

A Simple Quantum Integro-Differential Solver (SQulDS)[☆]Carlos A. Argüelles^{a,*}, Jordi Salvado^b, Christopher N. Weaver^c^a Department of Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA^b Departament de Física Quàntica i Astrofísica and Institut de Ciències del Cosmos, Universitat de Barcelona, Diagonal 647, E-08028 Barcelona, Spain^c Enrico Fermi Institute, University of Chicago, Chicago, IL 60637, USA

ARTICLE INFO

Article history:

Received 7 April 2020

Accepted 19 May 2020

Available online 1 June 2020

Keywords:

Quantum mechanics

Ordinary differential equations

SU groups

ABSTRACT

Simple Quantum Integro-Differential Solver (SQulDS) is a C++ code designed to solve semi-analytically the evolution of a set of density matrices and scalar functions. This is done efficiently by expressing all operators in an SU(N) basis. SQulDS provides a base class from which users can derive new classes to include new non-trivial terms from the right hand sides of density matrix equations. The code was designed in the context of solving neutrino oscillation problems, but can be applied to any problem that involves solving the quantum evolution of a collection of particles with Hilbert space of dimension up to six.

New version program summary

Program Title: SQulDS

CPC Library link to program files: <http://dx.doi.org/10.17632/hnchbntnybs.1>Developer's repository link: <https://github.com/jsalvado/SQulDS>

Licensing provisions: GNU Lesser General Public License, version 3.

Programming language: C++11.

External routines: GNU Scientific Library (<http://www.gnu.org/software/gsl/>).

Journal reference of previous version: Comput. Phys. Commun. 196 (2015) 569–591

Does the new version supersede the previous version?: Yes.

Nature of problem: Solve the evolution of open quantum systems of Hilbert space dimension N with self interactions and interaction with classical fields.

Solution method: The algebra is implemented as a C++ object and is embedded into the GSL ordinary differential equation solver.

Reasons for the new version: Added new features and improved runtime.

Summary of revisions:

- Library names have been moved into the 'squids' namespace.
- Support for extraction of SU vectors and basis transformations to GSL matrices.
- Implementation of matrix exponentiation for use cases like evolution with non-diagonalized Hamiltonians.
- Implementation of a number of useful matrix operations on SU vectors, including applying arbitrary unitary transformations, extracting real and imaginary components, transposition, and generalized, weighted rotations.
- Support for obtaining the eigensystem which diagonalizes an SU_vector.
- Explicit disallowance of creating or manipulating SU(1) vectors, which are degenerate with scalars.
- A fix to a severe issue in the use of the GSL ODE interface, now giving good output for much less stringent tolerance values.
- Optimization of element-wise operations on SU vectors without the need to allocate temporary storage.
- Redesign of SU vector operations for vectorizability, including optimized allocation of element storage with optimal alignment.
- Implementation of a more efficient version of SU vector time evolution which can share initialization costs across multiple evolutions by the same operator over the same time interval.

[☆] The review of this paper was arranged by Prof. J. Ballantyne.

* Corresponding author.

E-mail address: carguelles@icecube.wisc.edu (C.A. Argüelles).

- A more precise definition of the meter.
- Improvements to configuration and build system.
- Clean up of capitalization for file and class names.
- Rearrangement of project files to a more standard layout.
- Elimination of memory leaks.
- Limited support for the Intel and Portland Group compilers. These should not be considered guaranteed to work, but have been given minimal testing and appear compile the library correctly.
- Added additional unit conversions to SQulDS constants.

Additional comments including restrictions and unusual features: The code is only implemented up to Hilbert spaces of dimension six, but a Mathematica notebook is provided in order to generate higher dimensional solutions. Furthermore, only ordinary differential equation solution methods that require only the first derivative can be used.

Acknowledgements CAA is supported by NSF grant PHY-1912764. JS acknowledges financial support from EU Networks FP10ITN ELUSIVES (H2020-MSCA-ITN-2015-674896) and INVISIBLES-PLUS (H2020-MSCA-RISE2015-690575), by the MINECO grant FPA2016-76005-C2-1-P and by the Maria de Maeztu grant MDM-2014-0367 of ICCUB.

© 2020 Elsevier B.V. All rights reserved.
