jPublicFewBodySolver v.12.04 release notes

Vladimir Roudnev*and Mike Cavagnero

May 1, 2012

Abstract

Here are some notes on the preliminary release of the three-body code being prepared. We briefly explain how to use the code, the units employed, scaling factors to be used and give other information that may be useful.

General notes

The puprose of the code is performing quantum calculations of scattering and bound states of atomic three-body systems. The code is based on the numerical techniques, that are generally described in Refs. [1, 2, 3]. More particular description of the code will be provided later.

The code consists of two parts. The first part is a configurator that simplifies composing the necessary configuration files. The second part is the 3-body computational kernel that performs the actual calculations. The code comes with a preset of model potentials that have self-explaining names. The references to the original publications for the potentials will be added in the future versions of the code.

The current version of the code has some technical limitations. It does not calculate any bound-state related observables besides the energy spectrum. The scattering calculations are limited to scattering of a single s-wave bound state below the three-body or the first excitation threshold.

The code is developed with Java Development Kit (JDK) version 6 (it may work under Java Runtime Environment (JRE) starting version 1.2, but no tests have been performed.) It is reommended to download and install the latest version of the JRE from *http://www.java.com/en/download/index.jsp* .

Units

The potentials take their arguments in atomic units (Bohr) and return the values in a.u.e. for energy and a.u. for other observables. The actual calculations, however, are performed in scaled units.

*roudnev@pa.uky.edu, Department of Physics and Astronomy University of Kentucky Lexington, Kentucky 40506-0055

The kinetic energy in the Faddeev operator can be reduced to 6-dimensional Laplacian purely by rescaling Jacobi coordinates. Applying this approach directly, however, may lead to numerical instabilities, as in the scaled coordinates the box containing the numerical solution can become inappropriately large, which, in turn, may lead to critical loss of accuracy. In order to compensate for this kind of problem we first multiply the Faddeev operator by twice the highest reduced mass of all pair subsystems and perform the coordinate scaling afterwards. The appropriate conversion factor ("coupling") for the energy is reported when the three-body code starts (the exception is the MTV potential and the Bargman potential, that currently should be only employed for three identical particles in dimensionless units). The conversion factors between internal units of length and atomic units for all pairs of Jacobi coordinates are also reported and saved in the output file `scalingFactors.dat` .

Mention, that the results of calculations for He_2 and He_3 can differ substantially from the results reported in literature[5]. This difference comes from a few unit conversion factors that are commonly rounded to 4 significant figures. This "standard" round-off, however, can be critical for the accuracy of the near-threshold systems. In this version of the code we attempt to employ the best known values for all the physical constants.

Running the configurator

To run the configurator issue the command *`java -jar ThreeBodySolverConfigurator.jar`* . A dialog window should appear. There are two areas in this window.

The first one allows users to set the masses of the particles in a.u.m. (for instance, 4.002603 for ^{4}He) and to mark the identical particles. To mark all three particles as identical, one has to check the identical box at the first particle. To mark two of the particles as identical one has to check the identical box at the second particle. If two of the particles are marked as identical, one could also impose the corresponding symmetry by choosing symmetric or antisymmetric state for the appropriate pair. This option is not well tested yet and only the symmetric option is recommended until subsequent releases.

The second area in the lower part of the window allows user to configure the interaction potential, to construct an optimal nonuniform grid which is suitable to reproduce the two-body states for a given potential. To generate the grid choose the potential, choose an appropriate cutoff radius *`Xmax`* and press *`Generate the grid`* button. A new window should appear. First, the program starts showing iterations of the corresponding two-body subsystem s-wave spectrum being calculated using subsequent approximations to the optimal non-uniform grid [4]. After the grid shape is optimized, a convergence test for the two-body spectrum will be generated. Checking the convergence test results may help to estimate the number of point needed to reproduce the two-body spectrum with the desired accuracy. Varying the *`Xmax`* and analyzing the results one also can choose the most reasonable box size for the calculation.

After the grid has been constructed the user may choose the desired number

of grid points for y and z coordinates as well as the cutoff radius in y coordinate.

When the interactions and grids are set for all distinguishable pairs the configuration files can be generated by choosing the corresponding option in the *File* menu. The following files should be generated: `Discretization.conf`, `ID.conf`, `Masses.conf`, `Potentials.conf` and one or more `*.grd` files.

Running the three-body code

The configuration files are used by the three-body code as an input. The only other input that should be provided by user is the energy of the state with respect to the two-body threshold. If the energy is less than zero, the three-body code will attempt to calculate 2 eigenstates of the three-body system closest to the given value. An optional second parameter of the number of eigenvalues to calculate can be provided. If the energy is greater or equal to zero, a scattering code will be launched.

Run the code with the command `java -server -Xmx8192M -jar jPublicThreeBodySolver.jar` *Energy [number of states]*. If `-server` option is not availble for your JRE, it can be omitted. The `-Xmx8192M` option allows the code to allocate 8G of memory. Change this parameter according to the amount of memory available at your machine. The *Energy* parameter (in a.u.) is defined with respect to the two-body threshold, so that the total energy of the system is E_2+*Energy*. If the value is negative, the program attempts to calculate the energies of the bound states closest to the total energy. If the number of desired states is not provided the value of 2 is assumed. If the value is non-negative, the scattering code should start. For zero energy scattering length calculations will be performed. For positive energy the K-matrix will be calculated.

The bound state code will report subsequent iterative estimates for the 2 (or the given number of) eigenenergies closest to the given input energy. The energies are reported with respect to the three-body threshold. As those estimates are based on approximate solution of the Faddeev equations, it is suggested to make several runs with different values of the initial trial energy to account for the error coming from the approximate inversion of the Faddeev operator. It is recommended to choose the tryal energy not too close to the expected exact value, otherwise the approximate solver can become unstable, which may result in accuracy loss and very long computations. Using an estimate which differs from the exact value within 5-10% range can be reccomended. If the spectrum of the system is approximately known, putting 1 as the number of states can reduce the computation time substantially. It is recommended to make the first run using some reasonably small number of grid points (15-20) to make a rough estimate for the spectrum and refine each spectrum point separately on denser grids.

The scattering code will create files `Phase-`n`.DAT` and `Kmatr-`n`.DAT`. File `Kmatr-?.DAT` contains the elements of the K-matrix extracted at different values of y coordinate for the given finale state channel n. It is assumed, that there is only one open channel for each of the three partitionings. For zero initial state

energy the file contains the scattering length instead of the K-matrix. In the future more general functionality including multiple two-body channels for each partitioning will be added. File `Phase-`n`.DAT` contains the phaseshifts for the elastic scattering case.

Note. For certain values of the number of grid points the three-body code can stop with diagnostics "Error: imaginary angular momentum". In that case use smaller or greater number of grid points in z coordinate.

Parallel execution

The last version of the code can run in parrallel on SMP computers. The number of parallel threads is defined automatically as the number of available CPU cores minus one. If there is any need to change this parameter a file `NCPU.conf can be` added manually. This file should contain the maximal number of CPU cores the code attempts to use. If there is not enough memory, the code will switch to sequential memory-saving mode which can be much slower, but makes calculations with denser grids available.

Changes since version 0.12

1. Fixed a bug in cutoff radius in x coordinate. The previous version assumed the value in internal scaled units rather than a.u.
2. Fixed a bug in conversion of the energy units for scattering states.
3. Fixed a bug in identifying the region of free dynamics.
4. Improved trial vector initialization for bound state calculations.
5. Grid generation is being performed in a separate thread, which allows users to see the progress of grid generation and perform other tasks while an optimal grid is being computed.
6. The configurator attempts to read the last saved configuration from the current directory, so if only minor modifications are necessary there is no need to start configuring from scratch.
7. Some minor optimizations of the grid generation algorithm are made.
8. The z grid is modified to provide more stable results for the angular functions so that more rotational states of two-body subsystems can be accounted for.
9. Added an option of Lennard-Jones potential (undocumented).
10. The most computationally demanding parts of the code are paralleled.

11. K-matrix calculation for scattering with rearrangements added.
12. Authomatic switching to sequential mode when memory is insufficient.
13. Changes to this document.

Future versions

In the future we are going to extend the standard collection of potential models. We shall also provide a user mechanism for adding extra potentials to the collection and adding three-body forces. We shall be working on extending the functionality of the code. We expect our colleagues to produce a feedback that would allow us to concentrate our efforts on the most physically important aspects of the few-body calculations. The following features are planned for future versions:

- developing a parallel version of the memory saving mode;
- K-matrix calculation for excitation and deexcitation;
- variational estimates for the bound state energy;
- hide unconventional units from the user;
- variational estimates for the scattering amplitude;
- explicit output of the Faddeev components and the wave function for bound states;
- estimates of mean radius and mean inter-atomic separations for bound states;
- optimization of angular and reactive coordinate grids;
- better output of the bound state solver;
- generating error messages if the configuration is incomplete;
- extending documentation.

Please, let us know about the features that you are missing the most and we shall do our best to implement such features in the future versions.

Distribution

The preliminary release is distributed solely for the purpose of testing the package on the base of collaborative research. The code is distributed as java binary. If any of your colleagues are interested in running the code, they are welcome to participate in the collaborative research. Feel free to give the original archive containing the code and this note to our colleagues after notifying the authors.

Acknowledgments

This project is supported by the NSF grant PHY-0903956.

References

[1] V.A.Roudnev, Ultra-low energy elastic scattering in a system of three He atoms, Chem. Phys. Lett. **367** (2003) 95

[2] V. Roudnev, Localized component method: application to scattering in a system of three atoms, Nucl. Phys. A **737CF**, S292 (2003)

[3] V. A. Roudnev, S. L. Yakovlev, and S. A. Sofianos, Bound State Calculations for Three Atoms Without Explicit Partial Wave Decomposition, Few-Body Systems, **37**, 179-196 (2005)

[4] Vladimir Roudnev and Michael Cavagnero, Computer Physics Communications **182**, 2099-2106 (2011)

[5] Vladimir Roudnev and Michael Cavagnero, J. Phys. B, **45**, 025101 (2012)