**BtoVVana: the package for analysis of**

\[ B_s^0 \rightarrow J/\psi \phi \text{ and } B_d^0 \rightarrow J/\psi K^* \text{ decays} \]

S. Shulga

Laboratory of Particle Physics, Joint Institute for Nuclear Research,
141980 Dubna, Moscow region, Russia
Francisk Skarina Gomel State University, Gomel, Belarus

**Abstract**

C++ package BtoVVana is aimed to extract physics parameters in decays of the scalar neutral B-meson to two vector particles in the intermediate state, and two leptons and two pseudoscalar mesons in the final state. Improved angular moments method is implemented in the package BtoVVana: the method uses the time dependent weighting functions and time-integrated observables with variable upper time limit. These observables allow to keep full time informative contents of the decay. By using the package BtoVVana it was shown that the statistical errors of extraction of the observables strongly depends on the choice of the angular weighting functions. The best angular weighting functions, which are the linear combinations of the amplitude angular functions, are implemented in the package BtoVVana. The first version of the program contains algorithms to extract physics parameters in the case of untagged decays. The package BtoVVana and program codes for the channels \( B_s^0 \rightarrow J/\psi \phi \) and \( B_d^0 \rightarrow J/\psi K^* \) in the Monte Carlo generator SIMUB were written in parallel and mutually tested with high precision.
Program summary

Title of the program: BtoVVana
Catalogue identifier: XXXX
Program summary URL : XXXX
Program obtainable from: http://cmsdoc.cern.ch/~shulga/BtoVVana/BtoVVana.html
Computer: PC, two Intel 2.0 GHz processors, 512MB RAM
Operating system: Linux Red Hat 6.1, 7.2, 7.3 and other platforms which allow to set ROOT package
Programming language used: C++: gcc 2.96 or 2.95.2 compiler suite with g++
Size of the package: 2.3 MB (0.4 MB compressed distribution archive), without ROOT libraries (additional 120 MB) and without input files with events
Distribution format: tar gzip file
Additional disk space required: Depends on the number of events: 35 Mb for 100 000 events (output of SIMUB generator)
PACS: 02.70.Tt; 02.70.Uu; 07.05.Tt; 13.25.Hw;
Keywords: particle physics, decay simulation, Monte Carlo methods, exclusive B-meson decay, angular moments method, CP-violation
Nature of the physical problem: The package BtoVVana has been developed to study the performance of the angular moments method of the $\Delta \Gamma$ determination from analysis of untagged decays $B^0_s \to J/\psi(\to \mu^+\mu^-)\phi(\to K^+K^-)$. By using the package BtoVVana it was shown that the method of angular moments gives stable results and is found to be an efficient and flexible tool for measurements with $B^0_s \to J/\psi\phi$ and $B^0_d \to J/\psi K^*$ decays.
Method of solution: The method of angular moments allows to construct a sequential chain of extraction of the physics parameters from events: $\Delta \Gamma \to$ initial transversity amplitudes $\to$ weak CP-violating phase $\to$ strong CP-conserving phases $\to$ $\Delta M_s$. Standard angular moments method was improved by using the time dependent weighting function and time-integrated observables with variable upper time limit. These observables allow to keep full time informative contents of the decay. This improved method is used in the package BtoVVana. It was shown that the statistical errors of extraction of the observables strongly depends on the choice of the angular weighting functions. The best weighting functions were found and implemented in the package BtoVVana. The program BtoVVana includes methods of extractions of observables as independent modules. In frame of the package BtoVVana user can
include new analysis modules. The package BtoVVana and program codes for the channels $B_s^0 \rightarrow J/\psi \phi$ and $B_d^0 \rightarrow J/\psi K^*$ in the Monte Carlo generator SIMUB were written in parallel and mutually tested with high precision.

Restrictions on the complexity of the problem: Program processes any number of files with any number of events; program tested with about 200 files including $2 \times 10^6$ events.

Typical running time: On a PC/Linux with 2.0 GHz processors BtoVVana package spends 15 seconds for every 100 000 events.
1 Introduction

The package BtoVVana has been developed to analyze the decays of neutral pseudoscalar B-mesons into two vector mesons decaying into two muons and two pseudoscalar mesons. Two channels of this type are included in the B-physics generator SIMUB [1] used here for precise testing of the package BtoVVana: $B^0_s(t), \overline{B}^0_s(t) \rightarrow J/\psi(\rightarrow \mu^+\mu^-) \phi(\rightarrow K^+K^-)$ and $B^0_d(t), \overline{B}^0_d(t) \rightarrow J/\psi(\rightarrow \mu^+\mu^-) K^*(\rightarrow K\pi)$. These channels contain rich physics information because of nontrivial angular distributions and time dependence of the decays.

At present we have the information from D0 experiment at the TEVATRON about observation of 337 events with decay of the first type $B^0_s \rightarrow J/\psi\phi$ and 1370 events of the second type $B^0_d \rightarrow J/\psi K^*$ [2]. The D0 data sample corresponds to an integrated luminosity 0.22 fb$^{-1}$, collected in 2002-2004. The statistics will be increased up to 4000 $B^0_s \rightarrow J/\psi\phi$ events at the D0 in the nearest future. Assuming the $b\bar{b}$ cross section to be 100 $\mu$b, the number of expected $B^0_s \rightarrow J/\psi\phi$ events, using 2 fb$^{-1}$ of integrated luminosity, will be equal to 41400 in the BTEV experiment [3]. About 8500 $B^0_d \rightarrow J/\psi K^*$ events are reconstructed in a data sample taken by the BELLE detector in the KEKB $e^+e^-$ collider [4].

The systematic study of these decays will be performed after 2007 year on the LHC detectors. About 200 000 $B^0_s \rightarrow J/\psi\phi$ events at the CMS, 100 000 events at the ATLAS and 75 000 events at the LHCb are expected to be obtained during first year of low luminosity operation [5].

Possibilities of the current experiments for these important decays can be improved by using optimal methods of the analysis. For large statistics the best method is likelihood fit while for small statistics this method becomes unstable for big number of unknown physics parameters, and the angular moments method is more preferable [6]. The latest method allows one to build a sequential chain of extraction of the physics parameters [1]. The goal of the paper is to formulate an optimal scenario of this analysis which can be realized for real data by using the package BtoVVana pre-
2 Structure of the package

Files of the package BtoVVana are kept under the BtoVVana directory which contains subdirectories `mak`, `src`, `doc`, `res`: directory `src` contains the source codes of the program; `mak` contains `Makefile`, command files for compilation `make_release` and execution `run`; `bin` contains the results of the compilation (object files) and the executable file (this directory is created automatically by command `make_release`); `doc` includes documentations; `res` is the user directory for data and results.

The structure of the directory tree one can be found in `Makefile`:

```bash
source_dirs = ../mak
source_dirs+= T_run_Read
source_dirs+= T_Run_Read/T_Accum_Measure
source_dirs+= T_Run_Read/T_Accum_Measure/T_AngMomMethod
source_dirs+= T_Run_Read/T_Accum_Measure/T_AngMomMethod/T_WeightFunc
source_dirs+= T_Run_Read/T_Accum_Measure/T_AngMomMethod/T_StandardModel DG
source_dirs+= T_Run_Read/T_Accum_Measure/T_AngMomMethod/T_WeightFunc\    /T_TimeIntObs
source_dirs+= T_Run_Read/T_Accum_Measure/T_AngMomMethod/T_WeightFunc\    /T_TimeIntObs/T_Physics
source_dirs+= T_Utility
```

The directory tree reflects a logical structure and dependences of the classes of the package. Command `make_release` processes all `.C` files in directories included in the list above. As a rule the name of the directory coincides with the name of the class which is placed in the directory. The structure of the program is shown in Fig.1.

Class `T_run_Read` is intended for reading the events from external files. This class contains the main loop over events. For mode `Mode Loop = 2` the loop calls
the virtual dummy method which will be overlapped in the derived class. Class T_run_Read contains the methods for regime Mode_Loop = 3 in which the angle and time distributions are created (see Fig. 3).

The daughter class T_Accum_Measure overlaps some dummy methods of the mother class T_run_Read and is intended for regime Mode_Loop = 2. The main data members of the class are the so called ”containers” which accumulate the information event by event, average the observables, and write the results to the final listing.

The class T_AngMomMethod carries out the extraction of the observables by using the angular moments method. Two ”containers” are implemented in the class T_Accum_Measure: both of the ”containers” are described as objects of the same class T_AngMomMethod with different angular weighting functions.

The main data members of the class T_AngMomMethod are four objects of the class T_TimeIntObs and three objects of the class T_StandardModel_DG.

The class T_TimeIntObs contains descriptions of the time-integrated observables which are obtained by accumulation of the event weights.

The class T_StandardModel_DG performs calculations of the width Γ and width difference ΔΓ by using the found observables in case of the Standard Model prediction of the small CP-violating weak phase.

The weights of the events are found by using the class T_WeightFunc which is the mother class for class T_TimeIntObs.

Extraction of the final observables based on some predefined physics parameters with its errors which described in class T_Physics. The object of this class should be created in the main program by the user in case of processing the real data. In case of processing the data from generator SIMUB one needs to set logical parameter bRealData = false in the main function. In this case the program will read the generator physics parameters from the first event.

Directory T_Utility includes some auxiliary functions and classes.

The structure of the package BtoVVana is the same as the structure of the gener-
ator SIMUB [1] which allows to obtain the events $B_s^0(t), \overline{B}_s^0(t) \rightarrow J/\psi(\rightarrow l^+l^-) \phi(\rightarrow K^+K^-)$ and $B_d^0(t), \overline{B}_d^0(t) \rightarrow J/\psi(\rightarrow l^+l^-) K^*(\rightarrow K\pi)$ with full physics contents and high precision of generation of the kinematical variables.

3 Preparation of input files by the SIMUB generator

Subpackage SIMUB/BB_dec allows to obtain the events with $B_s^0 \rightarrow J/\psi \phi$ and $B_d^0 \rightarrow J/\psi K^*$ decays with full physics contents by setting the channel option "COPT SI1" in command file BB_dec/mak/run [1] and using the decay mechanism option "B0SM 1" ("B0DM 1") and decay channel option "B0SC 1" ("B0DC 1") for $B_s^0 (B_d^0)$ mesons. The format of the SIMUB event is defined by option FRMT. Subpackage SIMUB/BB_dec has a mode Mode_Loop==2 to produce events in the format readable by the program BtoVVana. The variable Mode_Loop is set in the file BB_dec/mak/run by option MODD 2. In this case the format defined by option FRMT is ignored and the events are written in ROOT tree in the format defined in the constructor of class T_Loop (subpackage SIMUB/BB_dec) in the following way:

```c
if(Mode_Loop==2){
  hfile= new TFile(fname,"RECREATE","B0s->J/PsiPhi or B0d->J/PsiK* tree");
  tree = new TTree("T","B0s->J/PsiPhi decay tree");
  tree->Branch("VB" ,&VB ,"x/D:y:z:t:tau");
  tree->Branch("PB" ,&PB ,"x/D:y:z:E:M");
  tree->Branch("Pa" ,&Pa ,"x/D:y:z:E:M");
  tree->Branch("Pa1",&Pa1,"x/D:y:z:E:M");
  tree->Branch("Pa2",&Pa2,"x/D:y:z:E:M");
  tree->Branch("Pb" ,&Pb ,"x/D:y:z:E:M");
  tree->Branch("Pb1",&Pb1,"x/D:y:z:E:M");
  tree->Branch("Pb2",&Pb2,"x/D:y:z:E:M");
  tree->Branch("DFG",&DFG,"delta1/D:delta2:Wphi:nA02:nAP2:nAT2:
```
Here the `fname` is a name of the output file defined by option `OUTF 'NAME.root'`, the branches `VB` and `PB` define the secondary 4-vertex of B-decay and 4-momentum of B-meson, branches `Pa` and `Pb` define 4-momentum of $J/\psi$ and $\phi$ ($K^*$) mesons, branches `Pa1` and `Pa2` define 4-momentum of $\mu^+$ and $\mu^-$ mesons, branches `Pb1` and `Pb2` define 4-momentum of $K^+$ and $K^-$ mesons ($K$ and $\pi$ for $K^*$), the branch `DFG` defines the physics parameters of the decay $B \to J/\psi\phi$ ($B_0^0 \to J/\psi K^*$): strong CP-conserving phases $\delta_1 = \arg(A^*_\parallel A_\perp)$ (named as `delta1`) and $\delta_2 = \arg(A_\perp)$ (`delta2`; we set $\arg(A_0) = 0$), weak CP-violating phase $\phi$ (`Wphi`), initial transversity amplitudes squared $A^2_0, A^2_\perp, A^2_\parallel$ (names `nA02`, `nAP2`, `nAT2`), $B_0^0 (B_0^0)$ width $\Gamma$, widths of light and heavy states $\Gamma_L, \Gamma_H, \Delta \Gamma = \Gamma_H - \Gamma_L$ (with the corresponding program names `GB`, `GL`, `GH`, `DG`), B mass $M_{B^0}$ and masses of light and heavy states $M_L, M_H$, $\Delta M \equiv M_H - M_L$ (program names `MB`, `ML`, `MH`, `DM`), generator resolutions [1] or detector resolution named by `dx`, `dy`, `dz`, `dt` for four kinematical variables $\cos \Theta_{\mu^+}$, $\cos \Theta_{K^+}$, $\chi$ and $t$, correspondently, where three angles are defined in Fig. 2, and $t$ is the B meson proper lifetime.

If the program `BtoVVana` is used for the generator data, then the data members from the branch `DFG` contain physics parameters from the generator. In case of analysis of real data these data members should contain theoretical predictions to calculate some values which depend on the measured values and external parameters and for comparison.

Any number of the input files with events can be treated by the program `BtoVVana`. The names of these files are written in the main program.
4 Quick start with the package BtoVVana

The package is tested on Linux (RedHat 7.x, 8.x) platforms and uses the ROOT package [7]. To run the package, it is necessary to set the environment variables of the ROOT.

To install the program, it is enough to untar the file BtoVVana.tar.gz in the working directory and the BtoVVana directory with the structure described above will be created.

The main function is placed in the file mak/T_Accum_Measure_main.C. The user can set here the following parameters:

- code of B-meson \texttt{KF\_Code} (Pythia KF-code is 531 for $B_0^s$ and 511 for $B_0^d$);
- number of the input files \texttt{NFiles} and its names collected in the array \texttt{cFile};
- \texttt{Mode\_Loop} = 2 to extract the physics parameters or \texttt{Mode\_Loop} = 3 to draw the angular and proper time distributions;
- maximal proper lifetime used to select the events (or for Monte Carlo generation in generator) \texttt{Time\_maximal\_mmDc};
- the upper limit for the proper lifetime of B-meson \texttt{Time\_UP\_mmDc} (in units of [mm/c]), \texttt{Time\_UP\_mmDc} = \texttt{Time\_maximal\_mmDc} as a rule;
- second choice of the upper limits for proper lifetime \texttt{Time\_Zero\_mmDc};
- parameter \texttt{GammaS\_mmDcm1} may be set as arbitrary; to minimize the errors of measurements, one needs to set \texttt{GammaS\_mmDcm1} closely to the true value of $\Gamma$ ($\Gamma$ is B-meson width in units [mm/c]$^{-1}$);
- logical parameter \texttt{bRealData} = \texttt{true} in case of analysis of real data, otherwise the program will process the events from the generator; the only difference is that in case of \texttt{bRealData} = \texttt{false} the user does not need to define the object of \texttt{T\_Physics \_Phys} in the \texttt{main} function because this object will be
defined within the program as the data member by method Parameter_SIMUB of the class T_AngMomMethod by using the special physics records in the events written by generator SIMUB (see branch DFG in the class T_run_Read which one can found in package BtoVVana and in generator SIMUB); in case of bRealData = true the user needs to set an object of the class T_Physics in the main function by using the data from the previous run of the program or from other sources;

• number of events NEVENTS (will be ignored if it is less than the number of the events in all the input files);

• to draw a simple histogram, one has to set bSIMPLE_HISTOG = true otherwise it will be drawn histograms with error bars as it is shown in Fig.3;

• logical variable DEBUG = true for extended information in the listing used for debug of the program;

• logical variable bShortINF = true for short information in the listing;

To compile the program, one has to go into mak directory and execute the command make_release. After that, one should start the program by the command run.

Example of the main program and listing with results of the program are placed in section ”Test Run Input and Output”.

5 Class T_run_Read: loop over the events

The main part of the class T_run_Read was done automatically by the ROOT method MakeClass of the TTree class. The constructor of the class T_run_Read contains the following parameters: KF_Bmeson is a PYTHIA KF-code of B-meson, NumbFiles is a number of the files for analysis, Mode_Loop is a processing mode, and cFile[] is a list of the file names.
Class `T_run_Read` contains the loop which reads the data and writes the event information in the following data members of the class:

double VB_x, VB_y, VB_z, VB_t, VB_tau, // Vertex and lifetime
    PB_x, PB_y, PB_z, PB_E, PB_M, // B momentum and mass
    Pa_x, Pa_y, Pa_z, Pa_E, Pa_M, // J/ψ momentum and mass
    Pa1_x, Pa1_y, Pa1_z, Pa1_E, Pa1_M, // μ⁺ momentum and mass
    Pa2_x, Pa2_y, Pa2_z, Pa2_E, Pa2_M, // μ⁻ momentum and mass
    Pb_x, Pb_y, Pb_z, Pb_E, Pb_M, // Φ/K₅ momentun and mass
    Pb1_x, Pb1_y, Pb1_z, Pb1_E, Pb1_M, // K⁺(K) momentum and mass
    Pb2_x, Pb2_y, Pb2_z, Pb2_E, Pb2_M; // K⁻(π) momentum and mass

    // Physics, kinematics and Monte Carlo parameters (branch DFG):
    double DFG_delta1, DFG_delta2, // strong CP conserving phases
    DFG_wphi, // weak CP violating phase
    DFG_nA02, DFG_nAP2, DFG_nAT2, // initial polarized amplitudes squared (0,||,T)
    DFG_GB, DFG_GL, DFG_GH, DFG_DG, // see definition in previous section
    DFG_MB, DFG_ML, DFG_MH, DFG_DM, // see definition in previous section
    DFG_dx, DFG_dy, DFG_dz, DFG_dt, // see definition in previous section
    DFG_tmax, // maximal allowed B lifetime
    DFG_cThb1, DFG_cTha1, // cos(Θ_K⁺) cos(Θ_μ⁺)
    DFG_sChi, DFG_cChi, DFG_ChI, // cos(Chi), sin(Chi), Chi
    DFG_Blifetime, // B Lifetime
    DFG_fDist; // value of distribution function for given
        // cos(Θ_K⁺), cos(Θ_μ⁺), Chi, B-lifetime
    int DFG_KFCode; // PYTHIA KF code of B meson

Using the method `Mom_To_LorentzVector` the event momenta and vertex are rewritten in ROOT Lorentz vector variables: `P_B_L, P_a_L, P_a1_L, P_a2_L, P_b_L, P_b1_L, P_b2_L, V_B_gen, V_B_dec, DVertex_B = V_B_dec - V_B_gen`. The data members - `cThe_b1_R, cThe_a1_R, Chi_R, B_lifetime_R` - present the values `cos Θ_μ⁺`, `cos Θ_K⁺`, `χ`, `t` (see Fig.2) which are calculated for the current event by the method
Reco_Helicity_Angles (if the position of a method is not given, then the method belongs to the class T.run.Read in this section by default).

The number of the input files for the processing is set in the constructor of the class T.run.Read by parameter N.umbFiles. The file names are collected in the character array in the main program and passed in the constructor by the parameter cFile[].

The method Loop(NEVENTS) performs the loop over all the events in the file chain. The user can find two logical keys bSIMUB_Angles and bSIMPLE_HISTOGR inside the method Loop. If bSIMUB_Angles = true, then the angles and B lifetime directly from generator SIMUB (from the branch DFG) will be used for histograms, otherwise the angles and B lifetime which were reconstructed from the momenta and vertex, are used for histograms; if bSIMPLE_HISTOGR = true, then we can obtain the pictures with simple histograms for the three angles and B lifetime, while for bSIMPLE_HISTOGR = false one can obtain histograms with error bars. The latest kind of pictures for decay $B_s^0 \to J/\psi \phi$ can be seen in Fig. 3 for 100 000 events. We have used the ”main setting” of physics parameters by default for $B_s^0 \to J/\psi \phi(B_d^0 \to J/\psi K^*)$ decays: $\delta_1 = \pi$, $\delta_2 = 0$, $\phi_c(s) = 0.04$, $A_0^2 = 0.54(0.56)$, $A_1^2 = 0.16$, $A_1^2 = 1 - A_0^2 - A_2^2$, $\Gamma = 1/\tau_B$, $\tau_B = 1.464(1.571)$ ps, $\Delta \Gamma/\Gamma = -0.2(-0.01)$, $T_{max} = 2$mm/c, $N_{reso} = 50000$, $dt = dx = dy = 2/N_{reso}, dz = 2\pi/N_{reso}$, $T_{max} = 2$ mm/c, $x \equiv \Delta M/\Gamma = 20(0.73)$, where the values for $B_d^0 \to J/\psi K^*$ are shown in brackets if they do not coincide.

Switch of regimes Mode_Loop are placed in the main program. Class T.run.Read includes methods in a simple mode Mode_Loop = 3. Methods for the Mode_Loop = 2 are placed in the derived class T.Accum.Measure (see the next section). All other values of Mode_Loop are not used in the current version of the program.

It is convenient to create a separate derived class T.Accum.Measure specially to treat the events.

The main loop over the events is placed in class T.run.Read with non virtual
definition and therefore it is not redefined in the derived class T_Accum_Measure. The program uses the same method Loop placed in the parent class T_run_Read for all regimes.

The method Loop contains the switch on different regimes of processing by the key Mode_Loop. This switch has the following form (it is a simplified form of the loop):

```cpp
for(jentry=0; jentry<NEVENTS ; jentry++) {
    if(Mode_Loop==2)
        if(!Reco_Accum_Measure()) out("$WARNING 1(goto next evt)",0);
        else if(Mode_Loop==3) Fill_Hist_RECO(); }
if   (Mode_Loop==2) Average_Measure_OutResult();
else if(Mode_Loop==3) Draw_Hist(0);
```

Within the class T_run_Read there are dummy virtual methods Reco_Accum_Measure and Average_Measure_OutResult which are overlapped in the derived class. The method Reco_Accum_Measure transforms the momenta of the particles into angles and accumulates observables. The method Average_Measure_OutResult performs averaging and printing the results of the observables extraction.

6 Class T_Accum_Measure: accumulation of the observables

Class T_Accum_Measure has T_run_Read as a parent class and its constructor has the form:

```cpp
T_Accum_Measure(bool DEBUG,
             int KF_B_To_Analyse, int NumbFiles, char *cFile[],
             int Mode_Loop, double TimeUP_mmDc, double Time_maximal_mmDc,
             double Time_Zero_mmDc, double GammaS_mmDcm1,
             bool bRealData, T_Physics *Phys);
```
where the parameters DEBUG, KFB_To_Analyse, NumbFiles, cFile[] and ModeLoop are used for the constructor of the parent class T_run_Read described in the previous section. All other parameters are described in section 4.

At present the class T_Accum.Measure includes the extraction of observables by the angular moments method described in [1].

The main methods of the class overlap the parent virtual methods Reco_Accum.Measure and Average.Measure_OutResult. These methods are called by derived method T_run_Read::Loop which is not overlapped in the class T_Accum.Measure. Class T_Accum.Measure has the ”containers” to accumulate the information in the loop over the events. These data are processed and printed in their final form in function Average.Measure_OutResult. ”Containers” are represented as the following data members of the class:

T_AngMomMethod *AngMomMethod_A;
T_AngMomMethod *AngMomMethod_B;

where AngMomMethod_A(AngMomMethod_B) are used to treat the events by using the angular moments method with the weighting function which is described in [1] as ”set A” (”set B”).

The user can introduce new ”containers” for other ways of treatment. For example, tagged events of decay $B_s^0(t)\overline{B}_s^0(t) \rightarrow J/\psi \phi$ containing the oscillation phenomenon allow to extract the parameter $\Delta M$. According to the scenario presented here, the extraction of the $\Delta M$ by using the tagged samples should be done after analysis of the untagged sample. The method to extract $\Delta M$ and corresponding ”container” will be developed in the next version of the program BtoVVana.

A pointer to the object of the class T_Accum.Measure *AM is defined in the main function. Constructor of the class T_Accum.Measure defines the ”containers” (parameters of the constructor will be described in the following section):

AngMomMethod_A = new T_AngMomMethod(1, ...);
AngMomMethod_B = new T_AngMomMethod(2, ...);
Then the main program calls the loop method inherited from the parent class \texttt{AM \rightarrow Loop(500000)}. It passes the number of events to the loop to be processed. The events from the external files are read and written into the data members of the parent class \texttt{T\_run\_Read}. The structure of the loop was shown in the previous section. As you can see, the loop calls the method \texttt{Reco\_Accum\_Measure} which is overlapped in the class \texttt{T\_Accum\_Measure} and has the following form:

\begin{verbatim}
bool T\_Accum\_Measure::Reco\_Accum\_Measure() {
    Mom\_To\_LorentzVector();
    Reco\_Helicity\_Angles();
    Count\_MeasureEv++; if(Count\_MeasureEv==1) Measure\_Init();
    Measure\_Accum(); return true; }
\end{verbatim}

The first method \texttt{Mom\_To\_LorentzVector} transforms the momenta to the ROOT Lorentz vectors and to the angles (in the frame shown in Fig. 2) and the B-meson proper lifetime calculated by using the function \texttt{Reco\_Helicity\_Angles}. The last function is placed in directory \texttt{T\_Utility/F\_Reco\_HelicityAngles}.

Before treatment of the first event the initialization is performed by means of the method \texttt{Measure\_Init}. In the treatment of events from the generator \texttt{SIMUB} the method \texttt{Measure\_Init} calls the method \texttt{Parameter\_SIMUB} to initialize the “containers” \texttt{AngMomMethod\_A} and \texttt{AngMomMethod\_B} by the physics parameters (in this section the methods without references have the placement in the class \texttt{T\_Accum\_Measure} by default).

These parameters are used to extract the observables from the time-integrated observables which have been obtained by class \texttt{T\_TimeIntObs}, and to perform analytical calculations to compare them with the approximate estimations obtained by the Monte Carlo method (see description of the class \texttt{T\_Physics} below). It is helpful to test the both the event generator and the analysing package \texttt{BtoVVana}.

As it is described in the previous section, the method \texttt{Loop} derived from class \texttt{T\_run\_Read} calls the virtual method \texttt{Reco\_Accum\_Measure} which overlapped in the class described in this section. Further the chain of calls is the following: \texttt{Reco\_Accum\_Measure()
Measure_Accum() \rightarrow Measure_Accum_AngMomMethod(). The structure of the latest method is very general:

```cpp
void T_Accum_Measure::Measure_Accum_AngMomMethod(){
    AngMomMethod_A->Init_Angles(cThe_b1_R, cThe_a1_R, Chi_R, B_lifetime_R);
    AngMomMethod_A->Accumulation_TimeIntObs ();
    AngMomMethod_A->Set_False_FlagInit_Angles();
    AngMomMethod_B->Init_Angles(cThe_b1_R, cThe_a1_R, Chi_R, B_lifetime_R);
    AngMomMethod_B->Accumulation_TimeIntObs ();
    AngMomMethod_B->Set_False_FlagInit_Angles();
}
```

It is not difficult to include other user "containers" into this method.

In method `T_run_Read::Loop(int)` after the loop the method `Average_Measure_OutResult` is called. This method performs averaging and printing the results of the measurements (it is a simplified form):

```cpp
void T_Accum_Measure::Average_Measure_OutResult(){
    AngMomMethod_A->Average_TimeIntObs ();
    AngMomMethod_A->Out_Result_TimeIntObs(" Set A ");
    AngMomMethod_B->Average_TimeIntObs ();
    AngMomMethod_B->Out_Result_TimeIntObs(" Set B ");
}
```

### 7 Class T_AngMomMethod: extraction of physics parameters by using the angular moments method

The main data members of the class `T_AngMomMethod` are four pointers to the objects of the class `T_TimeIntObs` and three pointers to the objects of the class `T_StandardModel_DG`:

- `T_TimeIntObs *Obs_TG0 ; // T , Gp = 0`
- `T_TimeIntObs *Obs_TGp ; // T , Gp = G'`
- `T_TimeIntObs *Obs_T0G0; // T0, Gp = 0`
Two different definitions of the time-integrated observables can be considered according to [1]: \( \tilde{O} \) and \( \hat{O} \). The first case corresponds to the zero parameters \( \Gamma' = 0 \) and the second one - to nonzero \( \Gamma' \) defined by the user (see the next section about definition of \( \Gamma' \)). In the program the user can set also two different values of the upper time limit \( T \) and \( T_0 \).

Constructor and methods of the class \texttt{T\_AngMomMethod} are

\begin{verbatim}
T_AngMomMethod (....);
void Init_Angles (....);
void Parameter_SIMUB(....)
void Accumulation_TimeIntObs ();
void Average_TimeIntObs ();
void Out_Result_TimeIntObs (const Char_t *cc);
virtual void Show (const Char_t *c);
\end{verbatim}

The constructor calls four constructors for four objects of the class \texttt{T\_TimeIntObs}. Each of the methods listed above calls four methods with the same name for the four objects of the class \texttt{T\_TimeIntObs} and for the three objects of the class \texttt{T\_StandardModel\_DG}. For example,

\begin{verbatim}
void T_AngMomMethod::Accumulation_TimeIntObs(){
    Obs_TGO ->Accumulation_TimeIntObs();
    Obs_TGp ->Accumulation_TimeIntObs();
    Obs_TOG0->Accumulation_TimeIntObs();
    Obs_TOGp->Accumulation_TimeIntObs();
    SM_DG_TGp_TOG ->Accumulation();
    SM_DG_TGp_TOGp->Accumulation();
    SM_DG_TOG_TOG0->Accumulation(); }
\end{verbatim}
Some features in using the objects of the classes \(T_{\text{TimeIntObs}}\) and \(T_{\text{StandardModel DG}}\) the user can find in the method \texttt{Parameter.SIMUB} (this method calls the methods of \(T_{\text{TimeIntObs}}\) objects only), and in the method \texttt{Average.TimeIntObs}:

```cpp
void T_AngMomMethod::Average_TimeIntObs(){
    Obs_TGO ->Average_TimeIntObs(); ...;
    SM_DG_TGp_TG0->T_StandardModel_DG_Init(*Obs_TGp, *Obs_TG0);
    SM_DG_TGp_TG0->Average(); SM_DG_TGp_TG0->Calc_DG(true); ...;}
```

Method \(T_{\text{StandardModel DG}}.\text{Init}\) initiates the object to extract \(\Gamma\) and \(\Delta\Gamma\). Method \texttt{Average} averages values to calculate the correlation between two types of the observables. Method \texttt{Calc_DG} calculates \(\Gamma\) and \(\Delta\Gamma\).

8 Class \(T_{\text{WeightFunc}}\): weighting functions

Two types of the angular weighting function are proposed in [6, 1] to extract the observables: \(w_i^{(A)}(\theta_{l+}, \theta_{K+}, \chi)\) and \(w_i^{(B)}(\theta_{l+}, \theta_{K+}, \chi)\) \((i = 1, 6)\). As it is shown in [1], it is helpful to consider the time dependent weighting functions which can be written in the general case as

\[
W_i^{(A)}(\theta_{l+}, \theta_{K+}, \chi, t; \Gamma', T) = \exp(\Gamma't)w_i^{(A)}(\theta_{l+}, \theta_{K+}, \chi)\Theta(T - t)
\]

and similarly for \(W_i^{(B)}\). Weighting function \(W_i^{(A/B)}\) is named in the class \(T_{\text{WeightFunc}}\) as method \texttt{wi} \((i = 0, 5)\).

The integer data member \texttt{WF.Type} of the class \(T_{\text{WeightFunc}}\) sets the type of weighting functions \((1 \rightarrow A, 2 \rightarrow B)\) and it is initialized by the class constructor. The parameters of the weighting functions \(\Gamma'\) and \(T\) are the constructor input parameters, named as \texttt{GammaS.mmDcm1} (in unit \((\text{mm}/c)^{-1}\)) and \texttt{TimeUP.mmDc} (in unit \((\text{mm}/c))\). The \(\Gamma'\) represents a first approximation for the measured value \(\Gamma\) to be corrected in the current analysis of the untagged decays.

Together with method \texttt{wi} there are also the methods with names \texttt{wi_cTha1}, \texttt{wi_cThb1}, \texttt{wi_Chi} and \texttt{wi_t} in the class \(T_{\text{WeightFunc}}\). These methods represent
the derivatives of the function \( w_i \) with respect to arguments \( \cos \Theta_{\mu}, \cos \Theta_{K}, \chi \) and \( t \), correspondently. The derivatives are used in class \texttt{T\_TimeIntObs} to calculate the systematical errors or to estimate the detector response \[1\].

9 Class T\_TimeIntObs: six time-integrated observables

Class \texttt{T\_TimeIntObs} is a derived class to the class \texttt{T\_WeightFunc}. It is used to accumulate and average the time-integrated observable, and to extract the physics parameters.

In the general form by means of the weighting function \( W_{ij}^{(X)} \) \((X = A, B)\) on the set of the \( N(T_{\text{max}}) \) events we have observables and their statistical errors in the following form \[1\]:

\[
\hat{b}_i^{(\text{exp})}(\Gamma', T) = \frac{1}{N(T_{\text{max}})} \sum_{j=1}^{N(T)} W_{ij}^{(X)}(\Gamma', T), \tag{1}
\]

\[
\delta \hat{b}_i^{(\text{exp})}(\Gamma', T) = \frac{1}{N(T_{\text{max}})} \sqrt{\sum_{j=1}^{N(T)} \left[ \hat{b}_i^{(\text{exp})}(\Gamma', T) - W_{ij}^{(X)}(\Gamma', T) \right]^2},
\]

where \( W_{ij}^{(X)}(\Gamma', T) \equiv W_{ij}^{(X)}(\theta_{ij}^+, \theta_{ij}^{K+}, \chi^j, t^j; \Gamma', T) \) is a value of the weighting function for the \( j \)-th event. Including the time dependent \( \Theta(T - t) \)-function in the weighting function allows to keep the time informative contents in the value \( \hat{b}_i^{(\text{exp})}(\Gamma', T) \) and in the same time to have good statistical errors which are caused by a large size of the proper time interval \([0, T]\).

The class constructor \texttt{T\_TimeIntObs} is defined by the following signature:

\[
\text{T\_TimeIntObs(char *cc, int Type\_WF,} \\
\text{double GammaS\_mmDcm1, TimeUP\_mmDc, TimeMAX\_mmDc,} \\
\text{bool DEBUG, bRealData, T\_Physics *P\_hys);} \text{;}
\]

Three parameters \texttt{Type\_WF}, \texttt{GammaS\_mmDcm1}, \texttt{TimeUP\_mmDc} are sent to the mother class \texttt{T\_WeightFunc} (see the previous section). Parameter \texttt{TimeMAX\_mmDc} defines the
By means of the logical parameter `bRealData` the user can set either the mode for the real data or the mode for the generator data. The latest case does not use the method parameter `P_hys` and data member `Phys` of the class will be filled by the generator data taken from the first event by the method `Parameter_SIMUB`.

In case of `bRealData = true` the user has to define the object of the class `T_Physics` in the main program and send it into the object of the class `T_Accum_Measure` defined in the main program.

The constructor and list of the methods of the class are as follows:

```cpp
T_TimeIntObs(char *cc, int Type_WF,
             double GammaS_mmDc, TimeUP_mmDc, TimeMAX_mmDc,
             bool DEBUG, b_RealData, T_Physics *P_hys);
void FinalObserv_TimeIntObs (const Char_t *c);
void Parameter_SIMUB(double delta1, delta2, TotalPhaseWeak,
                        nA0_02, nA0_P2, nA0_T2, Gamma_Bmes_GeV, Gp_GeV, Gm_GeV, Delta_G_GeV,
                        Gen_dcTb1, Gen_dcTa1, Gen_dChi, Gen_dt);
void Accumulation_TimeIntObs();
void Average_TimeIntObs ();
void Out_Result_TimeIntObs (const Char_t *c);
virtual void Show (const Char_t *c);
```

If we compare this list with the list of the methods of the class `T_AngMomMethod`, it is possible to note that the last 5 methods have the same names as the method names of the class `T_AngMomMethod` because the calls of these methods are the main purpose of the corresponding methods of the class `T_AngMomMethod` (see explanation in the description of the class `T_AngMomMethod`).

Method `FinalObserv_TimeIntObs` is called by method `Out_Result_TimeIntObs` and calculates the final observables or their combinations according to the formu-
\[ |A_0(0)|^2 = \frac{\hat{b}_1^{(\text{exp})}(\Gamma', T)}{\hat{b}_1^{(\text{exp})}(\Gamma', T) + \hat{b}_2^{(\text{exp})}(\Gamma', T) + \hat{b}_3^{(\text{exp})}(\Gamma', T)/\hat{\gamma}}, \]
\[ |A_\| (0)|^2 = \frac{\hat{b}_2^{(\text{exp})}(\Gamma', T)}{\hat{b}_1^{(\text{exp})}(\Gamma', T) + \hat{b}_2^{(\text{exp})}(\Gamma', T) + \hat{b}_3^{(\text{exp})}(\Gamma', T)/\hat{\gamma}}, \]
\[ |A_\perp (0)|^2 = \frac{\hat{b}_3^{(\text{exp})}(\Gamma', T)/\hat{\gamma}}{\hat{b}_1^{(\text{exp})}(\Gamma', T) + \hat{b}_2^{(\text{exp})}(\Gamma', T) + \hat{b}_3^{(\text{exp})}(\Gamma', T)/\hat{\gamma}}, \]
\[ \cos(\delta_2 - \delta_1) = \frac{\hat{b}_5^{(\text{exp})}(\Gamma', T) \sqrt{\hat{b}_1^{(\text{exp})}(\Gamma', T) \hat{b}_2^{(\text{exp})}(\Gamma', T)}}{\hat{b}_1^{(\text{exp})}(\Gamma', T) \hat{b}_2^{(\text{exp})}(\Gamma', T)}, \] (2)

where we consider the initial amplitudes normalized as \(|A_0(0)|^2 + |A_\| (0)|^2 + |A_\perp (0)|^2 = 1\). We have introduced the function \(\hat{\gamma}\):

\[ \hat{\gamma}(\Delta \Gamma_L, \Delta \Gamma_H, \cos \phi_c(s), T) = \frac{\hat{G}_H}{\hat{G}_L}, \]
\[ \hat{G}_L = (1 \pm \cos \phi_c(s)) \frac{e^{\Delta \Gamma_L T/2} - 1}{\Delta \Gamma_L} - (1 \mp \cos \phi_c(s)) \frac{e^{-\Delta \Gamma_H T/2} - 1}{\Delta \Gamma_H}. \] (4)

\(\Delta \Gamma_{L/H}\) are measured parameters related with physics values of widths of light and heavy states \(\Gamma_{L/H}\) by

\[ \Delta \Gamma_L = 2(\Gamma' - \Gamma_L), \quad \Delta \Gamma_H = -2(\Gamma' - \Gamma_H). \] (5)

The method \texttt{FinalObserv\_TimeIntObs} also calculates two values:

\[ \sin \phi_c(s) \cos \delta_{1,2} = \frac{\hat{b}_{1,6}^{(\text{exp})}(\Gamma', T)}{\sqrt{\hat{b}_{2,1}^{(\text{exp})}(\Gamma', T) \hat{b}_3^{(\text{exp})}(\Gamma', T)}} \hat{\beta}. \] (6)

where

\[ \hat{\beta}(\Delta \Gamma_L, \Delta \Gamma_H, \cos \phi_c(s), T) = \frac{\sqrt{\hat{G}_L \hat{G}_H}}{\hat{Z}}, \quad \hat{Z} = \frac{1 - e^{\Delta \Gamma_L T/2}}{\Delta \Gamma_L} + \frac{1 - e^{-\Delta \Gamma_H T/2}}{\Delta \Gamma_H}. \] (7)

Eqs. (6) contain the weak phase in the left- and right-handed side and might be helpful in case of large values of weak phase \(\phi_c(s)\) when a large violation of the Standard Model predictions, \(\phi_c(s) \approx 0.03\), takes place.

Values \(\hat{\gamma}\) and \(\hat{\beta}\) are calculated in the data member \texttt{T\_Physics *Phys} of the class \texttt{T\_TimeIntObs}.
Table 1: Time-integrated observables obtained by data member \texttt{Obs\_TG0} of class \texttt{T\_AngMomMethod}. The sample of 300 000 (3 000 in parentheses) untagged $B^0_s \rightarrow J/\psi \phi$ events from SIMUB generator with the ”main setting” of physics parameters (see section 5) is used

| $i$ | $\hat{b}_i(0, T)_{T=T_{\text{max}}=0}$ | $\hat{b}_i(0, T)_{T=T_{\text{max}}=2}$ | $\hat{b}_i^{(\text{exp})}(0, T)_{T=T_{\text{max}}=2}$ | Statistical errors |
|-----|---------------------------------|---------------------------------|---------------------------------|-----------------|
| 1   | 0.54                            | 0.5225                          | 0.5207 (0.488)                 | 0.0014 (0.014)  |
| 2   | 0.30                            | 0.2903                          | 0.2940 (0.380)                 | 0.0021 (0.021)  |
| 3   | 0.16                            | 0.1873                          | 0.1849 (0.132)                 | 0.0020 (0.020)  |
| 4   | 0.0088                          | 0.0009                          | 0.0032 (0.011)                 | 0.0020 (0.020)  |
| 5   | -0.4025                         | -0.3894                         | -0.3916 (-0.316)               | 0.0029 (0.029)  |
| 6   | -0.0118                         | -0.0012                         | -0.0021 (-0.002)               | 0.0031 (0.031)  |

The results of extraction of final observables (2) and (6) are given in Tables 1 and 2.

The direct numerical calculations have shown that the difference between the values of observables $\hat{b}_i(T)$ ($i = 1, 2, 3, 5$), calculated with $\phi_c^{(s)} = 0$ and $\phi_c^{(s)} = 0.04$, does not exceed 0.01%. Even in case of statistics of 100 000 events this difference is negligibly small as compared with statistical errors for these observables. Therefore, the assumption $\phi_c^{(s)} \approx 0$ is a good approximation in case of Standard Model.

To determine the initial transversity amplitudes in case of SM, it is convenient to use $\Gamma' = 0$ and for large $T$ we have $\hat{\gamma} \approx \Gamma_L/\Gamma_H$.

The first observable in the list of four observables of the class \texttt{T\_AngMomMethod} is \texttt{Obs\_TG0} which is initialized with setting $\Gamma' = 0$ and with $T = T_{\text{max}}$ as the upper time limit. The results from \texttt{Obs\_TG0} are presented by two tables in the listing (see section "Test Run Input and Output").

These data are given in Tables 1 and 2.
Table 2: Determination of values (2) and (6) by using the observables $\hat{b}_i^{(exp)}(0,T)$ ($T = 2/1/0.5/0.25/0.125$ mm/c$^{-1}$) for $300000 \ B_s^0 \rightarrow J/\psi \phi$ events with the "main setting" (see section 5) of physics parameters

| Parameter | Input | Extracted value by (2), (6) | Statistical error |
|-----------|-------|----------------------------|-------------------|
| $A_0^2/A_0^0$ | 0.556 | 0.565/0.568/0.565/0.568 | 0.005/0.005/0.006/0.007 |
| $A_\perp^2/A_0^0$ | 0.296 | 0.294/0.293/0.293/0.288 | 0.004/0.004/0.005/0.006 |
| $|A_0(0)|^2$ | 0.54 | 0.538/0.538/0.538/0.539 | 0.001/0.001/0.002/0.002 |
| $|A_\parallel|^2$ | 0.30 | 0.304/0.305/0.304/0.306 | 0.002/0.002/0.003/0.003 |
| $|A_\perp|^2$ | 0.16 | 0.158/0.157/0.158/0.155 | 0.002/0.002/0.002/0.003 |
| $\cos(\delta_2 - \delta_1)$ | -1 | -1.001/-1.000/-0.998/-0.999 | 0.009/0.009/0.010/0.012 |
| $\cos\delta_1 \sin\delta_\perp^{(s)}$ | 0.04 | 0.15/0.17/0.16/-0.04 | 0.09/0.12/0.22/0.50 |
| $\cos\delta_2 \sin\delta_\perp^{(s)}$ | -0.04 | -0.07/-0.07/0.03/-0.03 | 0.10 /0.14/0.26/0.61 |
Table 3: Time-integrated observables obtained by data member Obs\_TGp of class T\_AngMomMethod for 300 000 (3 000 in parentheses) untagged $B^0_s \to J/\psi\phi$ events with the "main setting" (see section 5) of physics parameters

| $i$ | $\hat{b}_i(\Gamma, T_{max})$, Eq. (8) | $\hat{b}_i^{(exp)}(\Gamma, T_{\text{max}})$, Eq. (1) | Stat.err. |
|-----|-------------------------------------|-----------------------------------------------|-----------|
| 1   | 2.117                               | 2.139 (1.95)                                  | 0.014 (0.12) |
| 2   | 1.176                               | 1.157 (1.40)                                  | 0.019 (0.17) |
| 3   | 0.989                               | 1.024 (0.78)                                  | 0.020 (0.16) |
| 4   | 0.010                               | 0.035 (0.07)                                  | 0.018 (0.19) |
| 5   | -1.578                              | -1.603 (-0.94)                                | 0.028 (0.24) |
| 6   | -0.013                              | -0.010 (0.11)                                 | 0.028 (0.31) |

As it is shown in the first two columns of Tabl. 1, the time dependence of observables is essential in case the number of events is more than 3 000 because for the 3 000 events the differences $\hat{b}_i(0, T)_{T=T_{max} \to 0} - \hat{b}_i(0, T)_{T=T_{max}=2}$ are comparable with statistical errors. This conclusion depends on the value of width difference $\Delta \Gamma$. In case of the $B^0_d \to J/\psi K^*$ channel the Standard Model predicts a small value of $\Delta \Gamma$ ("main setting" for $B^0_d \Delta \Gamma/\Gamma = -0.01$, see section 5) and in this case $\hat{b}_i(0, T)_{T=T_{max} \to 0} \approx \hat{b}_i(0, T)_{T=T_{max}=2}$.

The second observable in class T\_AngMomMethod is Obs\_TGp which is considered here with parameter $\Gamma' = \Gamma$ ($\Gamma$ is a true value of the $B$ meson width) and with $T_{\text{max}}$ as the upper lifetime limit. The results from listing tables of observable Obs\_TGp are shown here in Tabl. 3 and 4 and in listing of section "Test Run Input and Output".

It is better to use observables $\hat{b}_i^{(exp)}(0, T_{\text{max}})$ to extract the initial observables (2) and (6) (compare the statistical errors in Tabl. 2 and 4).

Extraction of combinations $\cos \delta_{1,2} \sin \phi_c^{(s)}$ is shown in Tabl. 5.
Table 4: Determination of values (2) and (6) by using the observables $\hat{\delta}_i^{(exp)}(\Gamma, T_{max})$ for $300\,000\ B^0_s \rightarrow J/\psi\phi$ events with the "main setting" (see section 5) of physics parameters

| Parameter | Input value | Extracted value | Stat.error |
|-----------|-------------|-----------------|------------|
| $A_2^2/A_0^2$ | 0.556 | 0.5409 | 0.010 |
| $A_3^2/A_0^2$ | 0.296 | 0.3038 | 0.0066 |
| $|A_0(0)|^2$ | 0.54 | 0.5421 | 0.0030 |
| $|A_2|^2$ | 0.30 | 0.2932 | 0.0045 |
| $|A_3|^2$ | 0.16 | 0.1647 | 0.0034 |
| $\cos(\delta_2 - \delta_1)$ | -1 | -1.019 | 0.020 |
| $\cos\delta_1 \sin\phi_{c}^{(s)}$ | 0.04 | 0.138 | 0.073 |
| $\cos\delta_2 \sin\phi_{c}^{(s)}$ | -0.04 | -0.029 | 0.081 |

Table 5: Determination of values depended on weak phase by using the observables $\hat{\delta}_i^{(exp)}(\Gamma, T_{max})$ extracted from Monte Carlo data for two values of $\phi_{c}^{(s)} = 0.4$ and 0.04.

The events sample has been generated for the case of "main setting" for physics parameters

| Parameter | Input value | 100 000 events | 200 000 events | 400 000 events |
|-----------|-------------|----------------|----------------|---------------|
| $\cos\delta_1 \sin\phi_{c}^{(s)}$ | 0.04 | 0.24 ± 0.14 | 0.13 ± 0.09 | 0.12 ± 0.07 |
| $\cos\delta_2 \sin\phi_{c}^{(s)}$ | -0.04 | -0.05 ± 0.15 | 0.10 ± 0.11 | -0.03 ± 0.07 |
| $\cos\delta_1 \sin\phi_{c}^{(s)}$ | 0.389 | 0.49 ± 0.16 | 0.44 ± 0.11 | 0.35 ± 0.08 |
| $\cos\delta_2 \sin\phi_{c}^{(s)}$ | -0.389 | -0.29 ± 0.16 | -0.31 ± 0.12 | -0.41 ± 0.09 |
10 Class T\_Physics: physics values

For observables $\hat{b}_i(\Gamma', T)$ we have the following formulae [1]:

\begin{align*}
\hat{b}_1(\Gamma', T) &= |A_0(0)|^2 \hat{G}_L(\Gamma', T)/\bar{L}(T_{\text{max}}), \\
\hat{b}_2(\Gamma', T) &= |A_{\parallel}(0)|^2 \hat{G}_L(\Gamma', T)/\bar{L}(T_{\text{max}}), \\
\hat{b}_3(\Gamma', T) &= |A_{\perp}(0)|^2 \hat{G}_H(\Gamma', T)/\bar{L}(T_{\text{max}}), \\
\hat{b}_4(\Gamma', T) &= |A_0(0)| |A_{\parallel}(0)| \hat{Z}(\Gamma', T) \cos \delta_1 \sin \phi_c^{(s)}/\bar{L}(T_{\text{max}}), \\
\hat{b}_5(\Gamma', T) &= |A_0(0)| |A_{\parallel}(0)| \hat{G}_L(\Gamma', T) \cos(\delta_2 - \delta_1)/\bar{L}(T_{\text{max}}), \\
\hat{b}_6(\Gamma', T) &= |A_0(0)| |A_{\perp}(0)| \hat{Z}(\Gamma', T) \cos \delta_2 \sin \phi_c^{(s)}/\bar{L}(T_{\text{max}}), \quad (8)
\end{align*}

where $\bar{L}(T_{\text{max}})$ is a normalization factor, which has the form:

\begin{align*}
\bar{L}(T_{\text{max}}) &= (|A_0(0)|^2 + |A_{\parallel}(0)|^2) \hat{G}_L(0, T_{\text{max}}) + |A_{\perp}(0)|^2 \hat{G}_H(0, T_{\text{max}}), \quad (9)
\end{align*}

From expressions (8) one can see that $\hat{b}_1(\Gamma', T) + \hat{b}_2(\Gamma', T) + \hat{b}_3(\Gamma', T) = 1$.

In the general case with sizable weak phase $\phi_c^{(s)}$ to extract the physics values we can use the following equations:

\begin{align*}
\frac{\hat{b}_{1,2,5}^{(\text{exp})}(\Gamma', T)}{\hat{b}_{1,2,5}^{(\text{exp})}(\Gamma'', T_0)} &= \frac{\mu_L(\Delta \Gamma_L, \Delta \Gamma_H; \cos \phi_c^{(s)}; [\Gamma', T], [\Gamma'', T_0])}{\mu_L(\Delta \Gamma_L, \Delta \Gamma_H; \cos \phi_c^{(s)}; [\Gamma', T], [\Gamma'', T_0])}; \quad (10)
\frac{\hat{b}_3^{(\text{exp})}(\Gamma', T)}{\hat{b}_3^{(\text{exp})}(\Gamma'', T_0)} &= \frac{\mu_H(\Delta \Gamma_L, \Delta \Gamma_H; \cos \phi_c^{(s)}; [\Gamma', T], [\Gamma'', T_0])}{\mu_H(\Delta \Gamma_L, \Delta \Gamma_H; \cos \phi_c^{(s)}; [\Gamma', T], [\Gamma'', T_0])}; \quad (11)
\frac{\hat{b}_{4,6}^{(\text{exp})}(\Gamma', T)}{\hat{b}_{4,6}^{(\text{exp})}(\Gamma'', T_0)} &= \frac{\rho(\Delta \Gamma_L, \Delta \Gamma_H; [\Gamma', T], [\Gamma'', T_0])}{\rho(\Delta \Gamma_L, \Delta \Gamma_H; [\Gamma', T], [\Gamma'', T_0])}, \quad (12)
\end{align*}

where

\begin{align*}
\mu_{L/H} &\equiv \frac{\hat{G}_{L/H}(\Delta \Gamma_L, \Delta \Gamma_H; \cos \phi_c^{(s)}; [\Gamma', T])}{\hat{G}_{L/H}(\Delta \Gamma_L, \Delta \Gamma_H; \cos \phi_c^{(s)}; [\Gamma'', T_0])}, \\
\rho &\equiv \frac{\hat{Z}(\Delta \Gamma_L, \Delta \Gamma_H; [\Gamma', T])}{\hat{Z}(\Delta \Gamma_L, \Delta \Gamma_H; [\Gamma'', T_0])}. \quad (13)
\end{align*}

Therefore, to extract tree parameters $\Delta \Gamma_L, \Delta \Gamma_H$ and $\cos \phi_c^{(s)}$ we need the experimental values of the time-integrated observables $\hat{b}_i^{(\text{exp})}$ only.
In case of small values $\hat{b}_{4,6}^{(\text{exp})}$ we have a small $\sin \phi_c^{(s)} \cos \delta_{1,2}$, but it does not mean the small weak phase $\phi_c^{(s)}$ as it is predicted by the Standard Model. The opposite case of the sizable observables $\hat{b}_{4,6}^{(\text{exp})}$ will directly show the signal of the beyond Standard Model physics. In the last case we can extract $\Delta \Gamma_L$ and $\Delta \Gamma_H$ from the two equations (12). If we use $\Gamma'$ as the true value of the B meson width (found from other sources), then we will have a single unknown value $\Delta \Gamma_L = \Delta \Gamma_H \equiv \Delta \Gamma$. Simplification of the system of equations (10)- (12) in case of the Standard Model will be considered below.

After extraction of $\Delta \Gamma_L$, $\Delta \Gamma_H$ and $\cos \phi_c^{(s)}$ we can calculate the values $\hat{\gamma}$ (3) and $\hat{\beta}$ (7). The values $\hat{\gamma}$ and $\hat{\beta}$ are calculated in class T_Photics and used in the method FinalObserv_TimeIntObs of the class T_TimeIntObs to calculate the final observables (2) and (6). It is the first goal of the class T_Photics.

The second goal of the class T_Photics is to calculate exact theoretical values of the observables (8) to compare them with the approximate values obtained by (1) in the class T_TimeIntObs.

For these purposes the class T_Photics collects all the physics values including transversity amplitudes with its errors and three user parameters $\Gamma'$, $T$ and $T_{\text{max}}$.

In the main method Init of the class T_Photics one can see the scenario to calculate of the physics values described above. The list of the input parameters of the method Init includes 8 physics parameters: delta1 and delta2 (strong CP-conserving phase $\delta_{1,2}$), TotalPhaseWeak (weak CP-violating phase $\phi_c^{(s)}$), nA0_02, nA0_P2, nA0_T2 (initial transversity amplitudes squared $|A_0(0)|^2$, $|A_\parallel(0)|^2$ and $|A_\perp(0)|^2 = 1 - |A_0(0)|^2 - |A_\parallel(0)|^2$), Gp and Gm (widths of light and heavy B meson states $\Gamma_L$ and $\Gamma_H$), their errors, and 3 user defined parameters: TimeUP (upper time limit $T$), TimeMAX (maximal time $T_{\text{max}}$) and Gprime ($\Gamma'$) to use them for calculation by formulae (8). Data member oEk_hat ($k = 0, 5$) of the class T_Photics are the observables (8) in case of $\Gamma'$ defined by user while oEk_tilde ($k = 0, 5$) are the observables (8) in case of $\Gamma' = 0$. 

26
Class T.StandardModel_DG : extraction of $\Gamma$ and $\Delta \Gamma$ in case of the Standard Model

The name of the class T.StandardModel_DG means that it includes the extraction of $\Gamma$ and $\Delta \Gamma$ in case of the Standard Model expectation for the weak phase: $\phi_c^{(s)} \approx 0$.

In this case we may not use the small values of the observables $\bar{b}_{4,6}^{(exp)}$ and, according to (4) and (7), we have:

$$
\mu_L = \mu_L(\Delta \Gamma_L; [\Gamma', T], [\Gamma'', T_0]),
\mu_H = \mu_H(\Delta \Gamma_H; [\Gamma', T], [\Gamma'', T_0]).
$$

It means that extraction of $\Delta \Gamma_L$ and $\Delta \Gamma_H$ can be performed by numerical solving of the separate equations.

Three cases numbered as $\text{KK} = 1, 2, 3$ (see method mu.DG_Equa_Solut) are realized in the class T.StandardModel_DG: $\text{KK} = 1$ in case of $\Gamma' \neq 0$, $\Gamma'' = 0$, $T_0 = T$ (used in data member SM.DG.TGp.TG0 of the class T.AngMomMethod), $\text{KK} = 2$ in case of $\Gamma' = \Gamma'' = 0$, $T_0 \neq T$ (used in data member SM.DG.TG0.TG0), and $\text{KK} = 3$ in case of $\Gamma' = \Gamma'' \neq 0$, $T_0 \neq T$ (used in SM.DG.TGp.TGp). Only the case $\text{KK} = 1$ is presented in listing which is shown in the section "Test Run Input and Output". In each case we have 3 equations (10) to determine $\Delta \Gamma_L$ (names DGL.0, DGL.1 and DGL.4 in output listing) and one equation (11) for $\Delta \Gamma_H$ (with name DGH.2 in output listing).

Combining three values $\Delta \Gamma_L$ from (10) with one value $\Delta \Gamma_H$ from (11), we have three value $\Delta \Gamma = 0.5(\Delta \Gamma_L + \Delta \Gamma_H)$ (names DG.0, DG.1 and DG.4 in listing of section "Test Run Input and Output") and three value $\Gamma = \Gamma' - 0.25(\Delta \Gamma_L - \Delta \Gamma_H)$ (names G.0, G.1 and G.4 in listing) corresponding to combinations of indices [1,3], [2,3] and [5,3] of observables.

Three cases $\text{KK}=1,2,3$ of extraction of $\Gamma$ and $\Delta \Gamma$ are presented as three tables in listing.

The results of the first method of extraction of $\Gamma$ and $\Delta \Gamma$ ($\text{KK}=1$) weakly dependent on user defined $\Gamma'$ value (see Tabl. 6 obtained by SM.DG.TGp.TG0). Due to weak dependences of the result on $\Gamma'$ we fix this parameter in the following tables as $\Gamma' = \Gamma$. 

27
Table 6: Determination of values $\Gamma$ and $\Delta\Gamma$ by using the object SM\_DG\_TGp\_TG0 ($KK = 1$) for $300\,000 B^{0}_s \rightarrow J/\psi\phi$ events with the ”main setting” of physics parameters.

Combinations of indices [1,3] [2,3] [5,3] are explained in section 11

| Parameter | $\Gamma'$ | Input | [1,3] | [2,3] | [5,3] |
|-----------|-----------|-------|-------|-------|-------|
| $\Gamma$, [mm/c]$^{-1}$ | 2.278 | 2.278 | 2.228 ± 0.015 | 2.268 ± 0.020 | 2.231 ± 0.020 |
| $\Delta\Gamma$, [mm/c]$^{-1}$ | 2.278 | -0.456 | -0.506 ± 0.030 | -0.587 ± 0.040 | -0.512 ± 0.040 |
| $\Gamma$, [mm/c]$^{-1}$ | 2.392 | 2.278 | 2.227 ± 0.015 | 2.268 ± 0.020 | 2.230 ± 0.020 |
| $\Delta\Gamma$, [mm/c]$^{-1}$ | 2.392 | -0.456 | -0.508 ± 0.030 | -0.591 ± 0.040 | -0.515 ± 0.041 |
| $\Gamma$, [mm/c]$^{-1}$ | 2.164 | 2.278 | 2.229 ± 0.015 | 2.268 ± 0.019 | 2.232 ± 0.020 |
| $\Delta\Gamma$, [mm/c]$^{-1}$ | 2.164 | -0.456 | -0.504 ± 0.029 | -0.582 ± 0.039 | -0.509 ± 0.039 |

The results of the second method of extraction of $\Gamma$ and $\Delta\Gamma$ ($KK = 2$) depend on the user defined time up limits $T$ and $T_0$ (see Tabl. 7 obtained by SM\_DG\_TG0\_TG0). The best choice is $T_0 = 0.5$ mm/c in case of maximal $T = T_{max} = 2$ mm/c.

The results of the third method of extraction of $\Gamma$ and $\Delta\Gamma$ ($KK = 3$) depend on time upper limits $T$ and $T_0$ defined by the user (see Tabl. 8 obtained by SM\_DG\_TGp\_TG0p). The best choice is $T_0 = 0.25$ mm/c in case of maximal $T = T_{max} = 2$ mm/c. The last method of extraction of $\Gamma$ and $\Delta\Gamma$ is better among all the considered methods.

To avoid the superfluous information, the user needs to set DEBUG = false and bShortINF = true. In this case the listing consists of two parts for two methods of treatment by using the ”set A” and ”set B” weighting functions [1] (in section ”Test Run Input and Output” you can see the listing with the case of ”set B” only). The ”set B” weighting functions are the linear combinations of six angular functions which define the amplitude of the process while the ”set A” weighting functions are not expressed linearly via the angular functions. This is the main reason why the statistical errors for observables in case of ”set B” is about 2 times smaller than for
Table 7: Determination of values $\Gamma$ and $\Delta\Gamma$ by using the object SM DG TG0 TOG0 (KK = 2) for 300 000 $B_s^0 \rightarrow J/\psi \phi$ events with the "main setting" of physics parameters. Combinations of indices [1,3] [2,3] [5,3] are explained in section 11

| Parameter | $T_0; T$, $\frac{mm}{c}$ | Input $\frac{mm}{c}$ | [1,3] $\frac{mm}{c}$ | [2,3] $\frac{mm}{c}$ | [5,3] $\frac{mm}{c}$ |
|-----------|--------------------------|---------------------|---------------------|---------------------|---------------------|
| $\Gamma$  | 1; 2                     | 2.278               | 2.246 ± 0.017       | 2.281 ± 0.022       | 2.255 ± 0.022       |
| $\Delta\Gamma$ | 1; 2                 | -0.456              | -0.478 ± 0.033      | -0.549 ± 0.043      | -0.498 ± 0.043      |
| $\Gamma$  | 0.5; 2                   | 2.278               | 2.242 ± 0.015       | 2.272 ± 0.019       | 2.243 ± 0.019       |
| $\Delta\Gamma$ | 0.5; 2                | -0.456              | -0.485 ± 0.029      | -0.544 ± 0.038      | -0.487 ± 0.039      |
| $\Gamma$  | 0.25; 2                  | 2.278               | 2.240 ± 0.021       | 2.249 ± 0.025       | 2.242 ± 0.025       |
| $\Delta\Gamma$ | 0.25; 2               | -0.456              | -0.509 ± 0.042      | -0.527 ± 0.049      | -0.512 ± 0.050      |
| $\Gamma$  | 0.125; 2                 | 2.278               | 2.257 ± 0.028       | 2.236 ± 0.032       | 2.210 ± 0.033       |
| $\Delta\Gamma$ | 0.125; 2              | -0.456              | -0.477 ± 0.056      | -0.434 ± 0.065      | -0.383 ± 0.066      |
Table 8: Determination of values $\Gamma$ and $\Delta \Gamma$ by using the object SM$_{DG}$TG$_{p}$TO$_{Gp}$ (KK = 3) for 300,000 $B^0_s \rightarrow J/\psi \phi$ events with the "main setting" of physics parameters. Combinations of indices [1,3] [2,3] [5,3] are explained in section 11.

| Parameter | $T_0; T, \frac{mm}{c}$ | Input $\frac{mm}{c}$ | [1,3] $\frac{mm}{c}$ | [2,3] $\frac{mm}{c}$ | [5,3] $\frac{mm}{c}$ |
|-----------|-------------------------|---------------------|---------------------|---------------------|---------------------|
| $\Gamma$  | 1; 2                    | 2.278               | 2.225 ± 0.019       | 2.279 ± 0.026       | 2.238 ± 0.026       |
| $\Delta \Gamma$ | 1; 2                    | -0.456             | -0.514 ± 0.039     | -0.622 ± 0.053     | -0.541 ± 0.053     |
| $\Gamma$  | 0.5; 2                  | 2.278               | 2.228 ± 0.015       | 2.268 ± 0.020       | 2.231 ± 0.020       |
| $\Delta \Gamma$ | 0.5; 2                   | -0.456           | -0.506 ± 0.030     | -0.587 ± 0.040     | -0.512 ± 0.040     |
| $\Gamma$  | 0.25; 2                 | 2.278               | 2.231 ± 0.014       | 2.263 ± 0.018       | 2.237 ± 0.018       |
| $\Delta \Gamma$ | 0.25; 2                 | -0.456           | -0.508 ± 0.029     | -0.573 ± 0.036     | -0.520 ± 0.037     |
| $\Gamma$  | 0.125; 2                | 2.278               | 2.238 ± 0.016       | 2.256 ± 0.019       | 2.222 ± 0.020       |
| $\Delta \Gamma$ | 0.125; 2                | -0.456         | -0.496 ± 0.032     | -0.532 ± 0.039     | -0.465 ± 0.039     |
the "set A". In previous sections we have described the results for "set B" only.

In previous sections we have used three keys for tuning the program BtoVVana: $\Gamma', T$ and $T_0$. To improve solving the equations for $\Delta\Gamma_{L,H}$, the user can change limits of arguments in the method mu_DG_Equa_Solut (class T_StandardModel_DG) which solves the equations.

12 Conclusion

Extraction of physics information in decays $B_s^0 \to J/\psi\phi$ and $B_d^0 \to J/\psi K^*$ by using the angular moments method with time dependent and time-integrated observables, has a number of attractive features, which are demonstrated by package BtoVVana:

- it is an unbinned method;
- it uses full time informative contents of time-dependent decays;
- it uses full available statistics;
- it gives stable results in case of small statistics;
- it allows one to separate the extraction of physics values;
- it allows one to use different scenarios in case of the signals beyond Standard Model, or in case when it is justified;
- it is a flexible tool because it has a different ways to tune the extraction of observables (for example, to tune solving the equations) in the process of the real data treatment;
- and it is a visual method (see, for example, the Fig.4 in [1] for dependences of $\hat{b}_{1,2}$ on $\Delta\Gamma$).

The detailed tests have been performed for the package BtoVVana by means of a precise generator SIMUB. The tests have checked mutually the both programs SIMUB and BtoVVana with high precision.
The complex BtoVVana - SIMUB can be used to test other methods of extraction of physics information from decays $B^0_s \to J/\psi\phi$ and $B^0_d \to J/\psi K^*$.  

The program BtoVVana has clear structure and can be used as template to include new methods of treatments.

**Acknowledgements**

This work is dedicated to memory of A.A.Belkov who was the initiator of this work.


References

[1] A. Belkov and S. Shulga, Comp.Phys.Com. 156 (2004) 221-240; A. Belkov
and S. Shulga, Part. Nucl. Lett., 2[117] (2003) 12, hep-ph/0301105; see also
http://cmsdoc.cern.ch/~shulga/SIMUB/SIMUB.html.

[2] V.M.Abazov e.a. FERMILAB-Pub-04/225-E, hep-ex/0409043 v3 (10 Dec 2004);
K.Yip, in Proceedings of 39th Rencontres de Moriond on QCD and High-Energy
Hadronic Interactions, La Thuile, Italy, 28 Mar - 4 Apr 2004, hep-ex/0405024.

[3] Proceedings of the Workshop ”The CKM matrix and the unitary triangle”, 13-
16 February, 2002, Chapt.3. Editors: M.Battaglia, A.J.Buras, P.Gambino and
A.Stocchi, CERN-2003-002-corr, 10 October 2003, hep-ph/0304132.

[4] K. Abe e.a. BELLE-CONF-0438/ICHEP04 8-0864.

[5] Proceedings of the Workshop on Standard Model Physics (and more) at the
LHC. Sect.”B decays”. Editors: G.Altarelli, M.L.Mangano, CERN 2000-004, 9
May 2000.

[6] A.S. Dighe, I. Dunietz and R. Fleischer, Eur. Phys. J. C6 (1999) 647.

[7] R. Brun and F. Rademakers, Nucl. Instrum. Meth. A389 (1997) 81; see also
http://root.cern.ch/.
Test Run Input and Output

User defined parameters are defined in section 4 and placed in main program:

```c
TROOT root("Program BtoVVana"," xxx ");

int main(int argc, char **argv) {
    TApplication *theApp= new TApplication("App", &argc, argv);
    cout<<" ***** main() Start ***** "<<endl;
    //============= user set, Files with events and parameters ======
    const Int_t NFiles = 10; const Char_t *cFile[NFiles];
    cFile[0] = "$DAT1/10000ev_B0sJPsiPhi_1.root";
    cFile[1] = "$DAT1/10000ev_B0sJPsiPhi_2.root";
    ...
    cFile[9] = "$DAT1/10000ev_B0sJPsiPhi_10.root";
    Int_t KF_Code = 531; // 531 -> B0s, B0s
    Bool_t DEBUG = false; // switch for extended listing
    Bool_t bShortINF = true;
    Int_t Mode_Loop = 2; // 2: DGamma measurement; 3: angle distributions
    Double_t Time_UP_mmDc=2., Time_maximal_mmDc=2., Time_Zero_mmDc=0.5; //mm/c
    Double_t GammaS_mmDcm1;
    if (KF_Code == 531) GammaS_mmDcm1 = 2.278;
    else if(KF_Code == 511) GammaS_mmDcm1 = 2.27844; // Gd = 2.12326;
    else out_exit("main: check KF_Code or set GammaS_mmDcm1. STOP.");
    Bool_t bRealData = false;
    Double_t ErrFactor = 0.2, Reso_CosThetap = 0.022, Reso_CosThetaKp = 0.0076,
                               Reso_Chi = 0.04, Reso_t = 0.03; // [Reso_t] = mm/c
    Int_t NEVENTS = 100000;
    Bool_t bSIMPLE_HISTOGR = false;
    //============= end user setting. ================
    T_Physics *Phys = new T_Physics();
    Phys->Init(Time_UP_mmDc, Time_maximal_mmDc, 0., ErrFactor*0.,
```
if(Mode_Loop==2){
T_Accum_Measure *AM = new T_Accum_Measure(DEBUG, bShortINF, 
KF_Code, NFiles, cFile, Mode_Loop, 
Time_UP_mmDc, Time_maximal_mmDc, Time_Zero_mmDc, 
GammaS_mmDcm1, bRealData, Phys);
AM->Loop(NEVENTS, bSIMPLE_HISTOGR);
}
else if(Mode_Loop==3){
T_run_Read *SRAM=new T_run_Read(KF_Code,NFiles,cFile,Mode_Loop,DEBUG);
SRAM->Loop(NEVENTS, bSIMPLE_HISTOGR, 0);
}
if(Mode_Loop==3) theApp->Run();
cout<<" ***** main() End ***** "<<endl;
}

The output listing in short format has the form:

***** main() Start *****
--- ctor T_Accum_Measure ---
weight func [set B]
--- ctor T_TimeIntObs: ---
Gamma^prime = 4.49511e-13 GeV = 2.278 [mm/c]^-1
T_imeUP_mmDc = 1.01355e+13 GeV^-1 = 2 mm/c
T_imeMAX_mmDc= 1.01355e+13 GeV^-1 = 2 mm/c
Inform from T_TimeIntObs::Parameter_SIMUB in case of bSIMUB_Test:

\[ \Gamma_{B0s} = 4.49598 \times 10^{-13} \text{ GeV} = 2.27844 \text{ [mm/c]}^{-1} \]
\[ D_G = -8.99197 \times 10^{-14} \text{ GeV} = -0.455689 \text{ [mm/c]}^{-1} \]
\[ G_{p,GL} = 4.94558 \times 10^{-13} \text{ GeV} = 2.50629 \text{ [mm/c]}^{-1} \]
\[ G_{m,GH} = 4.04638 \times 10^{-13} \text{ GeV} = 2.0506 \text{ [mm/c]}^{-1} \]
\[ 0.5(G_{m}+G_{p}) = 4.49598 \times 10^{-13} \text{ GeV} = 2.27844 \text{ [mm/c]}^{-1} \]
\[ (G_{m}-G_{p}) = -8.99197 \times 10^{-14} \text{ GeV} = -0.455689 \text{ [mm/c]}^{-1} \]

\[ D_G/G = -0.2 \]

RESOLUTION: \(d_cT_b_1, d_cT_a_1, d_Chi, d_t = 4 \times 10^{-5}, 4 \times 10^{-5}, 0.000125664, 4 \times 10^{-5}\)

BEGIN: T_Accum_Measure::Average_Measure_OutResult

*** [set B] *** T_Accum_Measure::Average_Measure_OutResult

\[ \text{NEVENT} = 100000 \text{ TimeUP} = 2 \text{ mm/c}, \quad \Gamma_{S} = 2.278 \text{ [mm/c]}^{-1} \]

------------- BEGIN TGp -2prime- BEGIN ---

\[ o_0 = 2.1242 \pm 0.024(1.1\%) \pm 0.0000(0\%) = 2.1169 = oE_0 \]
\[ o_1 = 1.0584 \pm 0.033(3.1\%) \pm 0.0000(0\%) = 1.1760 = oE_1 \]
\[ o_2 = 1.0787 \pm 0.035(3.2\%) \pm 0.0000(0\%) = 0.9889 = oE_2 \]
\[ o_3 = -0.0165 \pm 0.031(188.3\%) \pm 0.0000(0\%) = 0.0099 = oE_3 \]
\[ o_4 = -1.5242 \pm 0.049(3.2\%) \pm 0.0000(0\%) = -1.5778 = oE_4 \]
\[ o_5 = 0.0469 \pm 0.047(99.8\%) \pm 0.0000(0\%) = -0.0133 = oE_5 \]

From T_Reco_CumObs::Out_Result_CumObs***

-------------------------------------------------------------------------

cVal Theor MC er_MC(erV/V%) er_st_cor(e/eV%) er_phys(e/e V%)
A_{02}/A_{P2} 1.8000 2.0070 0.072(3.6\%) 0.072(3.6\%) 0.00 (0.0\%)
A_{P2}/A_{02} 0.5556 0.4983 0.018(3.6\%) 0.018(3.6\%) 0.00 (0.0\%)
A_{T2}/A_{02} 0.2963 0.3221 0.26 (79.5\%) 0.012(3.6\%) 0.26 (99.9\%)
A_{02} 0.5400 0.5493 0.077(14.1\%) 0.005(0.9\%) 0.077(99.8\%)
A_{P2} 0.3000 0.2737 0.039(14.3\%) 0.008(2.9\%) 0.039(97.9\%)
A_{T2} 0.1600 0.1769 0.12 (65.5\%) 0.006(3.4\%) 0.12 (99.9\%)

36
\[
\begin{align*}
cos(d_2-d_1) & = -1.0000 -1.0166 0.036\ (3.6\%) 0.036\ (3.6\%) 0.00\ (0.0\%)
\cos(d_1)\sin(\phi) & = 0.0400 -0.0672 0.14\ (201.4\%) 0.13\ (188.3\%) 0.048\ (35.5\%)
\cos(d_2)\sin(\phi) & = -0.0400 0.1350 0.17\ (122.7\%) 0.13\ (99.8\%) 0.096\ (58.2\%)
\end{align*}
\]

\[--------------------- \text{END TGp -2prime- END} \---------------------------\]

[... output for other three observables is placed here]

**SM\_DG\_TGp\_TG0** from **T\_AngMomMethod::Out\_Result\_TimeIntObs**# BEGIN

| cVal | Val\_TheorC | Val\_MC | -erVal\_MC(\%eV)+/- | er\_sys(\%eV) |
|------|-------------|---------|----------------------|---------------|
| DGL\_0 | -0.4566 | -0.4493 | 0.0388 | 8.6\% | 0.0000 | 0\% |
| DGL\_1 | -0.4566 | -0.8170 | 0.1113 | 13.6\% | 0.0000 | 0\% |
| DGH\_2 | -0.4548 | -0.6876 | 0.0912 | 13.3\% | -0.0000 | 0\% |
| DGL\_4 | -0.4566 | -0.5358 | 0.1068 | 19.9\% | 0.0000 | 0\% |
| DG\_0 | -0.4557 | -0.5685 | 0.0496 | 8.7\% | 0.0000 | 0\% |
| DG\_1 | -0.4557 | -0.7523 | 0.0720 | 9.6\% | 0.0000 | 0\% |
| DG\_4 | -0.4557 | -0.6117 | 0.0703 | 11.5\% | 0.0000 | 0\% |
| G\_0 | 2.2784 | 2.2184 | 0.0248 | 1.1\% | 0.0000 | 0\% |
| G\_1 | 2.2784 | 2.3103 | 0.0360 | 1.6\% | 0.0000 | 0\% |
| G\_4 | 2.2784 | 2.2401 | 0.0351 | 1.6\% | 0.0000 | 0\% |
| DG\_0/G\_0 | -0.2000 | -0.2563 | 0.0225 | 8.8\% | -0.0000 | 0\% |
| DG\_1/G\_1 | -0.2000 | -0.3256 | 0.0316 | 9.7\% | -0.0000 | 0\% |
| DG\_4/G\_4 | -0.2000 | -0.2731 | 0.0317 | 11.6\% | -0.0000 | 0\% |

[... output for other two types of DG extraction is placed here]

END *** T\_Accum\_Measure::Average\_Measure\_Out\_Result() END ***

############ T\_run\_Read::Loop End T\_run\_Read::Loop(). ############

***** main() End *****

37
Figure captions

**Figure 1.** Structure of the package BtoVVana. Inserted squares mean derived classes. Diamond inside the class means the data members of the classes shown by arrow.

**Figure 2.** Definition of physical angles to describe decays $B^{0}_s(t), \bar{B}^{0}_s(t) \to J/\psi(\to l^+l^-) \phi(\to K^+K^-)$ in the helicity frame [1].

**Figure 3.** Angular distributions obtained by means of class T_run_Read for 100 000 events with $B^{0}_s \to J/\psi(\to l^+l^-)\phi(\to K^+K^-)$ decays ("main" setting of physics parameters, see section 5).
Figure 1
Figure 2
Figure 3