Implementation od MK Single Pion Production Model in GENIE

Igor Kakorin, Kostantin Kuzmin and Vadim Naumov [1]

June 18, 2020

[1] Joint Institute for Nuclear Research, Dubna, Russia

XSecAlgorithmI → *MKSPPP*

- The *XSecAlgoritmI* implemented in the class *MKSPPP* based on C code provided by Minoo: for each separate channel there is a separate file.
- Now all channels are calculated in the single class.
- The class *MKSPPP* calculates differential cross section $\frac{d\sigma}{dQ^2\ dW\ d\cos\theta_\pi\ d\varphi_\pi}$ and $\frac{d\sigma}{dQ^2\ dW\ d\cos\theta_\pi}$. The second one is obtained by analytical integration of the first one and it helps to speed up the numerical integration of $\frac{d\sigma}{dQ^2\ dW\ d\cos\theta_\pi\ d\varphi_\pi}$ when calculating total cross-section.

- To specify required cross section we use the second argument, *kps* (kinematic phase space), of the method *XSecAlgorithmI::XSec(const Interaction* i, KinePhaseSpace_t kps)*: kps=kPSWQ2ctpphipfE for $\frac{d\sigma}{dQ^2\ dW\ d\cos\theta_\pi\ d\varphi_\pi}$ and kps=kPSWQ2ctpfE for $\frac{d\sigma}{dQ^2\ dW\ d\cos\theta_\pi}$. The advantage of this approach is that no double code required for calculation of essentially similar things[2].
- If the kinematic phase space differs from kPSWQ2ctpphipfE or kPSWQ2ctpfE one needs know its dimension to calculate the required cross section: $D = 4$ for $\frac{d\sigma}{dQ^2\ dW\ d\cos\theta_\pi\ d\varphi_\pi}$ and $D = 3$ for $\frac{d\sigma}{dQ^2\ dW\ d\cos\theta_\pi}$.

[2]The MK-model is not an exception, the situations are possible when the cross section for some cases can/must be calculated using slightly different formulas.

- To find the dimension of kinematic phase space, we calculate the number of commas in its string representation, for example, the dimension of kine phase space which has string representation <{W, Q2, cos(theta_pion|E)}>
equal to $\mathbf{3} = \mathbf{2}$ (number of commas) $+ \mathbf{1}$.
- **Our suggestion**: it would be much better to implement an additional method in the class *KinePhaseSpace*, which would return the dimension of a given kinematic phase space.
- What concerns physics: we faced a few problems and some questions arose when implementing the model. We set out them in detail in the accompanying documents "Report on the implementation of the MK-model for resonance single-pion production into GENIE". We remark all problematic points in the code with reference to that Report.

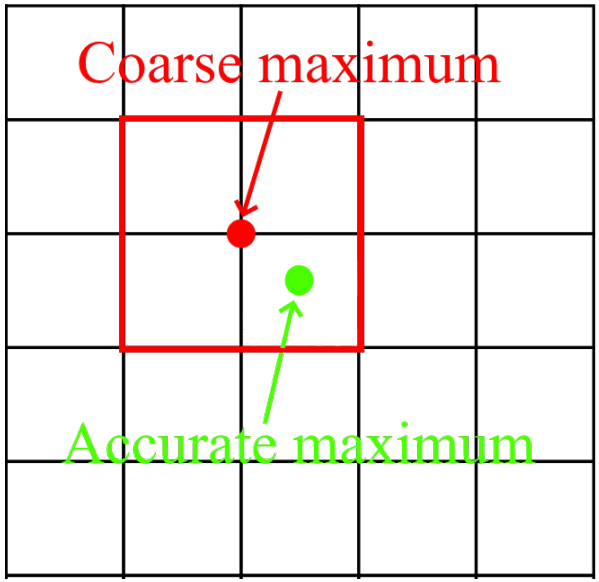
XSecIntegratorI → *MKSPPXSecWithCache* → *MKSPPXSec*

- The *XSecIntegratorI* implemented in the class *MKSPPXSecWithCache*, which is inherited by the class *MKSPPXSec*.
- The method *MKSPPP::XSec* for calculation $\frac{d\sigma}{dQ^2\ dW\ d\cos\theta_\pi}$ is called in the integrand.
- Numerical integration is carried out in tree variables. By default, we utilize the adaptive multi-dimensional integration using the Genz–Malik algorithm.

- The *EventRecordVisitorI* implemented in the class *RSPPEventGenerator*.
- The kinematics of pion, final nucleon, and lepton is selected.
- Isospin symmetry is implied.
- The main problem is maximum search for accept-rejection method. We have implemented three approaches, each has some advantages and disadvantages.
- **Straightforward method** can be found on branch `https://github.com/igor144/Generator/tree/MK_SPP_model`
- **Heuristic method based on MINUIT** can be found on branch `https://github.com/igor144/Generator/tree/MK_SPP_model_alt_max_search`
- **Method used FOAM algorithm** can be found on branch `https://github.com/igor144/Generator/tree/MK_SPP_model_foam`

Maximum search: Straightforward method

- At first we do brute force search of a "coarse" maximum among $N_1 \times N_2 \times N_3 \times N_4$ knots.
- Then we call MINUIT for delicate search of maximum in adjacent cells. The starting value for MINUIT is "coarse" maximum.



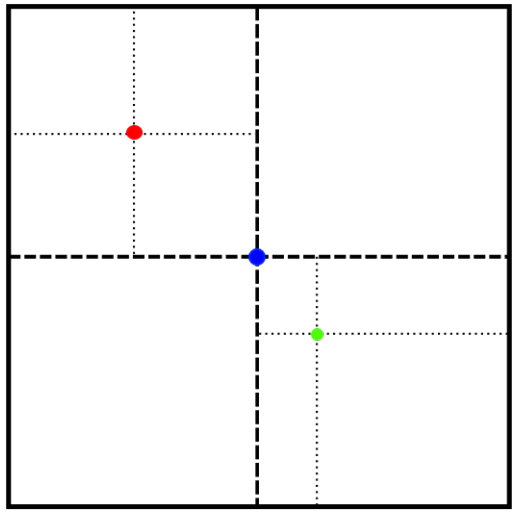
- Disadvantages of this approach:
  - If N_i are very small then the algorithm cannot find the global maximum, only local.
  - If N_i becomes big enough, it works very-very slow.

  We tested it for $N_i = 40$ and it takes about 250 s to find a maximum, but it was not the global one in about 10% cases.

Maximum search: Heuristic method

- We seek the global maximum by MINUIT. To choose proper starting value and limits on variables we use two approach.
- The first is for low-energy neutrino $E_\nu < 15$ GeV.
- The invariant mass W goes over all resonance masses, M_R, the maximum is sought in the limits $(M_R - \Gamma_R, M_R + \Gamma_R)$, where Γ_R is the width of given resonance.
- The region of φ_π-variation is divided into 4 parts $(0, \pi/2)$, $(\pi/2, \pi)$, $(\pi, 3\pi/2)$ and $(3\pi/2, 2\pi)$; the starting values are chosen equal to $0, \pi/2, \pi, 2\pi$.
- The region of $\cos\theta_\pi$-variation is divided into 2 parts $(-1, 0)$, $(0, 1)$; the starting values are chosen equal to ± 1.
- The region of Q^2-variation are (Q^2_{min}, Q^2_{max}) for E_ν <1 GeV with starting $(Q^2_{min} + Q^2_{max})/2$ and $(Q^2_{min}, 2Q^2_{min}/3 + Q^2_{max}/3)$ for 1 GeV< E_ν <15 GeV with starting value equal to $5Q^2_{min}/6 + Q^2_{max}/6$.

Maximum search: Heuristic method



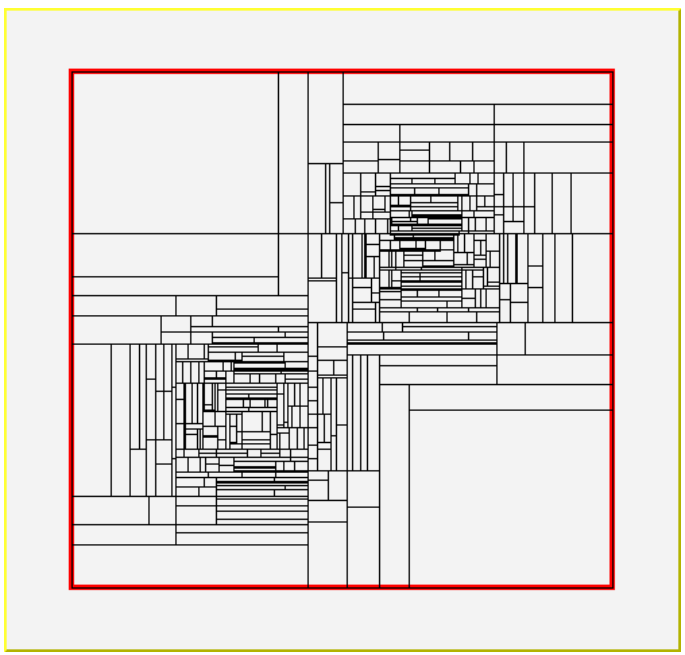
- The second method is for high-energy neutrino $E_\nu > 15$ GeV. We apply transformation: $x_1 = (W - W_{min})/(W_{max} - W_{min})$, etc. So the full kinematic space is represented by 4d-cube with vertex $(0, 0, 0, 0)$ and $(1, 1, 1, 1)$
- At first, we seek a maximum in full kinematic phase space with starting value at the centre of the 4d-cube: $(1/2, 1/2, 1/2, 1/2)$.

Maximum search: Heuristic method

- After the maximum is founded we divide the phase space into 16 4d-cubes, all them have founded maximum as a corner (•).
- Then we seek maximums in each of 16 4d-cube, again with starting value at the centre of 4d-cube.
- If founded maximum in one of sixths 4d-cube is distant from one from previous step, by a given value in all coordinates (•) then we set this maximum as dividing point, if not we set it as $(1/2, 1/2, 1/2, 1/2)$ (•), again divide this 4d-cube as in previous step and repeat maximum search in new cubes.
- Our tests show that heuristic method works faster in about 10 times and always find the global maximum.
- Disadvantage of this approach is in that it is not quite clear will it work properly in all cases, but if the number of events to generate is quite large then the dependence of the global maximum on energy will be interpolated eventually and the time of single event generation will decrease.

Maximum search: FOAM algorithm

- `https://root.cern/root/html524/TFoam.html`
- Official documentation: FOAM is the simplified version of the multi-dimensional general purpose Monte Carlo event generator. It creates hyper-rectangular "foam of cells", which is more dense around its peaks. See the 2-dim. example of the map of 1000 cells for doubly peaked distribution.



Maximum search: FOAM algorithm

- Implemented in ROOT.
- Disadvantage of this approach is in that FOAM need some time to build "foam" of cells, but then it can be saved on disk and used repetitive.
- Another disadvantage is in that the "foam" depends on probability distribution function, which in its turn depends on parameter – neutrino energy. So we build the "foam" for some reference energies interactively.
- We define the step between reference energies and if neutrino energy is close to existing one we load foam from disk and select kinematics according to closest by energy "foam" and set the weight of selected event
$w = \frac{d\sigma}{dQ^2\ dW\ d\cos\theta_\pi\ d\varphi_\pi}(E) / \frac{d\sigma}{dQ^2\ dW\ d\cos\theta_\pi\ d\varphi_\pi}(E_{ref})$
- In other case we build the "foam" for new reference energy and save it on disk.

Maximum search: FOAM algorithm

- The time to build "foam" of cells for given neutrino energy is about 30 s.
- Maybe the "foam" should be pre-calculated like XSec-splines? If yes, than we need to understand how to change the main framework for this.
- If the number of events to generate is quite large, then the generator eventually build all the needed "foams" and then the time of event generation will be defined by read/write operations from disk and time of work FOAM algorithm, which is quite small.
- It is advisable to choose the step energies less than granularity of flux.
- The weights of tested events as usual lie between 1 and 2.

Maximum search: any other ideas?

- Which method (among suggested) to find the global maximum should we choose?
- Seems like we have exhausted all our ideas. Maybe someone can offer something smarter?