ภาคผนวก

ภาคผนวก ก ตัวอย่างการใส่พารามิเตอร์ในแบบจำลองพลศาสตร์ควอนตัมเชิงโมเลกุล (QMD)ของปฏิกิริยาการชนกันของทองกับทอง ที่พลังงาน 0.15 A GeV

```
235523  1000       1
197 79 197 79
 500    0.2   500.
0150.    3.2245     5.9804
   1   1   1   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 ตัวอย่างผลการคำนวณในแบบจำลองพลศาสตร์ควอนตัมเชิงโมเลกุล (QMD)ของปฏิกิริยาการชนกันของทองกับทอง ที่พลังงาน 0.15 A GeV

5.045 1 400 1634 669

0 0 0 0 0 0 0

963 2 230 622 112 1 0 1

p_x	p_y	p_z		
-0.5929	-0.0857	0.1859	1	5
0.0828	-0.0935	-0.3717	1	1
0.1918	0.0243	-0.2560	1	5
0.0328	-0.0114	-0.3025	1	3
0.1426	-0.1913	-0.2992	1	4
-0.1725	0.1715	0.0657	1	8
0.2095	0.2995	-0.4911	1	4
0.1851	0.0673	-0.3086	1	3
-0.2064	0.0279	-0.0761	1	7
0.0374	0.1816	-0.2576	1	1
0.1201	0.1346	0.2317	1	3
-0.1643	0.2953	0.0593	1	3
-0.0951	0.0509	-0.3001	1	4
0.0695	-0.0146	-0.3145	1	17
-0.1143	-0.1150	-0.0202	1	6
-0.2145	0.2601	0.1864	1	12
0.1084	0.2567	-0.3622	1	3
-0.0317	0.0222	-0.3408	1	11
-0.0420	-0.0814	0.3849	1	12
-0.2275	0.1096	0.3926	1	3
0.1055	0.1360	-0.0814	1	11
-0.1587	-0.0537	-0.1776	1	2
-0.0550	0.2353	-0.3003	1	2

p_x	p_y	p_z		
0.0633	0.2635	0.1621	1	5
0.2134	0.0530	-0.1925	1	6
0.0122	0.0283	0.3839	1	7
0.3633	-0.0646	-0.1862	1	9
-0.0105	-0.0502	-0.0757	1	5
-0.2291	0.0774	0.0982	1	10
0.0907	0.0305	-0.1451	1	5
0.2808	-0.0193	-0.3550	1	15
0.0883	-0.0193	-0.3276	1	8
0.0696	-0.0440	0.0021	1	7
-0.2277	0.2070	-0.1006	1	13
-0.1185	-0.1199	-0.2909	1	3
-0.1152	-0.0749	0.0695	1	9
0.4382	-0.0300	-0.1571	1	9
0.0122	0.1156	-0.2735	1	4
0.2596	-0.1043	-0.1276	1	12
-0.1331	0.1075	-0.3712	1	2
0.2288	-0.0283	-0.2883	1	4
0.0128	-0.2870	-0.0023	1	10
-0.0407	-0.0194	-0.2340	1	4
0.0498	-0.1116	-0.2250	1	2
0.3422	0.0965	-0.1824	1	2
-0.0021	0.1497	-0.1367	1	6
0.1385	0.1223	-0.2291	1	5
0.1349	-0.1173	-0.4674	1	2
-0.0392	0.0856	-0.0277	1	11
0.1540	0.1647	-0.1388	1	8
0.2011	-0.0257	-0.3325	1	5

p_x	p_y	p_z		
0.0634	0.0760	-0.4336	1	2
-0.1210	0.1427	-0.1207	1	8
-0.1050	-0.5214	0.2356	1	7
0.2644	0.2545	-0.0936	1	3
0.1192	-0.1617	-0.4307	1	5
-0.1252	-0.0235	-0.2846	1	9
0.0662	-0.0497	-0.1670	1	9
-0.0349	0.2432	-0.1966	1	8
0.1170	-0.0716	-0.2841	1	4
-0.1403	-0.3319	-0.1826	1	2
0.1005	-0.2816	0.0548	1	4
0.0698	-0.0526	-0.2242	1	3
-0.1096	-0.0249	0.4094	1	6
0.0342	-0.2094	-0.4027	1	3
0.0258	0.0872	-0.3004	1	2
0.1073	0.0186	-0.1707	1	13
-0.0650	-0.2604	-0.0449	1	7
0.0218	0.0289	-0.5528	1	2
0.0761	0.0903	-0.1764	1	9
0.2528	-0.0160	0.0115	1	6
0.4108	0.1026	-0.1359	1	11
0.4026	0.0159	-0.2511	1	6
0.2448	0.0526	-0.2188	1	5
0.3145	-0.3211	-0.1528	1	4
0.0921	0.0028	-0.1940	1	2
-0.0479	0.0411	-0.1895	1	2
0.2467	0.2134	-0.1645	1	5
0.1445	0.1329	-0.1654	1	11

0.1300	0.0845	-0.1371	0	3
0.1840	0.0714	-0.1122	0	8
-0.2734	-0.3248	-0.0212	0	5
0.1477	-0.1053	-0.3485	0	6
0.1245	-0.1089	-0.3837	0	5
-0.2629	-0.1395	-0.4129	0	12
0.0290	-0.1814	-0.3103	0	3
-0.0613	0.0946	-0.2226	0	5

ภาคผนวก ค ตัวอย่างโปรแกรมการคำนวณการไหลเชิงตรงและเชิงวงรีของปฏิกิริยาการชนกันของทองกับทอง ที่พลังงาน 0.15 A GeV

```
c     Program cbfni.f is copied from  fca48.f and changed for
c     analysis of the radial transverse flow, directed flow, elliptic
c     flow, dN/dy, dN/dPt, and azmuthal angular distribution for
c     protons and neutrons in reactions of Ni(58,28) + Ni(58,28).
c     Yu-Ming Zheng  Nov. 20, 99. It has been revised on Dec. 12 of 01.
      program cbfni

      parameter (nbin = 26, nbin1 = 21, nb2 = 36, amu = 0.939)
c      parameter (pi = 3.1415926)
c.....variables for dndy
      dimension dndy(nbin), dndy1(nbin), dndy2(nbin),
c.....variables for dndpt
     &  dndpt(nbin1), dndpt1(nbin1), dndpt2(nbin1),
c.....variables for v1(pt)
     &  v1_pt(nbin1), dn_pt(nbin1), v1s_pt(nbin1),
     &  v1_y(nbin), v1s_y(nbin),
c.....variables for direct flow
     &  px_y(nbin), dn_y(nbin), px2_y(nbin),
c.....variables for v2(pt) (elliptic flow)
     &  v2_y(nbin), v2s_y(nbin), nru(17),
     &  v2_pt(nbin1), dn2_pt(nbin1), v2s_pt(nbin1),
c.....variables for dN-phi and dPt-phi (azmuthal angular distribution)
     &  dndphi(nb2), dndphi1(nb2), dndphi2(nb2),
     &  dn_phi(nb2), dpt_phi(nb2), dpt2_phi(nb2)

      pi = 4.0*atan(1.)
      massp = 197
```

```
      masst = 196
      e0 = 0.150          !* GeV/nucleon
      bb = 4.3645          !* fm
c     nkru = 5            !* amplification # of kaon production
c     bb = 7.5            !* fm    2000.11.24.
c     nkru = 10           !* amplification # of kaon production

      ebeam = amu + e0

      itot = 9
c     itot = 6
c     itot = 17
c     itot = 18

      nru(1) = 795        !* run # of the 1st file
      nru(2) = 1000       !* run # of the 1st file
      nru(3) = 1000       !* run # of the second file
      nru(4) = 1000       !* run # of the third file
      nru(5) = 798        !* run # of the third file
      nru(6) = 1000       !* run # of the fourth file
      nru(7) = 902        !* run # of the fourth file
      nru(8) = 1000       !* run # of f- file
      nru(9) = 1000       !* run # of g- file
c     nru(10) = 1000      !* run # of h- file
c     nru(11) = 1518      !* run # of i- file
c     nru(12) = 559       !* run # of j- file
c     nru(13) = 461       !* run # of k- file
c     nru(14) = 365       !* run # of l- file
c     nru(15) = 1027      !* run # of m- file
```

```
c      nru(16) = 889          !* run # of n- file
c      nru(17) = 260          !* run # of o- file
c      nru(18) = 1028         !* run # of o- file

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

       open (11, file = 'auswb1.5aout', status = 'old')
       open (12, file = 'auswb1.5bout', status = 'old')
       open (13, file = 'auswb1.5cout', status = 'old')
       open (14, file = 'auswb1.5eout', status = 'old')
       open (15, file = 'auswb1.5fout', status = 'old')
       open (16, file = 'auswb1.5hout', status = 'old')
       open (17, file = 'auswb1.5iout', status = 'old')
       open (18, file = 'auswb1.5kout', status = 'old')
       open (19, file = 'auswb1.5mout', status = 'old')
c      open (20, file = 'auswb1.5jout', status = 'old')
c      open (21, file = 'nih1.93b2.85kout', status = 'old')
c      open (22, file = 'nih1.93b2.85lout', status = 'old')
c      open (23, file = 'nih1.93b2.85mout', status = 'old')
c      open (24, file = 'nih1.93b2.85nout', status = 'old')
c      open (25, file = 'nih1.93b2.85oout', status = 'old')
c      open (26, file = 'nih1.93b2.85pout', status = 'old')
c      open (27, file = 'nih1.93b2.85qout', status = 'old')
c      open (28, file = 'nih1.93b2.85sout', status = 'old')
```

```
c     open (35, file = 'auoutppxy', status = 'unknown')
c     open (36, file = 'auoutpvpt', status = 'unknown')
c     open (30, file = 'auoutpv12', status = 'unknown')
c     open (31, file = 'auoutpdndpt', status = 'unknown')
      open (32, file = '1aus0.15pv12y', status = 'unknown')
c     open (33, file = 'auoutpdndy', status = 'unknown')
c     open (34, file = 'auoutpdnphi', status = 'unknown')

c.....initializati5o
      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

      bx = 0.0
      by = 0.0
      eep = ebeam + amu
      bz = pbeam/eep
      b2 = bx**2 + by**2 + bz**2
      gam = 1./sqrt(1. - b2)

     cpt = 0.0
     cpt2= cpt**2

     do i = 1, nbin
       px_y(i) = 0.
       dn_y(i) = 0.
       px2_y(i)= 0.
       v1_y(i) = 0.
```

```
    v1s_y(i)= 0.
    v2_y(i) = 0.
    v2s_y(i)= 0.
    dndy(i) = 0.
    dndy2(i)= 0.
  end do
  do i = 1, nbin1
    v1_pt(i) = 0.
    dn_pt(i) = 0.
    v1s_pt(i)= 0.
    v2_pt(i) = 0.
    dn2_pt(i)= 0.
    v2s_pt(i)= 0.
    dndpt(i) = 0.
    dndpt2(i)= 0.
  end do

  do i = 1, nb2
    dndphi(i) = 0.
    dndphi2(i)= 0.
    dn_phi(i) = 0.
    dpt_phi(i) = 0.
    dpt2_phi(i) = 0.
  end do
   aphi = 0.

  anp= 0.
  v1 = 0.
  v1s= 0.
```

```
      v2 = 0.
      v2s= 0.

      ymin = -1.25 * ycm
      ymax = 1.25 * 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)
      dphi = pi/18.

      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,ntest,e0,ebeam,cpt =',
     &          massp,masst,ntest,e0,ebeam,cpt
      write(30,*) 'ylab,ycm,dy,dphi,dpt =',ylab,ycm,dy,dphi,dpt
      write(30,*) 'bz, gam, nrun =', bz, gam, nrun
      write(30,*)

c.....loop over particles to take and analyze data
      massr = massp + masst
```

```
c       em = 1.0
        em = amu
        aaa = 0.0
        upbn = 0.0
        ccc = 0.0
        escap = 0.0
        escap1 = 0.0
        nevent = 0

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

        do i = 1, nbin
        dndy1(i)= 0.
        end do
        do i = 1, nbin1
        dndpt1(i)= 0.
        end do
        do i = 1, nb2
        dndphi1(i)= 0.
        end do
        an = 0.0

        read (10+ii, *)
        read (10+ii, *)
        read (10+ii, *)
c       masst = masst + massr
        write(6,*) nn
        do 300 j = 1, massr              !* loop over particles
```

```
      read (10+ii, *) px, py, pz, idp, idcol    !* printed out in N-N c.m. systerm.
      py = -py                  !* due to initial condition of QMD.
      pz = -pz
       aaa = aaa + 1.0
c      print *, px, py,pz,idp
        if(idp .ne. 1) goto 300           !* proton flow
c         if(lb .ne. 2) goto 300          !* neutron flow
c       print *, px, py,pz,idp
        en = sqrt(px**2 + py**2 + pz**2 + amu**2)
        y  = 0.5 * log((en + pz) / (en - pz))

        pt2= px**2 + py**2
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
c   correspond to the impact parameter b < 3.3 fm within a sharp cutoff
c   model (P312).   99.12.6.
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)
      pbzl = sqrt(pt2 + pzl**2)

c        if(pbzl .ge. 2.0) then
c          upbn = upbn + 1.0
c          go to 300            !* Up limit cut: PLab < 2.0 GeV/c
```

```
c          end if

           pt = sqrt(pt2)
c-------------------------------------------------------------------------
c      pt/amu cut   see: Z. Phys. A352(1995)355-357 (P357).  99.12.6.
c           if(pt/amu .le. 0.5) then
c            ccc = ccc + 1.0
c            goto 300              !* pt/amu cut: pt/amu > 0.5
c           end if
c-------------------------------------------------------------------------
c     |py|/amu cut   see: Z. Phys. A352(1995)355-357 (P357).  99.12.6.
c            pys = abs(py)
c           if(pys/amu .le. 0.5) goto 300    !* |py|/amu cut: |py|/amu > 0.5
c.....................................................................
c          mp = 0.938                  !*(GeV/c**2)
           mp = 1.007                  !*(amu)
           pp = sqrt(px**2 + py**2 + pz**2 )
              ep = sqrt(pp**2+mp**2)
              bt = pt/ep
           bt2=bt**2
           gamt = 1./sqrt(1. - bt2)
c.....................................................................
c          calculattion of transvers component
c          p=sqrt(px**2 + py**2)
           up=bz*gam
           ut=bt*gamt
c          pt2= px**2 + py**2
c          pt = sqrt(pt2)
c          ut = pt/amu
```

```
      ut0 = ut/up
c     ut0 = pt/p
c-----------------------------------------------------------------------
      if(ut0 .ge. 0.80) then !*compare to Fig.19 of Nuc Phys
       ccc = ccc + 1.0              !* A876(2012)1-60
       goto 300                     !*  pt > 0.80.
      end if
c-----------------------------------------------------------------------
      if(pt .le. 0.0000001) then
       escap = escap + 1.0          !* aaa = escap + anp
       go to 300
      end if
c-----------------------------------------------------------------------
c     for v1 analysis
       v0 = px/pt
       v00= v0**2
       v1 = v1 + v0
       v1s= v1s + v00
c     for v2 analysis
       v  = (px**2 - py**2) / pt2
       vv = v**2
       v2 = v2 + v
       v2s= v2s + vv
       anp = anp + 1.0
c.....pt cut for flow px-y, v1-y, v2-y, and dN/dy analysis.
      if(pt2 .lt. cpt2) then
c       if(pt/amu .le. 0.5) then      !* changed    99.11.23.
c        ccc = ccc + 1.0
       goto 300                     !* pt/amu cut
```

```
          else
           k = int((y - ymin)/dy) + 1
c           k = int((y + 2.)/dy) + 1
           if(k .gt. 0 .and. k .le. nbin) then
             px_y(k) = px_y(k) + px
             dn_y(k) = dn_y(k) + 1.
             px2_y(k)= px2_y(k) + px**2
             v1_y(k) = v1_y(k) + v0
             v1s_y(k)= v1s_y(k) + v00
             v2_y(k) = v2_y(k) + v
             v2s_y(k)= v2s_y(k) + vv
             dndy1(k)= dndy1(k) + 1.
           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
             px_y(k) = px_y(k) - px
             dn_y(k) = dn_y(k) + 1.
             px2_y(k)= px2_y(k) + px**2
             v1_y(k) = v1_y(k) - v0
             v1s_y(k)= v1s_y(k) + v00
             v2_y(k) = v2_y(k) + v
             v2s_y(k)= v2s_y(k) + vv
             dndy1(k)= dndy1(k) + 1.
           end if
          end if

c.....y cut for 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..... 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..... analasis for target rapidity region
            if(pt2 .ne. 0. and. yycm .le. -0.5
     &      .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
```