Project 2, deadline December 15

Nuclear Forces PHY989

National Superconducting Cyclotron Laboratory and Department of Physics and Astronomy, Michigan State University, East Lansing, USA

Fall semester 2017

Effective field theory

The aim of this project is to build your own Effective field theory (EFT) interaction model that approximates the phase shifts you extracted with the simple interaction model from project 1. We will thus use as input the phase shift analysis from the previous project and use these a our theoretical benchmark. We will follow closely Lepage's article *How to renormalize Schroedinger equation*

We will start by building a *pionless* EFT potential

$$LO = C_0, \tag{1}$$

$$NLO = C_0 + C_2(k'^2 + k^2), \tag{2}$$

where you would have to fit C_0, C_2, etc. to the low-energy phaseshifts from project 1.

They could then make "Lepage plots" to I) see the power-law improvement as you increase the order, and II) see the breakdown scale (which should be of order m$_{p}iorm_{p}i/2).They'dalsoseewhathappenswhenyouvarythecutoff(whichIwouldadvisetheyuseaGaussianorsu$
$Gaussiantoavoidsubtletiesinthe T-matrixcalculationwhentheinteractionhasasharpcutoff).$

You can then repeat the exercise now for a pionfull EFT potential where they explicitly include the OPE and refit the contacts. It would illustrate the same points as above, but now with the breakdown scale being moved higher to of the order of the middle-range meson mass.

$$LO = C_0 + V_{OPE}, \tag{3}$$

$$NLO = C_0 + C_2(k'^2 + k^2) + V_{OPE}. \tag{4}$$

Project 1a): Setting up the integration domain, mesh points and weights.

Project 1b): Adding a potential model.

Project 1c): Handling the principal value problem.

Project 1d): Adding realistic models for the nuclear forces.

Introduction to numerical projects

Here follows a brief recipe and recommendation on how to write a report for each project.

- Give a short description of the nature of the problem and the eventual numerical methods you have used.
- Describe the algorithm you have used and/or developed. Here you may find it convenient to use pseudocoding. In many cases you can describe the algorithm in the program itself.
- Include the source code of your program. Comment your program properly.
- If possible, try to find analytic solutions, or known limits in order to test your program when developing the code.
- Include your results either in figure form or in a table. Remember to label your results. All tables and figures should have relevant captions and labels on the axes.
- Try to evaluate the reliabilty and numerical stability/precision of your results. If possible, include a qualitative and/or quantitative discussion of the numerical stability, eventual loss of precision etc.
- Try to give an interpretation of you results in your answers to the problems.
- Critique: if possible include your comments and reflections about the exercise, whether you felt you learnt something, ideas for improvements and other thoughts you've made when solving the exercise. We wish to keep this course at the interactive level and your comments can help us improve it.
- Try to establish a practice where you log your work at the computerlab. You may find such a logbook very handy at later stages in your work, especially when you don't properly remember what a previous test version of your program did. Here you could also record the time spent on solving the exercise, various algorithms you may have tested or other topics which you feel worthy of mentioning.

Format for electronic delivery of report and programs

The preferred format for the report is a PDF file. You can also use DOC or postscript formats or as an ipython notebook file. As programming language we prefer that you choose between C/C++, Fortran2008 or Python. The following prescription should be followed when preparing the report:

- Use your github repository to upload your report. Indicate where the report is by creating for example a **Report** folder. Please send us as soon as possible your github username.
- Place your programs in a folder called for example **Programs** or **src**, in order to indicate where your programs are. You can use a README file to tell us how your github folders are organized.
- In your git repository, please include a folder which contains selected results. These can be in the form of output from your code for a selected set of runs and input parameters.
- In this and all later projects, you should include tests (for example unit tests) of your code(s).
- Comments from us on your projects, with score and detailed feedback will be emailed to you.

Finally, we encourage you to work two and two together. Optimal working groups consist of 2-3 students. You can then hand in a common report.