U.S. DEPARTMENT OF
ENERGY

PNNL-*draft*

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Technical Assistance for Brillouin Energy Corporation Concept of Low Energy Nuclear Weak Capture from Induced Confinement

RS Wittman

February 2013

Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by Battelle *Since 1965*

OFFICIAL USE ONLY

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights**. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Low Energy Nuclear Weak Capture from Induced Confinement

R.S. Wittman
Pacific Northwest National Laboratory Richland, WA 99352
TAP work performed for:
Robert Godes (CTO) *Brillouin Energy Corporation (BEC)*
http://www.brillouinenergy.com

1 Introduction

As part of an effort to understand excess energy produced in experiments conduced by Brillouin Energy Corporation (BEC), PNNL was contacted through the Technology Assistance Program (TAP) to model the possible contribution of excess energy from low energy nuclear reactions (LENR) – specifically lattice induced electron capture by protons (Fig. 1). The BEC experiments roughly consists of high (kHz) frequency current pulses through a hydrogen packed metal with an average input power of about 100 W. BEC believes that under certain conditions the current pulses collectively perturb the lattice to transfer energy to localized electronic states on the scale (780 keV) required to enable electron capture. The calculations reported here focus on the basic reaction of electron and proton capture by protons. The weak interaction physics is treated relativistically with the standard $V - A$ interaction. The specifics of the lattice are simply represented with an initial *e-p* state where confinement to a compressed lattice region is assumed to be externally induced. While this work provides basic tools for calculating electron capture rates, future work could develop a more realistic representation of the lattice using the lattice Hamiltonian discussed Ref. [1].

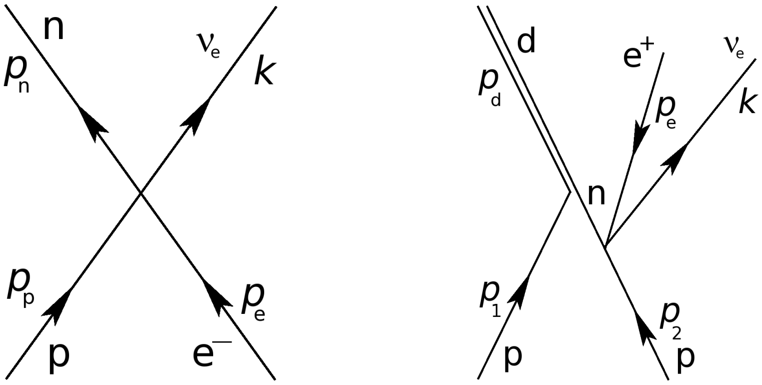

Figure 1-2. LENR elementary process for 1) electron capture and 2) and proton capture on protons. (labels are given for particle type and external four-momentum variables)

Computer programs were written for calculating the processes for both Figure 1 and Figure 2, but calculations are reported only for Figure 1. For the electron capture of Figure 1 in the center-of-momentum (c.m.) frame, the threshold kinetic energy of the electron is given by:

$$K_e = E_e - m_e = \frac{\left(M_n - m_e + m_\nu\right)^2 - M_p^2}{2\left(M_n + m_\nu\right)} \tag{1}$$

which is approximately 781.6 keV (or 783.1 keV in proton rest frame). Of course the proton rest frame is approximately the c.m. frame, but it should be mentioned that in the electron rest frame, the threshold kinetic energy required by a proton is approximately 2000X greater. Therefore, it is assumed that the energy transfer to produce a capture is to the c.m. frame *e-p* pair and represented by a confinement scale induced by the metallic lattice deformation. For the final state, the c.m. frame kinematics give the neutrino kinetic energy of:

$$K_\nu = k^0 - m_\nu = \frac{\left(E_p + E_e - m_\nu\right)^2 - M_n^2}{2\left(E_p + E_e\right)} \tag{2}$$

and a neutron kinetic energy of:

$$K_n = E_n - M_n = \frac{\left(E_p + E_e - M_n\right)^2 - m_\nu^2}{2\left(E_p + E_e\right)} \tag{3}$$

in terms of the total c.m. energy $E_p + E_e$. (Note: In units of $c = 1$, mass is assumed to have units of energy, *e.g.* $m_e \approx 511$ keV)

In the next section the basis for the elementary interactions is given; section 3 describes the rate calculations, and section 4 reports the results for neutron production and energy production rates.

2 Effective Lagrangian

The amplitudes of Figures 1 and 2 can be derived from the effective Lagrangian:[2–4]

$$\mathcal{L}_I = \frac{G_F}{\sqrt{2}}\overline{\psi}_p\left(G_V - G_A\gamma^5\right)\gamma^\mu\psi_n\overline{\psi}_e\gamma_\mu\left(1-\gamma^5\right)\psi_\nu + ig_{npd}\left(\overline{\psi}^c_p\gamma^\mu\psi_n - \overline{\psi}^c_n\gamma^\mu\psi_p\right)D^\dagger_\mu + h.c. \tag{4}$$

where the G_F is the Fermi coupling constant and the unit convention of $\hbar = c = 1$ is assumed. Charge conjugation is implied by $\overline{\psi}^c$ and D_μ is the effective deuteron field.

3 Reaction Observables

Based on the lowest order process [Figs. (1)], the cross section for $e^- + p \rightarrow n + \nu_e$ is given by:[5–7]

$$d\sigma_{ep} = \left(\frac{1}{2\pi}\right)^2 \frac{\overline{\sum_{fi}} \left|\mathcal{M}_{fi}\right|^2}{16\left|\mathbf{k}\cdot\left(E_n\mathbf{k} - k^0\mathbf{p}_n\right)\right|} \frac{k^3 p_e d\Omega_k}{\left|\mathbf{p}_e\cdot\left(E_p\mathbf{p}_e - E_e\mathbf{p}_p\right)\right|} \ . \tag{5}$$

The capture rate for a confined *e-p* pair is given according to:

$$d\lambda_{ep} = \left(\frac{1}{2\pi}\right)^2 \frac{\overline{\sum_{fi}} \left|\mathcal{M}_{fi}\left[\mathbf{p}_{ep} \to -i\nabla\right] \psi_{ep}\left(\mathbf{x}\right)\Big|_{\mathbf{x}=\mathbf{0}}\right|^2}{16E_pE_e\left|\mathbf{k}\cdot\left(E_n\mathbf{k} - k^0\mathbf{p}_n\right)\right|} k^3 d\Omega_k \tag{6}$$

where in the c.m. frame the three-momentum of the electron and proton are $\mathbf{p}_e = \mathbf{p}_{ep}$ and $\mathbf{p}_p = -\mathbf{p}_{ep}$, and $\psi_{ep}(\mathbf{x})$ is the wave function for the confined electron-proton initial state. For simplicity, the wave function is assumed to be that of the lowest energy for a confined cubic box with side length L

$$\psi_{ep}(\mathbf{x}) = \left(\frac{2}{L}\right)^{\frac{3}{2}} \cos\frac{\pi x}{L} \cos\frac{\pi y}{L} \cos\frac{\pi z}{L} \ . \tag{7}$$

The length L is presumably the lattice scale at a specified compression. For consistency with particle production processes, all kinematics and energies are treated relativistically (*e.g.* $E^2 = \mathbf{p}^2 + M^2$). For the confined electon and proton, the Eq. (7) wave function gives

$$E_e^2 = m_e^2 + 3\left(\frac{\pi}{L}\right)^2 \text{ and } E_p^2 = M_p^2 + 3\left(\frac{\pi}{L}\right)^2 \ . \tag{8}$$

The invariant matrix element of Eqs. (5&6) is

$$\mathcal{M}_{fi} = \frac{G_F}{\sqrt{2}} \overline{u}(p_n, s_n)\left(G_V - G_A\gamma^5\right)\gamma^\mu u(p_p, s_p)\overline{u}(k, s_k)\gamma_\mu\left(1-\gamma^5\right)u(p_e, s_e) \ , \tag{9}$$

where $G_F = 1.16637 \times 10^{-5}$ GeV^{-2} and the neutron lifetime is reproduced with $G_A = 1$ and $G_v = -1.285$.[7] The Dirac spinor convention is that of Itzykson and Zuber.[6]

The cross section for p + p → d + e$^+$ + ν_e (Fig. 2) is given by:

$$d\sigma_{pp} = \left(\frac{1}{2\pi}\right)^5 \frac{\overline{\sum_{fi}} \left|\mathcal{M}_{fi}\right|^2}{32E_e\left|\mathbf{p}_1\cdot\left(E_2\mathbf{p}_1 - E_1\mathbf{p}_2\right)\right|} \frac{k^3 p_1 d^3p_e d\Omega_k}{\left|\mathbf{k}\cdot\left(E_d\mathbf{k} - k^0\mathbf{p}_d\right)\right|} \tag{10}$$

and the capture rate for *p-p* pair is given according to:

$$d\lambda_{pp} = \left(\frac{1}{2\pi}\right)^5 \frac{\overline{\sum_{fi}} \left|\mathcal{M}_{fi}\left[\mathbf{p}_{pp} \to -i\nabla\right] \psi_{pp}\left(\mathbf{x}\right)\Big|_{\mathbf{x}=\mathbf{0}}\right|^2}{32E_eE_1E_2\left|\mathbf{k}\cdot\left(E_d\mathbf{k} - k^0\mathbf{p}_d\right)\right|} k^3 d\Omega_k d^3p_e \tag{11}$$

where in the p-p c.m. frame the three-momentum of the protons are $\mathbf{p}_1 = \mathbf{p}_{pp}$ and $\mathbf{p}_2 = -\mathbf{p}_{pp}$. and $\psi_{pp}(\mathbf{x})$ is the wave function for the confined p-p initial state. The invariant matrix element of Eqs. (10&11) is

$$\mathcal{M}_{fi} = \frac{G_F}{\sqrt{2}} g_{npd} \overline{v}(p_1, s_1) \not{\$}_d \frac{\not{p}_d - \not{p}_1 + M_n}{(p_d - p_1)^2 - M_N^2} \left(G_V - G_A \gamma^5\right) \gamma^\mu u(p_2, s_2)$$
$$\times\, \overline{u}(k, s_k) \gamma_\mu \left(1 - \gamma^5\right) v(p_e, s_e) - (1 \leftrightarrow 2)\ , \quad (12)$$

where $g_{npd} = 10.662$ reproduces the $p(n,\gamma)d$ cross section.

4 Results for Electron Capture

Given an assumed induced confinement scale according to Eqs. (7&8), Figure 3 shows the average energetics in a material such as a Ni-H solution with a density of about 7 g/cc. The electron capture threshold occurs at about $L = 0.009$ Å. Effectively, the black curve is the power consumed by electron capture and the red curve is the energy produced by a subsequent neutron capture by a proton (assuming that a fraction of 10^{-7} lattice sites are continuously under compression) – the orange curve is the power ratio.

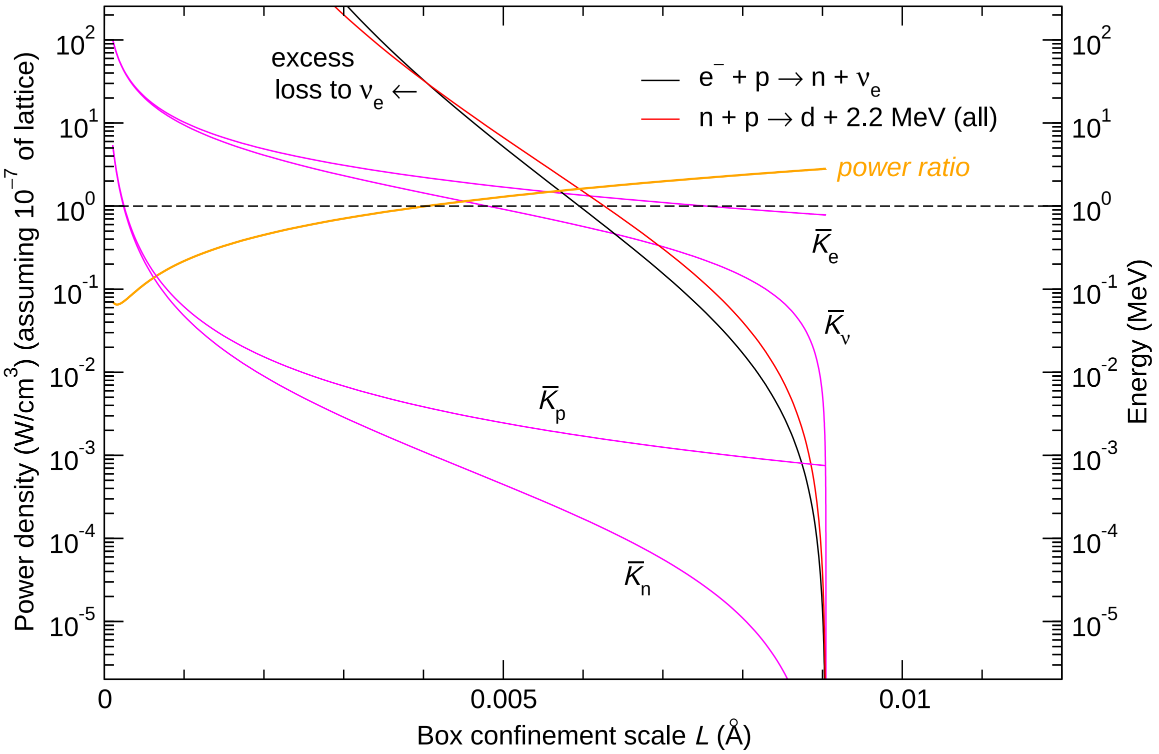

Figure 3. Electron capture energy balance as a function of confinement scale.

The power ratio is positive for L between 0.004 and 0.009 Åwith a maximum of almost a factor of 3. Of course, increases of this ratio result from considering more energetic neutron capture reactions. Additionally, the ratio is decreased by losses of the lattice energy to processes other than confinement. Both of these considerations will have to be investigated to give quantitative guidance for BEC experiments. As a first step, this work provides the tool to calculate the induced electron capture rate (see program listing at end of report) given the confinement scale. Further work is necessary to predict how the input energy controls the lattice dynamics and how the confinement scales assumed here are achieved.

References

[1] R.E. Godes, *The Quantum Reaction Hypothesis*, http://www.brillouinenergy.com/docs.php?doc=energy_hypothesis (2011).

[2] R. P. Feynman and M. Gell-Mann, *Theory of the Fermi Interaction*, Phys. Rev. **109**, 193 (1958);

[3] E.C.G. Sudarshan and R.E. Marshak, *Chirality Invariance and the Universal Fermi Interaction*, Phys. Rev. 109, 1860 (1958).

[4] A.N. Ivanov, H. Oberhummer, N.I. Troitskaya and M. Faber, *The Nambu–Jona–Lasinio model of light nuclei*, European Physical Journal, A7, 519 (2000).

[5] J.D. Bjorken and S.D. Drell, *Relativistic Quantum Mechanics* (McGraw-Hill Book Company, New York, 1964).

[6] C. Itzykson, J. Zuber, *Quantum Field Theory* (McGraw-Hill Book Company, New York, 1985).

[7] J. Beringer, *et al.*, (Particle Data Group), *Review of Particle Properties*, Phys. Rev. D **86**, 010001 (2012).

PROGRAM LISTING for: pe-REALbox-rate-pow.f

```
      implicit real*8 (a-b,d-h,o-z)
      implicit complex*16 (c)
      dimension ca(4,4),cb(4,4),cgamma(0:3,4,4),
     1 cgamma5(4,4),gmunu(0:3,0:3),cu1bar(4),cu2(4),cukbar(4),cue(4),
     2 p1(0:3),p2(0:3),pk(0:3),cs1(2),cs2(2),csk(2),
     3  cleft(4,4),cGVGA(4,4),xg(100),wxg(100),pg(100),wpg(100),
     4 pe(0:3),cse(2)
      common /dirac/ ci,cgamma,cgamma5,gmunu
      common /con/   pi,hbarc,rc,alpha,GF,GV,GA,dMp,dMn,dme,dmnu,dMd,gnpd
     1                   ,dkapp,dkapn,dufac,dN0,e

      call const
      call cdirac
      nx = 8
      call gaus(xg,wxg,nx)
      np = 8
      call gaus(pg,wpg,np)

      do ip=1,np
         pg(ip)  = (pg(ip)+1.d0)*pi
         wpg(ip) =        wpg(ip)*pi
      enddo

      Ekemin = ((dMn + dmnu - dme)**2 - dMp**2)/(2.d0*(dMn+dmnu))
      pmin = dsqrt((Ekemin+dme)**2 - dme**2)

      E2check = ((dMn + dmnu)**2 - (dMp+dme)**2)/(2.d0*dMp)

      write(*,*) Ekemin,pmin,E2check

      call copyg(cgamma5,ca)
      call chsgng(ca)

      cscal = 1.d0
      call saddg(cscal,ca,cleft)

      call copyg(cgamma5,ca)
      cscal = GA
      call smultg(cscal,ca,ca)
      call chsgng(ca)

      cscal = GV
      call saddg(cscal,ca,cGVGA)
      do is1 =1,2
          cs1(is1) = 0.d0
      enddo
      do is2 =1,2
          cs2(is2) = 0.d0
      enddo
      do ise =1,2
          cse(ise) = 0.d0
      enddo
      do isk =1,2
          csk(isk) = 0.d0
      enddo

      tran2 =  0.d0
c      GF = 1.d0
c      gndp = 1.d0
c      hbarc = 1.d0
       densep = 7.d0    ! g/cc   NiH density ??
       dNiMW  = 28.d0   ! g/mol  NiH MW ??
c   Number density of H in 1/(cm)^3
       dndens =  dN0*densep/dNiMW
c   Number density of H in (MeV)^3
       rhoep = ((hbarc*1.d-13)**3)*dndens
       Vwire = pi*((0.031d0/2.d0)**2)*5.d0 ! cm^3
       protN = Vwire*dndens
       write(*,*) densep,dndens,rhoep,protN
       write(*,”(1x)”)

      open(9,file='plot.dat',status=”UNKNOWN”)




      do iep=1,500

      pep = ( pmin + dexp(dlog(1.d-11)+dfloat(iep)*
     1          (dlog(100.d0)-dlog(1.d-11))/500.d0) )/dsqrt(3.d0)
      rlconf  =  1.d-5*hbarc*pi/pep

      test = 0.d0
      tran2 = 0.d0

      do ix=1,nx
      do ip=1,np

      do is1 =1,2
          cs1(is1) = 1.d0
      do is2 =1,2
          cs2(is2) = 1.d0
      do ise =1,2
          cse(ise) = 1.d0
      do isk =1,2
          csk(isk) = 1.d0

      do isgn1 = 1,2
      do isgn2 = 1,2
      do isgn3 = 1,2

      camp2 = 0.d0

      pep1 = pep*((-1.d0)**isgn1)
      pep2 = pep*((-1.d0)**isgn2)
      pep3 = pep*((-1.d0)**isgn3)

      call pekin(xg(ix),pg(ip),pep1,pep2,pep3,pe,p1,p2,pk)

          call spinubar(p1,dMn,cs1,cu1bar)
          call spinu(p2,dMp,cs2,cu2)
          call spinu(pe,dme,cse,cue)
          call spinubar(pk,dmnu,csk,cukbar)


      call vcopyg(0,cgamma,cb)
      call multg(cGVGA,cb,cb)
      call ubaru(cu1bar,cb,cu2,chad0)
      call vcopyg(0,cgamma,cb)
      call multg(cb,cleft,cb)
      call ubaru(cukbar,cb,cue,clep0)

c       write(*,*) clep0

      call vcopyg(1,cgamma,cb)
      call multg(cGVGA,cb,cb)
      call ubaru(cu1bar,cb,cu2,chad1)
      call vcopyg(1,cgamma,cb)
      call multg(cb,cleft,cb)
      call ubaru(cukbar,cb,cue,clep1)

      call vcopyg(2,cgamma,cb)
```

```
        call multg(cGVGA,cb,cb)
        call ubaru(cu1bar,cb,cu2,chad2)
        call vcopyg(2,cgamma,cb)
        call multg(cb,cleft,cb)
        call ubaru(cukbar,cb,cue,clep2)

        call vcopyg(3,cgamma,cb)
        call multg(cGVGA,cb,cb)
        call ubaru(cu1bar,cb,cu2,chad3)
        call vcopyg(3,cgamma,cb)
        call multg(cb,cleft,cb)
        call ubaru(cukbar,cb,cue,clep3)

        camp2 = camp2 +
     1         (chad0*clep0 - chad1*clep1 - chad2*clep2 - chad3*clep3)

        enddo
        enddo
        enddo

        fact =rc*((pep/(2.d0*pi))**3)*(GF**2)*(dsqrt(pk(0)**2-dmnu**2))**3
     1        /(512.d0*(pi**2)*p2(0)*pe(0)*hbarc*1.d-15*
     1  dabs((pk(1)*(p1(0)*pk(1)-pk(0)*p1(1))) +
     1         (pk(2)*(p1(0)*pk(2)-pk(0)*p1(2))) +
     1         (pk(3)*(p1(0)*pk(3)-pk(0)*p1(3)))))

        EKn = p1(0) - dMn
        EKe = pe(0) - dme

          tran2 = tran2 + wxg(ix)*wpg(ip)*
     1    dreal(camp2*dconjg(camp2))*fact*dndens
c       4       *((pe(0) - dme)+(p2(0) - dMp))*e   ! W power input
c       4     *(2.2d0+EKn)*e     ! W power output gamma
c         test =test + 2.d0*pi*wxg(ix)*wpep*(fbox(pep,rl)**2)*(pep**2)/16.d0


             csk(isk) = 0.d0
        enddo
             cse(ise) = 0.d0
        enddo
             cs2(is2) = 0.d0
        enddo
             cs1(is1) = 0.d0
        enddo
```

```
      enddo
      enddo

      EKp   = p2(0) - dMp
      EKn   = p1(0) - dMn
      EKe   = pe(0) - dme
      EKnu = pk(0) - dmnu

      write(*,*) rlconf,tran2,EKe,EKn
      write(9,"(10(1pe14.6))") rlconf,
     1   tran2*((pe(0) - dme)+(p2(0) - dMp))*e*1.d-7,
     1   tran2*(2.2d0+(p1(0) - dMn))*e*1.d-7,EKe,EKn,EKnu,Ekp
      enddo

      close(9)
      STOP
      end

      subroutine const
      implicit real*8 (a-h,o-z)
      common /con/   pi,hbarc,rc,alpha,GF,GV,GA,dMp,dMn,dme,dmnu,dMd,gnpd
     1                       ,dkapp,dkapn,dufac,dN0,e

      pi = 4.d0*datan(1.d0)
      hbarc = 197.3269631d0       ! MeV-fm
      rc    = 299792458.d0        ! m/s
      alpha = 7.2973525376d-3
      e     = 1.602176487d-13    ! J/MeV

      dN0  =    6.02214179d23      ! mol^-1
      dMp  =  938.272013d0         ! MeV
      dMn  =  939.565346d0         ! MeV
      dme  =    0.510998910d0      ! MeV
      dmnu =    1.d-8              ! MeV
      dMd  = 1875.612793d0         ! MeV
      GF   = 1.16637d-11           ! 1/MeV^2
      GV   =  1.d0 ! 1.013d0
      GA   = -1.285d0*GV           ! -1.267
      gnpd = 10.6617553d0          ! 11.3 in 9704031v5.pdf (pg 8-9)
      dkapp=  1.793d0
      dkapn= -1.913d0
      dufac=  0.d0

      RETURN
      end
      subroutine spin1(i,p,dM,s)
      implicit real*8 (a-h,o-z)
      dimension p(0:3),s(0:3)

      s(0) = p(i)/dM
      do j=1,3
         s(j) = p(j)*p(i)/(dM*(dM+p(0)))
      enddo

      s(i) = s(i) + 1.d0

      RETURN
      end

      subroutine spin1g(p,s)
      implicit real*8 (a-h,o-z)
      dimension p(0:3),s(0:3)

      do j=0,3
         s(j) = p(j)/p(0)
      enddo

      RETURN
      end

      subroutine pekin(x,phi,pep1,pep2,pep3,pe,p1,p2,pk)
      implicit real*8 (a-h,o-z)
      dimension p1(0:3),p2(0:3),pk(0:3),pe(0:3)
      common /con/ pi,hbarc,rc,alpha,GF,GV,GA,dMp,dMn,dme,dmnu,dMd,gnpd
     1                 ,dkapp,dkapn,dufac,dN0,e

c        write(*,*) EKe,x,prel(2)

      p2(0) = dsqrt(dMp**2 + pep1**2 + pep2**2 + pep3**2)
      p2(1) = -pep1
      p2(2) = -pep2
      p2(3) = -pep3

      pe(0) = dsqrt(dme**2 + pep1**2 + pep2**2 + pep3**2)
      pe(1) =  pep1
      pe(2) =  pep2
      pe(3) =  pep3

      Etot = p2(0) + pe(0)

      Eknu = ((Etot-dmnu)**2 - dMn**2)/(2.d0*Etot)
      Ekn  = ((Etot-dMn)**2 - dmnu**2)/(2.d0*Etot)
      pnu = dsqrt((Eknu+dmnu)**2 - dmnu**2)

      pk(0) = Eknu + dmnu
      pk(1) = pnu*dsqrt(1.d0 - x**2)*dcos(phi)
      pk(2) = pnu*dsqrt(1.d0 - x**2)*dsin(phi)
      pk(3) = pnu*x

      p1(0) = p2(0) + pe(0) - pk(0)
      p1(1) = p2(1) + pe(1) - pk(1)
      p1(2) = p2(2) + pe(2) - pk(2)
      p1(3) = p2(3) + pe(3) - pk(3)


c       write(*,*) pnu,pcheck,Eknu,Ekn

      RETURN
      end

      subroutine multg(ca,cb,cc)
      implicit real*8 (a-b,d-h,o-z)
      implicit complex*16 (c)
      dimension ca(4,4),cb(4,4),cc(4,4),cd(4,4)

      do i1=1,4
      do i2=1,4
         cd(i1,i2) = (0.d0,0.d0)
         do j=1,4
            cd(i1,i2) = cd(i1,i2) + ca(i1,j)*cb(j,i2)
         enddo
      enddo
      enddo

      do i1=1,4
      do i2=1,4
         cc(i1,i2) = cd(i1,i2)
      enddo
      enddo

      RETURN
      end

      subroutine smultg(cscal,ca,cb)
      implicit real*8 (a-b,d-h,o-z)
      implicit complex*16 (c)
      dimension ca(4,4),cb(4,4)

      do i1=1,4
      do i2=1,4
              cb(i1,i2) = ca(i1,i2)*cscal
      enddo
      enddo

      RETURN
      end

      subroutine saddg(cscal,ca,cb)
      implicit real*8 (a-b,d-h,o-z)
      implicit complex*16 (c)
      dimension ca(4,4),cb(4,4)

      do i1=1,4
      do i2=1,4
              cb(i1,i2) = ca(i1,i2)
      enddo
      enddo

      do i1=1,4
              cb(i1,i1) = cb(i1,i1) + cscal
      enddo

      RETURN
      end

      subroutine addg(ca,cb,cc)
      implicit real*8 (a-b,d-h,o-z)
      implicit complex*16 (c)
      dimension ca(4,4),cb(4,4),cc(4,4)

      do i1=1,4
      do i2=1,4
              cc(i1,i2) = ca(i1,i2) + cb(i1,i2)
      enddo
      enddo

      RETURN
      end
      subroutine subg(ca,cb,cc)
      implicit real*8 (a-b,d-h,o-z)
      implicit complex*16 (c)
      dimension ca(4,4),cb(4,4),cc(4,4)

      do i1=1,4
      do i2=1,4
              cc(i1,i2) = ca(i1,i2) - cb(i1,i2)
      enddo
      enddo

      RETURN
      end

      subroutine copyg(ca,cb)
      implicit real*8 (a-b,d-h,o-z)
      implicit complex*16 (c)
      dimension ca(4,4),cb(4,4)

      do i1=1,4
      do i2=1,4
              cb(i1,i2) = ca(i1,i2)
      enddo
      enddo

      RETURN
      end

      subroutine vcopyg(mu,ca,cb)
      implicit real*8 (a-b,d-h,o-z)
      implicit complex*16 (c)
      dimension ca(0:3,4,4),cb(4,4)

      do i1=1,4
      do i2=1,4
              cb(i1,i2) = ca(mu,i1,i2)
      enddo
      enddo

      RETURN
      end

      subroutine chsgng(ca)
      implicit real*8 (a-b,d-h,o-z)
      implicit complex*16 (c)
      dimension ca(4,4)

      do i1=1,4
      do i2=1,4
              ca(i1,i2) = -ca(i1,i2)
      enddo
      enddo

      RETURN
      end

      subroutine cdirac
      implicit real*8 (a-b,d-h,o-z)
      implicit complex*16 (c)
      dimension cgamma(0:3,4,4),cgamma5(4,4),gmunu(0:3,0:3)
      common /dirac/ ci,cgamma,cgamma5,gmunu
      ci = (0.d0,1.d0)

      do i1=1,4
      do i2=1,4
        cgamma5(i1,i2) = (0.d0,0.d0)
        do imu = 0,3
           cgamma(imu,i1,i2) = (0.d0,0.d0)
        enddo
      enddo
      enddo

      cgamma(0,1,1) =  1.d0
      cgamma(0,2,2) =  1.d0
      cgamma(0,3,3) = -1.d0
      cgamma(0,4,4) = -1.d0

      cgamma(1,1,4) =  1.d0
      cgamma(1,2,3) =  1.d0
      cgamma(1,3,2) = -1.d0
      cgamma(1,4,1) = -1.d0

      cgamma(2,1,4) = -ci
      cgamma(2,2,3) =  ci
      cgamma(2,3,2) =  ci
      cgamma(2,4,1) = -ci

      cgamma(3,1,3) =  1.d0
      cgamma(3,2,4) = -1.d0
      cgamma(3,3,1) = -1.d0
      cgamma(3,4,2) =  1.d0

      cgamma5(1,3) =  1.d0
      cgamma5(2,4) =  1.d0
      cgamma5(3,1) =  1.d0
      cgamma5(4,2) =  1.d0

      RETURN
      end

      subroutine spinu(p,rm,cs,cu)
      implicit real*8 (a-b,d-h,o-z)
      implicit complex*16 (c)
      dimension cgamma(0:3,4,4),cgamma5(4,4),cwu(4,2),
     1     cu(4),gmunu(0:3,0:3),
     2     cs(2),cphi(2),p(0:3)
      common /dirac/ ci,cgamma,cgamma5,gmunu

      fac = dsqrt(p(0) + rm)

      cphi(1) =  cs(1)*fac
      cphi(2) =  cs(2)*fac

      cwu(1,1) =   1.d0
      cwu(1,2) =   0.d0
      cwu(2,1) =   0.d0
      cwu(2,2) =   1.d0
      cwu(3,1) = ( p(3)          )/(p(0) + rm)
      cwu(3,2) = ( p(1)-ci*p(2)  )/(p(0) + rm)
      cwu(4,1) = ( p(1)+ci*p(2)  )/(p(0) + rm)
      cwu(4,2) = (-p(3)          )/(p(0) + rm)

      do i1=1,4
        cu(i1) = 0.d0
      do i2=1,2
        cu(i1) = cu(i1) + cwu(i1,i2)*cphi(i2)
      enddo
      enddo

      RETURN
      end

      subroutine spinubar(p,rm,cs,cubar)
      implicit real*8 (a-b,d-h,o-z)
      implicit complex*16 (c)
      dimension cgamma(0:3,4,4),cgamma5(4,4),cwu(4,2),
     1      cu(4),gmunu(0:3,0:3),cubar(4),
     2      cs(2),cphi(2),p(0:3)
      common /dirac/ ci,cgamma,cgamma5,gmunu

      fac = dsqrt(p(0) + rm)

      cphi(1) =  cs(1)*fac
      cphi(2) =  cs(2)*fac

      cwu(1,1) =    1.d0
      cwu(1,2) =    0.d0
      cwu(2,1) =    0.d0
      cwu(2,2) =    1.d0
      cwu(3,1) = ( p(3)            )/(p(0) + rm)
      cwu(3,2) = ( p(1)-ci*p(2)  )/(p(0) + rm)
      cwu(4,1) = ( p(1)+ci*p(2)  )/(p(0) + rm)
      cwu(4,2) = (-p(3)            )/(p(0) + rm)

      do i1=1,4
        cu(i1) = 0.d0
      do i2=1,2
        cu(i1) = cu(i1) + cwu(i1,i2)*cphi(i2)
      enddo
      enddo

      cubar(1) =   dconjg(cu(1))
      cubar(2) =   dconjg(cu(2))
      cubar(3) =  -dconjg(cu(3))
      cubar(4) =  -dconjg(cu(4))

      RETURN
      end

      subroutine spinv(p,rm,cs,cv)
      implicit real*8 (a-b,d-h,o-z)
      implicit complex*16 (c)
      dimension cgamma(0:3,4,4),cgamma5(4,4),
     1      cwv(4,2),cv(4),gmunu(0:3,0:3),
     2      cs(2),cchi(2),p(0:3)
      common /dirac/ ci,cgamma,cgamma5,gmunu

      fac = dsqrt(p(0) + rm)

      cchi(1) =  dconjg(cs(2))*fac
      cchi(2) = -dconjg(cs(1))*fac

      cwv(1,1) = ( p(3)           )/(p(0) + rm)
      cwv(1,2) = ( p(1)-ci*p(2)  )/(p(0) + rm)
      cwv(2,1) = ( p(1)+ci*p(2)  )/(p(0) + rm)
      cwv(2,2) = (-p(3)           )/(p(0) + rm)
      cwv(3,1) =   1.d0
      cwv(3,2) =   0.d0
      cwv(4,1) =   0.d0
      cwv(4,2) =   1.d0

      do i1=1,4
        cv(i1) = 0.d0
      do i2=1,2
        cv(i1) = cv(i1) + cwv(i1,i2)*cchi(i2)
      enddo
      enddo

      RETURN
      end

      subroutine spinvbar(p,rm,cs,cvbar)
      implicit real*8 (a-b,d-h,o-z)
      implicit complex*16 (c)
      dimension cgamma(0:3,4,4),cgamma5(4,4),
     1     cwv(4,2),cv(4),cvbar(4),gmunu(0:3,0:3),
     2     cs(2),cchi(2),p(0:3)
      common /dirac/ ci,cgamma,cgamma5,gmunu

      fac = dsqrt(p(0) + rm)

      cchi(1) =  dconjg(cs(2))*fac
      cchi(2) = -dconjg(cs(1))*fac

      cwv(1,1) = ( p(3)           )/(p(0) + rm)
      cwv(1,2) = ( p(1)-ci*p(2)  )/(p(0) + rm)
      cwv(2,1) = ( p(1)+ci*p(2)  )/(p(0) + rm)
      cwv(2,2) = (-p(3)           )/(p(0) + rm)
      cwv(3,1) =   1.d0
      cwv(3,2) =   0.d0
      cwv(4,1) =   0.d0
      cwv(4,2) =   1.d0

      do i1=1,4
        cv(i1) = 0.d0
      do i2=1,2
        cv(i1) = cv(i1) + cwv(i1,i2)*cchi(i2)
      enddo
      enddo

      cvbar(1) =  dconjg(cv(1))
      cvbar(2) =  dconjg(cv(2))
      cvbar(3) = -dconjg(cv(3))
      cvbar(4) = -dconjg(cv(4))

      RETURN
      end

      subroutine slash(p,cpslsh)
      implicit real*8 (a-b,d-h,o-z)
      implicit complex*16 (c)
      dimension cgamma(0:3,4,4),cgamma5(4,4),p(0:3)
     1     ,cpslsh(4,4),gmunu(0:3,0:3)
      common /dirac/ ci,cgamma,cgamma5,gmunu

      p(0) = -p(0)

      do i1=1,4
      do i2=1,4
        cpslsh(i1,i2) = (0.d0,0.d0)
        do imu = 0,3
         cpslsh(i1,i2) = cpslsh(i1,i2) - cgamma(imu,i1,i2)*p(imu)
        enddo
      enddo
      enddo

      p(0) = -p(0)

      RETURN
      end

      subroutine ubaru(cubar,ca,cu,cscal)
      implicit real*8 (a-b,d-h,o-z)
      implicit complex*16 (c)
      dimension cubar(4),ca(4,4),cu(4)

      cscal = 0.d0

      do i1=1,4
      do i2=1,4
```

```
            cscal = cscal + cubar(i1)*ca(i1,i2)*cu(i2)
      enddo
      enddo


      RETURN
      end

      subroutine antism(a,b,c,d)
      implicit real*8 (a-h,o-z)
      dimension a(0:3),b(0:3),c(0:3),d(0:3)

      d(0) =   a(1)*b(2)*c(3) + a(2)*b(3)*c(1) + a(3)*b(1)*c(2)
     1       - a(2)*b(1)*c(3) - a(3)*b(2)*c(1) - a(1)*b(3)*c(2)
      d(1) = - a(0)*b(2)*c(3) - a(2)*b(3)*c(0) - a(3)*b(0)*c(2)
     1       + a(2)*b(0)*c(3) + a(3)*b(2)*c(0) + a(0)*b(3)*c(2)
      d(2) = - a(1)*b(0)*c(3) - a(0)*b(3)*c(1) - a(3)*b(1)*c(0)
     1       + a(0)*b(1)*c(3) + a(3)*b(0)*c(1) + a(1)*b(3)*c(0)
      d(3) = - a(1)*b(2)*c(0) - a(2)*b(0)*c(1) - a(0)*b(1)*c(2)
     1       + a(2)*b(1)*c(0) + a(0)*b(2)*c(1) + a(1)*b(0)*c(2)

      RETURN
      end

      function fbox(p,rl)
      implicit real*8 (a-h,o-z)
      common /con/ pi,hbarc,rc,alpha,GF,GV,GA,dMp,dMn,dme,dmnu,dMd,gnpd
     1                  ,dkapp,dkapn,dufac,dN0,e

      fbox = dexp(-(rl*p)**2)*(2.d0*rl**2/pi)**(0.75d0)


      RETURN
      end

      function dgprod(Ekn)
      implicit real*8 (a-h,o-z)
      common /con/ pi,hbarc,rc,alpha,GF,GV,GA,dMp,dMn,dme,dmnu,dMd,gnpd
     1                  ,dkapp,dkapn,dufac,dN0,e

c n + p --> d + gamma
c Fit to ENDF/B-VII.1
      dgprod = 5.2828d-05/dsqrt(EKn)

      dgprod = dgprod*1.d-24  !  cm^2
      RETURN
      end

      function delast(Ekn)
      implicit real*8 (a-h,o-z)
      common /con/  pi,hbarc,rc,alpha,GF,GV,GA,dMp,dMn,dme,dmnu,dMd,gnpd
     1              ,dkapp,dkapn,dufac,dN0,e

c n + p --> d + gamma
c Fit to ENDF/B-VII.1
      delast = 20.d0

      delast = delast*1.d-24  !  cm^2

      RETURN
      end
```