Working with PHQMD short introduction

30.09.2024, Susanne Gläßel

Run PHQMD: steps

1) Run PHQMD code
2) Apply stabilisation routine
3) Convert PHQMD-output into one file (detector-input)

Parton-Hadron-Quantum-Molecular Dynamics

= n-body microscopic transport approach for the description of heavy-ion dynamics with dynamical cluster formation

PHSD + QMD + MST

Relativistic considerations

but: mean-field potentials
=> correlations are smeared out

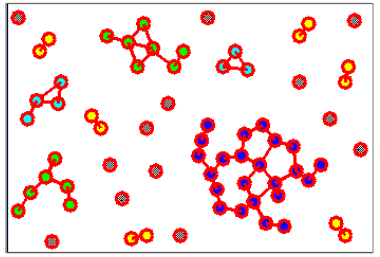
Correlations between baryons

n-body transport approach

=> formation of clusters due to potential interactions

Cluster recognition

search for accumulations of particles in coordinate space



PHQMD

Initial A+A collisions → Formation of QGP → Partonic phase → Hadronization → Hadronic phase

QMD Initialization nuclei → propagation of baryons

local $\varepsilon > \varepsilon_c$: dissolution of pre-hadrons

propagation of partons

interactions of partons

PHSD Primary collisions pre-hadronic states

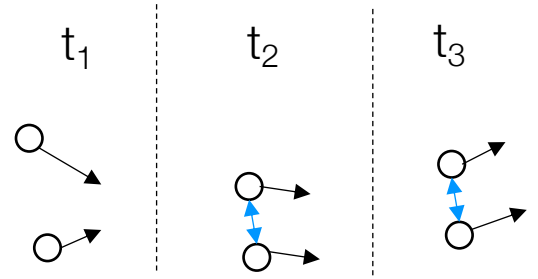
interactions of hadrons

propagation of mesons

MST

J. Aichelin et al., PRC 101 (2020) 044905 PHSD: W. Cassing, E. Bratkovskaya, PRC 78 (2008) 034919; NPA831 (2009) 215; W. Cassing, EPJ ST 168(2009)

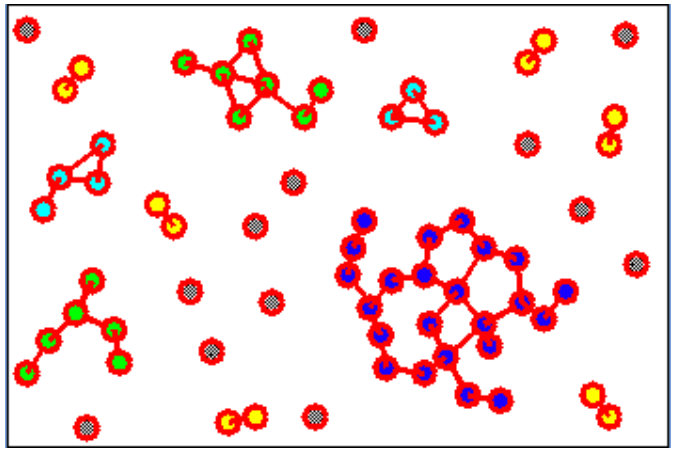
PHQMD Cluster recognition
Minimum Spanning Tree (MST)



Cluster formation due to attractive potential between nucleons in **QMD**

Cluster recognition criterion: distance of nuclei

Algorithm: search for accumulations of particles in coordinate space



1. Two particles i & j are bound if:

$$|r_i - r_j| < 4.0 \text{ fm}$$

2. Particle is bound to cluster if bound with at least one particle of cluster

cut on binding energy is applied during conversion

*R.K. Puri, J.Aichelin, J.Comp. Phys. 162 (2000) 245-266

Parton-Hadron-Quantum-Molecular Dynamics

= n-body microscopic transport approach for the description of heavy-ion dynamics with dynamical cluster formation

PHSD + QMD + MST

Relativistic considerations

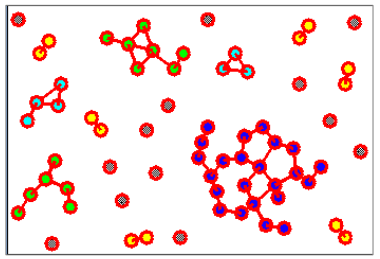
but: mean-field potentials
=> correlations are smeared out

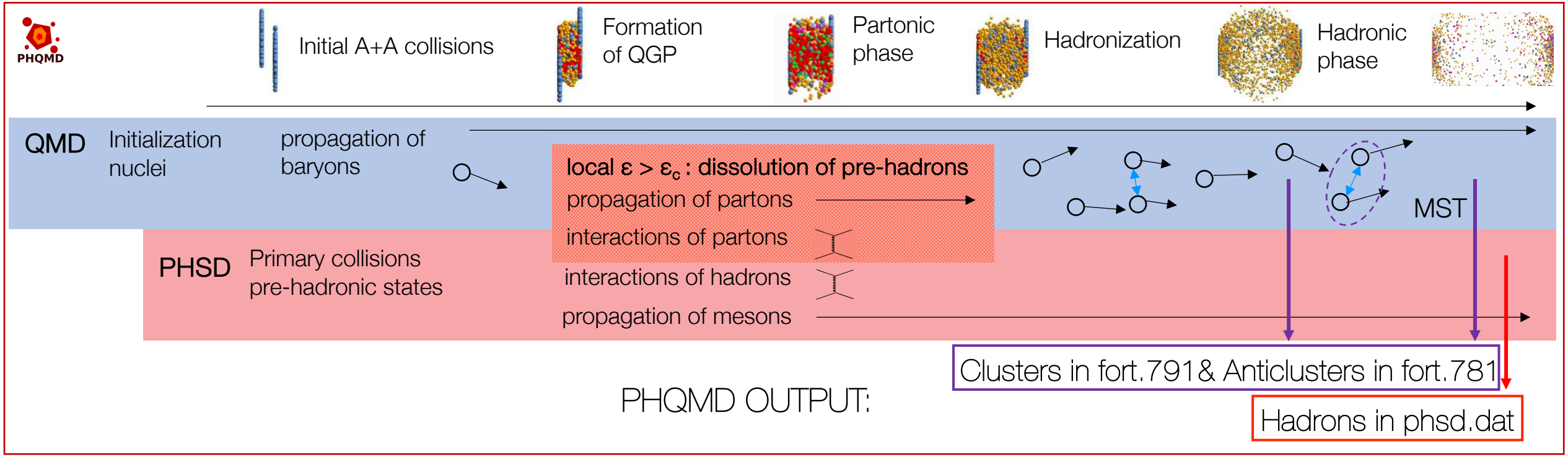
Correlations between baryons

n-body transport approach (Skyrme potential)
=> formation of clusters due to potential interactions

Cluster recognition

search for accumulations of particles in coordinate space





J. Aichelin et al., PRC 101 (2020) 044905 PHSD: W. Cassing, E. Bratkovskaya, PRC 78 (2008) 034919; NPA831 (2009) 215; W. Cassing, EPJ ST 168(2009)

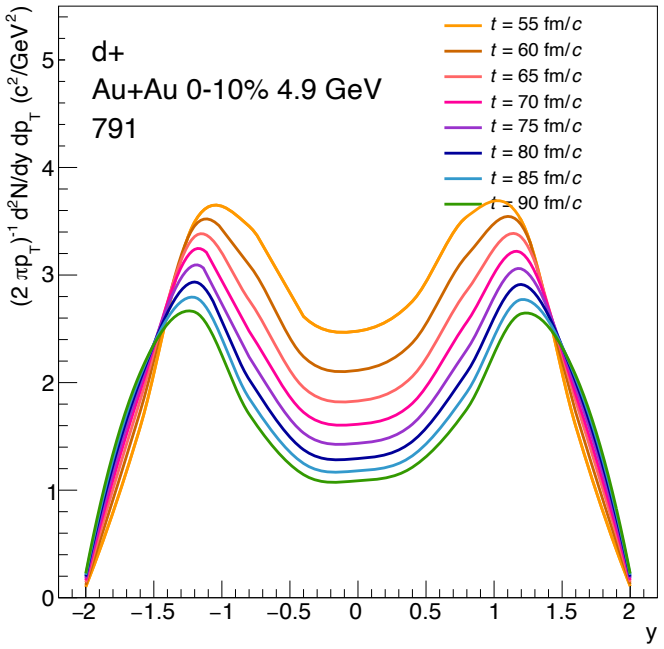
Run PHQMD: steps

1) Run PHQMD code
2) Apply stabilisation routine
3) Convert PHQMD-output into one file (detector-input)

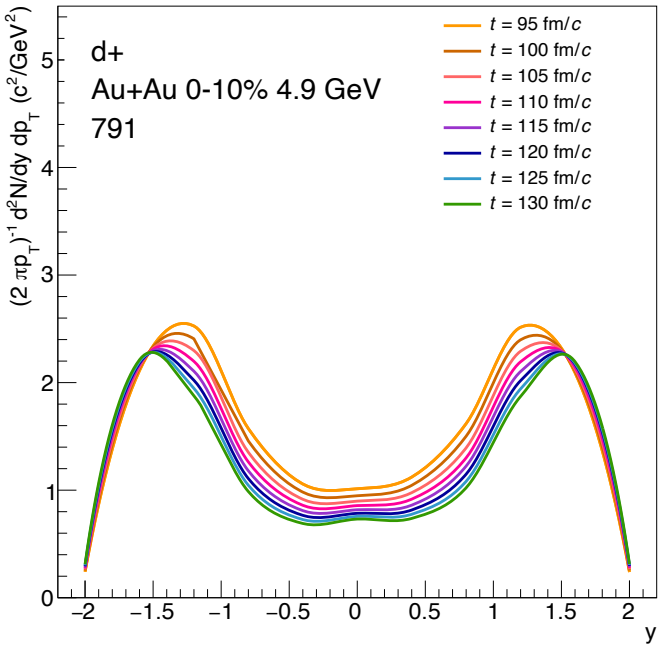
MST clusters for different times during the HIC

Deuteron y-distribution fort.791

t = 55 fm/c – 90 fm/c



t = 95 fm/c – 130 fm/c



Decreasing of cluster multiplicity:

- at early times: baryons collide with other baryons & escape cluster
- at all times: artificial dissolving of clusters, because:

QMD is a semiclassical model:

Skyrme potential is not relativistic

→ different times in computational frame and centre of mass frame of clusters

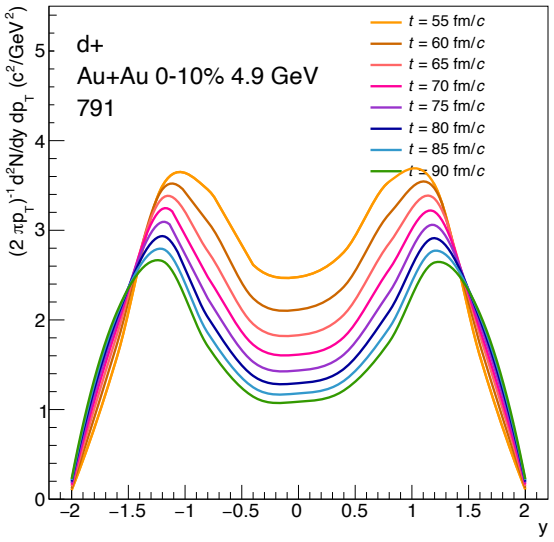
→ numerical artifact when calculating potentials for propagation

→ Stabilistation routine to overcome artificial dissolving

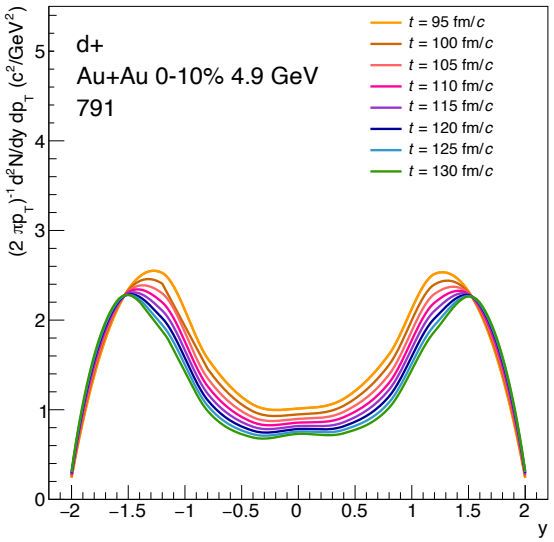
Stabilization of cluster output

Before stabilisation

t = 55 fm/c – 90 fm/c

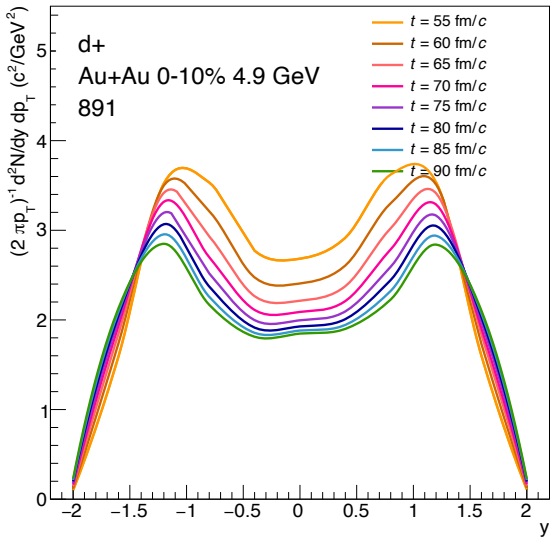


t = 95 fm/c – 130 fm/c

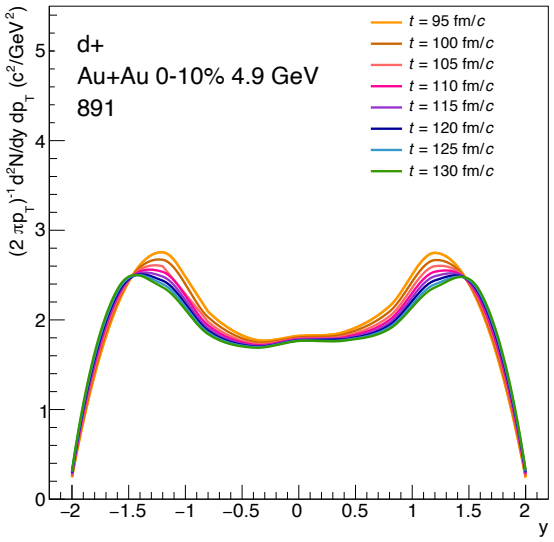


With stabilisation procedure (aMST)

t = 55 fm/c – 90 fm/c



t = 95 fm/c – 130 fm/c

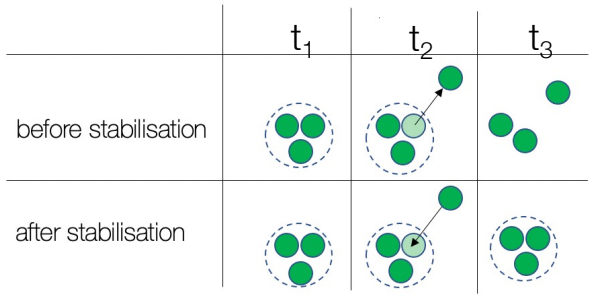


Stabilisation procedure Advanced MST (aMST):

Consider a cluster as stable after all baryons in the cluster are frozen

1) Check if dissolving of a cluster was artificial, i.e . if all cluster-baryons had their last collision.
2) Restore cluster.

fort.791&
fort.781 ↓
fort.891&
fort.881



Run PHQMD: steps

1) Run PHQMD code
2) Apply stabilisation routine
3) Convert PHQMD-output into one file (detector-input)

Convert PHQMD output into detector input

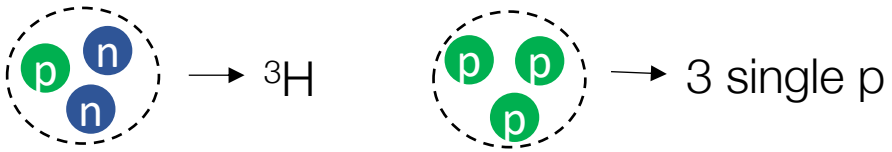
A) Get hadrons from phsd.dat (& remove hadrons bound in clusters)

B) Get clusters from fort.891 / 881

MST recognises clusters / anti-clusters independently of their physical existence

Conversion:

1) Check if cluster binding energy > 0
2) Check if cluster is physical (cluster table A < 9)
   & if not, count baryons as single baryons
   options for big clusters (A > 7):
   - count independent of physical existence
   - don't count



Convert PHQMD output into detector input

PHQMD output

inputPHSD

hadrons from *phsd.dat*

stable clusters from *fort.891*

stable anticlusters from *fort.881*

Detector input: unigen-file (root)

Hadrons & clusters & anticlusters

cluster-table

Nuclei	pdgcode	p	q=0	Λ	σ
Deuterium	1000010020	1	1	0	0
Tritium	1000010030	1	2	0	0
Helium3	1000020030	2	1	0	0
Helium4	1000020040	2	2	0	0
Helium6	1000020060	2	4	0	0
Helium8	1000020080	2	6	0	0
Lithium6	1000030060	3	3	0	0
Lithium7	1000030070	3	4	0	0
Lithium8	1000030080	3	5	0	0
Lithium9	1000030090	3	6	0	0
Berilium7	1000040070	4	3	0	0
Berilium9	1000040090	4	5	0	0
Bor8	1000050080	5	3	0	0
Hypertriton	3004	1	2	1	0
H4L	3005	1	3	1	0
H4LL	3008	1	4	2	0
H5LL	3010	1	5	2	0
Helium4L	3006	2	2	1	0
Helium5L	3007	2	3	1	0
Helium6LL	3011	2	4	2	0
Helium7LL	1020020070	2	5	2	0

The cluster_table.root can be modified by changing cluster_table.dat and by recreating it with write_clustertree.C.

UniGen output (root)

UEvent

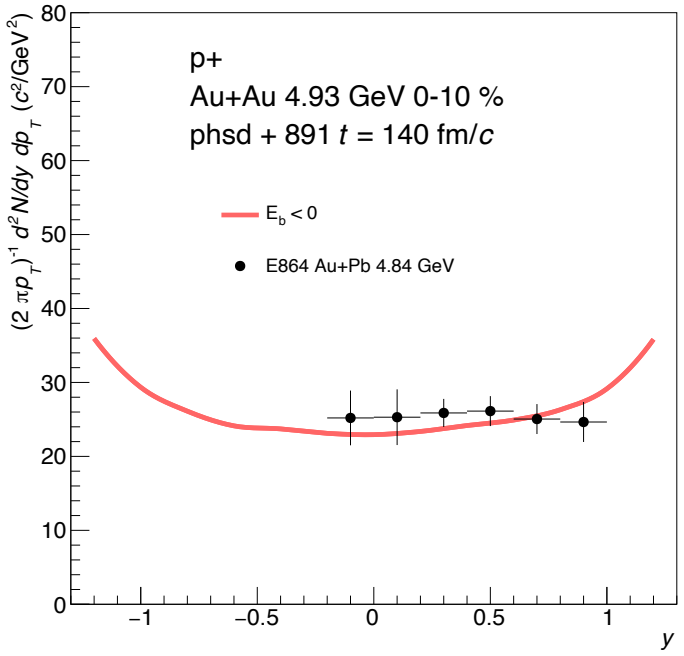
```
Int_t          fEventNr;        // Event number
Double_t       fB;              // Impact parameter (fm)
Double_t       fPhi;            // Reaction plane angle
Int_t          fNes;            // Number of event steps
Int_t          fStepNr;         // Event step number
Double_t       fStepT;          // Event step time
Int_t          fNpa;            // Number of particles
TString        fComment;        // Generator-specific information
TClonesArray* fParticles;       // Array of particles
```

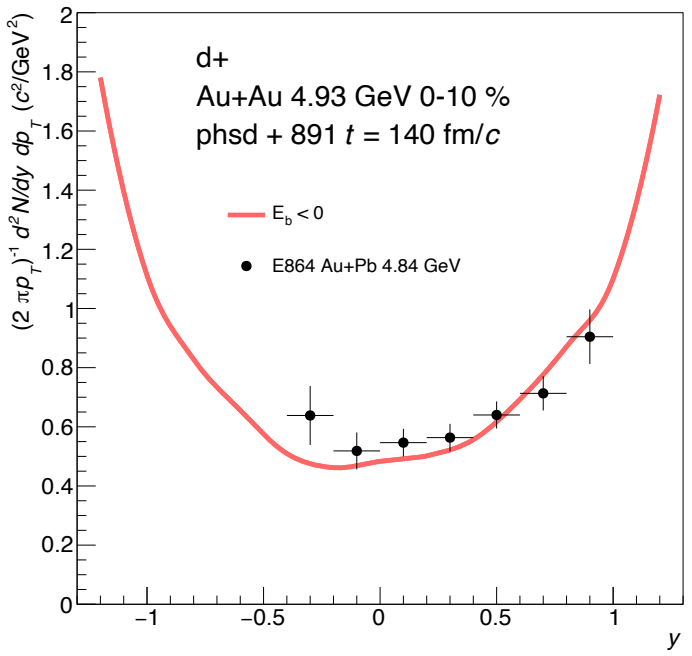
*available in actual version, other information can be added

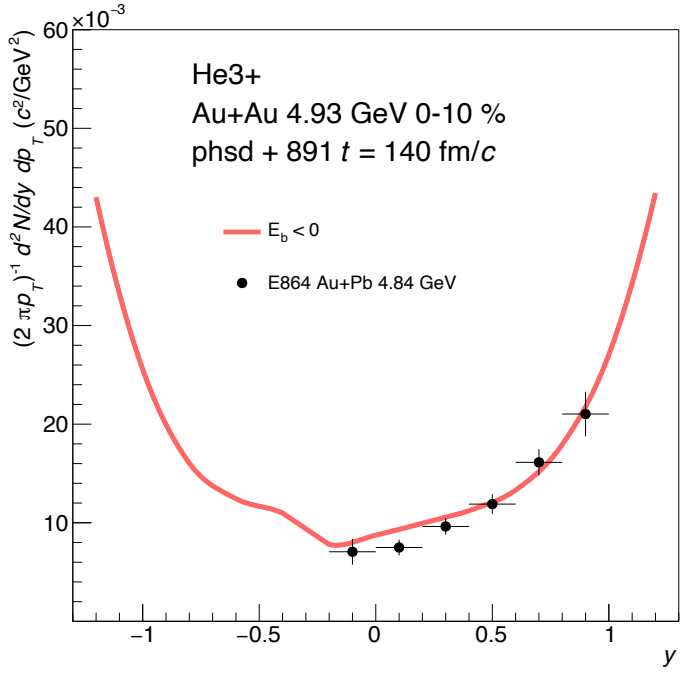
UParticle

```
Int_t    fIndex;          // index of this particle
Int_t    fPdg;            // PDG code
Int_t    fStatus;         // Status
Int_t    fParent;         // Index of parent
Int_t    fParentDecay;    // Parent decay index
Int_t    fMate;           // index of last collision partner
Int_t    fDecay;          // decay index (-1 if not decayed)
Int_t    fChild[2];       // index of first and last child
Double32_t fPx;           // px (GeV)
Double32_t fPy;           // py (GeV)
Double32_t fPz;           // pz (GeV)
Double32_t fE;            // Energy (GeV)
Double32_t fX;            // x (fm)
Double32_t fY;            // y (fm)
Double32_t fZ;            // z (fm)
Double32_t fT;            // t (fm)
Double32_t fWeight;       // weight
```

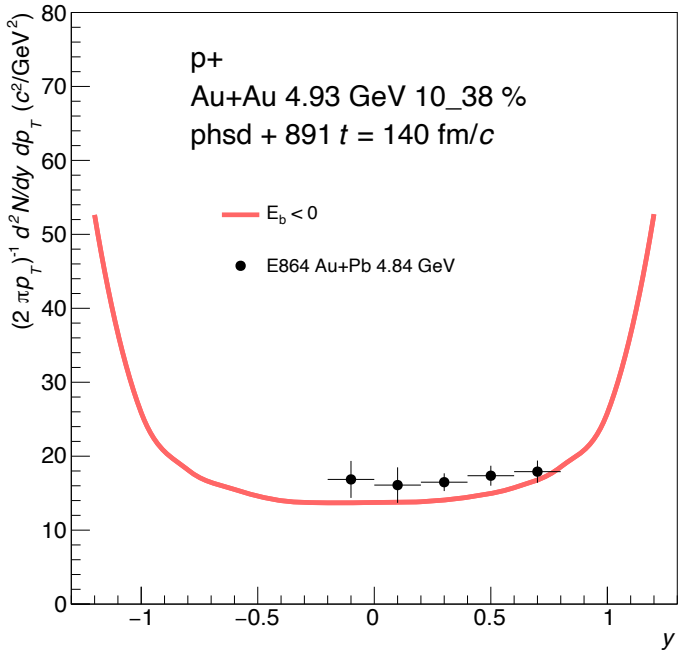
PHQMD Au-Au 4.93 GeV 0-10 % most central compared to experimental data

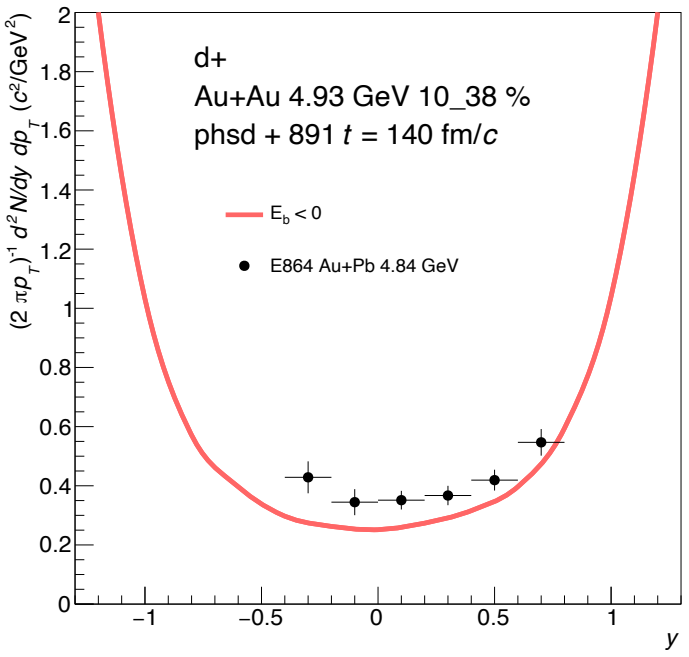


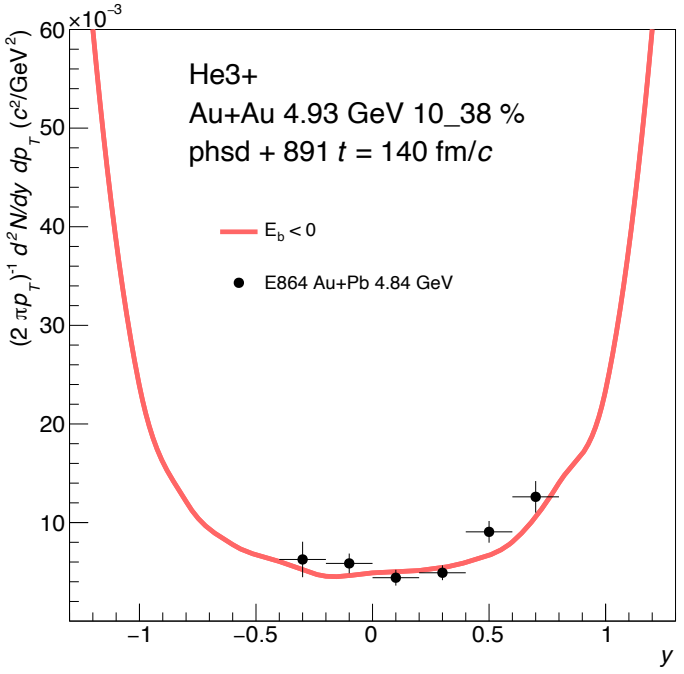




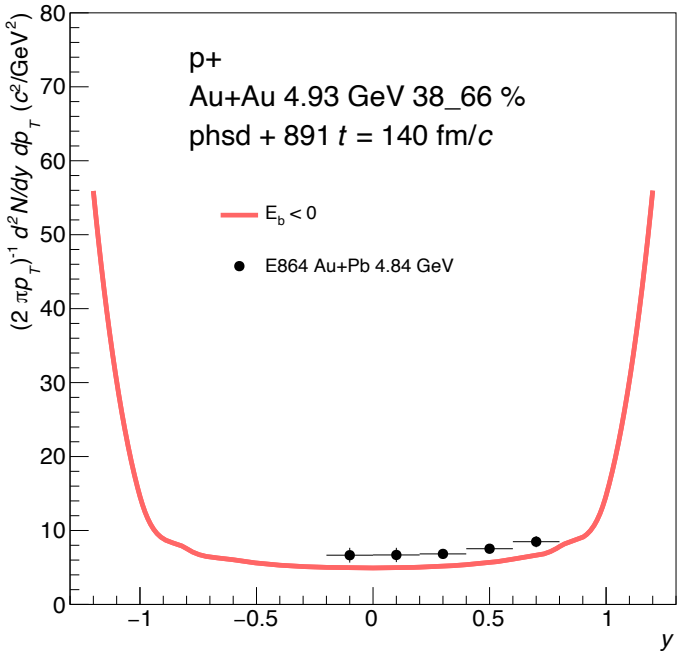
PHQMD Au-Au 4.93 GeV 10-38 % most central compared to experimental data

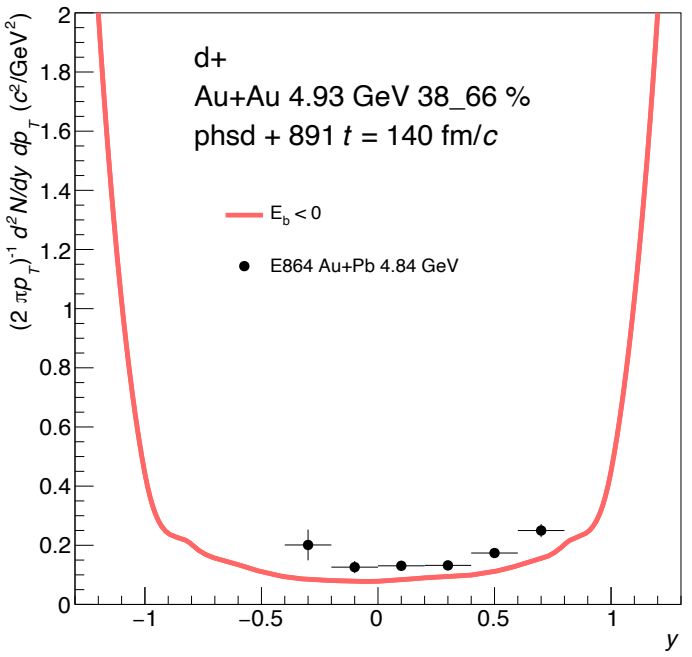


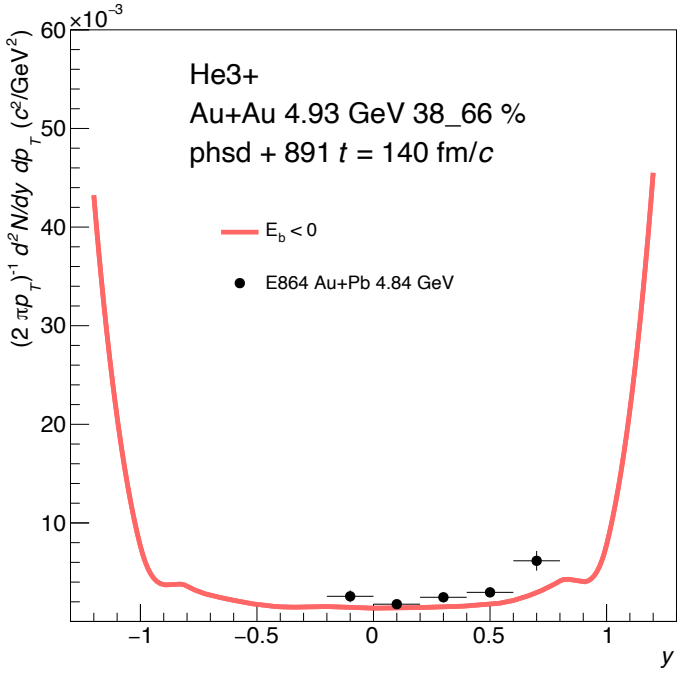




PHQMD Au-Au 4.93 GeV 38-66 % most central compared to experimental data







Run PHQMD, stabilisation & conversion at the batchfarm

All 3 steps* can be performed with:

. runPHQMD.sh

Only *runinfo.sh* needs to be modified.

The following files are needed (besides the PHQMD code):

- runinfo.sh	modification of specifications for run
- runPHQMD.sh	starts run on batch farm
- batch_runPHQMD.sh	executed on batch farm
- 791to891.exe	stabilisation routine
- convert_phqmd_detector_unigen.C	conversion PHQMD-output files into one root-file
- cluster_table.dat	contains information for definition of nuclei

runinfo.sh

Only runinfo.sh needs to be modified:

- selection of steps (run PHQMD code, stabilisation, conversion)
- choose option for cluster with A > 7 (all or only phyisical clusters are counted)
- choose if only clusters OR clusters & anti-clusters are converted
- select input information for PHQMD, e.g system, energy, number of parallel events, impactparamter etc.
- location information

runinfo.sh – run options, locations

```
## Options for running PHQMD code, stabilisation & conversion #

## Steps:
export RunPhqmdCode=1           ## = 1: run PHQMD code - input: inputPHSD, output: phsd.dat (hadrons), fort.791 (cluster-baryons), fort.891 (anti-
cluster-baryons)
export RunStabilisation=1       ## = 1: run stabilisation routine - input: fort.791 & fort.781, output: fort.891 & fort.881
export MakeDetectorInput=1      ## = 1: run conversion of PHQMD output into detector input - input: fort.891 & fort.881, output: unigen format

## Options for steps:
export ConvertAntiClusters=1    ## = 0: convert/stablise only clusters, = 1: convert clusters and anti-clusters
export CountAllClusters=0       ## = 0: only physical clusters are counted (unphysical clusters are counted as single baronys),
                                ## = 1: all clusters A > 7 are counted as clusters


## Delete & Merge
export DeleteFiles=0            ## = 1: deletes all output files, but the detector-input [dataset].phqmd.root
export RunHadd=0                ## = 1: add 100 jobs into one file
export RunHaddSim=0             ## = 1: add 10 jobs in one file for detector simulations



#############  Directories and files ##########################

export LOCATION=/lustre
export DIR=$LOCATION/cbm/users/$USER
export PHQMDDIR=$DIR/phqmd
export CBMROOT_DIR=$DIR/cbmsoft/cbmroot/build
export OUTDIR=$DIR/mc/phqmd/$SYSTEM"_"$ENERGY"GeV_"$IMPACTPARAMETER_MAX"fm_Num"$NUM"xSub"$ISUBS
```

runinfo.sh – input parameter, sbatch parameter

```
## Parameters for sbatch ######################################

export firstJob=1
export lastJob=2
export array=$firstJob-$lastJob
export last_hadd=$(printf "%0.0f" "$(echo " $lastJob / 100  " | bc -l)")
export array_hadd=1-$last_hadd
export last_hadd_simcbm=$(printf "%0.0f" "$(echo " $lastJob / 10  " | bc -l)")
export array_hadd_simcbm=$firstJob-$last_hadd_simcbm
export time="1-23:59:59"
export partition="long"


## Parameters for inputPHSD (more parameters availabe, but should usually not be changed)

export SYSTEM=auau              ## specification below for auau, pbpb, aupb, aupt - other systems need to be added below
export ENERGY=4.93              ## center of mass energy - !for sqrt(s) > 7 GeV setting for kinetic deuterons in PHQMD code should be changed!
export ELAB=$(echo " $ENERGY * $ENERGY / (2 * 0.938) - (2 * 0.938)" | bc -l)  ##  Lab energy per nucleon (needed for PHQMD)
export NUM=100                  ## number of parallel events (NUM=100 for 3 GeV & 4.9 GeV required)
export ISUBS=1                  ## number of subsequent runs
export nEvents=$(echo " $NUM * $ISUBS " | bc -l)

export IMPACTPARAMETER_MIN=0.0  ## minimal impact parameter in fm
export IMPACTPARAMETER_MAX=5.0  ## maximum impact parameter in fm ! Max impact parameter must not be > 15.0 fm !

export IGLUE=1                  ## =1 with partonic QGP phase (PHSD mode); =0 - HSD mode ! Needs to be set to = 0 for 3 GeV
export EOS=0                    ## EoS; =0: hard EOS without M.D.I; =1 soft EoS
export ICLUSTER=1               ## enable or disable CLUSTER output
```

PHQMD settings- important remarks

Energy:	needs to be given in units of lab energy per nucleon
Impact parameter:	maximum must be set b < 15.0 fm
NUM:	number of parallel events (needed for mean-field in PHSD)
	must be: NUM = 100 for 3.0 & 4.9 GeV, NUM = 200 for lower energies
IGLUE	: =1 with partonic QGP phase (PHSD mode); =0 - HSD mode
	must be set to = 0 for 3 GeV

Please be aware: It is possible that jobs do not finish due to time limit.

For higher energies $\sqrt{s} >$ 7 GeV: Setting for kinetic deuterons needs to be changed in the PHMQD code.

Options

⁎ Store PHQMD output files into root-format *EventPHQMD*

https://github.com/SusanneGlaessel/EventPHQMD

⁎ PHQMD with freeze-out time, position & momentum (avaible for hadrons):

change in PHQMD code & recompile: IFreezeOut=1 (in ini_param.f)

calculation of freeze-out time, position & momentum for clusters:

- freeze-out time of cluster: freeze-out-time of cluster-baryon with latest freeze-out-time
- freeze-out position & momentum: transport all cluster-baryons to cluster freeze-out-time

!!! with QMD (attractive potential) freeze-out-momentum ≠ final momentum
→ Format: *EventFreeze* to save freeze-out-momentum!!!

⁎ Write output from phsd.dat only (single hadrons)