The UrQMD Frankfurt Nantes Duke user guide

December, 9th 2009

Warning: This document is updated regularly. In its current form it describes the handling of UrQMD **revision** 3.3. If you are using a different version of UrQMD, please obtain the manual from http://urqmd.org/documentation
The authors give no warranty to the correct functioning of the UrQMD program. Use this program at your own risk. Please send all bug–reports to the following e-mail address:

urqmd@urqmd.org and a copy to
bleicher@th.physik.uni-frankfurt.de

General Information

The Ultra Relativistic Quantum Molecular Dynamics (UrQMD) model is a transport model for simulating heavy ion collisions in the energy range from SIS to RHIC (use at LHC is at your own risk). It runs on various UNIX-based computing platforms. Current implementations include IBM/AIX (xlf), GNU/Linux (gfortran, ifc), SGI/IRIX, DEC-UNIX and Sun/Solaris.
UrQMD is designed as multipurpose tool for studying a wide variety of heavy ion related effects ranging from multifragmentation and collective flow to particle production and correlations. For hard pQCD scatterings, the model includes the PYTHIA routines from the LUND group.
This document is no introduction to the physics of UrQMD. Its purpose is to serve as a short guide to the experienced physicist on how to run the program. A detailed model description can be found in the following two articles.

1. *Microscopic Models for Ultrarelativistic Heavy Ion Collisions*
   S. A. Bass, M. Belkacem, M. Bleicher, M. Brandstetter, L. Bravina, C. Ernst, L. Gerland, M. Hofmann, S. Hofmann, J. Konopka, G. Mao, L. Neise, S. Soff, C. Spieles, H. Weber, L. A. Winckelmann, H. Stöcker, W. Greiner, C. Hartnack, J. Aichelin and N. Amelin.
   Prog. Part. Nucl. Phys. **41** (1998) 225–370.

2. *Relativistic Hadron-Hadron Collisions and the Ultra-Relativistic Quantum Molecular Dynamics Model (UrQMD)*
   M. Bleicher, E. Zabrodin, C. Spieles, S.A. Bass, C. Ernst, S. Soff, H. Weber, H. Stöcker and W. Greiner.
   J. Phys. **G25** (1999), 1859–1896.

For the hybrid model including an ideal hydrodynamic evolution for the hot and dense stage please refer to

1. *Fully integrated transport approach to heavy ion reactions with an intermediate hydrodynamic stage*
   H. Petersen, J. Steinheimer, G. Burau, M. Bleicher and H. Stöcker.
   Phys. Rev. C **78** (2008) 044901.

The hybrid calculations are only tested and give reasonable results in the energy range from $E_{\rm lab} = 2A - 160A$ GeV.
The hydrodynamic evolution is carried out by the Smooth and Sharp Transport Algorithm (SHASTA) in its implementation for relativistic heavy ion collisions as it is decribed in

1. *Relativistic hydrodynamics for heavy ion collisions. 1. General aspects and expansion into vacuum*
   D. H. Rischke, S. Bernard and J. A. Maruhn.
   Nucl. Phys. A **595** (1995) 346.

2. *Relativistic hydrodynamics for heavy ion collisions. 2. Compression of nuclear matter and the phase transition to the quark - gluon plasma*
   D. H. Rischke, Y. Pursun and J. A. Maruhn.
   Nucl. Phys. A **595** (1995) 383.

For the different equations of state during the hydrodynamic evolution the reader is refered to the following references:

1. Hadron Gas:
   Particle ratios at RHIC: Effective hadron masses and chemical freeze-out
   D. Zschiesche, S. Schramm, J. Schaffner-Bielich, H. Stöcker and W. Greiner.
   Phys. Lett. B **547**, 7 (2002).

2. Bag Model:
   Relativistic hydrodynamics for heavy ion collisions. 2. Compression of nuclear matter and the phase transition to the quark - gluon plasma
   D. H. Rischke, Y. Pursun and J. A. Maruhn.
   Nucl. Phys. A **595**, 383 (1995).

3. Chiral EoS:
   (3+1)-Dimensional Hydrodynamic Expansion with a Critical Point from Realistic Initial Conditions
   J. Steinheimer, M. Bleicher, H. Petersen, S. Schramm, H. Stöcker and D. Zschiesche.
   Phys. Rev. C **77**, 034901 (2008).

User support

UrQMD users are encouraged to join the user@urqmd.org e-mail list and post questions concerning UrQMD to this list. When registering as UrQMD user at http://urqmd.org/, subscription to the list will happen automatically.
For further questions and bug reports, the UrQMD-developers can be contacted via urqmd@urqmd.org

Copyright

UrQMD source and documentation are provided freely for the purpose of checking and reproducing published results of the authors.
The Open Standard Codes and Routines (OSCAR)-Group has established - for good reasons - guidelines for reproducablity, usage and quality control of simlulations codes for pA and AA collisions.
UrQMD is a complex model. In order to ensure that it is used correctly, that all results are reproducible and that the proper credits are given we ask for your agreement to the following copyright and safeguard mechanisms in the OSCAR spirit.
The UrQMD collaboration favors cooperation and joint projects with outside researchers. We encourage experimental collaborations to compare their results to UrQMD. We support you and/or cooperate on any sensible project related to UrQMD.
If you are interested in a project, please contact us.
Projects without the participation of the UrQMD-Collaboration are accepted, if the project is not a current thesis topic of any UrQMD-Collaboration member.

We expect that the code authors are informed about any changes and modifications made to the code. Any changes to the official version must be documented.
The only official source for the UrQMD **program** is the web page http://urqmd.org.
The code or any fragments of it shall not be given away to third parties. Similarily, events generated with UrQMD shall not be given to third parties without consent of the code authors at Frankfurt.

Compiling and running the program

To compile UrQMD one needs a FORTRAN77 compiler and GNU-make. The GNU-make programm is available on ftp.gnu.org (note: on many old UNIX systems GNU-make is called gmake). Compilation is initiated by issuing the make command at the command-prompt in the UrQMD subdirectory. After successful compilation the binary has the name urqmd.ARCH where ARCH is the machine type as given by uname -m. For further possibilities of using make with UrQMD, type (g)make help.
In order to run UrQMD one needs to define the running parameters with an input file. The input file is made accessible to UrQMD by attaching its name to the environment-variable ftn09. The output files are attached in the same fashion via the environment variables ftn14 and ftn15. Figure 1 shows, how UrQMD is started on a generic UNIX system (here Linux using the Bash Shell). A sample file (runqmd.bash) is provided:

```
export ftn09=inputfile
export ftn13=outputfile_with_freezeout
export ftn14=outputfile
export ftn15=collisionfile
export ftn16=outputfile_with_decaying_resonances
export ftn19=outputfile_for_OSCAR97
export ftn20=outputfile_for_OSCAR99
urqmd.$(uname -m)
```

Figure 1: running the UrQMD program

The input file

Figure 2 shows a typical input–file for UrQMD. The general format of the inputfile is 1A3,1A77. This means that every input line consists of two sections: First a three character flag followed by a 77 character string, the contents of which varies according to the flag specified. A sample inputfile (inputfile) is included.

```
# this is a sample input file for urqmd
#  projectile
#    Ap Zp
pro  197 79
# optional: special projectile: ityp, iso3
# PRO 101 2
#  target
#    At Zt
tar  197 79
#  number of events
nev 10
#  time to propagate and output time-interval (in fm/c)
tim  200 200
#
#   incident beam energy in AGeV
elb  160.0
#
#   weighted impact parameter distribution (from 0-3 fm)
imp  -3.0
#
#   equation of state
eos  0        # CASCADE mode
#   some options and parameters
cto  4  1     # output of initialization
ctp  1  1.d0    # scaling for decay width of Resonances
#
f15  # no output to file15
#  end of file
xxx
```

Figure 2: sample input file for UrQMD

The input–file does not have a predefined sequence. However, it is mandatory that the input contains definitions for projectile, target, impact parameter and incident beam energy.
Table 1 shows a quick summary of all possible flags with their respective parameters.

label	arguments	description
#		comment line
xxx		last line of input–file
pro	Ap Zp	define projectile
PRO	ityp iso3	define special projectile
tar	At Zt	define target
TAR	ityp iso3	define special target
nev	nevents	number of events to calculate
tim	tottime outtime	define time of calculation and output
ene	ebeam	incident kinetic beam energy (lab frame)
elb	ebeam	incident kinetic beam energy (lab frame)
plb	pbeam	incident beam momentum (lab frame)
PLB	pmin pmax npbin	incident (min/max) beam momentum for excitation function
PLG	pmin pmax npbin	like PLB, log-weighted
ecm	srt	$\sqrt{s}$ for two particle collision
ENE	srtmin srtmax nsrt	incident min/max $\sqrt{s}$ for excitation function
ELG	srtmin srtmax nsrt	incident min/max $\sqrt{s}$ for excitation function (log-weighted)
imp	bmax	define impact parameter (bmin=0)
IMP	bmin bmax	define impact parameter
eos	EoS	define equation of state
box	dim edens solid para	define box for infinite matter calculation
bpt	ityp iso3 npart pmax	define particle population for box-mode
bpe	ityp iso3 npart	like bpt, for given energy density
rsd	seed	seed for random number generator
stb	ityp	keep particle stable
cdt	deltat	Δt between full collision load
f13		suppress output to unit 13
f14		suppress output to unit 14
f15		suppress output to unit 15
f16		suppress output to unit 16
f19		suppress output to unit 19
f20		suppress output to unit 20
ctp	index value	set optional parameter in CTParam array
cto	index value	set option in CTOption array

Table 1: possible flags in the input–file with their respective arguments

Input Parameters

In this section all input labels with their respective arguments are explained. A complete sample input file can be seen in figure 2.

comment

```
#  string
```

`string` can be used to insert comments into the input file.

end of file

```
xxx string
```

`string` should contain at least one blank, `xxx` marks the end of the input file. On some systems it might be necessary to add an additional empty line after the `xxx`.

define projectile

```
pro Ap Zp
PRO ityp iso3
```

`Ap` mass of projectile

`Zp` charge of projectile

Instead of defining an ordinary nucleus (with `pro` one can also define a special non-composite projectile with `PRO`:

`ityp` ID of projectile (see tables 2 and 3 for available itypes)

`iso3` $2 \cdot Isospin_3$ of particle

define target

```
tar Ap Zp
TAR ityp iso3
```

`Ap` mass of target

`Zp` charge of target

Instead of defining an ordinary nucleus (with `tar` one can also define a special non-composite target with `TAR`:

`ityp` ID of target (see tables 2 and 3 for available itypes)

`iso3` $2 \cdot Isospin_3$ of particle

define number of events

```
nev nevents
```

`nevents` number of events to calculate

define calculation time (in fm/c)

```
tim tottime outtime
```

`tottime` total time span (in fm/c) to calculate

`outtime` time interval (in fm/c) after which output is written to files 13 and 14

define incident beam energy (in GeV)

```
ene ebeam
elb ebeam
ecm srts
ENE srtmin srtmax nsrt
ELG srtmin srtmax nsrt
plb pbeam
PLB pmin pmax npbin
PLG pmin pmax npbin
```

`ebeam` kinetic energy of the beam-particle (in case of nuclei it is the energy per nucleon) in the laboratory frame

`srt` $\sqrt{s_{NN}}$ between projectile and target (in case of nuclei it is the energy per nucleon pair)

`srtmin` minimal value for $\sqrt{s_{NN}}$ between projectile and target particles (in case of nuclei it is the energy per nucleon)

`srtmax` maximal value for $\sqrt{s_{NN}}$ between projectile and target particles (in case of nuclei it is the energy per nucleon)

`nsrt` number of $\sqrt{s_{NN}}$ values from `srtmin` to `srtmax` for which events shall be calculated (excitation function)

`pbeam` momentum of the beam-particle (in case of nuclei it is the momentum per nucleon)

`pmin` minimal value for p_{lab} (for excitation functions)

`pmax` maximal value for p_{lab} (for excitation functions)

`npbin` number of p_{lab} values from `pmin` to `pmax` for which events shall be calculated (excitation function)

For single momenta/energies the definitions `ene`/`elb, ecm` and `plb` are used, for excitation functions `ENE, ELG, PLB, PLG` are needed. The binning of the excitation function is linear for `ENE` and `PLB` and logarithmic for `ELG` and `PLG`. In the case of an excitation function the number of events `nev` refers to the **full** excitation function, i.e. the number of events per bin would be `nev`/`nsrt` or `nev`/`npbin` respectively.
Only one of the above seven definitions must be given. Make sure to only use one of the above commands for the beam energy in order to avoid ambiguities in the input file.

define impact parameter

```
imp b
imp -bmax
IMP bmin bmax
```

`b` fixed impact parameter of b fm.

`-bmax` impact parameter range from $0 \ldots b_{max}$.

`bmin` minimum impact parameter.

`bmax` maximum impact parameter.

By default, the impact parameter is weighted quadratically (`CTOption(5)` is automatically set). However, with `CTOption(5)` it is possible to change the weighting characteristic also to a linear weighting (this is in contrast to the usual experimental trigger conditions).
A minimum bias calculation (including events without interaction!) can be performed with `bmin=0`,`bmax` $> R_p + R_t$.

infinite matter (box) calculations

```
box dim edens solid para
bpt ityp iso3 npart pmax
bpe ityp iso3 npart
```

`dim` width of (cubic) box

`edens` total energy content of box in GeV

`solid` 1: reflecting walls, 0: periodic boundary conditions

`para` 0: standard, 1: use "old" periodic boundary conditions

`ityp` ID of species (see tables 2 and 3 for available itypes)

`iso3` $2 \cdot Isospin_3$ of species

`npart` number of particles for species

`pmax` maximum momentum for fermi-sphere in momentum space.

define equation of state

```
eos EoS
```

`EoS` equation of state for the calculation

Currently only CASCADE mode (`EoS=0`) or a hard Skyrme equation of state (`EoS=1`) are available. The default mode is CASCADE, the hard Skyrme equation of state is limited to incident beam-energies below 4.0 GeV/nucleon.
Important: This option also changes the initialization mode (see `CTOption(24)`).
This option has nothing to do with the equation of state during the hydrodynamic evolution in the hybrid mode.

set random number generator seed

```
rsd seed
```

`seed` (integer) seed for random number generator

On many computer systems, `UrQMD` is able to extract a random seed from the local time at intervals of one second. However, we advise to check if this is indeed the case for your system. Especially, if running many `UrQMD` jobs in parallel it is mandatory to set different individual seeds to avoid synchronisation of the runs due to the same start time.

set forced collision load update interval

```
cdt deltat
```

`deltat` time interval (in fm/c) for the update of potentials and a full particle scan for the collision arrays

In the CASCADE mode a regular full particle scan for the collision arrays is not necessary, therefore this command should be only used in calculations including potentials, infinite matter calculations or for debug purposes.

suppress output files

```
f13
f14
f15
f16
f19
f20
```

The output to the respective files is omitted if the above command is used.

set particles stable

```
stb ityp
```

`ityp` ID of particle (see tables 2 and 3 for available itypes)

Treat all particles with this ID as stable particles. At the moment the number of particles to be set stable is limited to 20 via the parameter `maxstables` in `options.f`.

set special parameter

```
ctp index value
```

`index` index of `CTParam()` array

`value` value for `CTParam(index)` (see tables 4 and 5 for available parameters)

set special option

```
cto index value
```

`index` index of `CTOption()` array

`value` value for `CTOption(index)` (see table 6 for available options)

Please note that if you are running `UrQMD`from a self-generated old inputfile via `CTOption(40)=1` make sure that all the baryons are listed first followed by all the mesons. This particle order is necessary to avoid further complications.
If `CTOption(45)=1` is used the code needs 2GB of memory because of the dimensions of the hydrodynamic grid and the corresponding array sizes.

ID	nucleon	ID	delta	ID	lambda	ID	sigma	ID	xi	ID	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 2: Baryon-ID's used in UrQMD. A particle is fully defined, when its ityp and $2 \cdot I_3$ are known. Antibaryons carry a negative sign.

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

Table 3: Meson-ID's in UrQMD, sorted with respect to spin and parity, included into the UrQMD model. Mesons with strangeness -1 carry a negative sign.

CTParam(X)	default	function
1	1.d0	scaling factor for resonance widths
2	0.52d0	minimal stringmass and el/inel. cut in `makestr`
3	2.0d0	velocity exponent for modified AQM
4	0.3d0	transverse pion mass, used in `strexct` and `make22`
5	0.0d0	probability for quark rearrangement in cluster
6	0.37d0	strangeness probability in `makestr`
7	0.d0	charm probability (not yet implemented in `UrQMD`)
8	0.093d0	probability to create a diquark
9	0.35d0	kinetic energy cut-off for last string break
10	0.25d0	min. kinetic energy for hadron (in string)
11	0.0d0	percentage of non groundstate resonances (in string)
21	0.d0	deformation parameter
25	0.9d0	probability for diquark not to break
28	1.d0	scaling factor for transverse fermi motion
29	1.d0	double strange di-quark suppression factor
30	1.5d0	radius offset for initialization
31	1.6d0	σ of Gaussian for transverse momentum transfer
32	0.d0	$\alpha - 1$ for valence quark distribution
33	2.5d0	β_v for valence quark distribution
34	0.1d0	minimal x multiplied with $E_{c.m.}$
35	3.0d0	offset for cut for the FSM
36	0.275d0	fragmentation function parameter a (nucleons)
37	0.42d0	fragmentation function parameter b (nucleons)
38	1.08d0	diquark p_t scaling factor
39	0.8d0	strange quark p_t scaling factor
40	0.5d0	$\beta_s - 1$ for valence quark distribution
41	0.d0	distance between nuclei at initialization
42	0.55d0	width of Gaussian for p_t-distribution in string-fragmentation
43	5.0d0	maximum kinetic energy in mesonic cluster
46	8.0d6	maximum number of rejections during initialization of nuclei
47	1.0d0	Field-Feynman fragmentation func. parameter a (prod. part.)
48	2.0d0	Field-Feynman fragmentation func. parameter b (prod. part.)

Table 4: Optional parameters used in `UrQMD`

CTParam(X)	default	function
49	0.5d0	additional single strange diquark suppression factor
50	1.0d0	enhancement factor for 0^{-+} mesons
51	1.0d0	enhancement factor for 1^{--} mesons
52	1.0d0	enhancement factor for 0^{++} mesons
53	1.0d0	enhancement factor for 1^{++} mesons
54	1.0d0	enhancement factor for 2^{++} mesons
55	1.0d0	enhancement factor for 1^{+-} mesons
56	1.0d0	enhancement factor for $(1^{--})^{*}$ mesons
57	1.0d0	enhancement factor for $(1^{--})^{**}$ mesons
58	1.0d0	scaling factor for DPF time-delay
59	0.4d0	scaling factor for leading hadron cross-section (PYTHIA)
60	3.0d0	resonance/string transition energy for s-channel
61	0.2d0	cell size dx of the hydro code
62	200	`ngr` is the grid size of the hydro code
63	1.0d0	minimum $t_{\rm start}$ for hydro calculation
64	5.0d0	multiplied with ϵ_0 as freeze-out criterion
65	1.0d0	factor to be multiplied with $t_{\rm start}$
66	1.d10	rapidity cut for hydrodynamic decription
67	1.d0	integer number of testparticles per real particle

Table 5: Optional parameters used in `UrQMD`

CTOption(X)	**default / options**	**description**
1	0	mass dependent resonance decay widths
	0	enabled
	1	disabled
2	0	2-particle scattering plane:
	0	stochastic selection of $\varphi(1,2)$
	1	conserve plane
3	0	detailed balance selection
	0	take finite resonance widths into account
	1	use standard detailed balance
4	0	initial configuration output to file14
	0	output according to `tim` statement
	1	**additional** output of initialization
5	0	impact parameter weighting
	0	use `bmax` as fixed impact parameter
	1	random b from `bmin` to `bmax`, bdb weighted
	2	random b from `bmin` to `bmax`, flat distribution
6	0	first collisions within target/projectile
	0	block first collisions **within** proj./target
	1	all collisions allowed
7	0	suppress elastic NN collisions
	0	elastic collisions are allowed
	1	no elastic NN collisions; $\sigma_{in} = \sigma_{tot}$
8	0	mass dependent partial decay widths
	0	enabled
	1	disabled, use fixed widths
9	0	tabulated p+p inelastic cross sections
	0	enable table-lookup
	1	disable table-lookup
10	0	Pauli-blocker
	0	enable Pauli-blocker
	1	disable Pauli-blocker

CTOption(X)	**default / options**	**description**
11	0	mass reduction (binding energy) in CASCADE mode
	0	enable mass reduction according to binding energy
	1	disable mass reduction
12	0	string production
	0	enable string production
	1	disable string production
13	0	enhanced file16 output
	0	disabled
	1	enabled
14	0	angular distribution in binary scattering
	0	enable angular distribution
	1	disable distribution ($\cos(\vartheta) = 1$ forward peak)
15	0	meson-meson and meson-baryon scattering
	0	enable MM and MB scattering
	1	disable MM and MB scattering
16	0	molecular dynamics switch
	0	enable collision term
	1	propagate with forces only (disable collision term)
17	0	collision-table update mode
	0	update only collision partners after interaction
	1	initialize complete table after every interaction
18	0	decay of unstable particles at end of event
	0	perform decay after final timestep
	1	unstable particles do not decay after final timestep
19	0	$B\bar{B}$ annihilation
	0	enabled
	1	disabled
20	0	e^+e^- annihilation instead of $B\bar{B}$ annihilation
	0	disabled (normal $\bar{B}B$ mode)
	1	enabled (e^+e^- mode)

CTOption(X)	**default / options**	**description**
21	0	string fragmentation function
	0	field-Feynman fragmentation function
	1	Lund fragmentation function
	2	QGSM fragmentation function
22	1	string mass excitation
	-1	simple $1/M$ excitation
	1	FRITIOF ansatz
	2	QGSM ansatz
23	0	Lorenz contraction of projectile and target
	0	enabled
	1	disabled
24	1	initialization mode
	0	hard sphere (used for EOS$\neq$0)
	1	Woods-Saxon (used for CASCADE mode)
	2	Fast Woods-Saxon (used for CASCADE mode)
25	0	phase space correction for resonance masses
	0	disabled
	1	enabled
27	0	reference frame for calculation
	0	N.N. (equal speed) frame
	1	target (lab) frame
	2	projectile frame
29	2	p_t for last two particles in string (`clustr`)
	0	isotropic
	1	isotropic, baryon goes into forward hemisphere
	2	baryon goes into forward hemisphere, $p_t = 0$
30	1	frozen Fermi approximation in CASCADE mode
	0	disabled
	1	enabled
32	0	distribute resonance masses according to mass-dep. Breit-Wigner
	0	enabled
	1	disabled

CTOption(X)	default / options	description
33	0	use table-lookup for calculation of $\langle p_{CMS} \rangle$ in `pmean`
	0	enabled
	1	disabled
34	1	resonance life-times
	0	$\tau = 1/\Gamma(M)$
	1	$\tau = 1/\Gamma_{pole}$
	2	DPF formalism
35	1	generate high-precision tables (file `tables.dat`)
	0	disabled
	1	enabled
36	0	correct normalization for mass-dependent Breit-Wigner distributions
	0	enabled
	1	disabled
37	0	heavy quark clusters
	0	disabled
	1	enabled
38	0	scale p-pbar to b-bbar with equal p_lab instead of equal $\sqrt{s}$
	0	disabled
	1	enabled
39	0	compute collision densities via call to Pauli-blocker
	0	enabled
	1	disabled
40	0	use old file14 as initial state for calculation
	0	disabled
	1	enabled
41	0	extended file14 output (needed for `cto(40)`)
	0	disabled
	1	enabled
	2	different counting rules for `origin`
42	0	color fluctuations in high energy hadron-hadron collisions
	0	disabled
	1	enabled

CTOption(X)	**default / options**	**description**
44	1	Pythia call for hard scatterings
	0	disabled
	1	enabled
45	0	Hydro mode
	0	disabled
	1	enabled
46	0	Density calculation switch
	0	ρ_B
	1	$\rho_{q+\bar{q}}$
47	2	EoS for hydro evolution
	2	hadron gas (HG)
	3	bag model (BM)
	5	chiral+hadron gas (CH)
48	0	number of timesteps for hydro propagation
	0	usual run until freeze-out
	N	N timesteps
49	0	Spectator switch
	0	spectators are propagated in `UrQMD`
	1	spectators are propagated on the hydro grid
50	0	(Additional) f14/f19-output directly after the hydro to transport transition
	0	disabled
	1	enabled
52	0	Freeze-out switch for hydro mode
	0	Gradual transition scenario (GF)
	1	Isochronuous transition scenario
53	0	Improved momentum generation in Cooper-Frye
	0	enabled
	1	disabled

Table 6: available options in `UrQMD`

Output files

The UrQMD program has several different output files. The *standard output files* (file13 and file14) contain all particles of a given event at a certain time-step. The *collision history file* (file15) contains information on all collisions/decays of a given event. The *decay file* (file16) contains information on all particle decays as well as information on all stable particles after the final timestep. The *OSC files* (file19, file20) generates output compliant with the Open Standards And Codes (OSCAR) format. Consecutive timesteps (only file13 and file14) and events are added sequentially to the files.
Each event consists of a header and a body. The standard header is identical for file13, file14 and file16 and file15 use an abbreviated header and the format of file19 and file20 is fixed by the OSCAR requirements.

Standard output-files: file13/file14

The standard output files contain the phase-space of the event at a given timestep (e.g. final output after last timestep). file13 contains the same information as file14, but additionally lists the freeze-out coordinates in configuration- and momentum-space for all particles.
Figure 3 shows a standard header as used in file13, file14 and file16. Only one header per event is written to file. Consecutive time-steps of the same event are added body to body without additional headers between them. In the case of running UrQMD in box mode, file14 contains additional header-lines reporting the box-related parameters.
The general format of the standard-fileheader can be found in table 7, its contents is self-explanatory – please consult figure 3.
The format of the box-header can be found in table 8, please consult also figure 4.
All lines of the box-header are guaranteed to start with box. The first line contains the word boxmode, followed by length of the box, total energy and the parameters solid and para (see table 1) and the number of specified particle species npart. The second line contains no physical information. The rest of the box-header contains npart lines, each of them stating ityp, isospin, number and maximal momentum of the particles as specified in the bpt and bpe input file directives.
The body of the *standard output files* contains in its first line the number of particles N_{part} to follow (there are as many lines to follow as there are particles) and the time (in fm/c) of the output (two unformatted integers). The next line contains counters for the number of collisions, decays and produced resonances per event (format(8i8)), it is described in table 9.
The subsequent N_{part} lines then contain the information on the individual particles. The exact format of the particle vector depends on the chosen output file and the selected options. Table 10 lists the different possibilities. Figure 5 shows the beginning of a sample event body for file14.
The standard output files should suffice for most types of analysis. They provide the event information at a given timestep (mostly the final timestep). The contents of the particle vectors is described in table 11. All reference frame dependent values are given in the computational frame, which has been fixed by CTOption(27).

line#	format
1	`format(a20,3i7,a15,i2)`
2	`format(a13,a13,i4,i4,a12,a13,i4,i4,a1)`
3	`format(a36,3f11.7)`
4	`format(a36,3f6.2,a31,1f9.2)`
5	`format(a20,i3,a15,e11.4,a15,e11.4,a15,e11.4)`
6	`format(a7,i9,a13,i12,a9,a20,i7,a20,f11.3)`
7	`format(a2,15(i3,a2))`
8	`format(a2,15(i3,a2))`
9	`format(a2,15(i3,a2))`
10	`format(a2,12(e11.4,a2))`
11	`format(a2,12(e11.4,a2))`
12	`format(a2,12(e11.4,a2))`
13	`format(a2,12(e11.4,a2))`
14	`format(a171)`

Table 7: format for the standard event header.

line#	format
1	`format(a20,e14.6,a20,e14.6,a3,i1,a3,i1,a3,i3)`
2	`format(a35)`
3ff	`format(a5,2i4,i8,e14.6)`

Table 8: format for the box-header extension.

column#	contents
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

Table 9: description of the collision/decay counters in the standard output file

format	
`format(9e16.8,i11,2i3,i9,i5,i4)`	standard `file14` and `file16`
`format(9e16.8,i11,2i3,i9,i5,i10,3e16.8,i8)`	`file14` with `CTOption(41)=1`
`format(9e16.8,i11,2i3,i9,i5,i4,8e16.8)`	standard `file13`
`format(9e15.7,i11,2i3,i9,i5,i4,2i4)`	`file16` with `CTOption(13)`

Table 10: particle-vector format for different output-options in the standard output files.

Collision history file: file15

The collision file `file15` contains each binary interaction, resonance decay and string-excitation which occurred in the course of the heavy-ion reaction. It can be used to reconstruct the entire space-time evolution of the event. Each entry (collision, decay or annihilation) consists of a header line followed by 3 to N lines (three lines for annihilations/decays, four lines for scattering, possibly more lines for string-decays) with the individual particle information.
The event-header consists of a single line of the format:

`format(i8,i8,i4,i7,f8.3,4e12.4)`

The format is identical to the header line for the respective binary interactions and decays which follow in the file. In order to distinguish the beginning of an event from the beginning of a collision/decay entry, the first integer in the event header is a `-1`. It is then followed by the number of the event, the mass of projectile and target, the impact parameter, the two-particle c.m. energy of the heavy-ion reaction (i.e. $\sqrt{s}$ for proton-proton reactions), the total cross section of the heavy-ion reaction and the beam energy and momentum (per particle) in the laboratory frame.
Figure 6 shows a sample collision entry. The header line contains first the number of in- and outgoing particles (scattering: 2 2; decay: 1 2; annihilation: 2 1; Pauli-blocked collision: 2 0; Pauli-blocked decay: 1 0 and string decay with 5 outgoing particles: 2 5), the ID of the respective process (e.g. elastic scattering, decay, string excitation ...) then the number of the collision (in the respective event), the collision time in fm/c, the total CM energy ($\sqrt{s}$ in GeV), the total cross section (σ_{tot} in mbarn), the partial cross section (σ_i in mbarn) for the respective exit channel and finally the baryon density at the collision point. Note that the cross sections are ill-defined in the case of a decay.
One of the purposes of the collision file is to have the possibility to track the trajectory of a single particle in the course of the reaction or the time evolution of the available CM-energy per binary collision. The contents of the particle vectors has the format:

`format(i5,9e16.8,i11,2i3,i9,i5,i3,i15)`

and is described in table 12.

Decay output: file16

The header of the decay output file is identical to that of the *standard output files* (see figure 3 and table 7). The body of the *decay file* contains entries for each particle which has decayed during the event as well as a list of all stable particles after the final timestep of the event. Since the number of

column#	contents
1	t : eigentime of particle in fm/c
2	r_x : x coordinate in fm
3	r_y : y coordinate in fm
4	r_z : z coordinate in fm
5	E : energy of particle in GeV
6	p_x : x momentum component in GeV
7	p_y : y momentum component in GeV
8	p_z : z momentum component in GeV
9	m : mass of particle in GeV
10	$ityp$: particle-ID
11	$2 \cdot I_3$: isospin z-projection (doubled)
12	ch : charge of particle
13	index of last collision partner
14	N_{coll} number of collisions
15	history information (parent process)
16	t_{fr} : freeze-out time of particle in fm/c (`file13` only)
17	$r_{x,fr}$: freeze-out x coordinate in fm (`file13` only)
18	$r_{y,fr}$: freeze-out y coordinate in fm (`file13` only)
19	$r_{z,fr}$: freeze-out z coordinate in fm (`file13` only)
20	E_{fr} : freeze-out energy of particle in GeV (`file13` only)
21	$p_{x,fr}$: freeze-out x momentum component in GeV (`file13` only)
22	$p_{y,fr}$: freeze-out y momentum component in GeV (`file13` only)
23	$p_{z,fr}$: freeze-out z momentum component in GeV (`file13` only)
16	τ_{dec} decay time of particle (`file14` with `CTOption(41)=1`)
17	τ_{form} formation time of particle (`file14` with `CTOption(41)=1`)
18	R_σ cross section reduction factor (`file14` with `CTOption(41)=1`)
19	unique particle number (not ID!) (`file14` with `CTOption(41) =1`)
16	$ityp_{old,1}$: particle-ID of parent particle # 1 (`file16` with `CTOption(13)=1`)
17	$ityp_{old,2}$: particle-ID of parent particle # 2 (`file16` with `CTOption(13)=1`)

Table 11: contents of the particle vector in the standard output files

column#	contents
1	ind : index of particle
2	t : computational frame time of particle in fm/c
3	r_x : x coordinate in fm
4	r_y : y coordinate in fm
5	r_z : z coordinate in fm
6	E : energy of particle in GeV
7	p_x : x momentum component in GeV
8	p_y : y momentum component in GeV
9	p_z : z momentum component in GeV
10	m : mass of particle in GeV
11	$ityp$: particle-ID
12	$2 \cdot I_3$: isospin z-projection (doubled)
13	ch : charge of particle
14	index of last collision partner
15	N_{coll} number of collisions
16	S : strangeness
17	history information (parent process)

Table 12: contents of the particle vector in the collision file

decays and particles listed in the event-body is not determined at the point where the first output is written to file, the event body directly starts with particle-vector entries (see tables 10 and 11). The end of the event is marked with a line of the format `format(a1,8i8)`, which contains an `E` as a marker in the first column, followed by the collision counters listed in table 9.
For `CTOption(13)=1` all outgoing particles of all collisions and decays are listed instead of the decaying particles alone. The *decay output file* provides additional information about the produced baryon- and meson-resonances when compared to the *standard output file*. Since it contains particle output at different times during the event, however, one has to be very careful when extracting cross sections. All reference frame dependent values are given in the computational frame, which has been set by `CTOption(27)`.

The process identifier/parent process

The standard output file and the collision file contain information on the current/previous sub-process. In the collision file the information on the current process type is stored in the header of each individual reaction (position 3). In the standard output file entry 15 provides the ID of the parent process leading to the production of this particle. A list of process IDs is given in table 13.

```
UQMD   version:       10000   1000  10001  output_file  14
projectile:  (mass, char)   32  16   target:  (mass, char)   32  16
transformation betas (NN,lab,pro)       .0000000   .7183285  -.7183285
impact_parameter_real/min/max(fm):     .00   .00   .00  total_cross_section(mbarn):        .00
equation_of_state:    0  E_lab(GeV/u):  .2000E+01  sqrt(s)(GeV):  .2697E+01  p_lab(GeV/u):  .2784E+01
event#         1 random seed:  1944955121 (fixed)  total_time(fm/c):    60 Delta(t)_O(fm/c):   60.0
op  0    0    0    0    0    0    0    0    0    0    0    0    0    0
op  0    0    0    0    0    0    1    0    1    0    0    0    0    2    1
op  0    0    0    1    1    0    0    0    0    0    0    0    0    0    0
pa  .1000E+01    .5200E+00    .5000E+00    .3000E+00    .0000E+00    .3700E+00    .0000E+00    .9300E-01    .3500E+00    .2500E+00    .0000E+00    .5000E+00
pa  .2700E+00    .4900E+00    .2700E+00    .1000E+01    .1600E+01    .8500E+00    .1550E+01    .0000E+00    .0000E+00    .0000E+00    .0000E+00    .0000E+00
pa  .9000E+00    .5000E+02    .1000E+01    .1000E+01    .4000E+00    .1500E+01    .1600E+01    .0000E+00    .2500E+01    .1000E+00    .3000E+01    .2750E+00
pa  .4200E+00    .1080E+01    .8000E+00    .5000E+00    .0000E+00    .5500E+00    .5000E+01    .8000E+00    .5000E+00    .8000E+06    .1000E+01    .2000E+01
pvec: r0             rx             ry             rz             p0             px             py             pz             m          ityp 2i3 chg lcl#  ncl or
... start of event body ...
```

Figure 3: sample header of an UrQMD output file.

```
boxmode length(fm):   0.200000E+02 tot. energy (GeV):   0.228900E+04 s:1 p:0 #:  1
boxh ityp 2i3       N     pmax(GeV)
box:  101   1    2289  0.763000E+03
```

Figure 4: sample box-header in UrQMD output file.

```
 83 60
     248     105     141       2      78     165       0       0
   .60000000E+02  -.35597252E+02  -.76801184E+01   .30121505E+01   .11839331E+01  -.70109432E+00  -.16633296E+00   .51512840E-01   .93800002E+00    1  1  1     1    9  20
   .60000000E+02   .81996126E+01   .19904670E+02  -.24104662E+02   .11707300E+01   .14606801E+00   .47854791E+00  -.49032721E+00   .93800002E+00    1 -1  0     2   17  20
   .60000000E+02   .54067910E+01   .30092770E+02   .35340946E+02   .15291829E+01   .14462310E+00   .74024320E+00   .94322867E+00   .93800002E+00    1  1  1    37    3  30
   .60000000E+02   .72953637E+00  -.35051235E+01  -.14941324E+02   .96631053E+00   .47106723E-02  -.60615381E-01  -.22408834E+00   .93800002E+00    1  1  1    13    2  19
...
```

Figure 5: beginning of a sample body of an UrQMDstandard output file.

```
...
2       2   1      1    .398   .2650E+01   .4565E+02   .1834E+02   .1742E+01
   31   .39812730E+00   .10071119E+01   .11882873E+01   .10408219E+00   .13016664E+01  -.64792705E-01  -.13898806E-01   .91198600E+00   .92640464E+00    1 -1  0     0    0  0              0
   64   .39812730E+00   .28051872E+00   .15413296E+01   .10408219E+00   .13490995E+01   .11074780E+00   .15481079E-01  -.97761791E+00   .92294520E+00    1 -1  0     0    0  0              0
   31   .39812730E+00   .10071119E+01   .11882873E+01   .10408219E+00   .15827584E+01  -.15319538E+00   .62127758E-01  -.53196784E+00   .14814876E+01   17 -3 -1    64    1  0              1
   64   .39812730E+00   .28051872E+00   .15413296E+01   .10408219E+00   .10680074E+01   .19915048E+00  -.60545484E-01   .46633592E+00   .93800002E+00    1  1  1    31    1  0              1
2       2   5      2    .743   .2791E+01   .4448E+02   .8529E+00   .1857E+01
   25   .74262809E+00  -.11192341E+01  -.13384231E+01   .25707630E-01   .14261601E+01   .32413590E-01  -.82195855E-01   .10840402E+01   .92248724E+00    1 -1  0     0    0  0              0
   50   .74262809E+00  -.97716182E+00  -.18394695E+01   .25707630E-01   .13692160E+01  -.13287008E-01  -.56741371E-01  -.10054827E+01   .92755640E+00    1 -1  0     0    0  0              0
   25   .74262809E+00  -.11192341E+01  -.13384231E+01   .25707630E-01   .14831149E+01   .14886580E+00  -.27346802E+00  -.36638117E+00   .14030142E+01    2 -1  0    50    1  0              5
   50   .74262809E+00  -.97716182E+00  -.18394695E+01   .25707630E-01   .13122612E+01  -.12973922E+00   .13453079E+00   .44493867E+00   .12202985E+01   17 -1  0    25    1  0              5
...
1       2  20      4    .858   .1481E+01   .0000E+00   .1718E+02   .1928E+01
   31   .85784130E+00   .96261609E+00   .12063324E+01  -.50428488E-01   .15827584E+01  -.15319538E+00   .62127758E-01  -.53196784E+00   .14814876E+01   17 -3 -1    64    1  0              1
   31   .85784130E+00   .96261609E+00   .12063324E+01  -.50428488E-01   .11582730E+01  -.45590325E+00  -.19299762E+00  -.46546376E+00   .93800002E+00    1 -1  0    31    2  0             20
   65   .85784130E+00   .96261609E+00   .12063324E+01  -.50428488E-01   .42448534E+00   .30270787E+00   .25512538E+00  -.66504080E-01   .13800000E+00  101 -2 -1    31    1  0             20
...
```

Figure 6: excerpts of a sample body of an UrQMDcollision history file.

Process ID#	Description
1	NN→ND
2	NN→NN*
3	NN→ND*
4	NN→DD
5	NN→DN*
6	NN→DD*
7	NN → N*N*,N*D*,D*D*
8	ND→DD
10	MB→B'
11	MM→M'
13	BB (but not pp,pn) elastic scattering
14	inelastic scattering (no string excitation)
15	BB → 2 strings
17	pn-elastic
19	pp-elastic
20	decay
22	BBar elastic
23	BBar annihilation → 1 string
24	BBar diffractive → 2 strings
26	MB elastic scattering
27	MB,MM → 1 string
28	MB,MM → 2 strings
30	ND→NN
31	DD→DN
32	DD→NN
35	ND inelastic
36	Danielewicz forward delay (MB→B')
37	Danielewicz forward delay (MM→M')
38	MM elastic scattering
39	BBar inelastic scattering (no annihilation)

Table 13: list of process identifiers

OSC1997A output (OSCAR 1997A format): file19

The OSC output format has been defined by the OSCAR group in order to create a well defined easily accessible output-format which is supported by all OSCAR compliant transport models, event genera-

tors and other heavy-ion related models. For a full overview of the goals of the OSCAR collaboration, please consult the web-site

http://karman.physics.purdue.edu/OSCAR/

UrQMD supports the OSC1997A output format. The file-header consists of three lines: The first two lines have the format format(a12) and specify the format (e.g. OSC1997A) and the file contents. Currently, this is final_id_p_x, i.e. the final event output including particle ID, the momentum space information and the freeze-out coordinates in configuration space. The third line has the format

```
format (2(a8,2x),'(',i3,',',i6,')+(',i3,',',i6,')',2x,a4,2x,e10.4,2x,i8)
```

and contains first the model-name and version, followed by mass and charge of projectile and target, the reference frame of the calculation, the incident beam energy and the number of test-particles used per nucleon.

The event header consists of one line with the format:

format(i10,2x,i10,2x,f8.3,2x,f8.3)

This line lists the number of the event, the number of particles in the event, the impact parameter and an azimuthal angle ϕ with which the event-plane might be rotated with respect to the xz-plane. The subsequent particle entries of the event-body have the format:

format(i10,2x,i10,2x,9(e12.6,2x))

and contain first the particle number, its ID, then the four-momentum vector of the particle (p_x, p_y, p_z, E), followed by the mass of the particle and finally its freeze-out location (x_f, y_f, z_f, τ_f). The particle ID is given according to the definitions of the *Review of Particle Properties* Monte-Carlo naming scheme. Composite clusters (nuclei) are marked with 7AAAZZZ (AAA: mass, ZZZ: charge of cluster). If other objects than nuclei are used as projectile or target, then a -1 is listed in the mass-slot followed by the PDG-ID in the charge slot. Figure 7 shows a sample output in the OSC format.

OSC1999A output (OSCAR 1999A format): file20

The OSC1999A is an improvement to the OSC97A output and allows to write out the complete event history – starting with the initial state, including all binary collisions, string-fragmentations and hadronic decays. In it's scope it is comparable to the UrQMD file15 collision output file, but includes also the full initial configuration and final state information. An abbreviated OSC199A output format without the intermediate collision history can be used as replacement for the OSC97A output format. The first three lines of the header are almost identical to the OSC97A format, but are preceded by a comment marker # and a blank space in each line.

The first two lines thus have the format format(a20) and specify the format (e.g. # OSC1999A) and the file contents. Currently, this can be # full_event_history, the final output tag final_id_p_x referring to the OSC1997A format. The third line has the format

```
format ('# (',i3,',',i6,')+(',i3,',',i6,')',2x,a4,2x,e10.4,2x,i8)
```

and contains first the model-name and version, followed by mass and charge of projectile and target, the reference frame of the calculation, the incident beam energy and the number of test-particles used per nucleon. OSCAR further recommends that additional information be provided in subsequent comment lines, e.g.

```
OSC1997A
final_id_p_x
   UrQMD       1.0  ( 32,    16)+( 32,    16)  eqsp   .2000E+01         1
         1          83      .000      .000
         1        2212  -.701094E+00  -.166333E+00   .515128E-01   .118393E+01   .938000E+00  -.355973E+02  -.768012E+01   .301215E+01   .600000E+02
         2        2112   .146068E+00   .478548E+00  -.490327E+00   .117073E+01   .938000E+00   .819961E+01   .199047E+02  -.241047E+02   .600000E+02
         3        2212   .144623E+00   .740243E+00   .943229E+00   .152918E+01   .938000E+00   .540679E+01   .300928E+02   .353409E+02   .600000E+02
         4        2212   .471067E-02  -.606154E-01  -.224088E+00   .966311E+00   .938000E+00   .729536E+00  -.350512E+01  -.149413E+02   .600000E+02
         5        2212  -.225070E-01  -.148103E-01   .231440E+00   .966506E+00   .938000E+00  -.866336E+00  -.961651E+00   .125389E+02   .600000E+02
         6        2212  -.241272E+00   .159999E+00  -.574162E+00   .113724E+01   .938000E+00  -.126123E+02   .657565E+01  -.298675E+02   .600000E+02
...
```

Figure 7: Sample output in the OSC1997A format.

```
# Initial Condition:  Au + Au @ 200
GeV/c
# Cascade Time Ordering:  Center of
Mass
# [additional parameters supplied for
the run]
```

The remaining file after the comments contains the full history of each event in blocks of data. Each block describes one interaction and has the following format:

```
block header  (one line)
particle list (one or more lines)
```

The block header contains:

```
nin nout [optional information]
```

with `nin` and

tt nout being integers denoting the number of ingoing and outgoing particles of that particular reaction – e.g. `2 2` for two particles scattering into two particles or `1 2` for a resonance decaying into two particles. Optional information can be anything that fits on one line. For example one could put `g q` $\rightarrow$ `g q` to characterize the block as describing elastic quark-gluon scattering. Thus the minimum format for that line is `format (2(i7,2x))`. In `UrQMD` additional information is supplied analogously to the file15 output: after `nin` and `nout` the ID of the respective process (e.g. elastic scattering, decay, string excitation ...) is listed, followed by the number of the collision (in the respective event), the collision time in fm/c, the the total CM energy ($\sqrt{s}$ in GeV), the total cross section (σ_{tot} in mbarn), the partial cross section (σ_i in mbarn) for the respective exit channel and finally the baryon density at the collision point (in units of nuclear ground state density). Note that the cross sections are ill-defined in the case of a decay.

The next `nin` lines are incoming particles followed by `nout` lines of outgoing particles, the format of these lines is

```
ipart id ist px py pz p0 mass x y z t [optional information]
```

This format is identical to the particle entries of the OSC1997A format with two additions: `ist` is an integer containing additional information related to the particle ID – this is needed in some event generators indicating the status of the particular entry. The optional information can be anything useful or relevant to the particular model, but has to fit into the same output line. Thus, each particle entry line contains first the particle number, its ID, the 2nd ID-tag, then the four-momentum vector of the particle (p_x, p_y, p_z, E), followed by the mass of the particle (all in GeV) and finally its production vertex [(x, y, z, τ); in fm and fm/c]. The minimal format for this line is `format (3(i10,2x),9(e12.6,2x))`.

It should be noted that the particle number `ipart` is a unique particle identifier (not equivalent to the memory slot information used in OSC1997A) which is created for a particle at its production point and is retired for the duration of the event at the destruction/scattering vertex of the respective particle. It thus can be used to track trajectories of particles in the course of the reaction.

The particle ID is given according to the definitions of the *Review of Particle Properties* Monte-Carlo naming scheme. Composite clusters (nuclei) are marked with `7AAAZZZ` (`AAA`: mass, `ZZZ`: charge of cluster). If other objects than nuclei are used as projectile or target, then a `-1` is listed in the mass-slot followed by the PDG-ID in the charge slot.

The very first block of each event describes the initial distribution of the nucleons (partons, or other species). In this case the header contains `nin=0` and `nout` is the number of initial particles, followed by the respective particle vectors in the body of the block. OSCAR recommends optional information for the event header to include the event number, impact parameter and azimuthal angle/orientation of that event. Thus the header of the first event block is very similar to the OSC1997A format:

```
format (3(i7,2x),2(f8.3,2x))
```

listing a zero, the number of initial particles, the number of the event, the impact parameter and an azimuthal angle ϕ with which the event-plane might be rotated with respect to the xz-plane.
The very last block of each event describes the final (freezeout) configuration of all particles. Here the header comes with `nin` as the number of final particles and `nout = 0 .`
Figure 8 shows a sample output in the OSC1999A format.

Appendix A: changes from version 1.0/1.1 to 1.2

New OSCAR output

The OSC1999A OSCAR intermediate file output format has been added. Caution: to remain consistent with our input-file convention, you have to insert `f20` into your urqmd-inputfile in order **not** to get any file20 output. If you forget this, you will get a `fort.20` which will surely explode your quota, since the output on this file is **huge**.
This format is similar to our collision-file output format, but adheres to the new OSCAR OSC1999A output convention. Output is written to Fortran-unit 20.
New features include particle ID according to the PDG Monte-Carlo ID scheme and a new global quantum number, uid (stands for Unique Particle ID), i.e. a serial number the particle gets at creation and which is retired from the event after the particle undergoes an interaction. This ID is supposed to make the tracing of a particle through the collision file easier, since it does not change dynamically as our particle-slots/numbers do, due to the internal `UrQMD` memory management. Most file modifications are due to the introduction of this new uid-array. As usual all output statements are found in `output.f`

Bug-fixes and improvements

- In `GNUmakefile` there is a small modification for the code to run on Alpha-machines.
- Complete rewrite of `gnuranf`. `gnuranf` is now the default for Linux.
- A new angular distribution in the meson-baryon channel is used. I.e. isotropic resonance decays below an inv. mass of 6 GeV and a forward-backward behavior above. `angdis.f`: for collisions with sroot larger than 6 GeV zero degree scattering is enforced (only deflection from string decay).
- A sign-error in `angdis.f` has been corrected. This error led to a wrong symmetry in meson-baryon collisions (only visible when running `UrQMD` for elementary hadron-hadron collisions).

```
# OSC1999A
# full_event_history
# UrQMD 1.2
# (  4,     2)+(  4,     2)  nncm  0.5000E+02         1
      0        8        1     0.200     0.000
         1        2212           0  -.521296E-01  0.525298E-01  0.405082E+01  0.415387E+01  0.916548E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
         2        2212           0  -.723310E-01  0.239309E-01  0.517619E+01  0.525948E+01  0.929208E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
         3        2112           0  0.596589E-01  -.202471E-02  0.544394E+01  0.552250E+01  0.926260E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
         4        2112           0  0.648017E-01  -.744360E-01  0.487618E+01  0.496485E+01  0.928902E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
         5        2212           0  -.442186E-01  0.180513E-01  -.437103E+01  0.446873E+01  0.928092E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
         6        2212           0  -.385991E-01  0.383077E-01  -.431142E+01  0.441015E+01  0.926336E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
         7        2112           0  -.489669E-01  0.150743E-01  -.557163E+01  0.564798E+01  0.924068E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
         8        2112           0  0.131784E+00  -.714333E-01  -.532665E+01  0.540741E+01  0.918942E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
      2        4       15        1   0.118  0.9420E+01  0.3847E+02  0.2506E+02  0.1508E+01
         4        2112           0  0.648017E-01  -.744360E-01  0.487618E+01  0.496485E+01  0.928902E+00  0.669238E+00  -.290863E+00  -.383425E-01  0.117799E+00
         5        2212           0  -.442186E-01  0.180513E-01  -.437103E+01  0.446873E+01  0.928092E+00  -.205637E+00  0.215361E+00  -.383425E-01  0.117799E+00
         9        2114           0  0.376577E+00  -.473402E+00  0.223060E+01  0.261298E+01  0.121909E+01  0.669238E+00  -.290863E+00  -.383425E-01  0.117799E+00
        10        2212           0  -.311133E+00  -.221868E+00  -.318752E+01  0.334457E+01  0.938000E+00  -.205637E+00  0.215361E+00  -.383425E-01  0.117799E+00
        11       -3112           0  0.235318E+00  -.485507E+00  0.890289E+00  0.158258E+01  0.119200E+01  0.669238E+00  -.290863E+00  -.383425E-01  0.117799E+00
        12        3114           0  -.280179E+00  0.112439E+01  0.571773E+00  0.189345E+01  0.138400E+01  0.669238E+00  -.290863E+00  -.383425E-01  0.117799E+00
...
      2        2       19        3   0.290  0.9240E+01  0.3840E+02  0.7584E+01  0.1680E+01
        14        2112           0  -.425527E-01  0.836121E-01  0.366816E+01  0.378736E+01  0.938000E+00  -.101003E+01  -.746785E-01  0.898854E-01  0.290019E+00
         7        2112           0  -.489669E-01  0.150743E-01  -.557163E+01  0.564798E+01  0.924068E+00  -.888384E+00  0.244736E+00  0.869713E-01  0.290019E+00
        18        2112           0  0.845839E-02  -.167264E+00  0.366215E+01  0.378408E+01  0.938000E+00  -.101003E+01  -.746785E-01  0.898854E-01  0.290019E+00
        19        2112           0  -.999780E-01  0.265951E+00  -.556562E+01  0.565126E+01  0.938000E+00  -.888384E+00  0.244736E+00  0.869713E-01  0.290019E+00
...
      1        2       20        7   1.257  0.1687E+01  0.0000E+00  0.5665-265  0.7609E+00
        26        1212           0  -.163056E+00  -.115806E+00  -.522476E+01  0.549408E+01  0.168722E+01  0.516529E+00  -.837827E+00  -.977526E+00  0.125673E+01
        32        2214           0  -.352471E+00  -.203853E-01  -.320625E+01  0.345977E+01  0.125111E+01  0.516529E+00  -.837827E+00  -.977526E+00  0.125673E+01
        33        -211           0  0.189415E+00  -.954203E-01  -.201851E+01  0.203431E+01  0.138000E+00  0.516529E+00  -.837827E+00  -.977526E+00  0.125673E+01
...
     26        0
        39        2112           0  -.207591E+00  -.539088E-01  -.306519E+01  0.321267E+01  0.938000E+00  0.335363E+00  -.848305E+00  -.262550E+01  0.303501E+01
        28        2212           0  -.362984E+00  0.210145E+00  0.446340E+01  0.458015E+01  0.938000E+00  0.195594E+00  0.118682E+01  0.540054E-01  0.372621E+00
        35        2112           0  0.130921E+00  0.811300E-01  0.324282E+01  0.337927E+01  0.938000E+00  -.100510E+01  -.172068E+00  0.222217E+01  0.249329E+01
        46        2112           0  0.321851E+00  -.147599E+00  0.166698E+01  0.194526E+01  0.938000E+00  0.275070E+01  -.290751E+01  0.122909E+02  0.145606E+02
        21        2212           0  0.234665E+00  0.398535E+00  -.210678E+00  0.106683E+01  0.938000E+00  -.182715E+00  0.260743E+00  -.716466E-01  0.302219E+00
        29        2212           0  0.746550E+00  -.257338E+00  -.701843E+00  0.141279E+01  0.938000E+00  0.621357E+00  0.771038E+00  0.540054E-01  0.372621E+00
...
      0        0
```

Figure 8: Sample output in the OSC1999A format.

- New environment variable `URQMD_TAB` to find tables.dat (think of export `URQMD_TAB=tables.'uname'`)
- Higher meltpoint for resonant meson absorption on baryons (only eta, rho, omega and all hyperon channels).
- Bugfix in meson-meson annihilation cross section.
- `string.f`: subroutine ityp2id case of u-quark anti-u quark: quark ids corrected from 2 and -2 to 1 and -1.
- `string.f`: subroutine `gausspt` comment with respect to the calculated distribution is corrected
- `make22.f`: sighera warning only active if new logical variable - `warn` - is true. analogously defined as variables - `check` - and - `info` - in tt coms.f. As default the variable `warn` is set as false to avoid countless warnings.
- `coms.f`: new logical variable - `warn` - `nmax` = maximum number of particles increased from 5000 to 40000.
- `input.f` Due to the enhancement of nmax in coms.f a warning is added if calculations are performed for energies smaller than 200 A GeV = E_{lab} or p_{lab} or sroot smaller 20 A GeV: parameter `nmax` in `coms.f` may be decreased!
- `colltab.f`: `ncollmax`: maximum number of entries in collision table is increased from 5000 to 10000.

Appendix B: changes from version 1.2 to 1.3

Initialization

In UrQMD version 1.3 a new initialization method for cascade mode has been implemented. The initialization method used in UrQMD 1.0-1.2 led to an increased nucleon density on the surface of the nucleus and a too small total collision crossection. For calculations with Skryme equation of state (eos<>1) the old initialization method is still the recommended one.

```
Caution!  Due to the new initialization
method calculations made with UrQMD
version 1.3 may give results deviating
from the results published with earlier
versions of UrQMD!
```

To reproduce old results you can set `cto 24 0` to get the old initialization.

New Options

- New parameter `ctp 21` allows the initialization of deformed nuclei. The parameter 25 gives the deformation parameter (default is 0.0).
- New option `stb [ityp]` prevents the decay of a particle type [ityp]. Multiple definitions of `stb` in the input file are allowed.
- Phasespace correction for upper mass limit of resonances. This feature is important for correct dilepton spectra. To enable this feature set `cto 25` to 1. Default is 0.
- Alternate parametrisation of the p-pbar annihilation crosssection. You can switch with `cto 38`. For a detailed description of the two parametrizations see Phys.Rev.C66:054903,2002.
- New option `cto 41 2`:

  `origin(i)` modified: elastic collisions no longer overwrite the production process. Instead elastic collisions increment the 3rd digit of origin (origin+=100). A process with iline=27 and no color exchange is treated as an elastic collision.

New Channels

To improve the description of Kaon production at low energies the following channels have been implemented:

$$p + p \Longrightarrow p + \Sigma^+ + K^0 \quad (1)$$

$$p + p \Longrightarrow p + \Sigma^0 + K^+ \quad (2)$$

$$p + p \Longrightarrow p + p + f_0 \Longrightarrow p + p + K^+ + K^- \quad (3)$$
$$p + p \Longrightarrow p + p + a_0 \Longrightarrow p + p + \bar{K}^0 + \bar{K}^- \quad (4)$$
$$\pi^- + p \Longrightarrow N^* \Longrightarrow n + f_0 \Longrightarrow n + K^+ + K^- \quad (5)$$

Appendix C: changes from version 1.3 to 2.3

UrQMD version 2.0, 2.1 and 2.2 are considered as unstable development versions.

Inclusion of Pythia

PYTHIA 6.409 is included for hard scatterings from $\sqrt{s}_{\min} = 10$ GeV on. Hard collisions are presently defined as collisions with momentum transfer $Q > 1.5$ GeV. The transition between the low energy string routine and Pythia is smooth and given by the probability distribution for hard scatterings.

- `pythia6409.f`: This new file contains the Pythia code in version 6.409.
- `hepnam.f`,`hepchg.f`,`hepcmp.f`: New files which convert the PDG standard Id's into useful information, such as particle names, charges and other characteristics.
- `make22.f`: Minimal possible center of mass energy in the individual two particle reactions for a Pythia call is `minsrt` = 10 (GeV).
- `upmerge.f`: New file with Subroutine `upyth` which merges `UrQMD` and Pythia, e.g. converts particle arrays back and forth and finds the leading hadrons.
- `upmerge.f`: VINT(51)=Q the momentum transfer from Pythia.

  Pythia switch `CTOption(44)` is implemented.
- Leading particle cross sections for particle from PYTHIA are implemented. `leadfac` = 0.4*SUPPFAC in `upmerge.f` to reduce cross sections for leading hadrons out of Pythia as a simple way to account for coherence effects.

Adjustments of the interface PYTHIA/`UrQMD`

- Particles unknown to `UrQMD` obtain a shift in `ityp` by ± 1000 (sign depends on the sign of the `ityp`). Note that exotic PDG particle codes can now be encountered in the `UrQMD` and OSCAR output!
- Calculate the correct charge, but for simplicity we give strangeness zero to particles produced via Pythia and not known to `UrQMD`. For this purpose the functions `fchg(i3,i)` and `strit(i)` in `blockres.f` are modified.
- `dectim.f`: Unknown particles from Pythia are set stable (`dectim(i)`=10^{34}).

- `getspin.f`: Unknown particles from Pythia get spin zero.
- `ityp2pdg.f`: Transforms `UrQMD` ityps and shifted PDG-IDs to the correct PDG-Id's.
- `scatter.f`: In subroutine `collclass`: Unknown particles from Pythia do not interact collclass=0.

Inclusion of high mass resonances

High mass resonances are included in the energy regime between $\sqrt{s}_{coll} = 1.67$ GeV and $\sqrt{s}_{coll} = 3$ GeV (upper mass limit given by `CTParam(60)`). The formed particle excitations are treated as pseudo-resonances instead of strings. Below $\sqrt{s}_{coll} = 1.67$ GeV normal resonance excitation via `anndec` takes place. Above $\sqrt{s}_{coll} = 3$ GeV the normal `UrQMD`-Stringroutine `qstring` is called. Parameters for the unknown resonances are extrapolated from the nearest available resonance.

- `make22.f`: `CTParam(60)` is introduced and part for `iline 27` has been changed according to the above description.
- `string.f`: New subroutine `id2itypnew` is included to obtain a transformation from the quark IDs to `UrQMD`-ityps.

To fix the strangeness production cross section which was reduced because of the new production of high mass resonances instead of strings the branching ratios of high lying resonances are changed to the corresponding branching ratios obtained from string decays of the same mass. Further adjustments are made to keep the particle properties in line with the Particle Data Book 2006 in `blockres.f`.

Other new features

- New Regge-parametrisations for cross-sections at high energies are implemented in `make22.f`.
- Fix the mass distribution of the nucleon resonances N^* via inclusion of the Delta resonances in `iline 14` in `make22.f`: maxnuc $\rightarrow$ maxdel.
- Adjust Ξ- and Ω-production rates in pp-collisions to newly available data via the value of `CTParam(29)`.

Bug fixes

- `blockres.f`: New channel (39) for baryon-antibaryon interactions to allow for string production at high energies.
- `coload.f`: New if-statement in `ctupdate` to signal an array out of bounds error.
- `dwidth.f`: nrejmax 5,000 $\rightarrow$ 1,000,000

  (avoid warning, can be changed back if speed is more important than the details of the mass distribution)

- `erf.f`: function erf → real*8

  (declare the error function as double precision)

- `jdecay2.f`: New variable `ntry` introduced.

- `make22.f`: ntry 100 → 1,000

- `make22.f`: Bug in $p\bar{p}$-cross-section has been fixed. Now pp and $\bar{p}p$ cross sections and scattering processes become similar at high energies.

- `newpart.f`: mprt 200 → 1,000

- `string.f`: near "call getmas" 1. → 1d0

- `urqmd.f`: Reset Pauli-blocking to old value after final decay for next event.

- `urqmd.f`: Implement charge conservation check before and after the call of `scatter`.

- `whichres.f`: in function `pcms`, lt → le

New formats, options and parameters

- `output.f`: New subroutine `urqmdlogo` is implemented to display an `UrQMD` logo which is called in `init.f`.

- `output.f`: **Important: Output format has changed!!!**
  Set `ityp` and `lstcoll` long enough for Pythia output in formats 201, 210, 213, 501, 503: i5 → i11 and i6 → i9.
  Time format in standard event header has been changed (line 6) to format(a7,i9,a13,i12,a9,a20,i7,a20,f11.3).

- `string.f`: Single strange diquark suppression via `CTParam(49)` is set to 0.5 in `input.f`.

- `input.f`: Set `CTParam(29)` = 1, as new default value → no additional double strange diquark suppression.

  `CTParam(59)` = 0.4, scaling factor for leading hadron cross section for Pythia particles.

  `CTParam(60)` = 3, resonance/string transition energy for high mass resonances

  `CTOption(44)` = 1, (default) call Pythia for hard scatterings

  `CTOption(46)` = 0, Density calculation switch (default is baryon density)

- `cascinit.f`: New subroutine `nucfast` if CTOption(24)=2 which provides a faster initialization → needed for cosmic air shower simulations.

UrQMD at LHC energies

To run UrQMD at LHC energies the following arrays need to be adjusted:

- coms.f: Maximum particle number nmax should be increased from 40.000 to 100.000.
- colltab.f: Size of collision table ncollmax should be increased from 10.000 to 30.000.
- output.f: The output format statements need to be changed to accomodate a larger amount of significant digits. The following format statements need to be changed:

```
c standard particle information vector
 201  format(9e16.8,i11,2i3,i9,i5,i4)
LHC--> 201  format(9e24.16,i11,2i3,i9,i5,i4)

c special output for cto40 (restart of old event)
 210  format(9e16.8,i11,2i3,i9,i5,i10,3e16.8,i8)
LHC--> 210  format(9e24.16,i11,2i3,i9,i5,i10,3e24.16,i8)

c special output for mmaker
 203  format(9e16.8,i5,2i3,i6,i5,i4,i5,2e16.8)
LHC--> 203  format(9e24.16,i5,2i3,i6,i5,i4,i5,2e24.16)

c same with index for file15
 501  format(i5,9e16.8,i11,2i3,i9,i5,i3,i15)
LHC--> 501  format(i5,9e24.16,i11,2i3,i9,i5,i3,i15)

c enhanced file16
 503  format(9e15.7,i11,2i3,i9,i5,i4,2i4)
LHC--> 503  format(9e24.16,i11,2i3,i9,i5,i4,2i4)

c same including freeze-out coordinates
 213  format(9e16.8,i11,2i3,i9,i5,i4,8e16.8)
LHC--> 213  format(9e24.16,i11,2i3,i9,i5,i4,8e24.16)
```

Appendix D: Patch to version 2.3

A minor bug in the angular distribution of particles that are produced in string fragmentation (not via Pythia) has been fixed. This bug was not present in the previous published version 1.3. It has led to outgoing particles which have zero momentum in x- and y-direction in elementary p-p collisions. The multiplicities and particle spectra are unchanged by this bugfix. Thanks to Katarzyna Grebieszkow for pointing us to the problem. The following changes have been made

- `make22.f`: New variable pythflag indicates if the process has been handled via Pythia.
- `angdis.f`: The produced particles from `UrQMD`strings have to be rotated afterwards, while this is not necessary for Pythia strings.

A rewritten GNUmakefile has been added including the directory mk with the specifications for the different running platform. gfortran is now used as the standard Linux compiler. The name of the executable has changed, please have a look in the beginning of this guide for an example file how to run the code.
Therefore, some adjustments have been made

- `make22.f`: go to statement in iline 27 has been removed.
- `tabinit.f`, `getmass.f`: pause statements have been removed
- `upmerge.f`: the variable mm_to_fmc is now declared explicitly as real*8

Appendix E: changes from version 2.3 to 3.3

Charm rescattering

Implementation of charmed hadrons with the following itype's: D (133), D* (134), J/Ψ (135), Ψ' (136), χ_c (137)
Rescattering cross sections with pions and rho's included as well, both elastic and inelastic $D + \pi \leftrightarrow D^*$, $\rho + J/\Psi \leftrightarrow D + \bar{D}$ and $\rho + J/\Psi \leftrightarrow D^* + \bar{D}^*$
Cross sections have been parameterized from work done by Zi-Wei Lin: Nucl.Phys.A689:965-979,2001 and Phys.Rev.C62:034903,2000

UrQMD + Hydro

It is possible to run UrQMD with a hydrodynamic evolution for the hot and dense stage of the heavy ion reaction. Default calculations are still the cascade mode calculations. For the physics changes please refer to arXiv:0806.1695. The hydrodynamic evolution is calculated via the SHASTA algorithm.

- New files `1fluid.f`, `bessel.f`, `defs.f`, `uhmerge.f` and new directory with tables for the equation of state (eosfiles) have been added
- `output.f`: new entry f15outhy is implemented to generate output in f15 if hydro is called. nin is set to 9 and one header line and nine particle lines at the beginning and in the end of the hydrodynamic evolution is printed consisting only of zeroes except of the time information.
- New options and parameters:
  1. `CTOption(45)`=1 : hydro mode (default is cascade calculation)

2. `CTOption(47)`=2 : hadron gas EoS (default)
3. `CTOption(47)`=3 : Bag model EoS
4. `CTOption(47)`=5 : chiral + hadron gas EoS
5. `CTOption(48)`=N : flag for only N timesteps of hydro evolution (test case)
6. `CTOption(49)`: spectator switch: 0 (default)→spectators are propagated seperately; 1 → spectators are also put on the hydro grid
7. `CTOption(50)`=1: (additional) f14/f19-output directly after hydro evolution; time is equal to $t_{\mathrm{hydrostart}}$ because of back propagation, resonances decay immediately
8. `CTOption(52)`: freeze-out switch: 0 (default) → isochronous transverse slices; 1 → completely isochronous freeze-out os the whole system
9. `CTOption(53)`: switch for improved momentum generation, default is zero and any other number leads to old prescription with in any case high enough maxima
10. `CTParam(61)` = 0.2 fm : dx is the cell size for the hydro code
11. `CTParam(62)` = 200 : ngr is the grid size of the hydro code
12. `CTParam(63)` = 1. fm : is the minimal $t_{\mathrm{hydrostart}}$
13. `CTParam(64)` = 5 is the factor for the freezeout criterium ($x * \epsilon_0$)
14. `CTParam(65)` = 1 is multiplied with $t_{\mathrm{hydrostart}}$
15. `CTParam(66)`=1.d10 is the rapidity cut for the matter that is put on the hydrodynamic grid, necessary for calculations at higher energies than $E = 160A$ GeV.

- Output in timesteps according to tim statement in inputfile is not consistently possible during the hydrodynmic evolution
- The option for test cases cto 48 does not work when using the bag model equation of state.
- Cut in uhmerge.f to stabilize the Cooper-Frye Monte Carlo has been introduced.
- f15-output has been adjusted. There is now one collision entry before the hydro evolution with `npart` ingoing particles and no outgoing particles and the opposite after the hydro evolution. Therfore the format 502 of the header line has been changed from `format(i1,i8,i4,i7,f8.3,4e12.4)` to `format(i8,i8,i4,i7,f8.3,4e12.4)`. Since there is now an interaction with 0 ingoing particles which was the signal for a new event, the header-line of a new event starts now with a `-1`.

Bug Fixes

- Completely new makefile is written. Please use ”make help” for information.
- `anndec.f`: New subroutine `getbran` which gives reasonable values back even if summed cross section is very small.

- `scatter.f`: Disable elastic scattering for pp collisions now works stable (`CTOption(7)`)
- `blockres.f` Branching ratios for hyperon resonances are adjusted in order to ensure that they sum up to one (thanks to Pasi Huovinen).
- `CTParam(67)` allows for testparticle calculations (default is one testparticle per real particle). If this parameter is used with a value different from one the variable `ncollmax` in `colltab.f` has to increased (by CTParam(67)/2) and `AAmax` should be set to 300*CTParam(67) in `inputs.f`. After that the code has to be recompiled and the file 'tables.dat' has to be removed and newly generated. The output does not account for the testparticles and has to be scaled accordingly. Furthermore, the computig time increases when using this parameter.
- `scatter.f`: The freeze-out coordinates in position space are changed to take into account the formation times of particles produced in string fragmentation processes. Only formed hadrons are able to decouple from the system.

0.1 Changes for u3.3p1

Bugfix in `output.f`: wrong handling of charmed particles

Appendix F: Known problems and inconsistencies in `UrQMD`

- Meson-meson cross sections have discontinuities at the meltpoint to sighera cross section (1.7 GeV) . They should better be treated in a similar way as the meson-baryons.
- More sophisticated treatment of coherent scattering will be important at very high energies.
- Detailed balance is violated due to string decays and other multi-particle (n$\geq$3) decays, e.g. $\omega \rightarrow 3\pi$, for which no inverse reactions are implemented.
- The frame dependence of the code (target vs. projectile vs. CMS-frame) leads to slightly asymmetric $\leq 5\%$ distributions and different yields in forward-backward hemispheres at RHIC.

Thanks

We encourage all users to submit potential problems and bug reports to the following email address: urqmd@urqmd.org .
We would like to thank everybody who has been sending suggestions, bug reports and ideas how to fix them.
Especially, Dr. Hajo Drescher, Dieter Heck and Tanguy Pierog (CORSIKA), Vladimir Uzhinsky, the HADES collaboration.