User Manual

April, 2000

FDC

Jian-Xiong Wang

Theory Group, Institute of High Energy Physics, Academia

P. O. Box 918, Beijing 100039, China

Version 2.0

Contents

1 Introduction

1.0.1 Main Parts in FDC

In this project, we have the following items:

1 Build the physics model, i.e. construct the Lagrangian and quantize it.

2 Generation of all Feynman diagrams and amplitudes for a given process.

3 Analytic calculation of amplitudes of these diagrams and generation of the expression of the total squared amplitude

1.0.2 Computer language used

```
     REDUCE, RLISP     most part of the system use REDUCE and RLISP.
     FORTRAN           fortran source generated to do numerical calculation.
     C++, X-window     user interface and picture drawing.
     PS                PS file are generated to draw and print Feynman diagram.
> cd fdchome
> ls
psdraw  ------ C++ source for draw Feynman diagrams in PS file.
draw    ------ C++ source for draw Feynman diagrams in X-window.
fasl/   ------ compilered source of files in src.
f77/   ------  some prepared fortran source of the system
log/    ------ recording files of compiling of each source file.
mode/   ------ Physical Models.
bin/    ------ excutable files of each part of FDC.
src/    ------ all the REDUCE source of FDC.
util/   ------ all the files used to install or update the FDC.
xfdc    ------ C++ source for integration FDC system.

> cd src
> ls
fdc        -----  procedures for evaluation of tree and loop diagrams.
fdcad      -----  some procedures for loop momentum integration.
tlib5      -----  procedures for Feynaman diagram generation.
tm2        -----  some procedure for supper symmetry.
tmode1     -----  procedures for Feynman rules deduction
wtest      -----  procedures for simplification of Feynman rules.

> cd fasl
> ls
fdc.b fdcad.b tlib5.b tm2.b tmode1.b wtest.b

> cd mode
> ls
mode01     ---- Standard Model in R-ksi gauge.
mode02     ---- Standard Model in unitary gauge.
```

```
mode03     ---- Two Higgs Doublet Model in R-ksi gauge.
mode04     ---- Two Higgs Doublet Model in unitary gauge.

> cd bin
> ls

gmodel1*  ----- Generation matter fileds interaction (include Higgs
                self-interaction and Yukawa coupling ) for the input
                physical mode.

gmodel2*   ----- Generation and Quantization of the Lagrangian of the model.

gmodel3*   ----- Simplification of the deduced Feynman rules.

diag*     ----- Generation Feynman diagrams for a given process in the model.

amp*     ----- Evaluation of the Feynman Amplitudes and generation of
                Fortran source for the process.

kine*   -----  to generate kinematics and parameters FORTRAN source and
               makefile

psdraw*   ----- draw Feynman diagrams in PS file

idraw*    ----- draw Feynman diagrams in PS file

draw*     ----- draw Feynman Diagrams in X-window.

XFDC*      ----- Integrated FDC system on X-window.
```

2 Physical Model

First of all, the physical model must be built in. It can be done by just introducing all the Feynman rules, counterterms and constant obtained from a handbook of the physical model. This is not a so hard task. But it is very unconvenient for a user to inprt new gauge or new model, such as SU(5) or suppersymmetry models. When a physical model is constructed, to quantize it and give all Feynman rules and counterterms by hand needs quit a lot of work, through there is a standard procedure to do it. For this propose, we have constructed a package FDC1. It makes the input of the Physical model easy. The folling is the input date for Standard Model. The user do not needed to worry about ghost fields, which will appear when the gauge fields are quantized. In the input it is easy to change the gauge fixing terms to another gauge. The output of this part is the list of all Feynman rules and the expression of counterterms which depends on the scheme(choice of a set independent parameters as input quantities). The output are of two types, one is built in the systemas the physical model. Another appeares in REDUCE algebraic mode, shich is easy to be understood by user.

The user can pass the next two sections if he just want to use the existed models.

2.1 To add new model

One can built a new model in the following way: to copy all the input files of a model existed by:

```
>model_cp example_model new_model
```

where the "example_model" is the directory of the existed model and the "new_model" is the directory of the new model. the command "model_cp" copy the following input files

```
model.def
model_input
add_vertices
physical_parameters
sybmole_lsit
```

from the "example_model" to "new_model". Then one have to edit these files and change something to suit for his model. The detail of these input file will be explained in the following section.

There are four steps to build the model.

```
>cd new_model
>gmodel1
>gmodel2
>gmodel3
>glmodel
```

For the first step, the following is the list of the input and output files.

Input file name:	description
model.def	basic information of the model
model_input	description of physical model
add_vertices	phenemology vertices
Output file name:	
matterintinput	all possible Yukawa coupling terms
vertices_drop_list	about which vertices are droped
feynman_rule_show	list of all vertices in the model
system/midmodel	lagrangian of the model
model_state	the state of the model

To run the first step.

```
> gmodel1
```

Then the user should do the following things:

- Check the running recording file "out" to see if there are some thing wrong
- Check the file "model_state" to see if there is the line

```
model_generator1:=ok;
```

Otherwise, the user have to check the three input files to find out what is wrong.

- Edit the file "matterintinput" to choose the Yukawa coupling terms (such as in two Higgs doublet model).
- View the file "feynman_rule_show" to find the vertices to drop and add them to the vertices_drop_list in the file "vertices_drop_list"

For the second step, the following is the list of the input and output files.

Input file name:	description
add_vertices	output file of step 1
midmodel	output file of step 1
matterintinput	output file of step 1
vertices_drop_list	output file of step 1
Output file name:	description
particle_table	list of all particles in the model
feynman_rule_show	list of all vertices in the model
model_state	the state of the model
system/Feynman_rules	Feynman rules for FDC system using
system/model	model for FDC system using
system/particle_table	paticle list for FDC system using

For the third step, the following is the list of the input and output files.

Input file name:	description
model.def	
midmodel	output file of step 1
particle_table	output file of step 1
physical_parameters	physical parameters
symbole_list	latex notation of each particle
Output file name:	
model.tex	latex file about the detail of the model

To run the second and third steps.

```
> gmodel2
> gmodel3
```

Then the user should do the following things:

- Check the running recording file "out" to see if there are some thing wrong
- Check the file "model_state" to see if there is the line

```
model_generator1:=ok;
model_generator2:=ok;
model_generator3:=ok;
```

Otherwise, the user have to check the input files to find out what is wrong.

The fourth step is to built latex and ps file from the latex file of the model.

```
> glmodel
> lamodel
```

2.1.1 Input files

The following is the input files for Standard Model.

```
%————————— File: model.def ——————————————
%Please revise the following items when a new model is created.
model_home:=""/home/fdc/fdc96/model/smu/";
model_name:=""SM in UG";
model_author:=""Jian-Xiong Wang";
model_time:=""Oct 13, 1994";
;end;
```

```
\% ----------------------- File: mode_input -------------------- \\
\%                  The standard model in unitary gauge        \\

algebraic$
%------------------------ Gauge fields -------------------------------
% The first is the gauge field. So in the mode, you must input
% how many gauge fields and for each gauge field, declare SU(n) ( we
% have just realize SU(n) gauge field in the system), notation for the
% field, goupling constant, gauge fix term  and gauge parameter ( different
% gauge can be chosen by differnet gauge fix term and different gauge
% parameter), and you should indicate if the gauge symeetry is breaking
% or not.

gaugefields:=
%     No.  Name  SU(n)   notation  coupling   gaugefixterm gauge_pa breaking
   '((1    u     1       b         g1        gfix1        infinite   yes    )
     (2    su    2       a         g         gfix2        infinite   yes     )
     (3    su    3       gs        g3        gfix3        1         no     )
    )$
gfix1:=pd(b(v),v)$
gfix2:=pd(a(i1,v),v)$
gfix3:=pd(gs(i1,v),v)$
```

```
% where b is the u(1) field, a is the su(2) field and gs is the su(3) field.
%-------------------------------------------------------------------

%----------------------- Matter fields -------------------------------
% The second is the matter field. for each matter field, you must input
% the name,  2 times spin, chiral ( r --- right, l --- left or r-l
% together) and then the gauge group parameter for each gauge group
% ( for u(1), it is u(1) charge and for su(n), it is mutiplet number)
%
matterinput:={{name,  2*spin,  chiral, "|g",  1,     2,    3},
              { hig,   0  ,    rl  ,          1,     2,    0},
              { el ,   1,      l  ,          -1,     2,    0},
              { er ,   1,      r  ,          -2,     0,    0},
              { q1l ,   1,      l  ,          1/3,    2,    3},
              { q1dr,   1,      r  ,          -2/3,   0,    3},
              { q1ur,   1,      r  ,          4/3,    0,    3}
              }$
% where:
%  hig is the Higgs boson doublet with u(1) charge 1 and su(3) singelet.
%  el is the electron left-hand doublet with u(1) charge -1 and su(3) singelet.
%  er is the electron right-hand singelet with u(1) charge -2 and su(3)
%  singelet.
%  q1l is the first generation quark left-hand doublet with u(1) charge
%  1/3 and su(3) triplet.
%  q1dr is the first generation down quark right-hand singelet with u(1)
%  charge -2/3 and su(3) triplet.
%  q1ur is the first generation up quark right-hand singelet with u(1)
%  charge 4/3  and su(3) triplet.
%-------------------------------------------------------------------


%----------------------- Charge list of particles ----------------------
% In fact, The charge of particles can be calculated out by using the
% definition of charge and the definition of all the fields.

pchalist:={{hig,h0(0)},              %hig field's componet h0 is of charge 0.
           {el,nue(0),ee(-1)},       %el field's componets nue -- 0, ee -- -1.
           {er,ee(-1)},              %er field's componet ee is of charge -1.
           {q1l,qu(2/3),qd(-1/3)},   %q1l's componets qu -- 2/3, qd -- -1/3.
           {q1ur, qu(2/3)},          %q1ur's componet qu -- 2/3.
           {q1dr, qd(-1/3)},         %q1dr's componet qd -- -1/3.
           {a,w(1),w(-1),z(0),p(0)},%a's componets w -- 1, z -- 0, p --0.
           {b,p(0),z(0)},            %a's componets z -- 0, p --0.
           {gs,gs(0)}                %gs's componet gs -- 0.
          };
```

```
%--------------------------------------------------------------------



%------------------- Definition of mutiplet -------------------------
%
% To definition the mutiplet is not a trivial thing in general case,
% the definition of vector mutiplets can be done in the following way.
% the definition of tensor mutiplet are realized in SU(5) gauge theory
% case.
%
%         name  n_group   1_componet 2_componet   ...
 mdefl:={{hig, {2,        0   ,   (v0+h0)/2**0.5}},
         {el , {2,        nue   ,   ee}},
         {er , {2,        ee           }},
         {q1l , {2,       qu   ,  qd}},
         {q1dr, {2,       qd          }},
         {q1ur, {2,       qu            }},
         {qu,  {3,        qu(1),   qu(2) ,  qu(3)}},
         {qd,  {3,        qd(1)  , qd(2) ,  qd(3)}}
        }$

vancumexpectation:='((hig v0));

% where hig is the Higgs doublet for the second gauge group su(2)
% and the definition of it's componets are given in unitary gauge.
% the vancume expextation of hig is v0 so that in the 2_componet
% expression of hig there is only one higgs field h0.
% el, q1l are electron and first generation quark left-hand double for
% the second gauge field su(2).
% er, q1dr and q1ur are electron, first generation down quark and up quark
% right-hand singelet for the second gauge field su(2).
% the qu, qd which appeared in q1l,q1dr,q1ur componets are triplet for
% the third guage field su(3).
%--------------------------------------------------------------------

%------------------- Fermion mix -------------------------
%
% Definition of Fermion Generation:
% Generation is a very useful concept in physical model. We can construct
% the lagrangian for only one generation, then add almost the same terms
% for the other generation and make some rotations which are caused by
% Yukawa coupling or Higgs self-interaction terms. In the following, we
% just introduce the definition for fermion generations.
% There are three generation quark and lepton in the SM as shown in the
```

```
% following

realfamily:='((q1l q2l q3l) (q1dr q2dr q3dr) (q1ur q2ur q3ur)
              (qu qc qt) (qd qs qb)
              (el mul taul) (er mur taur)
              (ee mu tau) (nue numu nut)
              );

% where (q1l q2l q3l) is the three generations of left-hand double quark,
% (q1dr q2dr q3dr) is the three generations of right-hand singlet down quark,
% (q1ur q2ur q3ur) is the three generations of right-hand singlet up quark.
% (qu qc qt) and (qd qs qb) are the coressponding componets for the three
% generation quarks.
% where (el mul taul) is the three generations of left-hand double lepton,
% (er mur taur) is the three generations of right-hand singlet lepton,
% (ee mu tau) and (nue numu nut) are the coressponding componets for the
% three generation leptons.

% Mix Matrices:

mixlist:='(
            (l (s1 (qd qs qb)) (s2 (qu qc qt))
               (se1 (ee mu tau)) (se1 (nue numu nut))
            )
            (r (t1 (qd qs qb)) (t2 (qu qc qt))
               (te1 (ee mu tau))
            )
          );

% Definition of CKM Matrix: ]

mixlistcondition:={s2(-1)*s1(1)=>v_ckm(1)};

% where the mix matrix for the three left-hand generations are:
%   mix matrix s1 for (qd qs qb),  mix matrix s2 for (qu qc qt),
%   mix matrix se1 for (ee mu tau) and (nue numu nut).
%   The reason that the mix matrix of (nue numu nut) can be chosen as
%   the smae mix matrix of ( ee mu tau) is due to that nue, numu, nut
%   have the same mass so they can be mixed in any way. The choice of
%   the same mix matrix se1 for (ee mu tau) and (nue numu nut) make
%   representation in the simplist way. For eaxmple, If you have a model
%   with the neutrons (nue numu nut) are of different masses, you must
%   choose different mix matrices for leptons and neutrons and as a result
%   that you need to definition CKM matrix for leptons in the above
%   mixlistcondition.
% The mix matrix for the three right-hand generations are:
```

```
%   mix matrix t1 for (qd qs qb), t2 for (qu qc qt)
%   and te1 for (ee mu tau).
%-----------------------------------------------------------------------


%------------------------- Globel Symmetry -----------------------------
%
% The third is the globel symmetry of the model, there are lepton number
% conservation as globel symmetry in Standrad Model and the lepton numbers
% are set to each particles in the model. gbsymmetrylist is the globel
% symmetry list.

gbsymmetrylist:={len,lmuon,ltau};
gbsymmetrydata:={{len,  {el,1},   {er,1}},
                 {lmuon,{mul,1},  {mur,1}},
                 {ltau, {taul,1}, {taur,1}}}$

% where globel symmetry number len are 1 for el and 1 for er,
% globel symmetry number lmuon are 1 for mul and 1 for mur,
% globel symmetry number ltau are 1 for taul and 1 for taur.
% In the system, it is supposed that the lepton number of particles
% which do not appeared in each lepton number table in the gbsymmetrydata
% are set to zero and all the globel symmetry numbers for anti-particles
% are set to the minus times that of the particles.
%-----------------------------------------------------------------------

%-------------------- Masses and Other Paramters ------------------------
% The names of masses of all the particles are given in the following
% table phymass and all the zero mass particles are also set it's mass
% to zero in the table.

phymass:={{w, wm}, {z, zm}, {p, 0}, {h0, hm},
          {ee, fme},{nue, 0},
          {mu, fmmu},{numu, 0},
          {tau, fmtau},{nut, 0},
          {qd, fmd}, {qu, fmu},
          {qs, fms}, {qc, fmc},
          {qb, fmb}, {qt, fmt}
          }$

% The parameter set are chosen in table phyinput and this set is used
% as physical input of the model, or it can be called as the choice of scheme.
% To change scheme, your can choose different physical input in the table.
% For eaxmple, to use Z boson's mass zm instead of the w boson's mass wm,
% to use electronic coupling g1 instead of sin(thta), for this propose,
% the relation of g1,g and thta in the paramters relation table construles
```

```
% should be changed.

phyinput:={wm, hm, g, sin(thta),
           fme,
           fmmu,
           fmtau,
           fmd, fmu,
           fms, fmc,
           fmb, fmt
          }$


% -----------------  Mix of vector boson -----------------------
%

mixrules:={ a(1,~v) => (w(1,v)+w(-1,v))/2**0.5,
            a(2,~v) => (w(-1,v)-w(1,v))/2**0.5/i,
            a(3,~v) => z(0,v),
            b(~v)   => p(0,v) };

bosonmix:='((sxpz p z));
%-------------------------------------------------------------------

% -----------------  Mix of Higgs boson -----------------------
% The mix of Higgs boson will appeared in the model with more than
% one Higgs mutiplet. For eaxmple, in the two Higgs doublet model.
%-------------------------------------------------------------------

% ------------  Mix of Higgs boson and vector boson -------------
% The mix of Higgs boson and vector boson is required to be cancelled
% by some terms appeared in the gauge fix terms and the system will
% automatically add these terms to the gauge fix term given in above
% gauge fields section.
%-------------------------------------------------------------------

% --------- Mix Matrices of Higgs boson and vector boson---------

matrixlist:={ {sxpz,{cos(thta),sin(thta)},{-sin(thta),cos(thta)}}
            };

%where matrixlist is the list of all the boson mix matrices. the
%mix angle thta of photon and Z boson is introduce the it's mix
%matrix expression.
%-------------------------------------------------------------------

% --------- To solve parameter relations---------
```

```
% it is obvious that not all the parameters appeared in the model are
% indepandent and some of them should be expressed by others. In the
% following, there are two way to choose which parameters should be
% expressed by others. One is to put the expressions of the parameters
% into the list construles and another is to put the parameters in the
% list mixfix, then the system will give the expression when it deal the
% mix of particles.

mixfix:={};
construles:={ g1=>g*sin(thta)/cos(thta)
            };

% The following is the another way.
%mixfix:={g1};
%construles:={ };

%
%-------------------------------------------------------------------------

%---------------------- The charge definition --------------------------

charge:=tp(2,3)+tp(1,1)/2;

%where tp(n,m) represent the m-th generator of the n-th gauge group
;end$
%%-----------------------------------------------------------------------
```

2.1.2 Output files

2.2 Standard Model and Extensions

2.3 SM in UG and some vertices were droped

This model is in then directory "$fdc/model/smu1". and the basic information file of the model "model.def" is the following:

```
%Please revise the following items when a new model is created.
model_home:=""/home/fdc/fdc96/model/smu1/";
model_name:=""SM in UG and some vertices were droped";
model_author:=""Jian-Xiong Wang";
model_time:=""Oct 13, 1994";
;end;
```

The model input file "model_input" is described in detailed in the following

This is Standard Model in unitary gauge, include electro-weak interaction and QCD, Quark mixing terms are droped. Higgs particle couple to electron, muon, u-quark, d-quark are droped. It

is inputed by Jian-Xiong Wang, on Oct 13, 1994.

There are totally 18 particles, in which 12 are Fermions, 4 are Vector Bosons, 2 are Scalar Bosons , g_g is ghost particles.

There are totally 63 vertices in this model.

2.3.1 SM in UG and J/ψ,J/ψ_8,B_c

This model is in then directory "$fdc/model/smu2". and the basic information file of the model "model.def" is the following:

```
%Please revise the following items when a new model is created.
model_home:=""/home/fdc/fdc96/model/smu2/";
model_name:=""SM in UG and J/ψ,J/ψ8,Bc";
model_author:=""Jian-Xiong Wang";
model_time:=""Nov 16, 1996";
;end;
```

This is Standard Model in unitary gauge, include electro-weak interaction and QCD, Quark mixing terms are droped. Higgs particle couple to electron, muon, u-quark, d-quark was droped.

In this model, We have added effective vertices, $J/\psi c\overline{c}$, $J/\psi_8 c\overline{c}$, $B_c b\overline{c}$, to do some calculation related to these bound state It is inputed by Jian-Xiong Wang, on Nov 16, 1996.

There are totally 18 particles, in which 12 are Fermions, 4 are Vector Bosons, 2 are Scalar Bosons , g_g is ghost particles.

There are totally 67 vertices in this model.

2.3.2 SM in UG and Some Phenomenology Lagrangian

This model is in then directory "$fdc/model/smu_a1". and the basic information file of the model "model.def" is the following:

```
%Please revise the following items when a new model is created.
model_home:=""/home/fdc/fdc96/model/smu_a1/";
model_name:=""SM and Some Phenomenology Lagrangian";
model_author:=""Jian-Xiong Wang";
model_time:=""Nov 16, 1996";
;end;
```

This is Standard Model in unitary gauge, include electro-weak interaction and without QCD, Quark mixing terms are droped. There are two phenomenogly terms which represent some effects to V_{tb} due to some new physics

$$c_1 g\overline{\psi_{q3l}}\gamma_\nu F^{\nu,\mu}(a)\partial_v\psi_{q3l} + ic_2 g^2\overline{\psi_{q3l}}\gamma_\nu\tau_a\psi_{q3l}\overline{h}\tau_a\partial_v h;$$

where a is the $U(1)$ filed, q3l is the $SU(2)$ double of the third generation quark and h is the Higgs double. Higgs particle couple to electron, muon, u-quark, d-quark was droped. It is inputed by

Jian-Xiong Wang, on Dec 12, 1997.

There are totally 18 particles, in which 12 are Fermions, 4 are Vector Bosons, 2 are Scalar Bosons , g_g is ghost particles.

There are totally 172 vertices in this model.

2.4 Minimal Supper Symmetry Standard Model and Extensions

2.5 Phenomnology Model

3 Physical Process

There are two ways in FDC to calculate physical process. One is to prepare input and calculate a process, which is descriped in section 3.1. Another is to prepare input under some topic and then generate a series of processes, which is explained in section 3.4.

3.1 To calculate a process

At first, we must prepare a directory for the process and then prepare a input files "process.def", and create subdirectory "fort" and "diagram" in the directory.

Let's start from an example $e^+e^- \to w^+w^-H$, which is prepared in the the fdc system directory "$fdc/example". We can prepare the inputs by perform the following command.

> process_cp $fdc/example/p1 p

where the command "process_cp" is in the directory "$fdc/bin" and it creates the directory "p", copy input files "process.def" in "$fdc/example/p1" into "p", and create the subdirectory "fort" and "diagram" in "p"

The following is the detailed description of the input file "process.def":

```
model_home:='"/home/fdc/fdc96/model/smu1/"$   % The model
process_name:='"efb ef —>w h0 wb "$           % The name
process_author:='"Jian-Xiong Wang"$           % The user
process_time:='"Aug 13, 1997"$                % The date
namel:='(ef efb wb h0 w)$                     % The process
inpl:='(1 1 -1 -1 -1)$                        % Initial and Final
ncorrection:='((g 0) (g3 0))$                 % Tree process
squart_root_s=200                             % center mass energy
```

where "namel:=...." is the list of initial and final particles and the particle name notation "ef" (electron), "efb" (position), "wb" (w^+), w (w^-) and h0 (Higgs) are defined in the in the file "particle-table" in the home directory of choose physical model. "inpl:=..." is used to distinguish the initial and final particles, in which 1 means the initial particle and -1 mean final particle. "ncorrection:='((g 0) (g3 0))$" is used to give the perturbation order, in which g and g3 mean electro-weak coupling constant and strong interaction constant respectively. "(g 0)" means leading order of "g" and "(g3 0)" means leading order of "g3". for example, if there is "(g 1)", it means beyond the leading order of "g", it will calculate the next leading order diagrams of "g".

Now it is easy to perform the following steps to finished the calculation.

```
> cd p
> diag      % to generator the feynman diagrams
> amp       % to manipulate the amplitude and
            %generate FORTRAN source of amplitude.
> kine      % to generate kinematics FORTRAN source.
> cd fort   % all fortran source are placed in the
            % subdirectory fort.
> make      % to compile and link of the FORTRAN
            % source.
> int       % To do numerical integration
```

There are three output files in the subdirectory "fort", which are explained in the following:

fresult.dat is the final result

convergence.dat is the convergence behave of the integration

plot.dat contains all the distributions which the user asked

There are a few output files in subdirectory "diagram", in which, aa0a1, aa0a2, ... are the files contains all the Feynman amplitude data, diag1.ps, diag2.ps .... are the PS files of all the Feynman diagrams.

The example is used to calculate the total cross section with the default set in FDC system. However, there are many options could be set in the input file "process.def" by user, which are explained in section "Options for user". there are also many extra usages could be added in the input file, which are described in the section "Advanced usages".

The user can follow the same procedure as above to calculate a process. But he need to make some change in the input file according to his own need.

3.2 Options for user

The options are used to make different choose. Their default values are seted in the fdc system file "$fdc/src/option_s". User can change them by set their value to other possible value in the input file "process.def". In the following, all these options will be explained.

3.2.1 amp_method

The option is used to choose the method to calculate the amplitude square. The default method is: It use amplitude square alantica method[??1]] (amp_method:='amp$) in the case of less than four final state particles case, and to use numerical amplitude method[??2]] (amp_method:='tree$) in other case. The user can choose the method by adding the following line to input file "process.def".

```
amp_method:='tree$, or amp
```

3.2.2 addwidth

The option is used to controle the adding width of unstable particles in the amplitude calculation. The default choose is to add width. The user can choose not to add width by set the following value:

```
addwidth:='no$
```

It is obviously that to add width will break the gauge invariance in the calculation of amplitude. It will be a higher order breaking if there is no t-channel photon connected to a fermion line with small mass. Otherwise, the breaking is serious. To deal with this serious breaking, we introduce a method and controle it by the next option.

3.2.3 no_change_to_t_photon

The option is used to controle the way to deal with cancellation due to gauge invarince in the case of t-channel photon connected to fermion with small mass[??3]]. The default choose is to do the projection[??3] to the fermion line. The user can choose not to do any projection to the fermion line by set it to:

```
no_change_to_t_photon:='ok;
```

In the case of gauge invariance keeping, there will be the same reuslts from the different choose, but in the case of gauge invariance breaking by the width of unstable particles in the propagator formular, The result of no projection is wrong while the result with the projection is the right one.

3.2.4 no_change_to_s_photon

The option is similar to the above one but in the case of s-channel photon connected to fermion with small mass. the default choose is to do no projection, and the user can choose to do with projection by adding the following line to the input file "option".

```
no_change_to_s_photon:='no;
```

Where the two method will result in almost the same result.

3.2.5 To choose diagrams in amplitude calculation

The default is to choose all the Feynman diagrams in the amplitude calculation. To choose a few diagrams in the amplitude calculation, the user need to add the following line in the file "process.def".

```
diagram_keep_list:='(n1 n2 .... n);
```

where the diagram n1,n2,...n will be calculated and others will droped. n1, n2 ... are the order number of the diagrams, It can be found by looking at the generated diagrams. Therefor this option is added to the file "process.def" after all the Feynman diagrams generated. It is the same situation for the next option.

3.2.6 To choose diagrams in kinematics generation

The default is to choose all the Feynman diagrams in kinematics generation. In some case, the user need to consider just part of diagrams in kinematics generation. To do that by adding the following line to the input file "process.def".

```
keep_list:='(n1 n2 .. n);
```

where the diagram n1,n2,...n will be take into account in the kinematics generation. It meams that all the resonance, t-channel singularities, in these diagrams will be treated in the kinematics generation.

3.3 Advance Options

All the options in this section are set to their default values in the FDC system file "$fdc/src/option_s". where "$fdc" means the directory of FDC system. It is very rarely to have chance for the ordinary user to change them. They are changed by FDC system developer to debug the system or to choose different scheme. In the following, all these options will be explained.

3.3.1 selfenergy

The default value of selfenergy is

```
selfenergy:='ok$
```

which means that the self-energy diagrams will be keeped. the user can change it to

```
selfenergy:='no$
```

if it is needed to drop self-energy diagrams.

3.3.2 ncolor

The option is used to declare if the initial and final state are color singlet. It is not color singlet in some calculation, and it is color singlet in other case. The default value of it is zero which means the initial and final state are color singlet. The user can set it to 1 to indicat that the initial and final state is not color singlet by adding the line as

```
ncolor:='1\$
```

3.3.3 showmore

This option is used to debug the Feynman diagram generator. The default value is 0, means not to print any debug information when the generator is running. The vaule can be 0,1,....10. More debug information are printed when it's value is set to bigger number.

```
showmore:='n$
```

where n=0,1,....10.

3.3.4 justnumber

The default is to demand that all the output of each diagram are generated and oputput to output files aa0a1, aa0a2,..... in subdirectory "diagram" It can be set to

```
justnumber:='ok$
```

which means that only the counting number of Feynman diagram is generated and output to file.

3.3.5 ndiagfile

All the diagrams are generated and output to files aa0a1,aa0a2,.... Each of these files contains "ndiagfile" diagrams, and the number of diagrams in the last output file will be less or equal to "ndiagfile". The default vaule is:

```
ndiagfile:='50$
```

and It can be set to any number.

3.3.6 bv_to_loop

The default is to demand that bound-state vertex must be connected to loop in the Feynman diagram generation for a process in which there is a bound-state. The option can be set to

```
bv_to_loop:='no;
```

which means that bound-state vertex are not demanded to connecte to a loop in Feynman diagrams.

3.3.7 bvex_rule

The default is to demand that all the diagrams, in which other external particles directly connected to bound-state-vertex, are droped in Feynman diagram generation. The option can be set to

```
bvex_rule:='no;
```

all the droped diagrams by the default demand of this option will be keeped.

3.3.8 fwuvy_apply

The default is to apply fwuvy therom[?] in Feynman diagram generation. The option can be set to

```
fwuvy_apply:='no;
```

which means not to apply fwuvy therom in Feynman diagram generation.

3.3.9 check_kin

The option is used to check kinematics generation and print out the numerical check results when FORTRAN is runed. The default value is set to go on without the check. In the case to check the generated kinematics, the user need to add the following line to the file "process.def".

```
check_kin:='ok$
```

3.3.10 physical_cut

The opiton is used to controle the physical cut. The default value is set "ok", which means physical cut condition will be applied if there are. The user can ask the FDC not to applied any cut condition by adding the line

```
physical_cut:='no$
```

3.4 Advance use

In this section, We introduce a lot of extra usage for the user in FDC, such as the cut of phase space range, choose of polarize of beam or final and initial particles, choose of many kinds of distributions, choose of range of some free parameters and so on.

3.4.1 To change parameters

It is easy to change parameters in the calculation by adding the following table to file "process.def".

```
input_list:=' (
              (hm 100.0)      % the parameters could be changed in this way
              (acc1 0.01)     % acurcy of grid optimization in bases
              (acc2 0.01)     % acurcy of integration in bases
              (itmx1 10)      % number of iteration for grid optimization
              (itmx2 10)      % number of iteration for integration
              (ncall 10000)   % number of sample point in each iteration.
              (mxdim 50)      % number of maxium integration dimension.
              ......
              );
```

where all the numerical number can be changed by the user to the value he want.

3.4.2 To do approximation in amplitude square calculation

It is needed to do some approximation to simplify calculation in some complicate case, To do it, the user can add the following table

```
approximation_rules:={
                      cnew*cnew=¿0,
                      ......
                      fme=¿0)
                      };
```

where the first line means to drop terms contains $cnew^2$, the last line means to drop electron's mass in amplitude square. It is easy to add more line to drop more terms. It will cause divergence

to drop fermion's mass in the case if there is t-channel singularity related to the fermion. Therefor, a safe way to do it is introduced in the following section.

3.4.3 To drop fermion's mass safely

It is safe if we drop fermion's mass in the numinator of amplitude square and keep it in the denominator. To do it, the user can add the following table to the input file "process.def":

```
mass_drop_list:='(
                  (fme 0)
                  ......
                  );
```

where the first line means to drop electron's mass in numinator of amplitude square. It is easy to add more line to drop mass of other fermion.

3.4.4 To obtain results for set of points in parameter space

It is easy to change parameters by adding the following table in the file "process.def".

```
change_parameters:='(
(hm 65 69 70 75 80 90)
(squart_root_s 3.5 4 5 7 10 30 60 70 90 91 92 134 140 161 175 200)
......
);
```

Where "hm" is Higgs mass and the first line means that Higgs mass changes from 65,69, ...,90 GeV, "squart_root_s " is the center mass energy and the second line means it will changes from 3.5,4,...,200 GeV. More lines could be added to the table to express the change of parameters.

3.4.5 To plot distributions

To plot distributions and to do physical cut, the user should understand a little bitter about the momentum setting. The momentum setting is as following:

```
namel   :='(ef efb wb h0 w);
momentum:=  p1 p2  p3 p4 p5
```

where all the momentum are in the laboratory frame. In the case of two initial particles, z-axial is the beam direction. In the case of one initial particle z-axial is of no special meaning.

The components of n-th momentum is:

```
p(1,n),  the energy component
p(2,n),  x-axial component
p(3,n),  y-axial component
p(4,n),  z-axial component
```

There are four quantity defined in FDC as following:
$cos\theta$:

```
examples:
    wcos(p3,p4) means cos of the angle between p3 and p4
    wcos(p3,z)  means cos of the angle between p3 and z-axial
    wcos(p4,x)  means cos of the angle between p4 and x-axial
    ......
    wcos_lorentz(p4,p3,p3+p5) means then cos of angle between p4,p3
                in the center-mass frame of p3+p5.
```

s: is the invariant mass, which is defined as

$$sqrts(pn) = \sqrt{p(1,n)^2 + p(2,n)^2 + p(3,n)^2 + p(4,n)^2}$$

```
examples:
    sqrts(p3,p4) means the invariant mass of p3 and p4
    sqrts(p3,p4,p5) means the invariant mass of p3,p4 and p5
    ......
```

pt: is the transverse momentum, which is defined as

$$pt(pn) = \sqrt{p(2,n)^2 + p(3,n)^2}$$

```
examples:
    pt(p3) means the transverse momentum of p3
    pt(p3,p4) means the transverse momentum of p3 and p4
    ......
```

vpt: is the pseudo-velocity, which is defined as

$$vpt(pn) = \frac{1}{2} log \frac{p(1,n)+p(4,n)}{p(1,n)-p(4,n)}$$

```
examples:
    vpt(p3) means the pseudo velocity of p3
    vpt(p3,p4) means the pseudo velocity of p3 and p4
    ......
```

There are two kind of ways to describe the distributions. One is to plot distributions visa the four well defined quantity described above. In this case, the boundary can be find out out by the FDC, therefore it will use the default boundary if the user do not give the boundary. The format of each line is

```
{n,a,mg,name}
{n,a,mg,b0,b1,name}
```

where the "n" means the n-th distribution, "a" is the $cos\theta$ or s, or pt or vpt variable visa which the distribution is plotted, "mg" means to divide the variable range into "mg" grids, and "name" is just the name given by the user to refer to the distribution. b0 and b1 are the upper and lower boundary respectively.

Another is to plot distributions visa ambitory expressions of momentums (include the four well defined quantities). In this case, the boundary of the variable must be inputed. The format of each line is

```
{n,a,mg,b0,b1,name}
```

where the "n" means the n-th distribution, "a" is a kind of expression of variable visa which the distribution is plotted, "mg" means to divide the variable range into "mg" grids, "b0" is the lower boundary of "a", "b1" is upper boundary of "a", and "name" is just the name given by the user to refer to the distribution.

The following is an example. To obtain the distributions, the following part must be added to the input file "process.def".

```
histograms  :=
            {
            {1,wcos(p4,z),50,"cos_theta"},
            {2,sqrts(p3,p4),50,"invariant_mass"},
            {3,pt(p4),50,"transverse_momentum"},
            {4,vpt(p4),50,-1.5,1.5"psudo velocity"},
            {5,sqrt(p(2,4)**2+p(3,4)**2),50,5.0,15.0,"T_momentum"},
            {6,p(4,4),50,5.0,15.0,"logitud_momentum"},
            {7,wcos_lorentz(p4,p3,p3+p5),50,"cos(theta) between p4 and p3"},
            ...
            {n,cos(p4,z),50,-1,1,"cos_theta"}
            }$
```

where the line 1 is to ask for the distribution visa the cos of the angle between p4 and z-axial, "50" in the line means to divide the variable range into 50 grids, and "cos_theta" is just the name given by the user to refer to the distribution. The above line number 2 is to ask for the distribution visa the invariant mass of p3 and p4, and more similar lines could be added.

The following is an example. To obtain the two dimension distributions, the following part must be added to the input file "process.def".

```
scatters  :=
          {
          {1,sqrts(p3,p4),sqrts(p4,p5),120,120,"Daliz of s34 and s45"},
          {2,sqrts(p3,p4),sqrts(p3,p5),120,120,1.0,3,0,1,0,3.0,"Daliz of s34 and s35"},
          {6,p(4,4),sqrts(p3,p4),100,100,"Daliz of of p4_z and s34"},
          ...
          }$
```

Where the format of each line is

```
{n,x,y,mx,my,name}
{n,x,y,mx,my,x1,x2,y1,y2,name}
```

where the "n" means the n-th distribution, "x" and "y" are the $cos\theta$ or s, or pt or vpt variable visa which the distribution is plotted, "mx" and "my" means to divide the variable range into "mx" and "my" grids, and "name" is just the name given by the user to refer to the distribution. x1 and x2 are the upper and lower boundary of x respectively. y1 and y2 are the upper and lower boundary of y respectively.

3.4.6 To do calculation for corelation

$abc(p_m, p_n, p_k) = (p(3,m)*p(4,n) - p(4,m)*p(3,n))*p(2,k) + (p(4,m)*p(2,n) - p(2,m)*p(4,n))*p(3,k) + (p(2,m$

```
coreelation  :=
             (
             (p1 p3 p4 "measurement of cp by (p1 x p3) . p4 ")
             (p1 p3 p5 "measurement of cp by (p1 x p3) . p5 ")
             (p1 p3 s1 "measurement of cp by (p1 x p3) . s1 ")
             (p1 s2 s1 "measurement of cp by (p1 x s2) . s1 ")
             ...
             (s1,s2,s3,"measurement of cp by (s1 x s2) . s3 ")
             )$
```

3.4.7 To do calculation for polarizations

To do calculation for polarizations, the user should understand a little bitter about the spin setting. The spin setting is as following:

```
namel   :='(ef  efb wb  h0  w);
spin    :=  ns1 ns2 nn3 nn4 nn5
```

where "ns1,ns2,....." is for fermions and "nn1,nn2,....." is for boson. There are two state of fermion spin, the two state denoted as "1", "2". There is only one state for scalar boson so that nn4 could not changed. For vector boson, the two transverse spin polarization state are express as "x", "y". It is also another two polarization state "er" and "el" which could be defined from "x", "y" as

$$er = \frac{x + iy}{\sqrt{2}} \quad el = \frac{x - iy}{\sqrt{2}}$$

For massive vector boson, there is an additional longitude polarization state is denoted as "z". To describe the polarizations. The format of each line is

```
( spin_1 ... spin_n m1 ... m)
```

```
where spin_i=(nni s),
      s=1,2 and nni=ns1,ns2.... for fermions.
      s=x,y,z or er,el,z. nni=nn1,nn2.... for vector boson.
      m1, ...., m means the order number of each distributions
                to be ploted, which is described in last section,
                and 0 means the total cross section.
```

There is an example in the following. to do the calculation for polarization, one has to add the following part to the input file "process.def".

```
polarizations:='  (
                    (0 5 6)
                    ((nn3 x) (ns1 1) 0 1 2 3 )
                    ((nn3 x) (ns1 2) 0 1 2 3 4 5 6)
                    ((nn3 er) (ns1 1) 0 4 5 6)
                    ((nn3 er) (ns1 2) 0 1 2 3 4 5 6)
                    ((nn3 el) (ns1 1) 0 1 2 3 4 5 6)
                    ...
                    ((nn3 el)) (ns1 2) 0 1 2 3 4 5 6)
                )$
```

where the first line (0 5 6) mean to calculate the total cross section, the distribution 5 and 6 without polarization. The second line mean to calculate the total cross section and the distribution 1, 2 and 3 with the first particle (electron) polarization ns1=1, the third particle (w^-) polarization nn3=x, and the other particles are not polarized. The other line can also be explained in the same way.

3.4.8 To do physical cuts

To describe the cuts. The format of each line is

```
{n,a,b0,b1},
```

where the "n" means the n-th cut condition, "a" is the variable visa which the cut is done, "b0" is the lower boundary of "a", "b1" is upper boundary of "a". "a" can one of the four predefined quantity or an ambitory expressions

The following is an example. To do the physics cut, the following part must be added to the input file "process.def".

```
phys_cut:=  {
            {1,wcos(p4,z),-0.96,0.96},
            {2,s(p3,p4),0.1,infinit},
            {3,pt(p3),5.0,15.0},
            {4,sqrt(p(2,4)2+p(3,4)2),5.0,15.0},
            ...
            {n,0.5dlog((p(1,4)+p(4,4))/(p(1,4)-p(4,4))),-0.5,0.5}
}$
```

The cut 1 is to do cut on the cos of the angle between p4 and z-axial with the lower boundary -0.96 and upper boundary 0.96.

The cut 2 is to do cut on the invariant mass of p3 and p4 with the lower boundary 0.1 GeV and upper boundary "infinite", where "infinite" means that to ask the FDC to find out the upper boundary of the whole range.

The cut 3 is to do cut on the transverse momentum of p3 with the lower boundary 5.0 GeV and upper boundary 15.0GeV.

The cut 4 is to do cut on the variable of expression $\sqrt{p(2,4)^2 + p(3,4)^2}$ with the lower boundary 5.0 GeV and upper boundary 15.0GeV.

The cut n is to do cut on the variable of expression $0.5 * dlog((p(1,4)+p(4,4))/(p(1,4)-p(4,4)))$ with the lower boundary -0.5 GeV and upper boundary 0.5 GeV

3.4.9 To use structure function in e-proton type collider

It is needed to use structure function of proton, or some similiar things in the case of e-proton or proton-proton colliders. We have defined proton in the section "physical model". To use it in the calculation of process, we introduced the following notation:

```
parton:='(na n E)

  0    "na" is a point particle
n=1    "na" is a parton in Proton with structure function type 1
  2    "na" is a parton in Proton with structure function type 2
  3    "na" is a photon in electron with structure function type 3


  E     is the energy of the particle "na"  when n=0,
        is the energy of the proton  when n=1,2.
        is the energy of the electron  when n=3
```

where all the three type of structure function are defined in the section "physical model".

we must add the following line in the input file "process.def".

```
namell:='(parton1 parton2)$
```

The following is the input file "process.def" for the calculation of $gamma + e^- \to \bar{t} + b + \nu_e$ in e^+e^- collider, where the γ is from the e^+ with structure function type 3 defined in the seciton "physical model".

```
model_home:=""/home/fdc/fdc96/model/smu_a1/";
process_name:=""e- p —¿t_bar b nu, tree";
process_author:=""Jian-Xiong Wang";
process_time:=""Dec 16, 1996";
namel:='(p ef qtb qb nue);
namell:='((p 3 250) (ef 0 250));
inpl:='( 1 1 -1 -1 -1);
ncorrection:='( (cnew2 1) );
```

Where it is easy to know in the line

```
namell:='((p 3 250) (ef 0 250))$
```

that "ef" (electron) is point particle with energy 250GeV, and "p" is the photon in electron (the energy of electron is 250GeV) with structure function type 3.

3.4.10 To use structure function in proton-proton type collider

The following is a example of the input file "process.def" for the the calculation of $g + g --- > g + J/\psi$ in proton-proton collider.

```
model_home:=""/home/fdc/fdc96/model/SM_unitary6/";
process_name:=""g g —¿g j/psi, tree";
process_author:=""Jian-Xiong Wang";
process_time:=""March 22, 1997";
namel:='(gs gs gs jpsi);
namell:='((gs 1 900) (gs 1 900) (qt2 hjpm2));
inpl:='( 1 1 -1 -1);
ncorrection:='( (g 0) (g3 0) );
ncolor:='1;
```

Where it is easy to know in the line

```
namell:='((gs 1 900) (gs 1 900))$
```

that the first "gs" (gluon) is parton in the first proton (the energy of the proton is 900GeV) with structrure function type 1, and the second "gs" (gluon) is parton in the second proton (the energy of the proton is 900GeV), with structrure function type 1,

3.4.11 To Make LIKELEHOOD Fitting

The follwoing three terms should be added to the file "process.def".

```
mkfit:='yes;                    %declare to do Likelihood Fitting.
resonace_list:='( ..... );      %declare resonaces
fit_parameter_list:='( ..... ); %declare fited parameters
```

Then to run the following command as usual.

```
> diag      % generate Feynman diagram
> amp       % manipulate amplitude and generate Fortran source
> kine      % manipulate kinematics and generate Fortran source
> cd fort
> make      % compile and link Fortran source
```

There are five Executable file generated after "make". The user can choosed some of them to run according to his need.

```
> int
```

To integrate cross section and generate distribution. the result of cross section is in the file "fresult.dat" and the convergence behaive is in the file "convergence.dat". The distribution data is in the file "plot.hbook" for pawX11 and the file "plot.dat for directly text view. The information for event generate is in the file "bases.data".

```
> gevent
```

To generate event by using the information in the file "bases.data" and the generated 4-momentum data of event is in the file "pdata1.dat". The file "pdata1.dat" is of the fortmat as:

```
Data of each Event
..... .....
```

and each event is of the following format. There is no momentum of initial paticles, just all the momentum of final particles

$$\begin{array}{cccc} P_{3,x} & P_{3,y} & P_{3,z} & E_3 \\ P_{4,x} & P_{4,y} & P_{4,z} & E_4 \\ \ldots & \ldots & \ldots & \ldots \end{array}$$

By performing the following program

```
> kint
```

to integrate cross section and generate distribution. the result of cross section is in the file "kfresult.dat" and the convergence behaive is in the file "kconvergence.dat". The distribution data is in the file "kplot.hbook" for pawX11 and the file "kplot.dat for directly text view. The information for event generate is in the file "kases.data".

```
> gkevent
```

To generate event by using the information in the file "kases.data" and the generated 4-momentum data of event is in the file "pdata1.mc".

```
> fit
```

To do Likelihood fitting by using 4-momentum data file "pdata1.dat" of Experiment events, phase space Moto Calo 4-momentum data file "pdata1.mc" and amplitude square Fortran source. The result of parameter fitting is in the file "pep.res". The file "pep.res" is of exactly same index as the input file "fpara.inp", but the "fpara.inp" is input and "pep.res" is the output. The file "flag.inp" is used to control the behaive of "fit".

Example Of Input Items Or Files

resonace_list is the list of resonace in this proecss, you can change the mass and width of resonaces in input data file "fort/reson.inp" without changing of fortran source.

```
resonace_list:='  (
                  %(Ser.Nu   name   mass   width   )
                  (1         pr1    1.9    0.001   )
                  (2         pr2    1.0    0.1     )
                  (3         pr3    1.0    0.1     )
                  (4         pr4    1.0    0.1     )
                  );
```

fit_parameter_list is the list of parameters which you want to change their value when you run program "fit" to fit with experiment data. The parameters are in the input file "fort/fpara.inp" and you can change them in the file.

```
fit_parameter_list:='(
%(Ser.Nu        name   value   stepvalue   Minium   Maxium   )
(1              f44    0.1     0.01        -10      10       )
(2              f45    0.1     0.01        -10      10       )
(3              f46    0.1     0.01        -10      10       )
(4              f47    0.1     0.01        -10      10       )
(5              f48    0.1     0.01        -10      10       )
)                                                   ;
```

The follow input files, generated by FDC as input data for generated Fortran source, are in the subdirectory "fort". User can change all the parameters in them to control and change parameters in the calculation.

flag.inp

```
'ITER='      15      number of iteration to do LIKELIHOOD fit
'PLOT='      1
'do_mc='     'y'     this option is used to optiomization of
                     the calculation when you run "fit" at first time.
                     You should use 'do_mc=' 'n' after the first time.
'MC_num='    10000   number of sample points in phase space Moto Calo
                     file "kdata1.dat"
'da_num='    20000   number of events in experiment data file "pdata1.dat"
```

Where the detail of "do_mc" option is described in FDC resport.

fpara.inp

```
5
   1   0.1   0.01   -10   10
   2   0.1   0.01   -10   10
   3   0.1   0.01   -10   10
   4   0.1   0.01   -10   10
   5   0.1   0.01   -10   10
```

reson.inp

```
10                                  % number of resonace
    %Ser.Nu   mass    width
    1         1.9     0.001
    2         1.0     0.1
    3         1.0     0.1
    4         1.0     0.1
```

3.5 To draw Feynman diagrams

All the Feynman diagram are drawed into PS file automatically by FDC when a processes is generated. The tool to draw Feynman diagram can also be used manually. There is brief description in the following.

psdraw -[fxy] aaa > aa.ps take **aaa** and psnametable as input and generate the file **aa.ps** options:

-f to draw diagrams with frame to be printed out. The default is to draw diagrams without frame.

-x nx to place nx diagrams in x direction of the paper, the default value of nx is 4.

-y ny to place ny diagrams in y direction of the paper, the default value of ny is 6.

3.6 To generate and calculate a series processes under some topic

At first, the user must prepare the file "process.def" and then to run the following command set to do all the things.

```
> findall
> mkps
> mkamp
> doint
```

where the following things are done in *findall*

```
dofind   ---- search for all the process which match the given pattern.
mkall    ---- generate all the Feynman diagrams for each process.
```

and the following things are done in *mkps*

```
drawall  ---- draw Feynman diagrams of all the process
mkpsfile ---- make the reference book of all the processes in
              latex version and ps version.
```

and in *mkamp* the following command set will be done in each subdirectory of each process.

```
> amp
> kine
> cd fort
> make
```

where *amp, kine* are explained in the above, and *make* is the command in unix, which will use the makefile file in this subdirectory to perform compile Fortran source and link the object files and integration library *BASES*.

3.7 Examples

There are examples in "$fdc/example". It is easy to create all processes in e^+e^- collider by following procedures.

```
> process_cp $fdc/example/e-e-collider e-e-collider
> cd e-e-collider
> findall
> mkps
> mkamp
> doint
```

3.7.1 The input file: "process.def"

The following is the detailed description of the input file "process.def":

```
algebraic;
model_home:="/home/fdc/fdc96/model/smu1/";
process_name:="e+e- Collider Physics";
process_author:="Jian-Xiong Wang";
process_time:="Aug 13, 1997";
namel:='(ef efb (2 3 4));
inpl:='( 1 1 -1 -1 -1);
number_of_loop:=0;
ncorrection:='( (g 0) (g3 0) );
ncolor:='1;
ninitial:='2;
  mprocess:='  (
               (ef efb any_b any_b)
               (ef efb any_f any_f)
               (ef efb any_b any_f any_f)
               (ef efb any_b any_b any_b)
               (ef efb any_b any_b any_b any_b)
               (ef efb any_l any_l any_l any_l)
               (ef efb any_q any_q any_l any_l)
               (ef efb any_q any_q any_q any_q)
               );
  end;
```

Where each line in "mprocess:=.." means a type of processes which the user ask the fdc to generate. For example, the first line

```
(ef efb any_b any_b)
```

means all processes of $e^-e^+ \to$ two boson. All the particles notation are listed in physical model part. All the options and some of advanced usages could be used in the input file "process.def".

4 Examples

There are a lot of examples provided in FDC. All of them are subdirectores in "$fdc/example", and each of them contains some processes as shown in the following:

gggjpsi is $g + g \to g + J/\psi$ in proton-proton Collider

eptbnu is $e + \gamma \to t + \bar{b} + \nu_e$ in e-proton type Collider

wwh is $e^+e^- \to w^+w^-H$

zbb is $e^+e^- \to b\bar{b}z^0$

e-e-collider is all processes in e^+e^- collider

e-proton-collider is all processes in $e^-Proton$ collider

proton-proton-collider is all processes in $ProtonProton$ collider

e-mu-collider is all processes in $e^-\mu^+$ collider

e-p-collider is all processes in $e^-\gamma$ collider

mu-mu-collider is all processes in $\mu^-\mu^+$ collider

jpsi-in-ee-collider is all processes of $e^-e^+ \to J/\psi + X$

jpsi-in-ep-collider is all processes of $e^-Proton \to J/\psi + X$

jpsi-in-pp-collider is all processes of $ProtonProton \to J/\psi + X$

tau-e-background is all processes of $e^-e^+ \to J/\psi + X$