Last updates on TAUOLA

Jakub Zaremba

IFJ Kraków

Mini Workshop on Tau Physics, CINVESTAV, Mexico, 23 June 2017

Outline

1. Tauola-bbb – „BaBar” initialization of TAUOLA currents developed by collaboration
   a) Code validation
   b) Framework for adding user-defined modeling of decays
   c) How to plug hadronic current into TAUOLA
2. Fitting framework prototype
3. Few words on „projection operators”

Tauola-bbb

- Recently we have prepared new version of TAUOLA recreating BaBar setup of their basic simulations .
- We believe, it is beneficial for user to have setup of TAUOLA replicating what was used by renowned experiment
- Tarball available at:

https://twiki.cern.ch/twiki/bin/view/FCC/Tauola

http://annapurna.ifj.edu.pl/~jzaremba/resources/

- Full documentation at:

https://arxiv.org/abs/1609.04617v2 - sent to publication

Tauola-bbb - Code validation

- We have compared results of our new initialization of TAUOLA with BaBar collaboration production files
- Agreement was checked with MC-TESTER with samples of 1 617 945 000 decays
- The MC-TESTER-based tests accounted for over 133 decay channels (including those with multiple photons in final state generated by PHOTOS) as used by BaBar collaboration
- All invariant masses constructed from stable decay products were monitored

Tauola-bbb – Freamework for adding user-defined modeling of decays

- Our new version provides also more user flexibility to modify the generator physics initialization.
  - Most notably we added option to add/replace currents with pointers to user functions
  - To account for such modification we also gave possibility to change parameters of phase space generation (to improve efficiency of presamplers).
  - Core of TAUOLA code remain in FORTRAN but new version is a step towards C++, or another higher level programming language. Mixed language for models possible.
  - Examples are given in *demo-redefine* folder of *tauola-bbb*

How to plug hadronic current into TAUOLA

1. Code a **hadronic current** of desired channel in your preferred language.
2. Prepare function redefining currents

```
void name_of_your_redefinitions() {
vector<int> products(9);  //define products of your hadronic current
products[0] = -1; //pi-                    NOTE: those numbers  are TAUOLA  ID's, not PDG ID's.
products[1] =  2; //pi0                          Check Appendix C.1 of https://arxiv.org/abs/1609.04617 for codes used in TAUOLA
ChannelForTauola *pipi0_test1 = new ChannelForTauola(BR, products, "  Name used in output" , Your_function_of_hadronic_current );
RegisterChannel(channel_NO-2, pipi0_test1);  //see  *.output in any of demo  folders for channels list
}
```

3. Add relevant files into *makefile*

```
COMMAND_OBJECTS = taumain.o iniofc.o your_file1.o your_file2.o
EXTER_SRC1  = ../*.f ../tauola-c/*.c ../tauola-c/*.h your_file1.c your_file2.c
```

4. Call user redefinitions in your main program

Fortran:

```
CALL TauolaRedef ! register reinitialization function;
CALL INIofC    ! Has to be called before generation but after initialization
```

Or in simple words find:
CALL INIPHY
and paste those calls after after.

C++ :

```
In user program add: #include "Tauola/ChannelForTauolaInterface.h"
then use: Tauolapp::SetUserRedefinitions(name_of_your_redefinitions);
// for  fortran this is used in iniofc.c, where extern "C" void tauolaredef_() function  is defined
```

Hadronic current format

Format of hadronic current needs to follow the one used in TAUOLA.

Fortran:

SUBROTINE your_current(four_momentum_of_outgoing_particle1, …. , hadronic_current)

REAL four_momentum_of_outgoing_particle1(4), ….

COMPLEX hadronic_current(4)

if used later in C++ part it requires declaration:

extern "C" { void your_current_(float *four_moemntum_of_outgoing_particle1, …); }

Then used in same way as C++ function

C++ equivalent:

void your_current(float *four_momentum_of_outgoing_particle1, …. , complex<float> *hadronic_current)

Matrix element format

C++:

void Your_matrix_element(const float *Tau_4momentum, const float *Neutrino_4momentum, const float *Partilce1_4momentum, … , float &amplitude, float *polarimetric_vector)

FORTRAN:

SUBROUTINE Your_matrix_element(Tau_4momentum, Neutrino_4momentum, Partilce1_4momentum, … , amplitude, polarimetric_vector)

Class ChannelForTauola is overloaded and will recognize format of user provided function, therefore replacing hadronic currents and matrix elements works in exactly the same manner for all multiplicities of particles in final state.

Note: for LFV currents slot for neutrino momentum is used by one of particles.

amplitude = ω with all the constants in the folowing slide

polarimetric_vector = h_i in the following slide

Hadronic current and matrix element formats

Hadronic current: $J^\mu = N\{T_v^\mu[c_1(p_2-p_3)^v F_1 + c_2(p_3-p_1)^v F_2 + c_3(p_1-p_2)^v F_3] + c_4 q^\mu F_4 - \frac{i}{4\pi^2 f_\pi^2} c_5 \epsilon^\mu{}_{\nu\rho\sigma} p_1^\nu p_2^\rho p_3^\sigma F_5\}$

Matrix element: $\mathcal{M} = \frac{G}{\sqrt{2}} \bar{u}(N)\gamma^\mu(v + a\gamma_5)u(P)J_\mu$

$$2M\Gamma_x = \cos^2\theta_{Cabibo}\, G_{Fermi}^2 \frac{v^2+a^2}{2} \int dLips(Q; q_i; N)\omega(1+h_i s^i)$$

$$|\mathcal{M}|^2 = \cos^2\theta_{Cabibo}\, G_{Fermi}^2 \frac{v^2+a^2}{2} \omega(1+h_i s^i)$$

For some channels $\cos\theta_{cabbibo}$ of is to be replaced by sin.

$$\omega = P^\mu(\Pi_\mu - \gamma_{va}\Pi_\mu^5) \qquad H_\mu = \frac{1}{M}\left(M^2 g_\mu^\nu - P_\mu P^\nu\right)\left(\Pi_\nu^5 - \gamma_{va}\Pi_\nu\right)$$

$h_0 = 1$

$h_i = \frac{H_i}{\omega}$

$$\begin{aligned} \Pi_\mu &= 2\Big((J^* \cdot N)J_\mu + (J \cdot N)J_\mu^* - (J^* \cdot J)N_\mu\Big), \\ \Pi^{5\mu} &= 2\mathrm{Im}\ \epsilon^{\mu\nu\rho\sigma} J_\nu^* J_\rho N_\sigma, \\ \gamma_{va} &= -\frac{2va}{v^2+a^2}. \end{aligned}$$

Fitting framework prototype

- We are working on fitting framework for three pion currents that could work in semi analytical framework but would enable easy switch between models that you can plug into TAUOLA later
- In its core, framework uses for fitting minuit2 library
- Use of multiple cores is supported
- Main target of this framework is to allow simple way of fitting multidimensional distributions and calculate errors, correlation matrices etc.

Projection operators

- We are also working on „projection operators” to acces fully differential distributions of 3 scalar τ decay modes
- Once finished they should be easily to pluged into prepared fitting framework
- This is by far more complex that when it was used for the first time at CLEO collaboration
- Recent „low-energy” (Belle, BaBar) have energies much higher than CLEO, where taus were produced at rest
- Neutrino momentum needs to be reconstructed now and it is very complex
- Gate to full dynamic of medium energy QCD at ~0.2% precision level

Thank you for your attention !