NUKLEONIKA 2004;49(Supplement 2):S81–S83

PROCEEDINGS

CorrFit – a program to fit arbitrary two-particle correlation functions

Adam Kisiel

Abstract Two-particle correlation functions produced in heavy-ion collisions with small relative velocities provide unique space-time information about the system. In order to extract this information, a fitting procedure must be applied. In this work we describe the software main purpose of which is to provide a way to fit arbitrary two-particle correlation functions, using a complete knowledge of the CorrFit corresponding interaction.

Key words correlation function fitting • weight method

A. Kisiel
Faculty of Physics,
Warsaw University of Technology,
75 Koszykowa Str., 00-662 Warsaw, Poland,
Tel.: +48 22/ 660 73 75, Fax: +48 22/ 628 21 71,
e-mail: kisiel@if.pw.edu.pl

Received: 22 April 2004

Motivation

The CorrFit program has been originally created in order to fit non-identical particle (pion-kaon) correlation functions obtained in the STAR experiment [1]. As there is no analytical form of the correlation function for pion-kaon pair, it was necessary to develop a program, which would use the full information of the interaction potential (both Coulomb and strong interaction, as well as quantum statistics for identical pairs) and an arbitrary source model. The best-fit correlation function is found through the minimum χ^2 method.

Design specifications

The following requirements have been used when designing the CorrFit software:

- Use the knowledge of particle interaction in the form of weight calculators. Possibility to choose between multiple calculators.
- Use of arbitrary source models, describing the emitting source. Each source model needs to be parametrized. Possibility to use models with any number of parameters. Easy addition of new source models by the end user.
- Use of arbitrary correlation function types (1-dimensional, 3-dimensional, double-ratios). Possibility to add new correlation function types.
- Only data normally available in the experiment should be used in the fitting procedure – that is the correlation functions and the pair momentum distribution.
- Program execution is controlled by the parameter file, which is automatically generated. The end user only needs to fill the data relevant to his analysis.

The program would be distributed in the source form, available for use and modification by anyone.

Implementation

The program was designed to be fully object-oriented. It is written in C++, with the use of classes and file formats from the ROOT package [2].

The program starts with reading the parameter file. If it does not exists, it is automatically created for the user, who then needs to supply the necessary values. Then, the input data is read – the experimental correlations functions (ECF) and the pair momenta. Also the source model (SM) and the range of its parameters is determined. An empty χ^2 map (C2M) is then created, with model parameters on the axes. Then, the program progresses through all of the C2M bins sequentially. In each of them, the following actions are carried out. First, the SM parameters for the current bin are read out from the axes of the C2M. For each pair, the momentum is read from the input file, the position is generated by the SM, according to the current parameters, and a wave-function squared (called 'weight') for such a pair is calculated. The Calculated Correlation Function (CCF) is constructed using these weights. At the end, a χ^2 value for a given bin, which quantifies the difference between the ECF and the CCF for this bin, is calculated. This value is then stored in the C2M bin. When the whole map is filled in this way, the minimum of the map is sought, and the values of the parameters, which produced the CCF with lowest χ^2 value are taken as the best fit.

In the current version of the program two additional improvements were made. If the number of input pairs is insufficient for the meaningful calculation, it is possible to generate additional pairs. Also, after the first coarse map is done, a second map, smaller but also with smaller bin width, is filled to get the best-fit parameters with better accuracy.

Main classes

The main building classes of the CorrFit program are shown in Fig. 1. The CFFitter class manages the whole fitting process. It initiates the procedure of pair reading (through PairManager class), manages the C2M filling (through Chi2Map class) and has the knowledge of the ECFs and CCFs used in current analysis. The CFFitter is an abstract base class, with the actual fitter classes inheriting from it. In principle each type of correlation function class needs its own fitter. At the time of this writing, fitters were written for:

1. CFFitter1DHBT – 1D identical particle correlation functions.
2. CFFitterNonId – 1D non-identical correlation functions with double ratios.
3. CFFitterNonIdMult – several 1D non-identical correlation functions with double ratios fitted at the same time with the same model.

Each of this fitters can have (and usually has) a separate ECF and CCF class.

The ECF class ExpCF handles reading and storing of the experimental correlation function. It also takes care of momentum resolution and purity corrections. ExpCF is an abstract base class, each CF type requires its own ECF class.

The CCF class CalcCF handles calculation of the theoretical correlation functions. It manages the weight generator, which calculates the weights for pairs. It is also possible (through CFStorage class) to store and retrieve previously calculated CFs with the specific SM and SM parameter values. CalcCF is also an abstract base class; each CF type requires its own CCF class.

The weight calculator (WF) takes as an input pair momenta (p_1,p_2) and freeze-out coordinates (x_1,x_2), as well as particle types, then calculates the squared wave-function (a weight) for such pair. At the time of this writing, two WFs were implemented – one based on Fortran code by R. Lednický, the other based on the code provided by S. Pratt.

The SM class SourceModel is responsible for providing the freeze-out coordinates (x_1,x_2) for the pair, given its momenta (p_1,p_2). The SourceModel is an abstract base class, the real models should be built on top of it. The SM can have any form – from a simple Gaussian distribution to the parameterization of complex physical models. This class is the most important from the physical point of view, as it includes the description of the emitting source necessary for the interpretation of the fit results. New SM classes can be trivially added by the end user.

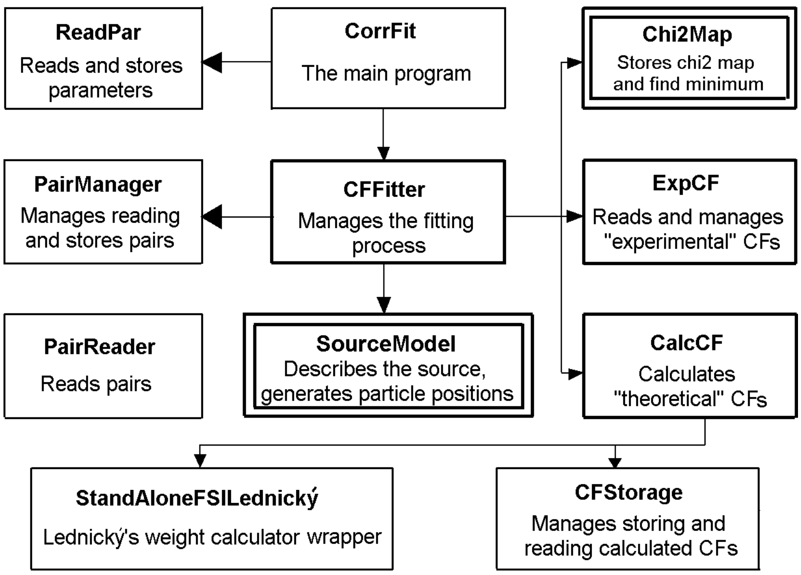

Fig. 1. Functional diagram of the CorrFit classes.

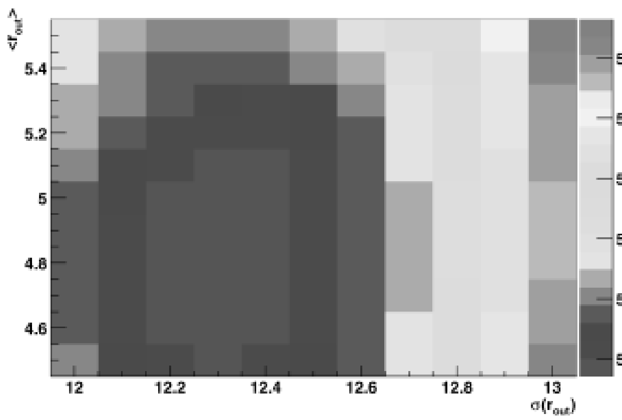

Fig. 2. An example of a χ^2 map produced by CorrFit.

Program output – examples

The CorrFit program produces several results. It outputs the best fit parameters for the given fit run. It stores the C2Ms produced during its run. An example of such a map is shown in Fig. 2. The axes of the map are the parameters of the model (in this case a sigma and mean of a Gaussian in r_{out} variable). The value in each cell is the χ^2 value between the fitted ECF, and the CCF generated with a given SM parameter values. Such a map can be later used to judge the quality of the fit, as well as to determine the systematic error. The program also generates a best fit correlation function which can be directly compared to the experimental one. An example of such a comparison is shown in Fig. 3.

The output of the program can be easily extended. For example, for theoretical model studies, it might be desirable to compare the theoretical source distributions with the fitted ones. In this case, the CorrFit program can save the source distribution it generated through the SM class. An example of such comparison for RQMD study is shown in Fig. 4. This plot can be used to judge how well the assumed SM describes the actual source distribution.

Conclusions and plans

The CorrFit software is still in the development phase, although it is fully functional and can perform the basic tasks of fitting an arbitrary correlation function with an arbitrary source model. At the moment, it is the only method to obtain quantitative information from non-identical particles correlations.

In the future the author considers a few additions to the software:

- Developing a way to fit three-dimensional correlation functions;
- To develop other parameterization sources (mostly based on input from theory and models);

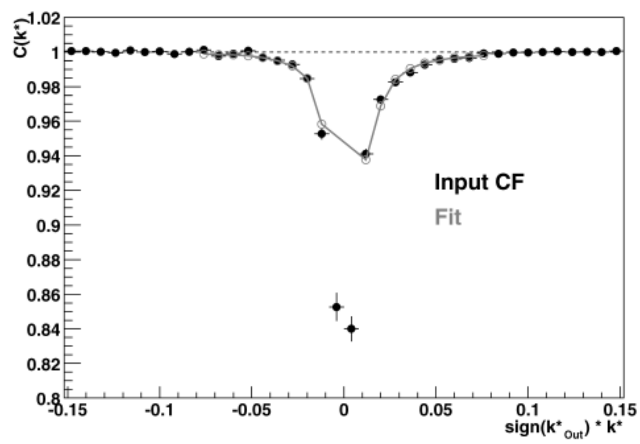

Fig. 3. Comparison of the experimental (input) and theoretical (fitted) correlation functions.

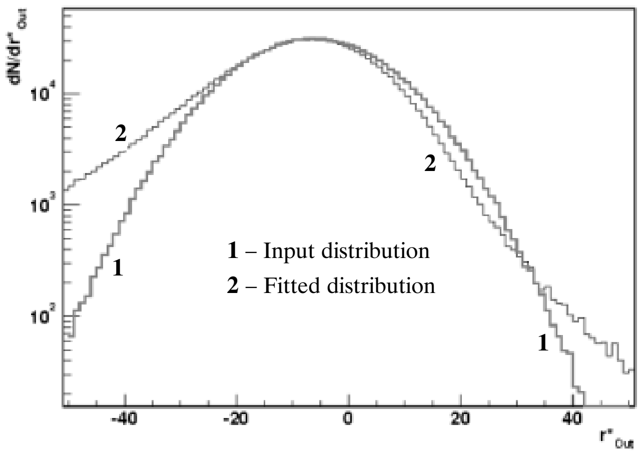

Fig. 4. A comparison of input and fitted source distributions.

- To implement a MINUIT minimalization method instead of a simple minimum determination;
- To implement and test new C2M map generation methods (pair-by-pair instead of bin-by-bin) to improve performance.

Acknowledgment This work was supported by the State Committee for Scientific Research in years 2003–2004 as a research project.

References

1. Adams J, Adler C, Aggarwal MM (2003) Pion-kaon correlations in central Au+Au collisions at sqrt{s_NN} = 130 GeV. Phys Rev Lett 91:262302
2. Brun R, Rademakers F (1997) ROOT – an object oriented data analysis framework. Nucl Instrum Meth Phys Res A 389:81–86 (see also: http://root.cern.ch)