fsiv.py documentation

Tomasz Golan

September 24, 2014

Contents

1 About

fsiv.py is the script to validate GENIE's FSI models. It uses runfast.pl to run required simulations and intranukeplotter.pl to prepare plots. Both perl scripts are assumed to be at the same directory[1].

fsiv.py gives the possibility to make a validation with the set of all available data or with selected subset of them. The precision can be modified by changing number of events for runfast.pl script. The script can be parametrized so runfast.pl and intranukeplotter.pl will be run on a grid.

2 How does it work

Preparing the list of available authors

At the beginning the script reads all *.dat files from $FSIV_DAT_DIR folder. If the environmental variable $FSIV_DAT_DIR is not defined by the user "pwd/data_files" is assumed. The list of objects of class Author is created. Each object stores informations about available process for given author (incident particle, type of cross section). Those informations are extract from the filename assuming the following convention:

- author-particle-target-fate.dat (for total cross section data)
- author-particle-*.dat (for differential cross section data)

For total cross section data it always keeps informations about available fates as well as a list of energies and maximum cross section for each fate. Those informations are extracted from the file with data.

[1] should we changed that?

Preparing the list of simulations for total cross section

In the next step the list of process for total cross section simulations (TotalProcess) is created. The script goes through the list of authors and creates new entries (or updates existing ones) for each channel. Each TotalProcess object contains informations about the incident particle, the target, the list of fates and for each fate: the list of kinetic energies and the maximum cross section.

This way it is possible to run one simulation for several authors if they have data for the same channel.

For some channels data are highly concentrated around several kinetic energy values, e.g.

37.0, 52.0, 68.0, 83.0, 100.0, 125.0, 151.0, 187.0, 233.0, 280.0, 584.0, 711.0, 870.0, 1213.0, 1446.0, 1860.0 [MeV]

In order to find the proper energy values for simulations, the script follows the following steps:

1. round to integer and skip the values less than 10 MeV:

   37, 52, 68, 83, 100, 125, 151, 187, 233, 280, 584, 711, 870, 1213, 1446, 1860

2. take all point for which difference is less than 25% and take average of them instead all of them (but leave first and last points):

   37, 52, 75, 100, 138, 210, 280, 647, 870, 1329, 1860

3. as long as the difference between some adjacent points is bigger than 200% add the point in the middle between them:

   37, 52, 75, 100, 138, 210, 280, 463, 647, 870, 1329, 1860

Preparing and running bash scripts

The script goes through the list of Authors (or TotalProcess) and for each entry it creates a bash script which then is run or passed to a grid. The bash scripts are stored in $FSIV_BSH_DIR[2].

Each bash script contains proper runfast and intranukeplotter commands. Before it runs intranukeplotter it looks for the most recent root file (or txt for total) for given process, so the proper date can be passed to intranukeplotter.

The result of simulations are stored in $FSIV_SIM_DIR[3]. The plots are saved in $FSIV_PNG_DIR[4]. The outputs of runfast and intranukeplotter are saved in $FSIV_LOG_DIR[5]. The general fsiv_[date].log file contains the summary of the validation job.

3 Using the script

Environmental variables

The script uses the following environmental variables:

[2] or 'pwd/bash_scripts' if $FSIV_BSH_DIR is not defined

[3] or 'pwd/sim_files' if $FSIV_SIM_DIR is not defined

[4] or 'pwd/png_files' if $FSIV_PNG_DIR is not defined

[5] or 'pwd/log_files' if $FSIV_LOG_DIR is not defined

- $FSIV_DAT_DIR - folder with data files
- $FSIV_SIM_DIR - folder for results of simulations
- $FSIV_PNG_DIR - folder for final plots
- $FSIV_BSH_DIR - folder for bash scripts
- $FSIV_LOG_DIR - folder for log files

Note, all environmental variables required by GENIE must be defined.

List of options

-h, --help → usage

-p, --pion → turn on validation for pions

-n, --nucl → turn on validation for nucleons

-k, --kaon → turn on validation for kaons

If non from aboves is chosen they all are set on

-t, --total → turn on total cross section validation

-e, --energy → turn on dsigma / denergy validation

-a, --angle → turn on dsigma / dangle validation

If non from aboves is chosen they all are set on

-s, --simulation → turn on 'runfast.pl' to generate root files

-f, --plot → turn on 'intrenukeplotter.pl' to generate plots

If non from aboves is chosen they all are set on

--nof_events_diff= → number of events for differential cross section (default=1,000,000)

--nof_events_total= → number of events for total cross section (default=100,000)

--authors= → list of authors to be done (only for differential xsec)

--isospins= → list of particles to be done (only for total xsec): pip,pim,p,n,kp,km

--command= → command to run bash scripts (default=bash)

--FNAL → set command for FNAL grid (i.e. jobsub -e GENIE -e LD_LIBRARY_PATH -e PATH)

Examples

- To run simulations and create plots for all available pion data sets for total cross section use:

  ```
  ./fsiv.py --pion --total
  ```

- To create plots (w/o running simulations) for author1 and author2 for nucleons data use:

  ```
  ./fsiv.py -n --plots --authors='author1 author2'
  ```

- To run simulations and create plots for all particles for dsigma/denergy data sets with custom command use:

  ```
  ./fsiv.py --energy --command='grid_command --grid_option1 --grid_option2...'
  ```

- To run only simulations for total cross section for positive pion and positive kaon with 1000 events use:

  ```
  ./fsiv.py -t --isospin='pip kp' --nof_events_total=1000
  ```

- To run full validation on FNAL grid use:

  ```
  ./fsiv.py --FNAL
  ```