

FTF and ElasticPP generators at SPD

FTF

- 1) Part of Geant4 toolkit, but is available in spdroot:
spdroot/spdgenerators/FtfGen/
- 2) Works at $\sqrt{s} < 10$ GeV
- 3) Generates only minimum bias events
- 4) Generated events are stored in input/FTF_tmp.root
- 5) Works in several modes:

Mode = -1

Read already generated FTF_tmp.root

Mode = 0

Generates events, all the settings are passed through
config file (see examples in spdgenerators/FtfGen/PP.mac)

NEW! Mode = 1

Generates events, all the settings
are provided by user

Incident particle

Target material (taken from G4 Nist tables)

Target mass number

Generator (usually ftfp)

Number of events

Plab or $\sqrt{s_{NN}}$ value

Examples

```
/* ++++++ DEFINE PRIMARY GENERATORS ++++++ */
SpdPrimaryGenerator* primGen = new SpdPrimaryGenerator();

//-----FTF GENERATOR-----
// mode = -1 - read the existing datafile
// mode = 0 - generate and read the datafile
// mode = 1 - generate events without config file
SpdFtfGenerator* ftfgen = new SpdFtfGenerator(1 /*mode*/);
ftfgen->SetRemoveDataFile(kFALSE);
ftfgen->SetSeed(seed /*seed*/, 0 /*events_to_skip+1*/);

ftfgen->SetSqrtS(5.0); //sqrt s energy
ftfgen->SetNEvents(nEvents); //generated number of events
ftfgen->SetParticle("proton"); //projectile particle
ftfgen->SetTartetMass(1); //mass number of target nucleus

// "ftfp" generator is couple of FTF model and simple Preco model of Geant4.
// "ftfb" generator is couple of FTF model and Binary cascade model of Geant4.
ftfgen->SetGenerator("ftfp");
ftfgen->SetMaterialName("G4_H"); //target particle

primGen->AddGenerator(ftfgen);
//-----
```

```
/* ++++++ DEFINE PRIMARY GENERATORS ++++++ */
SpdPrimaryGenerator* primGen = new SpdPrimaryGenerator();

//-----FTF GENERATOR-----
// mode = -1 - read the existing datafile
// mode = 0 - generate and read the datafile
// mode = 1 - generate events without config file
SpdFtfGenerator* ftfgen = new SpdFtfGenerator(1 /*mode*/);
ftfgen->SetRemoveDataFile(kFALSE);
ftfgen->SetSeed(seed /*seed*/, 0 /*events_to_skip+1*/);

ftfgen->SetSqrtS(5.0); //sqrt s energy
ftfgen->SetNEvents(nEvents); //generated number of events
ftfgen->SetParticle("deuteron"); //projectile particle
ftfgen->SetTartetMass(2); //mass number of target nucleus

// "ftfp" generator is couple of FTF model and simple Preco model of Geant4.
// "ftfb" generator is couple of FTF model and Binary cascade model of Geant4.
ftfgen->SetGenerator("ftfp");
ftfgen->SetMaterialName("G4_Deuterium"); //target particle

primGen->AddGenerator(ftfgen);
//-----
```

Generated events will be stored in `spdroot/input/FTF_tmp.root`

- Optional:
ftfgen->SetGenPath(«some path»);
will change the directory with generated events.

Elastic pp generation

Generates only pp-elastic events.

We believe that it works correctly at low $|t|$.

Different parametrization at

$$\sqrt{s} < 3.5 \text{ GeV}$$

$$3.5 < \sqrt{s} < 4.3$$

$$\sqrt{s} > 4.3 \text{ GeV}$$

Regime1: only hadronic Elastic scattering

Regime2: Coulomb, interference and hadronic parts

Generated events are stored in

spdgenerators/PPelastic/ppelast.root

An example of usage is available

spdroot/macro/primgen/TestElasticPP.C

V. Uzhinsky, A. Galoyan, Q. Hu, J. Ritman and H. Xu, “Empirical parametrization of the nucleon-nucleon elastic scattering amplitude at high beam momenta for Glauber calculations and Monte Carlo simulations,” Phys. Rev. C 94 (2016) no.6, 064003 doi:10.1103/PhysRevC.94.064003 [arXiv:1603.04731 [hep-ph]].

```
/* ++++++ DEFINE PRIMARY GENERATORS ++++++ */
SpdPrimaryGenerator* primGen = new SpdPrimaryGenerator();

SpdElPPGenerator* ppgen = new SpdElPPGenerator(0);
ppgen->SetRemoveDataFile(kFALSE);
ppgen->SetNEvents(nEvents);
ppgen->SetSqrtS(5.0);
ppgen->SetThetaMin(0.2);

primGen->AddGenerator(ppgen);
//-----
run->SetGenerator(primGen);
```