UrQMD (Fortran) calls ROOT (C++)

Fan Si

Status

- Generally successful (but with some cases to be determined / developed)
  - Define C++ functions, and let them called by Fortran at proper locations
  - Define C++ extern struct to link Fortran quantities (commons)

- Location: ui:/ustcfs/HICUser/fsi/urqmd

- Original UrQMD: /ustcfs/HICUser/fsi/urqmd/urqmd
  - Change: pythia6409.f −> pytahi6428.f (final pythia6), GNUmakefile updated
  - Change: new gfortran option -std=legacy in mk/Linux.mk

- New Fortran & C++ codes: /ustcfs/HICUser/fsi/urqmd/curqmd
  - c_*.f, curqmd.[cxx,h], GNUmakefile, run.sh

What to store (UrqmdDst)

column#		contents
0		"E" (only in ○16)
1	√	# of collisions
2	√	# of elastic collisions
3	√	# of inelastic collisions
4	√	# of Pauli-blocked collisions
5	√	# of decays
6	√	# of produced *hard* baryon resonances
7	√	# of produced *soft* baryon resonances
8	√	# of baryon resonances produced via a decay of another resonance

- Entry for each time step of events (few)
- Additional branch
  - b
  - time: event level (instead of track level)
  - Npart: Aproject + Atarget − #(pptype%100==0)
  - Ntracks
  - spin
  - iso3_old1 & iso3_old2

○13	○14	○15	○16	contents
		1		ind: index of particle (see `CTOption(56)`)
1	1	2	1	t: time of particle
2	2	3	2 √	r_x: x coordinate
3	3	4	3 √	r_y: y coordinate
4	4	5	4 √	r_z: z coordinate
5	5	6	5	E: energy of particle
6	6	7	6 √	p_x: x momentum component
7	7	8	7 √	p_y: y momentum component
8	8	9	8 √	p_z: z momentum component
9	9	10	9 √	m: mass of particle
10	10	11	10 √	ityp: particle-ID
11	11	12	11 √	iso3: $2 \cdot I_3$ (see Section 1.2)
12	12	13	12 √	ch : charge of particle
13	13	14	13 √	parent collision number (see Table 10)
14	14	15	14 √	N_{coll} number of collisions
		16		S : strangeness
15	15		15 √	parent process type (see Table 11)
		17		history information (debugging only)
16			√	t^{fr} : freeze-out time of particle
17			√	r_x^{fr} : freeze-out x coordinate
18			√	r_y^{fr} : freeze-out y coordinate
19			√	r_z^{fr} : freeze-out z coordinate
20				E^{fr} : freeze-out energy of particle
21			√	p_x^{fr} : freeze-out momentum x component
22			√	p_y^{fr} : freeze-out momentum y component
23			√	p_z^{fr} : freeze-out momentum z component
	16*		√	τ_{dec} decay time of particle
	17*		√	τ_{form} formation time of particle
	18*		√	R_σ cross section reduction factor
	19*		√	unique particle number (not ID!)
			16* √	$\mathrm{ityp}_1^{\mathrm{old}}$: particle-ID of parent particle # 1
			17* √	$\mathrm{ityp}_2^{\mathrm{old}}$: particle-ID of parent particle # 2

What to store (UrqmdCollisionDst)

column#		format	contents
1	√	(i8)	number of ingoing particles N_{in}
2	√	(i8)	number of outgoing particles N_{out}
3	√	(i4)	process ID (see Table 11)
4	√	(i7)	collision/entry counter
5	√	(f8.3)	collision time t_{coll} in fm/c
6	√	(e12.4)	center of mass energy of the collision $\sqrt{s}$ in GeV
7	√	(e12.4)	total cross-section of the collision σ_{tot} in mbarn
8	√	(e12.4)	partial cross-section of the actual sub-process σ_{i} in mbarn
9	√	(e12.4)	Baryon density at collision point ρ_{B} in units of ρ_0

- Not created or stored if storeCollision = false

- Entry for each collision of events (many)
- Additional branch
  - b
  - Ntracks
  - spin
  - iso3_old1 & iso3_old2

o13	o14	o15	o16	contents
		1		ind: index of particle (see CTOption(56))
1	1	2	1	t: time of particle
2	2	3	2 √	r_x: x coordinate
3	3	4	3 √	r_y: y coordinate
4	4	5	4 √	r_z: z coordinate
5	5	6	5	E: energy of particle
6	6	7	6 √	p_x: x momentum component
7	7	8	7 √	p_y: y momentum component
8	8	9	8 √	p_z: z momentum component
9	9	10	9 √	m: mass of particle
10	10	11	10 √	ityp: particle-ID
11	11	12	11 √	iso3: $2 \cdot I_3$ (see Section 1.2)
12	12	13	12 √	ch : charge of particle
13	13	14	13 √	parent collision number (see Table 10)
14	14	15	14 √	N_{coll} number of collisions
		16		S : strangeness
15	15		15 √	parent process type (see Table 11)
		17		history information (debugging only)
16				t^{fr} : freeze-out time of particle
17				r_x^{fr} : freeze-out x coordinate
18				r_y^{fr} : freeze-out y coordinate
19				r_z^{fr} : freeze-out z coordinate
20				E^{fr} : freeze-out energy of particle
21				p_x^{fr} : freeze-out momentum x component
22				p_y^{fr} : freeze-out momentum y component
23				p_z^{fr} : freeze-out momentum z component
	16*			τ_{dec} decay time of particle
	17*			τ_{form} formation time of particle
	18*			R_σ cross section reduction factor
	19*		√	unique particle number (not ID!)
			16* √	$\text{ityp}_1^{\text{old}}$: particle-ID of parent particle # 1
			17* √	$\text{ityp}_2^{\text{old}}$: particle-ID of parent particle # 2

What is new

- Can store spin, iso3_old
- Can store parent information (iType_old, iso3_old) during final output
- Can store f13 information @ each time step (original f13 only stores final)
- Can set quiet output to command screen
- Can skip empty event (Ncoll + Ndecays == 0)
- Can set random seed event-by-event

How to store parent information

- Integer (4 bytes): origin (origin definition)
  - = ppType + 100*(# scatters) + 1000*(|iType_old1| + 1000*|iType_old2|)
  - 0 <= ppType < 100, 0 <= |iType_old| < 1000
  - No iso3_old or sign of iType_old

```
if ( (itypt(1).eq.ityp(inew(i))).and.
     (iso3t(1).eq.iso3(inew(i))) ) then
   origin(inew(i))=origint(1)+100
   uid(inew(i))=uidt(1)
elseif ( (itypt(2).eq.ityp(inew(i))).and.
         (iso3t(2).eq.iso3(inew(i))) ) then
   origin(inew(i))=origint(2)+100
   uid(inew(i))=uidt(2)
```

ityp	nucleon	ityp	delta	ityp	lambda	ityp	sigma	ityp	xi	ityp	omega
1	N_{938}	17	Δ_{1232}	27	Λ_{1116}	40	Σ_{1192}	49	Ξ_{1317}	55	Ω_{1672}
2	N_{1440}	18	Δ_{1600}	28	Λ_{1405}	41	Σ_{1385}	50	Ξ_{1530}		
3	N_{1520}	19	Δ_{1620}	29	Λ_{1520}	42	Σ_{1660}	51	Ξ_{1690}		
4	N_{1535}	20	Δ_{1700}	30	Λ_{1600}	43	Σ_{1670}	52	Ξ_{1820}		
5	N_{1650}	21	Δ_{1900}	31	Λ_{1670}	44	Σ_{1775}	53	Ξ_{1950}		
6	N_{1675}	22	Δ_{1905}	32	Λ_{1690}	45	Σ_{1790}	54	Ξ_{2025}		
7	N_{1680}	23	Δ_{1910}	33	Λ_{1800}	46	Σ_{1915}				
8	N_{1700}	24	Δ_{1920}	34	Λ_{1810}	47	Σ_{1940}				
9	N_{1710}	25	Δ_{1930}	35	Λ_{1820}	48	Σ_{2030}				
10	N_{1720}	26	Δ_{1950}	36	Λ_{1830}						
11	N_{1900}			37	Λ_{1890}						
12	N_{1990}			38	Λ_{2100}						
13	N_{2080}			39	Λ_{2110}						
14	N_{2190}										
15	N_{2200}										
16	N_{2250}										

Table 1: Baryon-itypes used in **UrQMD**. Antibaryons carry a negative sign.

ityp	0^{-+}	ityp	1^{--}	ityp	0^{++}	ityp	1^{++}	ityp	charmed
101	π	104	ρ	111	a_0	114	a_1	133	D
106	K	108	K^*	110	K_0^*	113	K_1^*	134	D^*
102	η	103	ω	105	f_0	115	f_1	135	J/Ψ
107	η'	109	ϕ	112	f_0^*	116	f_1'	136	χ_c
ityp	1^{+-}	**ityp**	2^{++}	**ityp**	$(1^{--})^*$	**ityp**	$(1^{--})^{**}$	137	Ψ'
122	b_1	118	a_2	126	ρ_{1450}	130	ρ_{1700}	138	D_s
121	K_1	117	K_2^*	125	K_{1410}^*	129	K_{1680}^*	139	D_s^*
123	h_1	119	f_2	127	ω_{1420}	131	ω_{1662}		
124	h_1'	120	f_2'	128	ϕ_{1680}	132	ϕ_{1900}		

- ppType = origin%100
- # scatters = origin/100%10

```
   2       1  10    318   3.486  0.1665E+01  0.1983E+02  0.1983E+02  0.5118E+00
714  0.34855015E+01 -0.31837811E+01  0.30769649E+01 -0.22947721E+01  0.62539188E+00 -0.15451485E+00  0.31808005E+00 -0.49701232E+00  0.13800000E+00       101 -2 -1      277   1  0        1001015
233  0.34855015E+01 -0.26863519E+01  0.33516045E+01 -0.23172907E+01  0.44498825E+01  0.47432728E+00 -0.64771840E+00 -0.42751707E+01  0.93800002E+00         1  1  1      267   4  0        1017030
233  0.34855015E+01 -0.29350665E+01  0.32142847E+01 -0.23060314E+01  0.50752743E+01  0.31981242E+00 -0.32963835E+00 -0.47721830E+01  0.16654542E+01         4 -1  0      318   5  0        1101010
```

How to store parent information

- Integer (4 bytes): origin (new definition)
  - = ppType + 100*(# scatters) + 1000*((iso3_old1+3)+7*(t_iType_old1+100)) + 1400000*((iso3_old2+3)+7*(t_iType_old2+100))
  - 0 <= ppType < 100, 0 <= iso3_old+3 <= 6
  - t_iType_old = iType_old; if(>=100) −=40; if(<=−100) +=40
    - 0 <= t_iType_old+100 < 200
  - origin < 1.96e9 (4-byte integer max ~ 2.1e9)
  - !!!Also affect *.txt output
  - ppType = origin%100
  - # scatters = origin/100%10
  - If ppType does not contain # scatters, can be stored 1-byte

```
ppType = origin%1000;
if(origin<1000)
    iType_old1 = iso3_old1 = iType_old2 = iso3_old2 = 0;
else{
    iso3_old1 = origin/1000%7-3;
    iType_old1 = origin/7000%200-100;
    if(iType_old1>=60)
        iType_old1 += 40;
    if(iType_old1<=-60)
        iType_old1 -= 40;
    if(origin<1400000)
        iType_old2 = iso3_old2 = 0;
    else{
        iso3_old2 = origin/1400000%7-3;
        iType_old2 = origin/9800000-100;
        if(iType_old2>=60)
            iType_old2 += 40;
        if(iType_old2<=-60)
            iType_old2 -= 40;
    }
}
```

ityp	nucleon	ityp	delta	ityp	lambda	ityp	sigma	ityp	xi	ityp	omega
1	N_{938}	17	Δ_{1232}	27	Λ_{1116}	40	Σ_{1192}	49	Ξ_{1317}	55	Ω_{1672}
2	N_{1440}	18	Δ_{1600}	28	Λ_{1405}	41	Σ_{1385}	50	Ξ_{1530}		
3	N_{1520}	19	Δ_{1620}	29	Λ_{1520}	42	Σ_{1660}	51	Ξ_{1690}		
4	N_{1535}	20	Δ_{1700}	30	Λ_{1600}	43	Σ_{1670}	52	Ξ_{1820}		
5	N_{1650}	21	Δ_{1900}	31	Λ_{1670}	44	Σ_{1775}	53	Ξ_{1950}		
6	N_{1675}	22	Δ_{1905}	32	Λ_{1690}	45	Σ_{1790}	54	Ξ_{2025}		
7	N_{1680}	23	Δ_{1910}	33	Λ_{1800}	46	Σ_{1915}				
8	N_{1700}	24	Δ_{1920}	34	Λ_{1810}	47	Σ_{1940}				
9	N_{1710}	25	Δ_{1930}	35	Λ_{1820}	48	Σ_{2030}				
10	N_{1720}	26	Δ_{1950}	36	Λ_{1830}						
11	N_{1900}			37	Λ_{1890}						
12	N_{1990}			38	Λ_{2100}						
13	N_{2080}			39	Λ_{2110}						
14	N_{2190}										
15	N_{2200}										
16	N_{2250}										

Table 1: Baryon-itypes used in **UrQMD**. Antibaryons carry a negative sign.

ityp	0^{-+}	ityp	1^{--}	ityp	0^{++}	ityp	1^{++}	ityp	charmed
101	π	104	ρ	111	a_0	114	a_1	133	D
106	K	108	K^*	110	K_0^*	113	K_1^*	134	D^*
102	η	103	ω	105	f_0	115	f_1	135	J/Ψ
107	η'	109	ϕ	112	f_0^*	116	f_1'	136	χ_c
ityp	1^{+-}	**ityp**	2^{++}	**ityp**	$(1^{--})^*$	**ityp**	$(1^{--})^{**}$	137	Ψ'
122	b_1	118	a_2	126	ρ_{1450}	130	ρ_{1700}	138	D_s
121	K_1	117	K_2^*	125	K_{1410}^*	129	K_{1680}^*	139	D_s^*
123	h_1	119	f_2	127	ω_{1420}	131	ω_{1662}		
124	h_1'	120	f_2'	128	ϕ_{1680}	132	ϕ_{1900}		

How to set seed for random generator

- Random seed generator
  - iseed is actual seed when used
    - Changed after each use
  - ranseed is seed of iseed
  - 1. set ranseed < 0
  - 2. if < 0, set ranseed by time
    - if different from old
  - 3. set iseed = −ranseed
  - 4. modify iseed (extra if < 0)

```
      subroutine sseed(ranseed)
c
c     reset the random number generato
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      implicit none

      real*8 dummy, ran2
      integer iseed, ranseed, oldseed, time, timeseed
      common /seed/iseed,oldseed

      if (ranseed.le.0) then
         timeseed = abs(time())
         if (timeseed.eq.oldseed) return
         ranseed = timeseed
      endif
      oldseed = ranseed
      iseed = -ranseed
      dummy = ran2(iseed)

      return
      end
```

```
c     initialize random number generator
c     call auto-seed generator only for first event and if no seed was fixed
      if(.not.firstseed.and.(.not.fixedseed)) then
         ranseed=-(1*abs(ranseed))
         call sseed(ranseed)
      else
         firstseed=.false.
      endif
```

If .not.fixedseed, it can reset seed from time almost each event

If fixedseed, it use only one random sequence (follows last event)

```
      function ran2(idum)
c
c Long period (>2E18) random number generator of L'Ecuyer with
c Bays-Durham shuffle and added safeguards. Returns a uniform random
c deviate between 0.0 and 1.0 (exclusive of the endpoint values).
c Call with idum a negative integer to initialize; thereafter, do
c not alter idum between successive deviates in a sequence. RNMX
c should approximate the largest floating value that is less than 1.
c
C  (C) Copr. 1986-92 Numerical Recipes Software.
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      implicit none

      integer idum,IM1,IM2,IMM1,IA1,IA2,IQ1,IQ2,IR1,IR2,NTAB,NDIV
      real*8 ran2,AM,EPS,RNMX

      parameter (IM1=2147483563,IM2=2147483399,AM=1.0D0/IM1,IMM1=IM1-1,
     $     IA1=40014,IA2=40692,IQ1=53668,IQ2=52774,IR1=12211,IR2=3791,
     $     NTAB=32,NDIV=1+IMM1/NTAB,EPS=1.0D0-16,RNMX=1.0D0-EPS)
      integer idum2, j, k, iv(NTAB), iy
      save iv, iy, idum2
      data idum2/123456789/, iv/NTAB*0/, iy/0/

      if (idum.le.0) then
         idum=max(-idum,1)
         idum2=idum
         do 11 j=NTAB+8,1,-1
            k=idum/IQ1
            idum=IA1*(idum-k*IQ1)-k*IR1
            if (idum.lt.0) idum=idum+IM1
            if (j.le.NTAB) iv(j)=idum
 11      continue
         iy=iv(1)
      endif
      k=idum/IQ1
      idum=IA1*(idum-k*IQ1)-k*IR1
      if (idum.lt.0) idum=idum+IM1
      k=idum2/IQ2
      idum2=IA2*(idum2-k*IQ2)-k*IR2
      if (idum2.lt.0) idum2=idum2+IM2
      j=1+iy/NDIV
      iy=iv(j)-idum2
      iv(j)=idum
      if (iy.lt.1) iy=iy+IMM1
      ran2=min(AM*iy,RNMX)

      return
      end
```

How to set seed for random generator

- Set seed for gRandom by random_device
  - True random number if supported by compiler
- Set c_use_external_seed = true

```
    std::random_device rd;
    gRandom->SetSeed(rd());

    c_sets_1_.c_use_external_seed = kTRUE;
```

- Set ranseed uniformly in [-2147483648, 2147483647]
  - Not finalized yet
  - Set ranseed only positive?
  - Add effect of time?

```
void c_set_seed_(){
    sys_.ranseed = (gRandom->Rndm()-0.5)*4294967296;
    seed_.oldseed = sys_.ranseed;
    seed_.iseed = -sys_.ranseed;
    ran2_(&seed_.iseed);
}
```

- Call ran2 to modify iseed

```
c     initialize random number generator
c     call auto-seed generator only for first event and if no seed was fixed
      if(c_use_external_seed)then
        if(firstseed)firstseed=.false.
        if(fixedseed)then
          write(6,*)'c_use_external_seed is neglected due to fixedseed'
          c_use_external_seed=.false.
        else
          call c_set_seed
        endif
      elseif(.not.firstseed.and.(.not.fixedseed)) then
         ranseed=-(1*abs(ranseed))
         call sseed(ranseed)
      else
         firstseed=.false.
      endif
```

Other developments

- Parent particle Information in collision output
  - In temporary quantities named t*
  - No spin, or Nin, rho_b values
    - Saved to my defined global quantity
    - Extraction method is the same as other t*

- Now time step tree filled after final particle decays rather than after final time step
  - Unstable particles' decays after all time steps & before the end of event
  - The same as origin output f14

- If CTOption(4)==1, initial information (t==0) is stored in tree
  - The same as origin output f14

Other developments

- Pdgid (int) –> mass (float), iType (short), iso3 (char), charge (char)
  - Strange masses of protons
  - Different masses between UrQMD and PDG
  - Some particles in UrQMD has no pdgid in ityp2pdg.f
    - Such as iType = 12, 13, 15, 16

```
c Neutron
     .       1, -1,  2112,
c Proton
     .       1,  1,  2212,
c N*
     .       2, -1, 12112,        2,  1, 12212,
     .       3, -1,  1214,        3,  1,  2124,
     .       4, -1, 22112,        4,  1, 22212,
     .       5, -1, 32112,        5,  1, 32212,
     .       6, -1,  2116,        6,  1,  2216,
     .       7, -1, 12116,        7,  1, 12216,
     .       8, -1, 21214,        8,  1, 22124,
     .       9, -1, 42112,        9,  1, 42212,
     .      10, -1, 31214,       10,  1, 32124,
     .      14, -1,  1218,       14,  1,  2128,
c Delta
     .      17, -3,  1114,  17, -1,  2114,  17, 1,  2214,  17, 3,  2224,
     .      18, -3, 31114,  18, -1, 32114,  18, 1, 32214,  18, 3, 32224,
     .      19, -3,  1112,  19, -1,  1212,  19, 1,  2122,  19, 3,  2222,
     .      20, -3, 11114,  20, -1, 12114,  20, 1, 12214,  20, 3, 12224,
     .      21, -3, 11112,  21, -1, 11212,  21, 1, 12122,  21, 3, 12222,
     .      22, -3,  1116,  22, -1,  1216,  22, 1,  2126,  22, 3,  2226,
     .      23, -3, 21112,  23, -1, 21212,  23, 1, 22122,  23, 3, 22222,
     .      24, -3, 21114,  24, -1, 22114,  24, 1, 22214,  24, 3, 22224,
     .      25, -3, 11116,  25, -1, 11216,  25, 1, 12126,  25, 3, 12226,
     .      26, -3,  1118,  26, -1,  2118,  26, 1,  2218,  26, 3,  2228,
```

```
data baryon_names/
.     'Nukleon',
.     'N(1440)',
.     'N(1520)',
.     'N(1535)',
.     'N(1650)',
.     'N(1675)',
.     'N(1680)',
.     'N(1700)',
.     'N(1710)',
.     'N(1720)',
.     'N(1900)',
.     'N(1990)',
.     'N(2080)',
.     'N(2190)',
.     'N(2220)',
.     'N(2250)',
```

Wrong name?

ityp	nucleon	ityp	delta	ityp	lambda	ityp	sigma	ityp	xi	ityp	omega
1	N_{938}	17	Δ_{1232}	27	Λ_{1116}	40	Σ_{1192}	49	Ξ_{1317}	55	Ω_{1672}
2	N_{1440}	18	Δ_{1600}	28	Λ_{1405}	41	Σ_{1385}	50	Ξ_{1530}		
3	N_{1520}	19	Δ_{1620}	29	Λ_{1520}	42	Σ_{1660}	51	Ξ_{1690}		
4	N_{1535}	20	Δ_{1700}	30	Λ_{1600}	43	Σ_{1670}	52	Ξ_{1820}		
5	N_{1650}	21	Δ_{1900}	31	Λ_{1670}	44	Σ_{1775}	53	Ξ_{1950}		
6	N_{1675}	22	Δ_{1905}	32	Λ_{1690}	45	Σ_{1790}	54	Ξ_{2025}		
7	N_{1680}	23	Δ_{1910}	33	Λ_{1800}	46	Σ_{1915}				
8	N_{1700}	24	Δ_{1920}	34	Λ_{1810}	47	Σ_{1940}				
9	N_{1710}	25	Δ_{1930}	35	Λ_{1820}	48	Σ_{2030}				
10	N_{1720}	26	Δ_{1950}	36	Λ_{1830}						
11	N_{1900}			37	Λ_{1890}						
12	N_{1990}			38	Λ_{2100}						
13	N_{2080}			39	Λ_{2110}						
14	N_{2190}										
15	N_{2200}										
16	N_{2250}										

Table 1: Baryon-itypes used in **UrQMD**. Antibaryons carry a negative sign.

ityp	0^{-+}	ityp	1^{--}	ityp	0^{++}	ityp	1^{++}	ityp	charmed
101	π	104	ρ	111	a_0	114	a_1	133	D
106	K	108	K^*	110	K_0^*	113	K_1^*	134	D^*
102	η	103	ω	105	f_0	115	f_1	135	J/Ψ
107	η'	109	ϕ	112	f_0^*	116	f_1'	136	χ_c
ityp	1^{+-}	**ityp**	2^{++}	**ityp**	$(1^{--})^*$	**ityp**	$(1^{--})^{**}$	137	Ψ'
122	b_1	118	a_2	126	ρ_{1450}	130	ρ_{1700}	138	D_s
121	K_1	117	K_2^*	125	K_{1410}^*	129	K_{1680}^*	139	D_s^*
123	h_1	119	f_2	127	ω_{1420}	131	ω_{1662}		
124	h_1'	120	f_2'	128	ϕ_{1680}	132	ϕ_{1900}		

What to be determined

- Npart: Aproject + Atarget − #(pptype%100==0)
  - ppType = origin%100
  - # scatters = origin/100%10

  - Some particles only explore scatters, contribute to Npart?

```
     2       2  17    233   1.554  0.5056E+01  0.3978E+02  0.1166E+02  0.8930E+00
362  0.15541405E+01 -0.22213285E+01  0.35829759E+01 -0.71742345E+00  0.17039231E+01 -0.17381395E+00  0.13815719E+01 -0.29078794E+00  0.93800002E+00         1 -1  0      213    1  0          100
233  0.15541405E+01 -0.27680167E+01  0.33953480E+01 -0.60064611E+00  0.85553163E+01  0.20642659E+00  0.13534976E+00 -0.85059975E+01  0.88346555E+00         1  1  1        0    0  0            0
362  0.15541405E+01 -0.22213285E+01  0.35829759E+01 -0.71742345E+00  0.84217759E+01  0.32168377E+01  0.25335456E-01 -0.83631539E+01  0.93800002E+00         1 -1  0      233    2  0          200
233  0.15541405E+01 -0.27680167E+01  0.33953480E+01 -0.60064611E+00  0.18374634E+01 -0.28907113E+00  0.14915862E+01 -0.43363154E+00  0.93800002E+00         1  1  1      233    1  0          100
```

Check

```
root [1] UrqmdCollisionDst->Show(0);
======> EVENT:0
 b               = 13.9877
 number          = 1
 time            = 0.0746903
 sqrts           = 199.043
 sigma_total     = 52.0758
 sigma_partial   = 10.3074
 rho_B           = 14.489
 pType           = 17
 Nin             = 2
 Nout            = 2
 Ntracks         = 4
 uid             = 73,
                   352, 73, 352
 iType           = 1,
                   1, 1, 1
 iso3            = 1,
                   -1, 1, -1
 charge          = 1,
                   0, 1, 0
 spin            = 1,
                   -1, -1, 1
 x               = 1.65141,
                   0.385643, 1.65141, 0.385643
 y               = 0.0844311,
                   0.0137725, 0.0844311, 0.0137725
 z               = 0.0112218,
                   0.0112218, 0.0112218, 0.0112218
 px              = 0.0141267,
                   0.0926628, 2.14204, -2.03525
 py              = 0.148645,
                   -0.105794, -0.724586, 0.767437
 pz              = 98.2774,
                   -100.773, -100.748, 98.2526
 m               = 0.917558,
                   0.919356, 0.938, 0.938
 Nc              = 0,
                   0, 1, 1
 pcNumber        = 0,
                   0, 1, 1
 ppType          = 0,
                   0, 100, 100
 iType_old1      = 0,
                   0, 0, 0
 iso3_old1       = 0,
                   0, 0, 0
 iType_old2      = 0,
                   0, 0, 0
 iso3_old2       = 0,
                   0, 0, 0
```

```
pvec:ind   r0              rx              ry              rz              p0              px              py              pz              m           ityp 2i3 chg lcl#  nc
       2       2  17       1   0.075  0.1990E+03  0.5208E+02  0.1031E+02  0.1449E+02
  73  0.74690319E-01  0.16514065E+01  0.84431142E-01  0.11221819E-01  0.98281818E+02  0.14126706E-01  0.14864452E+00  0.98277421E+02  0.91755845E+00          1  1  1       0   0  0             0
 352  0.74690319E-01  0.38564307E+00  0.13772507E-01  0.11221819E-01  0.10077680E+03  0.92662766E-01 -0.10579351E+00 -0.10077251E+03  0.91935619E+00          1 -1  0       0   0  0             0
  73  0.74690319E-01  0.16514065E+01  0.84431142E-01  0.11221819E-01  0.10077745E+03  0.21420381E+01 -0.72458558E+00 -0.10074771E+03  0.93800002E+00          1  1  1       1   1  0           100
 352  0.74690319E-01  0.38564307E+00  0.13772507E-01  0.11221819E-01  0.98281171E+02 -0.20352486E+01  0.76743659E+00  0.98252621E+02  0.93800002E+00          1 -1  0       1   1  0           100
```

Check

```
    2       3  28      4   0.085  0.6184E+02  0.6747E+01  0.5664E+01  0.9469E+01
412  0.84880709E-01 -0.10829742E+01 -0.40724994E+00  0.21507895E-01  0.52972894E+02 -0.22741829E+00  0.10696294E+00 -0.52958522E+02  0.12080407E+01        17 -1  0       3    1  0      996111015
400  0.84880709E-01 -0.89035344E+00 -0.70689933E+00  0.21920565E-01  0.18058629E+02 -0.39930945E+00 -0.12071740E+00  0.18010336E+02  0.12521293E+01       113 -1  0       2    1 -1      993311015
418  0.84880709E-01 -0.10829742E+01 -0.40724994E+00  0.21507895E-01  0.52971212E+02 -0.22741600E+00  0.10695713E+00 -0.52956287E+02  0.12320000E+01        17 -1  0       4    2  0     1699021028
419  0.84880709E-01 -0.89035344E+00 -0.70689933E+00  0.21920565E-01  0.73279741E+01 -0.28391694E+00 -0.31171549E+00  0.73145324E+01  0.13800000E+00       101 -2 -1       4    2  0     1699021028
420  0.84880709E-01 -0.89035344E+00 -0.70689933E+00  0.21920565E-01  0.10732337E+02 -0.11539479E+00  0.19100390E+00  0.10693570E+02  0.88364039E+00       108  1  1       4    1 -1     1699021028
```

```
root [3] UrqmdCollisionDst->Show(3);
======> EVENT:3
b               = 13.9877
number          = 4
time            = 0.0848807
sqrts           = 61.8361
sigma_total     = 6.74686
sigma_partial   = 5.6639
rho_B           = 9.46934
pType           = 28
Nin             = 2
Nout            = 3
Ntracks         = 5
uid             = 412,
                  400, 418, 419, 420
iType           = 17,
                  113, 17, 101, 108
iso3            = -1,
                  -1, -1, -2, 1
charge          = 0,
                  0, 0, -1, 1
spin            = 1,
                  -2, 3, 0, 2
x               = -1.08297,
                  -0.890353, -1.08297, -0.890353, -0.890353
y               = -0.40725,
                  -0.706899, -0.40725, -0.706899, -0.706899
z               = 0.0215079,
                  0.0219206, 0.0215079, 0.0219206, 0.0219206
px              = -0.227418,
                  -0.399309, -0.227416, -0.283917, -0.115395
py              = 0.106963,
                  -0.120717, 0.106957, -0.311715, 0.191004
pz              = -52.9585,
                  18.0103, -52.9563, 7.31453, 10.6936
m               = 1.20804,
                  1.25213, 1.232, 0.138, 0.88364
Nc              = 1,
                  1, 2, 2, 1
pcNumber        = 3,
                  2, 4, 4, 4
ppType          = 15,
                  15, 28, 28, 28
iType_old1      = 1,
                  1, 17, 17, 17
iso3_old1       = 1,
                  1, -1, -1, -1
iType_old2      = 1,
                  1, 113, 113, 113
iso3_old2       = 1,
                  -1, -1, -1, -1
```

Check

```
root [4] UrqmdCollisionDst->Show(10);
======> EVENT:10
b                = 13.9877
number           = 11
time             = 3.86413
sqrts            = 1.32149
sigma_total      = 0
sigma_partial    = 0
rho_B            = 0.00949667
pType            = 20
Nin              = 1
Nout             = 2
Ntracks          = 3
uid              = 447,
                   462, 463
iType            = 118,
                   101, 104
iso3             = 2,
                   2, 0
charge           = 1,
                   1, 0
spin             = -2,
                   0, 0
x                = -0.446549,
                   -0.446549, -0.446549
y                = -1.2298,
                   -1.2298, -1.2298
z                = -2.79132,
                   -2.79132, -2.79132
px               = 0.242734,
                   0.16513, 0.0776042
py               = -0.285845,
                   -0.169923, -0.115922
pz               = -1.54135,
                   -0.0961053, -1.44524
m                = 1.32149,
                   0.138, 1.0194
Nc               = 1,
                   2, 1
pcNumber         = 6,
                   11, 11
ppType           = 28,
                   20, 20
iType_old1       = 101,
                   118, 118
iso3_old1        = 0,
                   2, 2
iType_old2       = 1,
                   0, 0
iso3_old2        = -1,
                   0, 0
```

```
    1       2  20     11   3.864  0.1321E+01  0.0000E+00  0.0000E+00  0.9497E-02
447  0.38641318E+01 -0.44654879E+00 -0.12298002E+01 -0.27913244E+01  0.20646336E+01  0.24273440E+00 -0.28584481E+00 -0.15413463E+01  0.13214902E+01       118  2  1        6   1  0       993730028
462  0.38641318E+01 -0.44654879E+00 -0.12298002E+01 -0.27913244E+01  0.29055463E+00  0.16513024E+00 -0.16992283E+00 -0.96105291E-01  0.13800000E+00       101  2  1       11   2  0         1251020
463  0.38641318E+01 -0.44654879E+00 -0.12298002E+01 -0.27913244E+01  0.17740790E+01  0.77604157E-01 -0.11592198E+00 -0.14452410E+01  0.10193991E+01       104  0  0       11   1  0         1251020
```

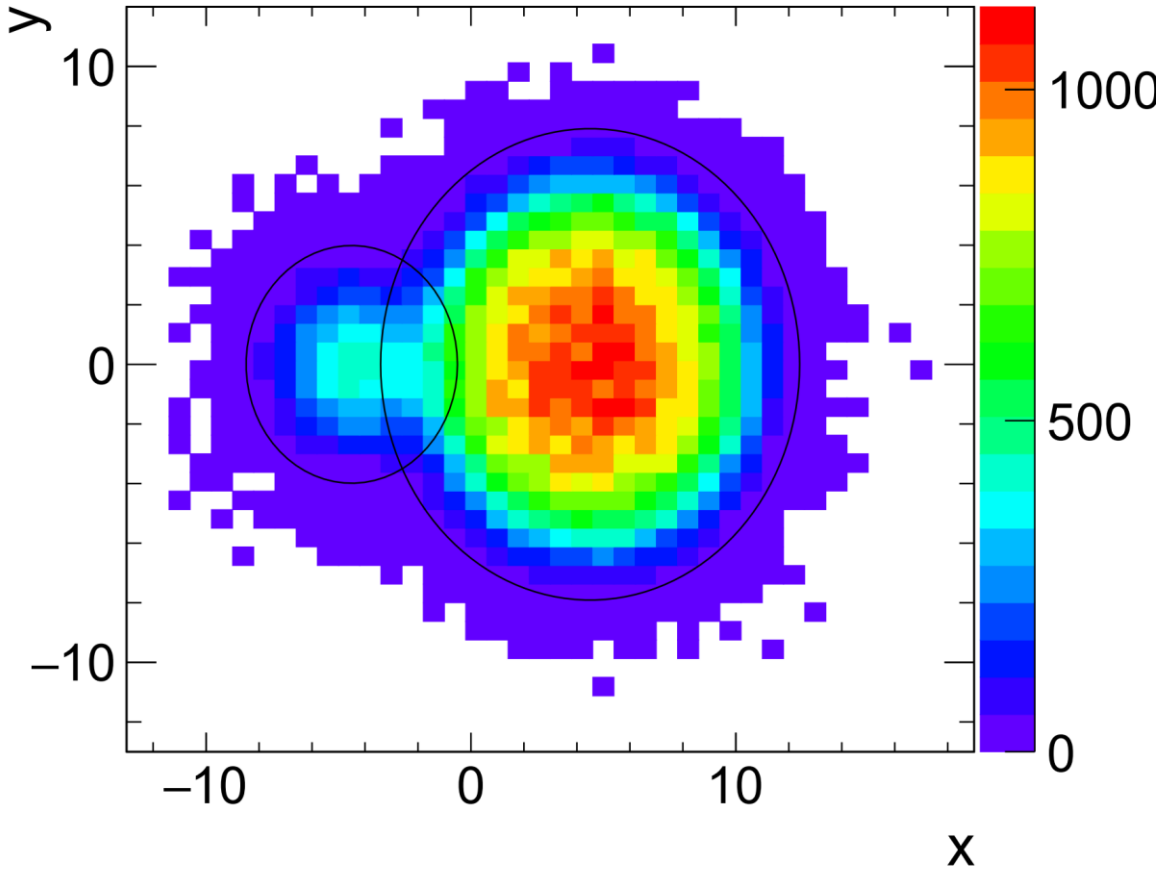
N_{part} calculation

- $(b_{max})_{max}$ = nucrad(Ap)+nucrad(At)+2*CTParam(30)
  - CTP(30): radius offset for initialization (1.5 default)
  - ^{197}Au+^{197}Au: 15.82 = 6.41+6.41+1.5*2

```
      function nucrad(AA)
      implicit none
      real*8 nucrad, r_0
      integer A,AA
      include 'coms.f'
      include 'options.f'

      A=abs(AA)/CTParam(67)

c root mean square radius of nucleus of mass A
c r_0 corresponding to rho0
      if (CTOption(24).eq.1) then
c root mean square radius of nucleus of mass A (Mayer-Kuckuck)
         nucrad = 1.128 * a**(1./3.) - 0.89 * a**(-(1./3.))
      else
         r_0 = (0.75/pi/rho0)**(1./3.)
c subtract gaussian tails, for distributing centroids correctly
         nucrad = r_0*(0.5*(a + (a**(1./3.)-1.)**3.))**(1./3.)
      endif

      return
      end
```

N_{part} calculation

- pro: 197 79
- tar: 16 8
- $\sqrt{s_{\text{NN}}}$ = 200 GeV
- b = 9 fm
- t = 0

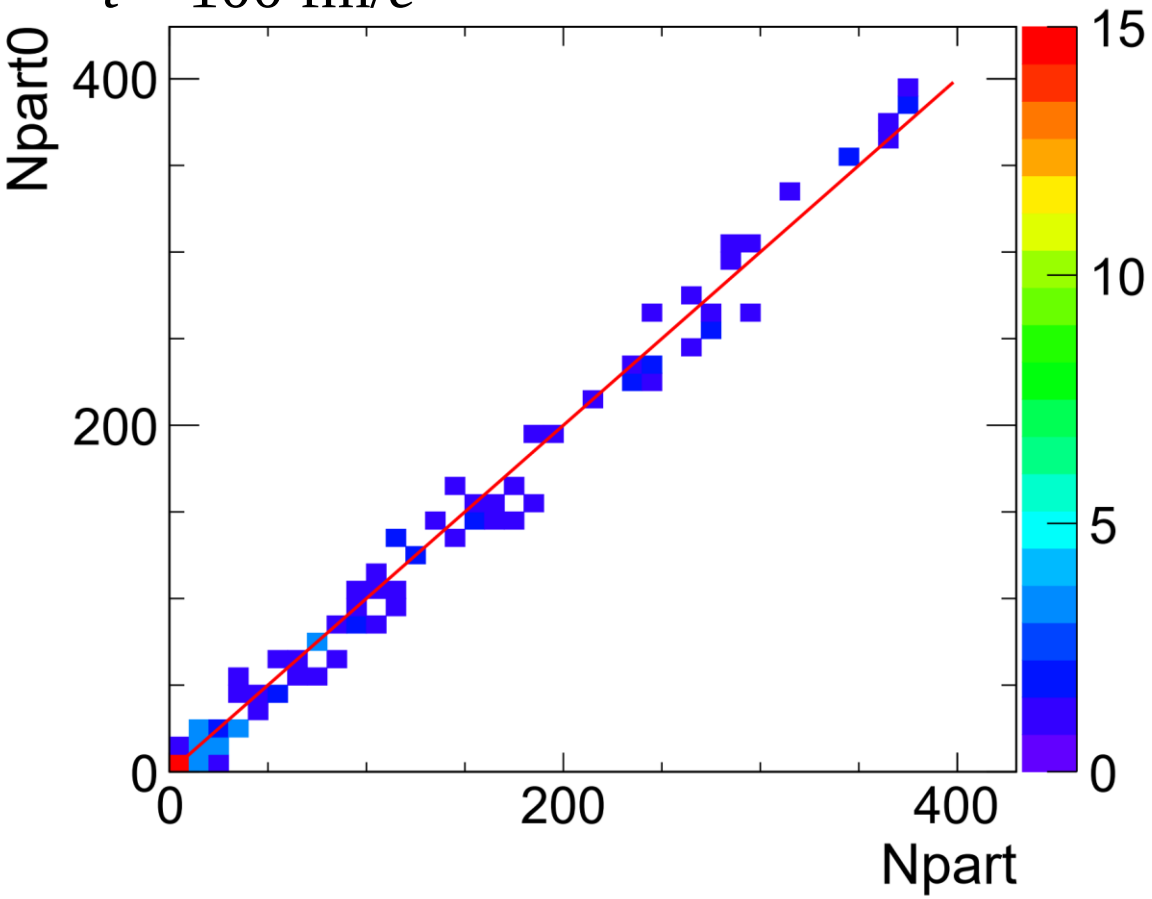


- Center: pro (b/2, 0), tar (−b/2, 0)
- Radius: nucrad(A)+CTParam(30)
  - But CTP(30) only contributes to $(b_{\text{max}})_{\text{max}}$ rather than nuclei initialization

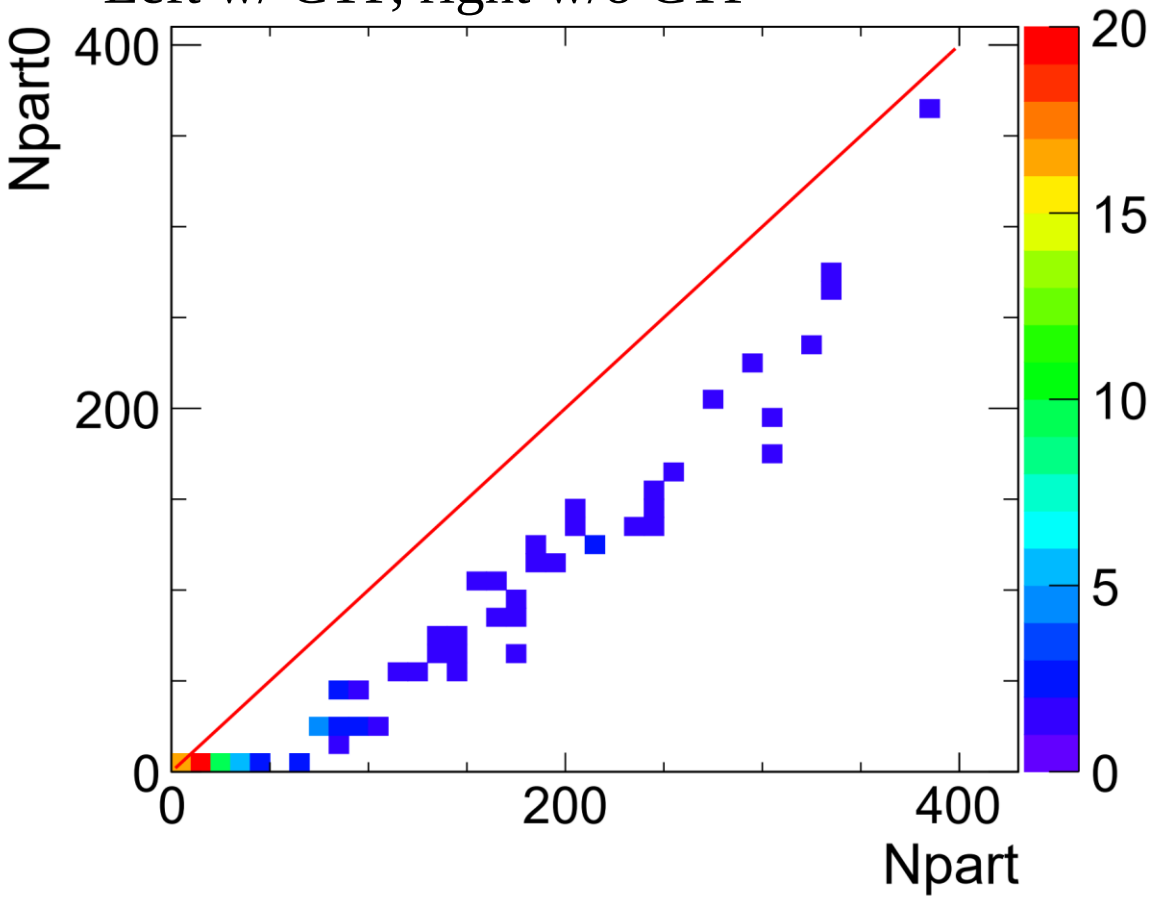
- N_{part} = #($(x-b/2)^2+y^2 < R(\text{pro})$ && $(x+b/2)+y^2 < R(\text{tar})$) at t = 0

N_{part} calculation

- $^{197}Au+^{197}Au$, $0 <= b < (b_{max})_{max}$
- $\sqrt{s_{NN}}$ = 200 GeV
- t = 100 fm/c



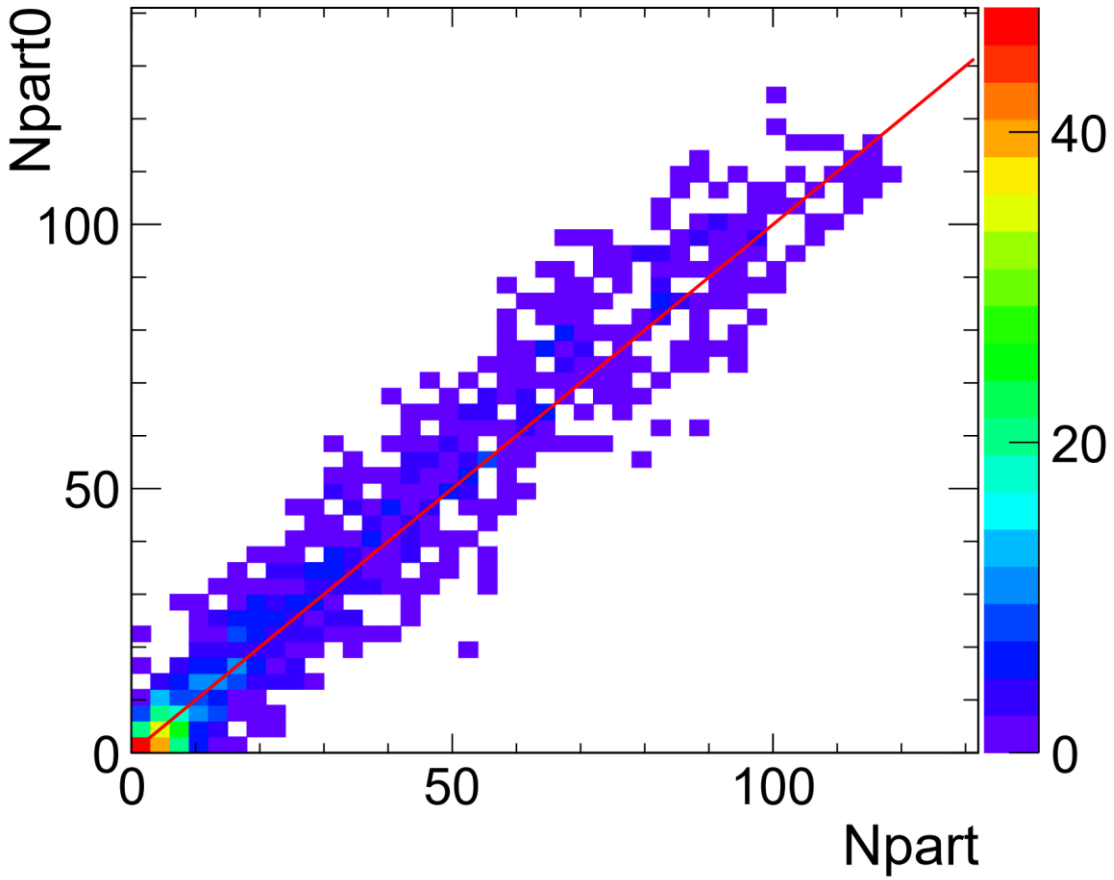
- Npart: old definition
- Npart0: new definition
  - Left w/ CTP, right w/o CTP

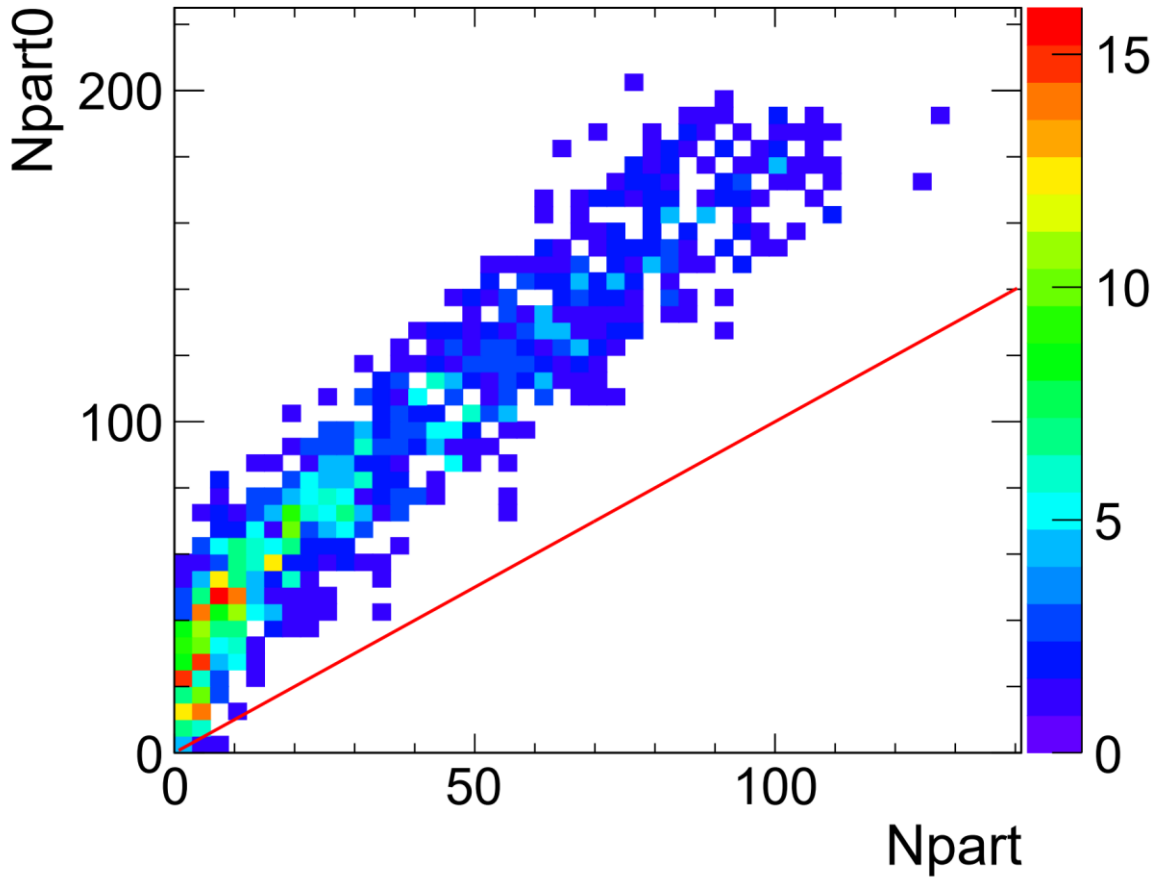


N_{part} calculation

- $^{197}Au+^{16}O$, $0 <= b < (b_{max})_{max}$
- $\sqrt{s_{NN}}$ = 200 GeV
- t = 100 fm/c

- Left: CTP(30) = 1.5 (default)
- Right: CTP(30) = 4





N_{part} calculation

```
      dstp=dstp+CTParam(30)
      dstt=dstt+CTParam(30)
         if(bdist.gt.(nucrad(Ap)+nucrad(At)+2*CTParam(30)))
     &        bdist=nucrad(Ap)+nucrad(At)+2*CTParam(30)
         if(bdist.gt.(nucrad(Ap)+nucrad(At)+2*CTParam(30)))
     &        bdist=nucrad(Ap)+nucrad(At)+2*CTParam(30)
      CTParam(30)=1.5
```

```
      real*8 nucrad,dstt,dstp,pcm,eb,embeam,emtarget
            dstp = nucrad(Ap)+CTParam(41)
            dstp = nucrad(Ap)*ratio**(2.0/3.0)+CTParam(41)
         dstp=dstp+CTParam(30)
c         dst=0.5d0*(dstt+dstp)
            dstp=0d0
     &                    pboost,0.5*bimp,-dstp)
     &                pboost,0.5*bimp,-dstp+CTParam(20))
                   call boostnuc(npart,npart,-pbeam,0.5*bimp,-dstp)
                   call boostnuc(npart,npart,0.d0,0.5*bimp,-dstp)
                   call boostnuc(npart,npart,-ppeq,0.5*bimp,-dstp)
      real*8 nucrad,dstt,dstp,pcm,eb,embeam,emtarget
```

```
      subroutine boostnuc(i1,i2,pin,b,dst)
      implicit none
      include 'coms.f'
      include 'options.f'
      integer i1,i2,i
      real*8 b,dst,ei,ti
      real*8 pin,beta,gamma

      do 1 i=i1,i2

      beta = pin/sqrt(pin**2+fmass(i)**2)
      gamma = 1.d0/sqrt(1.d0-beta**2)

c  Gallilei-Trafo in x-direction (impact parameter)
c  projectile hits at POSITIVE x
         rx(i) = rx(i) + b
c  distance between nuclei: projectile at NEGATIVE z for dst < 0
         if(CTOption(23).eq.0)then
           ti = r0(i)
           rz(i) = rz(i)/gamma+dst/gamma
         else
          rz(i) = (rz(i) + dst)
         end if


         Ei = p0(i)
         p0(i) = gamma*(p0(i) - beta*pz(i))
         pz(i) = gamma*(pz(i) - beta*Ei)

 1    continue
      return
      end
```

N_{part} calculation

24	1	initialization mode
	0	hard sphere (used for EOS≠0)
	1	Woods-Saxon (used for CASCADE mode)
	2	Fast Woods-Saxon (used for CASCADE mode)

```
if (CTOption(24).eq.1) then
  R2 = nucrad(A) + 10.0
else
  R2 = nucrad(A)
endif
```

```
subroutine nucfast(IA,JJ)

Performs very fast initialisation of nuclei
Adopted from QGSJET (S. Ostapchenko)
ccccccccccccccccccccccccccccccccccccccccccccccc
IMPLICIT DOUBLE PRECISION (A-H,O-Z)

include 'inputs.f'
real*8 nucrad

am=0.545
rad=nucrad(ia)/am
```

Classes to store UrQMD data

- UrqmdDst
  - TClonesArray*[4] containing the following classes
  - initOutputTree: returns TTree for writing
  - initInputChain: reads input file (*.list, *.root), and returns TChain for analyzing
    - chain->GetEntry(i);
    - urqmdDst-> …
- UrqmdEvent
- UrqmdTimestep
- UrqmdCollision
- UrqmdTrack

```
class UrqmdDst{
  public:
    enum {Event = 0, Timestep, Collision, Track, NUrqmdArrays};
    static const TString UrqmdArrayNames[NUrqmdArrays];
    static const TString UrqmdArrayTypes[NUrqmdArrays];
    static const UInt_t UrqmdArraySizes[NUrqmdArrays];

    UrqmdDst();
    virtual ~UrqmdDst();

    UrqmdEvent* event(){return (UrqmdEvent*)mUrqmdArrays[Event]->UncheckedAt(0);}
    UrqmdTimestep* timestep(UInt_t i){return (UrqmdTimestep*)mUrqmdArrays[Timestep]->UncheckedAt(i);}
    UrqmdCollision* collision(UInt_t i){return (UrqmdCollision*)mUrqmdArrays[Collision]->UncheckedAt(i);}
    UrqmdTrack* track(UInt_t i){return (UrqmdTrack*)mUrqmdArrays[Track]->UncheckedAt(i);}
    UInt_t numberOfTimesteps(){return mUrqmdArrays[Timestep]->GetEntriesFast();}
    UInt_t numberOfCollisions(){return mUrqmdArrays[Collision]->GetEntriesFast();}
    UInt_t numberOfTracks(){return mUrqmdArrays[Track]->GetEntriesFast();}

    TTree* initOutputTree(const TString outputTreeName = "urqmd");
    TChain* initInputChain(const TString inputFileName, const TString inputTreeName = "urqmd");

  protected:
    TClonesArray* mUrqmdArrays[NUrqmdArrays];

#ifdef __ROOT__
    ClassDef(UrqmdDst, 1)
#endif
};
```

Classes to store UrQMD data

- UrqmdEvent
  - id: event id
    - 0 ~ 4294967295
  - b: impact parameter
    - 0 ~ 26.214

```
class UrqmdEvent : public TObject{
  public:
    enum {BScale = 2500};

    UrqmdEvent() : mId(0), mB(0){}
    UrqmdEvent(const UInt_t id, const Double_t b) : mId(id), mB(TMath::Nint(b*Double_t(BScale))){
    virtual ~UrqmdEvent(){}

    UInt_t id() const{return mId;}
    Double_t b() const{return mB/Double_t(BScale);}

    void print() const;
    void set(const UInt_t id, const Double_t b);

  protected:
    UInt_t mId;
    UShort_t mB;

#ifdef __ROOT__
    ClassDef(UrqmdEvent, 1)
#endif
};
```

Classes to store UrQMD data

- UrqmdTimestep
  - time: Float16_t (24 bit, truncated mantissa)
  - nPart: old definition
  - nColl = Elastic + Inelastic + Bloacked
  - nElasticColl
  - nInelasticColl
  - nBloackedColl
  - nDecays
  - nHardRes
  - nSoftRes
  - nDecayRes
  - nTracks
  - startTrack: array index of 1st track at this timestep

```
class UrqmdTimestep : public TObject{
  public:
    UrqmdTimestep() : mTime(0.), mNPart(0), mNElasticColl(0), mNInelasticColl(0),
    UrqmdTimestep(const Double_t time, const UInt_t nPart, const UInt_t nElasticCo
    virtual ~UrqmdTimestep(){}

    Double_t time() const{return mTime;}
    UInt_t nPart() const{return mNPart;}
    UInt_t nColl() const{return nElasticColl()+nInelasticColl()+nBloackedColl();}
    UInt_t nElasticColl() const{return mNElasticColl;}
    UInt_t nInelasticColl() const{return mNInelasticColl;}
    UInt_t nBloackedColl() const{return mNBloackedColl;}
    UInt_t nDecays() const{return mNDecays;}
    UInt_t nHardRes() const{return mNHardRes;}
    UInt_t nSoftRes() const{return mNSoftRes;}
    UInt_t nDecayRes() const{return mNDecayRes;}
    UInt_t nTracks() const{return mNTracks;}
    UInt_t startTrack() const{return mStartTrack;}

    void print() const;
    void set(const Double_t time, const UInt_t nPart, const UInt_t nElasticColl, c

  protected:
    Float16_t mTime;
    UShort_t mNPart;
//  UShort_t mNColl;
    UShort_t mNElasticColl;
    UShort_t mNInelasticColl;
    UShort_t mNBloackedColl;
    UShort_t mNDecays;
    UShort_t mNHardRes;
    UShort_t mNSoftRes;
    UShort_t mNDecayRes;
    UShort_t mNTracks;
    UInt_t mStartTrack;

#ifdef __ROOT__
    ClassDef(UrqmdTimestep, 1)
#endif
};
```

Classes to store UrQMD data

- UrqmdCollision: ~ 20000 for 200 GeV, $b = 0$, $t = 50$ fm/c
  - time
  - Sqrts
  - sigmaTotal
  - sigmaPartial
  - rhoB: −1 ~ 30+
  - Type: < 100
  - nIn: < 2?
  - nOut
  - nTracks = nIn + nOut
  - startTrack
- An empty collision at index 0, so index = collision id

1,1 wall interaction (infinite matter calculations)
1,*N* > 1 decay
2,1 annihilation
2,2 scattering
2,*N* > 2 string excitation and decay
1,0 Pauli-blocked decay
2,0 Pauli-blocked scattering
N,**0** Fluidization: All particles at the beginning of the fluid-dynamical part (Process-ID "91"). **Please Note:** this also includes spectators and corona-particles that are not transferred to the fluid!

0;*N* Particlization: All particles at the end of the fluid-dynamical part (Process-ID "96"). Here, t_{coll}

```
class UrqmdCollision : public TObject{
  public:
    UrqmdCollision() : mTime(0.), mSqrts(0.), mSigmaTotal
    UrqmdCollision(const Double_t time, const Double_t sq
    virtual ~UrqmdCollision(){}

    Double_t time() const{return mTime;}
    Double_t sqrts() const{return mSqrts;}
    Double_t sigmaTotal() const{return mSigmaTotal;}
    Double_t sigmaPartial() const{return mSigmaPartial;}
    Double_t rhoB() const{return mRhoB;}
    UInt_t type() const{return mType;}
    UInt_t nIn() const{return mNIn;}
    UInt_t nOut() const{return mNOut;}
    UInt_t nTracks() const{return nIn()+nOut();}
    UInt_t startTrack() const{return mStartTrack;}

    void print() const;
    void set(const Double_t time, const Double_t sqrts, c

  protected:
//  UInt_t mId;
    Float16_t mTime;
    Float16_t mSqrts;
    Float16_t mSigmaTotal;
    Float16_t mSigmaPartial;
    Float16_t mRhoB;
    UChar_t mType;
    UChar_t mNIn;
    UChar_t mNOut;
    UInt_t mStartTrack;

#ifdef __ROOT__
    ClassDef(UrqmdCollision, 1)
#endif
};
```

Classes to store UrQMD data

- UrqmdTrack: ~86000 for 200 GeV, b = 0, t = 50 fm/c (collision+1timestep)
  - mType = (shifted_itype+100)+200((iso3+3)+7((charge+2)+5(spin+3)))
    - |itype| >= 140, shifted_itype = 0 (PdgId+1000 used for some c/b hadrons)
    - |itype| >= 100, shifted_itype = itype−40*|itype|/itype
    - iso3 = $2I_3$, $-3 \sim 3$
    - charge, $-2 \sim 2$
    - spin = $2s_3$?, $-3 \sim 3$
  - m: 0 − 6.5535
  - mOrigin
    - parentCollisionType, itypeOld1, iso3Old1, itypeOld2, iso3Old2
  - rSigma: 0 − 1
- If mTimestep=mCollision=0
  - √ ×

```
UInt_t id() const{return mId;}
UInt_t timestep() const{return mTimestep;}
UInt_t collision() const{return mCollision;}
Int_t itype() const;
Int_t iso3() const{return mType/200%7-3;}
Int_t charge() const{return mType/1400%5-2;}
Int_t spin() const{return mType/7000-3;}
Double_t x() const{return mX;}
Double_t y() const{return mY;}
Double_t z() const{return mZ;}
Double_t px() const{return mPx;}
Double_t py() const{return mPy;}
Double_t pz() const{return mPz;}
Double_t m() const{return mM/Double_t(MScale);}
UInt_t nCollisions() const{return mNCollisions;}
UInt_t parentCollision() const{return mParentCollision;}
UInt_t parentCollisionType() const{return mOrigin%1000;}
Int_t itypeOld1() const;
Int_t iso3Old1() const{if(mOrigin<1000) return 0; return mOrigin/200000%7-3;}
Int_t itypeOld2() const;
Int_t iso3Old2() const{if(mOrigin<1400000) return 0; return mOrigin/280000000-3;}
Double_t xFreezeOut() const{return mXFreezeOut;}
Double_t yFreezeOut() const{return mYFreezeOut;}
Double_t zFreezeOut() const{return mZFreezeOut;}
Double_t pxFreezeOut() const{return mPxFreezeOut;}
Double_t pyFreezeOut() const{return mPyFreezeOut;}
Double_t pzFreezeOut() const{return mPzFreezeOut;}
Double_t tauDecay() const{return mTauDecay;}
Double_t tauForm() const{return mTauForm;}
Double_t rSigma() const{return mRSigma/Double_t(RSigmaScale);}
```

```
  enum {MScale = 10000};
  enum {RSigmaScale = 250};
protected:
  UInt_t mId;
  UChar_t mTimestep;
  UShort_t mCollision;
  UShort_t mType;
//  Short_t mItype;
//  Char_t mIso3;
//  Char_t mCharge;
//  Char_t mSpin;
  Float16_t mX;
  Float16_t mY;
  Float16_t mZ;
  Float16_t mPx;
  Float16_t mPy;
  Float16_t mPz;
  UShort_t mM;
  UChar_t mNCollisions;
  UShort_t mParentCollision;
  UInt_t mOrigin;
//  UChar_t mParentCollisionType;
//  Short_t mItypeOld1;
//  Char_t mIso3Old1;
//  Short_t mItypeOld2;
//  Char_t mIso3Old2;
  Float16_t mXFreezeOut;
  Float16_t mYFreezeOut;
  Float16_t mZFreezeOut;
  Float16_t mPxFreezeOut;
  Float16_t mPyFreezeOut;
  Float16_t mPzFreezeOut;
  Float16_t mTauDecay;
  Float16_t mTauForm;
  UChar_t mRSigma;
```

Other updates

- Random seed (Int_t) =
  - (Int_t((gRandom->Rndm()-0.5)*0x100000000))^(Int_t(ULong64_t(gSystem->Now()))%0x10000)
    - gRandom->Rndm(): −1 ~ 1, ULong64_t(gSystem->Now()): current time in milliseconds
- All input settings summarized together
  - New input parameter: event ID offset
- Extended track information extracted for collision history
- GNUmakefile updated, rootcint compiler called
  - ROOT IO functions required by UrqmdDst

- New Npart definition in UrqmdEvent?
- Define 1 or 2 switches for some infrequently used track information?
  - Split UrqmdTrack to 2 or 3 classes

Result

- 10 events, 200 GeV, b = 0, t = 50 fm/c (collision+1timestep)
  - 32 MB
  - ~20000 collisions, ~86000 tracks
    - Generally, 3~4 tracks per collision (decay 3, collision 4)
    - If storeCollision is turned off, required space is much smaller

```
root [0] gSystem->Load("UrqmdDst/UrqmdDst.so");
root [1] UrqmdDst* u=new UrqmdDst();
root [2] TChain* c=u->initInputChain("out.root");
Read in root file out.root .
Total 1 file has been read in.
root [3] c->GetEntries()
(long long) 10
root [4] c->GetEntry(0);
root [5] u->event()->b()
(double) 0.0000000
root [6] u->numberOfTimesteps()
(unsigned int) 1
root [7] u->numberOfCollisions()
(unsigned int) 20629
root [8] u->numberOfTracks()
(unsigned int) 86390
root [9] u->timestep(0)->startTrack()
(unsigned int) 76058
root [10] u->track(76057)->collision()
(unsigned int) 20628
```

```
root [0] gSystem->Load("UrqmdDst/UrqmdDst.so");
root [1] UrqmdDst* u=new UrqmdDst();
root [2] TChain* c=u->initInputChain("test.list");
Read in root file out.root .
Total 1 file has been read in.
```