Implementing Quasielastic Spectral Function Model Into GENIE via Wrapper Infrastructure

Syrian V. Truong[1, 2]

[1] *Fermi National Accelerator Laboratory, Batavia, Illinois 60510, USA*
[2] *Department of Physics, University of Chicago, Illinois 60637, USA*

While GENIE is a robust and ubiquitous C++ based software framework, containing a Monte Carlo event generator utilized for neutrino scattering simulation and analysis, some aspects of its neutrino scattering models can be improved. Particularly, the quasielastic neutrino scattering models are known to have significant disagreement between GENIE-based simulation models and experimental data. However, there exists external code written in Fortran 90, outside of the GENIE framework, which offers better agreement between simulation and experimental data, but lacks many of the useful features already present within the GENIE framework for other aspects of the neutrino scattering simulations. As such, one solution to utilize the strength of both GENIE and this external code is to implement a wrapper-based infrastructure which will utilize the improved quasielastic model present within the Fortran 90 code, within GENIE's overall framework. Promising comparison results between the experimental data, external code simulation data, and wrapper-based infrastructure simulation data are shown in the results section. This wrapper-based infrastructure addition result success for GENIE motivates the possibility for further similar wrapper additions to improve GENIE's other reaction mechanism models, while still utilizing its overall framework.

I. INTRODUCTION

Fermilab is at the forefront of experimental research into the properties of neutrinos. With a plethora of data coming in from these experiments, much analysis through comparisons of simulated models for neutrino interactions is needed. Improved understanding of neutrino properties though this, could lead to discovering fundamentally new physics, since experimental results from neutrino interactions have indicated conflict with certain previously well established physical concepts. While Fermilab utilizes GENIE, a large software framework with a basis in Monte Carlo event generation of neutrino interactions for much of their neutrino simulation based research, improvements to this framework are still needed. [1] In particular, some of the neutrino scattering reaction mechanisms have not demonstrated a satisfactory agreement between experiment and simulation model results.

For this paper, we focus upon improving GENIE's quasielastic neutrino scattering reaction mechanism, through use of a wrapper infrastructure and external code with an improved quasielastic electron scattering reaction mechanism model, which extrapolates to an improved neutrino based version. From these results, we note the possibility of future work utilizing this wrapper infrastructure additions to improve other models in GENIE, such as the meson exchange current, resonance, and deep inelastic reaction mechanisms.

II. IMPROVEMENTS TO THE MODEL

Improvements to the quasielastic scattering models exist via Dr. Noemi Rocco's spectral function based model. [2] This model utilizes a combinaton of realistic initial target states with a fully relativistic interaction vertex and kinematics. [3][4] Further, it utilizes the fact that through extrapolated similarities between electrons and neutrinos, being that they interact similarly and have many identical nuclear effects, one can obtain information about neutrino quasielastic scattering models through solely looking at electron quasielastic scattering models. [5] Thus, this simulation focuses upon electron quasielastic scattering models, as by doing so we are able to utilize the plethora of information already known for electron scattering mechanisms. This bypasses some of our information based limitations regarding neutrino scattering mechanisms, while still allowing us to compare neutrino quasielastic scattering simulation models with their experimental counterparts, after proper conversions are made.

Example comparisons of differential cross sections as a function of electron energy transfer between the spectral function model and experimental data shown in Figure 1 provide good motivation for incorporation of this model into GENIE, demonstrating a rough visual agreement between the simulation and experimental data results (see [2] for more information on the accuracy of this model). Note, the quasielastic reaction mechanism contributions to the data dominate within low values of the electron energy transfer, but drop off during higher values where other reaction mechanism contributions begin to dominate. Thus, we are only concerned about the agreement between simulation and experimental data within low values of the electron energy transfer. With that said, due to the fact that the spectral function model calculations for these quasielastic scattering reaction mechanisms are written in Fortran 90, while GENIE is written in C++, this presents an immediate obstacle for utilizing both in tandem.

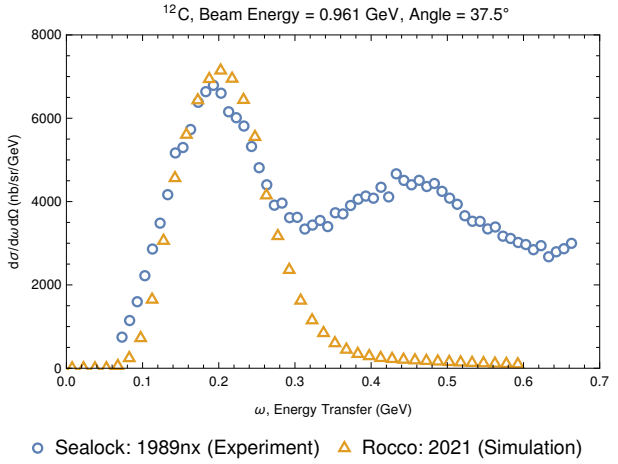

FIG. 1. Experimental (blue) and simulation (orange) data of electron scattering off of Carbon-12 nucleus with 0.961 GeV beam energy and 37.5 degree electron scattering angle. (Credit: Sealock 1989nx, Rocco 2021)

III. WRAPPER IMPLEMENTATION

There are more than a couple options for circumventing the difference in programming languages for both codes, when attempting to utilize the strengths of both GENIE and the spectral function model code. For example, one could attempt to rewrite the spectral function model directly into GENIE's C++ code. However, we instead choose to implement a wrapper infrastructure, by wrapping the spectral function model into GENIE. We do this in order to attempt to preserve as many advantages which come from the original Fortran 90 code execution, which might be lost during translation from one programming language to another. Further, it presents the possibility of a speedier process which might be less prone to human translation error, with the option of less invasive user customization via utilizing other external code which may benefit GENIE's other models in the future.

Generally, we removed GENIE's current quasielastic model related C++ code lines and replaced them with C++ code lines which call part of the slightly edited Fortran 90 spectral function model code. This is our wrapper, which executes only portions of Dr. Rocco's code that are necessary to simulate the electron quasielastic scattering model, slightly edited to fit the context of GENIE's difference in Monte Carlo based calculation procedure. Aside from this, one needs to pay attention to proper compilation procedures (including all necessary code files), generation of inputs from GENIE, conversion units between GENIE and the spectral function model, wrapper function-subroutine enabling lines, and resulting outputs from the wrapper code.

Thus, our wrapper infrastructure allows us to begin with similar GENIE initiation, generating the necessary inputs through GENIE for input into the spectral function model code, then receiving the desired differential cross sectional results back out from it, then finishing up calculations and output generation within GENIE.

IV. RESULTS

The wrapper-based infrastructure was a preliminary success, with rough agreement (with a few caveats and subject to further statistical analysis), as shown in Figure 2.

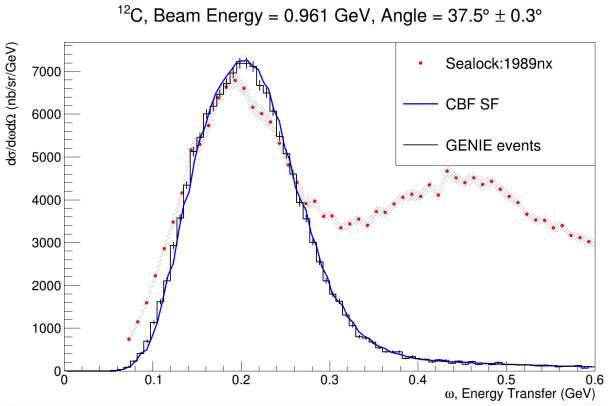

FIG. 2. Experimental (red), spectral function Fortran 90 only (blue), GENIE + wrapper (black) data of electron scattering off of Carbon-12 nucleus with 0.961 GeV beam energy and 37.5 degree electron scattering angle. (Credit: Gardiner and Truong 2021)

As such, this wrapper-based infrastructure addition result success for GENIE motivates the possibility for further similar wrapper additions to improve GENIE's other models (such as the meson exchange current, resonance, and deep inelastic reaction mechanisms), while still utilizing its overall framework.

ACKNOWLEDGMENTS

Huge thank you to Dr. Minerba Betancourt and Dr. Steven Gardiner for being both a supportive and understanding supervisor and mentor, respectively. This project was immensely helpful for further deciphering what my desired research direction would be for my dissertation research in my PhD program, along with bolstering my C++, Fortran, and general computational physics based skill set, in a large part thanks to their guidance. Further, I would like to thank Michael Geelhoed and the rest of Intern Group 3, for helping to make this virtual internship feel a little less isolating. Lastly, I would like to thank the Fermi National Accelerator Laboratory, the Graduate Education for Minorities (GEM) Fellowship Program, and the University of Chicago, without whom this internship opportunity would not have

been possible.

[1] C. Andreopoulos *et al.*, "The GENIE Neutrino Monte Carlo Generator," Nucl. Instrum. Meth. A **614**, 87–104 (2010), arXiv:0905.2517 [hep-ph].

[2] Noemi Rocco, Carlo Barbieri, Omar Benhar, Arturo De Pace, and Alessandro Lovato, "Neutrino-nucleus cross section within the extended factorization scheme," Phys. Rev. C **99**, 025502 (2019).

[3] Artur M. Ankowski and Jan T. Sobczyk, "Construction of spectral functions for medium-mass nuclei," Phys. Rev. C **77**, 044311 (2008).

[4] R. Bradford, A. Bodek, H. Budd, and J. Arrington, "A new parameterization of the nucleon elastic form factors," Nuclear Physics B - Proceedings Supplements **159**, 127–132 (2006), proceedings of the 4th International Workshop on Neutrino-Nucleus Interactions in the Few-GeV Region.

[5] A. Papadopoulou, A. Ashkenazi, S. Gardiner, M. Betancourt, S. Dytman, L. B. Weinstein, E. Piasetzky, F. Hauenstein, M. Khachatryan, S. Dolan, G. D. Megias, and O. Hen ($e4\nu$ Collaboration), "Inclusive electron scattering and the genie neutrino event generator," Phys. Rev. D **103**, 113003 (2021).

Appendix A: Wrapper Code

In essence, the wrapper infrastructure specific code only necessitates a few components. These components are contained within either your compilation procedure line or a makefile, within your established class, and possibly within any bridge code which may be necessary to have your external code work as intended. Some examples from our context are shown below.

For the makefile, our compilation lines required us to compile multiple C++ and Fortran 90 code files, where proper compilation options are important and specific to each programming language, as demonstrated by this code excerpt below:

```
# ...

test_xsec: test_xsec.o FortranWrapperQELPXSec.o
root_dictionary.o currents_opt_v1.o
xsec_fact_new.o diracmatrices.o nform.o
        $(CXX) $(ROOT_LIBS) $(GENIE_LIBS) -o $@
        $^ -lgfortran

%.o: %.cxx
        $(CXX) $(ROOT_CXX_FLAGS)
        $(GENIE_CXX_FLAGS) -o $@ -c $^

%.o: %.f90
        gfortran -o $@ -c $^

# ...
```

For the class, we required the following lines to partly enable the ability to call the specific Fortran 90 subroutines from this C++ class:

```
// ...

extern"C"
{
void diracmatrices_(double *xmn_in);
}

extern"C"
{
void cc1_(double *xq, double *w, double *wt,
double *xk, double *xp, double *ee0,
double *theta, int *ig, double *xsec,
double *nuphi);
}

// ...
```

which after obtaining the necessary proper input variables and inputting them, we would be able to call the specific Fortran 90 subroutines via the following lines:

```
// ...

diracmatrices_(&xmn_in);

cc1_(&xq, &w, &wt, &xk, &xp, &ee0, &theta, &ig,
&xsec, &nuphi);

// ...
```

For the bridge code, which in this context was necessary, we created the following Fortran 90 code file containing a subroutine structure for the purpose of properly executing a necessary precursor procedure and subroutine, in order for the main differential cross section calculating subroutine to work as intended:

```
subroutine diracmatrices(xmn_in)
        use dirac_matrices
        IMPLICIT NONE
        real*8 :: xmn_in

        call dirac_matrices_in(xmn_in)

        return

end subroutine diracmatrices
```