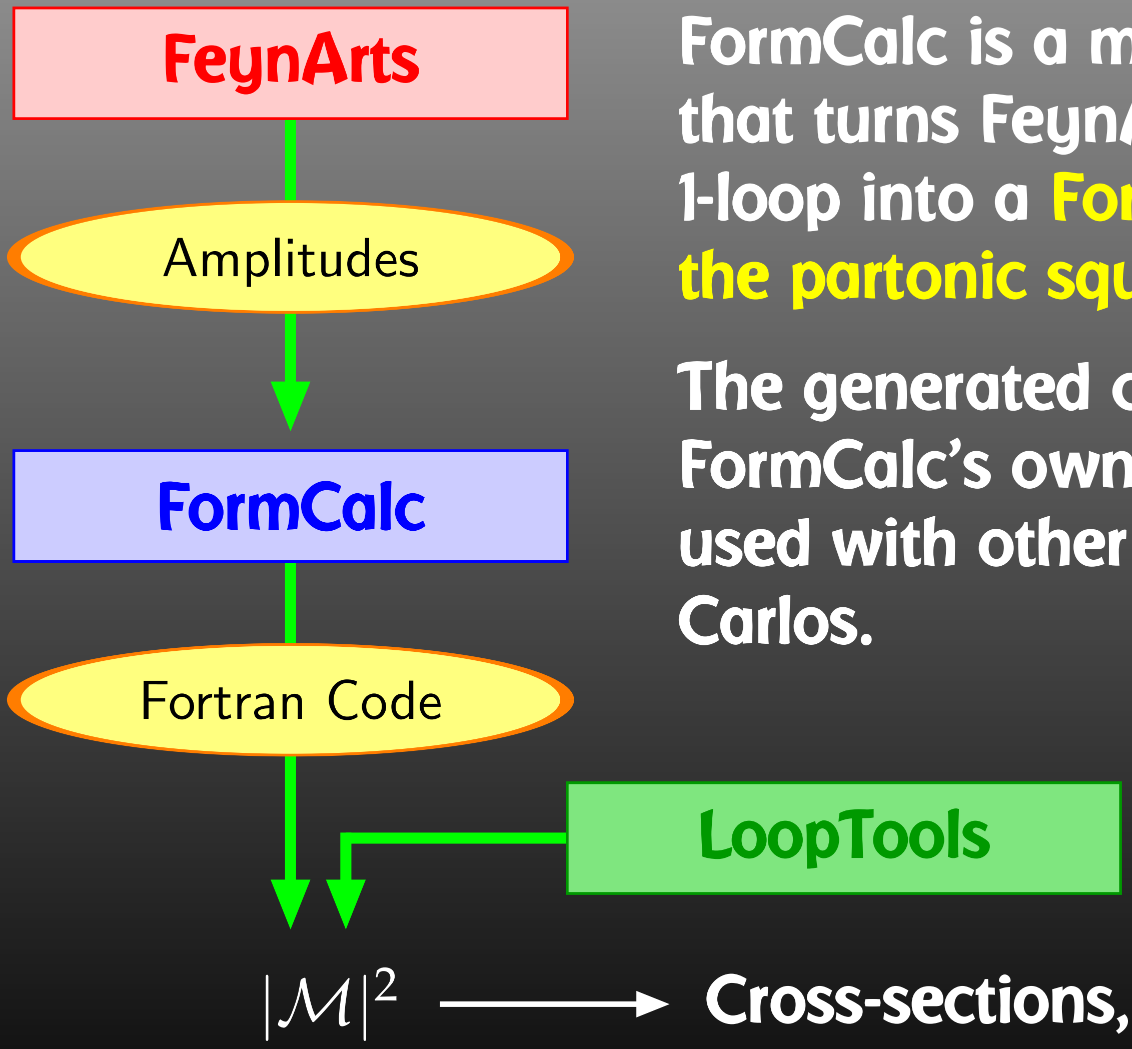
News from FormCalc and LoopTools

Thomas Hahn

Max-Planck-Institut für Physik
München

The System

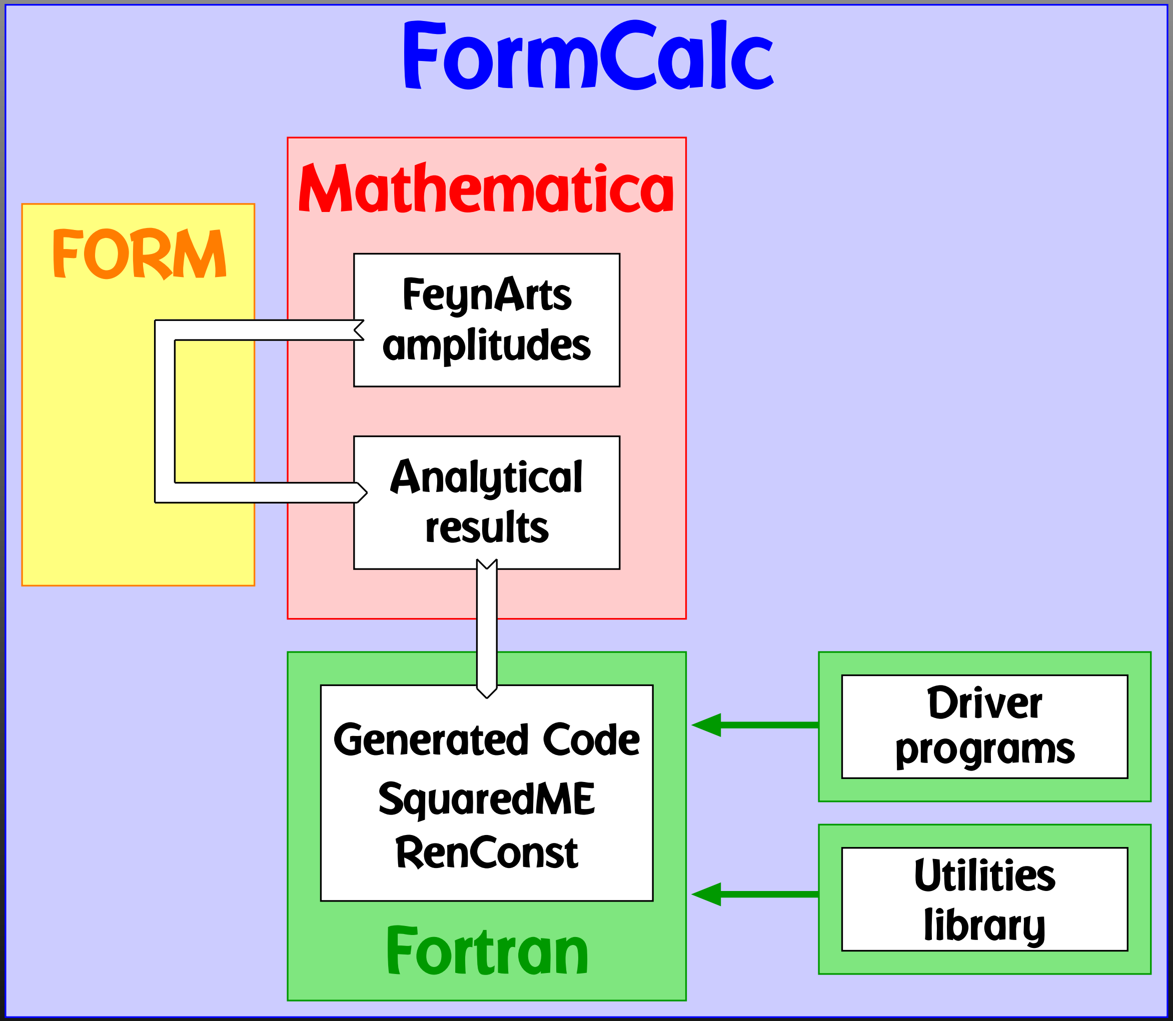


FormCalc is a matrix-element generator that turns FeynArts amplitudes up to 1-loop into a Fortran code for computing the partonic squared matrix element.

The generated code can be run with FormCalc's own driver programs, or used with other 'Frontends', e.g. Monte Carlos.

$|\mathcal{M}|^2$ ⟶ Cross-sections, Decay rates, . . .

FormCalc Internals



FormCalc

Mathematica

FORM

FeynArts amplitudes

Analytical results

Generated Code

SquaredME

RenConst

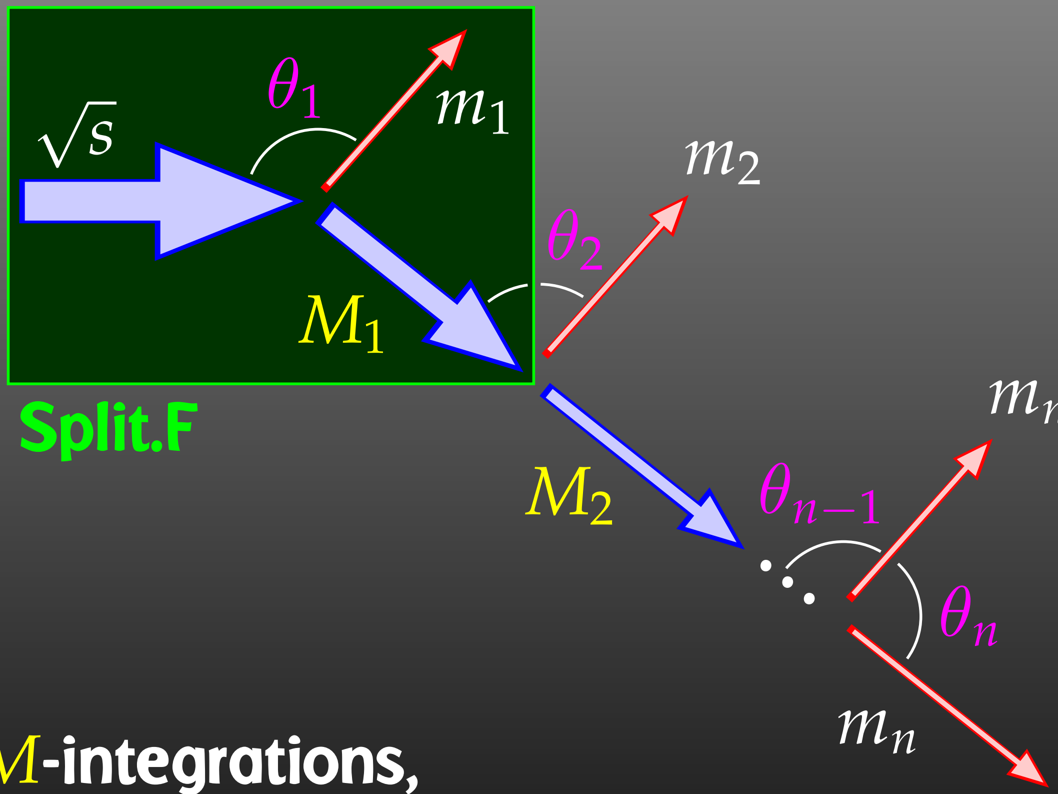
Fortran

Driver programs

Utilities library

New Kinematics

n-particle phase-space is built up iteratively:



$(n-1)$ M-integrations,

n $\cos\theta$-integrations,

n φ-integrations (φ_1 trivial because of axial symmetry)

Phase-space

Explicit phase-space parameterization:

$$\frac{1}{2\sqrt{s}} \int_{m_2+\cdots+m_n}^{\sqrt{s}-m_1} \mathrm{d}M_1\, \mathrm{d}\cos\theta_1\, \mathrm{d}\varphi_1\, \frac{k_1}{2}$$
$$\times \int_{m_3+\cdots+m_n}^{M_1-m_2} \mathrm{d}M_2\, \mathrm{d}\cos\theta_2\, \mathrm{d}\varphi_2\, \frac{k_2}{2}$$
$$\vdots$$
$$\times \int_{m_n}^{M_{n-2}-m_{n-1}} \mathrm{d}M_{n-1}\, \mathrm{d}\cos\theta_{n-1}\, \mathrm{d}\varphi_{n-1}\, \frac{k_{n-1}}{2}$$
$$\times \int \mathrm{d}\cos\theta_n\, \mathrm{d}\varphi_n\, \frac{k_n}{2}$$

k_i **and** $\cos\theta_i$ **are given in the decay's rest frame.**

Verbatim example from 2to3.F

```
   p = 0
   ex = 0
   ey = 0
   ez = 1
   minv = sqrtS
   msum = MASS3 + MASS4 + MASS5

   call Split(5, dble(MASS5), p, ex, ey, ez, minv, msum,
&    fac, key, Var(XMREM5), Var(XCOSTH5), Var(TRIVIAL))

   call Split(4, dble(MASS4), p, ex, ey, ez, minv, msum,
&    fac, 0, Var(FIXED), Var(XCOSTH4), Var(XPHI4))

   call VecSet(3, dble(MASS3), p, ex, ey, ez)
```

The `X` in `XMREM5` refers to the ratio, i.e. `XMREM5` runs from 0 to 1.
The integration borders are determined internally by Split.

Variables

FormCalc 5 contains a new homogeneous system for all (potential) integration variables. Each variable is referred to by a preprocessor constant, e.g. `SQRTS` or `XCOSTH`.

- `Var(`v`)` = the actual value of v.
- `Show(`v`)` = the value printed in the output - to print e.g. t instead of $\cos\theta$.
- `Lower(`v`)`, `Upper(`v`)`, `Step(`v`)` = the lower limit, upper limit, and step width of v. If the step is zero, the cross-section is integrated over v.
- `CutMin(`v`)`, `CutMax(`v`)` = the lower and upper cuts on v.

Two special variables: `FIXED` for fixed values and
`TRIVIAL` for trivial integrations.

Cuts

Split allows to place cuts on each M and $\cos\theta$ integration.
The φ integration is not modified in the present setup.

Cuts restricting M_i

Cut on	Key
M_i	CUT_MREM
E_i	CUT_MREM_E
k_i	CUT_MREM_K
$E_{T,i}$	CUT_MREM_ET
$k_{T,i}$	CUT_MREM_KT
y_i	CUT_MREM_RAP
η_i	CUT_MREM_PRAP

Cuts restricting $\cos\theta_i$

Cut on	Key
$\cos\theta_i$	CUT_COSTH
$\cos\hat\theta_i$	CUT_COSTHCMS
E_i	CUT_COSTH_E
k_i	CUT_COSTH_K

Application of Cuts

```
   key = 0

#ifdef E5MIN
   CutMin(XMREM5) = E5MIN
   key = key + Cut(CUT_MREM_E, CUT_MIN)
#endif

#ifdef COSTH5CUT
   CutMin(XCOSTH5) = -(1 - COSTH5CUT)
   CutMax(XCOSTH5) = +(1 - COSTH5CUT)
   key = key + Cut(CUT_COSTH, CUT_MIN + CUT_MAX)
#endif

   call Split(5, dble(MASS5), p, ex, ey, ez, minv, msum,
&    fac, key, Var(XMREM5), Var(XCOSTH5), Var(TRIVIAL))
   ...
```

Kinematical Information

Kinematical information is available for each external particle:

- `momspec(SPEC_M,`i`)` - **mass** m,
- `momspec(SPEC_E,`i`)` - **energy** E,
- `momspec(SPEC_K,`i`)` - **momentum** k,
- `momspec(SPEC_ET,`i`)` - **transverse energy** E_T,
- `momspec(SPEC_KT,`i`)` - **transverse momentum** k_T,
- `momspec(SPEC_RAP,`i`)` - **rapidity** y,
- `momspec(SPEC_PRAP,`i`)` - **pseudo-rapidity** η,
- `momspec(SPEC_DELTAK,`i`)` - **the difference** $E - k$.

Convolution

With the new variable system, the Convolution with arbitrary PDFs can easily be achieved.
Three modules are already included in FormCalc 5:

- `lumi_parton.F` = initial-state partons
  = no convolution.
- `lumi_hadron.F` = initial-state hadrons
  = convolution with hadronic PDFs
  from the LHAPDF library.
- `lumi_photon.F` = initial-state photons
  = convolution with CompAZ spectrum.

Modularity

As before, the two files `process.h` and `run.F` direct the choice of parameters and include one each of

`1to2.F`	`lumi_parton.F`	`model_sm.F`
`2to2.F`	`lumi_hadron.F`	`model_mssm.F`
`2to3.F`	`lumi_photon.F`	`model_thdm.F`

There is virtually **no cross-talk** between different modules which are in that sense 'universal.'

```
-rw-r--r--   1 hahn users  1501 2005-09-13 08:55 1to2.F
-rw-r--r--   1 hahn users  1652 2005-09-13 08:56 2to2.F
-rw-r--r--   1 hahn users  2153 2005-09-13 08:55 2to3.F

-rw-r--r--   1 hahn users  2003 2005-06-06 15:17 lumi_hadron.F
-rw-r--r--   1 hahn users   829 2005-06-06 15:17 lumi_parton.F
-rw-r--r--   1 hahn users  1353 2005-06-06 15:18 lumi_photon.F

-rw-r--r--   1 hahn users  5944 2005-05-31 09:31 model_sm.F
-rw-r--r--   1 hahn users 21460 2005-10-01 09:43 model_mssm.F
-rw-r--r--   1 hahn users  3647 2005-05-13 13:52 model_thdm.F
```

Code Reusability

The main program in FormCalc 5 only scans the command line and invokes

```
    call ProcessIni(...)
    call ParameterScan(...)
```

All further action is decoupled from the main program and can easily be called from any program. It is thus straightforward to use FormCalc-generated code in own programs.

Planned application:
Mathematica Interface for FormCalc-generated code, i.e. this makes the cross-section available as a Mathematica function.

Example use: Find minimum of cross-section w.r.t. model parameters using Mathematica's `FindMinimum`.

FormCalc Summary

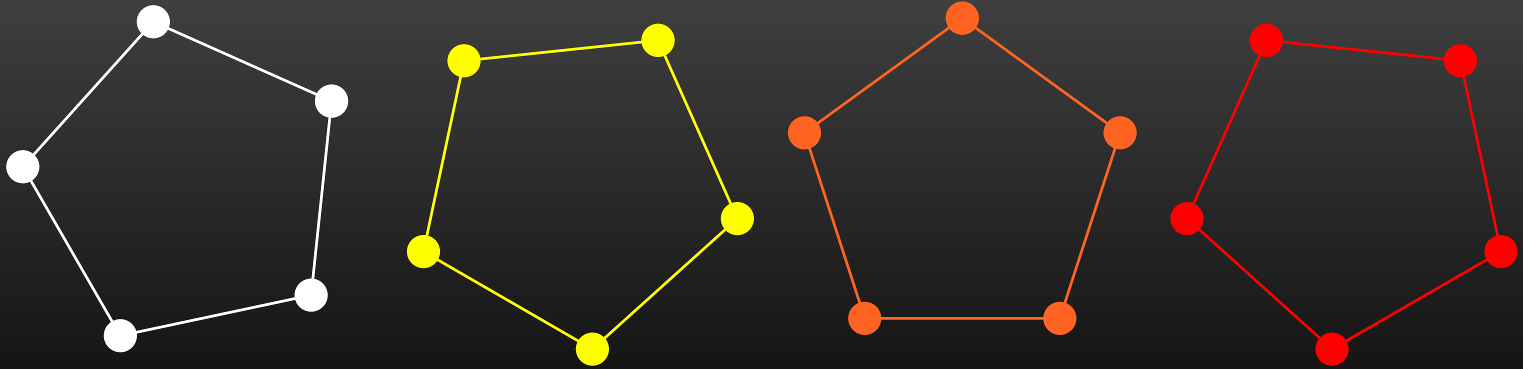
New FormCalc Features:

- **New kinematics routines** can build up arbitrary phase-spaces with a wide range of cuts possible.
- **Better modularity:** almost no cross-talk between modules. Calling FormCalc code from other programs ist straightforward.
- **Convolution** with arbitrary distribution functions included:
  - ▷ partons (no convolution),
  - ▷ hadrons (links with LHAPDF library),
  - ▷ photons (uses CompAZ).
- **Uniform variable system** + detailed kinematical information for all external legs.

LoopTools 2.2

LoopTools is a library for the one-loop integrals. It is based on FF and has a Fortran, C/C++, and Mathematica interface. It is referenced by the FormCalc-generated code.

- The five-point family of functions (scalar + tensor coefficients) have been implemented by M. Rauch, in both the Denner/Dittmaier and the Passarino/Veltman decomposition for comparison.



Bget

The **two-point functions have been united** into the `Bget` function which works similar to its `Cget`, `Dget`, and `Eget` counterparts, in particular it caches its results.

The reason is mainly **Cache Efficiency** in view of the five-point decomposition:

$$\texttt{Eget} \xrightarrow{\text{calls}} 5\,\texttt{Dget} \xrightarrow{\text{call}} (5\cdot 4)\,\texttt{Cget} \xrightarrow{\text{call}} (5\cdot 4\cdot 3)\,\texttt{Bget}$$

Also new: **Cache lookups use a binary search method.**

Compatibility routines for the old `B0`, `B1`, etc. are of course available.

Complex Versions

Versions of the LoopTools functions for Complex Parameters have been added as far as they are contained in FF, i.e. currently only some special cases for the complex D_0 are implemented.

B0i, Bget	CB0i, CBget
C0i, Cget	CC0i, CCget
D0i, Dget	CD0i, CDget
E0i, Eget	CE0i, CEget

Alternate Versions

For some functions alternate versions exist, most of which are based on an implementation by Denner. The user can choose at run-time which version to use, and whether checking is performed. This is determined by the Version Key:

0*key	compute version 'a' (mostly FF),
1*key	compute version 'b' (mostly Denner),
2*key	compute both, compare, return 'a',
3*key	compute both, compare, return 'b'.

Alternate versions are currently available for the following functions: A0, Bget, C0, D0, Eget, CEget.

Example:

```
call setversionkey(2*KeyD0 + 3*KeyBget)
```

Command-line Interface

The new **Command-line Interface** is useful in particular for testing and debugging.

It lists the N-point scalar and tensor coefficients corresponding to the number of arguments, i.e. 3 arguments = B, 6 arguments = C, etc.

```
> lt 500 80.4 91.1887
 p       =   500.0000000000000
 m1      =   80.40000000000001
 m2      =   91.18870000000000
 ====================================================
   FF 2.0, a package to evaluate one-loop integrals
 written by G. J. van Oldenborgh, NIKHEF-H, Amsterdam
 ====================================================
 for the algorithms used see preprint NIKHEF-H 89/17,
 'New Algorithms for One-loop Integrals', by G.J. van
 Oldenborgh and J.A.M. Vermaseren, published in
 Zeitschrift fuer Physik C46(1990)425.
 ====================================================
 bb0     = (-3.162048481416085,1.760724049383876)
 bb1     = (1.594933518284790,-0.8613661011403501)
 bb11    = (-7.269245396361010,-23.04426716231248)
 bb00    = (-1.086060618894423,0.4674785644394271)
 bb001   = (5.154082369023078,11.27351589602220)
 bb111   = (0.8310364075655215,-0.2737898598156348)
 dbb0    = (-3.5481451611823340E-003,3.8420899025032380E-003)
 dbb1    = (1.7079741517372242E-003,-1.9175856430236573E-003)
 dbb00   = (0.2544364496951837,-0.1969437683504615)
 dbb11   = (-7.6724543502407716E-004,1.2584010742473584E-003)

 total number of errors and warnings
 ===================================
 fferr: no errors
```

LoopTools Caches

With the addition of new functions, the internal caching mechanism is now used on:

`Bcache`	`Ccache`	`Dcache`	`Ecache`
`CBcache`	`CCcache`	`CDcache`	`CEcache`

Obviously the former system with `getcachelast` and `setcachelast` is no longer practicable for all those caches.

The new cache-management functions operate on all caches simultaneously:

- `call clearcache` - clears all caches.
- `call markcache` - marks the current position.
- `call restorecache` - reverts to the last marked position.

LoopTools Summary

New LoopTools features:

- Scalar 5-point function plus tensor coefficients available (thanks to M. Rauch).
- Complex versions available as far as implemented in FF.
- Checking can be enabled at run time, individually for all integrals which have alternate versions.
- Extended and improved cache system.
- Command-line interface. Useful for testing and debugging.
- Better internal accuracy in the tensor reduction. Uses LU-decomposition + quadruple precision at strategic points, available with (free) Intel ifort.