TOPAZ0 4.0 - Single channel version

First of all a reminder. All the internal flags of the program are hard coded and initialized in `SUBROUTINE TINIT`, which is called by the job-control. file. Here, for the sake of completeness, the complete list of inputs is given, devoting particular care in commenting the meaning of the new flags. For more details concerning flags already present in earlier versions, the reader is referred to the documentation of the previous releases.

<u>SE</u>: scaling factor for numerical integration error;

<u>OWEAK</u>: residual weak corrections: running (R) or fixed (F); at LEP 2 the correct choice is R;

<u>OWBOX</u>: weak boxes included (Y) or not (N). Away from the Z-resonance the *correct* choice is Y;

<u>OUO</u>: includes (S) the sub-leading two-loop $\mathcal{O}(\alpha^2 m_t^2)$ corrections. It is the *recommended* choice;

<u>OHC</u>: next-to-leading and higher order hard photon contribution included (Y) or not (N);

<u>ORAD</u>: selects the type of next-to-leading and higher order hard photon contributions to be included (A,D,E,F). F is the *recommended* choice;

<u>OAL</u>: selects the value of $\alpha_{em}^{-1}(M_Z^2)$: 128.896 (Y), 128.87 (N), user-defined (V); if OAL = 'V' then a new value for $\alpha_{em}^{-1}(M_Z^2)$ is selected through the calling procedure `CALL TCAQED(ALPHANEW)`.

<u>OAAS</u>: the scale in α_s in mixed $\mathcal{O}(\alpha\alpha_s)$ corrections;

<u>ONP</u>: array dimensioned as ONPT(NRTS) selecting (Y) or not (N) the pair production correction for each energy;

<u>ONIF</u>: array dimensioned as ONIF(NRTS) selecting (Y) or not (N) the effect of initial-finale state QED interference for each energy;

<u>OEXT</u>: selects the extrapolated (E) or cut (C) branch for μ and τ. With OEXT = 'P' only pseudo-observables are computed and control is returned;

<u>OFB</u>: selects (Y) or not (N) different treatment of cuts for cross sections and forward-backward asymmetries for μ and τ;

<u>OCUTS</u>: array dimensioned as OCUTST(NRTS); if OFB = Y selects (Y) or not (N) a cut on the invariant mass after initial-state radiation for μ and τ cross sections;

<u>OCHAN</u>: selects the full Bhabha scattering matrix element (F) (*default*) or only the s-channel part (S);

<u>OCUT</u>: array dimensioned as OCUT(NRTS); selects a cut on s': no cut (NC), cut on hadronic channels only (HC), cut on all channels (FC). Here s' is the invariant mass of the e^+e^- system after initial-state radiation;

<u>OCUTF</u>: as OCUT, but for final-state fermions invariant mass;

<u>OTHRMT</u>: if OEXT = C, selects invariant mass (M) or energy (E) threshold for μ and τ;

OCREE: if OEXT = E and OCHAN = S, i.e. for s-channel electrons, it allows for an s' cut;
OCUTES: array dimensioned as OCUTES(NRTS); if OCREE = Y, it allows (C) or not (N) for an s' cut for each energy point;
OTHRE: selects invariant mass (M) or energy (E) threshold for electrons;
OFS: selects the treatment of higher order final-state QED corrections; *default* (D) or perturbative (Z);
OCN: includes (Y) or not (N) the contribution of electromagnetic jets for final-state calorimetric electrons;
XMED: fictitious separator for 1-dimensional integrations to improve numerical convergence;
OVNAL: if OAL = V, enters the user-defined value $\alpha_{em}^{-1}(M_Z^2)$ = OVNAL;
SC: if OAAS = Y, defines the scale in α_s in mixed $\alpha\alpha_s$ corrections;
ZPCUT: the minimum fraction of squared invariant mass of the final state after radiation of the additional initial-state pair;
OXCUTS: if OCUTS = Y, it enters the value of the s' cut, s_0/s;
OXCUT: if OCUT = Y, it enters the value of the s' cut, s_0/s;
OXCUTF: if OCUTF = Y, it enters the cut on the value of invariant mass squared of the final-state fermions;
SOCUT: array dimensioned as SOCUT(3); SOCUT(1) is the minimum invariant mass (GeV) of the final-state electrons, used if OTHRE = M; SOCUT(2) and SOCUT(3) are the same as SOCUT(1) for μ and τ respectively, used if OTHRMT = M;
E0: array dimensioned as E0(3); E0(1) is the minimum energy (GeV) of the final-state electrons, used if OTHRE = E; E0(2) and E0(3) are the same as E0(1) for μ and τ respectively, used if OTHRMT = E;
THMIN: array dimensioned as THMIN(3); it enters the minimum scattering angle (deg) for electrons, μ and τ, respectively (*symmetrical angular acceptance* is understood); THMIN(2) and THMIN(3) used only if OEXT = C;
THMINP: the same as THMIN, for the antiparticles;
ACOLL: array dimensioned as ACOLL(3); it enters the maximum acollinearity angle (deg) for electrons, μ and τ, respectively; ACOLL(2) and ACOLL(3) used only if OEXT = C;
OXCUTES: if OCUTES(I) = C, it enters the s_0/s threshold for the I-th energy point;
DEL: the semi-aperture of the electromagnetic jet (deg); used only if OCN = 'Y'.

Changes in the new version are:

- SUBROUTINE TOPAZO(SE,NRTS,RTS,ZMT,TQMT,HMT,ALST,OTPPO,OTPRO).

The scaling factor for numerical integration error is now argument of the subroutine. This is important, since the needed CPU time is largely due to the choice of `SE`. Recommended is `SE = 1.D0`, especially when one computes all quantities in one stroke, but on a single-channel calculation it is feasible (and somewhat recommended) to reach even better precision, as with `SE = 0.1D0`. Remember that, far away from the Z-resonance al- l calculations are longer, much longer. For fitting procedures it might be advisable a lower precision, something like `SE = 10.D0` or even high- er. One should remember that for extrapolated observables there is a one-dimensional integral so we start with a requested relative error `REQ = 1.D-5*SE`. For observables with cuts we go up to three-dimensional integrals so we request `REQ = 1.D-4*SE` or even `REQ = 1.D-3*SE` (for 3-dim).

- The main branch in TOPAZ0 is governed by the flag `OEXT`: selects the extrapolated `(E)` or cut `(C)` branch for μ and τ. It is understood that `OEXT = 'P'` will be used only for pseudo observables. Now we have 3 additional flags. They are: `OBHABHA, OREST, OINDX`. The philosophy is as follows.

$$\boxed{\texttt{OEXT = 'E'}} \quad \rightarrow \quad \boxed{\texttt{OBHABHA = 'N'.or.'B'.or.'R'}} \tag{1}$$

1. `OBHABHA = 'N'` (default) everything is as before, i.e. Bhabha with cuts and the rest extrapolated with a, possible, s'-cut.

2. `OBHABHA = 'B'` Only Bhabha scattering (cross-section and forward-backward asymmetry) is computed.

3. `OBHABHA = 'R'` In this case `OREST = 'E'` and Bhabha scattering is not computed. One has the following options:

   a) `OINDX = 'MU'` Only the cross-section and the forward-backward asymmetry for muons (s'-cut) are computed.

   b) `OINDX = 'TA'` Only the cross-section and the forward-backward asymmetry for taus (s'-cut) are computed.

   c) `OINDX = 'BQ'` Only the cross-section and the forward-backward asymmetry for b-quarks (s'-cut) are computed.

   d) `OINDX = 'CQ'` Only the cross-section and the forward-backward asymmetry for c-quarks (s'-cut) are computed.

   e) `OINDX = 'HA'` Only the total cross-section for hadrons (s'-cut) is computed.

$$\boxed{\texttt{OEXT = 'C'}} \rightarrow \boxed{\texttt{OBHABHA = 'N'}} \tag{2}$$

Again, everything as before. But

$$\boxed{\texttt{OEXT = 'C'}} \rightarrow \boxed{\texttt{OBHABHA = 'R'}} \rightarrow \boxed{\texttt{OREST = 'C'}} \tag{3}$$

In this case `OINDX` can only assume two values `'MU'.or.'TA'` and one computes the cross-section and the forward-backward asymmetry for muons or taus with kinematical cuts. Now few examples:

```
OEXT= 'E'
CALL TINIT(NRS,OEXT)
OVAL= 'B'
OF7= 'OBHABHA'
CALL TCFLAG(OF7,OVAL)
```

will give you only Bhabha.

```
OEXT= 'E'
CALL TINIT(NRS,OEXT)
OVAL= 'R'
OF7= 'OBHABHA'
CALL TCFLAG(OF7,OVAL)
OV2= 'MU'
OF5= 'OINDX'
CALL TCFLAG(OF5,OV2)
```

Will give you only extrapolated muons.

```
OEXT= 'C'
CALL TINIT(NRS,OEXT)
OVAL= 'R'
OF7= 'OBHABHA'
CALL TCFLAG(OF7,OVAL)
OV2= 'MU'
OF5= 'OINDX'
CALL TCFLAG(OF5,OV2)
OVAL= 'C'
OF5= 'OREST'
CALL TCFLAG(OF5,OVAL)
```

Will give you only muons with kinematical cuts. Here I want to remind you how to change cuts. For instance, the following calling sequence

```
*    CUTS FOR BHABHA ARE CHANGED  (FOR MUONS 2, FOR TAUS 3)
      IND= 1
*    THIS WILL SET THE ANGLE FOR E^-
      THMN= 50.D0
*    THIS WILL SET THE ANGLE FOR E^+
      THMNP= 50.D0
*    THIS WILL SET ACOLLINEARITY
      ACLL= 30.D0
*    THIS WILL SET ENERGY THRESHOLD
      E0C= 1.D0
*    THE REST IS UNCHANGED
      DMY= -1.D0
*    CHANGE IS PERFORMED
      CALL TCUTSET(IND,DMY,THMN,THMNP,ACLL,E0C,DMY,DMY,DMY)
```

is used to change angles, acollinearity and energy thresholds. Alternatively

```
      IND= 1
      DMY= -1.D0
      XCUT= 0.8D0
      CALL TCUTSET(IND,DMY,DMY,DMY,DMY,DMY,XCUT,DMY,DMY)
```

will set the s'/s cut to 0.8 for all extrapolated quantities. More,

```
      IND= 1
      DMY= -1.D0
      XPCUT= 0.8D0
      CALL TCUTSET(IND,DMY,DMY,DMY,DMY,DMY,DMY,DMY,XPCUT)
```

will set the $z_{\rm min}$ cut for initial pair production to $z_{\rm min} = 0.8$. Etc, etc, etc.

Some example concerning CPU. Bhabha alone for 3 energies. For `SE = 1.D0` I get

```
 E_CM (GEV)            =          0.88450E+02

 SIGMA(E)   (NB)       =        0.3889544E+00 +/-        0.1525387E-04
 A_FB(E)               =        0.3899105E+00 +/-        0.5447768E-04

 E_CM (GEV)            =          0.91187E+02

 SIGMA(E)   (NB)       =        0.1064183E+01 +/-        0.1646520E-04
 A_FB(E)               =        0.1157290E+00 +/-        0.1723651E-04
```

```
E_CM (GEV)             =          0.93700E+02

SIGMA(E)   (NB)        =        0.3125899E+00 +/-        0.2235025E-04
A_FB(E)                =        0.1416559E+00 +/-        0.8131592E-04

 Charged CPU time:          0 00:00:53.32   Elapsed time:     0 00:00:56.39
```

while for `SE = 0.1D0` I get

```
E_CM (GEV)             =          0.88450E+02

SIGMA(E)   (NB)        =        0.3889543E+00 +/-        0.1901593E-05
A_FB(E)                =        0.3899121E+00 +/-        0.6791394E-05

E_CM (GEV)             =          0.91187E+02

SIGMA(E)   (NB)        =        0.1064184E+01 +/-        0.1926033E-05
A_FB(E)                =        0.1157299E+00 +/-        0.2011631E-05

E_CM (GEV)             =          0.93700E+02

SIGMA(E)   (NB)        =        0.3125812E+00 +/-        0.2282713E-05
A_FB(E)                =        0.1416393E+00 +/-        0.8307268E-05

 Charged CPU time:          0 00:07:20.19   Elapsed time:     0 00:07:36.75
```

It is up to the user to spend so much time for a smaller error. By the way, for Bhabha + rest extrapolated, at `SE = 1.D0`, one would need

```
 Charged CPU time:          0 00:05:40.19   Elapsed time:     0 00:05:53.71
```

So, Bhabha lovers can have it 6.4 times faster if they do not care about the rest. Another example: for muons, with cuts we go from `SE = 1.D0`

```
E_CM (GEV)             =          0.88450E+02

SIGMA(MU)  (NB)        =        0.2449078E+00 +/-        0.1956177E-04
A_FB(MU)               =       -0.2557341E+00 +/-        0.9986490E-04

E_CM (GEV)             =          0.91187E+02
```

```
 SIGMA(MU)  (NB)        =        0.1410461E+01 +/-        0.2350701E-04
 A_FB(MU)               =        0.9383254E-04 +/-        0.1665564E-04

 E_CM (GEV)             =          0.93700E+02

 SIGMA(MU)  (NB)        =        0.4703119E+00 +/-        0.2831399E-04
 A_FB(MU)               =        0.1405728E+00 +/-        0.6796475E-04

  Charged CPU time:          0 00:00:11.00   Elapsed time:     0 00:00:13.54
```

To SE = 0.1D0

```
 E_CM (GEV)             =          0.88450E+02

 SIGMA(MU)  (NB)        =        0.2449087E+00 +/-        0.1985238E-05
 A_FB(MU)               =       -0.2557254E+00 +/-        0.1015236E-04

 E_CM (GEV)             =          0.91187E+02

 SIGMA(MU)  (NB)        =        0.1410460E+01 +/-        0.2703623E-05
 A_FB(MU)               =        0.9437355E-04 +/-        0.1901329E-05

 E_CM (GEV)             =          0.93700E+02

 SIGMA(MU)  (NB)        =        0.4703196E+00 +/-        0.3949544E-05
 A_FB(MU)               =        0.1405793E+00 +/-        0.9536350E-05

  Charged CPU time:          0 00:00:55.55   Elapsed time:     0 00:00:59.15
```

So that muons with cuts can be obtained, even with high precision, without having to compute, notoriously time consuming Bhabha.