

UrQMD: Ultra relativistic Quantum Molecular Dynamics

Una breve introducción al generador de eventos UrQMD

Alejandro Guirado García

Universidad de Sonora

02 de Febrero del 2021



Contenido

- 1 Introducción
 - ¿Qué es UrQMD?
 - Modelos utilizados en UrQMD
 - Rangos de energías
 - UrQMD PID: Id de partícula
- 2 ¿Cómo usarlo?
 - Descargar de algún repositorio
 - Manual oficial de UrQMD
 - Compilar los archivos del código fuente
 - Editar archivo de entrada
 - Ejecutar UrQMD
 - Archivos de salida
 - Análisis con ROOT
 - Macro de análisis
- 3 Conclusión

Introducción: ¿Qué es UrQMD?

El generador de eventos llamado "Dinámica Cuántica Molecular ultra-Relativista" (UrQMD, por sus siglas en Inglés) es utilizado como generador de eventos en colisiones de iones pesados debido a que es rápido, fácil de utilizar, hay bastante documentación disponible, tiene varios métodos de propagación de partículas, es cómodo personalizarlo para diferentes tareas y está escrito en Fortran.

Descripción oficial

UrQMD está diseñado como herramienta multipropósito para estudiar una amplia variedad de efectos relacionados con iones pesados desde la multifragmentación y el flujo colectivo de partículas hasta la producción y correlaciones de partículas.

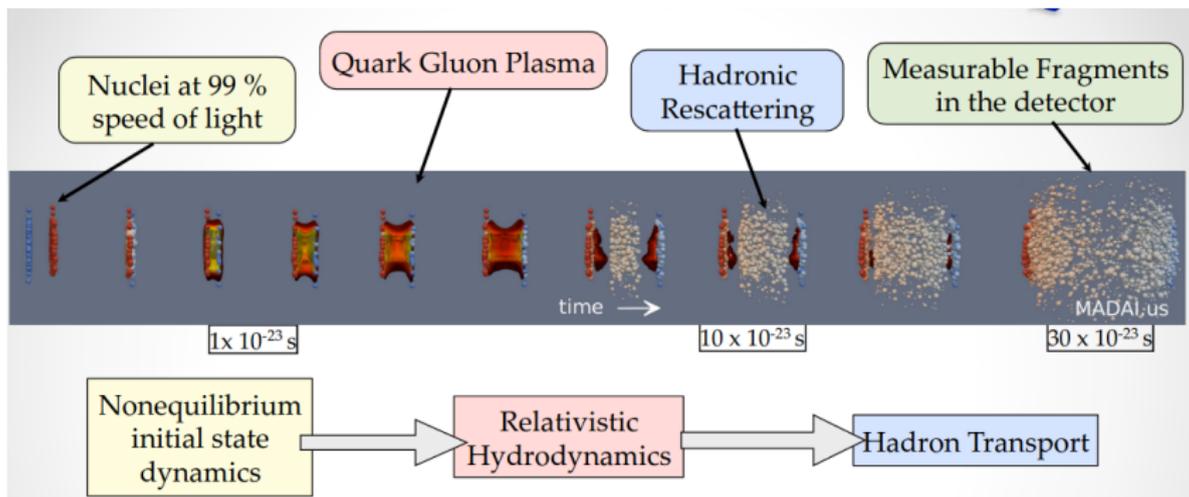


Figure 1: Evolución temporal de los eventos en UrQMD. (Dependiendo del modelo utilizado). Más información en: UrQMD-Presentation.

Modelos utilizados en UrQMD

En UrQMD los modelos de propagación de partículas pueden ser dos modelos de propagación :

- Ecs. canónicas de Hamilton: *Microscopic Models for Ultrarelativistic Heavy Ion Collisions*. arXiv:nucl-th/9803035.
- Ecuación de Boltzman: *Relativistic Hadron-Hadron Collisions in the Ultra-Relativistic Quantum Molecular Dynamics Model* arXiv:hep-ph/9909407v1
- Modelo híbrido: *Fully integrated transport approach to heavy ion reactions with an intermediate hydrodynamic stage*. arXiv:0806.1695

Rangos de energías

Los autores recomiendan usar UrQMD en rangos de 2 *GeV* a 200 *GeV*. Aunque se puede usar a energías mayores, p.e. del LHC(*TeV*), no se recomienda debido al exceso de partículas que se pueden producir.

- Para energías menores a 100 *GeV* los hadrones se modelan aun de forma puntual.
- A energías mayores a 100 *GeV* se recomienda modelar la dinámica temprana con aproximaciones hidrorrelativistas.

UrQMD PID: Id de partícula

Particle	PID	carga
Protón	1	1
Neutrón	1	0
Pión	101	1,0,-1

Table 1: PID de algunos hadrones en UrQMD.

nucleon	delta	lambda	sigma	xi	omega
N_{938}	Δ_{1232}	Λ_{1116}	Σ_{1192}	Ξ_{1315}	Ω_{1672}
N_{1440}	Δ_{1600}	Λ_{1405}	Σ_{1385}	Ξ_{1530}	
N_{1520}	Δ_{1620}	Λ_{1520}	Σ_{1660}	Ξ_{1690}	
N_{1535}	Δ_{1700}	Λ_{1600}	Σ_{1670}	Ξ_{1820}	
N_{1650}	Δ_{1900}	Λ_{1670}	Σ_{1750}	Ξ_{1950}	
N_{1675}	Δ_{1905}	Λ_{1690}	Σ_{1775}	Ξ_{2030}	
N_{1680}	Δ_{1910}	Λ_{1800}	Σ_{1915}		
N_{1700}	Δ_{1920}	Λ_{1810}	Σ_{1940}		
N_{1710}	Δ_{1930}	Λ_{1820}	Σ_{2030}		
N_{1720}	Δ_{1950}	Λ_{1830}			
N_{1900}		Λ_{1890}			
N_{1990}		Λ_{2100}			
N_{2080}		Λ_{2110}			
N_{2190}					
N_{2200}					
N_{2250}					

Figure 2: Bariones y resonancia de bariones incluidas en el modelo de UrQMD.

¿Cómo usar UrQMD?

De manera simple, para poder empezar a generar eventos en UrQMD, se necesita:

1. Descargar UrQMD.
2. Asegurarse de tener una versión de compilador GNU Fortran compatible. Una versión menor a 9 es útil.
3. Generar los archivos ejecutables.
4. Correr el archivo *runqmd.bash*.
5. Correr la macro *urqmdtoroot.C*.
6. Analizar resultados con ROOT.

Link de descarga

El código de UrQMD se puede descargar del siguiente link:

`urqmd-3.4.tar`

Mover la carpeta comprimida al workspace y descomprimir el archivo desde la terminal o el explorador de carpetas.

Manual de UrQMD

El manual de UrQMD tiene toda la información necesaria a nivel de usuario. Desde referencias, distintas configuraciones que se pueden utilizar, comentarios de compatibilidad y como utilizar este generador de eventos. El manual se encuentra en `../urqmd/docs/urqmd-user.pdf`.

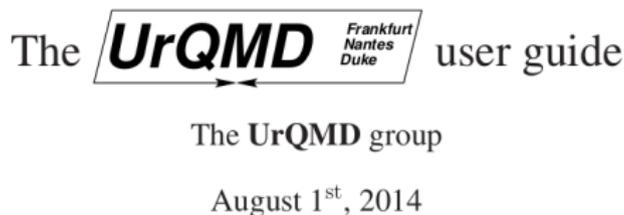


Figure 3: Portada del manual de urqmd para la versión 3.4.

Compilar los archivos del código fuente

Para saber la versión del compilador fortran que se tiene instalado, simplemente se usa el comando `gfortran --version`.

```
(base) kali@kali:~$ gfortran-8 --version
GNU Fortran (Debian 8.4.0-5) 8.4.0
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

(base) kali@kali:~$ gfortran --version
GNU Fortran (Debian 10.2.0-19) 10.2.0
Copyright (C) 2020 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Figure 4: Comando para verificar la versión instalada del compilador gfortran.

Después de tener instalado el compilador adecuado. Se utiliza el comando *make* para compilar los archivos de fortran siguiendo las instrucciones del archivo *GNUmakefile*.

```
gfortran-8 -O3 -mmodel=medium -c quadri.f90 -o obj_x86_64/quadri.o
gfortran-8 -O3 -mmodel=medium -c saveinfo.f -o obj_x86_64/saveinfo.o
gfortran-8 -O3 -mmodel=medium -c scatter.f -o obj_x86_64/scatter.o
```

```
#####
##                                     ##
## UrQMD 3.4      University of Frankfurt ##
##                                     ##
##                                     http://urqmd.org ##
##                                     bleicher@th.physik.uni-frankfurt.de ##
##                                     #####
##                                     ##
## Please cite when using this model: ##
## S.A.Bass et al., Prog.Part.Nucl.Phys. 41 (1998) 225 ##
## M.Bleicher et al., J.Phys. G25 (1999) 1859 ##
##                                     ##
##                                     #####
## UrQMD uses Pythia6.409 by T. Sjostrand ##
##                                     ##
##                                     #####
## If hydrodynamic evolution is switched on (CT0 45 1) ##
## UrQMD uses the SHASTA algorithm by D. Rischke ##
## Please cite when using the hybrid code: ##
## D. Rischke et al., Nucl.Phys. A 595 (1995) 346 ##
## D. Rischke et al., Nucl.Phys. A 595 (1995) 383 ##
## H. Petersen et al., Phys.Rev. C78 (2008) 044901 ##
##                                     ##
##                                     #####
```

```
Writing new table ...
O.K.
event#           1 1612141305
(W) No collision in event 1
no. of collisions = 0.0 (per event)
final particles   = 2.0 (per event)
empty events     : 1 = 100.0%
```

Editar inputfile

El archivo de entrada a UrQMD se llama `inputfile` (sin extensión).

```
pro 208 82//Az Z
tar 208 82//Az Z
nev 1 //Numero de eventos
imp 0.// Parametro de impacto fijo igual a 0 fm
elb 20.// Energia de lab igual a 20 GeV
tim 200 200// Tiempo final 200fm en pasos de 200fm
f13 //Archivo de salida test.f13 desactivado
#f14 //Archivo de salida test.f14 activado
f15
f16
f19
f20
```

Resultados

Opciones y parámetros especiales

En UrQMD se pueden agregar opciones(**CTO**) y parámetros(**CTP**) especiales.

La información detallada de estas opciones se encuentran en el manual y no es necesario utilizarlas para poder generar eventos, son simplemente para usarlas en caso de querer realizar un estudio específico.

Ejecutar UrQMD

Para ejecutar los archivos compilados de UrQMD es simple. Basta con cargar el archivo *runqmd.bash*. Este archivo se carga escribiendo desde la terminal en la carpeta de UrQMD el comando: *source runqmd.bash*.

En la siguiente diapositiva se muestra el archivo *runqmd.bash*.

```

run=${1-nothing}
exename="urqmd.$(uname -m)"

export ftn09=inputfile
export ftn13=test.f13
export ftn14=test.f14
export ftn15=test.f15
export ftn16=test.f16
export ftn19=test.f19
export ftn20=test.f20

# check if file lhc exists, or if user specifically asked for lhc run:
if [ -e lhc ] && [ "$run" ≠ "all" ] || [ "$run" = "lhc" ]; then
    if [ -x $exename.lhc ]; then
        echo "Running the LHC version of UrQMD"
        time ./$exename.lhc
    else
        echo "LHC executable not found, please compile with 'make lhc' first"
        exit 1
    fi
else
    if [ -x $exename ]; then
        echo "Running UrQMD"
        time ./$exename
    else
        echo "UrQMD executable not found, please compile with 'make' first"
        exit 1
    fi
fi

```

Figure 5: Script del archivo runqmd.bash

Resultados en la terminal

Resultados

Inputfile

```
Calculation at sroot.le.20 A GeV:
parameter nmax in coms.f may be decreased!
(info) no output on unit 13
(info) no output on unit 15
(info) no output on unit 16
(info) no output on unit 19
(info) no output on unit 20
event#          1 1612147394
(info) dsigma: calculating constants for ang. dist.
(info) dsigma: calculation finished
no. of collisions = 2025.0 (per event)
final particles   = 615.0 (per event)
empty events      :      0 = 0.0%

real    0m0.821s
user    0m0.753s
sys     0m0.017s
```

Archivos de salida

Los archivos de salida son:

- test.f13
- test.f14
- test.f15
- test.f16
- test.f19
- test.f20

La información detallada se puede encontrar en el manual de UrQMD. Sin embargo, explorando un poco cada archivo...

test.f13 y test.f14

Estos dos archivos de salida contienen toda la información relevante de las partículas para la mayoría de los análisis. testf14 test.f16

013	014	015	016	contents
		1		ind: index of particle (see CTOption (56))
1	1	2	1	<i>t</i> : 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	<i>E</i> : 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	<i>m</i> : 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	<i>ch</i> : charge of particle
13	13	14	13	parent collision number (see Table 10)
14	14	15	14	N_{coll} number of collisions
		16		<i>S</i> : strangeness
15	15		15	parent process type (see Table 11)
		17		history information (debugging only)

test.f15

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 II)
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

Figure 6: Información que se puede obtener del archivo test.f15

test.f16

			16*	$ityp_1^{old}$: particle-ID of parent particle # 1
			17*	$ityp_2^{old}$: particle-ID of parent particle # 2

Figure 7: El archivo `testf16` tiene la misma información que el archivo `test.f14`, también se agrega los datos de decaimientos.

test.f19 y test.f20: OSCAR files

Estos archivos son generados a partir de las lineamientos del grupo OSCAR, los cuales fueron diseñados para facilitar el acceso y uso de los datos.

column#	format	contents
1	(i10,2x)	event counter
2	(i10,2x)	number of particles in event
3	(f8.3,2x)	impact parameter (in fm/c)
4	(f8.3)	rotation of the event plane (fixed to 0 in UrQMD)

Figure 8: Información disponible en los encabezados de ambos archivos OSCAR.

column#		format	contents
OSC97	OSC99		
1	1	(i10, 2x)	particle counter (different meaning in OSC97/99, see text)
2	2	(i10, 2x)	HEP particle ID (see text)
	3	(i10, 2x)	particle state (always "0" in UrQMD)
3	4	(e12.6, 2x)	p_x : particle momentum
4	5	(e12.6, 2x)	p_y
5	6	(e12.6, 2x)	p_z
6	7	(e12.6, 2x)	E
7	8	(e12.6, 2x)	m : particle mass
8	9	(e12.6, 2x)	r_x^{fr} : Freeze-out position
9	10	(e12.6, 2x)	r_y^{fr}
10	11	(e12.6, 2x)	r_z^{fr}
11	12	(e12.6, 2x)	t^{fr}

Figure 9: Información disponible los archivos OSCAR. OSC97 es el archivo test.f19 y OSC99 es el archivo test.f20.

Análisis con ROOT

Después de realizar la generación de eventos con UrQMD y tener la información en los archivos de salida anteriormente comentados, es necesario saber como se puede analizar toda la información obtenida.

urqmdtoroot.C

Para poder utilizar el software de análisis ROOT, es necesario tener la información guardada en un árbol de ROOT. Para realizar esto, se utiliza la macro *urqmdtoroot.C*. → [Link de descarga](#).

Comentarios sobre *urqmdtoroot.C*

Esta macro crea un arbol de ROOT a partir de una *struct* de C++.

```
struct particula_t
{
Float_t time,X,Y,Z,E,Px,Py,Pz,Pt,P,Eta,m,id,isoespin,charge,lastcoll,
numbercoll,history,frezetime,frezeX,frezeY,frezeZ,frezeE,frezePx;
Float_t frezePy,frezePz,frezePt,frezeP,frezeEta;
} PARTICLE;
    particula_t  particle;

struct evento_t {Int_t nparticles,nevent;}  EVENT;
    evento_t  event;

TTree *tree = new TTree("T","An example of ROOT tree with a few branches");
tree->Branch("particle",&particle,"time:X:Y:Z:E:Px:Py:Pz:Pt:P:Eta:m:id:isoespin:
charge:lastcoll:numbercoll:history:frezetime:frezeX:frezeY:frezeZ:frezeE:frezePx:
frezePy:frezePz:frezePt:frezeP:frezeEta");
tree->Branch("event",&event,"ntrack/I:nevent/I");
```

Figure 10: Sección de código donde se crean las estructuras para llenar el árbol de ROOT. Además, se crean los objetos tree, branch de partículas y branch de eventos.

Comentarios sobre *urqmdtoroot.C* II

Es importante saber que se puede crear nuevas variables dentro del archivo de *urqmdtoroot.C* para guardar en el archivo y así, sea facilitar el análisis con otras macros. Por ejemplo, en el archivo para descargar se incluye la pseudorapidez y momento transverso.

```
Pt = TMath::Sqrt(Px*Px+ Py*Py);  
P  = TMath::Sqrt(Px*Px+Py*Py+Pz*Pz);  
  
if(TMath::Abs(P- Pz)>0.000000001)  
{  
    Eta = 0.5*TMath::Log((P+Pz)/(P-Pz));  
}
```

Figure 11: Definición de variables al crear el archivo de ROOT. En este caso, se agregaron el momento transverso, momento total y pseudorapidez.

Correr *urqmdtoroot.C*

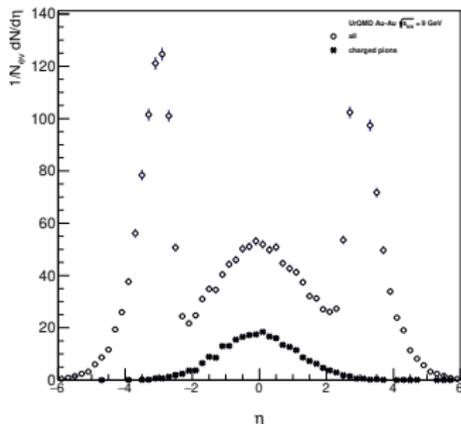
Para correr la macro:

1. Colocar la dirección del archivo test.f14 en *urqmdtoroot.C* y del archivo de salida en ROOT.
2. En la terminal colocarse en la carpeta donde se encuentra el archivo *urqmdtoroot.C*.
3. En la terminal cargar ROOT y escribir el comando: `.x urqmdtoroot.C`.

Macro de análisis

La macro de análisis ejemplo se llama *LeeEntradas.C*, simplemente carga los datos del archivo test.f13 o test.f14 despues de haber guardado los datos en un arbol de ROOT con la macro de urqmdtoroot.C y guarda dos histogramas seleccionando unicamente protones, piones cargados.

Descárgalo aquí!



Conclusión

UrQMD es un generador de eventos muy verstátil, fácil de usar y de instalar. Además, ofrece resultados aceptables y con un costo de procesamiento no tan elevado.

MExNICA

En la colaboración MExNICA se ha utilizado UrQMD para varios trabajos:

- *Hyperons from Bi+Bi collisions at MPD-NICA: Preliminary analysis of production at generation, simulation and reconstruction level* arXiv:2010.12593
- *Two-component source to explain Λ and Λ^- global polarization in non-central heavy-ion collisions* arXiv:2006.10015
- *A beam-beam monitoring detector for the MPD experiment at NICA* arXiv:1809.10553

En caso de cualquier duda sobre la instalación o uso de UrQMD pueden contactarme.

Gracias!!