Modification of NJOY2016.67 for JENDL-5 photonuclear sublibrary (Rev. 2)

It is necessary to modify NJOY2016.67 in order to process JENDL-5 photonuclear sublibrary because NJOY2016.67 produces ACE files of LAW=61 for JENDL-5 photonuclear sublibrary, where LAW=4 is adopted and MCNP6.2 and PHITS3.27 cannot use photonuclear ACE files of LAW=61. The conversion of the energy-angle distribution data of LAW=1 LANG=1 in ENDF-6 files to the energy-angle distribution data of LAW=61 in ACE files was introduced in NJOY2016.66, which includes other important revisions for photonuclear data (Some photonuclear libraries have MF6/MT18 with a neutron multiplicity equal to 1 instead of nubar. In this case the neutron multiplicity is replaced with the nubar data.). NJOY2016.65 converts the energy-angle distribution data of LAW=1 LANG=1 in ENDF-6 files to the energy distribution data of LAW=4 with isotropic angular distribution in ACE files, which MCNP6.2 and PHITS3.27 can use, but it could not produce complete ACE files of the JENDL-5 photonuclear sublibrary. Thus we modified NJOY2016.67 only in the conversion of the energy-angle distribution data of LAW=1 LANG=1 in ENDF-6 files based on NJOY2016.65 to produce the ACE file of the JENDL-5 photonuclear sublibrary, which can be used in MCNP6.2 and PHITS3.27. We also modified NJOY2016.67 to support multiple interpolation in MF6, MT5 data, to set negative heating number to 0.0 and to correct a double counting problem of heating number for MF6, MT5 LAW=2. Relativistic conversion between the center of mass system and the laboratory reference framework was not applied. Here we show the modifications of NJOY2016.67. You must use the modifications with your own responsibility.

(1) acepn.f90
line 891

```
! the next piece of code assumes the yield is given
! using one lin-lin interpolation region
! for now: error out and wait for this to come up to actually implement it
nr=nint(scr(5))
→ ! interporations except one lin-lin interpolation are accepted!
    nr=nint(scr(5))
```

line 895

```
if (nr.gt.1.and.nint(scr(8)).ne.2) then
    write(text,'("no linearised multiplicity for izap=",i4," in
mf=6/mt=",i3,".")')izap,mth
    call mess('acephn',text,'this is currently unsupported for photonuclear ACE
files.')
```

```
      endif
      → delete

line 903
      xss(nex+2)=0
      ne=nint(scr(6))
      xss(nex+3)=ne
      do i=1,ne
         xss(nex+3+i)=sigfig(scr(7+2*i)/emev,7,0)
         xss(nex+3+i+ne)=sigfig(scr(8+2*i),7,0)
      enddo
      nex=nex+4+2*ne
      → if ((nr.eq.1).and.(nint(scr(8)).eq.2)) then
            xss(nex+2)=0
            ne=nint(scr(6))
            xss(nex+3)=ne
            do i=1,ne
               xss(nex+3+i)=sigfig(scr(7+2*i)/emev,7,0)
               xss(nex+3+i+ne)=sigfig(scr(8+2*i),7,0)
            enddo
            nex=nex+4+2*ne
         else
            xss(nex+2)=nr
            xss(nex+3)=1
            do i=2,nr
               xss(nex+2+i)=nint(scr(5+(i-1)*2))
            enddo
            do i=1,nr
               xss(nex+2+nr+i)=nint(scr(6+i*2))
            enddo
            ne=nint(scr(6))
            xss(nex+3+2*nr)=ne
            do i=1,ne
               xss(nex+3+2*nr+i)=sigfig(scr(7+(nr-1)*2+2*i)/emev,7,0)
               xss(nex+3+2*nr+i+ne)=sigfig(scr(8+(nr-1)*2+2*i),7,0)
            enddo
            nex=nex+4+2*nr+2*ne
         endif
```

line 1110

```
      xss(phn+2+ie-it)=xss(phn+2+ie-it)+h*ss
      if (xss(tot+ie-1).ne.zero)&
        xss(thn+ie-1)=xss(thn+ie-1)&
        +h*ss/xss(tot+ie-1)
      → xss(phn+2+ie-it)=xss(phn+2+ie-it)+0.5*h*ss
          if (xss(tot+ie-1).ne.zero)&
            xss(thn+ie-1)=xss(thn+ie-1)&
            +0.5*h*ss/xss(tot+ie-1)
```

line 1461

```
      else if (law.eq.1) then
          ll=jscr
          call tab2io(nin,0,0,scr(ll),nb,nw)
          lang=nint(scr(ll+2))
          lep=nint(scr(ll+3))
          ne=nint(scr(ll+5))          ! number of incident energies
          if (lang.eq.1) then         ! legendre polynomials to law=61
              xss(last+1)=61            ! LAW
          else if (lang.eq.2) then ! Kalbach-Mann to law=44
              xss(last+1)=44            ! LAW
          else
              write(text,'("lang=",i3," not supported for law=",i2)')lang,law
              call error('acephn',text,'')
          endif
          xss(landp+jp-1)=-1        ! angular included in energy distribution
          nr=0
          xss(nex)=nr                ! NR set to 0
          lee=nex                     ! lee points to NR
          nex=nex+2*nr+1
          nle=2
          xss(nex)=nle               ! number of energies, NE, default to 2
          nex=nex+1+2*nle             ! leaving room for E(1:2), P(1:2).    nex points to
          xss(last+2)=nex-dlwp+1 ! IDAT
          nr=0
          xss(nex)=nr                ! LDAT(1) = NR set to 0
          nex=nex+1+2*nr
          xss(nex)=ne                ! LDAT(2) = NE set to number of incident energies
          nex=nex+1
```

```
          lle=nex                    ! lle points to LDAT(3) = E(1)
          nex=lle+2*ne               ! nex points to start of first distribution

          ! scr(llh) up to scr(lld-1) is set up for heating
      --> else if (law.eq.1) then
             ! scr(llh) up to scr(lld-1) is set up for heating

line 1492
      ! scr(llh) up to scr(lld-1) is set up for heating
      --> ! scr(llh) up to scr(lld-1) is set up for heating
          xss(landp+jp-1)=-1
          ll=jscr
          xss(last+1)=44
          call tab2io(nin,0,0,scr(ll),nb,nw)
          lang=nint(scr(ll+2))
          lep=nint(scr(ll+3))
          ne=nint(scr(ll+5))
          xss(nex)=0
          lee=nex
          xss(nex+1)=2
          nex=nex+2+2*2
          xss(last+2)=nex-dlwp+1
          xss(nex)=0
          xss(nex+1)=ne
          lle=nex+2
          nex=lle+2*ne

line 1521
      ng=nint(scr(lld+5))
      --> ng=nint(scr(lld+5))
          if (lang.eq.1) then
                xss(last+1)=4
                xss(landp+jp-1)=0
          endif

line 1604
          xss(nex+1+ig+4*ng)=sigfig(akal,7,0) ! a
      else
```

```
if (lang.eq.1) then
    xss(nex+1+ig+3*ng)=nexc-dlwp+1    !pointer to angdist table
    ! convert lang=1 list in scr to a normalized P(1) to P(NA) lis
    scr(ll)=0
    scr(ll+1)=scr(lld+6+ncyc*(ig-1))            !EOUT(ig)
    scr(ll+2)=0
    scr(ll+3)=0
    scr(ll+4)=na                                    !P(l) order (zero is a
    scr(ll+5)=0
    do ia=1,na
        lll=lld+7+ncyc*(ig-1)
        scr(ll+5+ia)=0
        if (scr(lll).ne.zero) then
            scr(ll+5+ia)=scr(lll+ia)/scr(lll) !P(n)/P(0)
        endif
    enddo

    call ptleg2(scr(ll))    !P(l) list in, tab1 (mu,pdf) out

    intmu=2
    xss(nexc)=intmu
    nmu=nint(scr(ll+5))
    xss(nexc+1)=nmu
    do imu=1,nmu
        xss(nexc+1+imu)=sigfig(scr(ll+6+2*imu),7,0)
        xss(nexc+1+nmu+imu)=sigfig(scr(ll+7+2*imu),7,0)
        if (imu.eq.1) then
            xss(nexc+1+2*nmu+imu)=0
        else if (imu.eq.nmu) then
            xss(nexc+1+2*nmu+imu)=1
        else
            del=scr(ll+6+2*imu)-scr(ll+4+2*imu)
            av=(scr(ll+7+2*imu)+scr(ll+5+2*imu))/2
            xss(nexc+1+2*nmu+imu)=&
                                    xss(nexc+1+2*nmu+imu-1)+del*av
            xss(nexc+1+2*nmu+imu)=&
                            sigfig(xss(nexc+1+2*nmu+imu),7,0)
        endif
    enddo
```

```
            nexc=nexc+2+3*nmu
          endif

      endif
      -->       xss(nex+1+ig+4*ng)=sigfig(akal,7,0) ! a
           endif

line 1673
      renorm=one/xss(nex+1+3*ng)
      --> renorm=1.0
          if (xss(nex+1+3*ng).ne.0.0) renorm=one/xss(nex+1+3*ng)

line 1683
      nex=nexc
      --> nex=nex+2+3*ng

line 1797
      xss(thn+i-1)=sigfig(xss(thn+i-1),7,0)
      --> xss(thn+i-1)=sigfig(xss(thn+i-1),7,0)
          If (xss(thn+i-1).lt.0.0d+0) xss(thn+i-1)=0.0d+0

line 1941
      integer::ip,locj,intmu,nmu,imu
      --> integer::ip,locj,intmu,nmu,imu
          integer::nr

line 2140
      ne=nint(xss(l2+3))
      write(nsyso,'(/"    yields:"//&
        &"                  energy            yield"/&
        &6x,2(2x,"------------"))')
      do ii=1,ne
         write(nsyso,'(6x,1p,2e14.6)')&
           xss(l2+4+ii-1),xss(l2+4+ne+ii-1)
      enddo
      --> nr=nint(xss(l2+2))
          if (nr.eq.0) then
             ne=nint(xss(l2+3))
             write(nsyso,'(/"    yields:   int=2"//&
```

```
        &"                  energy            yield"/&
        &6x,2(2x,"-----------"))')
      do ii=1,ne
        write(nsyso,'(6x,1p,2e14.6)')&
          xss(l2+4+ii-1),xss(l2+4+ne+ii-1)
      enddo
  else
      ne=nint(xss(l2+3+2*nr))
      write(nsyso,'(/"    yields:   ")')
      if (nr.eq.2) then
        write(nsyso,'(/"       point=",i4," - ",&
          &i4,"   int=",i2)') nint(xss(l2+2+1)),&
          &nint(xss(l2+2+2)),nint(xss(l2+2+nr+1))
        write(nsyso,'("       point=",i4," - ’’,&
          &i4,"   int=",i2)') nint(xss(l2+2+2)),ne,&
          &nint(xss(l2+2+nr+2))
      endif
      if (nr.eq.3) then
        write(nsyso,'(/"       point=",i4," - ",&
          &i4,"   int=",i2)') nint(xss(l2+2+1)),&
          &nint(xss(l2+2+2)),nint(xss(l2+2+nr+1))
        write(nsyso,'("       point=",i4," - ",&
          &i4,"   int=",i2)') nint(xss(l2+2+2)),&
          &nint(xss(l2+2+3)),nint(xss(l2+2+nr+2))
        write(nsyso,'("       point=",i4," - ",&
          &i4,"   int=",i2)') nint(xss(l2+2+3)),ne,&
          &nint(xss(l2+2+nr+3))
      endif
      write(nsyso,'(/"                  energy            yield"/&
        &6x,2(2x,"-----------"))')
      do ii=1,ne
        write(nsyso,'(6x,1p,2e14.6)')&
          xss(l2+4+2*nr+ii-1),xss(l2+4+2*nr+ne+ii-1)
      enddo
  endif
```