1. Introduction

NEUGEN is an event generator used to simulate neutrino - nucleus interactions over the energy range 100 MeV - 100 GeV. The program consists of approximately 17k lines of Fortran source code, mainly adhering to the Fortran77 standard, with a few F90 features like greater than 6 character variable names and the use of the 'include' statement. NEUGEN uses a pre-tabulated file of cross sections to speed execution. Generation of the table typically takes around 10 Cpu minutes, event generation itself is quite speedy, generating 100k events in about 75 cpu-sec on a 1.2 GHz Pentium III machine.

2. Installing

1. Set the environment variable NEUGEN3PATH to point to the directory into which NEUGEN has been installed.

2. Edit `neugen/Make.include` to point to the correct CERNLIB directory. NEUGEN also requires the STDHEP package. On many sites, libstdhep.a is installed in the CERNLIB directory already. If it is not, you may need to download it along with documentation from `http://www-cpd.fnal.gov/stdhep`.

3. Then just type `make all`.

   This will create several object libraries:

Library	Contents
libspi.a	coherent scattering code
libphys.a	physics models for various scattering processes
libsigcalc.a	routines for calculating cross sections
libnulook.a	routines for the nulook program
libnngen.a	routines for event generation

And 3 executables: `nulook` - program to display cross sections, `make_table` - creates the cross section tables used in neugen, `neugen_example` - an example program for calling the event generator.

3. Running NEUGEN

NEUGEN has a very minimal interface to the user. There is no disk I/O other than the reading of the cross section table file. User control over the way NEUGEN runs is established through calls to subroutines, and generated events are returned in the HEPEVT common block. The HEPEVT common is a standard common block interface for event generators, users who are not familiar with it should peruse the STDHEP documentation for more information.

To run neugen you will first need to create the cross section table, do this by running the make_table program. This will create a binary file called test_xsec.dat which neugen will access.

3.1. User Callable Routines

1. `generate_nu_event(nutype,e_nu,ia,iz)` where:

Call Parameter	Type	Description
nutype	int	neutrino type (PDG particle code)
e_nu	real	neutrino energy (GeV)
ia	int	A for the target nucleus
iz	int	Z for the target nucleus

This is the main event generation subroutine, which is called from the user program. If you give it an A/Z combo it doesn't know about it will let you know and stop.

2. `gen_control(ch_control,ival,iok)` where:

Call Parameter	Type	Description
ch_control	char*4	control word (more below)
ival	int	the value for the control variable
iok	logical	returns .true. if request succeeded

The control words one can set are:

Control Word	Value - Request
CCNC	0 = CC/NC in the right proportion
	1 = CC only
	2 = NC only
FREE	0 = bound nucleons
	1 = free nucleons
NOFF	0 = use INTRANUKE
	1 = turn INTRANUKE off
INTR	mask for specific interaction types:
	1 = turn off elastic / quasi-elastic
	2 = turn off resonance production
	3 = turn off DIS
	4 = turn off coherent production

3. `total_xsec(nutype,ia,iz,e,val)` where:

Call Parameter	Type	Description
nutype	int	PDG code for neutrino
ia	int	A for nucleus
iz	int	Z for nucleus
e	real	energy in GeV
val	real	cross section in $(10^{-38} cm^2)$

This subroutine returns the total cross section (`val`) for the specified neutrino, energy, and nucleus. Note that the calculation for the cross section is modified by the user masks and switches called in gen_control.

3.2. Useful Include Files

In addition to the HEPEVT common, NEUGEN keeps track of information related to the event generation in several different common blocks. This information, which can in principle also be extracted from the HEPEVT record, gives a useful summary of the details of the produced event.

1. `fkinematics.inc`: This include file contains a common block which is used to store the kinematic variables selected for the event (x,y,W,q^2,t). t is the four-momentum transfer to the nucleus and is meaningful for coherent events only.

```
real s,x_int,y_int,w_int,qqs_int, t_int
common /FKINEMATICS/ s, x_int, y_int, w_int, qqs_int, t_int
```

2. `icode.inc`: This include file summarizes information about the interaction type. `nutype` and `inuc` are the neutrino and nucleus type requested by the user (in Soudan 2 particle codes), `int_type` is the interaction type: (1) Elastic / Quasi-Elastic, (2) Resonance Production (3) Deep Inelastic Scattering (4) Coherent Pion Production, `iccnc` is (1) CC scattering (2) NC scattering, `ifc` is (1) = no flavor change (2) = flavor changing interaction, `im` (1) Interaction is $\nu + p$, (2) Interation is $\nu + n$, (3) Interaction is $\overline{\nu} + p$, (4) Interaction is $\overline{\nu} + n$.

```
      integer icode(6)
      integer nutype,inuc,int_type,iccnc,ifc,im
      equivalence (icode(1),nutype),(icode(2),inuc),
     +  (icode(3),int_type),(icode(4),iccnc),(icode(5),ifc),
     +  (icode(6),im)
      common /icode/icode
```

3. `stdptrs.inc`: This include file contains a common block that holds pointers to the most commonly accessed particles in the HEPEVT common block. `ip_neutrino` is the incoming neutrino. `ip_nucleus` is the target nucleus. `ip_target` is the nucleon within the nucleus that will be struck. `ip_remnant_nucleus` is the remnant nucleus. `ip_boson` is not set at present... `ip_lepton` is the outgoing lepton. `ip_struck` is the outgoing hadronic system.

```
       integer ip_neutrino, ip_nucleus, ip_target, ip_remnant_nucleus,
     +         ip_boson, ip_lepton, ip_struck
       common /STDPTRS/ ip_neutrino, ip_nucleus, ip_target,
     +   ip_remnant_nucleus, ip_boson, ip_lepton, ip_struck
```

4. Particle and Status Codes

4.1. Particle Codes

Particle codes input and output from NEUGEN are standard PDG codes, with a few exceptions. Internally the program uses its own particle codes (inherited from Soudan 2 and listed in `parcodes.inc`). Some intermediate state particles and nuclei do not have properly defined PDG codes. Intermediate states without a PDG counterpart are assigned as $10^7 + ID$, where ID is the Soudan 2 particle code, nuclear states are defined as $10^9 + A * 10^6 + Z * 10^3$.

A further wrinkle on the conventional PDG naming scheme is the inclusion of a 'ghost' particle to account for missing momentum and energy. Whenever a physical process in NEUGEN would result in lost energy or momentum, a placeholder particle is created which carries the missing E,p. In this way energy and momentum are conserved throughout the event generation process, a constraint that is quite helpful in identifying bugs in the program when they creep in. These particles are stored in the HEPEVT output with particle code 28. This particle code corresponds to a 'GEANTINO' in GEANT3, a completely non-interacting particle.

4.2. Status Codes

NEUGEN defines a few additional status codes in additon to the standard STDHEP ones. Status codes greater than 10 are NEUGEN-specific. As always, if the user is interested in stable final state particles they can select on those with ISTATUS = 1.

Status Code	Meaning
0	Initial State
1	Stable Final State
2	Intermediate State
3	Decayed State
11	Nucleon Target
12	DIS hadronic state (pre-fragmenation)
13	resonant state (pre-decay)
14	hadron in the nucleus

5. Example

Included in the tar file is a short program called `neugen_example` which shows how NEUGEN can be used. This particular program generated 10k events for a 3 GeV ν_μ neutrino on iron. The first 3 events are written out to standard output (through a call to the STDHEP routine heplst..), and several histograms are filled.

```
       program neugen_example
       implicit none
       real e_nu

c-- hbook variables
       integer nwpawc,istat,icycle,i1
       parameter (nwpawc=1000000)
       common /pawc/ h(nwpawc)
       real h

       integer i,j,id,ist
       real epi,costh,pfermi

       include 'stdhep.inc'
       include 'stdptrs.inc'

       e_nu = 3.
       call hlimit(nwpawc)

       call hropen(32,'tdir','example.dat','N',1024,istat)
       call hbook1(100,'pi energy',100,0.,e_nu,0)
       call hbook1(101,'cos lep',100,-1.,1.,0)
       call hbook1(102,'pfermi',100,0.,0.400,0)

       do i=1,1000
         if(mod(i,100).eq.0)  write(6,*) i

         call generate_nu_event(14,e_nu,56,26)

c  print out the first 3 events
         if(i.le.3) call heplst(1)
```

```
c  costh is the cosine of the angle between the lepton and neutrino
         costh = SNGL ( phep(3,ip_lepton) /
     +     sqrt( phep(1,ip_lepton)**2 + phep(2,ip_lepton)**2 +
     +           phep(3,ip_lepton)**2))

c  pfermi is the momentum of the struck nucleon
         pfermi = SNGL ( sqrt ( phep(1,ip_target)**2 +
     +     phep(2,ip_target)**2 + phep(3,ip_target)**2) )

c loop over all particles in the event
         do j=1,nhep
           id = idhep(j)
           ist = isthep(j)

c update histogram with particle energy for stable pions
           if((abs(id).eq.211.or.id.eq.111).and.
     +         ist.eq.1) then
             epi = SNGL(phep(4,j))
           endif

           call hf1(100,epi,1.)
         end do
         call hf1(101,costh,1.)
         call hf1(102,pfermi,1.)
       end do

       call hrout(0,icycle,' ' )
       call hrend('tdir')

       end
```

The output of this program consists of the HBOOK file `example.dat` which includes histograms of pion energies, lepton angle with respect to the beam direction, and fermi momentum of the struck nucleon. During execution, ascii summaries of the first 3 events are sent to stdout. The output may look like the following. The first event is DIS, producing 3 final state hadrons, (a neutron and 2 π^+). One of the pions inelastically scatters before getting out of the nucleus, losing energy, while the other emerges unscathed. The second event is a quasi-elastic CC event and the third is resonant production of a $\Delta(1232)$ which decays to a $p + \pi^+$. The π^+ elastically scatters before getting out of the nucleus.

```
                 Event listing (HEP format)           Event:        1

  I  particle/jet      ISTHEP    IDHEP     JMOHEP       JDAHEP     PHEP(1,I)   PHEP(2,I)   PHEP(3,I)   PHEP(4,I)   PHEP(5,I)
  1  !nu_mu!                0       14     0     0     5     5     0.00000     0.00000     3.00000     3.00000     0.00000
  2  !!                     0 ********     0     0     3     4     0.00000     0.00000     0.00000    52.10349    52.10349
  3  !p+!                  11     2212     2     2     6     6     0.10171     0.09142    -0.14342     0.92822     0.90682
  4                         1 ********     2     2     0     0    -0.10171    -0.09142     0.14342    51.17527    51.17488
  5  mu-                    1       13     1     1     0     0     0.30521    -0.34269     1.61779     1.68493     0.10566
  6  !omega(1420)!          3   100223     3     3     7     9    -0.20350     0.43411     1.23879     2.24329     1.80772
  7  !pi+!                 14      211     6     6    10    11    -0.00029    -0.25010     0.39977     0.49177     0.13957
  8  !pi+!                 14      211     6     6    12    12    -0.15758     0.38333    -0.02673     0.43815     0.13957
  9  n0                     1     2112     6     6     0     0    -0.04563     0.30087     0.86576     1.31337     0.93957
 10  pi+                    1      211     7     7     0     0     0.08393     0.11054    -0.11846     0.22973     0.13957
 11                         1       28     7     7     0     0    -0.08422    -0.36064     0.51823     0.26204     0.00000
```

```
  12  pi+                  1       211      8      8      0      0    -0.15758     0.38333    -0.02673     0.43815     0.13957
```

```
              Event listing (HEP format)           Event:        2

   I  particle/jet     ISTHEP     IDHEP     JMOHEP        JDAHEP      PHEP(1,I)   PHEP(2,I)   PHEP(3,I)   PHEP(4,I)   PHEP(5,I)
   1  !nu_mu!              0        14      0      0      5      5     0.00000     0.00000     3.00000     3.00000     0.00000
   2  !!                   0 ********      0      0      3      4     0.00000     0.00000     0.00000    52.10349    52.10349
   3  !n0!                11      2112      2      2      6      6    -0.13656    -0.20391     0.25496     0.93071     0.86080
   4                       1 ********      2      2      0      0     0.13656     0.20391    -0.25496    51.17278    51.17156
   5  mu-                  1        13      1      1      0      0     0.64174    -0.40972     2.10998     2.24564     0.10566
   6  !p+!                 3      2212      3      3      7      8    -0.77830     0.20581     1.14498     1.68507     0.93828
   7  p+                   1      2212      6      6      0      0    -0.75572     0.19984     1.11177     1.65150     0.93828
   8                       1        28      6      6      0      0    -0.02258     0.00597     0.03322     0.03357     0.00000
```

```
              Event listing (HEP format)           Event:        3

   I  particle/jet     ISTHEP     IDHEP     JMOHEP        JDAHEP      PHEP(1,I)   PHEP(2,I)   PHEP(3,I)   PHEP(4,I)   PHEP(5,I)
   1  !nu_mu!              0        14      0      0      5      5     0.00000     0.00000     3.00000     3.00000     0.00000
   2  !!                   0 ********      0      0      3      4     0.00000     0.00000     0.00000    52.10349    52.10349
   3  !p+!                11      2212      2      2      6      6    -0.01130    -0.23226     0.02767     0.92807     0.89804
   4                       1 ********      2      2      0      0     0.01130     0.23226    -0.02767    51.17542    51.17488
   5  mu-                  1        13      1      1      0      0    -0.10769     0.60759     2.10954     2.20047     0.10566
   6  !Delta++!            3      2224      3      3      7      8     0.09639    -0.83985     0.91813     1.72760     1.19457
   7  p+                   1      2212      6      6      0      0     0.13968    -0.76691     0.57709     1.34947     0.93828
   8  !pi+!               14       211      6      6      9     10    -0.04328    -0.07294     0.34104     0.37813     0.13957
   9  pi+                  1       211      8      8      0      0    -0.04415    -0.05788     0.34380     0.37813     0.13957
  10                       1        28      8      8      0      0     0.00087    -0.01506    -0.00276     0.00000     0.00000
```

REFERENCES