ภาคผนวก

ภาคผนวก ก ตัวอย่างการใส่ค่าพารามิเตอร์ต่างๆ ลงในแบบจำลองพลศาสตร์ควอนตัมเชิงโมเลกุล (QMD) ของปฏิกิริยาการชนของ ${}^{58}_{28}\mathrm{Ni}+{}^{58}_{28}\mathrm{Ni}$ ที่ระดับพลังงาน 1.93 A GeV

```
112564  1000       1
58  28      58  28
 250   0.2   250.
1930.   3.8000    6.5000
   1   1   2   0   1   0     1
   0   0   0   1
   1   1   1   1
5
****************************************************************
     read (5,*) iseed, nruns, icro
     read (5,*) nta, nzta, npr, nzpr
     read (5,*) nt, dt, wrtime
     read (5,*) epp, x00min, x00max
     read (5,*) iflag1, iflag2, iflag3, iflag4, iflag5, iflag6, iflagpi
     read (5,*) iflagr, iflagc, iflagpo, iflagpc
     read (5,*) iflagka, iflagks, iflagkc, iflagkp
     read (5,*) nkru
****************************************************************
c      iseed    initial value for the randomnumber generator
c      nruns    number of independent heavy ion collisions
c      icro
c      nta      number of nucleons in the target
c      nzta     number of protons in the target
c      npr      number of nucleons in the projectile
c      nzpr     number of protons in the projectile
c      nt       number of timesteps
```

```
c     dt        mesh of timesteps (fm/c)
c     wrtime    number of timestep after which the output is written
c     epp       laboratory input energy (MeV/nucleon)
c     x00min    minimum impact parameter to simulate
c     x00max    maximum impact parameter to simulate
c     iflag1    < 0 mean field only,
c               = 1 reduced collision (Nucleons in a nucleus do not allow to collide
c                   each other before their colliding with nucleon in another nucleus.)
c               > 1 full  collisions
c     iflag2    = 0 no coulomb,
c               = 1 including coulomb
c     iflag3    = 2 soft EOS used with the static mean field
c               = 1 hard EOS used with the static mean field
c               = 0 mean field generated  from G/matrix
c     iflag4    = 0 relativistic kinematics (Aichelin parametrization)
c               = 1       '           '     (Cugnon parametrization)
c               = 2 nonrelativistic kinematics
c     iflag5    = 0 no momentum dependent forces
c               = 1 momentum dependent forces ( rho    )
c               = 2   '      '        '    ( rho**2 )
c     iflag6    = 1 includes deuterons
c               = 0 no deuterons
c     iflagpi   = 0 Decay of resonances not allowed (i.e. no pions)
c               = 1 Decay of resonances allowed (pion production).
c     iflagr    = 0 nonrelativistic QMD of Aichelin
c               = 1 Relativistic QMD ( i.e. RQMD ) with generalized
c                   Skyrme forces (+ Optical Potential)
c               = 2 RQMD with self energies
c     iflagc    = 0 nonrelativistic QMD of Aichelin
```

```
c            = 1 Decay of resonances allowed (pion production).
c     iflagpo  = 0 pion propagation without medium-effects
c            = 1 pion propagation with potential from Weise
c            = 2 pion propagation with potential from Kapusta
c     iflagpc  = 0 pion propagation without pi-N Coulomb intr.
c            = 1 pion propagation with pi-N Coulomb intr.
c     iflagka  = 0 No kaon produced.
c            = 1 kaon production included.
c     iflagks  = 0 kaon propagation without K-N elastic collision.
c            = 1 kaon propagation with K-N elastic collision.
c     iflagkc  = 0 kaon propagation without kaon-N Coulomb intr.
c            = 1 kaon propagation with kaon-N Coulomb intr.
c     iflagkp  = 0 kaon production and propagation without medium-effects
c            = 1 kaon production and propagation with kaon potential
c     nkru       amplification number for kaon production
```

ภาคผนวก ข แสดงตัวอย่างผลการคำนวณที่ได้จากแบบจำลองพลศาสตร์ควอนตัมเชิงโมเลกุล (QMD) ของปฏิกิริยาการชนของ $^{58}_{28}\mathrm{Ni}+^{58}_{28}\mathrm{Ni}$ ที่ระดับพลังงาน 1.93 A GeV

ตาราง 6 แสดงตัวอย่างผลการคำนวณที่ได้จากแบบจำลองพลศาสตร์ควอนตัมเชิงโมเลกุล (QMD) ของปฏิกิริยาการชนของ $^{58}_{28}\mathrm{Ni}+^{58}_{28}\mathrm{Ni}$ ที่ระดับพลังงาน 1.93 A GeV

even	time step	p_x	p_y	p_z	Prob.	Channel
1	250	0.0212	-0.1915	0.4956	8.67E-07	1
1	250	-0.1874	-0.0959	-0.846	8.67E-07	1
1	250	0.0355	-0.0129	0.5931	8.67E-07	1
1	250	0.0975	-0.0209	0.5863	8.67E-07	1
1	250	-0.5308	0.1578	0.6654	8.67E-07	1
1	250	0.2036	0.0318	0.4623	1.06E-04	1
1	250	0.1542	0.0615	0.2254	1.06E-04	1
1	250	0.1637	0.2735	0.167	1.06E-04	1
1	250	-0.2741	-0.0194	-0.2558	1.06E-04	1
1	250	0.2146	0.3769	0.4447	1.06E-04	1
1	250	-0.2046	0.0213	0.6143	2.15E-04	1
1	250	-0.447	0.2979	-0.0455	2.15E-04	1
1	250	-0.2868	0.2879	0.3474	2.15E-04	1
1	250	-0.2577	0.0228	-0.0296	2.15E-04	1
1	250	0.017	0.3228	0.7268	2.15E-04	1
1	250	0.0561	-0.1011	0.3825	2.76E-05	1
1	250	-0.3714	0.3064	-0.0506	2.76E-05	1
1	250	-0.0701	-0.0115	-0.311	2.76E-05	1
1	250	-0.0352	-0.2324	0.1328	2.76E-05	1
1	250	0.0763	0.1337	0.6614	2.76E-05	1
1	250	-0.0671	-0.166	0.3248	6.35E-05	1

ตาราง 6 (ต่อ)

even	time step	p_x	p_y	p_z	Prob.	Channel
1	250	0.2331	0.0897	0.1387	6.35E-05	1
1	250	-0.0766	-0.0621	-0.261	6.35E-05	1
1	250	0.054	-0.1652	-0.3217	6.35E-05	1
1	250	-0.1916	-0.0638	-0.2701	6.35E-05	1
1	250	-0.2645	-0.0015	-0.3231	1.74E-04	1
1	250	-0.026	-0.3808	0.2488	1.74E-04	1
1	250	-0.3001	-0.0104	-0.4001	1.74E-04	1
1	250	0.12	0.1063	0.0245	1.74E-04	1
1	250	-0.2315	-0.0022	-0.2276	1.74E-04	1
1	250	-0.1487	-0.0694	0.2254	1.50E-04	1
1	250	-0.1332	0.2259	0.2009	1.50E-04	1
1	250	0.2868	-0.0076	0.3437	1.50E-04	1
1	250	0.1491	0.3749	0.442	1.50E-04	1
1	250	0.17	-0.051	0.5581	1.50E-04	1
1	250	-0.4494	-0.5241	-0.5074	7.38E-04	1
1	250	0.294	-0.3951	-0.1741	7.38E-04	1
1	250	0.1541	-0.1832	0.0153	7.38E-04	1
1	250	0.0856	-0.3699	0.2529	7.38E-04	1
1	250	0.5176	-0.099	0.1378	7.38E-04	1
1	250	0.2331	0.0897	0.1387	6.35E-05	1
1	250	-0.0766	-0.0621	-0.261	6.35E-05	1

ภาคผนวก ค ตัวอย่างโปรแกรมวิเคราะห์การกระจายอะซิมุททอลของปฏิกิริยาการชนของ $^{58}_{28}Ni + ^{58}_{28}Ni$

```
c    Program ckflowa.f is copied from  ckflow.f and changed to
c    analyze the transverse flow, directed flow, elliptic flow,
c    dN/dy, dN/dPt, and azmuthal angular distribution (dN_phi)
c    with errors given by a weighted statistics for kaons produced
c    in reactions of Ni(58,28) + Ni(58,28) at 1.93 AGeV;
c    using itot data-files.   Yu-Ming Zheng and C. Fuchs Dec. 10, 99.
c        It has been revised.  Yu-Ming Zheng    Dec. 12 of 2001.
      program ckflowa
      parameter (nbin = 23, nbin1 = 21, nb2 = 12, amu = 0.939)
c      parameter (pi = 3.1415926)
c.....variables for dndy, px-y (in plane transverse flow) and dpt-y
      dimension dny(nbin,0:5), dnye(nbin,0:5), dnsy(nbin,0:5),
     & pxy(nbin,0:5),pxsy(nbin,0:5), pby(nbin,0:5),pbsy(nbin,0:5),
     & dpty(nbin,0:5), dptsy(nbin,0:5), cpny(nbin,0:5), xc(0:5),
c.....variables for v1_y(direct flow) and v2_y (elliptic flow)
     &  v1y(nbin,0:5), v1sy(nbin,0:5), cpnye(nbin,0:5),
     &  v2y(nbin,0:5), v2sy(nbin,0:5), cpne(0:5), nru(17),
c.....variables for dndpt, v1_pt(differential flow) and v2_pt(elliptic flow)
     &  dnpt(nbin1,0:5), dnpte(nbin1,0:5), dnspt(nbin1,0:5),
     &  v1pt(nbin1,0:5), v1spt(nbin1,0:5), pbpt(nbin1,0:5),
     &  v2pt(nbin1,0:5), v2spt(nbin1,0:5), pbspt(nbin1,0:5),
     &  cpnpt(nbin1,0:5), cpnpte(nbin1,0:5), cpnphie(nb2,0:5),
c.....variables for dN-phi, dPt-phi, v1-phi and v2-phi (azmuthal angular distribution)
     &  dnphi(nb2,0:5), dnphiea(nb2,0:5), dnphie(nb2,0:5),
     &  dnsphi(nb2,0:5),
     &  dptphi(nb2,0:5), dptsphi(nb2,0:5), pbphi(nb2,0:5),
     &  v1phi(nb2,0:5), v1sphi(nb2,0:5), pbsphi(nb2,0:5),
```

```
     &  v2phi(nb2,0:5), v2sphi(nb2,0:5), cpnphi(nb2,0:5),
c.....variables for xkt(total probability of kaon), v1, v2_pt(elliptic flow), and <pt>
     &  v1(0:5), v1s(0:5), v2(0:5), v2s(0:5), apb(0:5),apbs(0:5),
     &  pta(0:5),pts(0:5), xkt(0:5), xke(0:5), xkts(0:5),cpn(0:5)

      pi = 4.0*atan(1.)
      massp = 58
      masst = 58

      e0 = 1.93              !* GeV/nucleon
      bb = 3.80              !* fm
      nkru = 5               !* amplification # of kaon production
c      nkru = 10             !* amplification # of kaon production

      ebeam = amu + e0
c      itot = 1              !* # of data-files
c      itot = 2
c      itot = 4
c      itot = 6
      itot = 5
c      itot = 18

      nru(1) = 999           !* run # of the 1st file
      nru(2) = 1000          !* run # of the second file
      nru(3) = 1000          !* run # of the third file
      nru(4) = 1000          !* run # of the fourth file
      nru(5) = 1000          !* run # of the fifth file
c      nru(6) = 129          !* run # of the sixth file
c      nru(7) = 1219         !* run # of the seventh file
```

```
c      nru(8) = 1219          !* run # of h- file
c      nru(9) = 135           !* run # of i- file
c      nru(10) = 889          !* run # of j- file
c      nru(11) = 1518          !* run # of k- file
c      nru(12) = 559          !* run # of l- file
c      nru(13) = 461          !* run # of m- file
c      nru(14) = 365          !* run # of n- file
c      nru(15) = 1027          !* run # of o- file
c      nru(16) = 889          !* run # of p- file
c      nru(17) = 260          !* run # of q- file
c      nru(18) = 1028          !* run # of s- file
c      nru(19) = 169          !* run # of r- file

      nrun = 0
      do i = 1, itot
      nrun = nrun + nru(i)
      end do

      open (11, file = 'sni1.93p0k0aka', status = 'old')
      open (12, file = 'sni1.93p0k0cka', status = 'old')
      open (13, file = 'sni1.93p0k0mka', status = 'old')
      open (14, file = 'sni1.93p0k0zka', status = 'old')
      open (15, file = 'sni1.93p0k0ka', status = 'old')
c      open (16, file = 'nih1.93b2.85fka', status = 'old')
c      open (17, file = 'nih1.93b2.85gka', status = 'old')
c      open (18, file = 'nih1.93b2.85hka', status = 'old')
c      open (19, file = 'nih1.93b2.85ika', status = 'old')
c      open (20, file = 'nih1.93b2.85jka', status = 'old')
c      open (21, file = 'nih1.93b2.85kka', status = 'old')
```

```
c     open (22, file = 'nih1.93b2.85lka', status = 'old')
c     open (23, file = 'nih1.93b2.85mka', status = 'old')
c     open (24, file = 'nih1.93b2.85nka', status = 'old')
c     open (25, file = 'nih1.93b2.85oka', status = 'old')
c     open (26, file = 'nih1.93b2.85pka', status = 'old')
c     open (27, file = 'nih1.93b2.85qka', status = 'old')
c     open (28, file = 'nih1.93b2.85ska', status = 'old')
c     open (29, file = 'nih1.93b2.85rka', status = 'old')

      open (37, file = 'ks1.93ppxy', status = 'unknown')
      open (38, file = 'ks1.93pvpt', status = 'unknown')
      open (30, file = 'ks1.93pv12', status = 'unknown')
      open (31, file = 'ks1.93pdndpt', status = 'unknown')
      open (32, file = 'ks1.93pv12y', status = 'unknown')
      open (33, file = 'ks1.93pdndy', status = 'unknown')
      open (34, file = 'ks1.93pdpty', status = 'unknown')
      open (35, file = 'ks1.93pdnphi', status = 'unknown')
      open (36, file = 'ks1.93pvphi', status = 'unknown')

c.....initializati5o
c   calculation of the velocity of the N-N CM seen from the
c   Lab [beta(CM)] and its corresponding gama [gama(CM)]:
      bx = 0.0
      by = 0.0
      eep = ebeam + amu
      bz = pbeam/eep
      b2 = bx**2 + by**2 + bz**2
      gam = 1./sqrt(1. - b2)
c   Rapidity of projectile in Lab system
```

```
      pbeam = sqrt(ebeam**2 - amu**2)
      ybeam = 0.5 * log((ebeam + pbeam) / (ebeam - pbeam))
      ylab = abs(ybeam)              !* It is Ylab.
      ycm = 0.5 * abs(ybeam)          !* Ycm

c     Rapidity of projectile in N-N CM system

      cpt = 0.0
      cpt2= cpt**2

      do ij = 0, 5
       xkt(ij) = 0.
       xkts(ij)= 0.
       v1(ij) = 0.
       v1s(ij)= 0.
       v2(ij) = 0.
       v2s(ij)= 0.
       pta(ij) = 0.
       pts(ij) = 0.
       apb(ij) = 0.
       apbs(ij)= 0.
       cpn(ij) = 0.

       do i = 1, nbin
        pxy(i,ij) = 0.
        pxsy(i,ij)= 0.
        v1y(i,ij) = 0.
        v1sy(i,ij)= 0.
        v2y(i,ij) = 0.
```

```
  v2sy(i,ij)= 0.
  dny(i,ij) = 0.
  dnsy(i,ij)= 0.
  dpty(i,ij) = 0.
  dptsy(i,ij)= 0.
  pby(i,ij) = 0.
  pbsy(i,ij)= 0.
  cpny(i,ij) = 0.
end do
do i = 1, nbin1
  v1pt(i,ij) = 0.
  v1spt(i,ij)= 0.
  v2pt(i,ij) = 0.
  v2spt(i,ij)= 0.
  dnpt(i,ij) = 0.
  dnspt(i,ij)= 0.
  pbpt(i,ij) = 0.
  pbspt(i,ij)= 0.
  cpnpt(i,ij) = 0.
end do
do i = 1, nb2
  dnphi(i,ij) = 0.
  dnsphi(i,ij)= 0.
  dptphi(i,ij) = 0.
  dptsphi(i,ij)= 0.
  v1phi(i,ij) = 0.
  v1sphi(i,ij)= 0.
  v2phi(i,ij) = 0.
  v2sphi(i,ij)= 0.
```

```
         pbphi(i,ij) = 0.
         pbsphi(i,ij) = 0.
         cpnphi(i,ij) = 0.
        end do
      end do

      ymin = -1.65 * ycm
      ymax = 1.65 * ycm
      dy = (ymax - ymin)/(nbin-1)
c       dy = 0.2
      ymin = ymin - 0.5 * dy
      ymax = ymax + 0.5 * dy

      phimin = 0.0
      phimax = 360.0
c       dphi = (phimax - phimin) / float(nb2)*pi/180.
      dphi = pi/6.

      ptmin = 0.
      ptmax = amu * cosh(ycm)
c       dpt = (ptmax - ptmin) / float(nbin-1)
c       dpt1= (ptmax - ptmin) / float(nbin1)
      dpt = 0.050
      dpt1= 0.050
       write(30,*) 'massp, masst, nrun, e0, ebeam, cpt =',
     &              massp, masst, nrun, e0, ebeam, cpt
       write(30,*) 'ylab,ycm,dy,dphi,dpt =',ylab,ycm,dy,dphi,dpt

c.....loop over particles to take and analyze data
```

```
massr = massp + masst

amk = 0.494
em = amk
totkn = 0.0
upkn  = 0.0
donkn = 0.0
cptm  = 0.0
 anp = 0.
  cy = 0.
     escap1 = 0.0
 nevent = 0
 nkaat = 0
write(30,*) 'amk, nkru =', amk, nkru
write(30,*) 'bz, gam =', bz, gam

do 100 ii = 1, itot                    !* loop over data-files
 do 200 nn = 1, nru(ii)                    !* loop over runs

 do ij = 0,5
  xke(ij) = 0.

  do i = 1, nbin
    dnye(i,ij)= 0.
  end do
  do i = 1, nbin1
    dnpte(i,ij)= 0.
  end do
  do i = 1, nb2
```

```
     dnphiea(i,ij)= 0.
     dnphie(i,ij)= 0.
    end do
   end do
    an = 0.0

    read (10+ii, *)
    read (10+ii, *)
    read (10+ii, *)
    read (10+ii, *) nkaa                     !* kaon number per run
     nkaat = nkaat + nkaa
    write(6,*) ii, nn

    do 300 j = 1, nkaa                       !* loop over particles
     read (10+ii, *) n1, ka,
c    &   xk, yk, zk, xtk,                    !* reduced output
     &   px, py, pz, ptk,                    !* printed out in N-N c.m. systerm.
     &   prk, ikaa, xcoll
      py = -py                              !* due to initial condition of QMD.
      pz = -pz

      totkn = totkn + 1.0
      pt2= px**2 + py**2
      en = sqrt(pt2 + pz**2 + em**2)
      y  = 0.5 * log((en + pz) / (en - pz))

c---------------------------------------------------------------------------------
c     Upper limit of Pklab < 0.5 GeV/c has been applied in FOPI analysis.
```

```
c   see: Nucl. Phys. A625(1997)307-324 (P310), in which the events analyzed
correspond
c   to the impact parameter b < 3.3 fm within a sharp cutoff model (P312).   99.12.6.
c
c  In Z. Phys. A352(1995)355-357 (P356), in which the events analyzed correspond
c  to the impact parameter b < 3.0 fm within a sharp cutoff model, upper limit
c  cut is: Plab < 0.6 GeV/c for kaon, PLab < about 0.6 GeV/c for pi+ and
c       PLab < 2.0 GeV/c for protons.
c-------------------------------------------------------------------------------
c     transformation to Lab frame
      pzl = pz + bz*gam*(bz*pz*gam/(gam + 1.) + en)
      pkl = sqrt(pt2 + pzl**2)

c        if(pkl .ge. 0.6) then
c          upkn = upkn + 1.0
c          goto 300            !* Upper limit cut: Plab < 0.6 GeV/c for kaon
c        end if

c-------------------------------------------------------------------------------
c    An absolute lower transverse momentum limit of 100 MeV/c exists in FOPI
c    analysis.    see: Nuclear Physics A625(1997)307-324 (P311).  99.12.6.
c-------------------------------------------------------------------------------
       pt = sqrt(pt2)
c        if(pt .le. 0.1) then
c          donkn = donkn + 1.0              !* totkn = upkn+donkn+cptm+anp
c          goto 300                   !* lower limit cut: pt > 0.1 GeV/c
c        end if
c-------------------------------------------------------------------------------
c    pt/em cut   see: Z. Phys. A352(1995)355-357 (P357).  99.12.6.
```

```
c          if(pt/em .le. 0.5) then
c           cptm = cptm + 1.0
c           goto 300                          !* pt/em cut: pt/em > 0.5
c          end if
c-----------------------------------------------------------------------------
c     pt cut : to compare to Fig. 2 of Phys. Lett. B486(00)6-12.  01.12.11.
        if(pt .le. 0.20 .or. pt .ge. 0.80) then
         cptm = cptm + 1.0
         goto 300
        end if
c-----------------------------------------------------------------------------
c    y/ybeam : to compare to Fig2 of PRL 95,012301(2005)
           y1 = y/ybeam
          if(y1 .le. 0.30 .and. y1 .ge. 0.70) then
        cy = cy + 1.0
        goto 300
       end if
c-----------------------------------------------------------------------------
      xprob = prk/float(nkru)            !* divide by amplification factor
       probk = xprob/float(nrun)           !* divide by total # of runs
c-----------------------------------------------------------------------------
c     # of kaons from different channels
c-----------------------------------------------------------------------------
      xc(0) = 1.                    !* total probability of kaon
      xc(1) = 0.                    !* NN
      xc(2) = 0.                    !* Npi
      xc(3) = 0.                    !* ND
      xc(4) = 0.                    !* Dpi
      xc(5) = 0.                    !* DD
```

```
      if(ikaa .eq. 1) xc(1) = 1.
      if(ikaa .eq. 2) xc(2) = 1.
      if(ikaa .eq. 3) xc(3) = 1.
      if(ikaa .eq. 4) xc(4) = 1.
      if(ikaa .eq. 5) xc(5) = 1.
c-----------------------------------------------------------------------------
c         for total probability, v1 and v2 analysis of kaons
          v0 = px/pt*probk
          v0s= (px/pt)**2*probk
          v  = (px**2 - py**2)/pt2*probk
          vs = ((px**2 - py**2)/pt2)**2*probk
          anp = anp + 1.0
         do ij = 0, 5
          xke(ij) = xke(ij) + xprob*xc(ij)
          apb(ij) = apb(ij) + probk*xc(ij)
          apbs(ij)= apbs(ij) + probk**2*xc(ij)
          cpn(ij) = cpn(ij) + xc(ij)
          v1(ij) = v1(ij) + v0*xc(ij)
          v1s(ij)= v1s(ij) + v0s*xc(ij)
          v2(ij) = v2(ij) + v*xc(ij)
          v2s(ij)= v2s(ij) + vs*xc(ij)
          pta(ij) = pta(ij) + pt*probk*xc(ij)
          pts(ij) = pts(ij) + pt2*probk*xc(ij)
         end do
c.....pt cut for flow px-y, v1-y, v2-y, and dN/dy analysis.

         if(pt2 .lt. cpt2) then
c-----------------------------------------------------------------------------
```

```
c         if(pt/em .le. 0.5) then              !* changed    99.11.23.
c          cptm = cptm + 1.0                    !* pt/em cut
c-------------------------------------------------------------------------------
         goto 300
        else
         k = int((y - ymin)/dy) + 1
c          k = int((y + 2.)/dy) + 1
         if(k .gt. 0 .and. k .le. nbin) then
          do ij = 0, 5
           dnye(k,ij) = dnye(k,ij) + xprob*xc(ij)
           pby(k,ij) = pby(k,ij) + probk*xc(ij)
           pbsy(k,ij)= pbsy(k,ij) + probk**2*xc(ij)
           cpny(k,ij)= cpny(k,ij) + xc(ij)
           pxy(k,ij) = pxy(k,ij) + px*probk*xc(ij)
           pxsy(k,ij)= pxsy(k,ij) + px**2*probk*xc(ij)
           v1y(k,ij) = v1y(k,ij) + v0*xc(ij)
           v1sy(k,ij)= v1sy(k,ij) + v0s*xc(ij)
           v2y(k,ij) = v2y(k,ij) + v*xc(ij)
           v2sy(k,ij)= v2sy(k,ij) + vs*xc(ij)
           dpty(k,ij) = dpty(k,ij) + pt*probk*xc(ij)
           dptsy(k,ij)= dptsy(k,ij) + pt2*probk*xc(ij)
          end do
         end if
c.....mirror average
         k = int((-y - ymin)/dy) + 1
c          k = int((-y + 2.)/dy) + 1
         if (k .gt. 0. .and. k .le. nbin) then
          do ij = 0, 5
           dnye(k,ij) = dnye(k,ij) + xprob*xc(ij)
```

```
          pby(k,ij) = pby(k,ij) + probk*xc(ij)
          pbsy(k,ij)= pbsy(k,ij) + probk**2*xc(ij)
          cpny(k,ij)= cpny(k,ij) + xc(ij)
          pxy(k,ij) = pxy(k,ij) - px*probk*xc(ij)
          pxsy(k,ij)= pxsy(k,ij) + px**2*probk*xc(ij)
          v1y(k,ij) = v1y(k,ij) - v0*xc(ij)
          v1sy(k,ij)= v1sy(k,ij) + v0s*xc(ij)
          v2y(k,ij) = v2y(k,ij) + v*xc(ij)
          v2sy(k,ij)= v2sy(k,ij) + vs*xc(ij)
          dpty(k,ij) = dpty(k,ij) + pt*probk*xc(ij)
          dptsy(k,ij)= dptsy(k,ij) + pt2*probk*xc(ij)
         end do
        end if
       end if

c.....y cut for dN-pt, v1-pt and v2-pt (elliptic flow) analysis
        yycm = y/ycm

c..... analasis for central rapidity region
c        if(pt2 .ne. 0. .and. abs(yycm) .le. 0.5) then
c
c..... analasis for projectile and target rapidity region
c        if(pt2 .ne. 0. .and. abs(yycm) .ge. 0.5
c    &     .and. abs(yycm) .le. 1.2) then

c..... analasis for projectile rapidity region
c        if(pt2 .ne. 0. .and. yycm .ge. 0.5
c    &     .and. yycm .le. 1.2) then
c
```

```
c..... analasis for target rapidity region
c         if(pt2 .ne. 0. .and. yycm .le. -0.5
c    &     .and. yycm .ge. -1.2) then
c
c..... compare to Fig.2 of Phys. Lett. B486(00)6-12. 01.12.12.
c         if(yycm .lt. -0.65 .and. yycm .gt. -1.2) then
c
        if(pt2 .ne. 0. ) then
          k = int((pt - ptmin)/dpt1) + 1
          if (k .gt. 0 .and. k .le. nbin1) then
            if (y .gt. 0.) then
              weight = 1.
            else
              weight = -1.
            end if
           do ij = 0, 5
            dnpte(k,ij) = dnpte(k,ij) + xprob*xc(ij)
            pbpt(k,ij) = pbpt(k,ij) + probk*xc(ij)
            pbspt(k,ij)= pbspt(k,ij) + probk**2*xc(ij)
            cpnpt(k,ij) = cpnpt(k,ij) + xc(ij)
            v1pt(k,ij) = v1pt(k,ij) + weight*v0*xc(ij)
            v1spt(k,ij)= v1spt(k,ij) + v0s*xc(ij)
            v2pt(k,ij) = v2pt(k,ij) + v*xc(ij)
            v2spt(k,ij)= v2spt(k,ij) + vs*xc(ij)
           end do
          end if
        end if

c..... for dN-phi and dPt-phi analysis
```

```
c         pt2 = px**2 + py**2
c         pt = sqrt(pt2)
        if(pt .gt. 0.0000001) then
         phi = acos(px/pt)
         k = int((phi -phimin)/dphi) + 1
          if(py .lt. 0.) k = 13 - k
         if(k .gt. 0 .and. k .le. nb2) then
          do ij = 0, 5
           dnphiea(k,ij) = dnphiea(k,ij) + xprob*xc(ij)
           pbphi(k,ij) = pbphi(k,ij) + probk*xc(ij)
           pbsphi(k,ij)= pbsphi(k,ij) + probk**2*xc(ij)
           cpnphi(k,ij) = cpnphi(k,ij) + xc(ij)
           dptphi(k,ij) = dptphi(k,ij) + pt*probk*xc(ij)
           dptsphi(k,ij)= dptsphi(k,ij) + pt2*probk*xc(ij)
           v1phi(k,ij) = v1phi(k,ij) + v0*xc(ij)
           v1sphi(k,ij)= v1sphi(k,ij) + v0s*xc(ij)
           v2phi(k,ij) = v2phi(k,ij) + v*xc(ij)
           v2sphi(k,ij) = v2sphi(k,ij) + vs*xc(ij)
c             an = an + 1.0    !* not to get total number of kaon
           an = an + xprob   !* to get total probability
          end do
         else
          escap1 = escap1 + 1.0
         end if
        end if
300     continue         !* end loop over particles

c       event by event analysis for dn_y
      nevent = nevent + 1
```

```
do ij = 0, 5
 xkt(ij) = xkt(ij) + xke(ij)
 xkts(ij)= xkts(ij) + xke(ij)**2

 do k = 1, nbin
  dny(k,ij) = dny(k,ij) + dnye(k,ij)
  dnsy(k,ij)= dnsy(k,ij) + dnye(k,ij)**2
 end do

 do k = 1, nbin1
  dnpt(k,ij) = dnpt(k,ij) + dnpte(k,ij)
  dnspt(k,ij)= dnspt(k,ij) + dnpte(k,ij)**2
 end do

   do k = 1, nb2
    dnphie(k,ij) = dnphiea(k,ij)/(an*dphi)
   end do

 do k = 1, nb2
  dnphi(k,ij) = dnphi(k,ij) + dnphie(k,ij)
  dnsphi(k,ij)= dnsphi(k,ij) + dnphie(k,ij)**2
 end do

end do

200     continue        !* end loop over runs
    close(10+ii)
100   continue          !* end loop over data files
```

```
c.....calculate  results and errors

      tote = float(nevent)
      tote2 = tote*2.0
c       2 comes in from mirror average

c          for total probability
      if(tote .gt. 1.1) then
      do ij = 0, 5
       xkt(ij) = xkt(ij)/tote
       xkts(ij)= sqrt((xkts(ij)/tote - xkt(ij)**2)/(tote - 1.))

       do i = 1, nbin
        dny(i,ij) = dny(i,ij)/tote2
        dnsyi = (dnsy(i,ij)/tote2 - dny(i,ij)**2)/(tote2 - 1.)
        dnsy(i,ij)=sqrt(dnsyi)
       end do

       do i = 1, nbin1
        dnpt(i,ij) = dnpt(i,ij)/tote
        dnspti = (dnspt(i,ij)/tote - dnpt(i,ij)**2)/(tote - 1.)
        dnspt(i,ij)=sqrt(dnspti)

       end do


       do i = 1, nb2
        dnphi(i,ij) = dnphi(i,ij)/tote
        dnsphii = (dnsphi(i,ij)/tote - dnphi(i,ij)**2)
```

```
 dnsphi(i,ij) = sqrt(dnsphii/(tote - 1.))
 end do

end do
end if

do ij = 0, 5
 if(apb(ij) .le. 0.0) then
  v1(ij) = 0.0
  v1s(ij)= 0.0
  v2(ij) = 0.0
  v2s(ij)= 0.0
  pta(ij) = 0.0
  pts(ij) = 0.0
 else
  v1(ij) = v1(ij)/apb(ij)
  v1s2 = v1s(ij)/apb(ij)
  v2(ij) = v2(ij)/apb(ij)
  v2s2 = v2s(ij)/apb(ij)
  pta(ij) = pta(ij)/apb(ij)
  pts2 = pts(ij)/apb(ij)
 end if

 if(apbs(ij) .le. 0.0) then
  v1s(ij) = 0.0
  v2s(ij) = 0.0
  pts(ij) = 0.0
 else
  xne = apb(ij)**2/apbs(ij)                !* the effective # of events
```

```
        cpne(ij) = xne
        if(xne .le. 1.0) then
         v1s(ij) = 0.0
         v2s(ij) = 0.0
         pts(ij) = 0.0
        else
         v1s(ij)=sqrt((v1s2 - v1(ij)**2)/(xne - 1.0))
         v2s(ij)=sqrt((v2s2 - v2(ij)**2)/(xne - 1.0))
         pts(ij)=sqrt((pts2 - pta(ij)**2)/(xne - 1.0))
        end if
       end if

      if(cpn(ij) .lt. 1.1) then
       apbs(ij)= 0.0
      else
       apb(ij) = apb(ij)/cpn(ij)
       apbs(ij)= sqrt((apbs(ij)/cpn(ij)-apb(ij)**2)/(cpn(ij)-1.0))
       apbs(ij)= apbs(ij)*cpn(ij)
       apb(ij) = apb(ij)*cpn(ij)
      end if

     end do

     do 400 ij = 0, 5
c          for dn_y (and dndy: */dy), px_y, v1_y and v2_y
      do i = 1, nbin
       pbye = pby(i,ij)
       if(pbye .le. 0.0) then
        pxy(i,ij) = 0.0
```

```
 pxsy(i,ij)= 0.0
 v1y(i,ij) = 0.0
 v1sy(i,ij)= 0.0
 v2y(i,ij) = 0.0
 v2sy(i,ij)= 0.0
 dpty(i,ij) = 0.0
 dptsy(i,ij) = 0.0
else
 pxy(i,ij) = pxy(i,ij)/pbye
 pxy2 = pxsy(i,ij)/pbye
 v1y(i,ij) = v1y(i,ij)/pbye
 v1y2 = v1sy(i,ij)/pbye
 v2y(i,ij) = v2y(i,ij)/pbye
 v2y2 = v2sy(i,ij)/pbye
 dpty(i,ij) = dpty(i,ij)/pbye
 dpty2 = dptsy(i,ij)/pbye
end if
 pbsye = pbsy(i,ij)
if(pbsye .le. 0.0) then
 pxsy(i,ij) = 0.0
 v1sy(i,ij) = 0.0
 v2sy(i,ij) = 0.0
 dptsy(i,ij)= 0.0
else
 xnye = pbye**2/pbsye                    !* the effective # of events
 cpnye(i,ij) = xnye
 if(xnye .le. 1.0) then
  pxsy(i,ij) = 0.0
  v1sy(i,ij) = 0.0
```

```
      v2sy(i,ij) = 0.0
      dptsy(i,ij)= 0.0
     else
      pxsy(i,ij) = sqrt((pxy2 - pxy(i,ij)**2)/(xnye - 1.))
      v1sy(i,ij) = sqrt((v1y2 - v1y(i,ij)**2)/(xnye - 1.))
      v2sy(i,ij) = sqrt((v2y2 - v2y(i,ij)**2)/(xnye - 1.))
      dptsy(i,ij)= sqrt((dpty2 - dpty(i,ij)**2)/(xnye - 1.))
     end if
    end if

    cpnys = cpny(i,ij)
    if(cpnys .le. 1.0) then
     pbsy(i,ij) = 0.0
    else
     pby(i,ij) = pby(i,ij)/cpnys
     pbsy(i,ij)=sqrt((pbsy(i,ij)/cpnys-pby(i,ij)**2)/(cpnys-1.))
     pbsy(i,ij) = pbsy(i,ij)*cpnys
     pby(i,ij) = pby(i,ij)*cpnys
    end if

   end do

c        for dn_pt (and dndpt: */dpt), v1_pt and v2_pt
   do i = 1, nbin1
    pbptx = pbpt(i,ij)
    if(pbptx .le. 0.0) then
     v1pt(i,ij) = 0.0
     v1spt(i,ij)= 0.0
     v2pt(i,ij) = 0.0
```

```
 v2spt(i,ij)= 0.0
else
 v1pt(i,ij) = v1pt(i,ij)/pbptx
 v1pt2 = v1spt(i,ij)/pbptx
 v2pt(i,ij) = v2pt(i,ij)/pbptx
 v2pt2 = v2spt(i,ij)/pbptx
end if
if(pbspt(i,ij) .le. 0.0) then
 v1spt(i,ij) = 0.0
 v2spt(i,ij) = 0.0
else
 xnpt = pbptx**2/pbspt(i,ij)          !* the effective # of events
 cpnpte(i,ij) = xnpt
 if(xnpt .le. 1.0) then
  v1spt(i,ij) = 0.0
  v2spt(i,ij) = 0.0
 else
  v1spt(i,ij)= sqrt((v1pt2 - v1pt(i,ij)**2)/(xnpt - 1.0))
  v2spt(i,ij)= sqrt((v2pt2 - v2pt(i,ij)**2)/(xnpt - 1.0))
 end if
end if

cpnc = cpnpt(i,ij)
if(cpnc .le. 1.0) then
 pbspt(i,ij) = 0.0
else
 pbpt(i,ij) = pbpt(i,ij)/cpnc
 pbspt(i,ij)=sqrt((pbspt(i,ij)/cpnc-pbpt(i,ij)**2)/(cpnc-1.))
 pbpt(i,ij) = pbpt(i,ij)*cpnc
```

```
 pbspt(i,ij)= pbspt(i,ij)*cpnc
end if

end do

c        for dn_phi (and dndphi: */dphi) and dpt_phi
do i = 1, nb2
 pbphix = pbphi(i,ij)
 if(pbphix .le. 0.0) then
  dptphi(i,ij) = 0.0
  dptsphi(i,ij)= 0.0
  v1phi(i,ij) = 0.0
  v1sphi(i,ij)= 0.0
  v2phi(i,ij) = 0.0
  v2sphi(i,ij)= 0.0
 else
  dptphi(i,ij) = dptphi(i,ij)/pbphix
  dptphi2 = dptsphi(i,ij)/pbphix
  v1phi(i,ij) = v1phi(i,ij)/pbphix
  v1phi2 = v1sphi(i,ij)/pbphix
  v2phi(i,ij) = v2phi(i,ij)/pbphix
  v2phi2 = v2sphi(i,ij)/pbphix
 end if
 if(pbsphi(i,ij) .le. 0.0) then
  dptsphi(i,ij) = 0.0
  v1sphi(i,ij) = 0.0
  v2sphi(i,ij) = 0.0
 else
  xx = pbphix**2/pbsphi(i,ij)          !* the effective # of events
```

```
      cpnphie(i,ij) = xx
      if(xx .le. 1.0) then
       dptsphi(i,ij) = 0.0
       v1sphi(i,ij) = 0.0
       v2sphi(i,ij) = 0.0
      else
       dptsphi(i,ij)= sqrt((dptphi2-dptphi(i,ij)**2)/(xx - 1.0))
       v1sphi(i,ij) = sqrt((v1phi2 - v1phi(i,ij)**2)/(xx - 1.0))
       v2sphi(i,ij) = sqrt((v2phi2 - v2phi(i,ij)**2)/(xx - 1.0))
      end if
     end if

     xcn = cpnphi(i,ij)
     if(xcn .le. 1.0) then
      pbsphi(i,ij) = 0.0
     else
      pbphi(i,ij) = pbphi(i,ij)/xcn
      pbsphi(i,ij)=sqrt((pbsphi(i,ij)/xcn-pbphi(i,ij)**2)/(xcn-1.))
      pbphi(i,ij) = pbphi(i,ij)*xcn
      pbsphi(i,ij)= pbsphi(i,ij)*xcn
    end if

    end do

400   continue

c.....write out results

    dphi = dphi*180./pi
```

```
      write(30, *)
      write(30, *) 'totkn,upkn,donkn,cptm,anp,escap1 =',
     &            totkn,upkn,donkn,cptm,anp,escap1
      write(30, *) 'tote, tote2 = ', tote, tote2
      write(30, *)

      do j = 0, 5

       if(cpn(j) .eq. 0.0) then
        cpnj = 0.0
       else
        cpnj = 1./sqrt(cpn(j))
       end if
        if(cpne(j) .eq. 0.0) then
         cpnej = 0.0
        else
         cpnej = 1./sqrt(cpne(j))
        end if
       write(30, *) 'j, xkt(j), xkts(j) =', j, xkt(j), xkts(j)
       write(30, *) 'j, apb(j), apbs(j) =', j, apb(j), apbs(j)
       write(30, *) 'j, cpn(j), cpne(j) =', j, cpn(j), cpne(j)
       write(30, *) 'j, pta(j), pts(j) =',  j, pta(j), pts(j)
       write(30, *) 'j, v1(j), v1s(j), cpnej =', j, v1(j),v1s(j),cpnej
       write(30, *) 'j, v2(j), v2s(j), cpnj =', j, v2(j), v2s(j), cpnj

c      squeeze-out ratio
       rout = dnphi(9,j)+dnphi(10,j)+dnphi(27,j)+dnphi(28,j)
       rin  = dnphi(36,j)+dnphi(1,j)+dnphi(18,j)+dnphi(19,j)
```

```
      write(30, *) 'j, Rout/in(j) =', j, rout, rin, rout/rin
      write(30, *)

      y = ymin + 0.5*dy
      do i = 1, nbin
c       y = float(2*i - 1)*0.5*dy - 2.
       if(cpny(i,j) .eq. 0.0) then
        cpnyij = 0.0
       else
        cpnyij = 1./sqrt(cpny(i,j))
       end if
       write(37, 24) y/ycm, pxy(i,j)/em, 0.0, pxsy(i,j)/em,
     &               pxy(i,j), 0.0, pxsy(i,j), cpnye(i,j)
       write(32, 24) y/ycm, v1y(i,j), 0.0, v1sy(i,j),
     &               v2y(i,j), cpnyij, v2sy(i,j), cpny(i,j)
       write(33, 24) y/ycm, dny(i,j), 0.0, dnsy(i,j),
     &               dny(i,j)/dy, 0.0, dnsy(i,j)/dy
       write(34, 24) y/ycm, dpty(i,j), 0.0, dptsy(i,j),
     &               pby(i,j), 0.0, pbsy(i,j)
       y = y + dy
      end do
      write(37, *) 'j', j
      write(32, *) 'j', j
      write(33, *) 'j', j
      write(34, *) 'j', j

24    format(1x,f7.3,2(1x,e11.4,1x,f4.1,1x,e11.4),f12.2)

      pt = ptmin + 0.5*dpt1
```

```
 do i = 1, nbin1
  if(cpnpt(i,j) .eq. 0.0) then
   cpnptij = 0.0
  else
   cpnptij = 1./sqrt(cpnpt(i,j))
  end if
  write(38, 24) pt, v1pt(i,j), 0.0, v1spt(i,j),
&        v2pt(i,j), cpnptij, v2spt(i,j), cpnpt(i,j)
  write(31, 24) pt, dnpt(i,j), 0.0, dnspt(i,j),
&        pbpt(i,j), 0.0, pbspt(i,j), cpnpt(i,j)
  pt = pt + dpt1
 end do
 write(38, *) 'j', j
 write(31, *) 'j', j

 phi = phimin + 0.5*dphi
 do i = 1, nb2
  if(cpnphi(i,j) .eq. 0.0) then
   cpnphiij = 0.0
  else
   cpnphiij = 1./sqrt(cpnphi(i,j))
  end if
  write(35, 24) phi, dnphi(i,j), 0.0, dnsphi(i,j),
&      dptphi(i,j), 0.0, dptsphi(i,j), cpnphie(i,j)
  write(36, 24) phi, v1phi(i,j), 0.0, v1sphi(i,j),
&      v2phi(i,j), cpnphiij, v2sphi(i,j), cpnphi(i,j)
  phi = phi + dphi
 end do
 write(35, *) 'j', j
```

```
write(36, *) 'j', j

end do

stop
end
```

ภาคผนวก ง ตัวอย่างโปรแกรมวิเคราะห์ภาคตัดขวางของปฏิกิริยาการชนของ $^{197}_{79}\mathrm{Au} + ^{197}_{79}\mathrm{Au}$

```
c     Program cb.f is made to calculate the impact parameter
c     of p(1,1) + C(12,6) collisions with r_0 parameter
c     if one hopes to get the averaged value, Obar, of one physical
c     obvervable O(b) by the method of equal-area analyse.
c                Yu-Ming Zheng     July 13, 2004.
      program cb

      parameter (n1=5, amu = 0.939)
      parameter (pi = 3.1415926)
      dimension r0(n1), rho0(n1), ra01(n1), ra58(n1), ra67(n1),
     & ra60(n1), ra62(n1), ra64(n1), ra66(n1), ra56(n1),
     & ra55(n1), ra61(n1), ra12(n1), ra197(n1)

c      pi = 4.0*atan(1.)

      open (18, file = 'cb.d', status = 'unknown')

       r0(1) = 1.142          !* fm
       r0(2) = 1.142
       r0(3) = 1.143
       r0(4) = 1.148
       r0(5) = 1.180
       r0(6) = 1.200

        cc = 3.0/4.0/pi
c        write(18, 23)
c23     format(1x,'r0(fm)',1x,'rho(fm-3)  ra55   ra56  ',
```

```
c     &  'ra58(fm) ra60(fm)',1x, 'ra61   ra62   ra64(fm)',
c     &  ' ra66   ra67   ra197(i)')
      do i = 1, n1
       r03 = r0(i)**3
       rho0(i) = cc/r03
       ra01(i) = 1.0**(1.0/3.0)*r0(i)
       ra12(i) = 12.0**(1.0/3.0)*r0(i)
       ra55(i) = 55.0**(1.0/3.0)*r0(i)
       ra56(i) = 56.0**(1.0/3.0)*r0(i)
       ra58(i) = 58.0**(1.0/3.0)*r0(i)
       ra60(i) = 60.0**(1.0/3.0)*r0(i)
       ra61(i) = 61.0**(1.0/3.0)*r0(i)
       ra62(i) = 62.0**(1.0/3.0)*r0(i)
       ra64(i) = 64.0**(1.0/3.0)*r0(i)
       ra66(i) = 66.0**(1.0/3.0)*r0(i)
       ra67(i) = 67.0**(1.0/3.0)*r0(i)
       ra197(i) = 197.0**(1.0/3.0)*r0(i)
c        write(18, 24) r0(i),rho0(i),ra01(i),ra55(i),ra56(i),ra58(i),
c     &   ra60(i),ra61(i),ra62(i),ra64(i),ra66(i),ra67(i), ra12(i)
      end do
24     format(2x,f5.3,1x,10(f7.4,1x)/7x,'bmax(fm)=')
       bmax = ra01(1)+ra12(1)
       write(18, 25)
25     format(1x,'r0(fm)',1x,'rho(fm-3) ra12(fm) bmax(fm)  ')
       write(18, 26) r0(1), rho0(1),ra12(1), bmax
26     format(2x,f5.3,1x,f7.4,4x,3(f7.4,2x)/)
c
       n = 1
10      write(18, 27) n
```

```
27    format(/1x,'n = ', i3/'   i    bi(fm)')
      nn = 2*n - 1
      do i = 1, nn
      y = (-1)**i
      if(y .lt. 0) then
       bi = sqrt(float(i)/float(2*n))*bmax
       write(18, 28) i, bi
      endif
      end do
28    format(2x,i3,1x,f8.5)
      if(n .lt. 13) then
       n = n + 1
       goto 10
      endif

    stop
    end
```

ภาคผนวก จ ตัวอย่างผลโปรแกรมวิเคราะห์ภาคตัดขวางของปฏิกิริยาการชนของ $^{197}_{79}Au + ^{197}_{79}Au$

r0(fm) rho(fm-3) ra197(fm) bmax(fm)

1.142 0.1603 6.6449 13.2898

n = 1

i bi(fm)

1 9.39730

n = 2

i bi(fm)

1 6.64490

3 11.50930

n = 3

i bi(fm)

1 5.42554

3 9.39730

5 12.13186

n = 4

i bi(fm)

1 4.69865

3 8.13830

5 10.50650

7 12.43146

n = 5

i bi(fm)

1 4.20260

3 7.27912

5 9.39730

7 11.11904

9 12.60780

n = 6

i bi(fm)

1 3.83643

3 6.64490

5 8.57852

7 10.15025

9 11.50930

11 12.72401

n = 7

i bi(fm)

1 3.55185

3 6.15198

5 7.94217

7 9.39730

9 10.65554

11 11.78014

13 12.80636

n = 8

i bi(fm)

1 3.32245

3 5.75465

5	7.42922
7	8.79037
9	9.96734
11	11.01931
13	11.97926
15	12.86779

n = 9

i	bi(fm)
1	3.13243
3	5.42554
5	7.00434
7	8.28764
9	9.39730
11	10.38911
13	11.29415
15	12.13186
17	12.91536

n = 10

i	bi(fm)
1	2.97169
3	5.14711
5	6.64490
7	7.86235
9	8.91506
11	9.85597
13	10.71457
15	11.50930

17 12.25258

19 12.95329

n = 11

i	bi(fm)
1	2.83339
3	4.90758
5	6.33566
7	7.49645
9	8.50018
11	9.39730
13	10.21594
15	10.97368
17	11.68238
19	12.35047
21	12.98424

n = 12

i	bi(fm)
1	2.71277
3	4.69865
5	6.06593
7	7.17731
9	8.13830
11	8.99723
13	9.78102
15	10.50650
17	11.18503
19	11.82468

21	12.43146
23	13.00998

n = 13

i	bi(fm)
1	2.60634
3	4.51432
5	5.82796
7	6.89573
9	7.81903
11	8.64426
13	9.39730
15	10.09432
17	10.74623
19	11.36078
21	11.94376
23	12.49958
25	13.03171

ภาคผนวก ฉ แสดงตัวอย่างโปรแกรมที่คำนวณการเกิดภาคตัดขวางของอนุภาค K^+ มีซอนจากปฏิกิริยาการชนของ ${}^{197}_{79}Au + {}^{197}_{79}Au$ ที่พลังงาน 1.5 A GeV

```
c      Program ckflow21.f is copied from  ckflow6.f and changed
to
c      analyze the transverse flow
c      with errors given by a weighted statistics for kaons
produced
c      in reactions of Au(197,79) + Au(197,79) at 1.5 AGeV;
c      using itot data-files.   Yu-Ming Zheng and C. Fuchs Dec.
10, 99.
c          It has been revised.  Yu-Ming Zheng    Dec. 12 of
2001.
c           edit by Ukrit Chaimongkon for cross section (Phys
Rew 204906)
c           12 Jun 2012 Naresuan University
c      Edit by Ukrit chaimongkin for cross section Naresuan
University 1 july 2012
c----------------------------------------------------------------
-----------
      program
      parameter (nbin = 22, amu = 0.939, amk = 0.494)
      parameter (pi = 3.1415926)
c-----------variables for cross section -----------------------
-----------
      dimension  xc(0:5), nru(30), dsima(nbin, 0:5),
dsimas(nbin,0:5),
     &           dsimae(nbin, 0:5)
c        pi = 4.0*atan(1.)


       e0 = 1.8                    !* GeV/nucleon
       bb = 5.72357                    !* fm
       nkru = 5                     !* amplification # of kaon
production

       ebeam = amu + e0
       itot = 1                    !* # of data-files

       nru(1) = 1000                   !* run # of the 1st file
c       nru(2) = 100                   !* run # of the second file
c       nru(3) = 400
c       nru(4) = 300
c       nru(5) = 100
c       nru(6) = 100
c       nru(7) = 200
c       nru(8) = 200
c       nru(9) = 100
c       nru(10)= 100
c       nru(11)= 100
c       nru(12)= 100
```

```
c        nru(13)= 200
c        nru(14)= 300
c        nru(15)= 100
c        nru(16)= 100
c        nru(17)= 100
c        nru(18)= 100
c        nru(19)= 300
c        nru(20)= 100
         nrun = 0
        do i = 1, itot
         nrun = nrun + nru(i)
        end do

c-------------------------------------------------------------
----------
        open (11, file = 'giihwo1.8b1Cka', status = 'old')
c        open (12, file = 'auhw15b1gka', status = 'old')
c        open (13, file = 'auhw15b1ika', status = 'old')
c        open (14, file = 'auhw15b1kka', status = 'old')
c        open (15, file = 'auhw15b1lka', status = 'old')
c        open (16, file = 'auhw15b1oka', status = 'old')
c        open (17, file = 'auhw15b1pka', status = 'old')
c        open (18, file = 'auhw15b1qka', status = 'old')
c        open (19, file = 'auhw15b1tka', status = 'old')
c        open (20, file = 'auhw15b1uka', status = 'old')
c        open (21, file = 'auhw15b1xka', status = 'old')
c        open (22, file = 'auhw15b1zka', status = 'old')
c        open (23, file = 'auhw15b1bbka', status = 'old')
c        open (24, file = 'auhw15b1eeka', status = 'old')
c        open (25, file = 'auhw15b1ffka', status = 'old')
c        open (26, file = 'auhw15b1ggka', status = 'old')
c        open (27, file = 'auhw15b1iika', status = 'old')
c        open (28, file = 'auhw15b1llka', status = 'old')
c        open (29, file = 'auhw15b1mmka', status = 'old')
c        open (30, file = 'auhw15b1nnka', status = 'old')
c-------------------------------------------------------------
-----------
        open (39, file = 'Ed2s_p2d0dp', status = 'unknown')
c-------------------------------------------------------------
-----------
c---------------------------initialization-------------------
-----------
c----------calculation of the velocity of the N-N CM seen from
the--------
c----------Lab [beta(CM)] and its corresponding gama
[gama(CM)]:----------
        bx = 0.0
        by = 0.0
        eep = ebeam + amu
        pbeam = sqrt(ebeam**2 - amu**2)
        bz = pbeam/eep
        b2 = bx**2 + by**2 + bz**2
        gam = 1./sqrt(1. - b2)
```

```
c-------------------------------------------------------------
-----------
      do ij = 0, 5
       do i = 1, nbin
        dsima(k,ij) = 0.
        dsimas(k,ij) = 0.
       end do
      end do
c---------------------------Please cut------------------------
-----------
      theta = 90. !*Degrees
      dtheta = 4. !*Degrees
      dzeta = 10. !*Degrees
      dzetasr = dzeta*(pi/180.) !*sr
      deplabx = 0.02
      delpc = 3.5
      bmax = 5.72357 !* fm
      bmax2 = bmax**2 !fm^2
      emin = 0.0
      ekk = 0.02
c------------loop over particles to take and analyze data------
-----------
         nevent = 0
         nkaat = 0
      do 100 ii = 1, itot                    !*loop over data-files
       do 200 nn = 1, nru(ii)                !*loop over runs

       do ij = 0,5
          do i = 1, nbin
             dsimae(i,ij) = 0.
        end do
       end do

        read (10+ii, *)
        read (10+ii, *)
        read (10+ii, *)
        read (10+ii, *) nkaa                  !*kaon number per run
         nkaat = nkaat + nkaa
        write(6,*) ii, nn

         do 300 j = 1, nkaa                                !* loop
over particles
         read(10+ii,*) n1,ka,
c    &   xkaa1,xkaa2,xkaa3,xtkaa,    ! reduced output
     &   pkaa1,pkaa2,pkaa3,ptkaa,
     &   prkaa,ikaa
         pkaa2 = -pkaa2                       !*due to initial
condition of QMD.
         pkaa3 = -pkaa3
c-------------------------------------------------------------
-----------
       ptsq  = pkaa1**2 + pkaa2**2
       psq   = ptsq + pkaa3**2
```

```
       ek = sqrt(amk**2 + psq)
c-------------------------------------------------------------------
-----------
c        pb = bx*pkaa1 + by*pkaa2 + bz*pkaa3
c        gg = (gam - 1.)*pb/b2 + gam*ek
c        pkla1 = gg*bx + pkaa1
c        pkla2 = gg*by + pkaa2
c        pkla3 = gg*bz + pkaa3
       pmomen = sqrt(psq)
c        pmomen = sqrt((pkla1**2)+(pkla2**2)+(pkla3**2))
       if(pmomen.eq. 0.) goto 300
       along = pkaa3/pmomen
       along = acos(along)      !* unit: sr
       along = along/pi         !* unit: degree
       along = along*180.
c-------------------------------------------------------------------
----------
          xprob = prkaa/float(nkru)    !* divide by
amplification factor
           probk = xprob/float(nrun)  !* divide by total # of
runs
c-------------------------------------------------------------------
-----------
c      # of kaons from different channels
c-------------------------------------------------------------------
-----------
          xc(0) = 1.                   !* total probability of
kaon
          xc(1) = 0.                   !* NNdsimae(k,ij)
          xc(2) = 0.                   !* Npi
          xc(3) = 0.                   !* ND
          xc(4) = 0.                   !* Dpi
          xc(5) = 0.                   !* DD

        if(ikaa .eq. 1) xc(1) = 1.
        if(ikaa .eq. 2) xc(2) = 1.
        if(ikaa .eq. 3) xc(3) = 1.
        if(ikaa .eq. 4) xc(4) = 1.
        if(ikaa .eq. 5) xc(5) = 1.
c-------------------------------------------------------------------
------------
       if ((along .gt. (theta - dzeta)) .and.
     &     (along .lt. (theta + dzeta))) then
           pcm = sqrt(psq)
           cosa = pkaa3/pcm
           sina = sqrt(1 - cosa**2)
           if (sina .lt. 0) go to 300
           dOlab = (2*pi*sina*dzetasr)
           delpdOlab = psq*delpc*dOlab
           sima = (pi*bmax2*xprob)
           d2sima = ((ek*sima)/delpdOlab)
           ekcm = ek - amk
           iekcm = int(ekcm/ekk) + 1
```

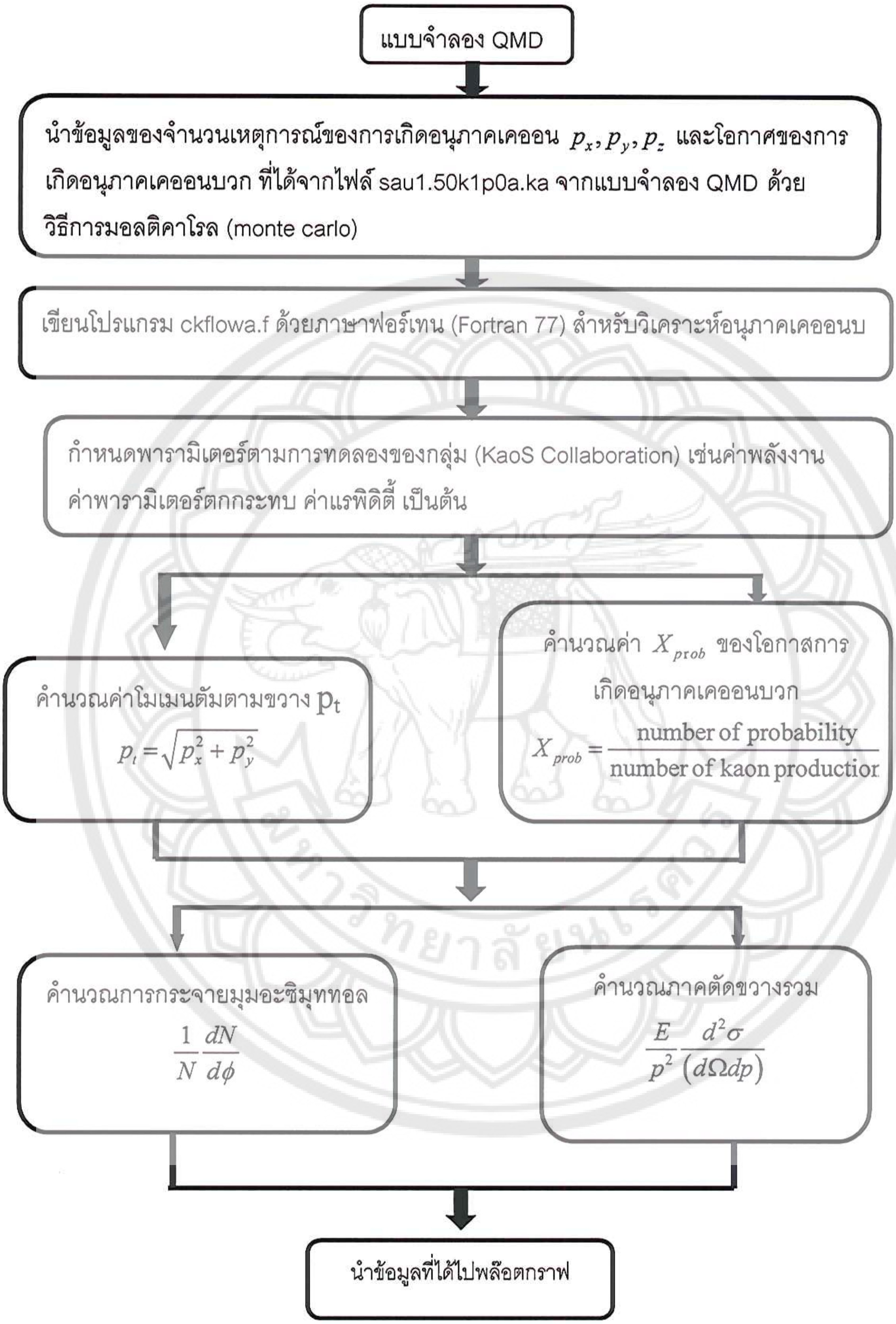
```
          if ((iekcm .gt. 0.) .and. (iekcm .lt. nbin)) then
              do ij = 0, 5
                  dsimae(iekcm,ij) = dsimae(iekcm,ij)
     &            + d2sima*xc(ij)
              end do
          end if
      end if
c-------------------------------------------------------------
------------
300   continue                    !* end loop over particles
c-------------------------------------------------------------
------------
      nevent = nevent + 1                   !* end loop over
particles
      do ij = 0, 5
        do iekcm = 1, nbin
               dsima(iekcm,ij)= dsima(iekcm,ij) +
dsimae(iekcm,ij)
               dsimas(iekcm,ij) = dsimas(iekcm,ij)
     &         + dsimae(iekcm,ij)**2
        end do
      end do
c-------------------------------------------------------------
------------
200   continue                    !* end loop over runs
      close (10+ii)
100   continue                    !* end loop over data files
c-------------------------------------------------------------
------------
      tote = float(nevent)
      if(tote .gt. 1.1)then
      do ij = 0, 5
        do iekcm = 1, nbin

         dsima(iekcm,ij)= dsima(iekcm,ij)/tote
         dsimai = (dsimas(iekcm,ij)/tote
     &   - dsima(iekcm,ij)**2) /(tote - 1.)
         dsimas(iekcm,ij)=sqrt(dsimai)

        end do
      end do
      end if
c-------------------------------------------------------------
------------
c-----------------------write out results----------------------
------------
      do ij = 0, 5
       ek = emin + ekk
        do iekcm = 1, nbin

          write(39,24) ek, dsima(iekcm,ij), 0.0,
dsimas(iekcm,ij)
       ek = ek + ekk
```

```
          end do
        write(39, *) 'ij', ij
24        format(1x,f7.3,2(1x,e11.4,1x,f4.1,1x,e11.4),f12.2)
        end do
        stop
        end
```



ภาพ 35 แผนผังการวิเคราะห์โปรแกรม ckflowa.f สำหรับอนุภาคเคออนบวก