Overview and Use of SIMC

Dave Gaskell

NPS Collaboration Meeting

July 17-18, 2022

1. SIMC overview
2. Basics of using SIMC

What is SIMC?

SIMC is the standard Hall C Monte Carlo for coincidence reactions (similar to MCEEP) → written in FORTRAN (now gfortran compatible …)

Features:

→ Optics (COSY) and "aperture checking" Monte Carlos of spectrometers

[HMS, SOS, SHMS, HRS's, BigCal,…]

→ Includes radiative effects, multiple scattering, ionization energy loss, particle decay

→ Simple prescriptions available for FSIs, Coulomb Corrections, etc.

Reactions implemented:

1. Elastic and quasielastic → H(e,e'p), A(e,e'p)
2. Exclusive pion production

   → H(e,e'π^+)n, A(e,e'$\pi^{+/-}$) [quasifree or coherent]
3. Kaon electroproduction → H(e,e'K^+)Λ,Σ, A(e,e'$K^{+/-}$)
4. H(e,e'$\pi^{+/-}$)X, D(e,e'$\pi^{+/-}$)X [semi-inclusive]
5. H(e,e'$K^{+/-}$)X, D(e,e'$K^{+/-}$)X [semi-inclusive]
6. H(e,e'$\rho \rightarrow \pi^+\pi^-$)p, D(e,e'$\rho \rightarrow \pi^+\pi^-$) [diffractive ρ]
7. *H(e,e'π^0)p, D(e,e'π^0)p/n, (e,e'π^0)X* ← **Peter's updates underway**

What SIMC is NOT…

SIMC is NOT a full detector response simulation a la GEANT

SIMC does NOT simulate a large class of processes simultaneously to generate backgrounds (like Pythia for example)

SIMC is not a generic event generator → processes are generated over a limited phase space with the specific purpose of being “thrown” into a spectrometer

SIMC is not hard to modify → if you want it to do something else (different cross section model, add new spectrometer, etc.) it is pretty easy to update + I can help

Overview

- Initialization
  - Choose reaction, final state (if appropriate)
  - Disable/enable implementation of (or correction for) raster, eloss …
- Event generation
  - Select vertex based on target size, position, raster size, beam spot size
  - Determine energy, angle generation that will populate 100% of the acceptance (accounting for radiation, energy loss, …)
- Physics Processes
  - Event-by-event multiple scattering, radiative corrections, particle decay, coulomb corrections
- Acceptance
  - Can apply geometric cuts or spectrometer model. Default spec. models include target/spec. offsets, model of magnetic elements, apertures at front, back, middle of magnets, collimators, detector active area
- Event Reconstruction
  - Tracks are fitted in the focal plane and reconstructed to the target. Apply (average) energy loss, fast raster corrections. Calculate physics quantities for Ntuple/Root tree

Example: Exclusive Pion Electroproduction

Initialize limits:

Generate vertex:

Generate scattering kinematics:

H(e,e'π): Generate θ_e, $\phi_{e,}$ θ_π, ϕ_π, p_e – calculate p_π

D(e,e'π): Generate θ_e, $\phi_{e,}$ θ_π, ϕ_π, $p_{e,}$ θ_N ϕ_N, p_N → calculate p_π

Bound nucleon momentum/direction

Modifications to kinematics:

Follow particles:

Apply cross section weighting (use model for free proton/neutron in γ-N center of mass)

Simulate particle decay

→ Decay the meson and follow the decay product(s)

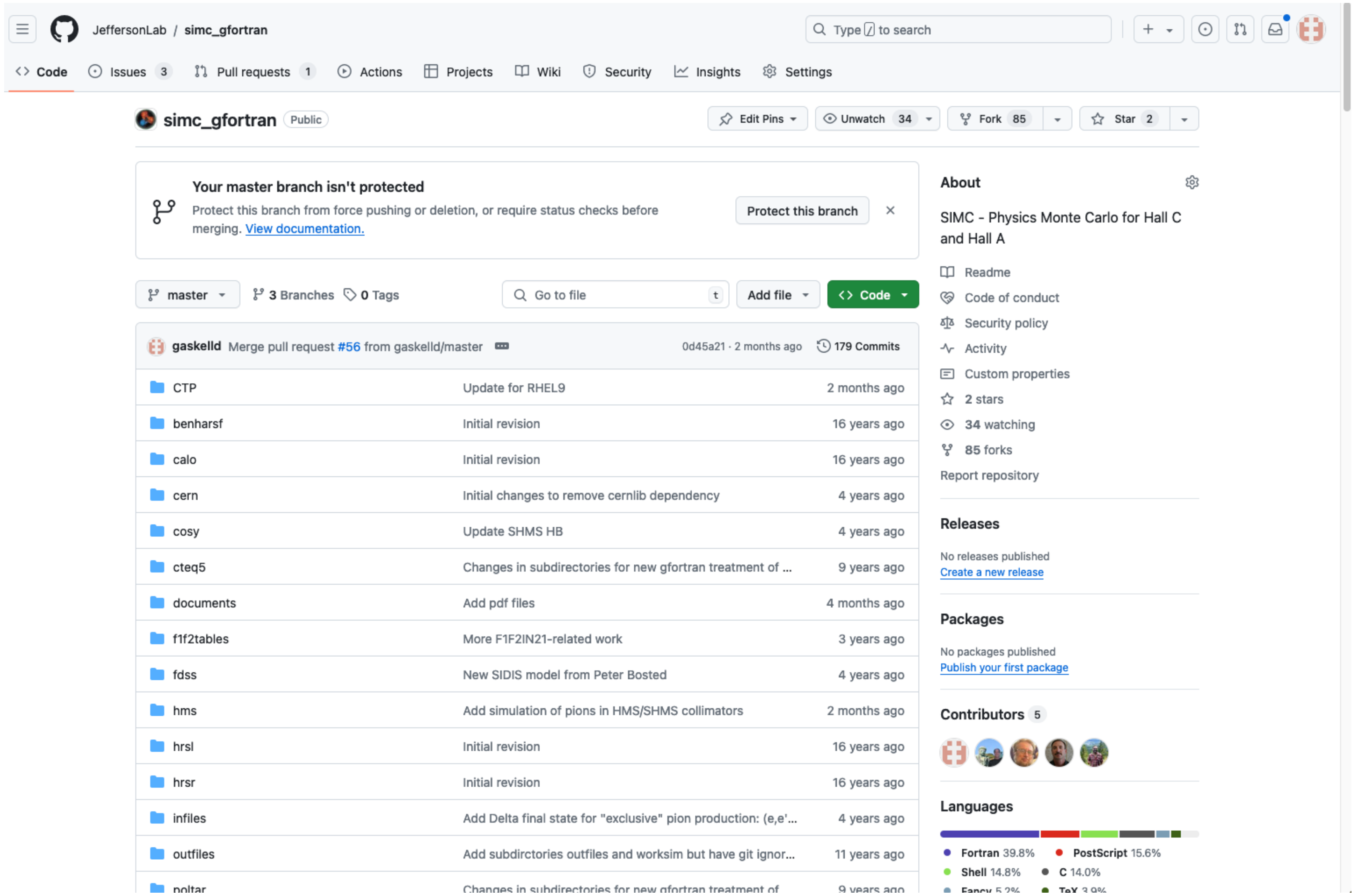
Calculate Normalization Factors:

Spectrometer Models (HMS, SHMS, HRS ...)

- Magnetic elements simulated using COSY
- Real apertures
  - collimators, vacuum pipes
  - magnet apertures (front, middle, and back)
- Fiducial cuts for detectors (depends on analysis, trigger)
- Sequential transformations for magnetic elements, continuous drift for field free regions (important for decay)
- Reconstruct track
  - Use smeared position at drift chamber
  - Fit track at focal plane
  - Reconstruct to target using matrix elements fit from MC data
- Can easily disable spectrometer, apply geometric cuts, add new spectrometer/detector (for example, NPS)

Getting SIMC

https://github.com/JeffersonLab/simc_gfortran



JeffersonLab / simc_gfortran

Code | Issues 3 | Pull requests 1 | Actions | Projects | Wiki | Security | Insights | Settings

simc_gfortran Public

Edit Pins | Unwatch 34 | Fork 85 | Star 2

Your master branch isn't protected

Protect this branch from force pushing or deletion, or require status checks before merging. View documentation.

Protect this branch

master | 3 Branches | 0 Tags | Go to file | Add file | Code

gaskelld Merge pull request #56 from gaskelld/master — 0d45a21 · 2 months ago — 179 Commits

CTP	Update for RHEL9	2 months ago
benharsf	Initial revision	16 years ago
calo	Initial revision	16 years ago
cern	Initial changes to remove cernlib dependency	4 years ago
cosy	Update SHMS HB	4 years ago
cteq5	Changes in subdirectories for new gfortran treatment of ...	9 years ago
documents	Add pdf files	4 months ago
f1f2tables	More F1F2IN21-related work	3 years ago
fdss	New SIDIS model from Peter Bosted	4 years ago
hms	Add simulation of pions in HMS/SHMS collimators	2 months ago
hrsl	Initial revision	16 years ago
hrsr	Initial revision	16 years ago
infiles	Add Delta final state for "exclusive" pion production: (e,e'...	4 years ago
outfiles	Add subdirctories outfiles and worksim but have git ignor...	11 years ago
poltar	Changes in subdirectories for new gfortran treatment of	9 years ago

About

SIMC - Physics Monte Carlo for Hall C and Hall A

Readme
Code of conduct
Security policy
Activity
Custom properties
2 stars
34 watching
85 forks
Report repository

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 5

Languages

● Fortran 39.8% ● PostScript 15.6%
● Shell 14.8% ● C 14.0%
● Fancy 5.2% ● TeX 3.9%

SIMC Basics

- Compiling: just type "make" in the main directory
- Important directories
  - infiles → where the input files for your simulation live
  - worksim → SIMC event-by-event output (root trees)
  - outfiles → where the *.hist file lives → need this for normfac
  - util → "helper" applications to generate output
- Helper applications
  - SIMC default output is a FORTRAN binary file
  - Helper applications in util directory convert to root trees or paw ntuple
  - util/ntuple and util/root_tree → must be compiled separately
  - Choose appropriate Makefile (Makefile.rhel9 or Makefile.rhel7)

SIMC: structures → derived types

- Long ago – had to convert from “structures” to ”derived types”

```
structure /double_arm/
        structure /arm/ e
                real*8          delta, yptar, xptar, z
        end structure
        structure /arm2/ p
                real*8          delta, yptar, xptar, z
        end structure
end structure
```

Use as variables, like:

```
recon.e.delta
vertex.p.xptar
```

```
type arm
                sequence
                real*8          delta, yptar, xptar, z
end type

type arm2
                sequence
                real*8          delta, yptar, xptar, z
end type

type double_arm
                sequence
                type(arm)::e
                type(arm2)::p
end type
```

Use as variables, like:

```
recon%e%delta
vertex%p%xptar
```

SIMC RHEL9/Alma9 updates: derived types in modules

- First try to transition to Alma9 – all the derived types were broken
- After lots of googling found a helpful web page that described the problem

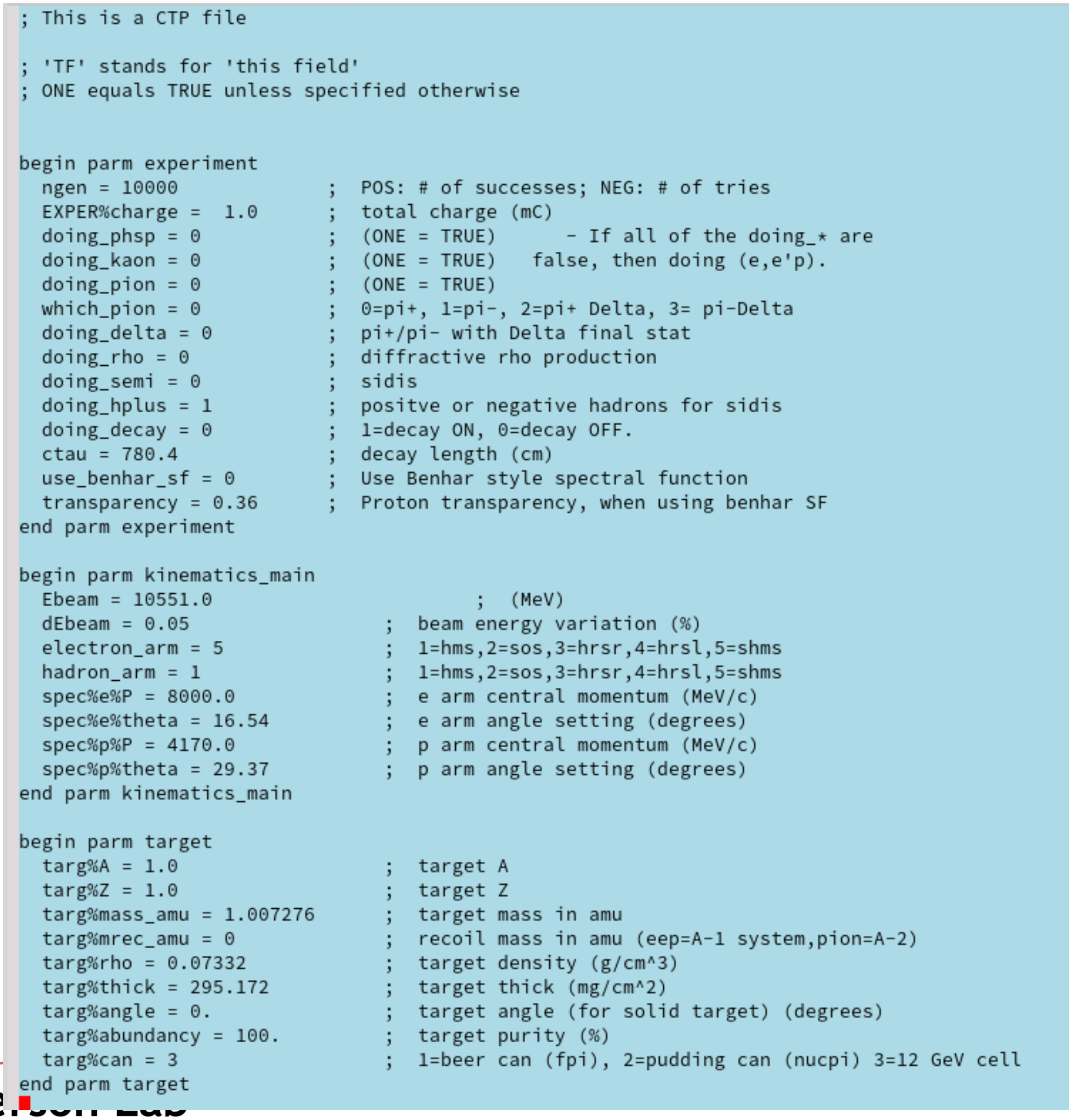
- Compilers can handle derived types in 2 different ways
  - Name equivalence: two types are equal if they have the same name
  - Index equivalence: each time a type is declared/defined, it's given a unique index, even if they have the same name!
  - Old compiler used first option, new gfortran uses second
- How to fix/deal with this?
  - Put the derived types in a module and load the module in each relevant subroutine
  - All the derived types that were in "structures.inc" are now in "modules.f"

```
 MODULE structureModule
! Define some BASIC record structures and associated parameters
! ... generic cut -->  initialized with MAX large and MIN small
        type cutstype
               sequence
               real*8           min, max
        end type cutstype
…
…
…
```

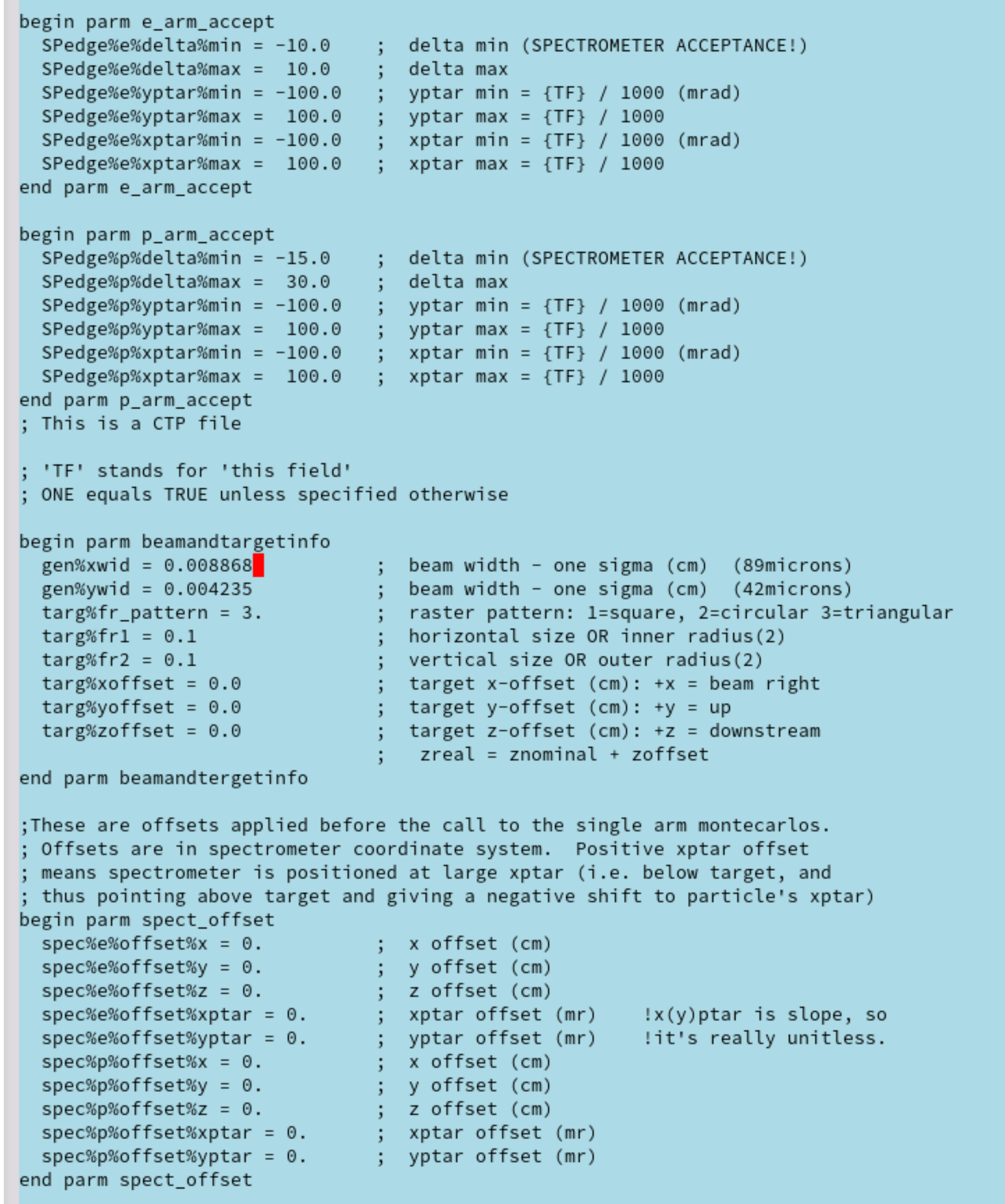
In each relevant subroutine, added this statement:

```
USE structureModule
```

SIMC Input File



```
; This is a CTP file

; 'TF' stands for 'this field'
; ONE equals TRUE unless specified otherwise


begin parm experiment
  ngen = 10000               ;  POS: # of successes; NEG: # of tries
  EXPER%charge =  1.0        ;  total charge (mC)
  doing_phsp = 0             ;  (ONE = TRUE)      - If all of the doing_* are
  doing_kaon = 0             ;  (ONE = TRUE)   false, then doing (e,e'p).
  doing_pion = 0             ;  (ONE = TRUE)
  which_pion = 0             ;  0=pi+, 1=pi-, 2=pi+ Delta, 3= pi-Delta
  doing_delta = 0            ;  pi+/pi- with Delta final stat
  doing_rho = 0              ;  diffractive rho production
  doing_semi = 0             ;  sidis
  doing_hplus = 1            ;  positve or negative hadrons for sidis
  doing_decay = 0            ;  1=decay ON, 0=decay OFF.
  ctau = 780.4               ;  decay length (cm)
  use_benhar_sf = 0          ;  Use Benhar style spectral function
  transparency = 0.36        ;  Proton transparency, when using benhar SF
end parm experiment

begin parm kinematics_main
  Ebeam = 10551.0                         ;  (MeV)
  dEbeam = 0.05                  ;  beam energy variation (%)
  electron_arm = 5               ;  1=hms,2=sos,3=hrsr,4=hrsl,5=shms
  hadron_arm = 1                 ;  1=hms,2=sos,3=hrsr,4=hrsl,5=shms
  spec%e%P = 8000.0              ;  e arm central momentum (MeV/c)
  spec%e%theta = 16.54           ;  e arm angle setting (degrees)
  spec%p%P = 4170.0              ;  p arm central momentum (MeV/c)
  spec%p%theta = 29.37           ;  p arm angle setting (degrees)
end parm kinematics_main

begin parm target
  targ%A = 1.0                   ;  target A
  targ%Z = 1.0                   ;  target Z
  targ%mass_amu = 1.007276       ;  target mass in amu
  targ%mrec_amu = 0              ;  recoil mass in amu (eep=A-1 system,pion=A-2)
  targ%rho = 0.07332             ;  target density (g/cm^3)
  targ%thick = 295.172           ;  target thick (mg/cm^2)
  targ%angle = 0.                ;  target angle (for solid target) (degrees)
  targ%abundancy = 100.          ;  target purity (%)
  targ%can = 3                   ;  1=beer can (fpi), 2=pudding can (nucpi) 3=12 GeV cell
end parm target
```



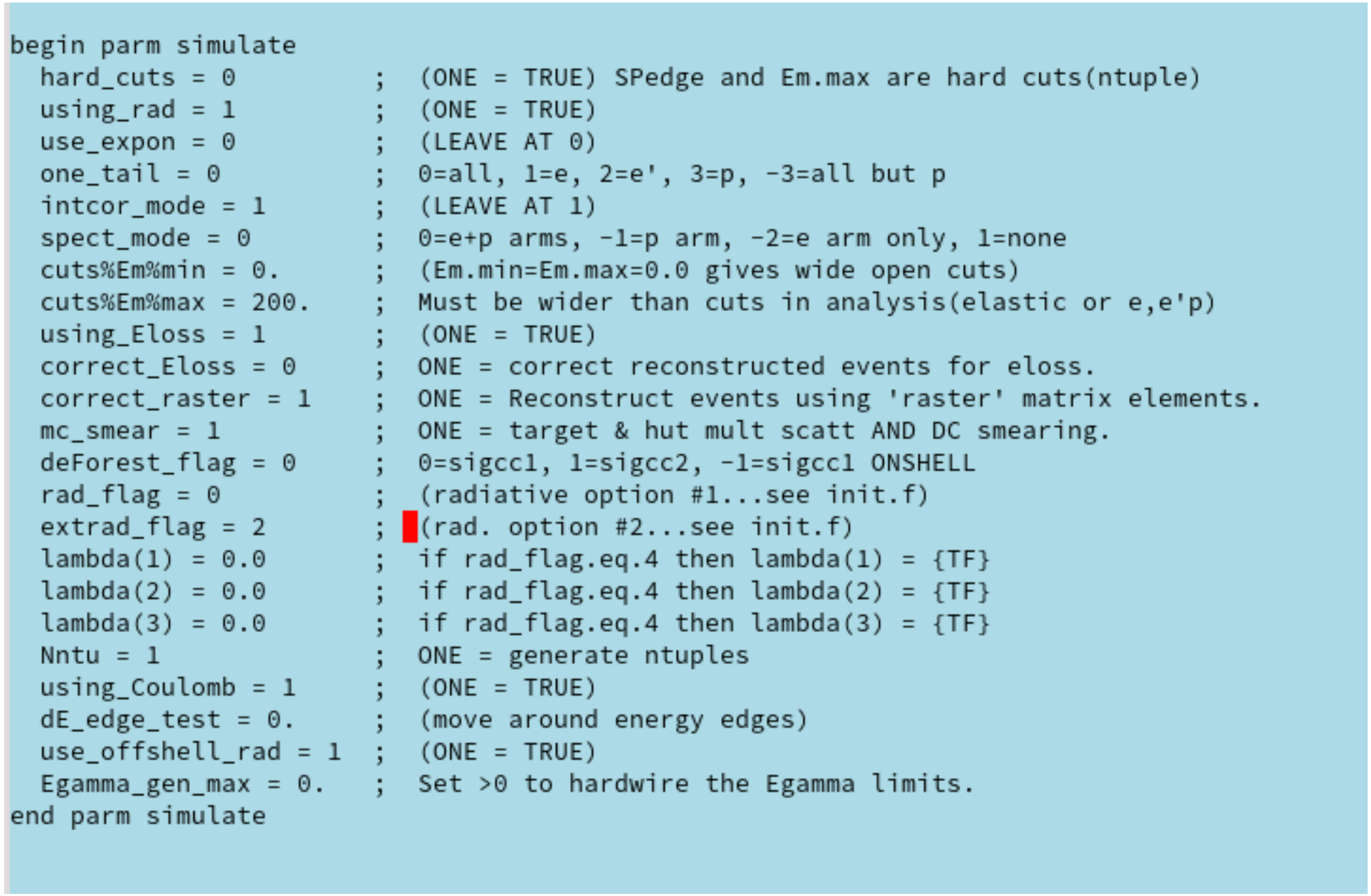
```
begin parm e_arm_accept
  SPedge%e%delta%min = -10.0      ;  delta min (SPECTROMETER ACCEPTANCE!)
  SPedge%e%delta%max =  10.0      ;  delta max
  SPedge%e%yptar%min = -100.0     ;  yptar min = {TF} / 1000 (mrad)
  SPedge%e%yptar%max =  100.0     ;  yptar max = {TF} / 1000
  SPedge%e%xptar%min = -100.0     ;  xptar min = {TF} / 1000 (mrad)
  SPedge%e%xptar%max =  100.0     ;  xptar max = {TF} / 1000
end parm e_arm_accept

begin parm p_arm_accept
  SPedge%p%delta%min = -15.0      ;  delta min (SPECTROMETER ACCEPTANCE!)
  SPedge%p%delta%max =  30.0      ;  delta max
  SPedge%p%yptar%min = -100.0     ;  yptar min = {TF} / 1000 (mrad)
  SPedge%p%yptar%max =  100.0     ;  yptar max = {TF} / 1000
  SPedge%p%xptar%min = -100.0     ;  xptar min = {TF} / 1000 (mrad)
  SPedge%p%xptar%max =  100.0     ;  xptar max = {TF} / 1000
end parm p_arm_accept
; This is a CTP file

; 'TF' stands for 'this field'
; ONE equals TRUE unless specified otherwise

begin parm beamandtargetinfo
  gen%xwid = 0.008868             ;  beam width - one sigma (cm)   (89microns)
  gen%ywid = 0.004235             ;  beam width - one sigma (cm)   (42microns)
  targ%fr_pattern = 3.            ;  raster pattern: 1=square, 2=circular 3=triangular
  targ%fr1 = 0.1                  ;  horizontal size OR inner radius(2)
  targ%fr2 = 0.1                  ;  vertical size OR outer radius(2)
  targ%xoffset = 0.0              ;  target x-offset (cm): +x = beam right
  targ%yoffset = 0.0              ;  target y-offset (cm): +y = up
  targ%zoffset = 0.0              ;  target z-offset (cm): +z = downstream
                                  ;   zreal = znominal + zoffset
end parm beamandtergetinfo

;These are offsets applied before the call to the single arm montecarlos.
; Offsets are in spectrometer coordinate system.  Positive xptar offset
; means spectrometer is positioned at large xptar (i.e. below target, and
; thus pointing above target and giving a negative shift to particle's xptar)
begin parm spect_offset
  spec%e%offset%x = 0.            ;  x offset (cm)
  spec%e%offset%y = 0.            ;  y offset (cm)
  spec%e%offset%z = 0.            ;  z offset (cm)
  spec%e%offset%xptar = 0.        ;  xptar offset (mr)    !x(y)ptar is slope, so
  spec%e%offset%yptar = 0.        ;  yptar offset (mr)    !it's really unitless.
  spec%p%offset%x = 0.            ;  x offset (cm)
  spec%p%offset%y = 0.            ;  y offset (cm)
  spec%p%offset%z = 0.            ;  z offset (cm)
  spec%p%offset%xptar = 0.        ;  xptar offset (mr)
  spec%p%offset%yptar = 0.        ;  yptar offset (mr)
end parm spect_offset
```

SIMC Input File

These parameters should not change much in general

→ ”using_Eloss” should always be on to simulate ionization energy loss, but “correct_Eloss” should be off since hcana does not apply a correction

→ Don’t change the radiation flags, except maybe “one_tail” → might want to turn off proton radiation for neutral particles



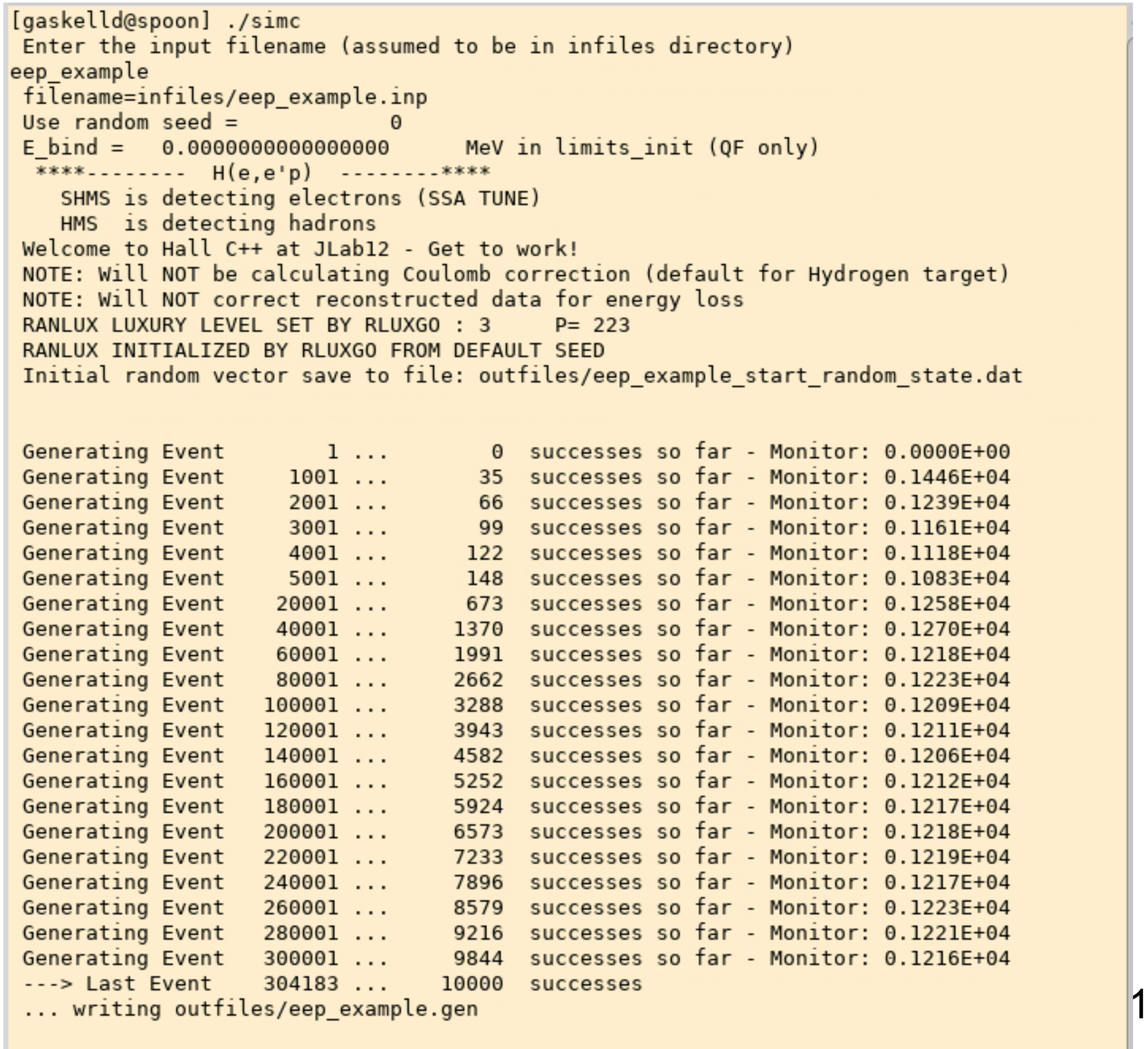
```
begin parm simulate
  hard_cuts = 0          ;  (ONE = TRUE) SPedge and Em.max are hard cuts(ntuple)
  using_rad = 1          ;  (ONE = TRUE)
  use_expon = 0          ;  (LEAVE AT 0)
  one_tail = 0           ;  0=all, 1=e, 2=e', 3=p, -3=all but p
  intcor_mode = 1        ;  (LEAVE AT 1)
  spect_mode = 0         ;  0=e+p arms, -1=p arm, -2=e arm only, 1=none
  cuts%Em%min = 0.       ;  (Em.min=Em.max=0.0 gives wide open cuts)
  cuts%Em%max = 200.     ;  Must be wider than cuts in analysis(elastic or e,e'p)
  using_Eloss = 1        ;  (ONE = TRUE)
  correct_Eloss = 0      ;  ONE = correct reconstructed events for eloss.
  correct_raster = 1     ;  ONE = Reconstruct events using 'raster' matrix elements.
  mc_smear = 1           ;  ONE = target & hut mult scatt AND DC smearing.
  deForest_flag = 0      ;  0=sigcc1, 1=sigcc2, -1=sigcc1 ONSHELL
  rad_flag = 0           ;  (radiative option #1...see init.f)
  extrad_flag = 2        ;  (rad. option #2...see init.f)
  lambda(1) = 0.0        ;  if rad_flag.eq.4 then lambda(1) = {TF}
  lambda(2) = 0.0        ;  if rad_flag.eq.4 then lambda(2) = {TF}
  lambda(3) = 0.0        ;  if rad_flag.eq.4 then lambda(3) = {TF}
  Nntu = 1               ;  ONE = generate ntuples
  using_Coulomb = 1      ;  (ONE = TRUE)
  dE_edge_test = 0.      ;  (move around energy edges)
  use_offshell_rad = 1   ;  (ONE = TRUE)
  Egamma_gen_max = 0.    ;  Set >0 to hardwire the Egamma limits.
end parm simulate
```

Input File Notes

- Kinematics
  - Units are in MeV
  - No need to account for ionization energy loss – that's in simulation
  - Include synchrotron energy loss in beam energy (if relevant)
  - If you find offsets to nominal kinematics in your analysis – apply those to the SIMC kinematics → will impact the model cross section calculation
- Make sure to include contraction of target cell at low temperatures in target information
- Target info mixes units, density in **g**/cm^3, but thickness in **mg**/cm^2
- Generation volume: must be larger than nominal acceptance to get correct yield and cross sections. You can easily check in the *.hist file if the "found" values are too close to the generation limits
- Despite the names, "targ%yoffset" and "targ%xoffset" are beam position offsets

Running SIMC and Generating Output

- Create your input file: eep_example.inp
- run SIMC

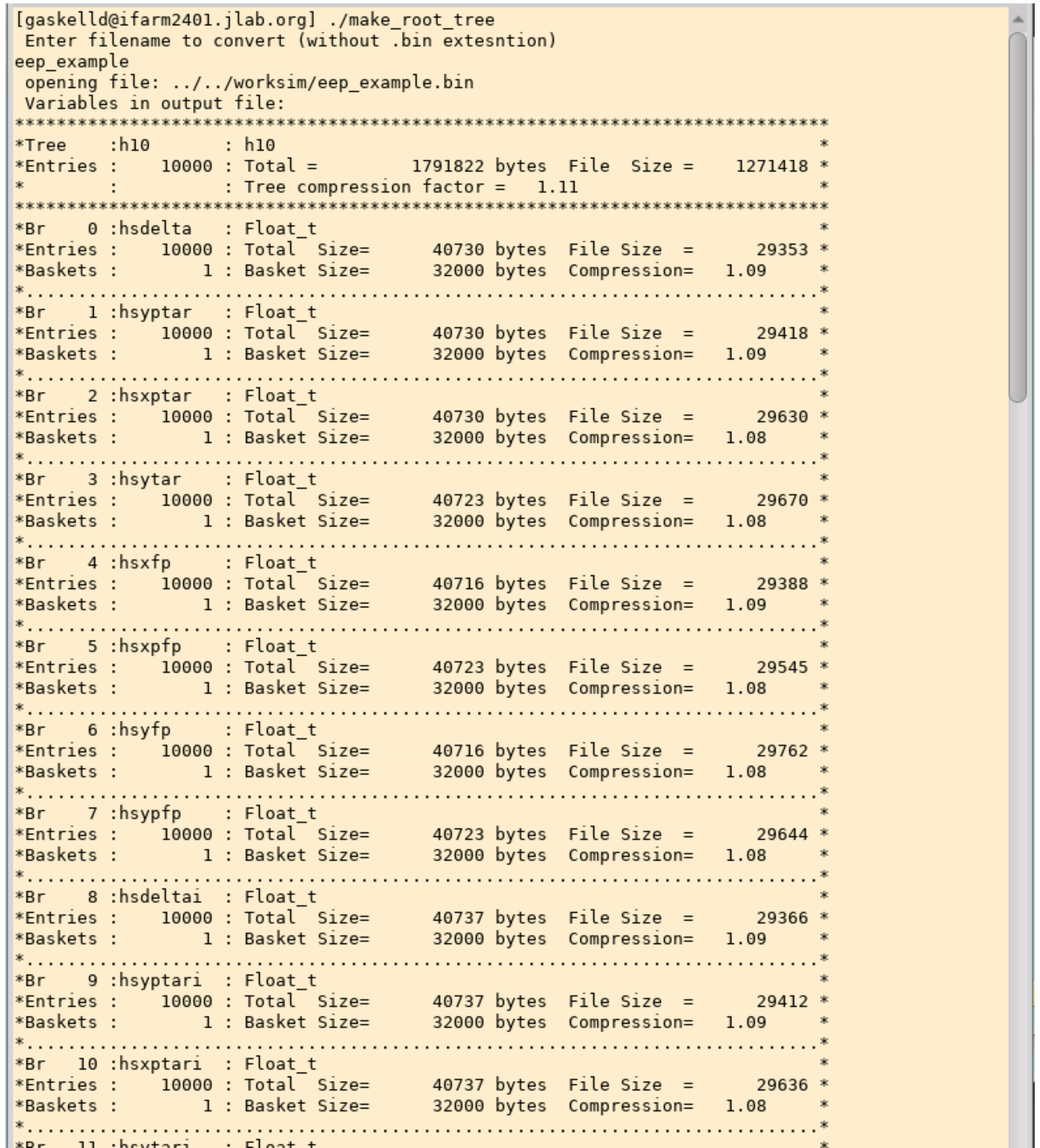


```
[gaskelld@spoon] ./simc
 Enter the input filename (assumed to be in infiles directory)
eep_example
 filename=infiles/eep_example.inp
 Use random seed =              0
 E_bind =   0.0000000000000000      MeV in limits_init (QF only)
  ****--------  H(e,e'p)  --------****
    SHMS is detecting electrons (SSA TUNE)
    HMS  is detecting hadrons
 Welcome to Hall C++ at JLab12 - Get to work!
 NOTE: Will NOT be calculating Coulomb correction (default for Hydrogen target)
 NOTE: Will NOT correct reconstructed data for energy loss
 RANLUX LUXURY LEVEL SET BY RLUXGO : 3      P= 223
 RANLUX INITIALIZED BY RLUXGO FROM DEFAULT SEED
 Initial random vector save to file: outfiles/eep_example_start_random_state.dat


 Generating Event        1 ...          0  successes so far - Monitor: 0.0000E+00
 Generating Event     1001 ...         35  successes so far - Monitor: 0.1446E+04
 Generating Event     2001 ...         66  successes so far - Monitor: 0.1239E+04
 Generating Event     3001 ...         99  successes so far - Monitor: 0.1161E+04
 Generating Event     4001 ...        122  successes so far - Monitor: 0.1118E+04
 Generating Event     5001 ...        148  successes so far - Monitor: 0.1083E+04
 Generating Event    20001 ...        673  successes so far - Monitor: 0.1258E+04
 Generating Event    40001 ...       1370  successes so far - Monitor: 0.1270E+04
 Generating Event    60001 ...       1991  successes so far - Monitor: 0.1218E+04
 Generating Event    80001 ...       2662  successes so far - Monitor: 0.1223E+04
 Generating Event   100001 ...       3288  successes so far - Monitor: 0.1209E+04
 Generating Event   120001 ...       3943  successes so far - Monitor: 0.1211E+04
 Generating Event   140001 ...       4582  successes so far - Monitor: 0.1206E+04
 Generating Event   160001 ...       5252  successes so far - Monitor: 0.1212E+04
 Generating Event   180001 ...       5924  successes so far - Monitor: 0.1217E+04
 Generating Event   200001 ...       6573  successes so far - Monitor: 0.1218E+04
 Generating Event   220001 ...       7233  successes so far - Monitor: 0.1219E+04
 Generating Event   240001 ...       7896  successes so far - Monitor: 0.1217E+04
 Generating Event   260001 ...       8579  successes so far - Monitor: 0.1223E+04
 Generating Event   280001 ...       9216  successes so far - Monitor: 0.1221E+04
 Generating Event   300001 ...       9844  successes so far - Monitor: 0.1216E+04
 ---> Last Event    304183 ...      10000  successes
 ... writing outfiles/eep_example.gen
```

Jefferson Lab

Running SIMC and Generating Output

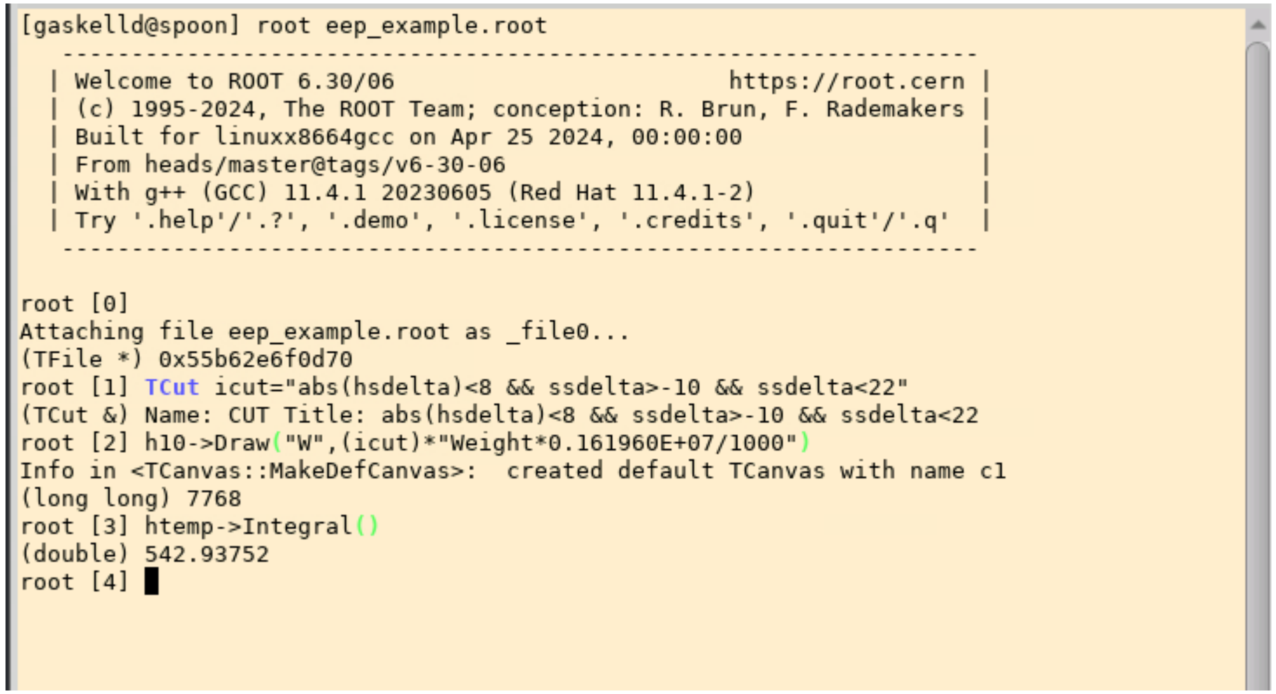
- Go to util/root_tree directory
- ./make_root_tree



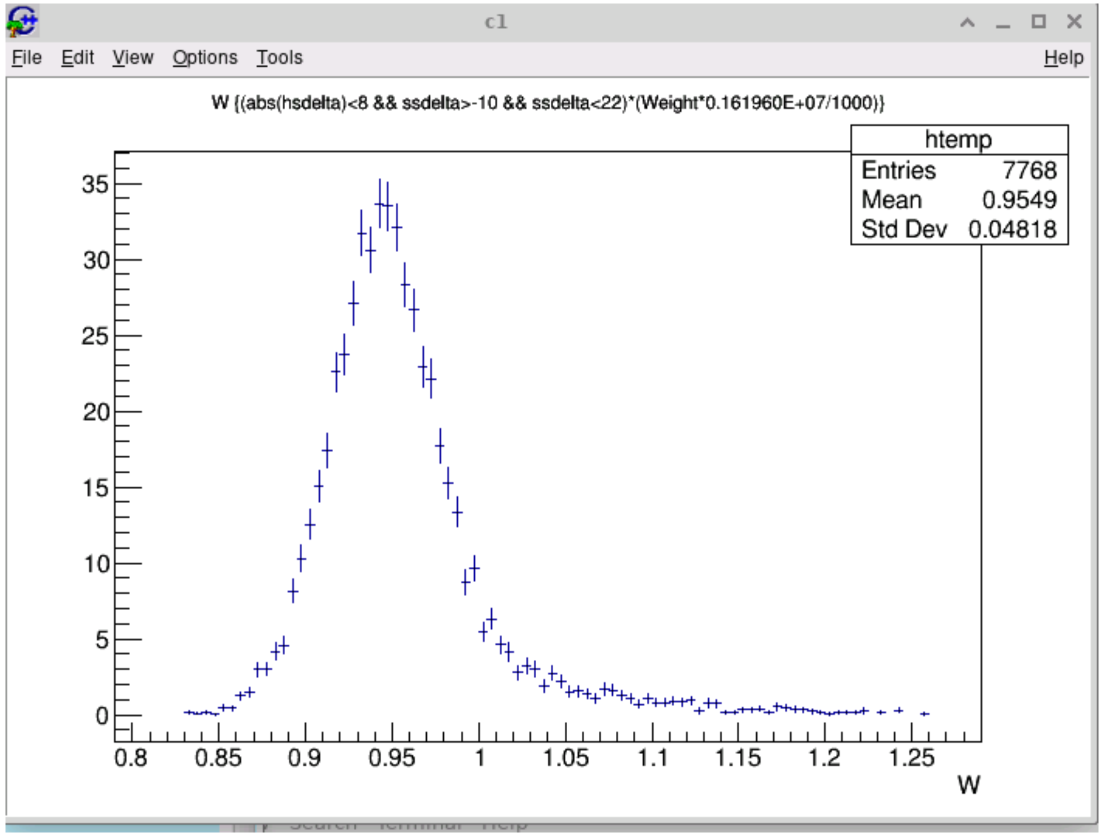
```
[gaskelld@ifarm2401.jlab.org] ./make_root_tree
 Enter filename to convert (without .bin extesntion)
eep_example
 opening file: ../../worksim/eep_example.bin
 Variables in output file:
******************************************************************************
*Tree    :h10       : h10                                                    *
*Entries :    10000 : Total =         1791822 bytes  File  Size =    1271418 *
*        :          : Tree compression factor =   1.11                       *
******************************************************************************
*Br    0 :hsdelta   : Float_t                                                *
*Entries :    10000 : Total  Size=      40730 bytes  File Size  =      29353 *
*Baskets :        1 : Basket Size=      32000 bytes  Compression=   1.09     *
*............................................................................*
*Br    1 :hsyptar   : Float_t                                                *
*Entries :    10000 : Total  Size=      40730 bytes  File Size  =      29418 *
*Baskets :        1 : Basket Size=      32000 bytes  Compression=   1.09     *
*............................................................................*
*Br    2 :hsxptar   : Float_t                                                *
*Entries :    10000 : Total  Size=      40730 bytes  File Size  =      29630 *
*Baskets :        1 : Basket Size=      32000 bytes  Compression=   1.08     *
*............................................................................*
*Br    3 :hsytar    : Float_t                                                *
*Entries :    10000 : Total  Size=      40723 bytes  File Size  =      29670 *
*Baskets :        1 : Basket Size=      32000 bytes  Compression=   1.08     *
*............................................................................*
*Br    4 :hsxfp     : Float_t                                                *
*Entries :    10000 : Total  Size=      40716 bytes  File Size  =      29388 *
*Baskets :        1 : Basket Size=      32000 bytes  Compression=   1.09     *
*............................................................................*
*Br    5 :hsxpfp    : Float_t                                                *
*Entries :    10000 : Total  Size=      40723 bytes  File Size  =      29545 *
*Baskets :        1 : Basket Size=      32000 bytes  Compression=   1.08     *
*............................................................................*
*Br    6 :hsyfp     : Float_t                                                *
*Entries :    10000 : Total  Size=      40716 bytes  File Size  =      29762 *
*Baskets :        1 : Basket Size=      32000 bytes  Compression=   1.08     *
*............................................................................*
*Br    7 :hsypfp    : Float_t                                                *
*Entries :    10000 : Total  Size=      40723 bytes  File Size  =      29644 *
*Baskets :        1 : Basket Size=      32000 bytes  Compression=   1.08     *
*............................................................................*
*Br    8 :hsdeltai  : Float_t                                                *
*Entries :    10000 : Total  Size=      40737 bytes  File Size  =      29366 *
*Baskets :        1 : Basket Size=      32000 bytes  Compression=   1.09     *
*............................................................................*
*Br    9 :hsyptari  : Float_t                                                *
*Entries :    10000 : Total  Size=      40737 bytes  File Size  =      29412 *
*Baskets :        1 : Basket Size=      32000 bytes  Compression=   1.09     *
*............................................................................*
*Br   10 :hsxptari  : Float_t                                                *
*Entries :    10000 : Total  Size=      40737 bytes  File Size  =      29636 *
*Baskets :        1 : Basket Size=      32000 bytes  Compression=   1.08     *
*............................................................................*
*Br   11 :hsytari   : Float_t
```

Jefferson Lab

Make some plots and get some yields



```
[gaskelld@spoon] root eep_example.root
   ------------------------------------------------------------------
  | Welcome to ROOT 6.30/06                        https://root.cern |
  | (c) 1995-2024, The ROOT Team; conception: R. Brun, F. Rademakers |
  | Built for linuxx8664gcc on Apr 25 2024, 00:00:00                 |
  | From heads/master@tags/v6-30-06                                  |
  | With g++ (GCC) 11.4.1 20230605 (Red Hat 11.4.1-2)                |
  | Try '.help'/'.?', '.demo', '.license', '.credits', '.quit'/'.q'  |
   ------------------------------------------------------------------

root [0]
Attaching file eep_example.root as _file0...
(TFile *) 0x55b62e6f0d70
root [1] TCut icut="abs(hsdelta)<8 && ssdelta>-10 && ssdelta<22"
(TCut &) Name: CUT Title: abs(hsdelta)<8 && ssdelta>-10 && ssdelta<22
root [2] h10->Draw("W",(icut)*"Weight*0.161960E+07/1000")
Info in <TCanvas::MakeDefCanvas>:  created default TCanvas with name c1
(long long) 7768
root [3] htemp->Integral()
(double) 542.93752
root [4]
```



Normalized yield by applying "Weight" from tree

→ Includes cross section, radiative corrections, small jacobian

→ What's this 0.161960E+07 thing?

  → "normfac" from eep_example.hist file

  → Includes target thickness, simulated charge, generation volume

→ Also need to divide by number of events in tree

Result from this simulation (with these cuts) was:
Yield = 542.9 counts/mC

Jefferson Lab

Reconstructed Quantities in SIMC

- Calculation of reconstructed quantities (W, x, Q2, etc.) may not match how these quantities are calculated in hcana
  - Reconstruction was originally intended to match calculations in Hall C 6 GeV fortran analyzer
- Found to be a problem for H(e,e'p) analysis for KaonLT/PionLT → definitions of missing momentum components weren't even consistent
- Ideally, we should make calculations in simc match hcana
  - In the short term, should use stand-alone scripts to make comparable output
  - Richard Trotta has done this for H(e,e'p) and exclusive Kaon production:

https://github.com/trottar/simc_gfortran/tree/fall_2023_kaon_xsects/recon_hcana

Updates for NPS

- Peter B. has done most of the work needed to make SIMC work for NPS
- Adding event generation for ***exclusive and semi-inclusive*** π^0 is straightforward → same as π^+ and π^-
  - Additional complication is decay of π^0 into 2 photons (extra subroutine)
- DVCS event generation same as exclusive mesons, just changing mass to zero!
- Physics models needed for cross sections:
  - SIDIS π^0 → average of π^+ and π^-
  - Exclusive π^0
    - Large W (>2 GeV): Average of π^+ and π^-, removing pion pole contributions
    - Low W (<2 GeV): MAID model
- NPS calorimeter can just be an aperture for checking acceptance. Simple resolution smearing can be added later if desired
- I'm working on incorporating Peter's updates into the main branch on version on github

SIMC Summary

- SIMC is a pretty simple tool in many ways, but powerful
- If full GEANT4 simulations are needed (e.g. for NPS) SIMC can be combined with those simulations
  - SBS is doing this for d(e,e'N) simulations
- Main drawback is that it's fortran
  - Modern cross section models are usually in C++, etc.
  - Easy work-around is to use look-up tables → Peter has done this for the MAID model of pion electroproduction for example
- Peter has already implemented many of the needed updates, integration underway