# Development of a New Model Framework for Simulating Heavy Ion Collisions

Joelle Pospisil
Advisors: Dong Jo Kim, Maxim Virta
August 8 2025

Abstract

This document describes the development of a new model framework for simulating heavy ion collisions, called the Jyväskylä event generator. The Jyväskylä event generator is designed to be used for conducting a Bayesian analysis. The framework includes multiple heavy ion collision models which simulate the initial conditions of the collision, the viscous hydrodynamics of the quark gluon plasma, hadronization, and the evolution of the resulting hadron gas. In this document a comparison of different heavy ion collision models will be made and then key steps in the process of developing the new event generator will be outlined. With the Jyväskylä event generator new Bayesian analyses can be performed and can be compared to prior Bayesian analyses in order to reduce the model dependence of the estimated Bayesian parameters.

# Contents



# 1 Introduction

The goal of the ALICE experiment at CERN is to understand Quantum Chromodynamics (QCD) through the production of ultra-relativistic heavy ion collisions. When heavy ions collide at ultra-relativistic energies the confined quarks and gluons inside the ions are released and form a medium called Quark Gluon Plasma (QGP). QGP resides in a strongly coupled regime where both perturbative and non-perturbative techniques are limited. While these theoretical tools cannot fully describe the complexity of QGP, multi-stage heavy ion collision models combined with experimental measurements have proven fairly effective. The stages of these models typically include pre-equilibrium, viscous hydrodynamics, hadronization, and evolution of the resulting hadron gas.

In most multi-stage heavy ion collision models there are 10-20 free parameters of which only a few can be estimated theoretically, leaving the remaining parameters to be extracted from experimental observations such as the particle yields, mean transverse momentum, and more. Determining model parameters from experimental observables is challenging since they are often linked through complex relationships. A Bayesian analysis is a technique which is currently being employed to determine these model parameters from experimental observables provided by ALICE. In a Bayesian analysis, initial information about the parameters is encoded into prior distributions $P(x)$. Using the prior distributions, simulations of heavy-ion collisions are performed. The likelihood $P(y|x)$ is then found by comparing the simulated observables to experimental data. Lastly, the posterior distributions $P(x|y)$ of the model parameters are calculated using Bayes' Theorem such that:

$$P(x|y) \propto P(x|y) \cdot P(x) \tag{1}$$

For more information about Bayesian analyses and parameter estimation with regards to determining the properties of QGP, see [1]. To perform a Bayesian analysis an event generator is needed which links together the models that simulate the different stages of a heavy ion collision. In previous Bayesian analyses the Duke event generator was implemented [1] [2]. The Duke event generator used the T$_R$ENTo and FreeStream models to simulate the initial conditions, OSU-hydro to simulate the hydrodynamics of the QGP, Frzout to model the hadronization process, and UrQMD to model the scattering and evolution of the hadrons. However, it is crucial to perform multiple Bayesian analyses with different simulation models in order to remove model dependence when determining the free parameters of heavy ion collisions. This document will detail the development of a new framework, the Jyväskylä event generator, which will function much like the Duke event generator, but with a different collection of heavy ion collision models. The Jyväskylä event generator will use IP-Glasma to simulate the initial conditions, MUSIC to simulate the hydrodynamics of the QGP, iSS to model the hadronization process, and UrQMD to model the scattering and evolution of the hadrons. First a discussion will be made explaining the reasons behind why each model was chosen for the framework followed by a discussion of the key details in designing the Jyväskylä event generator.

# 2 Heavy Ion Collision Models

The main stages of a heavy ion collision are the pre-equilibrium dynamics, viscous hydrodynamics, hadronization, and evolution of the hadron gas. The pre-equilibrium dynamics stage models the initial collision of the two nuclei and their evolution until the point at which the QGP forms. In the Duke event generator pre-equilibrium conditions were simulated by T$_R$ENTo and freestream. These models will be replaced with IP-Glasma in the Jyväskylä event generator. Following the pre-equilibrium stage, the viscous hydrodynamics stage models the evolution of the QGP. The viscous hydrodynamics are modeled by OSU-hydro in the Duke event generator and by MUSIC in the Jyväskylä event generator. Hadronization is the process by which the QGP transitions into a hadron gas. The Duke event generator uses the model Frzout for this process while the Jyväskylä event generator uses a model called iSS. The evolution of the hadron gas handles the transport and scattering of hadrons until the time at which the particles are detected. Both event generators use UrQMD to model this final stage. Figure 1 illustrates the differing model structures of the two event generators.

## 2.1 T$_R$ENTo vs. IP-Glasma

Initial condition models such as T$_R$ENTo and IP-Glasma simulate the pre-equilibrium dynamics of heavy ion collisions by generating either energy or entropy profiles based upon the initial conditions given by the user such as the type of nuclei involved in the collision, their impact parameter, and other physical values. These energy profiles are then passed on to the viscous hydrodynamics model for simulating the QGP's fluid dynamics. T$_R$ENTo is a non-dynamic model which means that instead of explicitly simulating the pre-equilibrium evolution of the collision T$_R$ENTo generates static energy or entropy profiles. To create

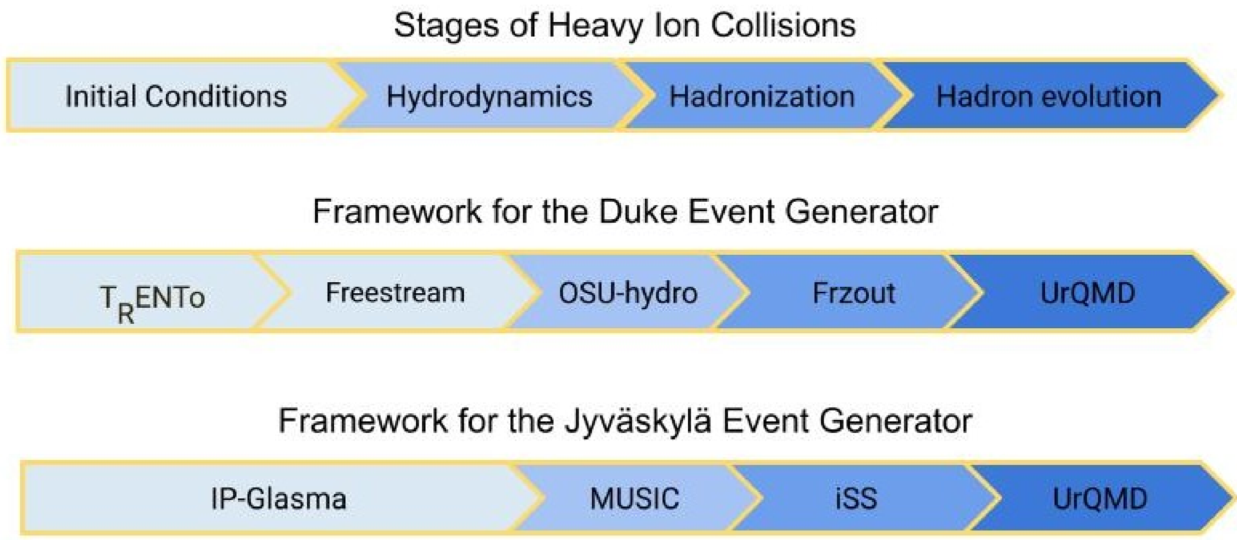


Figure 1: A visual comparison of the models chosen for the Duke event generator and the Jyväskylä event generator along with the main stages of a heavy ion collision. Each models is color coded according to the stage that it simulates.

these profiles T$_R$ENTo deposits energy based upon the reduced thickness function of the two nuclei. For more information on how to calculate the reduced thickness function and about T$_R$ENTo itself, see [3].

In contrast to T$_R$ENTo, IP-Glasma is a dynamic model which means that it does explicitly simulate the initial state and pre-equilibrium evolution of the collision. IP-Glasma uses weakly coupled color-glass condensate effective field theory and classical Yang-Mills evolution to calculate the evolution of the system. For more information about IP-Glasma, see [4]. IP-Glasma was chosen for the Jyväskylä event generator because it uses completely different techniques to simulate the pre-equilibrium dynamics compared to T$_R$ENTo. By performing a Bayesian analysis with a dynamic model such as IP-Glasma and then comparing the results to Bayesian analyses performed with a non-dynamic model such as T$_R$ENTo, the resulting parameter predictions will be more robust to model dependencies.

## 2.2 OSU-hydro vs. MUSIC

Hydrodynamic models such as OSU-hydro and MUSIC simulate the viscous hydrodynamics of the QGP after it has formed. This simulation is performed by numerically solving the five hydrodynamic conservation equations until the freeze-out time at which the QGP hadronizes. In order to simplify the calculations, OSU-hydro uses a boost invariant approximation which eliminates effects in the longitudinal direction. Therefore, OSU-hydro is considered to be a 2+1D model. In contrast, MUSIC is a 3+1D model as it does include hydrodynamic effects in the longitudinal direction. Including the hydrodynamic effects in the longitudinal direction increases both the model's complexity and computational cost, but is considered to be far more accurate as reality itself is a 3+1D system. MUSIC also includes additional viscosity terms that improve upon the OSU-hydro model. For more information about OSU-hydro and MUSIC see [5] and [6] respectively. These additional longitudinal effects and viscosity terms make MUSIC substantially different from OSU-hydro and will help to reduce model dependencies when combined with the results of Bayesian analyses performed with the Duke event generator.

### 2.3 Frzout vs. iSS

iSS and Frzout handle the hadronization process by generating individual particle samples from the final QGP hypersurface produced by MUSIC at the freeze-out time. This particlization process is performed using the Cooper-Frye formula which takes a continuous liquid-like object and mathematically segments it into particles with mass. For more information about the Cooper-Frye formula and how it is implemented in algorithms such as iSS, see [5]. Frzout and iSS have nearly identical functionalities and use the same physical assumptions. The reason iSS was chosen instead of Frzout for the Jyväskylä event generator is that iSS was designed to be used in tandem with MUSIC whereas Frzout was built for OSU-hydro. In order to use Frzout with MUSIC an additional step would have to be taken to convert the format of the hypersurface data file produced by MUSIC into a format that could be interpreted by Frzout. Converting the format of the hypersurface data so that it could be read by Frzout would consume unnecessary time and computing power when iSS can be used instead without any significant loss of functionality. Therefore, iSS was chosen for the Jyväskylä event generator.

### 2.4 UrQMD

UrQMD is a widely used model for simulating the evolution of hadron gases. UrQMD performs sequences of binary collisions and resonance decays in order to simulate the system's evolution and solve the Boltzmann equation for the distribution function of all hadrons in the system. Particle collisions are determined by comparing the minimum distance between the particles and the cross section of the collision type while resonance decays are determined using the concept of detailed balance. Further information about the calculations behind the simulation of particle collisions and about the concept of detailed balance would require a lengthy discussion which is beyond the scope of this document. For more information abut how UrQMD simulates the evolution of a hadron gas, see [7]. UrQMD was chosen for the Jyväskylä event generator since it is currently considered one of the best models for simulating the evolution of hadrons.

## 3 Developing the Event Generator

### 3.1 Creating Input Files

Between IP-Glasma and MUSIC there are over a hundred parameters that must be set by the user. Most of those parameters are static settings that can be set to be either on or off. Only a handful of the parameters vary the actual physics of a simulation and are considered useful for a Bayesian analysis. As a result, an input creator script is needed which can take a small number of dynamic inputs from the user and generate two complete input files, one for IP-Glasma and one for MUSIC. The input creator script included within the Jyväskylä event generator takes a text file as input. The text file must be formatted such that the IP-Glasma input parameters are listed first, with one parameter per line, followed by the statement, "EndOfIpglasmaParameters". Next, the MUSIC parameters should be listed. The entire file should end with the statement "EndOfMusicParameters".

When listing parameters and their values, each line should start with a parameter's name followed by a space or tab and then its value. Lists of all possible parameters can be found in the code repositories for IP-Glasma and MUSIC. An example of a valid input file can be found within the Jyväskylä event generator repository. To generate the IP-Glasma and MUSIC input files the script first starts with a while loop in which every line of text is written to the IP-Glasma input file. This while loop is broken when the phrase "EndOfIpglasmaParameters" is read. Next, every other parameter needed for IP-Glasma, along with its default value is added to the IP-Glasma input file.

Writing parameters to MUSIC is more complicated as MUSIC requires each parameter to be listed in a specific order. To accomplish this ordering, first the script reads a line from the user's input file. It then iterates over a list of dynamic parameter names until the parameter name stored in the list matches the parameter name read in the user's input file. The current list index is then updated to include not only the parameter's name, but also the parameter's value found within the user's input file. This process is repeated until the statement "EndOfMusicParameters" is reached. After this, each static parameter required for MUSIC and its corresponding value is written to the MUSIC input file along with the dynamic parameters stored in the aforementioned list. This process ensures that all dynamic parameters required for MUSIC are listed in the MUSIC input file in the correct order no matter how they are listed in the user's input file.

## 3.2 Adding Nuclear Fluctuations to IP-Glasma

The density distribution of a heavy ion is determined by Eq. (2) where $R(\theta, \phi)$ is given by Eq. (3).

$$\rho(r, \theta, \phi) = \frac{\rho_0}{1 + e^{(r - R(\theta,\phi))/a_0}} \tag{2}$$

$$R(\theta, \phi) = R_0 \left( 1 + \beta_2 \left[\cos\gamma Y_{2,0} + \sin\gamma Y_{2,2}\right] + \beta_3 \sum_{m=-3}^{3} \alpha_{3,m} Y_{3,m} + \beta_4 \sum_{m=-4}^{4} \alpha_{4,m} Y_{4,m} \right) \tag{3}$$

In the equation for $R(\theta, \phi)$, $\beta_2$ is the quadrupole term, $\gamma$ is the triaxial term, $\beta_3$ is the octuple term and $\beta_4$ is the hexadecapole term. These parameters are known collectively as the nuclear structure parameters since they determine the shape of the heavy ion's nucleus. Figure 2 provides a visual representation of how each term affects the nucleus's structure. In IP-Glasma the user can either include values for these parameters in the input file or IP-Glasma will set default parameters based upon the types of ions being used in the collision.

However, setting a static value for $\beta_2$ and $\gamma$, as is done within IP-Glasma, is not physically accurate as these two parameters fluctuate. Several modifications were made to IP-Glasma in order to account for these fluctuations. First of all, two new parameters were added to IP-Glasma, namely beta2_std and gamma_std. If beta2_std is set to be less than or equal to zero, $\beta_2$ will be set as a static parameter as defined by the user.

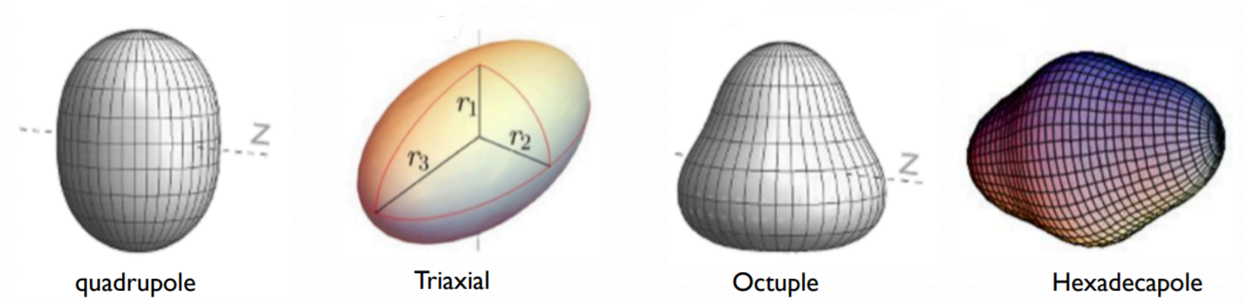


Figure 2: Illustration of the shape of different nuclear structures as determined by the nuclear parameters. Image taken from Zhou's 2025 Quark Matter presentation [8]

If beta2_std is greater than zero then the parameter $\beta_2$ will represent the mean value for $\beta_2$ and beta2_std will represent the standard deviation of $\beta_2$. A normal Gaussian distribution will then be generated using the mean and standard deviation. From that distribution a randomly generated value for $\beta_2$ will be selected, replicating the fluctuations of $\beta_2$. This same exact process is also applied to $\gamma$ where if gamma_std is greater than zero, a Gaussian distribution will be created with $\gamma$ as the mean and gamma_std as the standard deviation of $\gamma$. $\gamma$ will then be assigned a randomly generated value from the Gaussian distribution.

### 3.3 Linking iSS to UrQMD

IP-Glasma, MUSIC, and iSS were all naturally built to be compatible. MUSIC and iSS were designed such that the output files from IP-Glasma could be passed as input files to MUSIC, and that the output files from MUSIC could be passed as input files to iSS. However, this is not the case for iSS and UrQMD. iSS outputs a binary data file which stores variables such as the particle's PDG code, the particle's momentum and energy, and the particle's position in space and time. UrQMD expects a binary data file which contains the particle's position in space and time, momentum, mass, UrQMD I.D., isospin, charge, and resonant decay time. Therefore, in order to pass the output of iSS to UrQMD a conversion script is needed.

To perform this task, first the conversion script reads the iSS output file and determines if the current line is either a header line or a line containing particle data. If the line is a header then the event number and number of particles within that event are saved as variables. A header which matches the format expected by UrQMD and which includes the event number and number of particles is then written to the UrQMD input file. If the current line contains particle data, the script first stores the particle's position, momentum, energy, and PDG code as variables. The particle's mass is calculated from its momentum and energy as $mc^2 = \sqrt{E^2 - p_x^2 - p_y^2 - p_z^2}$.

The UrQMD I.D. functions similarly to a PDG code except that particles of the same type and different charges are given the same UrQMD I.D. For example, $\pi^0$ and $\pi^+$ have different PDG codes but share the same UrQMD I.D. To find the UrQMD I.D. and the particle's isospin, the iSS to UrQMD converter contains a large array consisting of three columns. The first column lists the particle's UrQMD I.D., the second column lists the particle's isospin, and the third column lists the particle's PDG code. The absolute value of the particle's PDG code, taken from the iSS output file, is matched to a PDG code in the converter script's array and the corresponding isospin and UrQMD I.D. are saved as variables. If the particle is an anti-particle then a negative sign is added to both the isospin and UrQMD I.D. The UrQMD I.D. and isospin are then used to find the particle's charge from an array listing charge values. To find the decay time, the particle's width is needed. This value is found by using the particle's UrQMD I.D. as an index for an array of particle widths stored in the converter script. With these values the decay time constant tau is calculated such that $\tau \propto -\ln(1-r)/w$ where $w$ is the particle's width and $r$ is a randomly generated number between 0 and 1. Once the mass, UrQMD I.D., isospin, charge, and decay time have all been found, these values along with the particle's position and momentum are written to the UrQMD input file. This process then continues for every particle listed in the iSS output file.

## 3.4 Creating an Output File for Analyses

The default output file from UrQMD version 3.4 is a binary data file which contains information such as the particle's I.D., momentum, energy, mass, and freeze-out position coordinates. In the Duke event generator UrQMD had been altered to instead output a binary data file containing the particle's PDG code, charge, transverse momentum, transverse energy, transverse mass, azimuthal angle of the transverse momentum, rapidity, and pseudorapidity. This data file was then converted into an HDF5 file within the Duke event generator script. An HDF5 file is a container which can store multiple heterogeneous data objects, or datasets, enabling it to efficiently store and organize large quantities of data. The analysis script needed to perform a Bayesian analysis was originally written to be compatible with the HDF5 file created by the Duke event generator. For the purpose of running an analysis with the Jyväskylä event generator, alterations needed to be made to both UrQMD and the event generator script so that an HDF5 file could be produced in the format expected by the analysis script.

To first change the information stored in the binary data file, a new output file format was added to UrQMD. Note that UrQMD is written in Fortran. The new output file format was copied from the Duke event generator's version of UrQMD and involved the steps necessary to calculate the variables required by the analysis script. It then saved those variables to a binary file. Within the Jyväskylä event generator script there is a function which is designed to then convert the binary file into an HDF5 file. It does this by first filtering out all of the UrQMD header lines. Next, each variable in the binary data file is linked to a header name for that variable. Finally, the header names and their values for each particle are used to generate a dataset for the HDF5 file. At this point the HDF5 file is ready to be used as an input for the analysis script.

# 4 Conclusions

Heavy ion collisions are complex processes which require a different model to simulate each stage of the process from the pre-equilibrium stage wherein the two nuclei collide, to the viscous hydrodynamics of the QGP, to the evolution of the resulting hadron gas. There are many different types of models for each stage of the process, some based upon simplified estimations and others based upon complex calculations of QCD theory. The Jyväskylä event generator is a new framework whose main goal is to produce simulations using a new combination of heavy ion collision models for the purpose of running a Bayesian analysis. This Bayesian analysis will then be compared to previous analyses which use different models in order to remove model dependence from the estimation of the free parameters which characterize QGP formation. Features of the Jyväskylä event generator include a script which creates input files for IP-Glasma and MUSIC from a small set of dynamic inputs, the inclusion of nuclear fluctuations within IP-Glasma, a conversion script which goes directly from iSS to URQMD, and additional formatting such that the output of UrQMD can be used directly within Bayesian analysis scripts.

# References


[1] J. E. Parkkila, A. Onnerstad, S.F. Taghavi, C. Mordasini, A. Bilandzic, M. Virta, and D.J. Kim. New constraints for QCD matter from improved bayesian parameter estimation in heavy-ion collisions at LHC. *Physics Letters B*, 835, 2022.

[2] M. Virta, J. Parkkila, and D.J. Kim. Enhancing bayesian parameter estimation by adapting to multiple energy scales in rhic and lhc heavy-ion collisions. *Physical Review C*, 111, 2025.

[3] J.S. Moreland, J.E. Bernhard, and S.A. Bass. Alternative ansatz to wounded nucleon and binary collision scaling in high-energy nuclear collisions. *Physical Review C*, 92, 2015.

[4] B. Schenke, P. Tribedy, and R. Venugopalan. Fluctuating Glasma initial conditions and flow in heavy ion collisions. *Physical Review Letters*, 108, 2012.

[5] C. Shen, Z. Qui, H. Song, J. Bernhard, S. Bass, and U. Heinz. The IEBE-vishnu code package for relativistic heavy-ion collisions. *Computer Physics Communications*, 199, 2016.

[6] B. Schenke, S. Jeon, and C. Gale. (3+1)D hydrodynamic simulation of relativistic heavy-ion collisions. *Physical Review C*, 82, 2010.

[7] 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, Ch. Hartnack, J. Aichelin, and N. Amelin. Microscopic models for ultrarelativsitc heavy ion collisions. *Progress in Particles and Nuclear Physics*, 41, 1998.

[8] Y. Zhou. Pobing nuclear structures in heavy-ion collisions from small to large systems. In *Qaurk Matter 2025*, 2025.