



Implementation of vertex form factors in grid-oriented version of CompHEP

A.Kryukov

(kryukov@theory.sinp.msu.ru)

Skobeltsyn Institute of Nuclear Physics
Moscow State University

Outline

- Introduction
 - CompHEP features
 - N(N)LO correction in CompHEP
- User interface
- Implementation
- Conclusion
 - Using FORM

Introduction

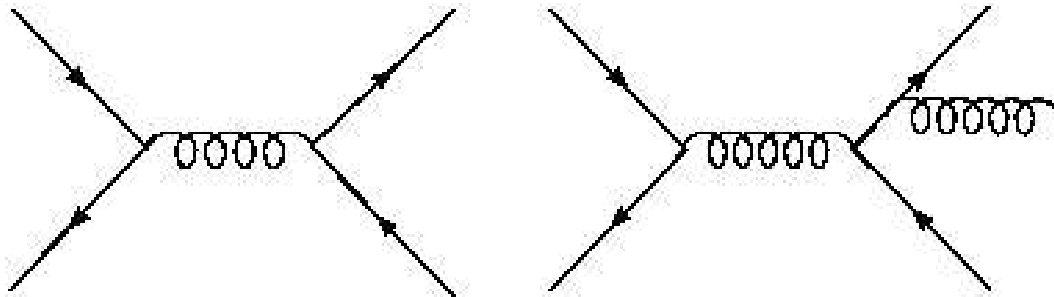
- CompHEP (<http://theory.sinp.msu.ru/comphep>) is a program for calculation of differential cross sections and MC generation of events flow.
- Last version is 4.4.3
- CompHEP Collaboration: E.Boos, V.Bunichev, M.Dubinin, L.Dudko, V.Edneral, V.Ilyin, A.Kryukov, V.Savrin, A.Semenov, A.Sherstnev

Introduction (cont.)

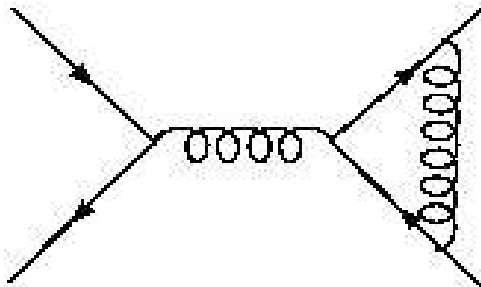
- User friendly interface
- Predefined main particle physics models
- New models defined by users
- Automatic generation of Feynman rules from Lagrangian by LanHEP.
- Symbolic (exact) calculation of differential cross sections (uses the squared matrix element technique)
- MC integration (VEGAS)
- Event flow generator
- Histogramming

N(N)LO correction in CompHEP

- Initial and final state radiation
- Next order of tree calculation

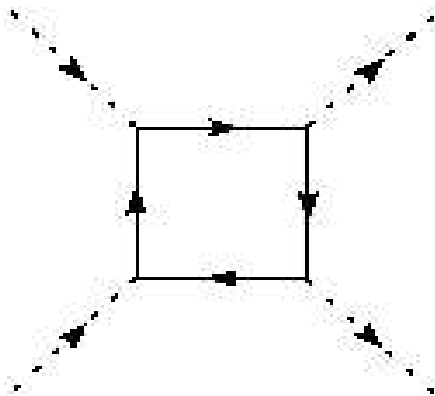


- No loop corrections



Scalar formfactor

- We call scalar formfactor - the function depends on scalar product of incoming momenta:



$$V = f(p_1^2, p_2^2, p_3^2, p_1 \cdot p_2, p_1 \cdot p_3, p_2 \cdot p_3) T$$

where T is tensor (Lorentz part).

User interface

- In the CompHEP model (Lagrangian table) the scalar factor designate by special prefix @.

Clr	Rest	Del	Size	P1	P2	P3	P4	>	Factor	< >	d
				A	W+	W-			-EE		m3.
				A	W+						
				A	W+	A.C					
				A	W+				-i*EE*MW		
				A.C	W+	A			@HAA		
				A.C	W-	B			EE/3		
				A.c	W+						
				A.c	W+						
				A.c	W-						
				A.c	W-						
				A	A	H			@HAA		m1.
				B	b	A			EE/3		G(m
				B	b	G			GG		G(m
				B	b	H			-EE*Mb/(2*MW*SW)		1
				B	b	Z			-EE/(12*CW*SW)		2*S
				B	b	Z.f			-i*EE*Mb/(2*MW*SW)		G5
				B	c	W-			-EE*Sqrt2*Vcb/(4*SW)		G(m
				B	c	W.f			-i*EE*Sqrt2*Vcb/(4*MW*SW)		Mb*
				B	t	W-			-EE*Sqrt2*Vtb/(4*SW)		G(m
				B	t	W.f			-i*EE*Sqrt2*Vtb/(4*MW*SW)		Mb*
				B	u	W-			-EE*Sqrt2*Vub/(4*SW)		G(m

User interface (cont.)

- User should realized the numeric procedure on C language for calculation of formfactor by using the following template:

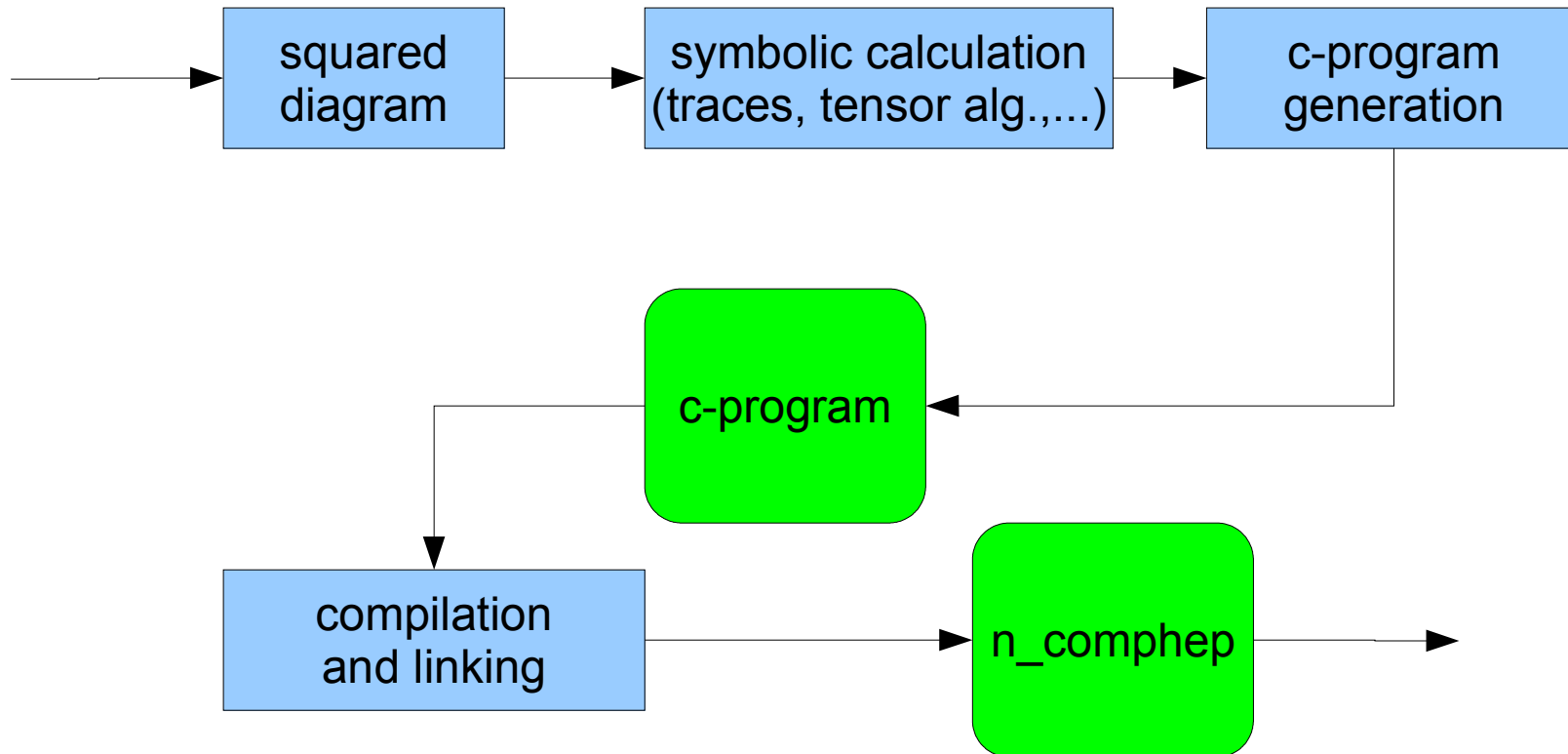
```
double ff(double p11,double p22,double p33,  
          double p12,double p13,double p23)  
/* here pij is a scalar product (pi.pj) */  
{  
    double res=0; /* return value */  
    /* here the numerical calculation of formfactor */  
    return res;  
}
```

and save it in file with **ff.c** name in directory **\$WDIR/ffactor**

User interface (cont.)

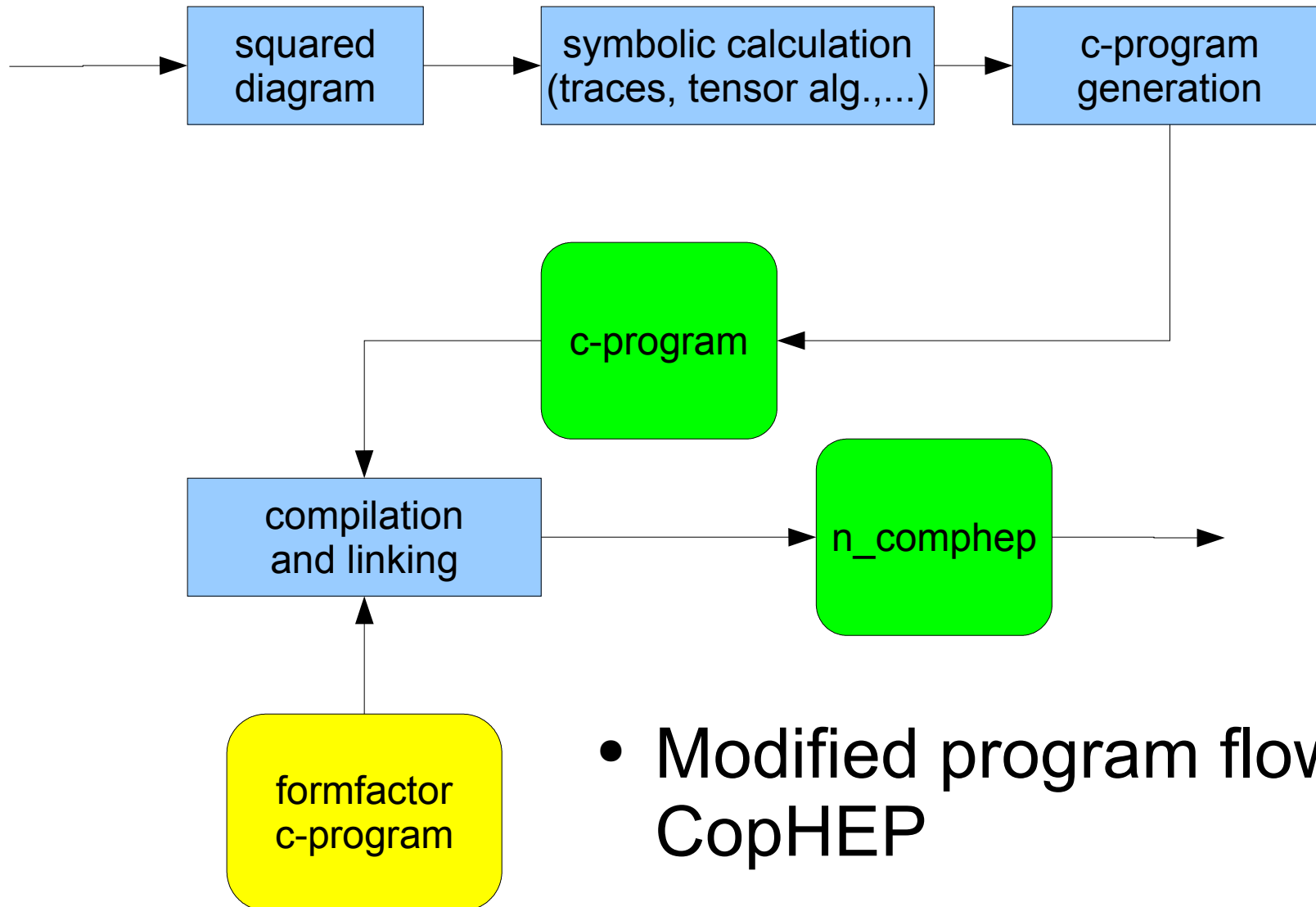
- All C-program in directory `ffactor` will be compiled and linked with numerical part of CompHEP automatically.
- During numeric (Monte-Carlo) calculation of cross section and/or event generation the formfactor functions will be call with proper set of arguments.

Implementation



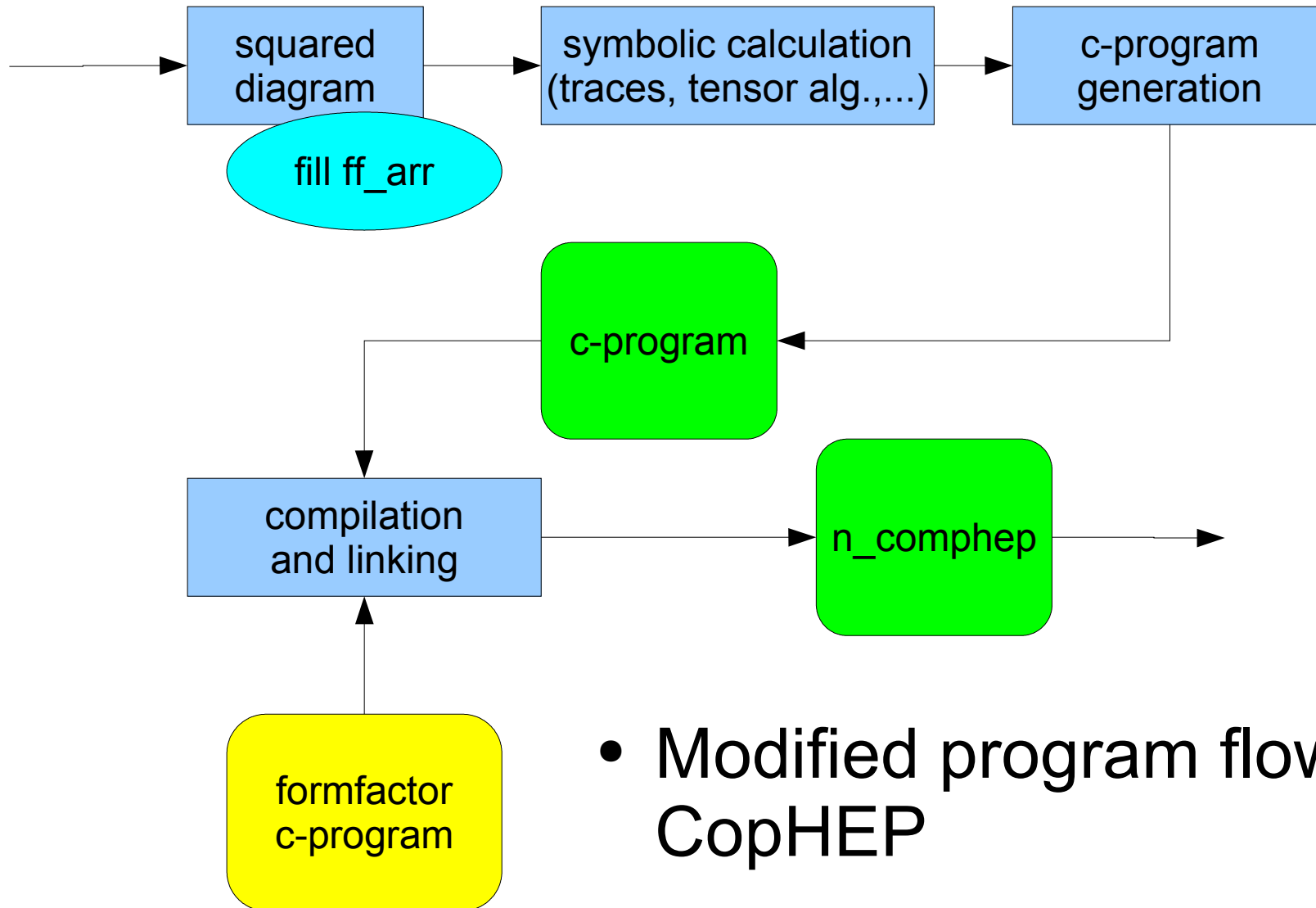
- Program flow in CopHEP

Implementation



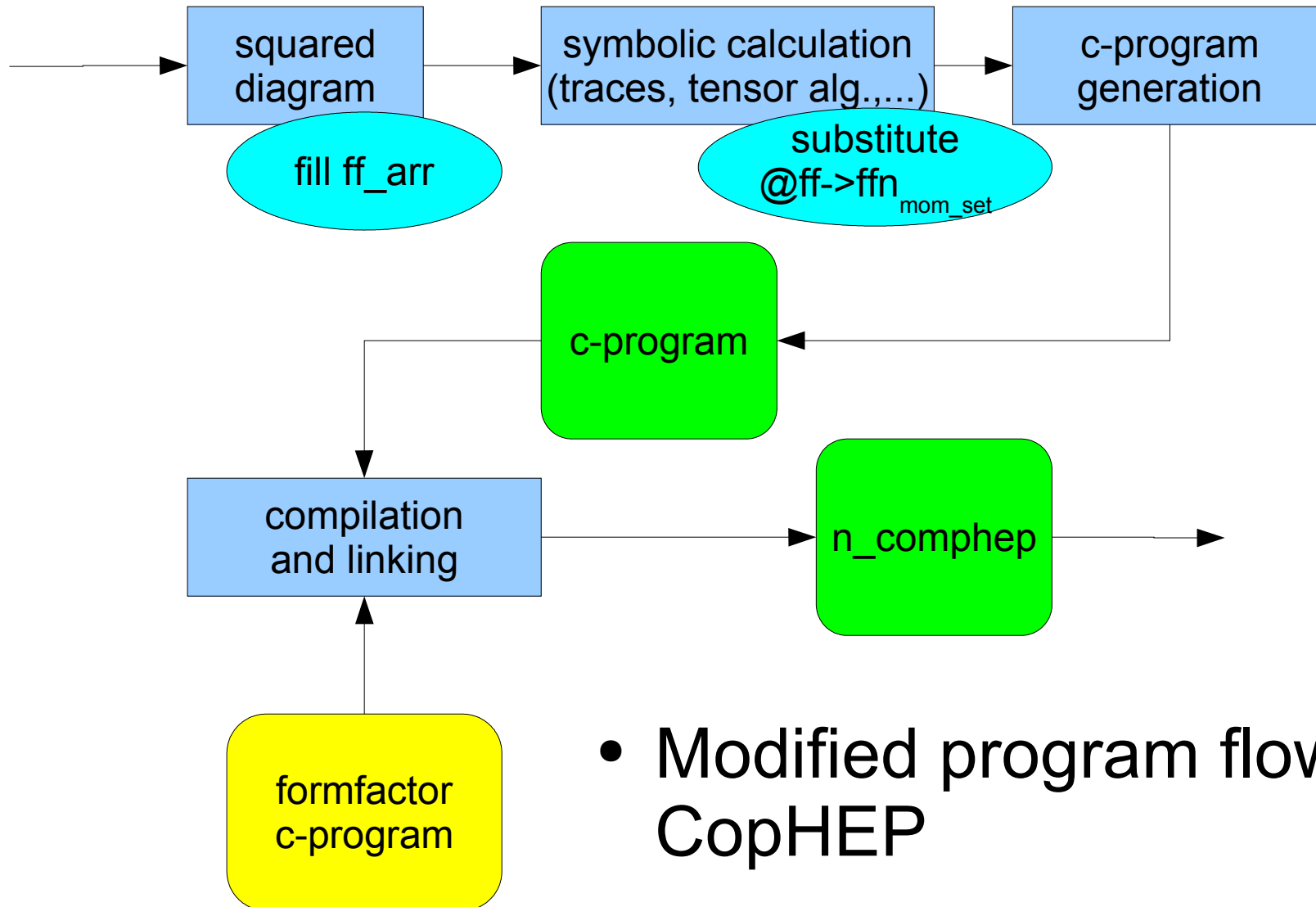
- Modified program flow in CopHEP

Implementation



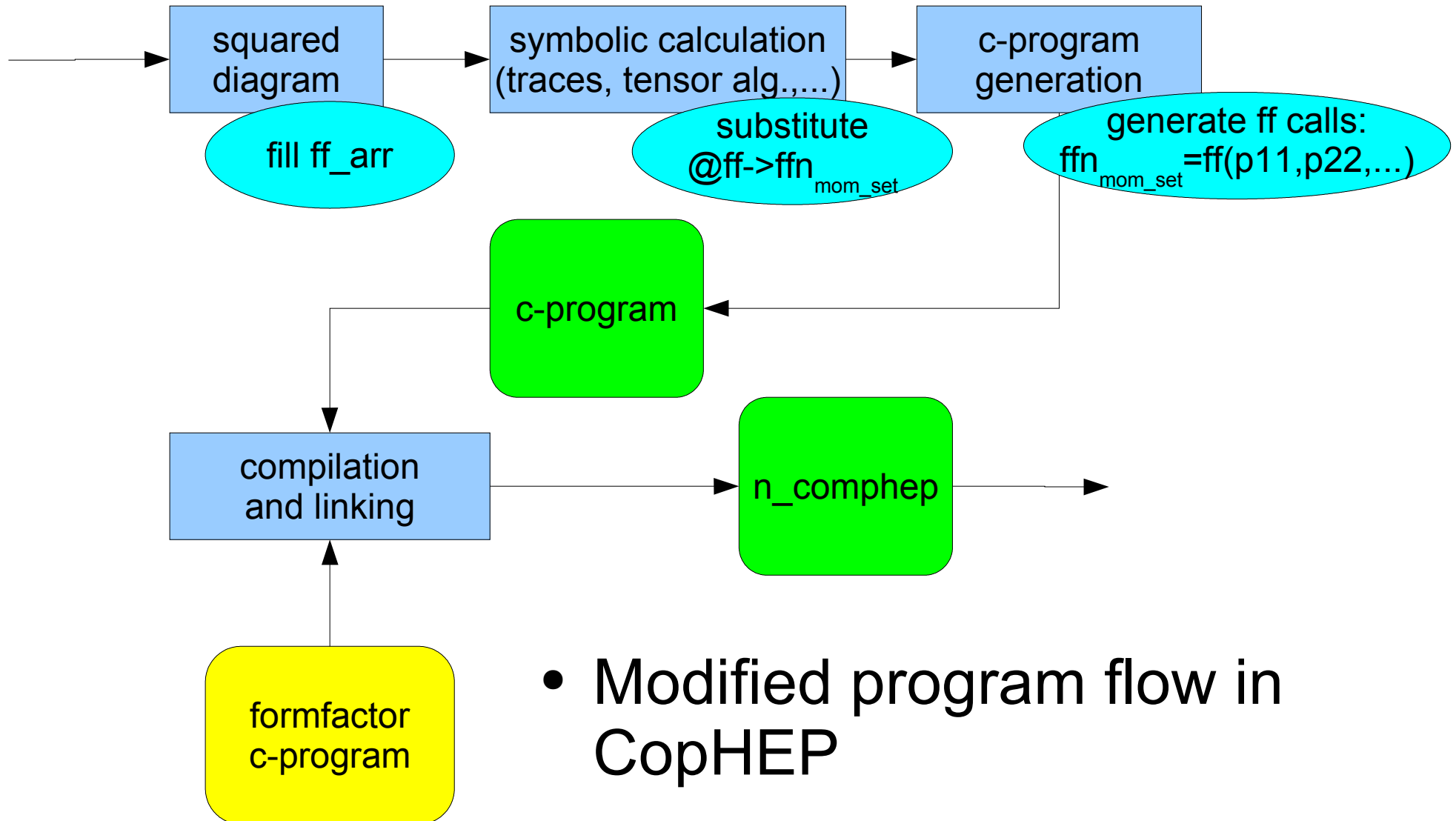
- Modified program flow in CopHEP

Implementation



- Modified program flow in CopHEP

Implementation



- Modified program flow in CopHEP

Implementation

- Mark of squared diagram. Index of set momenta is assigned to each vertex. This information is stored into additional C-structute:

```
structure ff_struct{
    int vnum; /* number of vertex */
    int pset; /* number of momenta set */
} ff_struct;

struct ff_struct ff_arr[MAXVERT];
```

Implementation (cont.)

- Generation C-code for squared matrix element. Special call if formfactor function is generated in fnn.c file:

```
...  
HAA1=HAA(0,0,DP[4],0);  
HAA2=HAA(DP[1],DP[2],DP[3],0);  
RNUM=+HAA1*DP[3]*DP[2]+HAA2*DP[4]*DP[1];  
...
```

Here DP[i] – scalar product of momenta defined in n_comphep.

Conclusions

- Implementation of scalar formfactor in CompHEP improve
 - taking into account of N(N)LO correction;
 - simplify realization SUSY models;
 - open new area of application like J/Psi production;
 - and so on.
- The similar approach will use in future version of CompHEP that should be based on FORM symbolic calculation program by J.Vermaseren.

Conclusion

CompHEP collaboration (incomplete list)

