

FORM in CompHEP

Slava Bunichev, Moscow State University

in collaboration with A.Kryukov

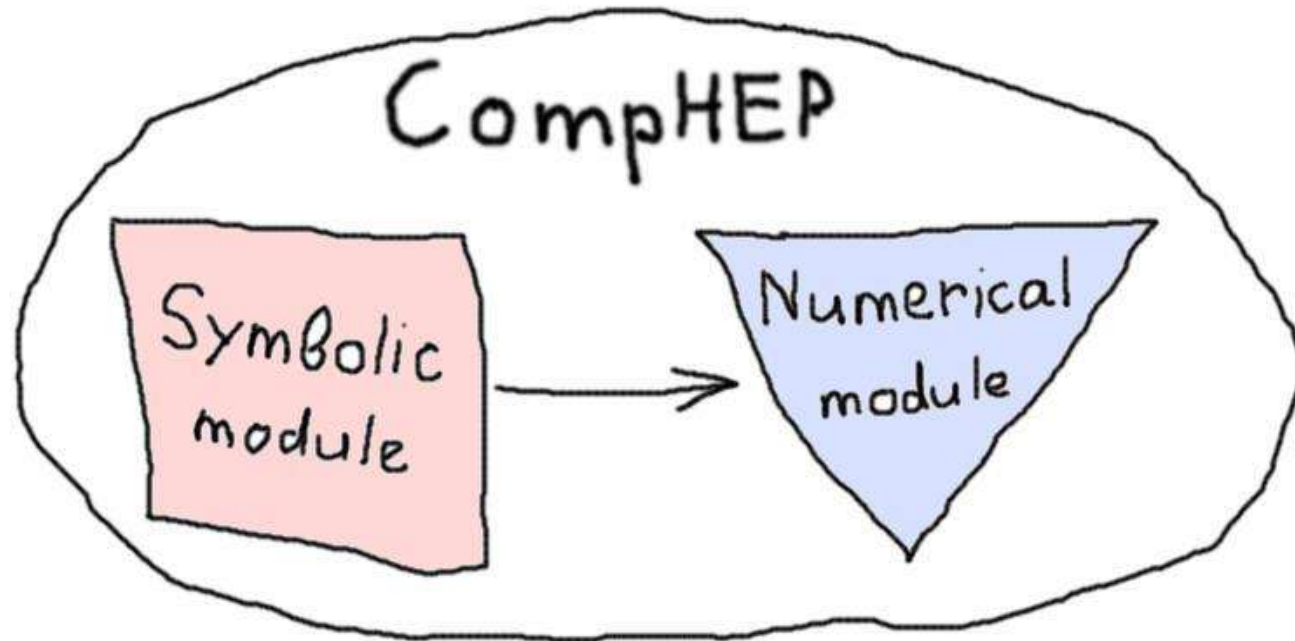
CompHEP

- **CompHEP** is a package for computation of Feynman diagram in quantum field theory from Lagrangian until event flow.
- Main feature of the package is an exact symbolic calculation of tree level diagrams for **Standard Model** and beyond (**2HDM**, **MSSM**, **mSUGRA**, ...).
- **CompHEP** is able to compute processes with many particles (**up to 6**) in the finale state taking into account all the **QCD** and **EW** diagrams, **masses** of fermions and bosons and **width** of unstable particles.
- **CompHEP** has user-friendly **GUI** and highly optimized for fast calculations and effective memory usage.

Price for this approach was **high specialization** of the module for the symbolic calculations. For example, a **fixed number of symbolic structures** allows in symbolic expressions under evaluation.

Such specialization is too heavy for further development of
CompHEP package!

CompHEP structure:



The **symbolic** part includes diagram generator and symbolic calculator.

The **numerical** part includes Monte Carlo integration module and event generator.

We have to change the module for **symbolic** calculations.

FORM

To have more advantage symbolic part of **CompHEP** we incorporate the computer algebra **FORM** for automatic evaluation of squared Feynman diagrams.

We choose the **FORM** because:

FORM is rather fast and good quality computer algebra program,
FORM is a standard de-facto in **HEP** applications.

CompHEP-FORM-CompHEP

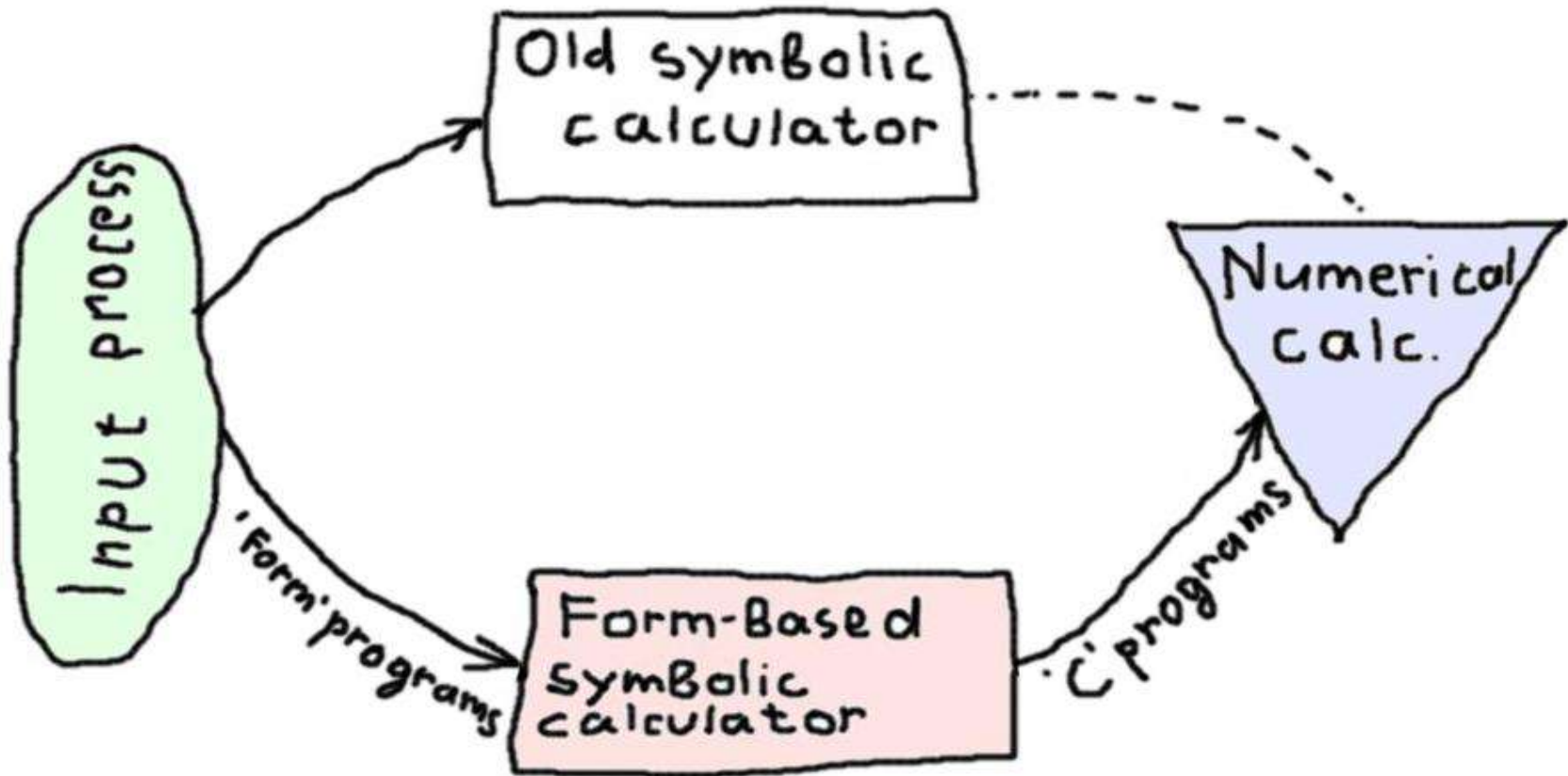
To save the compatibility of old version of **CompHEP** with new one and to minimize programming we decide to incorporate **FORM** in a parallel.

Thus, the new scheme was realized by adding the following modules:

“**form_code.c**” is the FORM code generator

“**procedur.prc**” is a set of auxiliary FORM programs

Scheme of the **CompHEP-FORM-CompHEP** interface

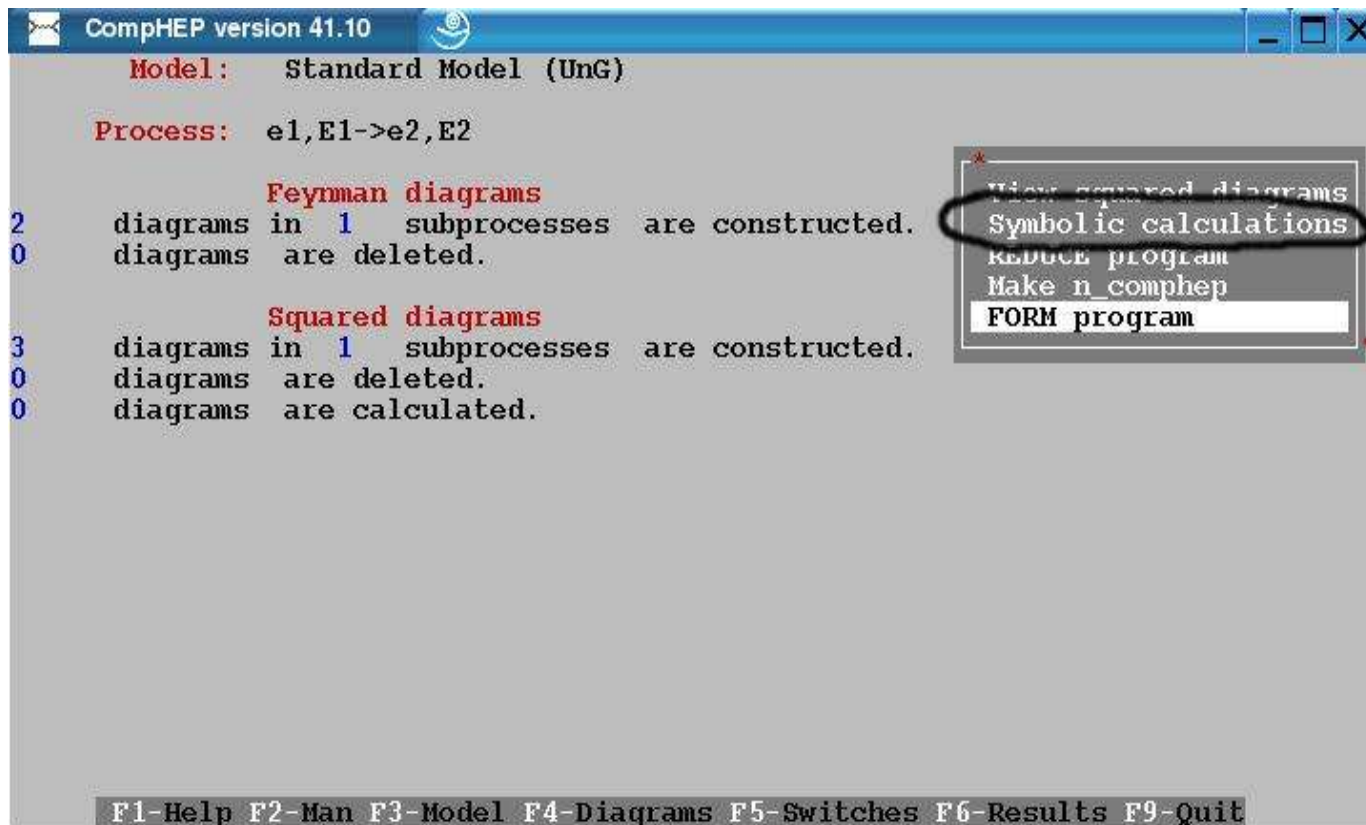


Example of diagrams generating

$$e^+, e^- \rightarrow \mu^+, -\mu^-$$

The image displays two screenshots of the CompHEP software interface, version 41.10, used for generating Feynman diagrams. The top window, titled "CompHEP version 41.10" and "2/2", shows a grid of diagrams. The first diagram in the grid is highlighted and shows a tree-level process where an incoming electron (e1) and positron (E1) annihilate into a virtual photon (A), which then decays into a muon (e23) and an anti-muon (E24). The second diagram in the grid shows a similar process but with a virtual Z boson (Z) instead of a photon. The bottom window, titled "CompHEP version 41.10" and "1/3", shows a more complex diagram. It features two tree-level diagrams side-by-side. The left diagram shows an incoming electron (e1) and positron (E1) annihilating into a virtual photon (A), which then decays into a muon (e2) and an anti-muon (E2). The muon and anti-muon then interact via a t-channel exchange of a muon (e2) and an anti-muon (E2) to produce a final state electron (e1) and positron (E1). The right diagram is similar but uses a virtual Z boson (Z) for the annihilation and exchange. The bottom window also includes a footer with keyboard shortcuts: "F1 - Help, F2 - Man, PgUp, PgDn, Home, End, #, Esc".

Example of symbolic calculations



CompHEP version 41.10

Model: Standard Model (UnG)

Process: e1,E1->e2,E2

Feynman diagrams
2 diagrams in 1 subprocesses are constructed.
0 diagrams are deleted.

Squared diagrams
3 diagrams in 1 subprocesses are constructed.
0 diagrams are deleted.
0 diagrams are calculated.

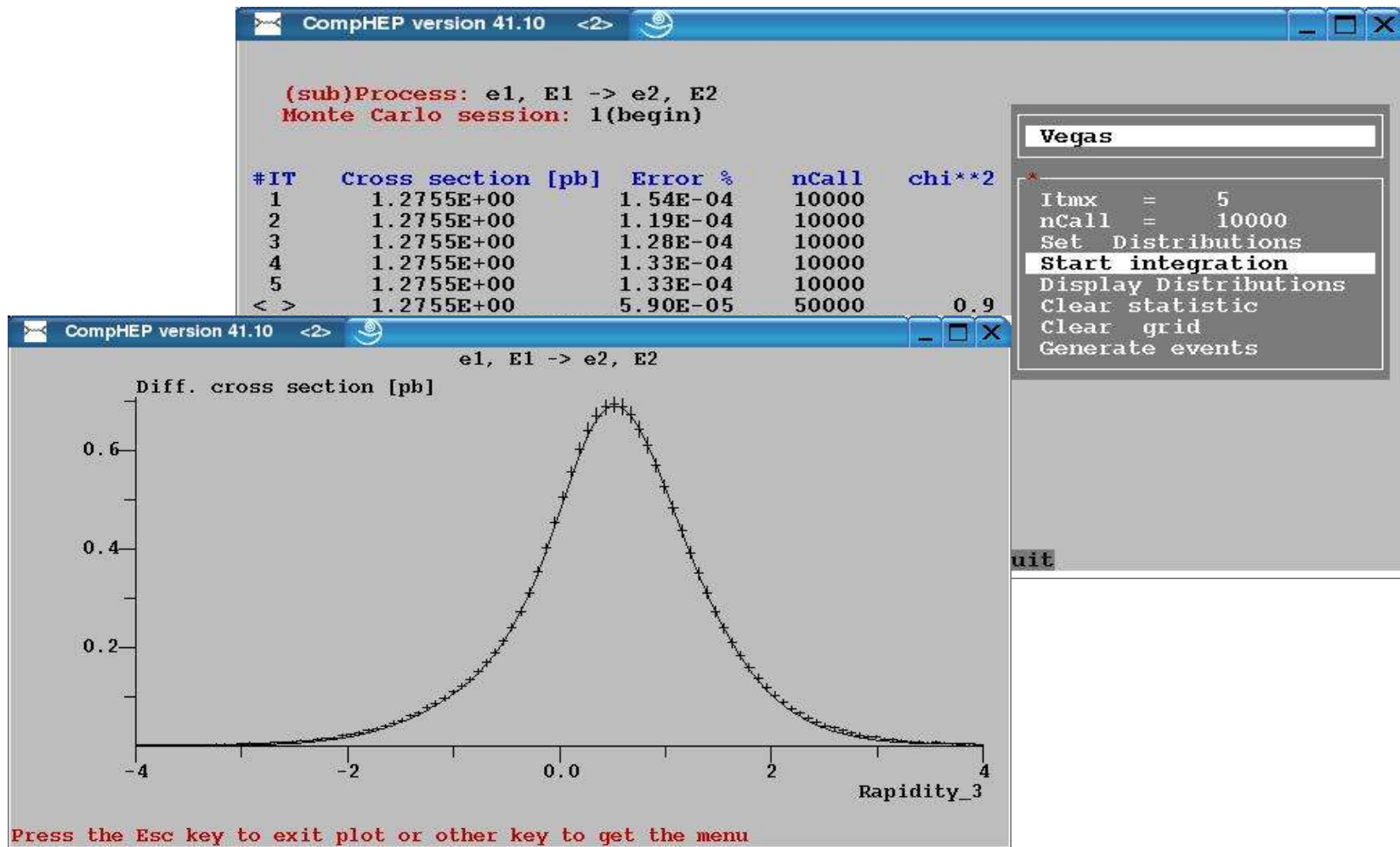
- View squared diagrams
- Symbolic calculations**
- REDUCE program
- Make n_comphep
- FORM program

F1-Help F2-Man F3-Model F4-Diagrams F5-Switches F6-Results F9-Quit

Old calc.

Form-based calc.

Example of numerical calculations



Example of the FORM output

```
#include<math.h>
#define real double
#include"out_ext.h"
#include"out_int.h"
#include"var_def.h"
FNN F1;
real F1(void)
{
real FACTOR,RNUM,DENOM,result;
FACTOR= 1./4.*pow(EE,4);

RNUM= (32*pow(Mm,2) + 32*dp(0) - 64*dp(1))*dp(0) + 64*pow(dp(1),2);

DENOM= 4*pow(dp(0),2);
result=FACTOR*RNUM/DENOM;
return result;
}
```

Optimization

We have to optimize **C**-code for improving calculation time in **numerical part**.

So we introduce the algorithm for reducing number of multiplications in algebraic expressions.

(results of working this algorithm is similar with Horner scheme)

Bu-algorithm

$$3*d1*d2*d3 + 5*d2^2*d3 + 7*d2*d3^2 + 2*d1*d3 + 3*d2 =$$

rename: $d1=d11$, $d1^2=d11*d12$, ...

$$3*d11*d21*d31 + 5*d21*d22*d31 + 7*d21*d31*d32 + 2*d11*d31 + 3*d21 =$$

find most rare multiplier:

$$\begin{aligned} 3*d11*d21*d31 + 5*d21*d22*d31 + 7*d21*d31*(d32) + 2*d11*d31 + 3*d21 = \\ 3*d11*d21*d31 + 5*d21*(d22)*d31 + 7*d21*d31*(d32) + 2*d11*d31 + 3*d21 = \end{aligned}$$

combine terms with similar terms:

$$3*d11*d21*d31 + (5*d22 + 7*d32)*d21*d31 + 2*d11*d31 + 3*d21 =$$

repeat procedure:

$$\begin{aligned} 3*(d11)*d21*d31 + (5*d22 + 7*d32)*d21*d31 + 2*(d11)*d31 + 3*d21 = \\ (3*d11 + 5*d22 + 7*d32)*d21*d31 + 2*(d11)*d31 + 3*d21 = \\ (3*d11 + 5*d22 + 7*d32)*(d21)*d31 + 2*(d11)*d31 + 3*(d21) = \\ ((3*d11 + 5*d22 + 7*d32)*d21 + 2*d11)*d31 + 3*(d21) = \end{aligned}$$

$$((3*d1 + 5*d2 + 7*d3)*d2 + 2*d1)*d3 + 3*d2$$

Conclusion

- We completely incorporate **FORM** in **CompHEP**
- Crosscheck (we get the same results for both calculators)
- We optimize **FORM**-output for numerical calculation

Future plans:

- We plan to implement new algorithms to **increase efficiency** of symbolic calculations
- Perform calculations in theories with **extra dimensions**
- Introduce new complicated structures in the vertices (e.g. **Form-factors**)
- Use the **dimentional regularization**
- Perform **polarized calculations** by introducing the corresponding density matrices for external lines of squared diagrams.
- Include calculations with extention to **1-loop** case
- Incorporate the **gauge invariant classes of diagrams**, **etc.**

