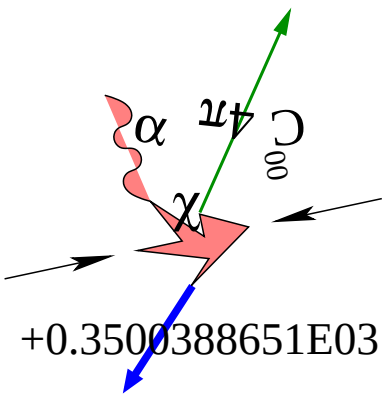

*a*n ı̊ntegrated Tool for Automated Loop Calculations

*a*ı̊TALC

Version 1.3.1: *The 2 to 2 fermion release*

September 27, 2006

Alejandro Lorca – Tord Riemann

Copyright © 2004 Alejandro Lorca – Tord Riemann

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Cover art by Alejandro Lorca

Nature of the problem

*a*ÅTALC is designed as a tool to perform automated perturbative calculations of cross sections in high energy physics.

The main goal is to create numerical programs directly from Feynman rules. It is actually developed for few known models like the electroweak standard model and QED, and the limit of application is, for the moment, the $2 \to 2$ particles reactions involving only external fermions.

The user will be expected to cook his or her own process within a set of basic and/or advanced ingredients. As a reward to such effort comes the output of *a*ÅTALC: Fast and reliable FORTRAN code with differential and integrated cross sections, analytical expressions for the transition amplitude and nice drawings of the individual Feynman graphs. It was never so easy to get all of it at once!

The tool was intended to integrate only free-of-charge packages existing on the market. As the calculation proceeds, in a modular fashion, we will profit from the DIANA package [TF00] (based itself on the QGRAF [Nog] code) for the generation and analysis of Feynman graphs, the FORM [Ver] language when dealing analytically with the large expressions in the amplitudes and, finally, the LOOPTOOLS [HPV99] library (also integrating the FF package [vO91]) for the numerical calculation of the loop integrals.

Last but not least, this tool owns its birth to the evolution of the static FORTRAN code TOPFIT [FLRW03], intended for the calculation of top-pair production. Some of the features then available such as the hard-photon bremsstrahlung or separate weak-QED contributions were discarded in pro of a full automation, higher technical precision and flexibility for many other processes.

Contents

1 Installation **1**
- 1.1 Technical requirements . . . 1
- 1.2 Downloading . . . 2
- 1.3 Extracting . . . 2
- 1.4 Where are the paths? Configuration . . . 2
- 1.5 Compiling the modules . . . 3
- 1.6 Error report . . . 3

2 How to work with *a*i̊TALC? **4**
- 2.1 A preliminary example at tree level: μ-pair production . . . 4
- 2.2 The environment through Makefile . . . 6
- 2.3 DIANA and the driver file . . . 7
  - 2.3.1 Basic driver file: Input . . . 7
  - 2.3.2 Advanced setup . . . 8
  - 2.3.3 What can we get? . . . 9
- 2.4 Using the FORM libraries . . . 11
  - 2.4.1 `#procedures` without end . . . 11
- 2.5 Getting numbers: the FORTRAN code . . . 12
  - 2.5.1 Understanding and controlling the output . . . 12
  - 2.5.2 A greedy possibility: quadruple precision . . . 13
- 2.6 Examples including loop level . . . 13

3 Hot potatoes in the scope **16**

Acknowledgements **18**

A References for the package DIANA **19**

B Implemented models **20**

C List of FORM procedures in KITFORM3 **22**

D Settings for `main.f` **25**

E Log of changes to previous version **27**
E.0.1 Changes in 1.3.1 to version 1.3 27
E.1 Changes in 1.3 to version 1.2 27
E.2 Changes in 1.2 to version 1.1 27
E.3 Changes in 1.1 to version 1.0 28

F GNU Free Documentation License **29**
F.1 APPLICABILITY AND DEFINITIONS 29
F.2 VERBATIM COPYING 30
F.3 COPYING IN QUANTITY 30
F.4 MODIFICATIONS 31
F.5 COMBINING DOCUMENTS 32
F.6 COLLECTIONS OF DOCUMENTS 32
F.7 AGGREGATION WITH INDEPENDENT WORKS 32
F.8 TRANSLATION 33
F.9 TERMINATION 33
F.10 FUTURE REVISIONS OF THIS LICENSE 33
ADDENDUM: How to use this License for your documents 33

Chapter 1

Installation

Often the installation of scientific software products is a nightmare. This chapter will help you to dribble such initial difficulties and hopefully run your code. Please pay attention to next sections, be a bit patient and cross fingers!

1.1 Technical requirements

We present a list of the necessary conditions to install and run *a*ÎTALC on your computer.

System compulsory requirements:

- A computer under LINUX[1] operating system with standard[2]
  - GNU MAKE utility
  - C compiler
  - FORTRAN77 compiler (with preprocessing ability)

Suggested optional features:

- CPU speed $\geq$ 333 MHz
- RAM Memory $\geq$ 128 MB
- At least from 1.5 to 10 MB[3] free space on disk for installation

☛ Further limitations given by the individual modules

[1] Installation on other UNIX systems might be possible if you already have some modules ready. The package has been succesfully build in SOLARIS for testing purposes.

[2] Tip: If lacking, free packages can be obtained from `http://www.gnu.org` under the names `make` and `gcc`.

[3] The fewer modules you have already installed the larger the size for including them. Another extra 30 MB will be needed if you build the three examples proposed.

1.2 Downloading

Most of the times, one finds this step as a circular reference: the information about where to download a code appears only in the tutorial attached to the code. Once the reader has broken this closed circuit we acknowledge the official site.

```
http://www-zeuthen.desy.de/theory/aitalc
```

Official site of *aiTALC*

1.3 Extracting

The file `aitalc-1.3.1.tar.gz` is a compressed group of files. Wherever the installer decided to save it[4], the next two commands

```
gunzip aitalc-1.3.1.tar.gz
tar -xvf aitalc-1.3.1.tar
```

Uncompress and unpack *aiTALC* at your computer

will create a new system directory called `aitalc-1.3.1`. From now on we will refer to it as the direcory `$AITALCHOME` in which all the necessary files will be placed.

1.4 Where are the paths? Configuration

In order to share the information between the different modules of the package, the installation routine should know where to look for the libraries or execute binaries. A full installation of *aiTALC* creates the default structure given in Tab. 1.1. Of course, if some of the components are already installed, it will be preferrable to find such valuable parts instead of performing a duplicate installation. Run the next command with such an intention.

```
cd ${AITALCHOME:=‘pwd‘/aitalc-1.3.1}
./configure
```

Search and establish the path for each module

If the script is not clever enough to find them or they are not installed on your machine, you might be prompted to request the missing modules from the net. Answer ‘yes’ (`Y`) if your internet conection is fast enough!

[4]Make sure you have write permission in that directory!!!

1.5 Compiling the modules

It should be straightforward, just type

```
make
```

Compiles everything needed

and the tool will be available. Let's do then interesting physics.

Module	→ **Directory**
DIANA	→ `$AITALCHOME/diana/diana-2.37/`
QGRAF	→ `$AITALCHOME/diana/diana-2.37/qgraf/`
TEDI	→ `$AITALCHOME/diana/diana-2.37/bin/`
FORM	→ `$AITALCHOME/form/bin/`
KITFORM3	→ `$AITALCHOME/form/kitFORM3/`
FORTRAN	→ `$AITALCHOME/fortran/`
LOOPTOOLS	→ `$AITALCHOME/fortran/LoopTools-2.2/`

Table 1.1: Default path location for the different software components after a full installation.

1.6 Error report

If you feel desperate because something went wrong, probably you will never want to hear about *a*ÍTALC again. Give it a chance and do not hesitate to contact the authors immediately, reporting the installation or other kind of bug. Maybe the problem can be easily solved and anyone benefit from an appropriate patching in the package.

```
mailto:alejandro.lorca@physik.uni-freiburg.de
mailto:tord.riemann@desy.de
```

Bug!

Chapter 2

How to work with *aïTALC*?

Are we ready to run *aïTALC*? As being mandatory in any prestigious code, we have a `'Hello world'` testing example; it will be a basic tree level process. Furthermore the description of each of the modules is contained in this chapter and provides the user with sufficient material to customize her or his own calculations.

2.1 A preliminary example at tree level: μ-pair production

Let's satisfy the impatient user and see what this weapon is able to do in unexperienced hands.

The directory `examples/` contains three prepared processes you can play with, one of them is inside `muon_production/`.

☞ From now on we assume you have write access on `examples/muon_production/`. If your system administrator installed *aïTALC* for you, probably you won't be able to write in `$AITALCHOME`. In this case copy the directory `examples/` into a suitable location for running.

```
cd examples/muon_production
make
cd tree
gv muon_production.ps &
```

Go inside the examples and type in successively the commands to run the process of μ-pair production and watch the diagrams

The compilation and execution will take a while, depending on your system from few seconds up to one minute or so, please be patient. When no error messages popped out, then your are viewing the Fig. 2.1. It is a graphical representation of the Feynman diagrams that add up into the scattering amplitude for the selected process. In them, the time coordinate is the horizontal axis that, as usual, is not shown. Incoming particles appear at the left of the diagram. Note that both fields e^- and e^+ are represented by the same symbol e. In order to identify the particle

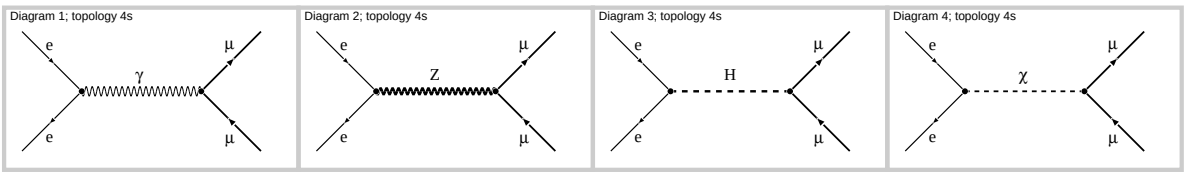

Figure 2.1: Diagrams appearing according to the `EWSM.model` for $e^-e^+ \to \mu^-\mu^+$ at tree–level.

(antiparticle) you have to check whether the fermion arrow points forward (backward) along the time direction; the same applies to all fermions.

If your wish is not so graphical, but instead the numerical achievement of the calculation, consider the following:

```
cd ../fortran
./main.out
cat main.log
```

Enter in this directory to check the numerics of the process

and immediately your screen will be filled with the content of Table 2.1, that you should interpret as the following: *I got in the tree–level approximation for the process* $e^-e^+ \to \mu^-\mu^+$ *the following set of differential cross sections*[1] *(in pb), at different scattering angles for a beam energy of 500 GeV, and maximum soft photon energy of 50 GeV.*

If after such a show you are still interested, read the rest of this chapter which describe extensively the more powerful usage of *a*ÎTALC, part by part.

```
#=============================================================================================
# aITALC Version 1.0 (29.10.2004) by A.Lorca -- T.Riemann
#---------------------------------------------------------------------------------------------
#sqrtsman     cost      dcs(Born)    dcs(loop)    dcs(soft)    dcs(B+l+s)   dcs(loop^2)
 0.5000000E+3 -.9000000 0.9458936E-1 0.0000000E+0 0.0000000E+0 0.9458936E-1 0.0000000E+0
 0.5000000E+3 -.5000000 0.8929449E-1 0.0000000E+0 0.0000000E+0 0.8929449E-1 0.0000000E+0
 0.5000000E+3 0.0000000 0.1503198E+0 0.0000000E+0 0.0000000E+0 0.1503198E+0 0.0000000E+0
 0.5000000E+3 0.5000000 0.2865049E+0 0.0000000E+0 0.0000000E+0 0.2865049E+0 0.0000000E+0
 0.5000000E+3 0.9000000 0.4495681E+0 0.0000000E+0 0.0000000E+0 0.4495681E+0 0.0000000E+0
#=============================================================================================
#Energy given in GeV, Cross sections in pb, setfracomega= 0.1000000
```

Table 2.1: File `main.log`, produced by the run of `main.out`.

[1]By default they are given by the value of $\cos\theta$. The scattering angle θ refers to the angle between the 1st incoming and the 1st outgoing particle in a $2 \to 2$ fermion process.

2.2 The environment through Makefile

Automatization requires effective inter–communication and a smart way to organize the tasks. We found that this second concept was by far much easier to implement under a Makefile environment than, let's say, programming everything from the scratch in C++.

The Makefile environment handles with

- Simplification of user interface
- Building the sections `tree`, `loop` and `fortran`
- Running the driver process file
- Writing the intermediate information DIANA → FORM → FORTRAN in a row
- Compiling the numerical program leading to the final results

in a modular fashion. The execution takes place sequentially following the control flow structure depicted in Fig. 2.2

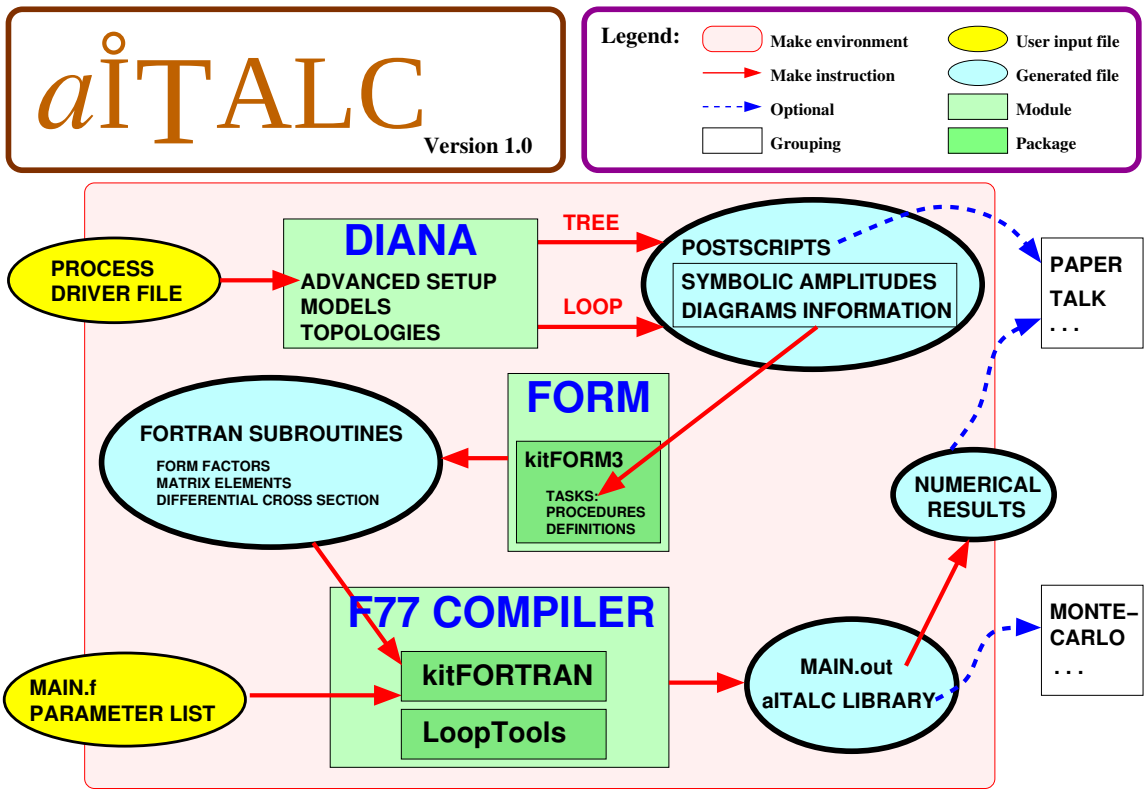

Figure 2.2: Logical flow chart structure for *a*ÎTALC.

2.3 DIANA and the driver file

Let us introduce some basic concepts about DIANA. If we are thinking of some process, a very intuitive approach is to draw some Feynman diagrams to get a feeling and support the calculation which comes later.

Such a task is very likely to be implemented with the help of computers since it is basically a combinatorial problem. Once we define a model, the external particles, and the level of loops, the problem can be solved. Indeed, DIANA will do it for us.

2.3.1 Basic driver file: Input

The driver file (e.g. `process.ini`) serves as basic input. It contains primarily:

- incoming and outgoing particles,
- physical model,
- process control flags and options.

These items are crucial for the determination of the DIANA "create" file, in which the `Makefile` also determines the number of loops. Then DIANA is able to offer some of the information to QGRAF and create the possible diagrams appearing at each level. Additionally extra information about the global process and for each individual graph is attached into the output file at each level e.g. `tree/$processname.in`.

Be careful with the momenta definitions. They all should be assigned to be incoming and clockwise indexed (e.g. `p1` addressed to be the first incoming particle, `-p2` to be the first outgoing one and so on). Follow the convention used in the examples, otherwise the results are not guaranteed to be correct.

We suggest the user to have a look to an example driver file, for instance the `process.ini` from the muon production as shown in Fig. 2.3.

As you probably realized, the comment character is `*` and the `** MODIFY HERE! **` sentences indicate where some modifications are accepted. At this basic level the modifications concern: processname, model name, incoming and outgoing particles and QGRAF options. Regarding the flags for the initial settings, the topology editor (TEDI) can be launched at running time by uncommenting the line: `*SET _TOPOLOGYEDITORNAME="tedi"`. TEDI is designed to allow the user ultimate flexibility on topology naming, shaping and even defining momenta. It requires the X window server in your system, but it's quite intuitive to manage as one may imagine by looking to the Fig. 2.4.

process.ini Page 1/2

```
************************************************************************
** Driver file for Diana 2.35 <process.ini>                           **
** This file is part of aITALC                                        **
** Copyright (C) 2004 Alejandro Lorca <alejandro.lorca@desy.de>       **
**                                                                    **
** Please, modify the file with your taste, mainly after the          **
** MODIFY HERE! sentence.                                             **
************************************************************************

* DIANA RUNNING SETTINGS
**====================================================================
** MODIFY HERE! **
SET _processname=muon_production

** To assign direction in loop momenta and naked integration line
*SET _MARK_LOOP="YES"
** Next means remember loop momenta
*SET rlm = "rlm"

** Next (YES/NO) avoid/load
** generic topology tables and internal abstract momenta
SET _NO_TABLES="YES"
*SET _NO_TABLE_MOMENTA="NO"
*--------------------------------------------------------------------
** To be adjusted by autoconf
SET _DIANANAME=diana35
** if you want to edit the topologies:
*SET _TOPOLOGYEDITORNAME=tedi

only interpret
\openlanguage(create.tml)
\MINVERSION(2.35)
*--------------------------------------------------------------------

\Begin(program)

* MODEL TYPE
**====================================================================
** MODIFY HERE! **
*\Begin(model,QED.model)
\Begin(model,EWSM.model)
\End(model)
**--------------------------------------------------------------------

* DESCRIPTION OF OUR PROCESS (INGOING AND OUTGOING PARTICLES)
**====================================================================
\Begin(process)
** MODIFY HERE! **
ingoing le(;p1),Le(;p4);
outgoing lm(;-p2),Lm(;-p3);
** To be adjusted in the Makefile
loops=N;
\End(process)
```

process.ini Page 2/2

```
**--------------------------------------------------------------------
\loopmomenta(q)
\zeromomentum(zero)
* For the topology tables
*\token(bridge,b)
*\token(chord,k)
*\token(loop,q)
**--------------------------------------------------------------------
\include(runningrestrictions.prg)

* LOADING/CREATION OF TOPOLOGIES
**====================================================================
** When creating a topology from the scratch, please save it into
** a file and rerun Diana with that topology file.
**--------------------------------------------------------------------
** To be adjusted with the Makefile
*\tables(4legsNloops.tab)
\Begin(topology,4legsNloops.top)
\End(topology)
**--------------------------------------------------------------------
\Begin(qgrafoptions)
** MODIFY HERE! **
*options=;
options=notadp,onshell;
\End(qgrafoptions)
**--------------------------------------------------------------------

* POSTSCRIPT FLAGS AND OPTIONS
**====================================================================
\Begin(header)
**--------------------------------------------------------------------
** MODIFY HERE! **
SET MakePs = "!"
SET MakeInfo = "!"
SET MakeEps  = "!"
SET MakeInfoEps  = "!"
**--------------------------------------------------------------------
debug off
forced execute
extra call
messages disable
\End(header)
**--------------------------------------------------------------------

* TML LANGUAGE
**====================================================================
\include(thetmlstuff.prg)
**--------------------------------------------------------------------
\indices(mu1,mu2,mu3,mu4,mu5,mu6,mu7,mu8,mu9,mu10,mu11,mu12)
\vectors(p,p1,p2,p3,p4,p5,p6,k,k1,k2,k3,k4,k5,k6,q,q1,q2,zero)
**--------------------------------------------------------------------
\End(program)
```

Figure 2.3: Snapshot for `process.ini` driver file.

2.3.2 Advanced setup

But what about if we want to exclude some couplings or modify the aspect of the diagrams?

After the installation of *a*ÎTALC it is also possible to use DIANA[2] as an independent package (see App. A). For our present purpose, we will show shortly the tested modifications that can be performed at this level.

If you have already been running the example, inside of the `tree` directory (also the `loop` directory in other examples) you can select some files to be considered instead of the default coming from the following directory: `$AITALCHOME/diana/prg`. This is automatically done by placing those files on the `examples/muon_production/` directory.

We do not suggest to copy and modify any other file inside `prg` since they are mostly functions using advanced DIANA declarations, so just consider to play with the following:

- `particleaspect.prg`: Defines how every propagator and particle label looks like in the `.ps` and `.eps` files. Syntax is not described (just a couple of comments), but intuitive. The field label `CT` is assigned to counterterms.

[2]This extends also to FORM and LOOPTOOLS.

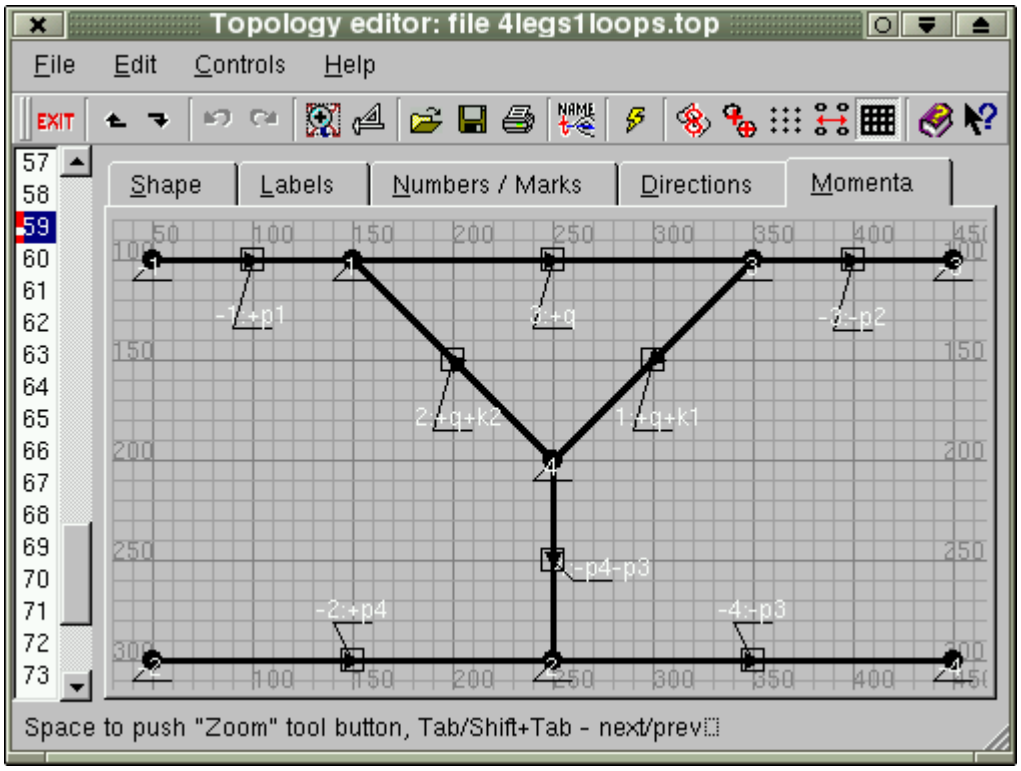

Figure 2.4: Snapshot for the momenta tab in TEDI

- `runningrestrictions.prg`: Exclude some couplings or fields here. As before uncomment or replace as desired.
  Warning: As soon as you may discard some fields or couplings, be aware that the renormalization of parameters and fields is FIXED and complete within each model and does not know about your intentions, so perform always a consistency check of your results!

2.3.3 What can we get?

Let's have a look inside a level directory (e.g. `tree` or `loop`): The organizer file is called `Makefile`. It has the instructions to create, organize and fill the different directories by given commands under the MAKE environment. This means you have to do nothing but wait while monitoring the abundant screen messages. Depending on the process, modifications, hardware and compiler the time for each module varies from seconds till several minutes[3]. You can keep trace about how much it took by looking the time stamps in the `Makefile.log` file at Fig. 2.5.

[3]If large files are to be compiled, occasionally your system might overload. Doubt about everything taking longer than half an hour.

```
alorca@linux:~AITALCHOME/examples/muon_production/tree> ls -XFl
total 183
drwxr-xr-x    2 alorca   users     2048 Oct 14 15:02 EPS/
drwxr-xr-x    2 alorca   users     2048 Oct 14 15:02 FFmuon_production/
drwxr-xr-x    2 alorca   users     2048 Oct 14 15:02 InfoEPS/
-rw-r--r--    1 alorca   users     3291 Oct 14 15:02 Makefile
-rw-r--r--    1 alorca   users     3710 Oct 14 15:02 muon_production.cnf
-rw-r--r--    1 alorca   users       20 Oct 14 15:02 fermioncurrentsnames.in
-rw-r--r--    1 alorca   users     3870 Oct 14 15:02 muon_production.in
-rw-r--r--    1 alorca   users     4531 Oct 14 15:02 Make_kit.log
-rw-r--r--    1 alorca   users      149 Oct 14 15:02 Makefile.log
-rw-r--r--    1 alorca   users     1088 Oct 14 15:02 do_amplitude.log
-rw-r--r--    1 alorca   users      245 Oct 14 15:02 joinff.1.log
-rw-r--r--    1 alorca   users     1059 Oct 14 15:02 joinff.log
-rw-r--r--    1 alorca   users      629 Oct 14 15:02 joinkinematics.log
-rw-r--r--    1 alorca   users   121102 Oct 14 15:02 joinmm.log
-rw-r--r--    1 alorca   users    14779 Oct 14 15:02 muon_production.ps
-rw-r--r--    1 alorca   users    18277 Oct 14 15:02 muon_productionInfo.ps
```

Figure 2.5: Listing for the `tree` directory inside our muon production example.

Diagram info

As it comes out of DIANA, the file `$processname.in` contains the required information about both, global process and individual diagrams. Specially useful for the FORM routines are the preprocessed variables defined for each diagram.

Graphical information

The global files `$processname.ps` and `$processnameInfo.ps` summarize all the Feynman graphs appearing at the process. Inside the `EPS/` (and `EPSInfo/`) directories one may find .eps files with all individual diagrams (and momenta-label definition).

Analytical expressions (FORM processed)

In the files `do_amplitude.log` and `joinmm.log`, information will be found about the individual evaluation of each diagram and the kinematical factor corresponding to the cross product of matrix elements, respectively. Also, extra kinematical information can be found in `joinkinematics.log`. As an intermediate step, individual amplitudes are stored in the `FF$processaname/` directory.

2.4 Using the FORM libraries

As soon as DIANA has finished one of the levels, a complete description of the process falls into our hands, but still in an encoded language. Different FORM programs will prepare the amplitudes in order to render numerically evaluable expressions. Tracing, commuting γ matrices, applying the Dirac equation to external fermions or introducing the Mandelstam variables is part of the physical achievement at this module.

There are basically two kind of programs: The main routines and the procedures. We describe the four routines:

- `do_amplitude.frm`: Takes care of chewing the individual amplitude for each diagram, extracting out the form factors in a given basis of matrix elements.
- `joinkinematics.frm`: Defines how the differential cross section should be composed in terms of fermion currents (`S`, `T` or `U`), the normalization of the incoming flux in the cross section formula, soft photon emission and the definition of the Mandelstam variables.
- `joinmm.frm`: The standard set of matrix elements is considered by this program, giving, as output, the multiplication of any two elements appearing in the amplitude decomposition.
- `joinff.frm`: Once all the diagrams were considered, this part pastes together the form factor contributions sorted by topologies. The final translation into FORTRAN code is also done at this step.

2.4.1 `#procedures` without end

One of the most important principle in any calculation is the correctness of the expression after every step. By using procedures with a clear physical or mathematical intention, we ensure a systematic processing for the Feynman amplitudes and an easy way to eliminate potential bugs.

The listing of the procedures is long, and the description of each of them is shortly presented in App. C.

At least we would like to draw the attention to one option. If you want to neglect the masses of particles (e.g. light fermion masses), you can easily discard any term proportional to them including your costumized `Neglectedmasses.inc` file.

```
cd ..
cp $AITALCHOME/diana/prg/runningrestrictions.prg .
cp $AITALCHOME/form/kitFORM3/Neglectedmasses.inc .
(edit the copied files)
make clean
make
```

Activate your own settings (e.g. neglecting particle's mass terms)

2.5 Getting numbers: the FORTRAN code

Considering the `fortran` module, its structure can be outlined as follows:

- `main.f, parameterlist.hf`: they both give the user access to modify the output structure of the program and the explicit values of the model respectively. In the `main.f` file we find many settings as logical flags, input of kinematics and output control.

- KITFORTRAN: consists of the full set of FORTRAN subroutines and functions needed to compile the `main.f` program. We can divide them into two categories, according to the role they play during running time.

  - **Global** are those that were fixed at installation time and considered to be process independent. They are stored at `$AITALCHOME/fortran/src`
  - **Local** are written at running time by the `tree` and `loop` modules and depend explicitely on the application. They get placed at `$processname/fortran/src`.

- LOOPTOOLS: with the purpose of calculating the loop integrals this library is called.

2.5.1 Understanding and controlling the output

If after all, the desired scattering reaction was calculated, it will not be a waste of time to give a bit of explanation on the results and how to adapt them to your needs.

In the table of differential cross sections (given always in pb), the first column corresponds to the beam energy ($\sqrt{s} =$ `setsqrtsman`) given in GeV, the second one to the $\cos\theta$ (`setcost`), θ being the angle between the three-momenta $\mathbf{p_1}$ and $-\mathbf{p_2}$. Later on we have successively the columns of the Born approximation (`Born`), the interference terms of the tree and loop levels (`loop`) and the soft photonic corrections (`soft`). These three columns (third ,fourth and fifth) are summed into the sixth column (`B+corr`) resulting into the finally corrected cross section. The last column `loop^2` returns back the amplitude concerning the one-loop squared terms in the perturbative approach. The maximum energy fraction that the soft photon may gain out of $\sqrt{s}$ is limited by the variable `setfracomega`.

If the calculation is correct, all the columns should be constant[4] under variation of the ultraviolet (UV) parameter (you can check this by turning on the flag `luvcheck`). The same check for the infrared (IR) behaviour (`lricheck`) will change the explicit values in columns `loop` and `soft`, but the sum should still be invariant! The last column may also suffer against IR variation since it is not compensated by double soft photon emission. Those `loop^2` numbers are crucial when they become the leading order in e.g. flavour changing neutral current processes (FCNC), or just as an estimator of the order of magnitude for the error in the next level of perturbation theory. When integrated cross sections are under study, the integration region is limited by the

[4]Warning: as a result of numerical variation, the round off of intermediate results will modify the last digits in your results. Such variation increases at the collinear cases due to cancellations.

`setlimcost` variable, the `ics` and `fba` being defined as

$$\texttt{ics} = \sigma_{\rm tot} = \int_{-\texttt{setlimcost}}^{\texttt{setlimcost}} \mathrm{d}\cos\theta\ \texttt{dcs} \tag{2.1}$$

$$\texttt{fba} = \frac{\sigma_{\rm fw} - \sigma_{\rm bw}}{\sigma_{\rm tot}} = \int_{0}^{\texttt{setlimcost}} \mathrm{d}\cos\theta\ \frac{\texttt{dcs}}{\texttt{ics}} - \int_{-\texttt{setlimcost}}^{0} \mathrm{d}\cos\theta\ \frac{\texttt{dcs}}{\texttt{ics}}. \tag{2.2}$$

$$\tag{2.3}$$

The integration algorithm is based on a Richardson extrapolation to the Romberg integration [PFTV92] with four steps. The estimated error is supplied in short forms in extra columns. If it does not suffice or for cross checking, the same subroutine with eight steps (`ICS8`) can be called instead of the faster `ICS`. A complete survey of variables is presented in App. D.

2.5.2 A greedy possibility: quadruple precision

Not every compiler[5] allows for extended or quadruple precision. This is a feature outside of the ANSI FORTRAN77. Even if *a*ĩTALC contains some code's implementations outside the ANSI syntax, they are quite common (e.g. using underscore, or double complex), so decent compilers won't complain.

But on high level computation, or simply as comparison, switching on the quadruple precision could stabilize numerical results or render amazing agreement with other calculations. Next is the roadmap for an activation of this possibility:

1. Have a look in your working process. Open the file called `Fortran.mk` and look to the first lines, where `#QUADRUPLE` is firstly mentioned. If the right hand side is set to `"yes"`, then you can try to uncomment it and see what happens.

2. Check whether your system has access to the quadruple precision LOOPTOOLS libraries (`libooptoolsQ.a`). In case they are not in your path, try to compile them by looking to the `qprec/` directory in the LOOPTOOLS distribution and following the instructions there.

3. Allow for longer printouts inside of the last formats statments in `main.f`. With this you can access to the sector of digits between 16 and 33, where the quadruple precision play a role.

4. Keep calm since this feature is outside of the automation chain, and therefore some minor changes are required in `Makefile`s and so... Be aware of the screen and file logs!

2.6 Examples including loop level

To finish this descriptive chapter of how to work, we would like to present the other examples available with the distribution. They can be tested in a similar way as µ-pair production. Just

[5] Until now, the use of quadruple precision has been only positively tested under INTEL FORTRAN compiler (see `http://www.intel.com/software/products/compilers/flin/noncom.htm`).

for reference, the default numerical result can be viewed in Tabs. 2.3−2.4. A bit of the physics behind of the examples here given might be found on [LR04] and the references therein.

With the purpose of giving flavour of timings, expected to take during the execution, all the examples were studied on two different machines, representing the today's typically available computers. The results are shown on Tab. 2.2. There we can see that when the size of the produced FORTRAN code begin to be large (i.e. more than ∼200kB for a single subroutine to be compiled), then the compiling time for the GNU compiler gets extremely long, in comparison with the magnitude of the other processes.

	muon_production		leLe.bS		bhabha_QED	
Module	Desktop	Laptop	Desktop	Laptop	Desktop	Laptop
tree	9”	3”	4”	1”	30”	9”
loop	–	–	2’20”	52”	37”	12”
fortran	15”	4”	4:35’56”	33”	26”	8”
Total	24”	7”	4:38’20”	1’26”	1’33”	29”

Table 2.2: Typical timings for the different modules. The technical specifications for system Laptop are: Intel Centrino 1.5GHz cpu, 512MB RAM, Intel icc and ifort version 8.1 compilers. For Desktop: Intel Pentium III 853MHz cpu, 256MB RAM, GNU gcc and g77 version 3.3.3 compilers.

✎ Exercise: Repeat in another directory the example of leLe.bS, in which you recreate the process $e^-e^+ \to b\bar{s}$, but this time neglecting analytically the terms proportional to the electron mass.
a) How many diagrams can you avoid to be generated?
b) How large is the numerical difference for the default differential cross sections?
c) How much time did you save on the compilation of the fortran module?
d) Which from both codes is more stable against UV-parameter check?

☆ Hint: Make use of the overriding possibilities with the files
$AITALCHOME/diana/prg/runningrestriction.prg,
$AITALCHOME/form/kitFORM3/Neglectedmasses.inc
and the llongoutput variable in main.f.

```
#=============================================================================================================================================
# aITALC Version 1.0 (29.10.2004) by A.Lorca -- T.Riemann
#---------------------------------------------------------------------------------------------------------------------------------------------
#sqrtsman      cost      dcs(Born)             dcs(loop)             dcs(soft)             dcs(Born+loop+soft)   dcs(loop^2)
 0.5000000E+03 -.9000000 0.523873625293772E+00 0.104514245255543E+00 -.219313307956275E+00 0.409074562593040E+00 0.923544633715657E-02
 0.5000000E+03 -.5000000 0.611600720838859E+00 0.176786637556097E+00 -.292280088588001E+00 0.496107269806955E+00 0.172748338575392E-01
 0.5000000E+03 0.0000000 0.117253629320705E+01 0.397406280604550E+00 -.606584657480458E+00 0.963357916331143E+00 0.420671139611335E-01
 0.5000000E+03 0.5000000 0.550440648756388E+01 0.212664263210930E+01 -.306463515349883E+01 0.456641396617436E+01 0.240547593782348E+00
 0.5000000E+03 0.9000000 0.189118378731798E+03 0.870334082480635E+02 -.116499974637966E+03 0.159651812341895E+03 0.108965601218541E+02
#=============================================================================================================================================
#Energy given in GeV, Cross sections in pb, setfracomega= 0.1000000
```

Table 2.3: File `main.log`, for the `bhabha_QED` example and `llongoutput=.true.`

```
#=============================================================================================================================================
# aITALC Version 1.0 (29.10.2004) by A.Lorca -- T.Riemann
#---------------------------------------------------------------------------------------------------------------------------------------------
#sqrtsman      cost      dcs(Born)             dcs(loop)             dcs(soft)             dcs(Born+loop+soft)   dcs(loop^2)
 0.5000000E+03 -.9000000 0.000000000000000E+00 0.000000000000000E+00 0.000000000000000E+00 0.000000000000000E+00 0.653411795813974E-08
 0.5000000E+03 -.5000000 0.000000000000000E+00 0.000000000000000E+00 0.000000000000000E+00 0.000000000000000E+00 0.709542147963438E-08
 0.5000000E+03 0.0000000 0.000000000000000E+00 0.000000000000000E+00 0.000000000000000E+00 0.000000000000000E+00 0.104868950766147E-07
 0.5000000E+03 0.5000000 0.000000000000000E+00 0.000000000000000E+00 0.000000000000000E+00 0.000000000000000E+00 0.208494230993482E-07
 0.5000000E+03 0.9000000 0.000000000000000E+00 0.000000000000000E+00 0.000000000000000E+00 0.000000000000000E+00 0.464395594522522E-06
#=============================================================================================================================================
#Energy given in GeV, Cross sections in pb, setfracomega= 0.1000000
```

Table 2.4: File `main.log`, for the `leLe.bS` example and `llongoutput=.true.`

Chapter 3

Hot potatoes in the scope

At the end of the day one expects the results to be correct, but how to proceed when we have done some intuitive manipulations not very well (or not at all) described before?

The following pretends to clarify some hot points, historically considered without no difficulties, and the way that were implemented in *a*ITALC.

1. **Renormalization**: Since the renormalization of the parameters and fields is model dependent, it was our solution to implement a fixed FORTRAN subroutine with the required elements in any $2 \to 2$ process with only external fermions. Therefore, whenever one makes modifications on the process settings on the DIANA or FORM modules, possible violations can arise. A typical example of this is to neglect any photon field to generate a gauge independent subset of pure weak diagrams. In such a case, be sure you also comment out any photon contribution inside of the `renormalization.f` or `renormalization_QED.f` files (indeed that is a tough job!).

2. **Running with a finite gauge boson width**: This feature is strictly going beyond a classical one-loop calculation, but still needed if one requests an accuracy of $\mathcal{O}(\alpha)$ of the corrections to the tree-level results with dressed propagators. The algorithm followed here is quite straightforward, based on the fix-width scheme

   - Self-energies of the finite width gauge boson are discarded to avoid double-counting
   - Complex masses are taken into account where needed in propagators and loop integrals to avoid singularities or discontinuities.

3. **Stability at the kinematical edges**: Experience taught us that in the limits, $\cos\theta \to \pm 1$, large numerical cancellations between different topologies and their loop integral values will occur. Thus, it is strongly advised to perform ultraviolet checks in your region of study, that will inform you how strong or weak is the stability of the numbers. Moreover, when computing integrated cross sections in that limits, the error of the calculation will never be better than the limit of precision shown by the differential cross section itself.

4. **Treatment of CKM-type couplings**: We adopt the solution of not implementing counterterms for the CKM elements. This has some limitation on charge current processes but, for neutral current ones, correct expressions can be retrieved by considering actively the loops on the external legs (usually discarded with the `onshell` option of QGRAF). You can have a look to the `leLe.bS` in the `examples/` directory to see how this is done.

5. **Extensions beyond 2 $\to$ 2 fermion processes**: They should be dealt carefully. There is plenty of code that can still be useful in a treatment of processes with more external legs or computation of simple self-energies (as it was done for the counterterms). Nevertheless, it requires a deep monitoring and understanding of the physics and technicalities behind. Then you would better forget about the FORTRAN module and run individually the MAKE commands in an analytical level by adding your own FORM programs or DIANA functions. But this is only a hint, the world of possibilities is then in your hands.

Acknowledgements

A.L. would like to thank Nils Christian for his endless lessons on CVS, autoconf, and many other UNIX tools. Without his support this project would have never become a reality.

The authors also express their gratitude to Misha Tentyukov and Jochem Fleischer for both, technical and scientific suggestions and critical reading of the tutorial.

This work was partially supported by the European's 5-th Framework under contract HPRN–CT–2000–00149 Physics at Colliders and by the Deutsche Forschungsgemeinschaft under contract SFB/TR 9–03.

Appendix A

References for the package DIANA

For the sake of completeness, we would like to extend the referencing of any potential user further interested in DIANA.

Tutorials

The perhaps slightly outdated web site of DIANA still contains most of the available information:

- `http://www.physik.uni-bielefeld.de/~tentukov/diana.html`.

Alternative access to the documents is found at our mirror under

- `http://www-zeuthen.desy.de/theory/aitalc/downloads/doc/diana.ps`
- `http://www-zeuthen.desy.de/theory/aitalc/downloads/doc/diana2.ps`

for the first and second versions respectively. Additionally there is a draft describing extra features like the automatization of momenta distributions at
`http://www.physik.uni-bielefeld.de/~tentukov/topoltables.ps.gz`.

Articles

Besides the tutorial and log files, the user may find interesting the following collection of articles and contributions [TF99], [FT00], [TF03], [FTT03], [TF04], [GLR04] and [FLR04]. They compile somehow a long heritage of the code's evolution.

Contact

In case you want to use DIANA independently of *a*i̊TALC, it might be sometimes useful to contact its authors. They provided us with useful comments and wise guidance in our adventure.
`mailto:tentukov@physik.uni-bielefeld.de`, `fl@physik.uni-bielefeld.de`.

Appendix B

Implemented models

Code	Arguments	Particle	Name	Lorentz type
Lepton				
`le`, `Le`	`(;`p_i`)`	e^-, e^+	electron, positron	fermion
Gauge boson				
`A`	`(`μ_i`;`p_i`)`	γ	photon	boson

Table B.1: Particle content for the QED model (`QED.model`)

Code	Arguments	Particle	Name	Lorentz type
Leptons				
`ne , Ne`		ν_e, $\bar{\nu}_e$	neutrino-e, antineut-e	
`nm , Nm`	`(;`p_i`)`	ν_μ, $\bar{\nu}_\mu$	neutrino-μ, antineut-μ	fermion
`nt , Nt`		ν_τ, $\bar{\nu}_\tau$	neutrino-τ, antineut-τ	
`le , Le`		e^-, e^+	electron, positron	
`lm , Lt`	`(;`p_i`)`	μ^-, μ^+	muon, antimuon	fermion
`lt , Lt`		τ^-, τ^+	tau, antitau	
Quarks				
`u , U`		u, $\bar{u}$	up, antiup	
`c , C`	`(;`p_i`)`	c, $\bar{c}$	charm, anticharm	fermion
`t , T`		t, $\bar{t}$	top, antitop	
`d , D`		d, $\bar{d}$	down, antidown	
`s , S`	`(;`p_i`)`	s, $\bar{s}$	strange, antistrange	fermion
`b , B`		b, $\bar{b}$	bottom, antibottom	
Gauge bosons				
`A`		γ	photon	
`Z`	`(`μ_i`;`p_i`)`	Z	Z-boson	vector
`Wm , Wp`		W^-, W^+	W-boson	
Higgs sector				
`H`		H	Higgs	
`G0`	`(;`p_i`)`	χ	"the would-be	scalar
`Gm , Gp`		ϕ^-, ϕ^+	Goldstone bosons"	
Faddeev-Popov ghosts				
`ghA, GhA`		η_γ, $\bar{\eta}_\gamma$	photon ghosts	
`ghZ, GhZ`	`(;`p_i`)`	η_Z, $\bar{\eta}_Z$	Z-ghosts	scalar
`ghm, Ghm`		η_{W^-},$\bar{\eta}_{W^-}$	W^--ghosts	(with fermion statistics)
`ghp, Ghp`		η_{W^+}, $\bar{\eta}_{W^+}$	W^+-ghosts	

Table B.2: Particle content for the electroweak sector in the Standard Model (`EWSM.model`)

Appendix C

List of FORM procedures in KITFORM3

Name	Args.	Description
`LTtoPV`		Translates the indexing of loop integrals from [Den93] into [tHV79] convention.
`aitalcnotation`	N	Translates the FORM code into FORTRAN readable. The argument $N = 1$ allows for complex masses while $N = 0$ does not
`analyzeterms`		Part of `simplifygrams`
`argsymmetries`		Applies symmetry properties on arguments for loop functions and Gram determinants
`canceldens`		To cancel composition of `den(a,b)*(a-b)=1`
`chisholm`		Applies Chisholm identities by converting all chains of γ matrices that are contracted by their dimensional value
`contgammas`		Simplifies contractions in products of gamma matrices
`contractepsilon`		It contracts the product of two ϵ tensors
`ctfeynmanrules`		Insertion of Feynman rules with counterterms
`definegrams`		Defines a FORM global variable with the explicit expression of a Gram determinant
`derivateself`		Derivates self energies functions
`dimensionfour`		To reach the limit $D \to 4$ taking care of the UV-behaviour of scalar integrals
`diracequation`		Applies Dirac equation to available spinors
`dummytovar`	N	Part of `simplifygrams`. N stands for the amount of arguments in `dummyN` function
`equalindex`		Gives the same index to potentially different contractions keeping them ordered by number

continued on next page

continued from previous page

Name	Args.	Description
`externalmomenta`		To substitute the internal momenta in loops with external ones
`gamma3to1`		Introduces an ϵ tensor and γ_5 to remove the product of three γ's
`gammaalgebra`		Prepares γ chain structures for `diracequation`
`gammamunu`		Introduces indices in the vector structures
`gramback`	N	Part of `simplifygrams`. Returns the Gram determinant of order N if no simplification was found
`gramsubstitution`	N, w	It substitutes the w-element to match posssible order N Gram determinants
`identifyintegrals`		Picks up the different loop integrals and provides a set of FORM `$`-variables with the right substitutions, inserting the function `LoopIntegral=LI(n)`
`integrationde93`		Procedure to integrate 4,3,2,1-point functions with internal momenta decomposition like in [Den93]
`invgammamunu`		Undoes the `gammamunu` effects
`keepUVloops`		Keeps only integrals actually UV divergent
`loopsymmetries`		Minimize loop integrals in boxes by symmetrizing arguments
`lorentzinvariants`		Adapts scalar product of vectors to Lorentz invariant squared masses
`massconvention`		To use single variables with M instead of MM outside arguments
`massivefofa`	j, c, N	Calculates and saves the formfactors as [Lor05]. `j` stands for topology, `c` for left-right or $\mathbb{1}$-γ_5 and `N=1(0)` do (not) print the formfactors
`movepslash`		Places the desired $\not{p}_i$ to the side of fermion chain to apply `diracequation`
`neglectmass`	`zero`	Neglects positive powers of terms as indicated in the file `Neglectedmasses.inc`. `zero` is the substituted value
`nosymboliccouplings`		To put explicit values of charges and weak couplings
`onshell`		Substitutes external momenta for external particle masses and Mandelstam variables
`pslashaway`		Reorder away the non desired position of $\not{p}_i$
`pushgamma5`		Pushes γ_5 to the right in fermion chains
`pushomegas`		Orders and pushes Left or Right projectors ($\omega_{L,R}$) to the right in fermion chains

continued on next page

continued from previous page		
Name	Args.	Description
`recoverargumentsde93`		Catchs back the arguments of loop integrals in combination with `integrationde93`
`reductionDD02`	N	Reduction algorithm from scalar tensor integrals to master integrals given in [DD03]. $N = 1$ sorts after each integral type, $N = 0$ does not sort at all
`reductionLT`		`reductionDD02(1)` with LOOPTOOLS notation
`simplifygrams`		Tries to remove inverse Gram determinants by looking to the numerator (time and memory consuming!)
`simplihelp`	N, r	It helps `simplifygrams` to run sequentially r times from the highest power of one variable for N order Gram determinants
`storeself`	j, c, N	Stores self energies, `j` stands for topology, `c` for left-right or $\mathbb{1}$-γ_5 and `N=1(0)` do (not) print the form factors
`threegammastoepsilon`		Introduces and removes the ϵ tensor simplifying chains of 3 γ matrices that naturally don't dissappear after `diracequation`
`tracefermiloops`		Traces possible fermion loops in self-energies
`transvorlongit`	x	Sets the longitudinal and transverse part of self energies, keeping off-shell but with mass^2 argument x
`unityCKM`		Makes $\mathrm{CKM}_{ij} = \delta_{ij}$
`usegamma5`		Uses γ_5, instead of $\omega_{L,R}$ projectors
`useomegas`		The opposite to `usegamma5`
`vartodummy`		Part of `gramback`, opposite as `dummytovar`

Appendix D

Settings for `main.f`

Variable	Type	Default Value	Meaning
Kinematical			
`icostloop`	integer	`5`	Dimension of `setcostarray`
`isqrtsloop`	integer	`5`	Dimension of `setsqrtsmanarray`
`setcostarray`	double prec.	`data //`	Set of default values for the `setcost` ($\cos\theta$) do loop in the `dcs` evaluation
`setsqrtsmanarray`	double prec.	`data //`	Set of default values for the `setsqrtsman` ($\sqrt{s}$) do loop in the `ICS` evaluation
`setfracomega`	double prec.	`0.1d0`	Limit for the maximum soft photon energy
`setlimcost`	double prec.	`0.9999d0`	Limit of integration for `ics` and `fba`
Process flags			
`lrenorm`	logical	`.true.`	Call the renormalization subroutine to calculate the counterterms parameters
`lwidth`	logical	`.false.`	Call the loop integrals with complex values for boson masses. For the four-point integral, only the case with one massless boson (photon) is considered.
`lidentCKM`	logical	`.false.`	Brings the CKM mixing matrix in a diagonal form, so no mixing occurs
Checking flags			
`luvcheck`	logical	`.false.`	Perform a shift in the `mudim` and `delta` parameters in LOOPTOOLS
`lircheck`	logical	`.false.`	Perform a shift in the `lambda` parameter in LOOPTOOLS

continued on next page

continued from previous page			
Variable	Type	Default Value	Meaning
Output flags			
lcostloop	logical	.true.	Performs a do-loop over setcost
lprintics	logical	.false.	Performs a do-loop over setsqrtsman and prints the results of ics
lprintfba	logical	.false.	Performs a do-loop over setsqrtsman and prints the results of fba
llongoutput	logical	.false.	Prints the cross sections in double precision format

Appendix E

Log of changes to previous version

E.0.1 Changes in 1.3.1 to version 1.3

26.09.06 FORM files with `#write` sentence and `configure.in`: Bug fixing for 64bit machines (AMD Opteron and EM64T tested), and new DIANA (2.37). TEDI path also corrected. Variable `$HOST` changed to `‘uname -n‘`.

E.1 Changes in 1.3 to version 1.2

04.09.06 `configure.in makefile.in ...`: Major modifications, removing all hand-made 1-,2- and 3-point functions with complex masses in favour of LOOPTOOLS ones. Final compilation.

05.07.05 `configure.in`: Several improvements and adaptations to new versions of DIANA (2.36) and QGRAF (3.1)

04.07.05 `diana/prg/analyze_looptype.prg`: Exported variable `looptypenumber` is given an integer (0) for the `twoloops` case, avoiding error.

04.07.05 `examples/Default/Level.mk.in`: `$(USER)` and `$(HOST)` variables are initialized within make, avoiding possible empty strings.

E.2 Changes in 1.2 to version 1.1

14.12.04 `form/kitFORM3/joinkinematics.frm`: Closing quote mark (") was absent in a comment line when creating the subroutine `dcs.f` for the case of identical particles in final state. Fixed

13.12.04 `diana/prg/particleaspect.prg`: Nicer postscript fonts with Times-Italic instead of Times-Roman[1].

13.12.04 `examples/Default/parameterlist.hf`: Width for the Higgs boson adequated to theoretical expectation.

08.12.04 `fortran/src/d0wdd0.f`: `d0wdd0` function improved for the case of two equal complex numbers giving quotient unity.

E.3 Changes in 1.1 to version 1.0

07.12.04 `fortran/src/cden.f` Function improved to correctly cancel infrarred divergencies with the given `D0w` and `C0w` fixed-width implementation.

07.12.04 `form/kitFORM3/aitalcnotation.prc`: The translation of real mass variables (`MM`s) into complex ones (`cMM`s) did not replace more than once the same variable, leading to inconsistent calls for the `D0w` and `C0w` loop functions for complex masses. This is fixed and tested.

07.12.04 `configure.in`: The command '`which gmake`' does not always return the GNU-MAKE location. This is fixed.

23.11.04 `Makefile.in`: Automatic installation of LoopTools fails when the FORTRAN compiler requires no optimization (`-O0` option) to properly run the FF routine `ffinit`. Now `-O0` is default.

[1] See for instance the standard postscript fonts under
`http://www.tac.dk/cgi-bin/info2www?(fontname.info)Standard%20PostScript%20fonts`.

Appendix F

GNU Free Documentation License

Version 1.2, November 2002
Copyright ©2000,2001,2002 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

F.1 APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The **"Document"**, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as **"you"**. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A **"Modified Version"** of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A **"Secondary Section"** is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding

them.

The **"Invariant Sections"** are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The **"Cover Texts"** are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A **"Transparent"** copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called **"Opaque"**.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The **"Title Page"** means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section **"Entitled XYZ"** means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as **"Acknowledgements"**, **"Dedications"**, **"Endorsements"**, or **"History"**.) To **"Preserve the Title"** of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

F.2 VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

F.3 COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

F.4 MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

F.5 COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

F.6 COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

F.7 AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that

bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

F.8 TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

F.9 TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

F.10 FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.
See `http://www.gnu.org/copyleft/`.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright ©*YEAR YOUR NAME*.
> Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with . . . Texts." line with this:

> with the Invariant Sections being *LIST THEIR TITLES*, with the Front-Cover Texts being *LIST*, and with the Back-Cover Texts being *LIST*.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Bibliography

[DD03] A. Denner and S. Dittmaier, *Reduction of one-loop tensor 5-point integrals*, Nucl. Phys. **B658**, 175–202 (2003), hep-ph/0212259.

[Den93] A. Denner, *Techniques for calculation of electroweak radiative corrections at the one loop level and results for W physics at LEP-200*, Fortschr. Phys. **41**, 307–420 (1993).

[FLR04] J. Fleischer, A. Lorca and T. Riemann, *Automatized calculation of 2-fermion production with* DIANA *and a*ĩTALC, To appear in the proceedings of International Conference on Linear Colliders (LCWS 04), Paris, France, 19-24 Apr (2004), hep-ph/0409034.

[FLRW03] J. Fleischer, A. Leike, T. Riemann and A. Werthenbach, *Electroweak one-loop corrections for e^+e^- annihilation into $t\bar{t}$ including hard bremsstrahlung*, Eur. Phys. J. **C31**, 37–56 (2003), hep-ph/0302259, Code reachable via `http://www.ifh.de/~riemann/doc/topfit/topfit.html`.

[FT00] J. Fleischer and M. Tentyukov, *A Feynman diagram analyser* DIANA*: Graphic facilities*, Contribution to the proceedings of 7th International Workshop on Advanced Computing and Analysis Techniques in Physics Research (ACAT 2000), 193-195, Batavia, Illinois, 16-20 Oct (2000), hep-ph/0012189.

[FTT03] J. Fleischer, M. Tentyukov and O. V. Tarasov, DIANA *and selected applications.*, Nucl. Phys. B (Proc. Suppl.) **116**, 348–352 (2003), hep-ph/0211209.

[GLR04] J. Gluza, A. Lorca and T. Riemann, *Automated use of* DIANA *for two-fermion production at colliders*, To appear in the proceedings of 9th International Workshop on Advanced Computing and Analysis Techniques in Physics Research (ACAT 03), Tsukuba, Japan, 1-5 Dec 2003. (2004), hep-ph/0409011.

[HPV99] T. Hahn and M. Perez-Victoria, *Automatized one-loop calculations in four and D dimensions*, Comput. Phys. Commun. **118**, 153–165 (1999), hep-ph/9807565, LoopTools available at `http://www.feynarts.de/looptools`.

[Lor05] A. Lorca, *Calculation and automatization of higher order corrections to 2 to 2 fermion processes*, PhD thesis, Universität Bielefeld, Bielefeld, Germany, 2005, In preparation.

[LR04] A. Lorca and T. Riemann, *Automated calculations for massive fermion production with a*ÅITALC, Nucl. Phys. B (Proc. Suppl.) **135**, 328–332 (2004), hep-ph/0407149.

[Nog] P. Nogueira, *An introduction to Qgraf 2.0*, Available at `http://cfif.ist.utl.pt/~paulo/qgraf.html`.

[PFTV92] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes in FORTRAN: The Art of Scientific Computing, 2nd ed.*, chapter 4.3 Romberg integration, and 16.4 Richardson Extrapolation and the Bulirsch-Stoer Method, pages 134–135 and 718–725, Cambridge University Press, Cambridge, U.K., 1992.

[TF99] M. Tentyukov and J. Fleischer, DIANA*, a program for Feynman diagram evaluation*, Talk given at 6th International Workshop on New Computing Techniques in Physics Research: Software Engineering, Artificial Intelligence Neural Nets, Genetic Algorithms, Symbolic Algebra, Automatic Calculation (AIHENP 99), Heraklion, Crete, Greece, 12-16 Apr (1999), hep-ph/9905560.

[TF00] M. Tentyukov and J. Fleischer, *A Feynman diagram analyser* DIANA, Comput. Phys. Commun. **132**, 124–141 (2000), hep-ph/9904258, Available at `http://www.physik.uni-bielefeld.de/~tentukov/diana.html`.

[TF03] M. Tentyukov and J. Fleischer, *A Feynman diagram analyzer* DIANA*: Recent development*, Nucl. Instrum. Meth. **A502**, 570–572 (2003), hep-ph/0210179.

[TF04] M. Tentyukov and J. Fleischer, *Parallel computation of Feynman diagrams with* DIANA, Comput. Phys. Commun. **160**, 167–186 (2004), hep-ph/0311111.

[tHV79] G. 't Hooft and M. J. G. Veltman, *Scalar one loop integrals*, Nucl. Phys. **B153**, 365–401 (1979).

[Ver] J. A. M. Vermaseren, *New features of FORM*, math-ph/0010025, Available at `http://www.nikhef.nl/~form`.

[vO91] G. J. van Oldenborgh, *FF: A Package to evaluate one loop Feynman diagrams*, Comput. Phys. Commun. **66**, 1–15 (1991), Available at `http://www.xs4all.nl/~gjvo/FF.html`.