Fermilab U.S. DEPARTMENT OF ENERGY Office of Science

Extended Factorization scheme and a Fortran Interface for GENIE

Noah Steinberg
Joint Theory and Experiment Meeting
April 27th 2022

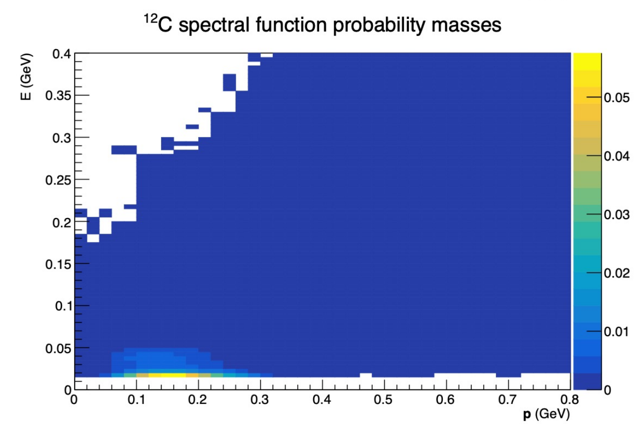
Motivation: Theory API + SF

- Extend GENIE's capabilities of incorporating new cross section models by developing a Fortran wrapper for computing hadronic response tensors
  - Move beyond precomputed tables of responses/tensor elements
  - Gives user ability to compute Hadronic response tensor on the fly
  - Utilize existing theory code already written in Fortran

- Combine state of the art Spectral Functions with an Extended factorization scheme to compute lepton-nucleus cross sections in GENIE

$$J_A^\mu \longrightarrow \sum_i j_i^\mu \,, \qquad |X\rangle \longrightarrow |x, \mathbf{p}_x\rangle \otimes |R, \mathbf{p}_R\rangle$$

$$d\sigma_A = \int dE d^3k \; d\sigma_N \; P(k, E)$$



12C spectral function probability masses in GENIE, implemented by Steven Gardiner

Details: Fortran Interface

- Developing a generalized Fortran interface for differential cross section calculations in GENIE
  - GENIE deals with phase space sampling and setting up kinematics for each event

  - Necessary kinematic information passed to Fortran
    - (Four-vectors of initial and final state particles
    - Information from nuclear model
    - Nucleon Form factors

  - GENIE calls Fortran code (not the other way around)
    - May pass redundant information, but too much is better than too little

  - Make it general enough that any theorist can plug in their Fortran code and it works out of the box

  - GENIE side of calculation is completely configurable via xml files as usual
    - Can use xml files to turn Fortran knobs as well

Implementation – Event Generation

- Developed currently for 1p1h QE interactions: EM, CC, and NC
  - FortanWrapperQELPXSec.cxx
  - FortranWrapperEventGenerator.cxx
  - FortranWrapperXSecIntegrator.cxx

- New in Cross section calculation

  - Theme (Use as much of GENIE as possible)
    - Leptonic Tensor
      - Class written by S. Gardiner (computes elements of 4x4 leptonic tensor)
        - Implemented for electrons and CC/NC neutrinos

    - Nucleon form factors
      - Utilize these, configurable via. XML file

  - Universal class for EM, CC, NC
    - Inspired by work Steven Gardiner has already done
    - Only thing that changes is the couplings and pre-factors for the cross section
      - G_F vs. α_{EM}

  - Utilize existing Fortran code to compute the nuclear side of the cross section

Implementation – Cross Section calculation

- New in Event Generation
  - FortranWrapperEventGenerator uses accept/reject to sample values of initial struck nucleon 4-momentum as well as 4-momentum of outgoing lepton **and** nucleon
    - Initial nucleon is sampled from Spectral function
    - Cross section is computed as $d\sigma/d^4k d^4p$, a universal form of the cross section so that the cross section in any other observable can be computed from this
      - kPSFullNBody phase space
      - Doesn't require a specific set of phase space variables which require a custom solution to sample
      - LHC-style generators use this – well established techniques to sample full 4-vectors
        - See [HepPh 2110.15319] J. Isaacson, et al.

Implementation – Cross Section calculation

- Brief theory overview of PWIA cross section
  - Start with inclusive differential cross section

$$d\sigma = \frac{\alpha^2 E_{k'}}{Q^4 E_k} R^{\mu\nu} L_{\mu\nu} d\Omega_{k'} dE_{k'}$$

  - Nuclear response tensor as a convolution of single nucleon matrix elements and spectral function

$$R^{\mu\nu}(p,q) = \int d^3p dE P(p,E) \frac{m_N^2}{E_p E_{p+q}} \langle p|j^{\mu\dagger}|p+q\rangle\langle p+q|j^\nu|p\rangle \delta(\tilde{\omega} - E + m_N - E(\vec{p}+\vec{q}))$$

  - Define a nucleon level response tensor

$$\tilde{R}^{\mu\nu}(p,q) = \frac{m_N^2}{E_p E_{p+q}} \langle p|j^{\mu\dagger}|p+q\rangle\langle p+q|j^\nu|p\rangle$$

- In the current set up, GENIE computes the leptonic part of the cross section as well as does the integral over the spectral function

- Fortran module simply computes the nucleon level response tensor

Implementation – Cross Section calculation

- Example: EMQE scattering
  - Start with inclusive differential cross section

$$\frac{d^2\sigma}{dE'd\Omega'} = \frac{\alpha^2}{Q^4}\frac{E'}{E}L_{\mu\nu}R^{\mu\nu}$$

  - Pass kinematic information to Fortran module to compute Hadronic response tensor

```
//____________________________________________________________________________
extern"C"
{
void compute_hadron_tensor_(double *xq, double *w, double *wt, double *xk, double *xp,
   std::complex<double> resp[4][4]/*response tensor*/, double *nuphi, double *f1v,
  double *f2v, double *ffa);
}
```

FortranWrapperQELPXsec.cxx
- Set's up kinematic factors for cross section for specific process
- Computes form factors relevant to process (EMQE, CCQE, NCQE)
- Passes set of kinematic information to Fortran Module

xsec_fact.f90
- loads kinematics and form factors
- Sets up four vectors for initial and final state nucleons

currents_opt_v1.f90
- Sets up dirac spinors for initial and final nucleons
- Computes current Γ^μ and matrix element J^μ
- Computes Response tensor (explicitly summing over spins)

FortanWrapperQELPXSec.cxx
- Loads response tensor from Fortran into derived class from Rank2LorentzTensor
- Contracts with Leptonic tensor from GENIE
- Returns $d\sigma/d^4k d^4p$, differential in outgoing momenta of final state particles

Fermilab

Implementation – Cross Section calculation

- Xsec_fact.f90
  - Implements compute_hadron_tensor() as a fortran subroutine
  - Loads form factors and groups nucleon kinematics into 4-vectors
  - Passes these to another module, currents_opt_v1.f90
- Currents_opt_v1.f90
  - Computes four spinors for initial and final nucleons, in both spin states
  - Computes nucleon current and matrix element
  - Finally, computes nucleon level response tensor

$$u^{(s)}(\vec{p}) = \frac{\not{p} + m}{\sqrt{2m(E+m)}} u^{(s)}(\vec{0}) = \sqrt{\frac{E+m}{2m}} \begin{bmatrix} \phi^{(s)} \\ \frac{\vec{\sigma}\cdot\vec{p}}{E+m}\phi^{(s)} \end{bmatrix}$$

FortranWrapperQELPXsec.cxx
- Set's up kinematic factors for cross section for specific process
- Computes form factors relevant to process (EMQE, CCQE, NCQE)
- Passes set of kinematic information to Fortran Module

→

xsec_fact.f90
- loads kinematics and form factors
- Sets up four vectors for initial and final state nucleons

→

currents_opt_v1.f90
- Sets up dirac spinors for initial and final nucleons
- Computes current Γ^{μ} and matrix element J^{μ}
- Computes Response tensor (explicitly summing over spins)

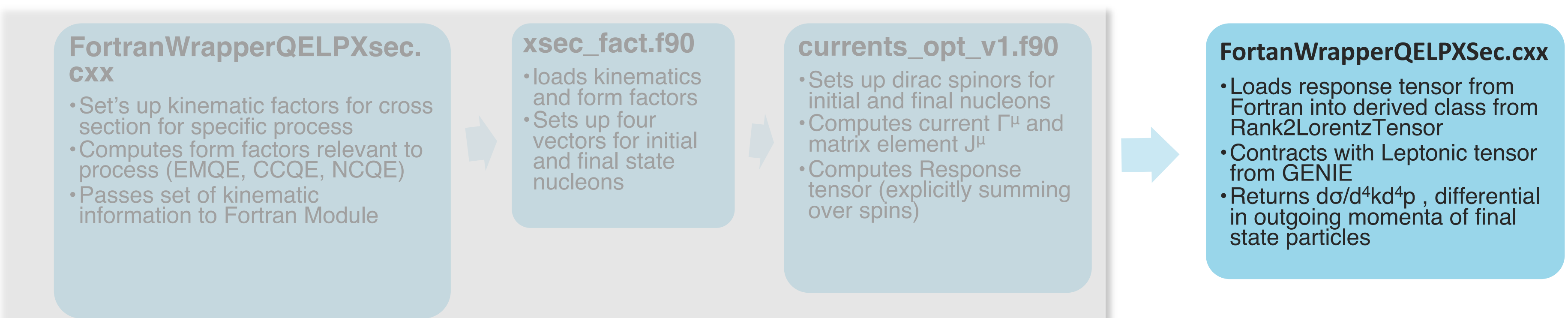
→

FortanWrapperQELPXSec.cxx
- Loads response tensor from Fortran into derived class from Rank2LorentzTensor
- Contracts with Leptonic tensor from GENIE
- Returns $d\sigma/d^4k d^4p$, differential in outgoing momenta of final state particles

Implementation – Cross Section calculation

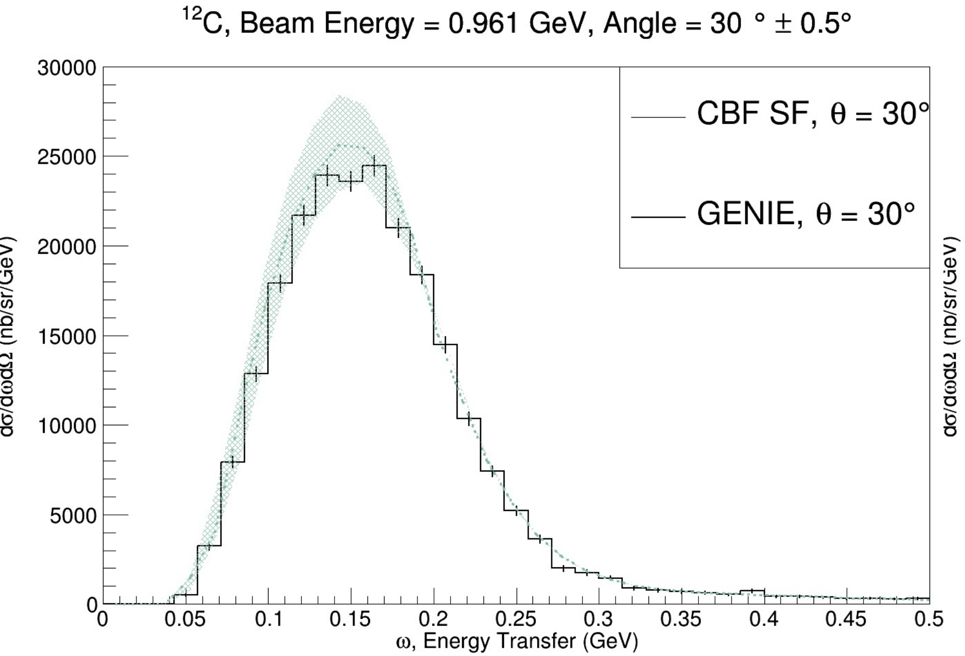
- Nucleon Response tensor is passed back to GENIE as a 4x4 array std::complex<double>
  - *Nota bene:* Fortran stores arrays in column ordering, so must transpose array before passing back to GENIE

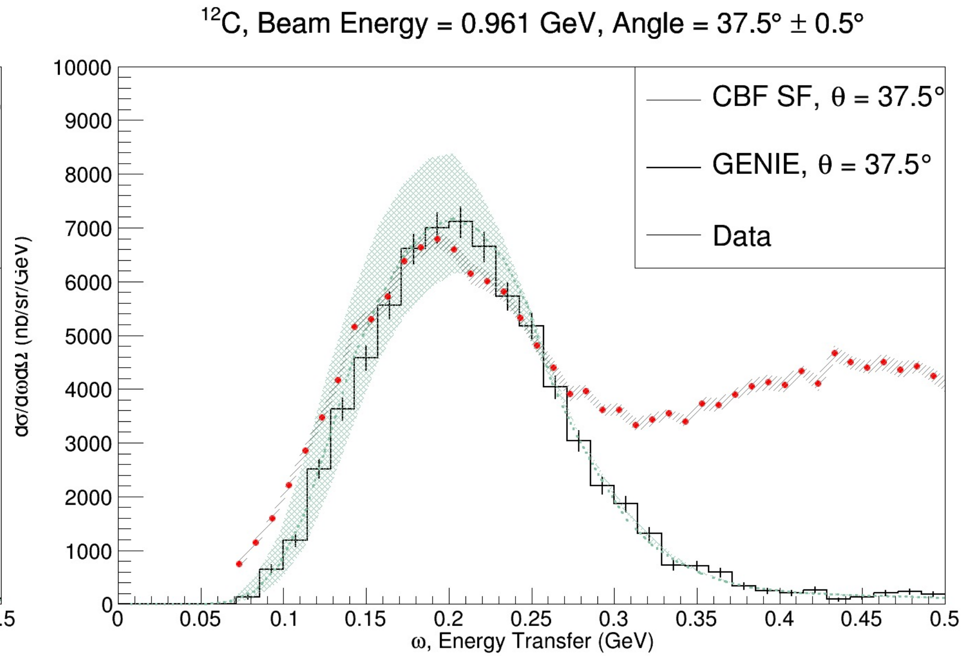
- Leptonic and nucleon level response tensor are contracted
  - This returns a cross section differential in lepton kinematics
  - In 1p1h case, transformation is trivial
    - Reintroduce d^3p and momentum conserving delta function



Testing it Out

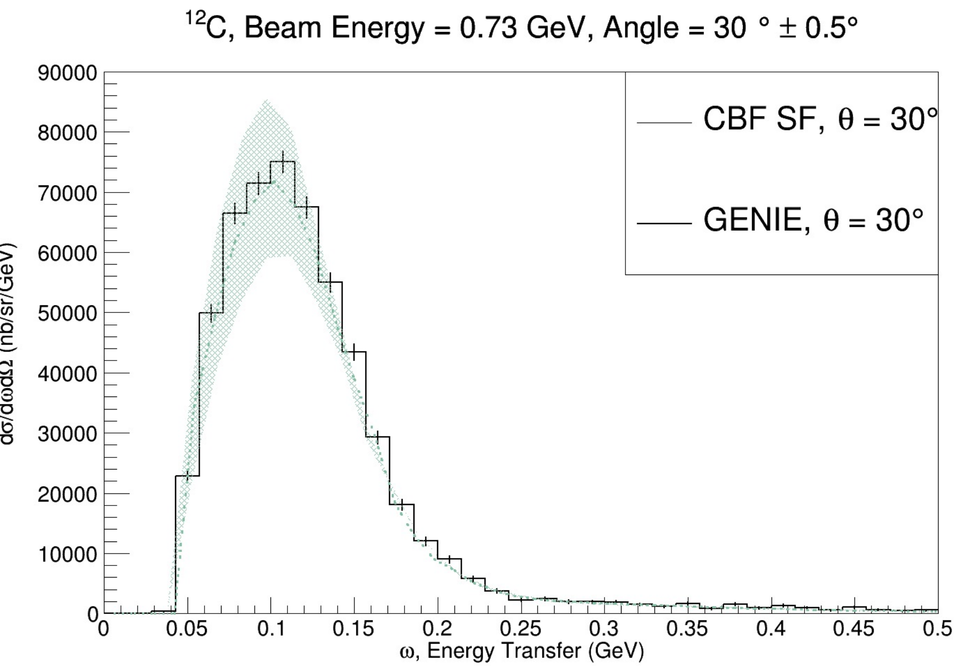
- Dipole Form Factor (Don't expect excellent agreement)
- 961 MeV
  - 3M events

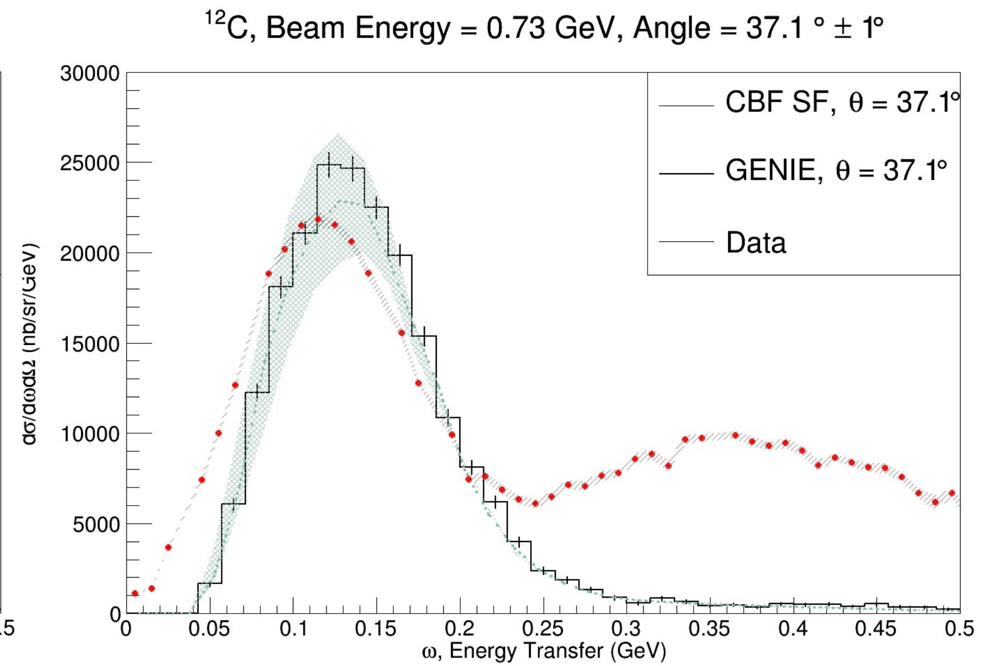




Testing it Out

- 730 MeV
  - 1M events





Fermilab

Validating now: Neutrinos

- Same cross section code, difference in kinematic pre-factors as well as coupling constants, leptonic tensor, etc.
  - Test on vector parts of CC interactions
  - Introduce axial couplings
    - Form factors fully configurable
  - Test NC
  - Total cross sections

More General Highlights

- This implementation
  - Extended factorization scheme can include MEC, resonance production, and DIS
    - Plan to implement MEC after electron and neutrino 1p1h is validated
  - Doesn't use precomputed tables of hadronic response tensors (contrast with SuSAv2)
  - Isn't limited to a particular nuclear model
    - Can use any of GENIE's built in nuclear models
  - Can be connected to any theory model of nuclear response tensor via Fortran
    - Many theorists use Fortran for their calculations anyways