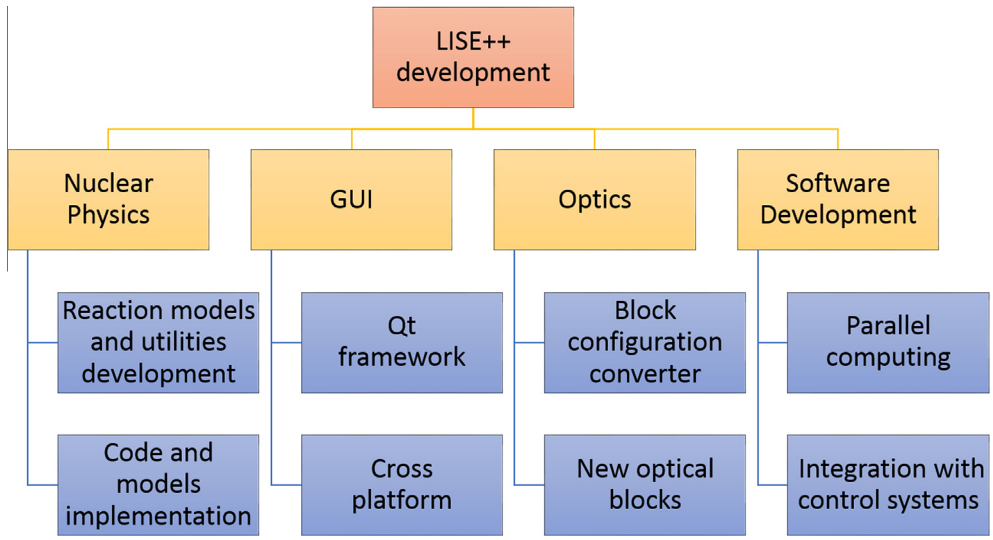

Fig. 1. A schematic diagram of the LISE++ development plans.

Parallel computing. To take advantage of modern computing architecture, parallel computing methods are essential in achieving faster computation. Once transportation of LISE++ to the Qt graphics framework is complete, we will be able to implement parallel computing on personal computers using OpenMP. OpenMP, whose name derives from Open Multi-Processing, is an application programming interface (API) for shared-memory parallel processing [5]. This API is useful for multi-core CPUs, such as those found in a personal computer. In the future, large-scale calculations using supercomputers or many-core machines using MPI is also planned. MPI stands for Message Passing Interface, which is an API for communication between processors [6]. Therefore, MPI can be used on distributed memory systems, such as supercomputers, for massively parallel computations.

Integration with control systems. In order to directly assist the tuning of a separator, the LISE++ program will be integrated with control systems. This will be tested at labs such as NSCL and GSI. This will be accomplished, for example, by direct integration of the LISE++ software with the A1900 controls at NSCL in order to update inputs to calculations based on experimental measurements.

4. Ion optics

New features are being implemented to improve the optics in LISE++. There are two different configuration modes in LISE++, segmented and extended. The segmented configuration is comprised of Sector (S) blocks, which are dispersive blocks that can contain quadrupole and dipole magnets, drift elements, along with other optical components. In the extended mode, all elements are separated into their own blocks, called Element (E) blocks, and their matrices can be calculated by the LISE++ code. For example, a single dipole S block in the A1900 fragment separator is the equivalent of 16 E blocks. Some of these features include a way to group and ungroup optical blocks within an extended configuration, the addition of minimization procedures, and new gas optical blocks.

Block configuration converter. This new tool will be built around a new type of block, labeled G (Group), which allows the grouping and ungrouping of E blocks, see Fig. 2. The tool can be applied to create sector configurations for fast analytical calculations. E-block properties of a G-block will be used in Monte Carlo mode for high order optics transmission calculations, and S-block properties in the *Distribution* mode for experiment planning. The *Distribution* mode uses an analytical solution to calculate the evolution of phase space distributions with use of transport integral methods [7] that is much faster than traditional Monte-Carlo methods.

Minimization procedures. 1st and 2nd order minimization of optics by varying magnetic fields (and/or electric voltages) of E-blocks has been applied to Distribution mode using the levmar package [8], which is now available in LISE++ since the version 9.9.10. The levmar package is a C implementation of the Levenberg–Marquardt algorithm, and iterative algorithm to find local minima. High-order minimization for distribution in Monte-Carlo modes will also be implemented. High-order minimization for distribution in Monte-Carlo modes will also be implemented.

LISE++ *blocks.* Gas-filled optical blocks will be developed, by allowing existing magnetic block types to have the option to be filled with gas. E-versions will be created for some already existing S blocks, and multipole properties of the drift block will be moved to a special block.

5. LISE++ transportation to a new graphics framework

On March 4, 2015, the LISE++ Qt utilities package was released. This is a cross platform package of the satellite utilities distributed with LISE++ to Qt. The package consists of five stand-alone programs useful for experiment design. They are PACE4, Charge, Global, the Kantele spectrometric calculator, and a units converter. Versions are available for Windows, Mac, and Linux. New documentation for the utilities is available on the LISE++ website [9]. The utilities package is shown in Fig. 3. Work is now underway to transport the rest of the LISE++ software to the Qt graphics framework.

6. Summary

Work is underway to update the LISE++ software suite. New code capabilities such as parallel computing, cross-platform functionality, and integration with control systems are planned and work is in progress. In order to implement these changes, the code is being transferred to a new graphics framework, Qt, and work is being done to ensure compliance with ISO C++ for use with modern compilers. The first release within the new framework is the LISE++ Qt utilities package, which was first released on March 4, 2015 [9]. The package contains five satellite utilities that are

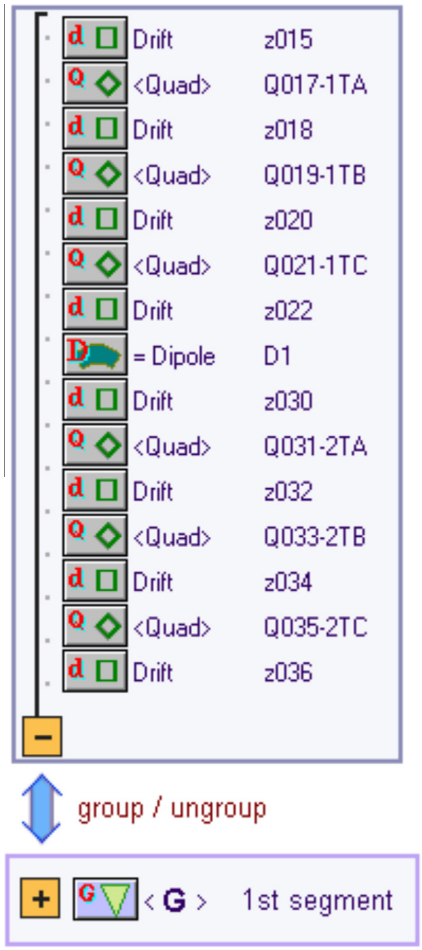

Fig. 2. The new LISE++ feature for converting a series of extended blocks to a single segmented block.

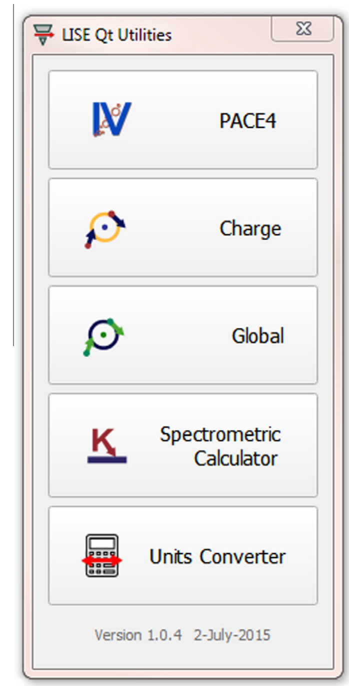

Fig. 3. This is the home interface for the LISE++ Qt utilities package, which launches the five satellite programs listed.

distributed in the LISE++ software suite. Computational speedup has become highly requested from users at many facilities, and computation time becomes even more crucial with the new large-scale nuclear physics facilities under construction, such as FRIB. Facilities such as NSCL and GSI have keen interest in integrating the LISE++ software with their control systems. With the completion of this work, we aim to address many requests of the LISE++ user base.

Acknowledgements

This material is based upon work supported by the Department of Energy National Nuclear Security Administration under Award Number: DE-NA0000979 through the Nuclear Science and Security Consortium. This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or limited, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

References

[1] O.B. Tarasov, D. Bazin, LISE++: radioactive beam production with in-flight separators, Nucl. Inst. Methods B 266 (2008) 4657–4664.
[2] LISE++ website, <http://lise.nscl.msu.edu>.
[3] O.B. Tarasov, D. Bazin, LISE++: exotic beam production with fragment separators and their design, these proceedings, 2015.
[4] O.B. Tarasov, LISE++ development: abrasion – fission, NSCL Preprint MSUCL1300, 2005.
[5] The OpenMP Architecture Review Board, Openmp, the openmp api specification for parallel programming, <http://openmp.org/wp/about-openmp/>, 2012.
[6] The MPI Forum, Message passing interface forum, <http://mpi-forum.org>, 2015.
[7] D. Bazin, B.M. Sherrill, Transport integral: a method to calculate the time evolution of phase-space distributions, Phys. Rev. E 50 (1994) 4017–4021, http://dx.doi.org/10.1103/PhysRevE.50.4017. URL: <http://link.aps.org/doi/10.1103/PhysRevE.50.4017>.
[8] M.I.A. Lourakis, levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++, http://www.ics.forth.gr/ lourakis/levmar/, 2004.
[9] LISE++ Qt utilities package, <http://lise.nscl.msu.edu/porting>.