

BCVEGPY2.2: A Newly Upgraded Version for Hadronic Production of the Meson B_c and Its Excited States

Chao-Hsi Chang*

*State Key Laboratory of Theoretical Physics, Institute of Theoretical Physics,
Chinese Academy of Sciences, P.O. Box 2735, Beijing 100080, P.R. China and
CCAST (World Laboratory), P.O. Box 8730, Beijing 100080, China*

Xian-You Wang†

*State Key Laboratory of Theoretical Physics, Institute of Theoretical Physics,
Chinese Academy of Sciences, Beijing 100080, P.R. China and
National Center for Nanoscience and Technology of China No.11,
Beiyitiaozhongguancun, Beijing 100190, P.R. China*

Xing-Gang Wu‡

*Department of Physics, Chongqing University, Chongqing 401331, P.R. China
(Dated: June 20, 2018)*

A newly upgraded version of the BCVEGPY, a generator for hadronic production of the meson B_c and its excited states, is available. In comparison with the previous one [C.H. Chang, J.X. Wang and X.G. Wu, Comput. Phys. Commun. **175**, 624 (2006)], the new version is to apply an improved hit-and-miss technology to generating the un-weighted events much more efficiently under various simulation environments. The codes for production of $2S$ -wave B_c states are also given here.

NEW VERSION PROGRAM SUMMARY

Title of program : BCVEGPY2.2

Program obtained from : CPC Program Library

Reference to original program : BCVEGPY2.1

Reference in CPC: Comput. Phys. Commun. **175**, 624 (2006)

Does the new version supersede the old program: Yes

Computer : Any LINUX based on PC with FORTRAN 77 or FORTRAN 90 and GNU C compiler as well

Operating systems : LINUX

Programming language used : FORTRAN 77/90

Memory required to execute with typical data : About 2.0 MB

No. of bytes in distributed program : About 2 MB, including PYTHIA6.4

Distribution format : .tar.gz

Nature of physical problem : Hadronic Production of B_c meson and its excited states.

Method of solution : To generate un-weighted events of B_c meson and its excited states by using an improved hit-and-miss technology.

Reasons for new version : Responding to the feedback from users, such as those from CMS and LHCb groups, we create a new hit-and-miss algorithm for generating the un-weighted events. Furthermore, the relevant codes for generating the $2S$ -excited state of B_c meson are added, because the excited state production may be sizable in the new LHC run.

Typical running time : It depends on which option is chosen to match PYTHIA when generating the full events and also on which state of B_c meson, either its ground state or its excited states, is to be generated. Typically on a 2.27GHz Intel Xeon E5520 processor machine, for producing the B_c meson ground state: I) If setting [IDWTUP=3 and `unwght=.true.`], it shall adopt the new hit-and-miss technology to generate the un-weighted events, and to generate 10^5 events takes 30 minutes; II) If setting [IDWTUP=3 and `unwght=.false.`] or [IDWTUP=1 and IGENERATE=0], it shall generate the weighted events, and to generate 10^5 events takes 2 minutes only (the fastest way, for theoretical purpose only); III) As a comparison, if setting [IDWTUP=1 and IGENERATE=1], it shall, as the same as the previous version, adopt the PYTHIA inner hit-and-miss technology to generate the un-weighted events, and to generate 1000 events takes about 22 hours. Thus, the efficiency (and accuracy also) for generating the un-weighted events obviously is greatly increased.

* zhangzx@itp.ac.cn

† xianyouwang@itp.ac.cn

‡ wuxg@cqu.edu.cn

Keywords : Event generator; Hadronic production; B_c meson; Un-weighted events

Summary of the changes : 1). We improve the approach for generating un-weighted events. 2). Responding to the feedback from users, we adjust part of the codes to make it work more user-friendly. More specifically, we explain main changes in the following :

- **Event generation.**

If each simulated event comes with a weight, it will make the data analysis much more complicated. Thus the un-weighted events are usually adopted for Monte Carlo simulations. As an external process of PYTHIA, the generator BCVEGPY [1–4] shall call the PYTHIA inner hit-and-miss mechanism to generate the un-weighted events by setting $IDWGTUP = 1$ and $IGENERATE = 1$ [5], i.e. the Von Neumann method is used for generating the un-weighted B_c events.

Every events bearing a weight ($xwgtup$) respectively, when inputting them to PYTHIA, they are suffered from being accepted or rejected, all the fully generated events at the output become to have a common weight. The Von Neumann method states that the event should be accepted by the PYTHIA subroutine PYEVNT with a probability $\mathcal{R} = xwgtup/xmaxup$. This can be achieved by comparing \mathcal{R} with a random number that is uniformly distributed within the region of $[0, 1]$. Namely if \mathcal{R} is bigger than such a random number then the event is accepted, otherwise it should be rejected. Here $xmaxup$ stands for the maximum event weight.

The von Neumann method works effectively for the cases when all the weights of input events are mod-

erate in the whole phase-space. However if the input events' weights vary greatly, such as varying logarithmically, then its efficiency shall be greatly depressed, since too much time shall be wasted for calculating $xwgtup$ of the rejected events. Thus it is helpful to find a new method for generating un-weighted events.

We will adopt the new hit-and-miss strategy suggested by Ref.[6] to do the B_c meson un-weight simulation. Extra switches for calling this new technology are added to BCVEGPY, e.g. the new hit-and-miss technology shall be called by setting $IDWTUP=3$ and $unwght=.true.$. Details for this new technology can be found in Ref.[6]. For self-consistency, we repeat its main idea here.

To be different from previous versions, BCVEGPY2.2 uses the VEGAS [7] and the MINT [8] as a combined way to generate the un-weighted events. The whole phase space shall be separated to a multi-dimensional phase-space grid. The main purpose of VEGAS [7] is to perform the adaptive Monte Carlo multi-dimensional integration, which uses the importance-sampling method to improve the integration efficiency. Each event shall generally result in a different weight, recorded by $xwgtup$, and the maximum weight within each grid shall be simultaneously recorded into the importance-sampling grid file (with the suffix `.grid`). Then following the idea of MINT, the Von Neumann method is used in each phase-space grid. Within this small grid region, the von Neumann algorithm works effectively, thus the efficiency for generating un-weighted events are greatly increased.

To implement the new hit-and-miss algorithm into BCVEGPY2.2, we change the original VEGAS subroutine as

```
vegas(fxn,ndim,ncall,itmx,nprn,xint,xmax,imode)
```

Three new variables $xint$, $xmax$ and $imode$ are added in the VEGAS subroutine. The $xmax$ array is used to record the maximum weights in all cells and $imode$ is a flag. $xint$ stands for the output cross-section when setting $imode=0$, which shall be used to initialize the $xmax$ array when setting $imode=1$. For convenience, the generated $xmax$ array will be stored in the same grid file in which the importance sampling function is stored.

In the initialization stage, the VEGAS subroutine shall be called by the subroutine `evntinit` twice by setting $imode=0$ and $imode=1$ respectively to generate both the upper bound grid $xmax$ for all cells and the importance sampling function.

A subroutine `gen(fxn,ndim,xmax,jmode)` is defined in the file `vegas.F` with the purpose to generate the un-weighted events. Three options for calling `gen` subroutine are defined: $jmode=0$ is to initialize the parameter; $jmode=3$ is to print the generation statistics; $jmode=1$ is the key option, which is to use the new hit-and-miss technology to generate the un-weighted events. More explicitly, by calling `gen(fxn,ndim,xmax,jmode=1)`, three steps shall be executed:

1. Call the `phase_gen` subroutine to generate a random phase-space point and to calculate its weight $xwgtup$.
2. Judge the point locates in which cell and read

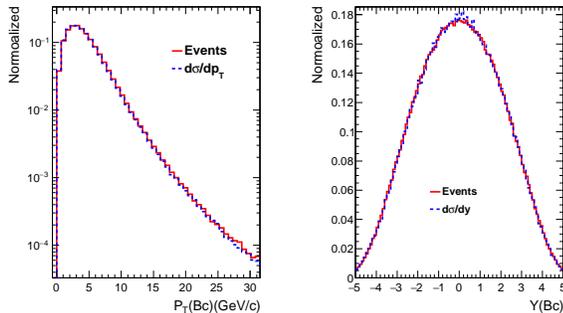


FIG. 1. Comparison of the normalized B_c transverse momentum (P_T) and rapidity (y) distributions derived by setting `unwght=.true.` (events) and `unwght=.false.` (differential distributions), which are represented by solid and dotted lines, respectively.

from the `xmax` array and get the upper bound value `xmaxup` for this particular cell.

- Judge whether such point be kept or not by using the Von Neumann method with the help of the probability `xwgtup/xmaxup`.

To be more flexible, we add one parameter `igenmode` for generating or using the existed `.grid` files. When setting `igenmode=1`, the VEGAS subroutine shall be called to generate the `.grid` files. When setting `igenmode=2`, the VEGAS subroutine shall be called to generate more accurate `.grid` files from the existed `.grid` files. When setting `igenmode=3`, one can directly use the existed `.grid` files to generate events without running VEGAS. Importantly, before using the existed `.grid` files, one must ensure all the parameters be the same as the previous generation.

- **A script for setting the parameters and a cross-check of the un-weighted events.**

We put an additional file, `bcvegpu_set_par.nam`, in the new version for setting the parameters. This way the user does not need to compile the program again if only the parameter values are changed.

As a cross-check of the new technology, we compare the un-weighted B_c event distributions derived by setting `unwght=.true.` with the weighted B_c differential distributions derived by setting

`unwght=.false.` The results are shown in FIG.1. Those two distributions after proper normalization agree well with each other, that shows our present scheme for un-weighted events is correct.

- **$B_c(2S)$ generation.**

In 2014 the ATLAS collaboration reported an observation about an excited state of B_c meson, which most probably is $B_c(2S)$ state [9]. With more data being collected at LHC detectors, it is hopeful that more observations on the excited B_c states will be issued. Therefore in addition to the production via color-singlet $B_c(1S)$, $B_c(1P)$ and color-octet $B_c(1S)$ states, the $B_c(2S)$ production is involved in BCVEGPY2.2. It is achieved by replacing the $1S$ -wave bound-state parameters `pmb`, `pmc` and `fbc` with those of the $2S$ -wave one. Here `pmb`, `pmc` and `fbc` are for b -quark mass, c -quark mass and the radial wave function at the zero ($|R(0)|$), respectively. For the $2S$ -wave case, their default values are set as `pmb=5.234` GeV, `pmc=1.633` GeV and `fbc=0.991` GeV^{3/2} [10] if the mass of the $2S$ -wave B_c state is 6.867 GeV.

More explicitly, two new values for `ibcstate` are added: `ibcstate=9` is to generate 2^1S_0 state and `ibcstate=10` is to generate 2^3S_1 state. Detailed technologies for deriving the production properties of all the mentioned ten B_c meson states can be found in Refs.[11–13]. Furthermore, the values for `mix_type` are rearranged. `mix_type=1` is to generate the mixing events for all mentioned states. `mix_type=2` is to generate the mixing events for 1^1S_0 and 1^3S_1 states. `mix_type=3` is to generate the mixing events for the four $1P$ -wave states and the two color-octet 1^1S_0 and 1^3S_1 states. `mix_type=4` is to generate the mixing events for 2^1S_0 and 2^3S_1 states.

Acknowledgments: This work was supported in part by National Basic Research Program of China (973 program, No.2013CB932804), by the Natural Science Foundation of China under Grant No.11275243, 11747001 and No.11275280, and by the Fundamental Research Funds for the Central Universities under the Grant No.CDJZR305513.

-
- [1] C. H. Chang, C. Drriouichi, P. Eerola and X. G. Wu, “BCVEGPY: An Event generator for hadronic production of the B_c meson,” *Comput. Phys. Commun.* **159**, 192 (2004).
 - [2] C. H. Chang, J. X. Wang and X. G. Wu, “BCVEGPY2.0: A Upgrade version of the generator BCVEGPY with an

addendum about hadroproduction of the P-wave B(c) states,” *Comput. Phys. Commun.* **174**, 241 (2006).

- [3] C. H. Chang, J. X. Wang and X. G. Wu, “An Upgraded version of the generator BCVEGPY2.0 for hadronic production of B(c)meson and its excited states,” *Comput. Phys. Commun.* **175**, 624 (2006).

- [4] X. Y. Wang and X. G. Wu, “A Trick to Improve the Efficiency of Generating Unweighted B_c Events from BCVEGPY,” *Comput. Phys. Commun.* **183**, 442 (2012).
- [5] T. Sjostrand, S. Mrenna and P. Z. Skands, “PYTHIA 6.4 Physics and Manual,” *JHEP* **0605**, 026 (2006).
- [6] X. Y. Wang and X. G. Wu, “GENXICC2.1: An Improved Version of GENXICC for Hadronic Production of Doubly Heavy Baryons,” *Comput. Phys. Commun.* **184**, 1070 (2013).
- [7] G. P. Lepage, “A New Algorithm for Adaptive Multidimensional Integration,” *J. Comput. Phys.* **27**, 192 (1978).
- [8] P. Nason, “MINT: A Computer program for adaptive Monte Carlo integration and generation of un-weighted distributions,” arXiv:0709.2085 [hep-ph].
- [9] G. Aad *et al.* [ATLAS Collaboration], “Observation of an Excited B_c^\pm Meson State with the ATLAS Detector,” *Phys. Rev. Lett.* **113**, 212004 (2014).
- [10] E. J. Eichten and C. Quigg, “Mesons with beauty and charm: Spectroscopy,” *Phys. Rev. D* **49**, 5845 (1994).
- [11] C. H. Chang and X. G. Wu, “Uncertainties in estimating hadronic production of the meson B_c and comparisons between TEVATRON and LHC,” *Eur. Phys. J. C* **38**, 267 (2004).
- [12] C. H. Chang, J. X. Wang and X. G. Wu, “Hadronic production of the P-wave excited B_c -states $B^*(cJ, L=1)$,” *Phys. Rev. D* **70**, 114019 (2004).
- [13] C. H. Chang, C. F. Qiao, J. X. Wang and X. G. Wu, “The Color-octet contributions to P-wave B_c meson hadroproduction,” *Phys. Rev. D* **71**, 074012 (2005).