

# EPOS model

(Installation - Generation - basic macro - transport -reconstruction)

3rd Computing/Analysis Workshop of the MexNICA  
Collaboration

Valeria Zelina Reyna Ortiz

02.02.2021

# El modelo de EPOS

EPOS es un generador de eventos de Montecarlo para interacciones hadrónicas a MB que se usa principalmente para simular colisiones de iones pesados y rayos cósmicos. [\[1\]](#). Y está basado principalmente en Parton-Based Gribov Regge Theory. [\[2\]](#)

Su distribución es libre junto al paquete Cosmic Ray Monte Carlo generator en: <https://web.i kp.kit.edu/rulrich/crmc.html>.

# Principio básico

## *Parton ladder o "Pomeron":*

Representan la evolución de partones del lado del proyectil y del target hacia el centro. [\[3\]](#)

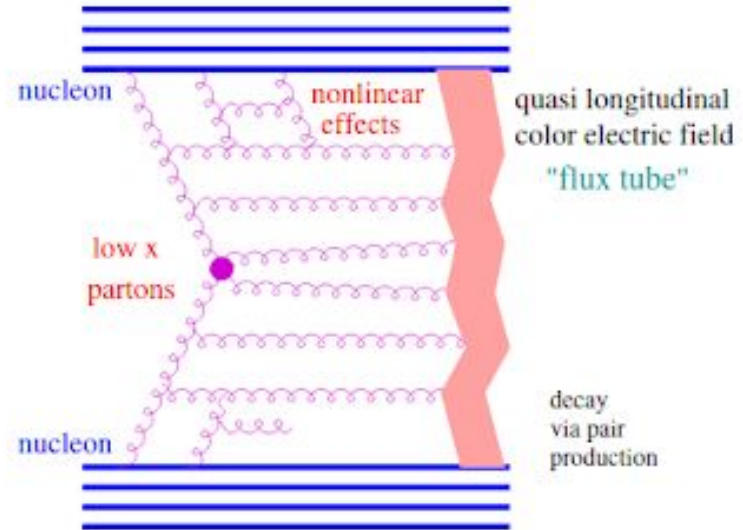


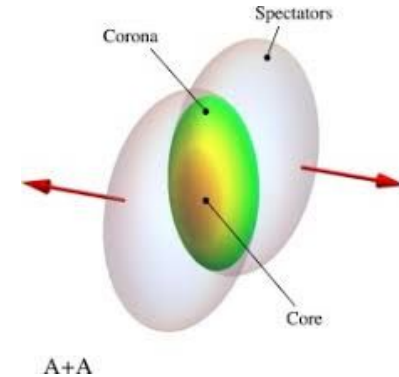
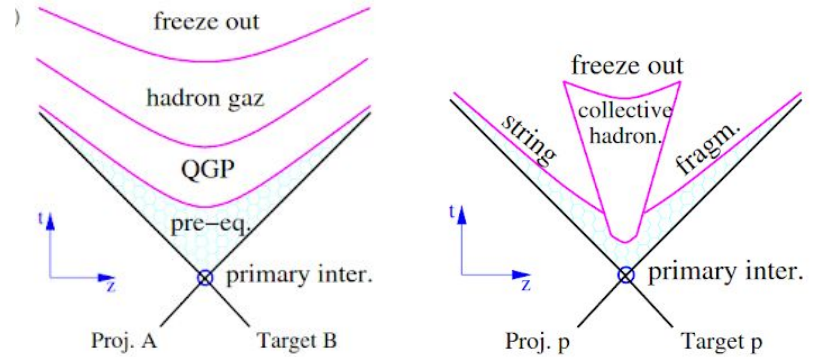
Figure 2. Elementary interaction in the EPOS model.

# Collective hadronization.

Las condiciones iniciales de este estado no son las de la colisión.

La condición inicial del estado del sistema es después del estado de radiación de Jets, pero antes de que los partones se dispersen como en HI.

Se considera un tiempo  $t_0$  antes de que se formen los hadrones en la colisión, donde se distinguen segmentos de cuerda en áreas de alta y baja densidad.

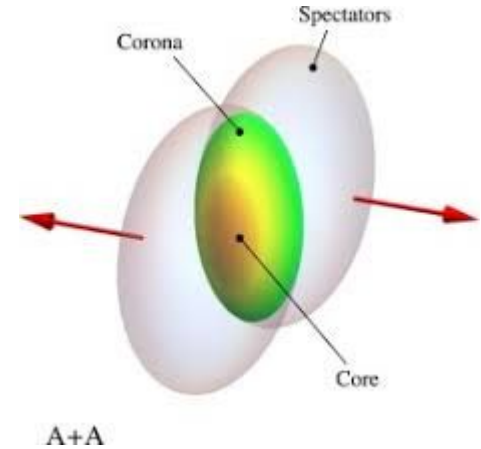


# Core - Corona

La discusión del Core es única en EPOS, otros modelos tienen un enfoque en la densidad de la Corona.

El Core está formado de clusters que se hadronizan vía procedimientos microcanónicos.[4] El proceso conserva energía, momento y sabor.

Una parte de los segmentos de cuerda se hadronizan hacia la corona y una parte se usa para crear un Core con hadronización colectiva.



# Instalación

Pre-requisitos: Fairsoft (Boost, HepMC, Root), FastJet. Creamos el directorio de instalación:

```
$ mkdir -p /home/$USER/Software/EPOS  
$ export EPOS= /home/$USER/Software/EPOS  
$ cd $EPOS
```

Instalación de FastJet:

```
$ wget http://fastjet.fr/repo/fastjet-3.3.4.tar.gz  
$ tar zxvf fastjet-3.3.4.tar.gz  
$ cd fastjet-3.3.4.tar.gz  
$ ./configure --prefix=/home/$USER/Software/EPOS/fastjet-3.3.4/build  
$ make  
$ make check  
$ make install
```

## Declaramos variables de ambiente \*:

```
$ export FAIRSOFT=/home/$USER/Software/NICA/fairsoft/install
$ export HEPMC_ROOT=$FAIRSOFT
$ export BOOST_ROOT=$FAIRSOFT
$ export ROOTSYS=$FAIRSOFT
$ export FASTJET_PREFIX=/home/$USER/Software/EPOS/fastjet-3.3.4
```

## Descargar e instalar:

```
$ wget
https://gitlab.ikp.kit.edu/AirShowerPhysics/crmc/repository/archive.tar.gz?ref=
=svn/crmc.v1.7.1 cerm.v1.7.1.tgz
$ tar -xvzf cerm.v1.7.1.tgz
$ cd cerm.v1.7.1.tgz
$ mkdir build
$ cd build
$ cmake ..
$ make
$ make install
```

# Generar eventos

Verificar la instalación: `$ make test ARGS=-V`

Ejecutar para ayuda: `$ /bin/crmc -h`

Ejemplos: 100 eventos pp a 7Tev con EPOS-LHC:

```
$ bin/crmc -o hepmc -S7000 -n100 -m0
```

100 eventos PbPb A 1.38TeV con EPOS 1.99:

```
$ bin/crmc -o hepmc -p1380 -P-1380 -n100 -i208 -I208 -m1
```

1000 eventos pp, salida root , 10 GeV con EPOS 1.99 con nombre diferente:

```
bin/crmc -o root -f crmc_epos199_p_p_10GeV_100000.root -S10 -n1000 -m1
```

# Macro para leer eventos

En el Github:

```
$ git clone https://github.com/MexNICA/Epos.git  
$ cd basicRead
```

Código básico:      `read_Epos.C` y `read_Epos.h`

```
Correr:      $ root -l  
             $ root [0]                .L read_Epos.C  
             $ root [1]                read_Epos t  
             $ root [2]                t.Loop()
```

# Transporte a MPD-ROOT (con un ejemplo)

Salida en formato hepmc, (1000 eventos pp, 9GeV, con EPOS 1.99):

```
$ bin/crmc -o hepmc -f crmc_epos199_p_p_9GeV_1000 -S9 -n1000 -m1
```

En la carpeta de Transporte:

```
$ mv crmc_epos199_p_p_9GeV_1000 /home/$USER/EPOS/mpdroot  
$ cd /home/$USER/EPOS/mpdroot
```

Editar: **runMC.C** (definir número de eventos y nombre de salida)

Dentro del Macro: `infile = "crmc_epos199_p_p_9GeV_1000"`

Cargar MpdRoot y correr runMC.C

```
$ source ../../mpdroot/build/config.sh  
$ root -b runMC.C
```

# Reconstrucción

La reconstrucción es análoga a la de los otros generadores,

Editar: **reco.C** (definir número de eventos y nombre de entrada y salida)

Dentro del Macro: `inFile = "crmc_salida_transporte.root"`

Correr reco.C :

```
$ root -b reco.C
```

Se generan los archivos de salida estándar **mpddst.root** y **mpddst.Minidst.root**. El código para leer estas salidas sigue siendo **readDST.C**