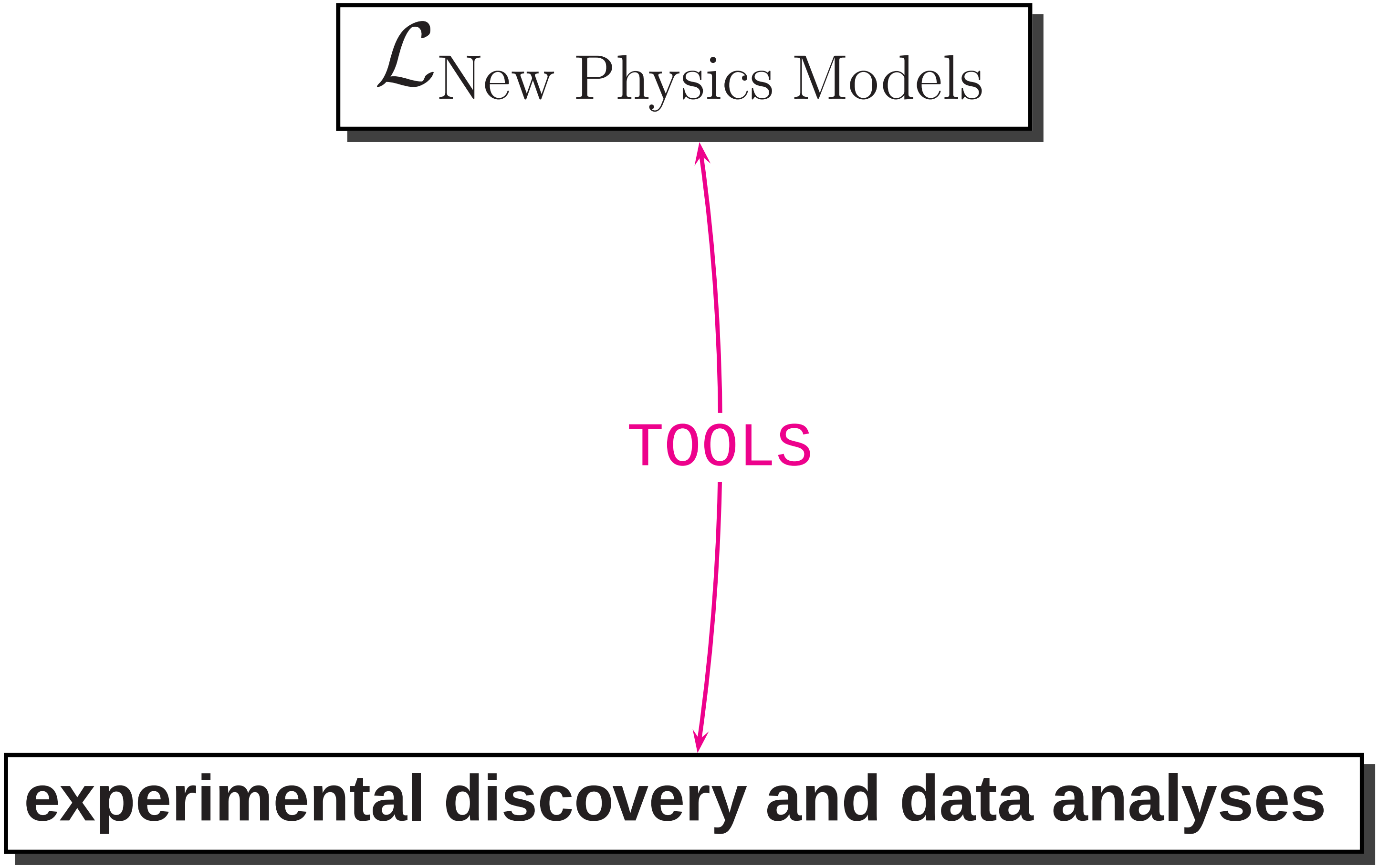
Tools and Monte-Carlos
for the
New Physics

Fawzi BOUDJEMA
LAPTH-Annecy, France

Part B

Outline

- Limitation of PS codes: need for spin correlation and multiparticle matrix element
- Basic ingredients for Matrix Element Generators: techniques
- A simple home made tool
- MEG at helicity amplitude level, GRACE/HELAS: automation
- Automation of Feynman rules: LanHEP, FeynRules
- Small “Tutorial” LanHEP and interface to CalcHEP/FormCalc-FeynArts/micrOMEGAs
- Play with CalcHEP
- One-loop extensions, example
- Modular Structure of codes, Putting all together, Les Houches Accords
- (Conclusions and Outlook)



$\mathcal{L}$ New Physics Models

TOOLS

experimental discovery and data analyses

PS Recap: Limitations, Warning

- **Divide and Conquer : each $\mathcal{P}_{\rm i}$ handled in turn →** Modular Structure
- event generators with PS,...generate events with as much details as possible: in our example with W production

  W will decay...uniformly?

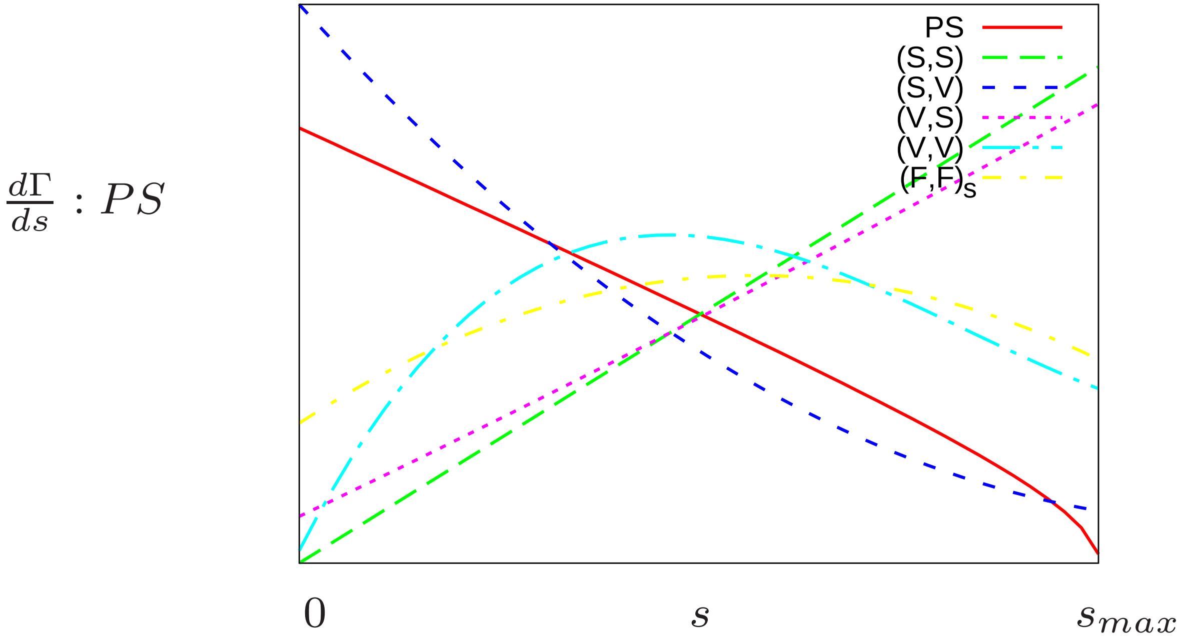
  production comes with non negligible radiation, but is all of the radiation accounted for?
- $\sigma_{\rm final\ state} = \sigma_{\rm hard\ process}\, \mathcal{P}_{\rm tot}$
- $\mathcal{P}_{\rm tot} = \mathcal{P}_{\rm decay}\mathcal{P}_{\rm ISR}\mathcal{P}_{\rm FSR}\mathcal{P}_{\rm remnants}\mathcal{P}_{\rm hadronise}\mathcal{P}_{\rm ord.\ dec.}$
- $\mathcal{P} \times \mathcal{P}_{\rm ord.\ dec.}$, means a combination of $ab \to XY$ followed by $X(Y) \to c, d$ or $X \to X + g$ could do....

  is $a, b \to X(Y) \to cd = a, b \to Xcd$ resonant vs non resonant contribution

  $a, b \to X(Y) \to Yg = a, b \to Xg$ PS vs full MEG

Spin and distributions: NP scenarios often differ but their spin content

Edelhauser, Porod, Ritesh (2010)



Differential width divided by a phase space factor PS for the different decays $X \longrightarrow f\bar{f}Y$, $X, Y \in S, V$ taking $m_f = 0$, $m_X/m_Y = 0.1$ and all couplings equal. In addition the phase space factor is drawn.

ME vs PS: Limitations of PS

- The majority of the processes in the general purpose event generators were implemented one by one, most of them are $2 \to 2$ processes. ▶

ME vs PS: Limitations of PS

- The majority of the processes in the general purpose event generators were implemented one by one, most of them are $2 \to 2$ processes. ▶
- However for the LHC we are often interested in higher multiplicity final states.
- These require specialized matrix element generators
- These are usually dedicated specialised stand-alone codes or recently some of the EG have built in MEG or at least part of the components of these MEG

Pythia

ffff 1
Currently implemented processes, complete with respect to groups, but with some individual processes missing for lack of space (represented by "..."). In the names, a "2" separates initial and final state, an "(s:X)", "(t:X)" or "(l:X)" occasionally appends info on an s- or t-channel- or loop-exchanged particle X.

ProcessGroup	ProcessName
SoftQCD	minBias,elastic, singleDiffractive, doubleDiffractive
HardQCD	gg2gg, gg2qqbar, qg2qg, qq2qq, qqbar2gg, qqbar2qqbarNew, gg2ccbar, qqbar2ccbar, gg2bbbar, qqbar2bbbar
PromptPhoton	qg2qgamma, qqbar2ggamma, gg2ggamma, ffbar2gammagamma, gg2gammagamma
WeakBosonExchange	ff2ff(t:gmZ), ff2ff(t:W)
WeakSingleBoson	ffbar2gmZ, ffbar2W, ffbar2ffbar(s:gm)
WeakDoubleBoson	ffbar2gmZgmZ, ffbar2ZW, ffbar2WW
WeakBosonAndParton	qqbar2gmZg, qg2gmZq, ffbar2gmZgm, fgm2gmZf qqbar2Wg, qg2Wq, ffbar2Wgm, fgm2Wf
Charmonium	gg2QQbar[3S1(1)]g, qg2QQbar[3PJ(8)]q, ...
Bottomonium	gg2QQbar[3S1(1)]g, gg2QQbar[3P2(1)]g, ...
Top	gg2ttbar, qqbar2ttbar, qq2tq(t:W), ffbar2ttbar(s:gmZ), ffbar2tqbar(s:W)
FourthBottom, FourthTop, FourthPair (fourth generation)	
HiggsSM	ffbar2H, gg2H, ffbar2HZ, ff2Hff(t:WW), ...
HiggsBSM	h, H and A as above, charged Higgs, pairs
SUSY	qqbar2chi0chi0 (not yet completed)
NewGaugeBoson	ffbar2gmZZprime, ffbar2Wprime, ffbar2R0
LeftRightSymmmetry	ffbar2ZR, ffbar2WR, ffbar2HLHL, ...
LeptoQuark	ql2LQ, qg2LQl, gg2LQLQbar, qqbar2LQLQbar
ExcitedFermion	dg2dStar, qq2uStarq, qqbar2muStarmu, ...
ExtraDimensionsG*	gg2G*, qqbar2G*, ...

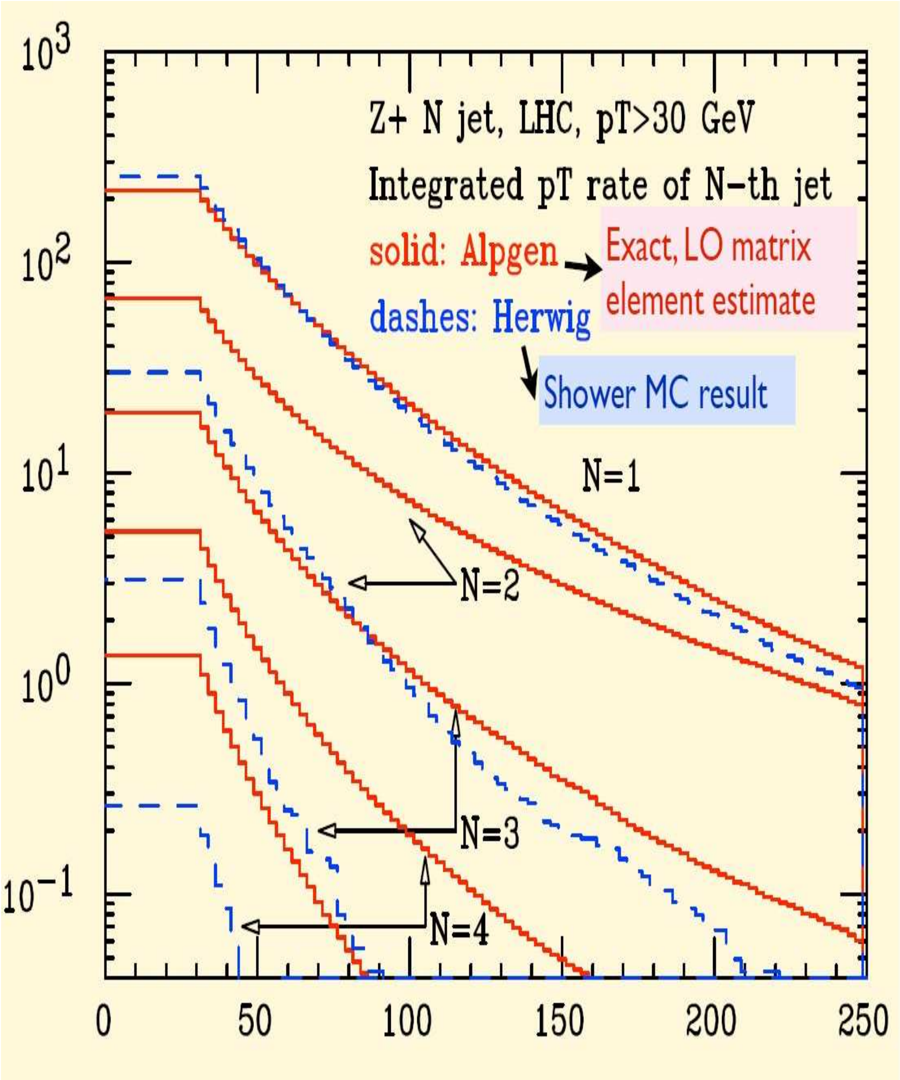
Pythia

ffff 1
Currently implemented processes, complete with respect to groups, but with some individual processes missing for lack of space (represented by "..."). In the names, a "2" separates initial and final state, an "(s:X)", "(t:X)" or "(l:X)" occasionally appends info on an s- or t-channel- or loop-exchanged particle X.

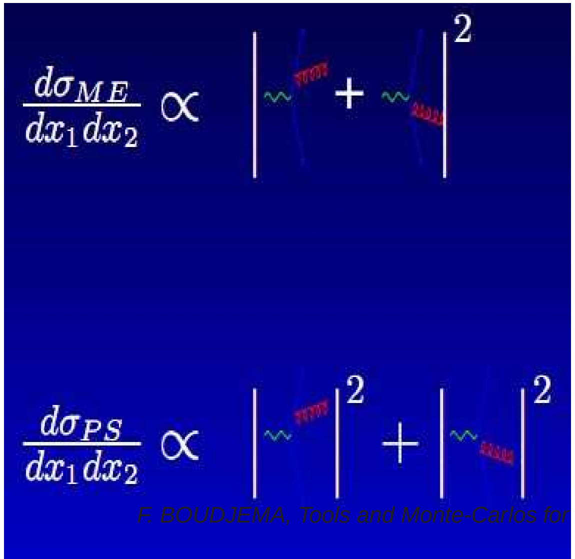
ProcessGroup	ProcessName
SoftQCD	minBias,elastic, singleDiffractive, doubleDiffractive
HardQCD	gg2gg, gg2qqbar, qg2qg, qq2qq, qqbar2gg, qqbar2qqbarNew, gg2ccbar, qqbar2ccbar, gg2bbbar, qqbar2bbbar
PromptPhoton	qg2qgamma, qqbar2ggamma, gg2ggamma, ffbar2gammagamma, gg2gammagamma
WeakBosonExchange	ff2ff(t:gmZ), ff2ff(t:W)
WeakSingleBoson	ffbar2gmZ, ffbar2W, ffbar2ffbar(s:gm)
WeakDoubleBoson	ffbar2gmZgmZ, ffbar2ZW, ffbar2WW
WeakBosonAndParton	qqb ffbar2gmZgm, fgm2gmZf qqb r2Wgm, fgm2Wf
Charmonium	gg2QQbar[3S1(1)]g, qg2QQbar[3PJ(8)]q, ...
Bottomonium	gg2QQbar[3S1(1)]g, gg2QQbar[3P2(1)]g, ...
Top	gg2ttbar, qqbar2ttbar, qq2tq(t:W), ffbar2ttbar(s:gmZ), ffbar2tqbar(s:W)
FourthBottom, FourthTop, FourthPair (fourth generation)	
HiggsSM	ffbar2H, gg2H, ffbar2HZ, ff2Hff(t:WW), ...
HiggsBSM	h, H and A as above, charged Higgs, pairs
SUSY	qqbar2chi0chi0 (not yet completed)
NewGaugeBoson	ffbar2gmZZprime, ffbar2Wprime, ffbar2R0
LeftRightSymmmetry	ffbar2ZR, ffbar2WR, ffbar2HLHL, ...
LeptoQuark	ql2LQ, qg2LQl, gg2LQLQbar, qqbar2LQLQbar
ExcitedFermion	dg2dStar, qq2uStarq, qqbar2muStarmu, ...
ExtraDimensionsG*	gg2G*, qqbar2G*, ...

mostly $2 \to 2$

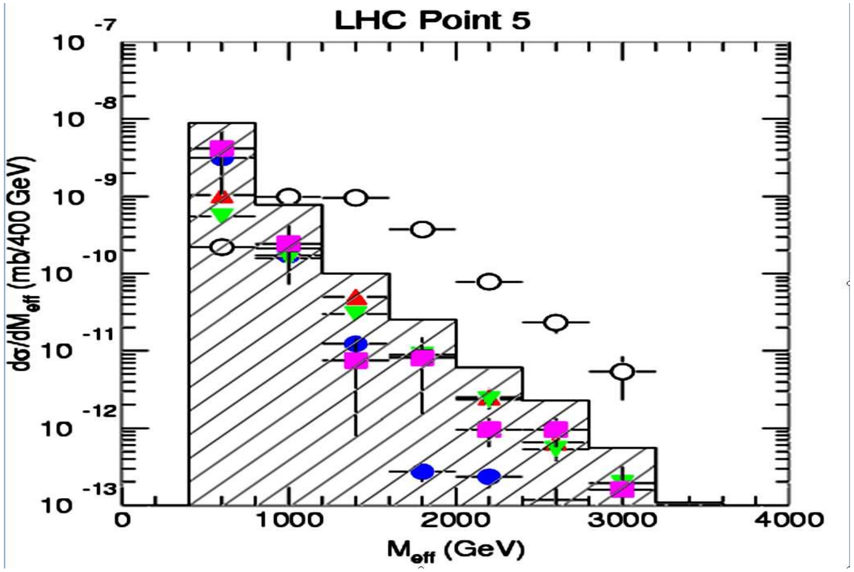
ME vs PS: Limitations of PS



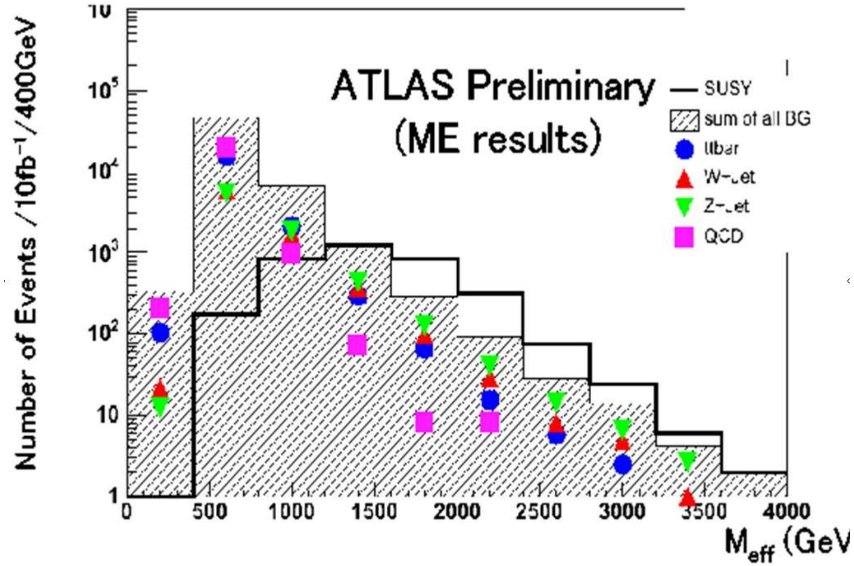
- PS do not describe hard jets
- ME do but in practice can not produce as many jets as PS
- ME evaluates the complete set of all diagrams/configurations: costly
- some real progress has been made in interfacing ME with PS
- CKKW, MLM (say more if time permits) Peter has done this



ATLAS TDR (same with CMS)



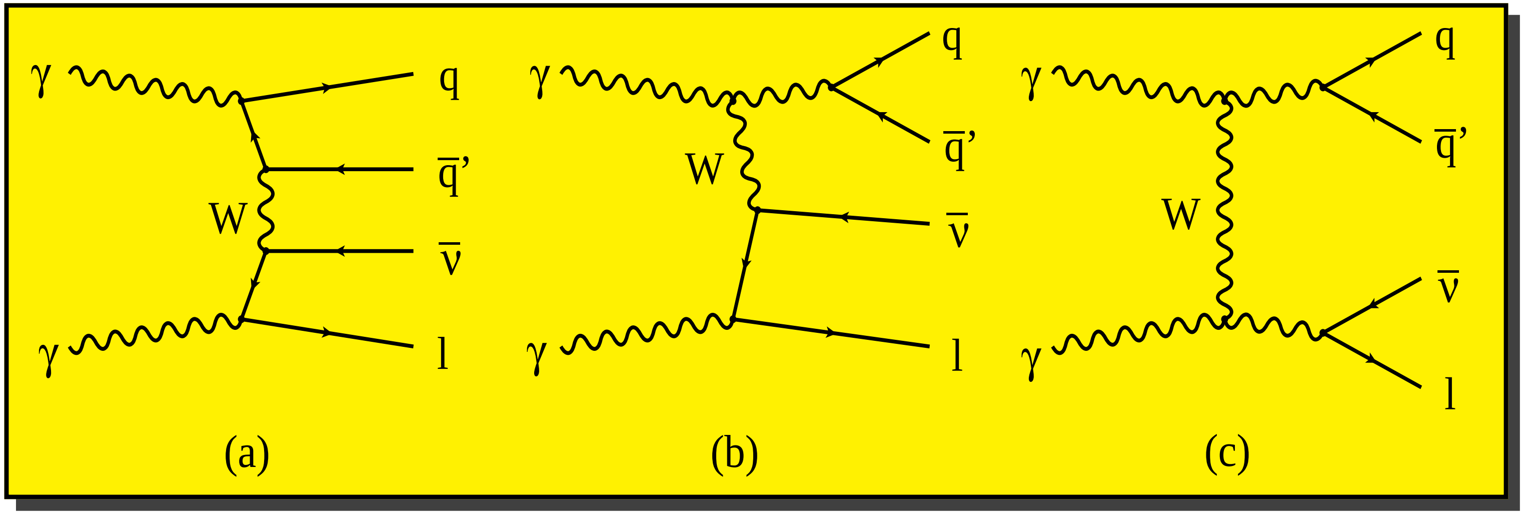
ATLAS TDR 98
(mSUGRA point, PreWMAP)



ATLAS 2006

QCD and SM processes can also produce hard jets! and these are/were lacking in PS/MC

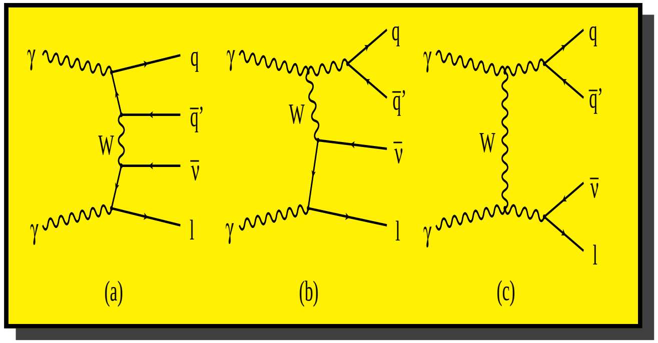
ME vs PS: Resonant vs non resonant



(a) Peripheral (b) single W production (c) Resonant diagram

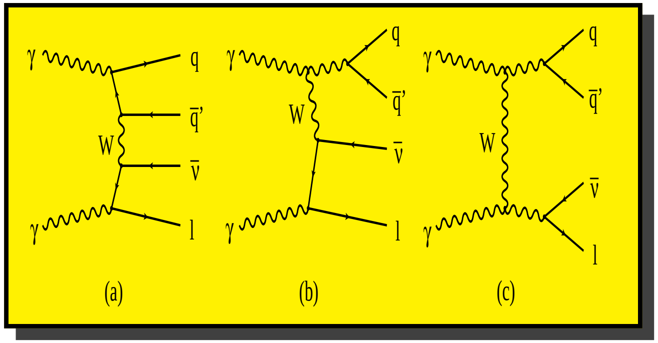
$$\gamma\gamma \to W^- W^+ \; W^\pm \to l\nu_l, W^\mp \to jj' \text{ with } l = e, \mu.$$

ME vs PS: Resonant vs non resonant



- Full calculation needs to evaluate $|\mathcal{M}|^2$
- draw all Feynman diagrams,
- associate Feynman rules to each vertex,
- sum over all diagrams $\mathcal{M} = \sum_i \mathcal{M}_i$

ME vs PS: Resonant vs non resonant

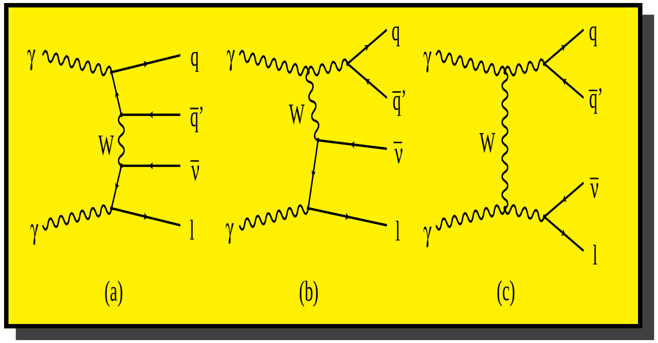


- Full calculation needs to evaluate $|\mathcal{M}|^2$
- draw all Feynman diagrams,
- associate Feynman rules to each vertex,
- sum over all diagrams $\mathcal{M} = \sum_i \mathcal{M}_i$

Squaring technique

- Most textbook will tell you to **square** summing over all polarisations, with tricks like spinors leading to traces, pola. vectors to completeness relations $\sum_\lambda (\epsilon_\mu \epsilon_\nu) \to -g_{\mu\nu}$,....
- with more than $2 \to 2$ this is intractable wit huge number of terms (due to *interference*) $\mathcal{M}_i \mathcal{M}_j^*$ with long expressions from squaring
- Any info on polarisation (initial or final, that can be crucial) is lost
- This technique can be automatised and is used in `CompHEP, CalcHEP`

ME vs PS: Resonant vs non resonant

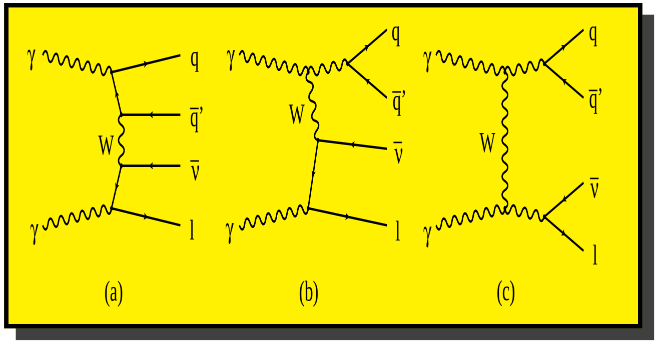


- Full calculation needs to evaluate $|\mathcal{M}|^2$
- draw all Feynman diagrams,
- associate Feynman rules to each vertex,
- sum over all diagrams $\mathcal{M} = \sum_i \mathcal{M}_i$

Helicity amplitude technique

- Calculate each $\mathcal{M}_i$ for a helicity configuration h_α, $\mathcal{M}_i(h_\alpha)$
- each $\mathcal{M}_i(h_\alpha)$ is a c-number
- Sum over i to get the full amplitude (sum before squaring) $\sum_i \mathcal{M}_i(h_\alpha) = \mathcal{M}(h_\alpha)$
- Store $\mathcal{M}(h_\alpha)$ to get polarised/unpolarised/spin-correlation
- Used in GRACE, MadGraph,...

ME vs PS: Resonant vs non resonant

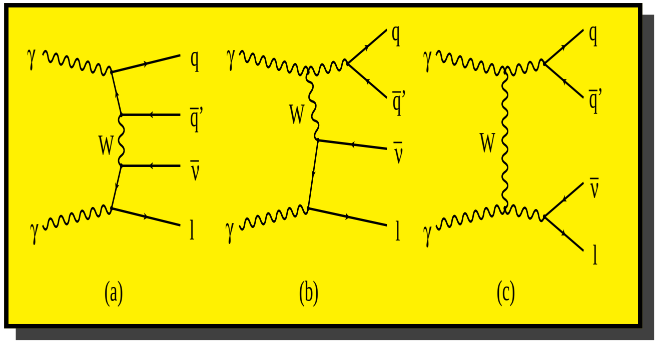


- Full calculation needs to evaluate $|\mathcal{M}|^2$
- draw all Feynman diagrams,
- associate Feynman rules to each vertex,
- sum over all diagrams $\mathcal{M} = \sum_i \mathcal{M}_i$

Recursion relations (mostly for massless states within QCD)

- not based on Feynman diagrams
- idea is to build amplitude for $N+1$ leg from N leg recursively
- Methods based on off-shell currents (Berends-Giele)
- New techniques MHV (Maximum Helicity Violating) based on *formal, twistor inspired* work by Cachazo-Svreck-Witten(CVS)
- Many developments (BCFW, Britto-Cachazo-Feng-Witten) including some applications to massive states
- SYM (N=4 based)/Wilson loops (tree-level and beyond)

ME vs PS: Resonant vs non resonant



- Full calculation needs to evaluate $|\mathcal{M}|^2$
- draw all Feynman diagrams,
- associate Feynman rules to each vertex,
- sum over all diagrams $\mathcal{M} = \sum_i \mathcal{M}_i$

Approximation: Resonant diagrams only?

- Production × decay
- $\sigma = \sigma_{WW} \times Br_{W^+} Br_{W^-}$ is OK only for total cross section without cuts
- distribution sensitive to the spin of decaying particles
- improve with full spin correlation

ME vs PS: Resonant vs non resonant, Density Matrix, Full Spin Correlation

$$\frac{d\sigma(\gamma(\lambda_1)\gamma(\lambda_2) \to W^+W^- \to f_1\bar{f}_2 f_3\bar{f}_4)}{d\cos\theta\ \ d\cos\theta^*_-\ \ d\phi^*_-\ \ d\cos\theta^*_+\ \ d\phi^*_+} = Br_W^{f_1\bar{f}_2}\, Br_W^{f_3\bar{f}_4}\, \frac{\beta}{32\pi s}\left(\frac{3}{8\pi}\right)^2 \qquad \times$$

$$\sum_{\lambda_-\lambda_+\lambda'_-\lambda'_+} \mathcal{M}_{\lambda_1,\lambda_2;\lambda_-\lambda_+}(s,\cos\theta)\ \mathcal{M}^*_{\lambda_1,\lambda_2;\lambda'_-\lambda'_+}(s,\cos\theta)\times$$

$$D_{\lambda_-\lambda'_-}(\theta^*_-,\phi^*_-)D_{\lambda_+\lambda'_+}(\pi-\theta^*_+,\phi^*_+ +\pi)$$

$$\equiv \frac{d\sigma(\gamma(\lambda_1)\gamma(\lambda_2) \to W^+W^-)}{d\cos\theta}\left(\frac{3}{8\pi}\right)^2 Br_W^{f_1\bar{f}_2}\, Br_W^{f_3\bar{f}_4}$$

$$\sum_{\lambda_-\lambda_+\lambda'_-\lambda'_+} \rho^{\lambda_1,\lambda_2}_{\lambda_-\lambda_+\lambda'_-\lambda'_+}\ D_{\lambda_-\lambda'_-}(\theta^*_-,\phi^*_-)\ D_{\lambda_+\lambda'_+}(\pi-\theta^*_+,\phi^*_+ +\pi)$$

with $$\rho^{\lambda_1,\lambda_2}_{\lambda_-\lambda_+\lambda'_-\lambda'_+}(s,\cos\theta) = \frac{\mathcal{M}_{\lambda_1,\lambda_2;\lambda_-\lambda_+}(s,\cos\theta)\ \mathcal{M}^*_{\lambda_1,\lambda_2;\lambda'_-\lambda'_+}(s,\cos\theta)}{\sum_{\lambda_-\lambda_+}|\mathcal{M}_{\lambda_1,\lambda_2;\lambda_-\lambda_+}(s,\cos\theta)|^2},$$

ME vs PS: Resonant vs non resonant, Density Matrix, Full Spin Correlation

$$\frac{\mathrm{d}\sigma(\gamma(\lambda_1)\gamma(\lambda_2) \to W^+W^- \to f_1\bar{f}_2 f_3\bar{f}_4)}{\mathrm{d}\cos\theta\ \ \mathrm{d}\cos\theta^*_-\ \ \mathrm{d}\phi^*_-\ \ \mathrm{d}\cos\theta^*_+\ \ \mathrm{d}\phi^*_+} = Br_W^{f_1\bar{f}_2} Br_W^{f_3\bar{f}_4} \frac{\beta}{32\pi s}\left(\frac{3}{8\pi}\right)^2 \quad \times$$

$$\sum_{\lambda_-\lambda_+\lambda'_-\lambda'_+} \mathcal{M}_{\lambda_1,\lambda_2;\lambda_-\lambda_+}(s,\cos\theta)\ \mathcal{M}^*_{\lambda_1,\lambda_2;\lambda'_-\lambda'_+}(s,\cos\theta)\times$$

$$D_{\lambda_-\lambda'_-}(\theta^*_-,\phi^*_-)D_{\lambda_+\lambda'_+}(\pi-\theta^*_+,\phi^*_+ + \pi)$$

$$\equiv \frac{\mathrm{d}\sigma(\gamma(\lambda_1)\gamma(\lambda_2) \to W^+W^-)}{\mathrm{d}\cos\theta}\left(\frac{3}{8\pi}\right)^2 Br_W^{f_1\bar{f}_2} Br_W^{f_3\bar{f}_4}$$

$$\sum_{\lambda_-\lambda_+\lambda'_-\lambda'_+} \rho^{\lambda_1,\lambda_2}_{\lambda_-\lambda_+\lambda'_-\lambda'_+}\ D_{\lambda_-\lambda'_-}(\theta^*_-,\phi^*_-)\ D_{\lambda_+\lambda'_+}(\pi-\theta^*_+,\phi^*_+ + \pi)$$

with $$\rho^{\lambda_1,\lambda_2}_{\lambda_-\lambda_+\lambda'_-\lambda'_+}(s,\cos\theta) = \frac{\mathcal{M}_{\lambda_1,\lambda_2;\lambda_-\lambda_+}(s,\cos\theta)\ \mathcal{M}^*_{\lambda_1,\lambda_2;\lambda'_-\lambda'_+}(s,\cos\theta)}{\sum_{\lambda_-\lambda_+}|\mathcal{M}_{\lambda_1,\lambda_2;\lambda_-\lambda_+}(s,\cos\theta)|^2},$$

ME vs PS: Resonant vs non resonant, Density Matrix, Full Spin Correlation

$$\frac{\mathrm{d}\sigma(\gamma(\lambda_1)\gamma(\lambda_2) \to W^+W^- \to f_1\bar{f}_2 f_3\bar{f}_4)}{\mathrm{d}\cos\theta\ \ \mathrm{d}\cos\theta^*_-\ \ \mathrm{d}\phi^*_-\ \ \mathrm{d}\cos\theta^*_+\ \ \mathrm{d}\phi^*_+} = Br_W^{f_1\bar{f}_2}\, Br_W^{f_3\bar{f}_4}\, \frac{\beta}{32\pi s}\left(\frac{3}{8\pi}\right)^2 \qquad \times$$

$$\sum_{\lambda_-\lambda_+\lambda'_-\lambda'_+} \mathcal{M}_{\lambda_1,\lambda_2;\lambda_-\lambda_+}(s,\cos\theta)\ \mathcal{M}^*_{\lambda_1,\lambda_2;\lambda'_-\lambda'_+}(s,\cos\theta)\times$$

$$D_{\lambda_-\lambda'_-}(\theta^*_-,\phi^*_-)D_{\lambda_+\lambda'_+}(\pi-\theta^*_+,\phi^*_+ + \pi)$$

$$\equiv \frac{\mathrm{d}\sigma(\gamma(\lambda_1)\gamma(\lambda_2)\to W^+W^-)}{\mathrm{d}\cos\theta}\left(\frac{3}{8\pi}\right)^2 Br_W^{f_1\bar{f}_2}\, Br_W^{f_3\bar{f}_4}$$

$$\sum_{\lambda_-\lambda_+\lambda'_-\lambda'_+} \rho^{\lambda_1,\lambda_2}_{\lambda_-\lambda_+\lambda'_-\lambda'_+}\ D_{\lambda_-\lambda'_-}(\theta^*_-,\phi^*_-)\ D_{\lambda_+\lambda'_+}(\pi-\theta^*_+,\phi^*_+ + \pi)$$

with $$\rho^{\lambda_1,\lambda_2}_{\lambda_-\lambda_+\lambda'_-\lambda'_+}(s,\cos\theta) = \frac{\mathcal{M}_{\lambda_1,\lambda_2;\lambda_-\lambda_+}(s,\cos\theta)\ \mathcal{M}^*_{\lambda_1,\lambda_2;\lambda'_-\lambda'_+}(s,\cos\theta)}{\sum_{\lambda_-\lambda_+}|\mathcal{M}_{\lambda_1,\lambda_2;\lambda_-\lambda_+}(s,\cos\theta)|^2},$$

W decay functions

$$D^{W^-}_{\lambda,\lambda'}(\theta^*,\phi^*) \equiv D_{\lambda,\lambda'}, \qquad D_{\lambda,\lambda'} = D^*_{\lambda',\lambda}$$

$$D_{+,-} = \frac{1}{2}(1-\cos^2\theta^*)e^{2i\phi^*}, \qquad D_{\pm,0} = -\frac{1}{\sqrt{2}}(1\mp\cos\theta^*)\sin\theta^* e^{\pm i\phi^*},$$

$$D_{\pm,\pm} = \frac{1}{2}(1\mp\cos\theta^*)^2, \qquad D_{0,0} = \sin^2\theta^*.$$

ME vs PS: Resonant vs non resonant

λ_1 λ_2	Inv. Mass. Cuts	Narrow Width Improved	All diag. $\Gamma_W(M_W^2)$	All diag. $\Gamma_W(s)$	All diag. Fudge	"Resonant" Subset
$\sqrt{s} = 400$ GeV						
$+ +$	None	2288	2310	2312	2309	2354
$+ -$	None	1893	1926	1927	1923	1975
$- +$	None	1890	1927	1927	1926	1975
$- -$	None	2186	2184	2183	2183	2252
$+ +$	$\Delta_{jj}, \Delta_{l\nu} < 5$ GeV	1759	1762	1764	1761	1761
$+ -$	$\Delta_{jj}, \Delta_{l\nu} < 5$ GeV	1455	1458	1458	1456	1456
$- +$	$\Delta_{jj}, \Delta_{l\nu} < 5$ GeV	1454	1457	1458	1457	1456
$- -$	$\Delta_{jj}, \Delta_{l\nu} < 5$ GeV	1681	1683	1681	1682	1682
$\sqrt{s} = 1600$ GeV						
$+ +$	None	377	389	389	389	456
$+ -$	None	320	335	335	335	388
$- +$	None	320	336	336	336	391
$- -$	None	427	447	447	447	490
$+ +$	$\Delta_{jj}, \Delta_{l\nu} < 5$ GeV	290	291	291	291	291
$+ -$	$\Delta_{jj}, \Delta_{l\nu} < 5$ GeV	246	246	246	246	246
$- +$	$\Delta_{jj}, \Delta_{l\nu} < 5$ GeV	246	246	247	246	247
$- -$	$\Delta_{jj}, \Delta_{l\nu} < 5$ GeV	328	329	329	330	329

Spin Correlation and Radiation

Usual techniques require production and decay to be generated at the same time

In a generator the difficulty is that

- We need to generate QCD radiation before particle decays
- There may be a long chain of sequential decays
- The particle may have different decay channels
- There may be a few final state particles

Need an algorithm for the production and decay to be done separately

Complexity should not grow more than the number of external particles

in HERWIG spin correlation is implemented not always the case for Pythia (even without radiation)

Matrix Elements Generation and Automation: Feynman diagrams

Automation of LO calculations of (partonic) processes $2 \to N$ are now automatised including integration, for (say) $N < 8$ based on different methods

- `ALPGEN` (non Feynman based) quite powerful especially for multiparticle Sm background (not much BSM, Z'...)
- `CompHEP/CalcHEP` could be slow with lots of particles but interface to Dark Matter codes, LanHEP
- `Grace` various guises some public others not
- `HELAS/PHEGAS`
- `MADGRAPH/MADEVENT` major activity (you've had tutorials here)
- `O'Mega/WHIZARD`
- `SHERPA/Amegic` powerful, CKKW, integrated within SHERPA evt generator

Matrix Elements Generation and Automation: Feynman diagrams

Automation of LO calculations of (partonic) processes $2 \to N$ are now automatised including integration, for (say) $N < 8$ based on different methods

- `ALPGEN` (non Feynman based) quite powerful especially for multiparticle Sm background (not much BSM, Z'...)
- `CompHEP/CalcHEP` could be slow with lots of particles but interface to Dark Matter codes, LanHEP
- `Grace` various guises some public others not
- `HELAS/PHEGAS`
- `MADGRAPH/MADEVENT` major activity (you've had tutorials here)
- `O'Mega/WHIZARD`
- `SHERPA/Amegic` powerful, CKKW, integrated within SHERPA evt generator

the more particles one deals with a (Feynman) diagrammatic calculations is costly as the number of diagrams grows $N!$

#gluons	2	3	4	5	6	7	8
#diagrams	4	25	220	2485	34300	0.5M	80M

Interfaces with Herwig/Pythia via LHA

Basics and ingredients of automated MEG

How would I go about calculating a Matrix element or/and a cross section without a dedicated tool?

tool: does not include pen/chalk, computer, (Symbolic manipulation software)

Basics and ingredients of automated MEG

How would I go about calculating a Matrix element or/and a cross section without a dedicated tool?

tool: does not include pen/chalk, computer, (Symbolic manipulation software)

I need $\mathcal{L}_{\text{New Physics Models}}$

Basics and ingredients of automated MEG

How would I go about calculating a Matrix element or/and a cross section without a dedicated tool?

tool: does not include pen/chalk, computer, (Symbolic manipulation software)

I need $\mathcal{L}_{\text{New Physics Models}}$

Figure out what my particles are

spin assignment, colour, charges,...quantum numbers

Label them

Basics and ingredients of automated MEG

How would I go about calculating a Matrix element or/and a cross section without a dedicated tool?

tool: does not include pen/chalk, computer, (Symbolic manipulation software)

I need $\mathcal{L}_{\text{New Physics Models}}$

Get the set of Feynman rules

get them from a trustworthy source (text book?)

better to have the complete set!

getting the full from different sources is asking for trouble

derive the rules myself

Basics and ingredients of automated MEG

How would I go about calculating a Matrix element or/and a cross section without a dedicated tool?

tool: does not include pen/chalk, computer, (Symbolic manipulation software)

I need $\mathcal{L}_{\rm New\ Physics\ Models}$

Keep a table for numerics, parameters

masses, couplings, combinations of such

Basics and ingredients of automated MEG

How would I go about calculating a Matrix element or/and a cross section without a dedicated tool?

tool: does not include pen/chalk, computer, (Symbolic manipulation software)

I need $\mathcal{L}_{\text{New Physics Models}}$

Would need a tool or efficient way for algebraic manipulations

Mathematica, Maple, Form,..

compilers for numerics also

Basics and ingredients of automated MEG

How would I go about calculating a Matrix element or/and a cross section without a dedicated tool?

tool: does not include pen/chalk, computer, (Symbolic manipulation software)

I need $\mathcal{L}$New Physics Models

Define the Process

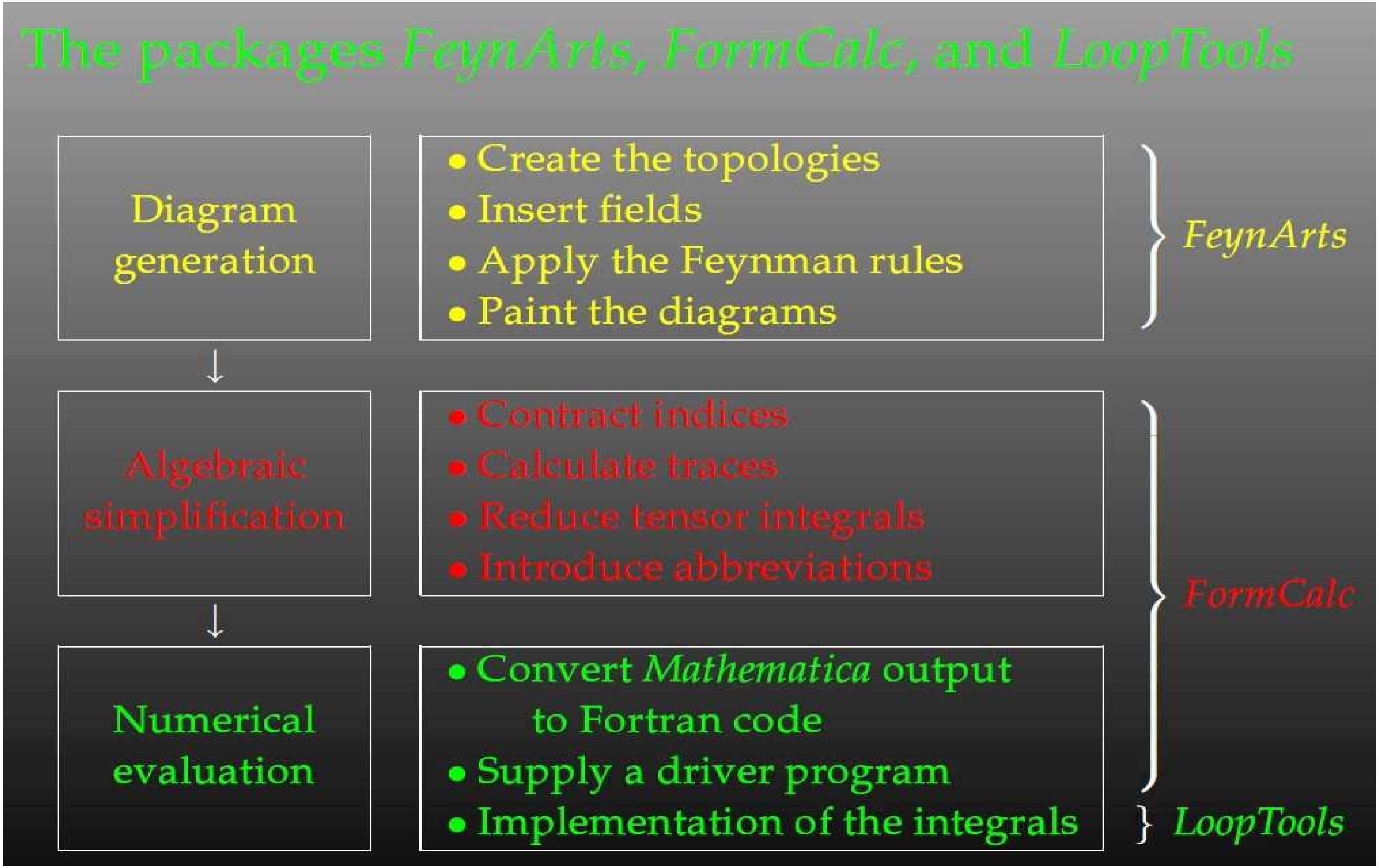
Squaring technique or helicity amplitude?

Integration over phase space. Analytical?, numerical, MC?

Feynman Recipe, knitting with vertices and propagators

Draw all possible types of diagrams	topology
Figure out which particles can run on each type of diagram	combinatorics
Translate diagrams into expressions applying the Feynman rules	data-base look up
contract indices, take traces, (multiply add blocks)	algebra
Collect and write up the results as a computer code	programming
integrate over phase space	coding/computing
run the program to get numerical values	waiting!

Your desiderata and wishes come true!



The packages *FeynArts*, *FormCalc*, and *LoopTools*

Diagram generation

- Create the topologies
- Insert fields
- Apply the Feynman rules
- Paint the diagrams

} *FeynArts*

↓

Algebraic simplification

- Contract indices
- Calculate traces
- Reduce tensor integrals
- Introduce abbreviations

↓

Numerical evaluation

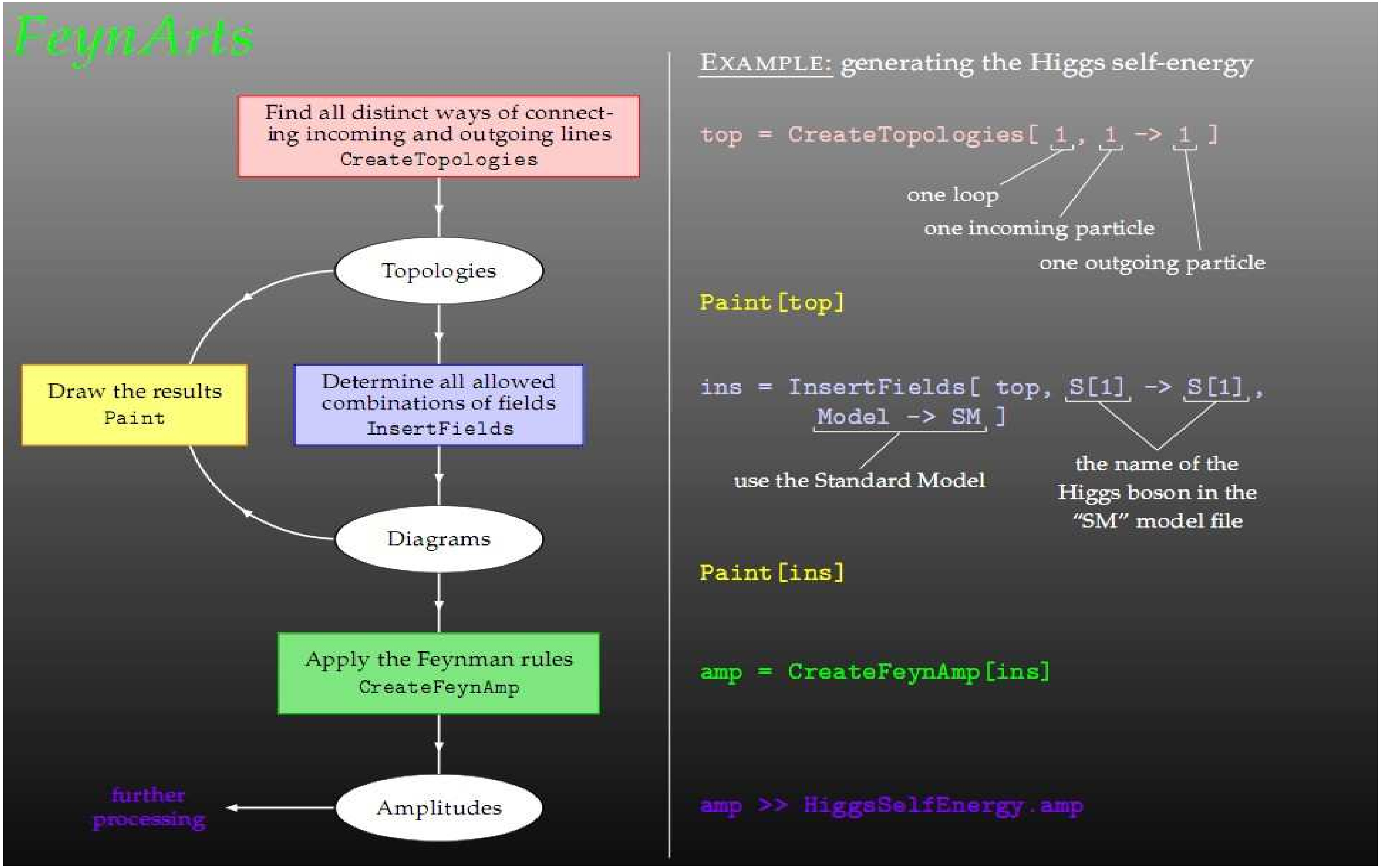
- Convert *Mathematica* output to Fortran code
- Supply a driver program

} *FormCalc*

- Implementation of the integrals

} *LoopTools*

Feyn Package



FeynArts

Find all distinct ways of connecting incoming and outgoing lines
`CreateTopologies`

Topologies

Draw the results
`Paint`

Determine all allowed combinations of fields
`InsertFields`

Diagrams

Apply the Feynman rules
`CreateFeynAmp`

Amplitudes

further processing

EXAMPLE: generating the Higgs self-energy

```
top = CreateTopologies[ 1, 1 -> 1 ]
```

one loop

one incoming particle

one outgoing particle

```
Paint[top]
```

```
ins = InsertFields[ top, S[1] -> S[1],
        Model -> SM ]
```

use the Standard Model

the name of the Higgs boson in the "SM" model file

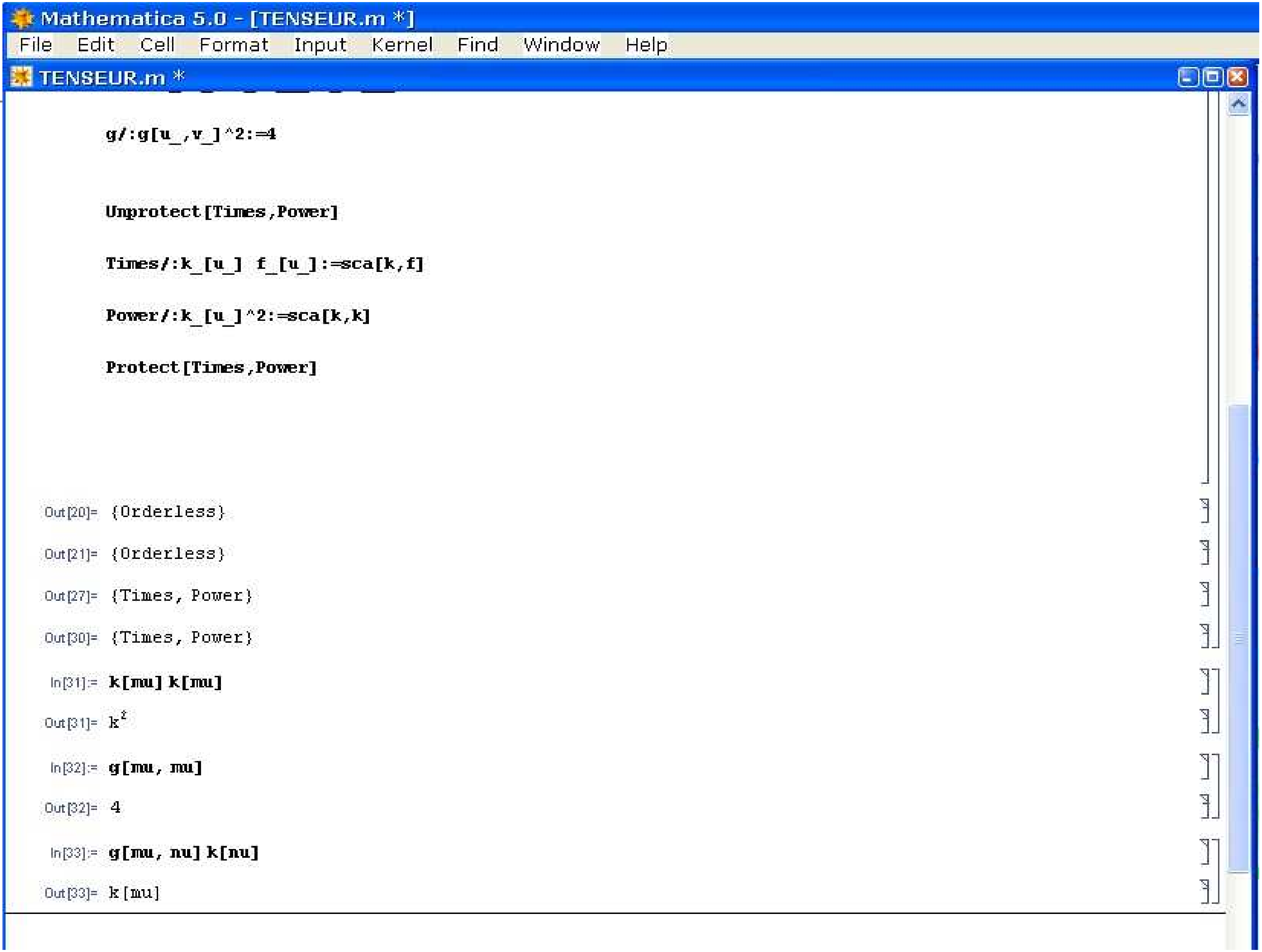
```
Paint[ins]
```

```
amp = CreateFeynAmp[ins]
```

```
amp >> HiggsSelfEnergy.amp
```

Very simple and compact tools for spin-1 manipulations, 20lines

```
ok /: ok[x_] := FreeQ[{sca, Times, List, Dot, Plus, Power, Cos, Sin}, x]

tens /: tens := % /. (f_)?ok[y__] → SequenceForm[f, Superscript[SequenceForm[y]]]

Format[sca[k_, k_]] := k^2

Format[sca[k_, f_]] := SequenceForm[k, ".", f]


Attributes[g] = {Orderless}

Attributes[sca] = {Orderless}

sca /: sca[x_, y_ + z_] := sca[x, y] + sca[x, z]

sca /: sca[-x_, y_] := -sca[x, y]

g /: g[u_, u_] := 4

g /: g[u_, v_] k_[x___, u_, y___] := k[x, v, y]

g /: g[u_, v_]^2 := 4

Unprotect[Times, Power]

Times /: k_[u_] f_[u_] := sca[k, f]

Power /: k_[u_]^2 := sca[k, k]

Protect[Times, Power]
```



```
g/:g[u_,v_]^2:=4


Unprotect[Times,Power]

Times/:k_[u_] f_[u_]:=sca[k,f]

Power/:k_[u_]^2:=sca[k,k]

Protect[Times,Power]



Out[20]= {Orderless}

Out[21]= {Orderless}

Out[27]= {Times, Power}

Out[30]= {Times, Power}

In[31]:= k[mu] k[mu]

          2
Out[31]= k

In[32]:= g[mu, mu]

Out[32]= 4

In[33]:= g[mu, nu] k[nu]

Out[33]= k[mu]
```

FeynCalc.m, 8000 lines (dirac, PV reduction,..)

```
Dot @@ (Reverse[{Spinor[-pe1, ma1], yz, Spinor[-pe2, ma2]}]/.
                    DiracGamma[5]->(-DiracGamma[5])/.
                   {DiracGamma[6] :> DiracGamma[7],
                    DiracGamma[7]:>DiracGamma[6]}
       );

 ComplexConjugate /: ComplexConjugate[ComplexConjugate[x_]]:=x;
 cLIndex[x_, dime___] := LorentzIndex[ComplexConjugate[x], dime];
 cSIndex[x_] := SUNIndex[ComplexConjugate[x]];
 Unprotect[Conjugate];
 Conjugate[x_] := x /. {Polarization[k_,1,in___] :> Polarization[k,-1,in],
                        Polarization[k_,-1,in___] :>Polarization[k,1,in]
                       }/.  Complex[a_, b_] -> Complex[a, -b] /.
                       Dot -> rev /. rev -> Dot /.
                       LorentzIndex -> cLIndex /.
                       SUNIndex  -> cSIndex;
 Protect[Conjugate];
 Polarization/:Momentum[Polarization[k_,i___],di_Symbol
                               ]:=Momentum[Polarization[k,i]];
 Polarization/:Momentum[Polarization[k__], di_Symbol - 4 ]:= 0;
(* #################################################################### *)
(*                             Main14                                  *)
(* #################################################################### *)
(* MetricTensordef *)
 Options[MetricTensor]={Dimension->4};
MetricTensor[x__]:=MetricTensor[x]=metricTensor[x];
 loin1[x_,___]:=x;
 metricTensor[a_ b_,opt___]:=metricTensor[a,b,opt];
 metricTensor[a_^2 ,opt___]:=metricTensor[a,a,opt];
 metricTensor[ x__ ]:=(metricTensor@@({x}/.LorentzIndex->loin1));
 metricTensor[x_,x_,op_:{}]:=(Dimension/.op/.Options[MetricTensor]);
 metricTensor[ x_, y_,op_:{} ] :=
    Pair[ LorentzIndex[x,Dimension/.op/.Options[MetricTensor] ],
          LorentzIndex[y,Dimension/.op/.Options[MetricTensor] ]
        ];
(* PolarizationVectordef *)
Polarization[k_]:=Polarization[k]=Polarization[k,1];
PolarizationVector[x__]:=PolarizationVector[x]=polarizationVector[x];

(* By default a second argument "1" is put into Polarization *)
(* This is changed to "-1" for conjugate polarization vectors *)
 polarizationVector[k_,mu_]:=
       FourVector[Polarization[k,1], mu, Dimension->4 ];
 polarizationVector[k_,mu_,glu_]:=
```

Feynman rules within tenseur.m, SM couplings!

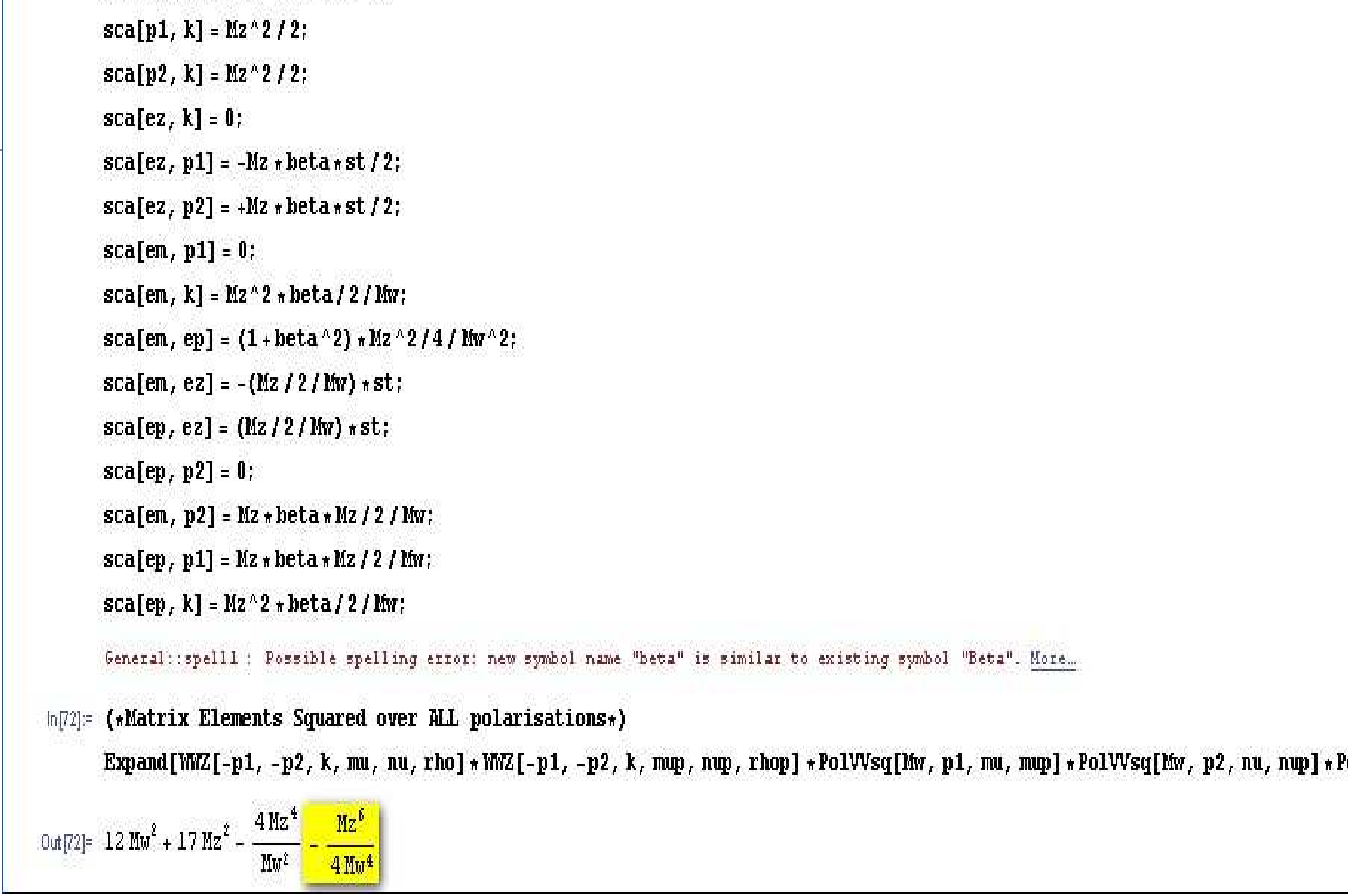
```
(*trilinear vertex W-(m,mu) W+(p,nu) Z/gamma (z,rho) all momenta entering*)
(*-i e*)WWZ[m_, p_, z_, mu_, nu_, rho_] := g[mu, nu] (m[rho] - p[rho]) +
   (z[nu] g[mu, rho] - z[mu] g[nu, rho]) + (p[mu] g[nu, rho] - m[nu] g[mu, rho]);
(*quadratic*)
(*-i e^2*)
WWZZ[mu_, nu_, rho_, sig_] :=
  -2 g[mu, nu] g[rho, sig] + g[mu, rho] g[nu, sig] + g[mu, sig] g[nu, rho];
(*Propagator of massive spin-1*)
(*-i*)
PropagV[M_, p_, mu_, nu_] := (g[mu, nu] - p[mu] p[nu] / M^2) / (sca[p, p] - M^2);
(*Sum on polarisations*)
PolVVsq[M_, p_, mu_, nu_] := g[mu, nu] - p[mu] p[nu] / M^2;
```

Example, Z'decay, kinematics

```
(*Z (k,rho,ez) to W-(p1,mu,em) W+(p2,nu,ep)*)
(*kinematics*)
sca[k, k] = Mz^2;
sca[p1, p1] = Mw^2;
sca[p2, p2] = Mw^2;
sca[p1, p2] = Mz^2/2 - Mw^2;
sca[p1, k] = Mz^2/2;
sca[p2, k] = Mz^2/2;
sca[ez, k] = 0;
sca[ez, p1] = -Mz*beta*st/2;
sca[ez, p2] = +Mz*beta*st/2;
sca[em, p1] = 0;
sca[em, k] = Mz^2*beta/2/Mw;
sca[em, ep] = (1+beta^2)*Mz^2/4/Mw^2;
sca[em, ez] = -(Mz/2/Mw)*st;
sca[ep, ez] = (Mz/2/Mw)*st;
sca[ep, p2] = 0;
sca[em, p2] = Mz*beta*Mz/2/Mw;
sca[ep, p1] = Mz*beta*Mz/2/Mw;
sca[ep, k] = Mz^2*beta/2/Mw;
```

Z'decay matrix element squared technique

```
(*Matrix Elements Squared over ALL polarisations*)
Expand[WWZ[-p1, -p2, k, mu, nu, rho] * WWZ[-p1, -p2, k, mup, nup, rhop] *
  PolVVsq[Mw, p1, mu, mup] * PolVVsq[Mw, p2, nu, nup] * PolVVsq[Mz, k, rho, rhop]]
```

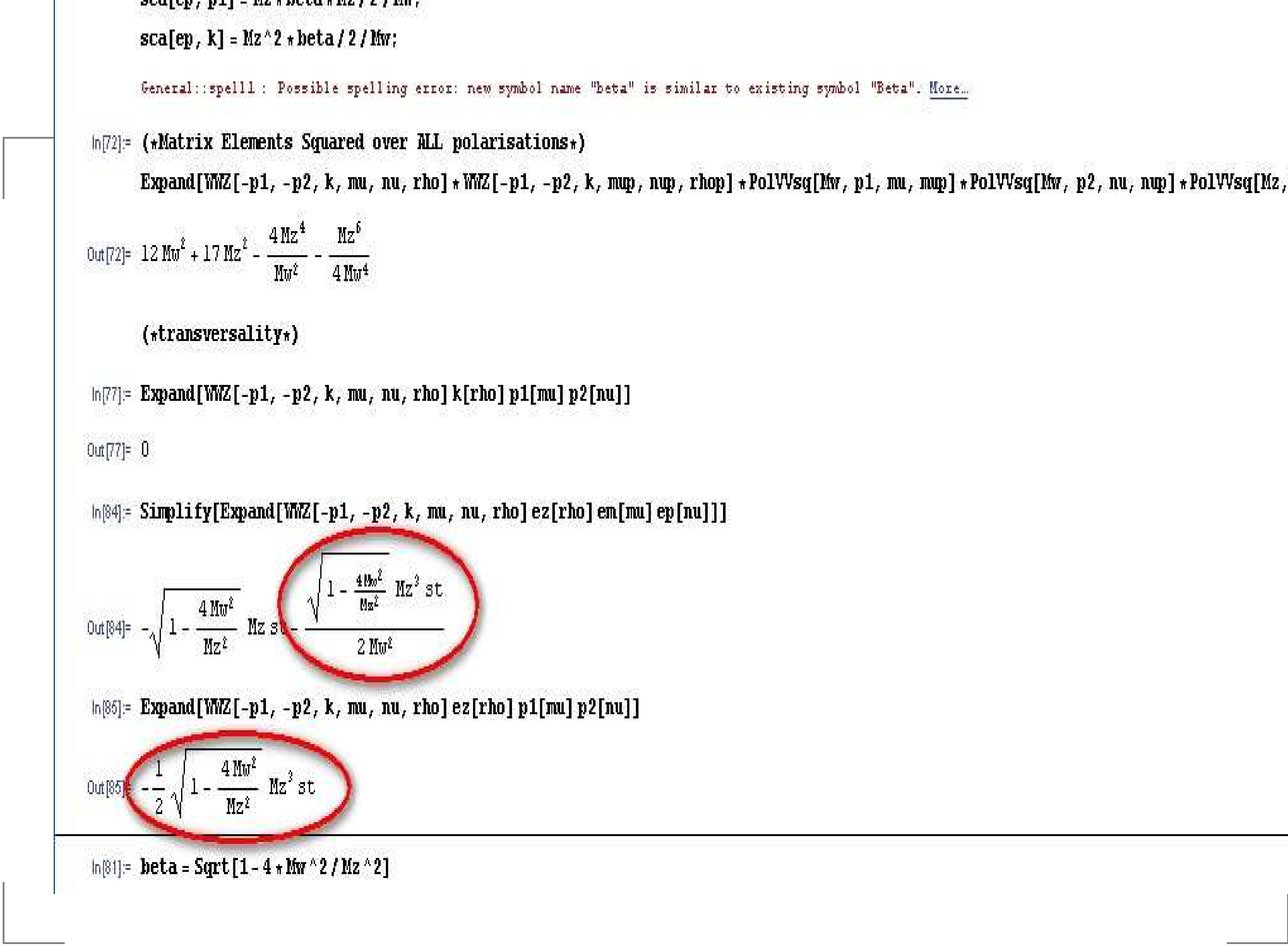


```
sca[p1, k] = Mz^2 / 2;
sca[p2, k] = Mz^2 / 2;
sca[ez, k] = 0;
sca[ez, p1] = -Mz * beta * st / 2;
sca[ez, p2] = +Mz * beta * st / 2;
sca[em, p1] = 0;
sca[em, k] = Mz^2 * beta / 2 / Mw;
sca[em, ep] = (1 + beta^2) * Mz^2 / 4 / Mw^2;
sca[em, ez] = -(Mz / 2 / Mw) * st;
sca[ep, ez] = (Mz / 2 / Mw) * st;
sca[ep, p2] = 0;
sca[em, p2] = Mz * beta * Mz / 2 / Mw;
sca[ep, p1] = Mz * beta * Mz / 2 / Mw;
sca[ep, k] = Mz^2 * beta / 2 / Mw;

General::spell1 : Possible spelling error: new symbol name "beta" is similar to existing symbol "Beta". More…

In[72]:= (*Matrix Elements Squared over ALL polarisations*)
         Expand[WWZ[-p1, -p2, k, mu, nu, rho] * WWZ[-p1, -p2, k, mup, nup, rhop] * PolVVsq[Mw, p1, mu, mup] * PolVVsq[Mw, p2, nu, nup] * P
```

Out[72]= $12\,Mw^2 + 17\,Mz^2 - \frac{4\,Mz^4}{Mw^2} - \frac{Mz^6}{4\,Mw^4}$

Z'decay helicity amplitude

```
Simplify[Expand[WWZ[-p1, -p2, k, mu, nu, rho] ez[rho] em[mu] ep[nu]]]
```



```
sca[ep, p1] = Mz * beta * Mz / 2 / Mw;
sca[ep, k] = Mz^2 * beta / 2 / Mw;

General::spell1 : Possible spelling error: new symbol name "beta" is similar to existing symbol "Beta". More…

In[72]:= (*Matrix Elements Squared over ALL polarisations*)
         Expand[WWZ[-p1, -p2, k, mu, nu, rho] * WWZ[-p1, -p2, k, mup, nup, rhop] * PolVVsq[Mw, p1, mu, mup] * PolVVsq[Mw, p2, nu, nup] * PolVVsq[Mz,
```

Out[72]=

$$12\,Mw^2+17\,Mz^2-\frac{4\,Mz^4}{Mw^2}-\frac{Mz^6}{4\,Mw^4}$$

```
(*transversality*)

In[77]:= Expand[WWZ[-p1, -p2, k, mu, nu, rho] k[rho] p1[mu] p2[nu]]

Out[77]= 0

In[84]:= Simplify[Expand[WWZ[-p1, -p2, k, mu, nu, rho] ez[rho] em[mu] ep[nu]]]
```

Out[84]=

$$-\sqrt{1-\frac{4\,Mw^2}{Mz^2}}\;Mz\;st-\frac{\sqrt{1-\frac{4\,Mw^2}{Mz^2}}\;Mz^3\;st}{2\,Mw^2}$$

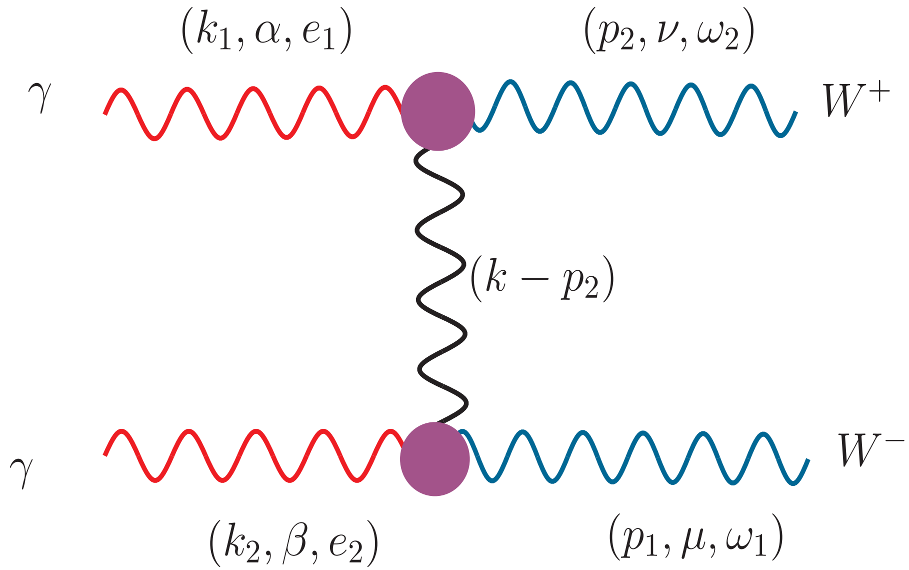
```
In[85]:= Expand[WWZ[-p1, -p2, k, mu, nu, rho] ez[rho] p1[mu] p2[nu]]
```

Out[85]=

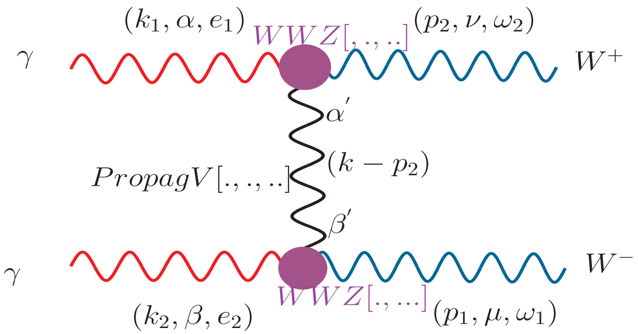
$$-\frac{1}{2}\sqrt{1-\frac{4\,Mw^2}{Mz^2}}\;Mz^3\;st$$

```
In[81]:= beta = Sqrt[1 - 4 * Mw^2 / Mz^2]
```

$\gamma\gamma \to W^+W^-$

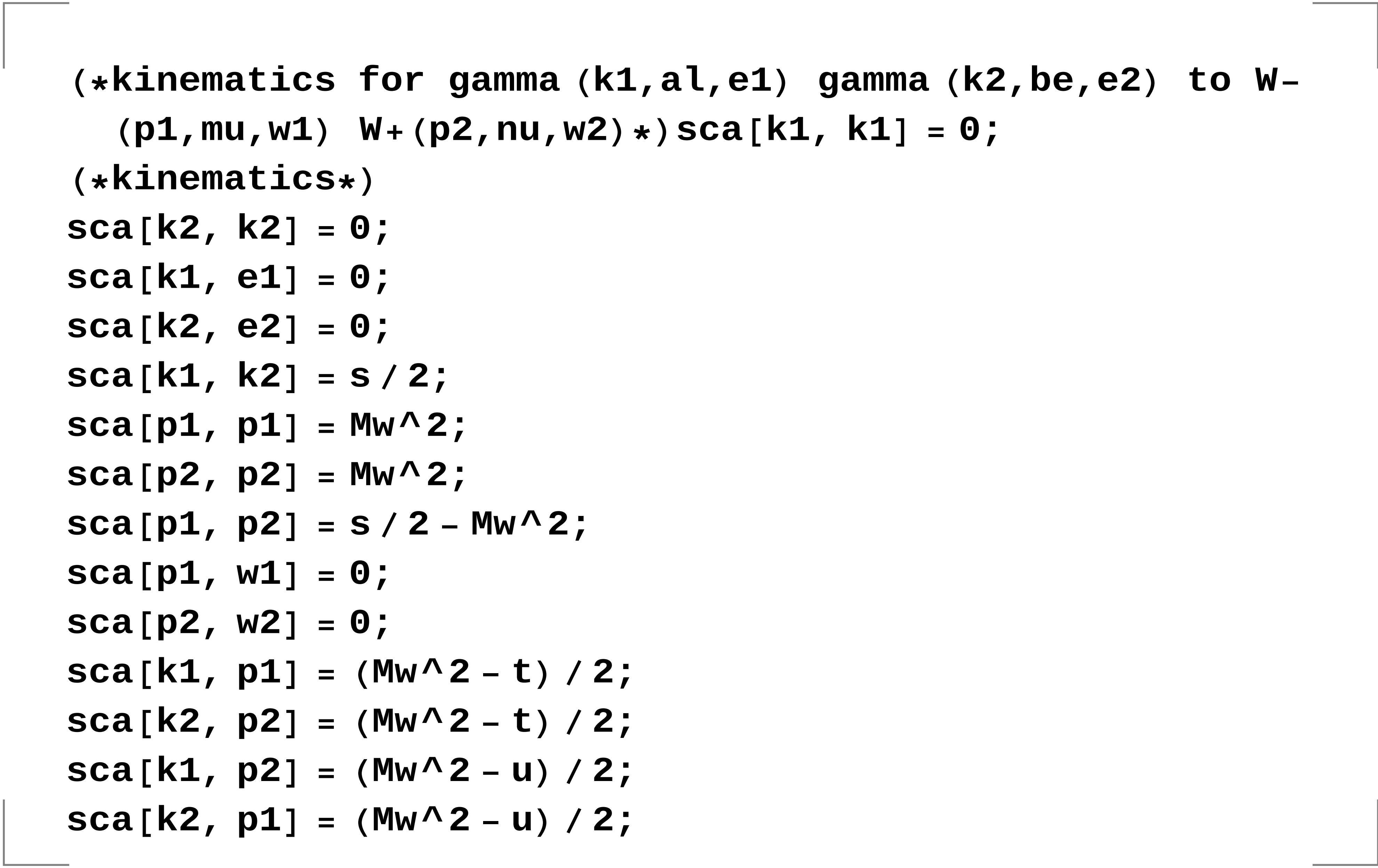


Writing the amplitude for $\gamma\gamma \to W^+W^-$



```
(*First diagram*)
S1abmn = WWZ[p2 - k1, -p2, k1, alp, nu, al]
   PropagV[Mw, k1 - p2, alp, bep] WWZ[k1 - p2, -p1, k2, bep, mu, be];
(*second diagram*)
S2abmn = WWZ[p2 - k2, -p2, k2, bep, nu, be]
   PropagV[Mw, k2 - p2, bep, alp] WWZ[k2 - p2, -p1, k1, alp, mu, al];
(*third diagram*)
Qabmn = -WWZZ[al, be, mu, nu];
```

Kinematics for $\gamma\gamma \to W^+W^-$



```
(*kinematics for gamma(k1,al,e1) gamma(k2,be,e2) to W-
  (p1,mu,w1) W+(p2,nu,w2)*)sca[k1, k1] = 0;
(*kinematics*)
sca[k2, k2] = 0;
sca[k1, e1] = 0;
sca[k2, e2] = 0;
sca[k1, k2] = s / 2;
sca[p1, p1] = Mw^2;
sca[p2, p2] = Mw^2;
sca[p1, p2] = s / 2 - Mw^2;
sca[p1, w1] = 0;
sca[p2, w2] = 0;
sca[k1, p1] = (Mw^2 - t) / 2;
sca[k2, p2] = (Mw^2 - t) / 2;
sca[k1, p2] = (Mw^2 - u) / 2;
sca[k2, p1] = (Mw^2 - u) / 2;
```

Helicity amplitude for $\gamma\gamma \to W^+W^-$

The photons with helicity λ_1 (λ_2) are in the $+z$ ($-z$) direction and the outgoing W^- (W^+) with helicity λ_- (λ_+) and 4-momentum p_- (p_+):

$$p^\mu_\mp = \frac{\sqrt{s}}{2}(1, \pm\beta\sin\theta, 0, \pm\beta\cos\theta) \quad ; \quad \beta = \sqrt{1-4/\gamma}\ ;\ \gamma = s/M_W^2 .$$

The polarisations for the helicity basis are defined as

$$\epsilon_1^\mu(\lambda_1) = \frac{1}{\sqrt{2}}(0, -\lambda_1, -i, 0) \qquad \epsilon_2^\mu(\lambda_2) = \frac{1}{\sqrt{2}}(0, \lambda_2, -i, 0) \quad \lambda_{1,2} = \pm$$

$$\epsilon_-^\mu(\lambda_-)^* = \frac{1}{\sqrt{2}}(0, -\lambda_- \cos\theta, i, \lambda_- \sin\theta) \qquad \epsilon_+^\mu(\lambda_+)^* = \frac{1}{\sqrt{2}}(0, \lambda_+ \cos\theta, i, -\lambda_+ \sin\theta) \quad \lambda_\pm =$$

$$\epsilon_-^\mu(0)^* = \frac{\sqrt{s}}{2M_W}(\beta, \sin\theta, 0, \cos\theta) \qquad \epsilon_+^\mu(0)^* = \frac{\sqrt{s}}{2M_W}(\beta, -\sin\theta, 0, -\cos\theta) \quad \lambda_\pm = 0.$$

Helicity amplitude for $\gamma\gamma \to W^+W^-$, pretty compact

$$\mathcal{M}_{\lambda_1\lambda_2;\lambda_-\lambda_+} = \frac{4\pi\alpha}{1-\beta^2\cos^2\theta}\;\mathcal{N}_{\lambda_1\lambda_2;\lambda_-\lambda_+},$$

where

$$\begin{aligned}
\mathcal{N}_{\lambda_1\lambda_2;00} &= -\frac{1}{\gamma}\left\{-4(1+\lambda_1\lambda_2)+(1-\lambda_1\lambda_2)(4+\gamma)\sin^2\theta\right\},\\
\mathcal{N}_{\lambda_1\lambda_2;\lambda_-0} &= \sqrt{\frac{8}{\gamma}}\;(\lambda_1-\lambda_2)(1+\lambda_1\lambda_-\cos\theta)\sin\theta, \qquad \lambda_-=\pm\\
\mathcal{N}_{\lambda_1\lambda_2;0,\lambda_+} &= -\sqrt{\frac{8}{\gamma}}\;(\lambda_1-\lambda_2)(1-\lambda_1\lambda_+\cos\theta)\sin\theta, \qquad \lambda_+=\pm\\
\mathcal{N}_{\lambda_1\lambda_2;\lambda_-\lambda_+} &= \beta(\lambda_1+\lambda_2)(\lambda_-+\lambda_+)+\frac{1}{2\gamma}\{-8\lambda_1\lambda_2(1+\lambda_-\lambda_+)\\
&\qquad +\gamma(1+\lambda_1\lambda_2\lambda_-\lambda_+)(3+\lambda_1\lambda_2)\\
&+\quad 2\gamma(\lambda_1-\lambda_2)(\lambda_--\lambda_+)\cos\theta-4(1-\lambda_1\lambda_2)(1+\lambda_-\lambda_+)\cos^2\theta\\
&+\quad \gamma(1-\lambda_1\lambda_2)(1-\lambda_-\lambda_+)\cos^2\theta\} \qquad\qquad \lambda_\pm=\pm.
\end{aligned}$$

Matrix Elements Generation and Automation: Feynman diagrams

Take as an example GRACE and $e^+e^- \rightarrow W^+W^-\gamma$

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Model="sm.mdl";
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Process;
  ELWK=3;
  Initial={electron, positron};
  Final  ={photon, W-plus, W-minus};
  Kinem="2302";
Pend;
```

Matrix Elements Generation and Automation: Feynman diagrams

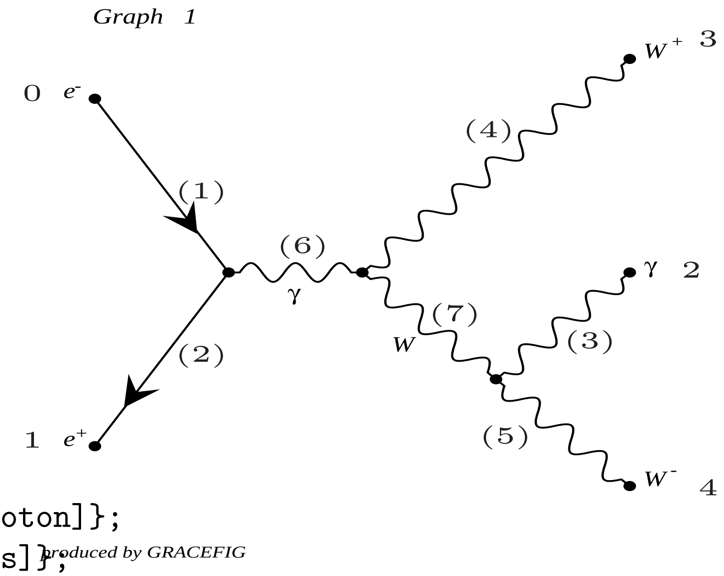
Take as an example `GRACE` and $e^+e^- \rightarrow W^+W^-\gamma$

1. Generate the number of vertices.
   The number of vertices is restricted by the order of the coupling constants for the physical process. Each vertex has a fixed number of propagators and external particles to be connected.
2. Connect vertices with propagators or external particles.
   There are multiple ways to connect vertices. All possible configuration are to be generated.
3. Particle assignment.
   Particles are assigned to propagators confirming that the connected vertex is defined in the model. As there will be many ways to assign particles to propagators, all possible configurations are to be generated.
4. Conservation laws such as electric charge and fermion numbers conservation will be employed in order to avoid fruitless trials.
5. Avoid duplication, use graph theory (edges and nodes)
6. `QGRAPH`: Powerful generator of graphs

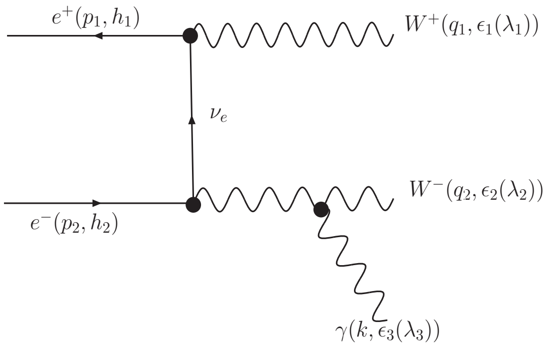
Matrix Elements Generation and Automation: Feynman diagrams

Take as an example GRACE and $e^+e^- \to W^+W^-\gamma$

```
Process=1; External=5;
      0= initial electron;
      1= initial positron;
      2= final photon;
      3= final w-plus;
      4= final w-minus;
   Eend; elwk=3;Loop=0;

   Graph=1; Gtype=1; Sfactor=-1; Vertex=3;
      0={      1[positron]};
      1={      2[electron]};
      2={      3[photon]};
      3={      4[w-plus]};
      4={      5[w-minus]};
      5[order={1,0}]={     1[electron],     2[positron],     6[photon]};
      6[order={1,0}]={     4[w-minus],     6[photon],     7[w-plus]};
      7[order={1,0}]={     3[photon],     5[w-plus],     7[w-minus]};
   Vend; Gend;

   Graph=2;
       ...
```



Matrix Elements Generation and Automation: Helicity amplitude, How it works



$$T_{fi}\big((p_1,h_1),(p_2,h_2),(q_1,\lambda_1),(q_2,\lambda_2),(k,\lambda_3)\big) =$$

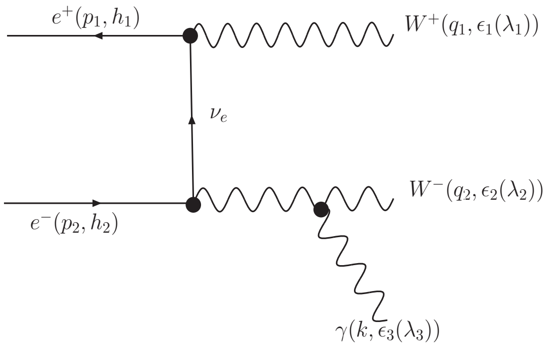
$$\overline{v}(p_1,h_1)\, c^{\eta}_{eW}\, \epsilon_{1\eta}(q_1)\, S_F(-p_1+q_1,0)\, c^{\mu}_{eW}\, u(p_2,h_2)$$

$$\times D_{V\,\mu\nu}(q_2+k,M_W)\, c^{\nu\rho\sigma}_{WW\gamma}(q_2+k,-q_2,-k)\, \epsilon_{2\rho}(q_2)\, \epsilon_{3\sigma}(k)$$

$$c^{\mu}_{eW} = \frac{eM_Z}{\sqrt{2(M_Z^2-M_W^2)}}\gamma^{\mu}\frac{1-\gamma_5}{2}$$

$$c^{\nu\rho\sigma}_{WW\gamma}(p,q,r) = e[(p-q)^{\sigma}g^{\nu\rho}+(q-r)^{\nu}g^{\rho\sigma}+(r-p)^{\rho}g^{\sigma\nu}]$$

Matrix Elements Generation and Automation: Helicity amplitude, How it works



$$T_{fi}\big((p_1,h_1),(p_2,h_2),(q_1,\lambda_1),(q_2,\lambda_2),(k,\lambda_3)\big) =$$

$$\overline{v}(p_1,h_1)\, c^{\eta}_{eW}\, \epsilon_{1\eta}(q_1)\, S_F(-p_1+q_1,0)\, c^{\mu}_{eW}\, u(p_2,h_2)$$

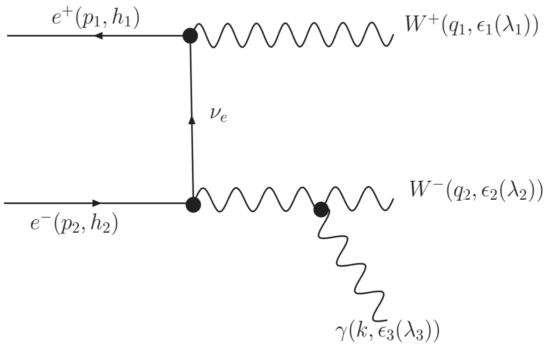
$$\times D_{V\,\mu\nu}(q_2+k,M_W)\, c^{\nu\rho\sigma}_{WW\gamma}(q_2+k,-q_2,-k)\, \epsilon_{2\rho}(q_2)\, \epsilon_{3\sigma}(k)$$

$$c^{\mu}_{eW} = \frac{eM_Z}{\sqrt{2(M_Z^2-M_W^2)}}\gamma^{\mu}\frac{1-\gamma_5}{2}$$

$$c^{\nu\rho\sigma}_{WW\gamma}(p,q,r) = e[(p-q)^{\sigma}g^{\nu\rho}+(q-r)^{\nu}g^{\rho\sigma}+(r-p)^{\rho}g^{\sigma\nu}]$$

$$S_F(p,m) = (\not{p}+m)D(p,m) \qquad D_{V\,\mu\nu}(p) = \Big(-g_{\mu\nu}+\frac{p_\mu p_\nu}{M^2}\Big)D(p,m) \qquad D(p,m) = \frac{1}{p^2-m^2}$$

Matrix Elements Generation and Automation: Helicity amplitude, How it works



$$T_{fi}\big((p_1,h_1),(p_2,h_2),(q_1,\lambda_1),(q_2,\lambda_2),(k,\lambda_3)\big) =$$

$$\overline{v}(p_1,h_1)\, c^{\eta}_{eW}\, \epsilon_{1\eta}(q_1)\, S_F(-p_1+q_1,0)\, c^{\mu}_{eW}\, u(p_2,h_2)$$

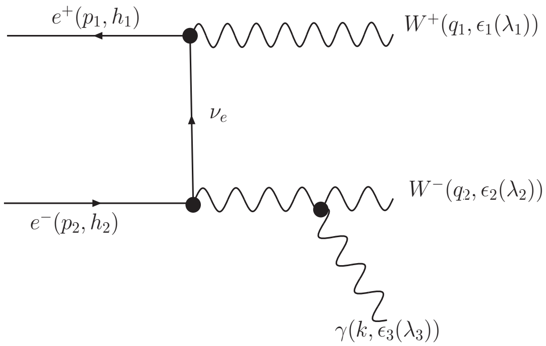
$$\times D_{V\,\mu\nu}(q_2+k,M_W)\, c^{\nu\rho\sigma}_{WW\gamma}(q_2+k,-q_2,-k)\, \epsilon_{2\rho}(q_2)\, \epsilon_{3\sigma}(k)$$

$$c^{\mu}_{eW} = \frac{eM_Z}{\sqrt{2(M_Z^2-M_W^2)}}\gamma^{\mu}\,\frac{1-\gamma_5}{2}$$

$$c^{\nu\rho\sigma}_{WW\gamma}(p,q,r) = e[(p-q)^{\sigma}g^{\nu\rho} + (q-r)^{\nu}g^{\rho\sigma} + (r-p)^{\rho}g^{\sigma\nu}]$$

$$S_F(p) = \frac{\sum_{\alpha i} w_{\alpha,i}\, U^{\alpha}(h^{(i)},p^{(i)})\, \overline{U}^{\alpha}(h^{(i)},p^{(i)})}{p^2-m^2}, \qquad D_{V\,\mu\nu}(p) = \frac{\sum_i\, w_i\, \epsilon^{(i)}_{\mu}(p)\, \epsilon^{(i)}_{\nu}(p)}{p^2-m^2}$$

Matrix Elements Generation and Automation: Helicity amplitude, How it works



$$T_{fi}\big((p_1,h_1),(p_2,h_2),(q_1,\lambda_1),(q_2,\lambda_2),(k,\lambda_3)\big) =$$

$$\overline{v}(p_1,h_1)\, c^{\eta}_{eW}\, \epsilon_{1\eta}(q_1)\, S_F(-p_1+q_1,0)\, c^{\mu}_{eW}\, u(p_2,h_2)$$

$$\times D_{V\,\mu\nu}(q_2+k,M_W)\, c^{\nu\rho\sigma}_{WW\gamma}(q_2+k,-q_2,-k)\, \epsilon_{2\rho}(q_2)\, \epsilon_{3\sigma}(k)$$

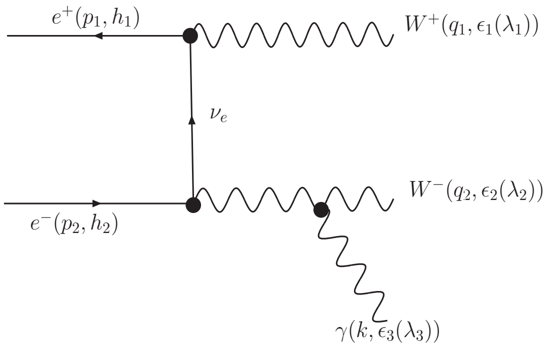
$$c^{\mu}_{eW} = \frac{eM_Z}{\sqrt{2(M_Z^2-M_W^2)}}\gamma^{\mu}\frac{1-\gamma_5}{2}$$

$$c^{\nu\rho\sigma}_{WW\gamma}(p,q,r) = e[(p-q)^{\sigma}g^{\nu\rho}+(q-r)^{\nu}g^{\rho\sigma}+(r-p)^{\rho}g^{\sigma\nu}]$$

$$S_F(p) = \frac{\sum_{\alpha i} w_{\alpha,i}\, U^{\alpha}(h^{(i)},p^{(i)})\, \overline{U}^{\alpha}(h^{(i)},p^{(i)})}{p^2-m^2}, \qquad D_{V\,\mu\nu}(p) = \frac{\sum_i w_i\, \epsilon^{(i)}_{\mu}(p)\, \epsilon^{(i)}_{\nu}(p)}{p^2-m^2}$$

$$T_{fi} = D(-p_1+q_1,0)\; D(q_2+k,m_W)\sum_{\alpha,i} w_{\alpha,i}\sum_{l} w_l \times V^{(\alpha,i)}_{eW+}\, V^{(\alpha,i,l)}_{eW-}\, V^{(l)}_{WW\gamma},$$

Matrix Elements Generation and Automation: Helicity amplitude, How it works



$$T_{fi}\big((p_1,h_1),(p_2,h_2),(q_1,\lambda_1),(q_2,\lambda_2),(k,\lambda_3)\big) =$$

$$\overline{v}(p_1,h_1)\, c^{\eta}_{eW}\, \epsilon_{1\eta}(q_1)\, S_F(-p_1+q_1,0)\, c^{\mu}_{eW}\, u(p_2,h_2)$$

$$\times D_{V\,\mu\nu}(q_2+k,M_W)\, c^{\nu\rho\sigma}_{WW\gamma}(q_2+k,-q_2,-k)\, \epsilon_{2\rho}(q_2)\, \epsilon_{3\sigma}(k)$$

$$c^{\mu}_{eW} = \frac{eM_Z}{\sqrt{2(M_Z^2-M_W^2)}}\,\gamma^{\mu}\,\frac{1-\gamma_5}{2}$$

$$c^{\nu\rho\sigma}_{WW\gamma}(p,q,r) = e[(p-q)^{\sigma}g^{\nu\rho}+(q-r)^{\nu}g^{\rho\sigma}+(r-p)^{\rho}g^{\sigma\nu}]$$

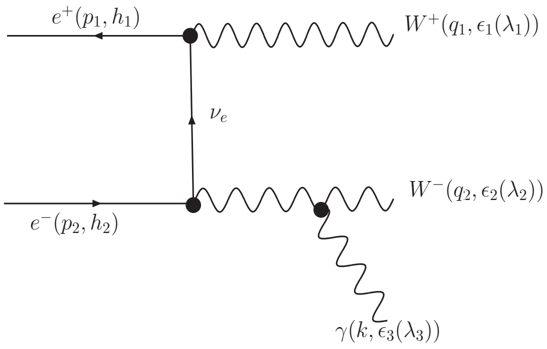
$$S_F(p) = \frac{\sum_{\alpha i} w_{\alpha,i}\, U^{\alpha}(h^{(i)},p^{(i)})\,\overline{U}^{\alpha}(h^{(i)},p^{(i)})}{p^2-m^2}, \qquad D_{V\,\mu\nu}(p) = \frac{\sum_i w_i\, \epsilon^{(i)}_{\mu}(p)\, \epsilon^{(i)}_{\nu}(p)}{p^2-m^2}$$

$$T_{fi} = D(-p_1+q_1,0)\; D(q_2+k,m_W)\; \sum_{\alpha,i} w_{\alpha,i} \sum_{l} w_l \times V^{(\alpha,i)}_{eW+}\, V^{(\alpha,i,l)}_{eW-}\, V^{(l)}_{WW\gamma},$$

The building blocks: c-numbers Library subroutines

$$\begin{aligned}
V^{(\alpha,i)}_{eW+} &= \overline{v}(p_1,h_1)\, c^{\eta}_{eW}\, \epsilon_{1\eta}(q_1)\, U^{\alpha}((-p_1+q_1)^{(i)},h^{(i)}), \quad \text{FFV}\\
V^{(\alpha,i,l)}_{eW-} &= \overline{U}^{\alpha}(p^{(i)},h^{(i)})\, c^{\mu}_{eW}\, \epsilon^{(l)}_{\mu}(q_2+k)\, u(p_2,h_2),\\
V^{(l)}_{WW\gamma} &= c^{\nu\rho\sigma}_{WW\gamma}(q_2+k,-q_2,-k)\, \epsilon^{(l)}_{\nu}(q_2+k)\, \epsilon_{2\rho}(q_2)\, \epsilon_{3\sigma}(k) \quad \text{VVV}.
\end{aligned}$$

Matrix Elements Generation and Automation: Helicity amplitude, How it works



$$T_{fi}((p_1,h_1),(p_2,h_2),(q_1,\lambda_1),(q_2,\lambda_2),(k,\lambda_3)) =$$

$$\overline{v}(p_1,h_1)\, c^{\eta}_{eW}\, \epsilon_{1\eta}(q_1)\, S_F(-p_1+q_1,0)\, c^{\mu}_{eW}\, u(p_2,h_2)$$

$$\times D_{V\ \mu\nu}(q_2+k,M_W)\, c^{\nu\rho\sigma}_{WW\gamma}(q_2+k,-q_2,-k)\, \epsilon_{2\rho}(q_2)\, \epsilon_{3\sigma}(k)$$

$$c^{\mu}_{eW} = \frac{eM_Z}{\sqrt{2(M_Z^2-M_W^2)}}\,\gamma^{\mu}\,\frac{1-\gamma_5}{2}$$

$$c^{\nu\rho\sigma}_{WW\gamma}(p,q,r) = e[(p-q)^{\sigma}g^{\nu\rho}+(q-r)^{\nu}g^{\rho\sigma}+(r-p)^{\rho}g^{\sigma\nu}]$$

$$S_F(p) = \frac{\sum_{\alpha i} w_{\alpha,i}\, U^{\alpha}(h^{(i)},p^{(i)})\, \overline{U}^{\alpha}(h^{(i)},p^{(i)})}{p^2-m^2}, \qquad D_{V\ \mu\nu}(p) = \frac{\sum_i w_i\, \epsilon^{(i)}_{\mu}(p)\, \epsilon^{(i)}_{\nu}(p)}{p^2-m^2}$$

$$T_{fi} = D(-p_1+q_1,0)\; D(q_2+k,m_W)\, \sum_{\alpha,i} w_{\alpha,i} \sum_{l} w_l \times V^{(\alpha,i)}_{eW+}\, V^{(\alpha,i,l)}_{eW-}\, V^{(l)}_{WW\gamma},$$

The building blocks: c-numbers Library. If New Physics? extend library?

read in new Feynman rules, calculate new entries for subroutines

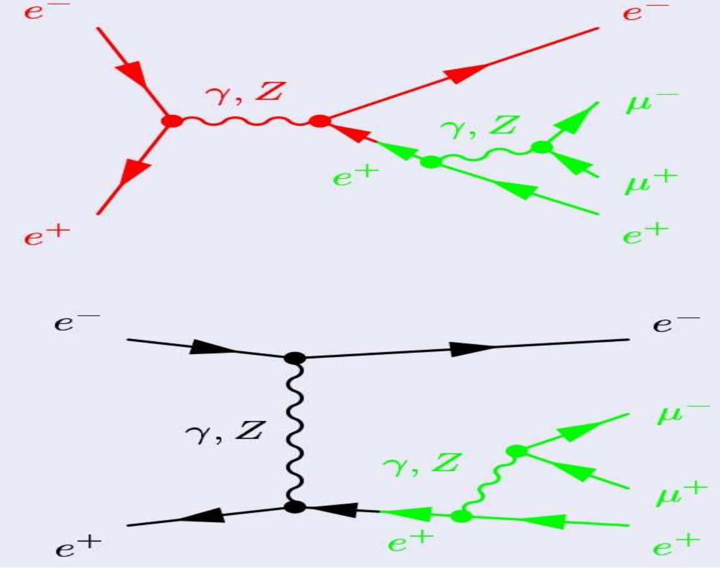
$$\begin{aligned}
V^{(\alpha,i)}_{eW+} &= \overline{v}(p_1,h_1)\, c^{\eta}_{eW,NP}\, \epsilon_{1\eta}(q_1)\, U^{\alpha}((-p_1+q_1)^{(i)},h^{(i)}),\\
V^{(\alpha,i,l)}_{eW-} &= \overline{U}^{\alpha}(p^{(i)},h^{(i)})\, c^{\mu}_{eW,NP}\, \epsilon^{(l)}_{\mu}(q_2+k)\, u(p_2,h_2),\\
V^{(l)}_{WW\gamma} &= c^{\nu\rho\sigma}_{WW\gamma,NP}(q_2+k,-q_2,-k)\, \epsilon^{(l)}_{\nu}(q_2+k)\, \epsilon_{2\rho}(q_2)\, \epsilon_{3\sigma}(k).
\end{aligned}$$

Matrix Elements Generation and Automation: Helicity amplitude, CHANEL(GRACE)/HELAS(MadGraph

- This is now used in SHERPA and HERWIG
- The method is purely numerical.
- The amplitude for each Feynman graph is first decomposed into vertex sub-amplitudes
- Each of these sub-amplitudes is read from a pre-defined model file library
- **drawback:** Libraries exists for Standard Couplings (renormalisable),
  for example $(a + b\gamma\gamma_5)\gamma_\mu$ type OK
  higher order operators need to be generated from scratch
  anomalous VVV couplings not assuming the general gauge VVV need to be generated
- **ALOHA** is on the way
- spin > 2 (but even theory needs firm ground)

Speed up

can speed up by reusing common pieces `GRACE, AMEGIC`



MEG, Matrix Elements Generators (Tree-level)

Int/Amp.	Squaring	Helicity	Off-Shell
Adaptive	CompHEP/CalcHEP	GRACE	ALPGEN
Multi-Channel	-	MadGraph/Sherpa	
	-	HELAC/Whizard	

More automation

but we need to feed in the Feynman rules

what if the new physics is like the MSSM? huge number of vertices?

need to input new models quickly and efficiently

LanHEP (A. Semenov) as prototype for automatic Feynman rules generation

`http://theory.sinp.msu.ru/˜semenov/lanhep.html`

- LanHEP was developed since 1994 as a part of CompHEP project to help to create new models (complete set of Feynamn rules) starting from the Lagrangian, the first goal was MSSM.
- can now output to FeynArts/FeynCalc
- Lagrangian writes in a texbook format, outputs also to LateX
- extremely powerful, extended to one-loop: generates counterterms and new vertices
- A model in a MEG (CompHEP/CalcHEP/FeynArts,..) is defined by the tables of parameters, particles and interaction vertices with implicit Lorentz structure.
- Flexible model format allows to introduce into these MEG new gauge theories as well as various anomalous terms.
- Not restricted to dim-4 (renormalisable) operators.
- Gauge theories highly automated (gauge-fixing, ghost, BRST)
- Powerful use of compact objects (multiplets, supermultiplets,..)

and thus SUSY-friendly: 2-component fermions and superpotential notation

LanHEP as prototype for automatic Feynman rules generation

- The LanHEP program is written in C, external mathematical software is NOT required.
- LanHEP reads an input file which describes the physical model by a set of statements.
- Large projects can be split into several files.
- Conditional processing of the model file allows the user to use the same input file(s) for several species of the physical model. This feature allows, for example, to chose gauge fixing and MSSM extensions by setting some switches instead of creating several slightly different input files.
- Command-line tool: no graphical interface means easy compilation on any platform where 32-bit C compiler exists.

An example: Lanhep in CompHEP/CalcHEP

A physical model in CompHEP/CalcHEP is defined by the (3/4) tables of

- parameters
- particles
- interaction vertices with implicit Lorentz structure (any Lorentz structure is allowed)
- a file for book-keeping (constraints, dependent parameters)

A physical model in CompHEP is defined by the tables of parameters, particles and interaction vertices

Parameters

```
EE    |0.31345      |Electromagnetic coupling constant (<->1/127.9)
MW    |MZ*CW
```

Particles

```
photon       |A  |A  |2     |0     |0     |1    |G  |A
Z boson      |Z  |Z  |2     |MZ    |wZ    |1    |G  |Z
W boson      |W+ |W- |2     |MW    |wW    |1    |G  |W^+
electron     |e  |E  |1     |Me    |0     |1    |   |e
```

Vertices

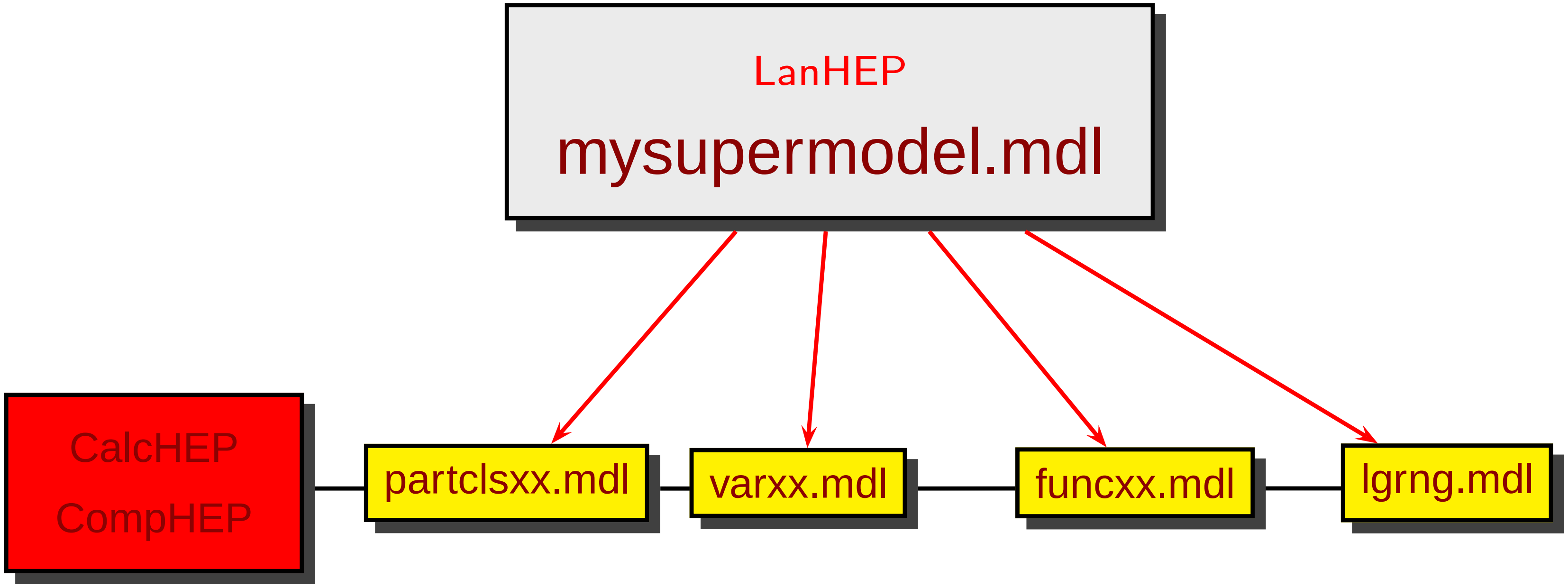
```
E    |e    |A    |     |EE                      |G(m3)
E    |e    |H    |     |-EE*Me*ca/(2*MW*SW*cb)  |1
E    |e    |H3   |     |i*EE*Me*tb/(2*MW*SW)    |G5
E    |e    |Z    |     |EE/(2*S2W)              |C2W*G(m3)*(1-G5)-2*SW^2*
E    |e    |Z.f  |     |-i*EE*Me/(2*MW*SW)      |G5
E    |e    |h    |     |EE*Me*sa/(2*MW*SW*cb)   |1
E    |ne   |H-   |     |EE*Me*Sqrt2*tb/(4*MW*SW)|(1-G5)
E    |ne   |W-   |     |-EE*Sqrt2/(4*SW)        |G(m3)*(1-G5)
E    |ne   |W-.f |     |-EE*Me*Sqrt2/(4*MW*SW)  |(1-G5)
```

LanHEP to MEG

LanHEP

mysupermodel.mdl

LanHEP to MEG



LanHEP

mysupermodel.mdl

CalcHEP

CompHEP

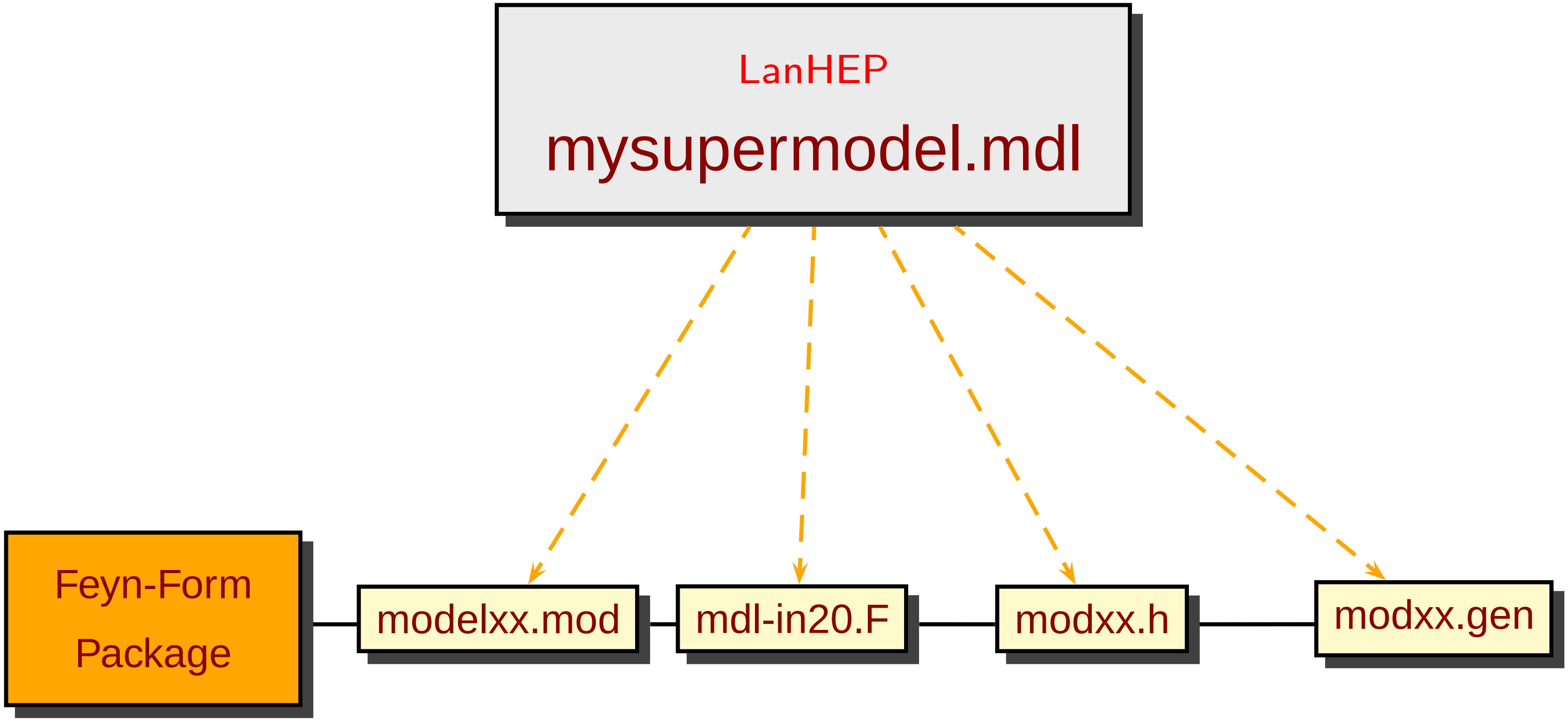
partclsxx.mdl

varxx.mdl

funcxx.mdl

lgrng.mdl

LanHEP to MEG



LanHEP

mysupermodel.mdl

Feyn-Form

Package

modelxx.mod

mdl-in20.F

modxx.h

modxx.gen

Description of the physical model for LanHEP

- The user declares the physical parameters to be included in the Lagrangian. The value of a parameter can be a number or an expression:

```
parameter ee=0.31333:'elementary electric charge'.
parameter sw=0.478:'sinus of weak angle'.
parameter cw=Sqrt(1-sw**2):'cosine of weak angle'.
```

- The user declares scalar, spinor, vector, (also spin 3/2 and 2) particles . It is possible to prescribe the colour structure for a particle:

```
spinor e1/E1:(electron, mass Me=0.000511).
spinor q/Q:(quark, color c3, mass Mq=10).
vector A/A:(photon, gauge).
```

- New symmetry groups are also possible. They can be defined in a way like color $SU(3)$ symmetry is defined, as well as corresponding matrices and structure constants:

```
group color:SU(3).
repres color:(c3/c3b,c8).
special lambda:(color c3, color c3b, color c8).
```

Description of the physical model for LanHEP **(cont)**

- The user can define the substitution rules, for example for covariant derivative $F^{\mu\nu} = \partial^\mu A^\nu - \partial^\nu A^\mu$:

```
let F^mu^nu = deriv^mu*A^nu - deriv^nu*A^mu.
```

- It is possible to define multiplets, and their components:

```
let l1 = {n1,e1}, L1 = {N1,E1}.
```

- The user can write Lagrangian terms with Lorenz and multiplet indices explicitly or omit indices (all or part of them):*(QED vertex* $\bar{e}(x)\gamma^\mu A_\mu(x)e(x)$*)*

```
lterm E1^a*gamma^a^b^mu*A^mu*e1^b.
lterm E1*gamma^mu*A^mu*e1.
lterm E1*gamma*A*e1.
```

- LanHEP performs explicit summation over the indices of Lagrangian terms, if the corresponding components for multiplets and matrices are introduced.

LanHEP features

- 2-component fermion notation makes possible the introduction of supersymmetric Lagrangian in a more natural way, closer to the form used in most textbooks on the supersymmetry.
- Superpotential can be used for supersymmetric theories; this option allows to introduce easily various extensions of MSSM (R-parity violation, NMSSM, etc). Yukawa and F^*F terms are now automatically derived by the program.
- Generating Hermitian conjugate terms allow to simplify model description.
- Constructing the ghost Lagrangian from BRST transformation.
- Conterterms can be generated if the necassy shifts for parameters and fields are prescribed.

LanHEP features

- Checking the correctness of the model
  - Electric charge conservation
  - Hermiticity
  - Probing kinetic and mass terms, the mass matrix is extracted
  - BRST invariance
  - Extracting classes of vertices
- Simplifying the expression for vertices
  - Orthogonal (and hermitian) matrices
  - Trigonometric expressions $(\sin\alpha \pm \beta)$
  - Lengthly expressions in the vertices can be transferred to the table of parameters.

Some new Lagrangians implemented by LanHEP

- Complete MSSM in unitary and t'Hooft-Feynman gauges with the Higgs sector by linking with the FeynHiggs, effective potential is used to take into account radiative corrections to Higgs masses and interaction; mSUGRA and GMSB by means of SLHA interface
- MSSM extensions include:
    - MSSM with R-parity violation
    - Model with gravitino and sgoldstinos
    - NMSSM (an extension of the MSSM by a gauge singlet N with hypercharge 0)
    - MSSM with CP violation
- Complete Leptoquark model which includes Yukawa couplings for all types of LQ, gauge couplings and anomalous gauge couplings for vector LQ
- Complete two-Higgs-doublet model with conserved or broken CP invariance
- Anomalous quartic vector bosons self-couplings

More new Lagrangians implemented by LanHEP

- A new signature for color octet pseudoscalars at the LHC, in theories of extra-dim. Alfonso R. Zerwekh, Claudio O. Dib, Rogerio Rosenfeld;
- Minimal Higgsless model, Chivukula et al;
- Inert Doublet Model, Pierce and Thaler;
- Excited fermions, Boos et al;
- Technihadrons, technicolour, Zerwekh;
- Little Higgs Models, Phenomenology of littlest Higgs model with T-parity: including effects of T-odd fermions. Alexander Belyaev, Chuan-Ren Chen, Kazuhiro Tobe, C.-P. Yuan (Michigan State U.);
- Universal extra-dim, Matchev et al.

Particle table format can be tuned

New option allows to modify the format of the output particle table and to add new proprties (new columns in the table). One can add, say, PDG particle number to the table:

```
prtcformat fullname:' Full   Name ',
           name:' p ',
           aname:' ap',
           spin2,color,mass,width, aux,
           pdg:'PDG ID',
           texname:'  latex P name ',
           atexname:' latex aP name ' .
```

Then the new property value can be written in the particle declaration statement:

```
scalar h:(higgs, mass Mh, pdg 123, width wh).
```

- Electric charge can be extracted automatically from the photon interaction and then added to the table.

Vertices table format: explicit colour structure

Color matrices and dot products can be optionally written in the Lorentz Part, e.g. QCD plus quark-photon interactions produces the following vertices file:

```
P1   |P2   |P3   |P4   |>   Factor   <|> dLagrangian/ dA(p1) dA(p2) dA(p3
G    |G    |G    |     |gg            |m2.p3*m1.m3*F(c1,c2,c3)
     |     |     |     |               -m1.p3*m2.m3*F(c1,c2,c3)
     |     |     |     |               +m3.p1*m1.m2*F(c1,c2,c3)
     |     |     |     |               -m2.p1*m1.m3*F(c1,c2,c3)
     |     |     |     |               -m3.p2*m1.m2*F(c1,c2,c3)
     |     |     |     |               +m1.p2*m2.m3*F(c1,c2,c3)
G.C  |G.c  |G    |     |-gg           |m3.p2*F(c1,c2,c3)
Q    |q    |G    |     |gg            |L(c1,c2,c3)*G(m3)
Q    |q    |A    |     |ee/3          |c1.c2*G(m3)
G    |G    |G    |G    |gg^2          |m1.m3*m2.m4*F(c1,c2,c0)*F(c3,c4,c0
     |     |     |     |               -m1.m4*m2.m3*F(c1,c2,c0)*F(c3,c4,c
     |     |     |     |               +m1.m2*m3.m4*F(c1,c3,c0)*F(c2,c4,c
     |     |     |     |               -m1.m4*m2.m3*F(c1,c3,c0)*F(c2,c4,c
     |     |     |     |               +m1.m2*m3.m4*F(c1,c4,c0)*F(c2,c3,c
     |     |     |     |               -m1.m3*m2.m4*F(c1,c4,c0)*F(c2,c3,c
```

Claude Duhr (in collaboration with N. D. Christensen and B. Fuks)

`http://feynrules.phys.ucl.ac.be`

- FeynRules has been developed since 2008 originally as a part of the MadGraph
- FeynRules is a Mathematica package that allows to derive Feynman rules from a Lagrangian.
- The syntax of FeynRules is an extension of the syntax used in FeynArts
- The only requirements on the Lagrangian are:

  All indices need to be contracted (Lorentz and gauge invariance)

  Locality

  Supported field types: spin 0, 1/2, 1, 2 and ghosts (ghost Lagrangian not automatically derived though)
- In progress

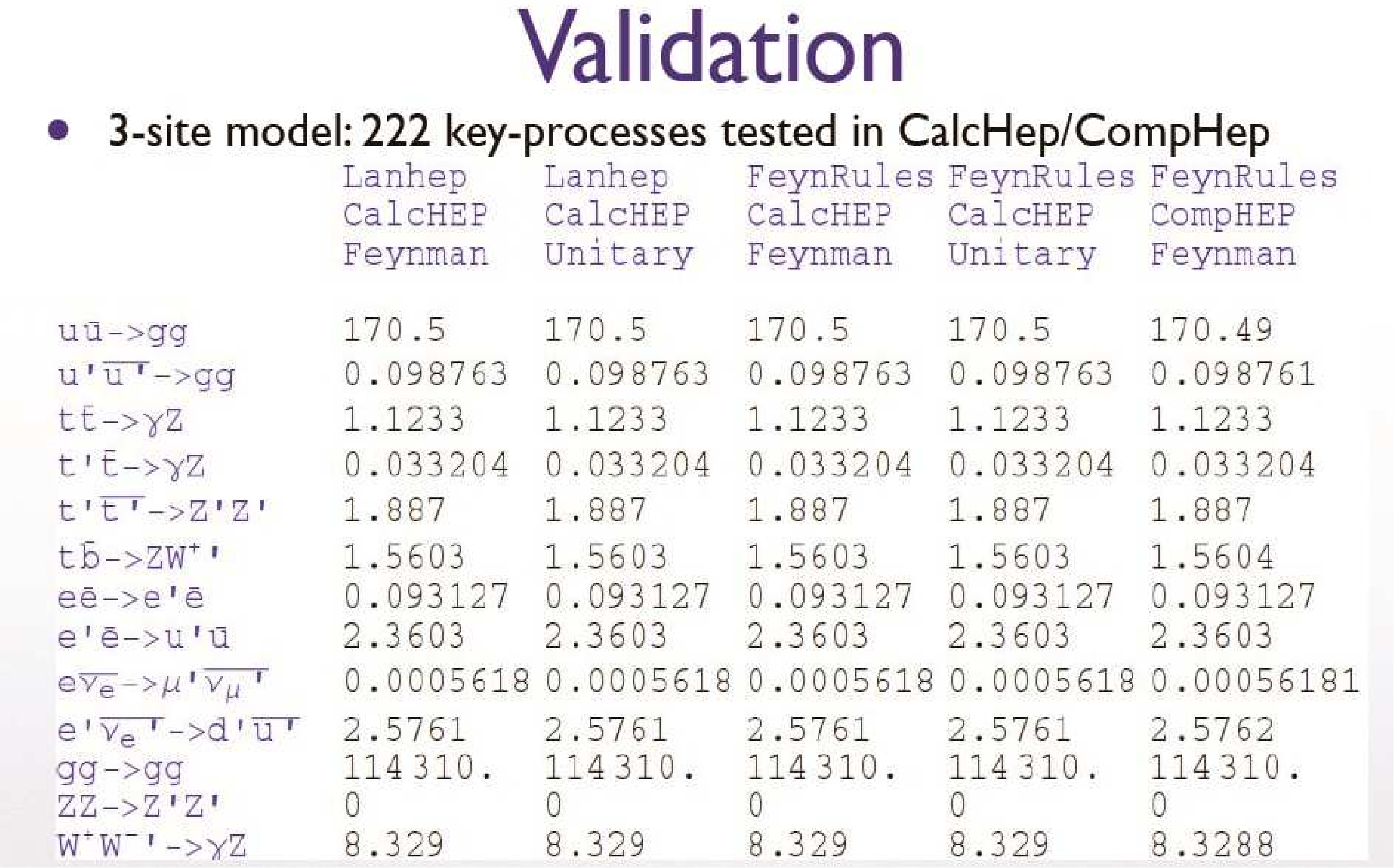
  Support for Weyl fermions and superfields

  Diagonalisation of mass matrices
- can export the Feynman rules into a TeX file.

Implemented Models in FeynRules

- Standard Model (CD, N. Christensen)
- Most general two Higgs doublet model (CD, M. Herquet)
- Minimal Higgsless Model (N. Christensen)
- Validation of the models:
- Full MSSM (B. Fuks)
- NMSSM (B. Fuks)
- R-symmetric MSSM (B. Fuks)
- RPV MSSM (B. Fuks)
- Universal Extra Dimensions (P. de Aquino)
- Large extra dimensions (P. de Aquino)
- Randall-Sundrum I (P. de Aquino)
- Strongly interacting Little Higgs (C. Degrande)
- Composite Top model (C. Degrande)
- Chiral perturbation theory (C. Degrande)

FeynRules early validation

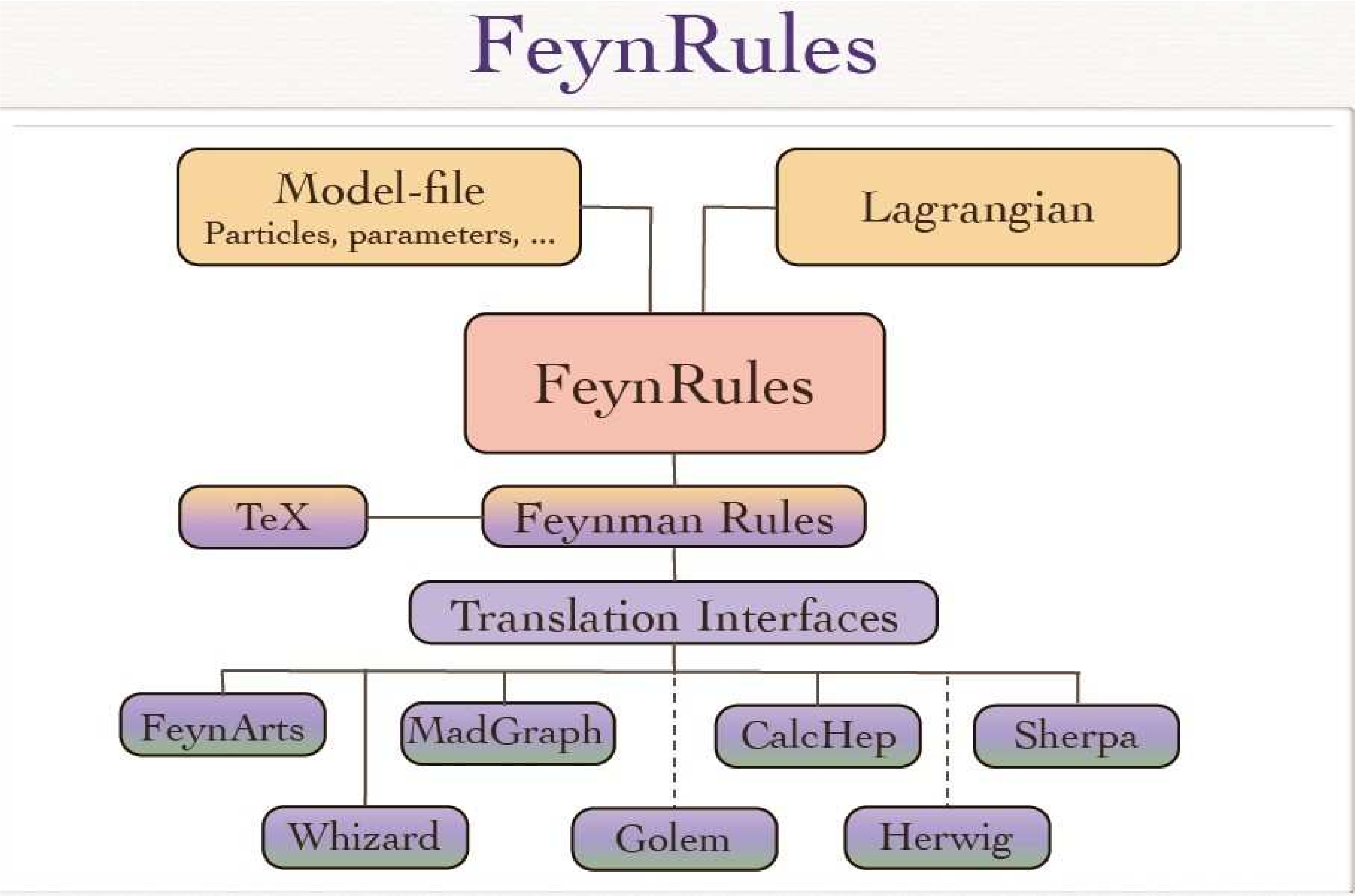


Validation

- 3-site model: 222 key-processes tested in CalcHep/CompHep

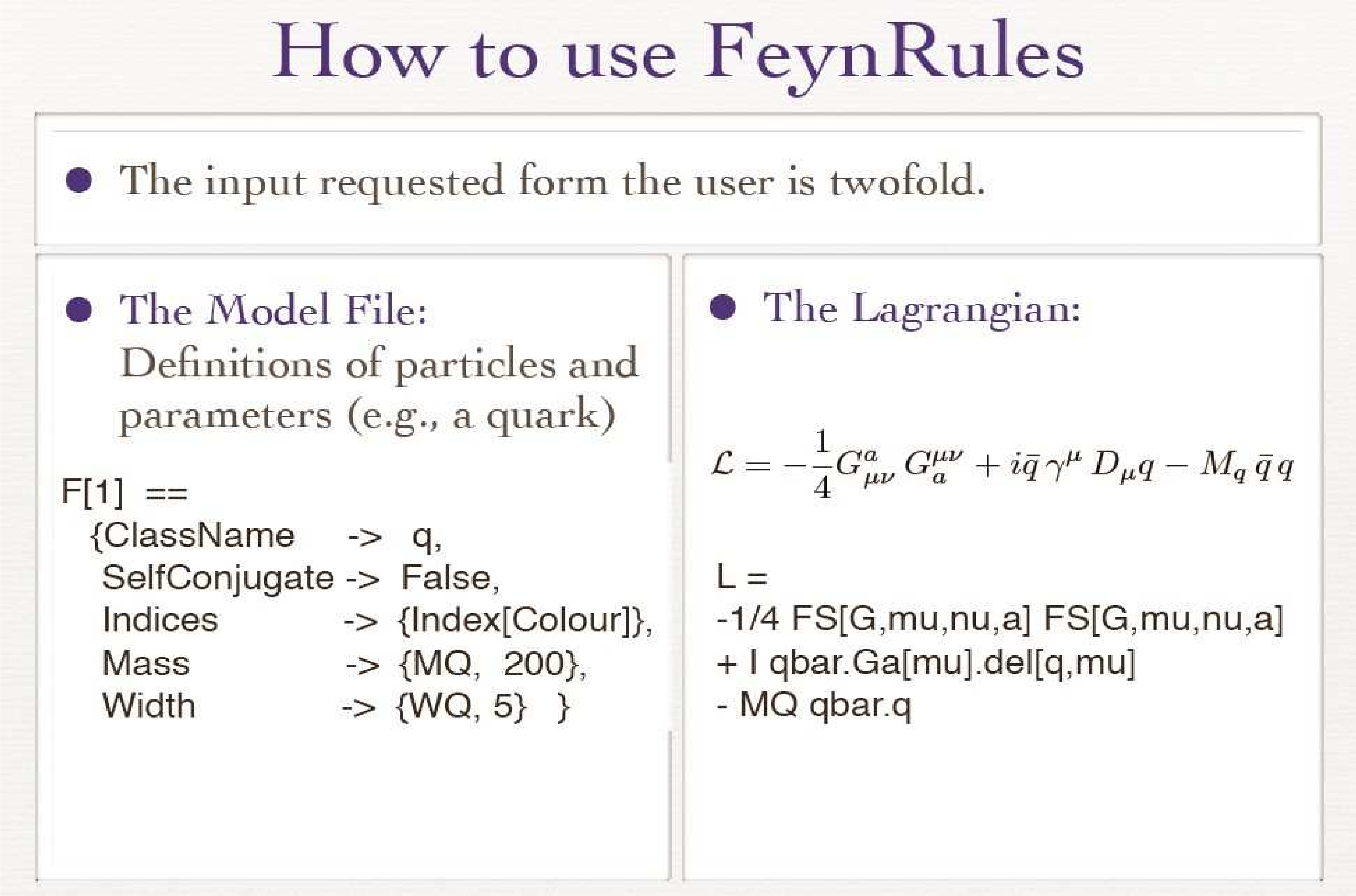
	Lanhep CalcHEP Feynman	Lanhep CalcHEP Unitary	FeynRules CalcHEP Feynman	FeynRules CalcHEP Unitary	FeynRules CompHEP Feynman
$u\bar{u}$->gg	170.5	170.5	170.5	170.5	170.49
$u'\overline{u'}$->gg	0.098763	0.098763	0.098763	0.098763	0.098761
$t\bar{t}$->γZ	1.1233	1.1233	1.1233	1.1233	1.1233
$t'\bar{t}$->γZ	0.033204	0.033204	0.033204	0.033204	0.033204
$t'\overline{t'}$->$Z'Z'$	1.887	1.887	1.887	1.887	1.887
$t\bar{b}$->$ZW^{+}{}'$	1.5603	1.5603	1.5603	1.5603	1.5604
$e\bar{e}$->$e'\bar{e}$	0.093127	0.093127	0.093127	0.093127	0.093127
$e'\bar{e}$->$u'\bar{u}$	2.3603	2.3603	2.3603	2.3603	2.3603
$e\overline{\nu_e}$->$\mu'\overline{\nu_\mu{}'}$	0.0005618	0.0005618	0.0005618	0.0005618	0.00056181
$e'\overline{\nu_e{}'}$->$d'\overline{u'}$	2.5761	2.5761	2.5761	2.5761	2.5762
gg->gg	114310.	114310.	114310.	114310.	114310.
ZZ->$Z'Z'$	0	0	0	0	0
$W^{+}W^{-}{}'$->γZ	8.329	8.329	8.329	8.329	8.3288

Feynrules flow, note the need for 2 input files



Strong feature: could output to many MEG,.... in principle (higher order operators? specific Lorentz structures?,...)

Feynrules input



How to use FeynRules

- The input requested form the user is twofold.

- The Model File:
  Definitions of particles and parameters (e.g., a quark)

```
F[1]  ==
  {ClassName      ->   q,
   SelfConjugate ->  False,
   Indices          -> {Index[Colour]},
   Mass              -> {MQ,  200},
   Width            -> {WQ, 5}   }
```

- The Lagrangian:

$$\mathcal{L} = -\frac{1}{4} G^{a}_{\mu\nu}\, G^{\mu\nu}_{a} + i\bar{q}\,\gamma^{\mu}\, D_{\mu} q - M_q\, \bar{q}\, q$$

```
L =
-1/4 FS[G,mu,nu,a] FS[G,mu,nu,a]
+ I qbar.Ga[mu].del[q,mu]
- MQ qbar.q
```

How to use FeynRules

- Once this information has been provided, FeynRules can be used to compute the Feynman rules for the model:

  FeynmanRules[L]

- Equivalently, we can export the Feynman rules to a matrix element generator, e.g., for MadGraph 4,

  WriteMGOutput[L]

Feynrules and ALOHA project, in planning

- MEG based on squaring techniques (CalcHEP/CompHEP and FeynArts/FomCalc with some(!) tweaking) are not restricted to particular Lorentz structures: dim-4, gauge structures,...

Feynrules and ALOHA project, in planning

- MEG based on squaring techniques (CalcHEP/CompHEP and FeynArts/FomCalc with some(!) tweaking) are not restricted to particular Lorentz structures: dim-4, gauge structures,...
- MEG based on helicity amplitudes are based on HELAS/Grace subroutines so a vertex must be recognised first so a general output of LanHEP or FeynArts will be of no use

Feynrules and ALOHA project, in planning

- MEG based on squaring techniques (CalcHEP/CompHEP and FeynArts/FomCalc with some(!) tweaking) are not restricted to particular Lorentz structures: dim-4, gauge structures,...
- MEG based on helicity amplitudes are based on HELAS/Grace subroutines so a vertex must be recognised first so a general output of LanHEP or FeynArts will be of no use
- Madgraph/FeynArts development: ALOHA (Automatic Languageindependent Output of Helicity Amplitudes) a code that allows to create HELAS routines from...

Feynrules and ALOHA project, in planning

- MEG based on squaring techniques (CalcHEP/CompHEP and FeynArts/FomCalc with some(!) tweaking) are not restricted to particular Lorentz structures: dim-4, gauge structures,...
- MEG based on helicity amplitudes are based on HELAS/Grace subroutines so a vertex must be recognised first so a general output of LanHEP or FeynArts will be of no use
- Madgraph/FeynArts development: **ALOHA** (Automatic Languageindependent Output of Helicity Amplitudes) a code that allows to create HELAS routines from...
- **UFO** Universal FeynRules Output

  Idea: Create Python modules that can be linked to other codes and contain all the information on a given model.

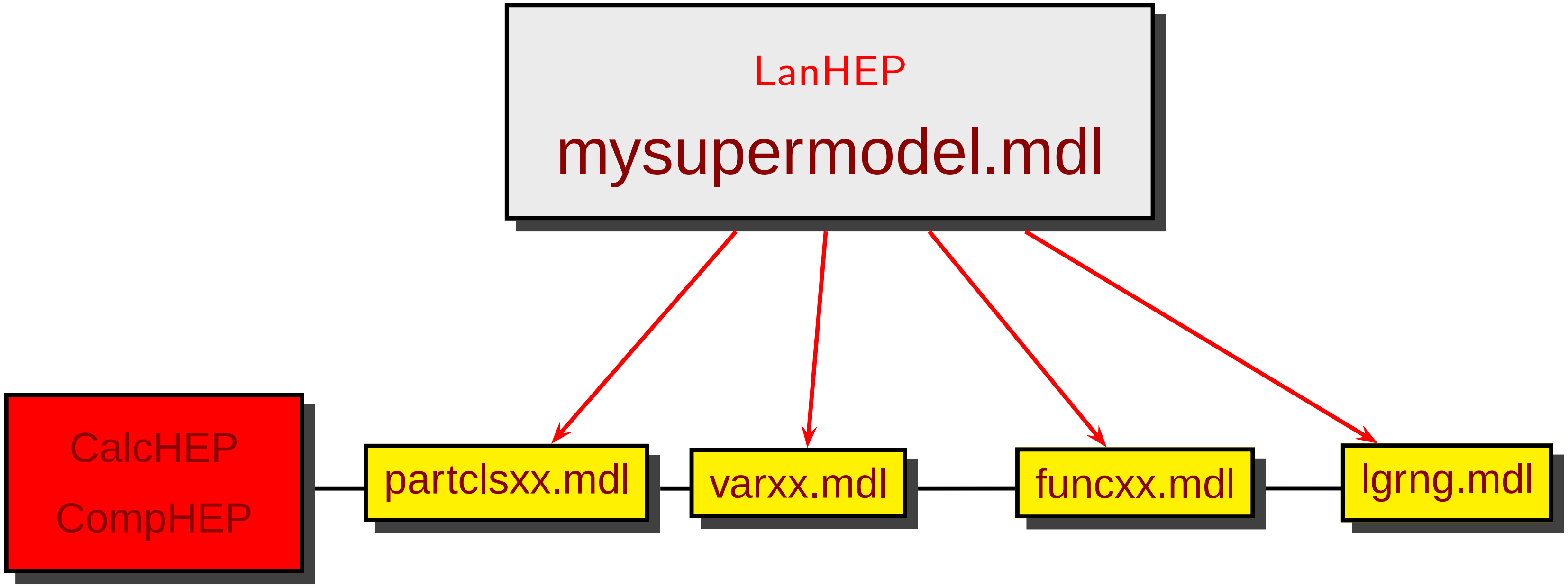
  The UFO is a self-contained Python code, and not tied to a specific matrix element generator.

LanHEP to MEG

LanHEP

mysupermodel.mdl

LanHEP to MEG



LanHEP

mysupermodel.mdl

CalcHEP

CompHEP

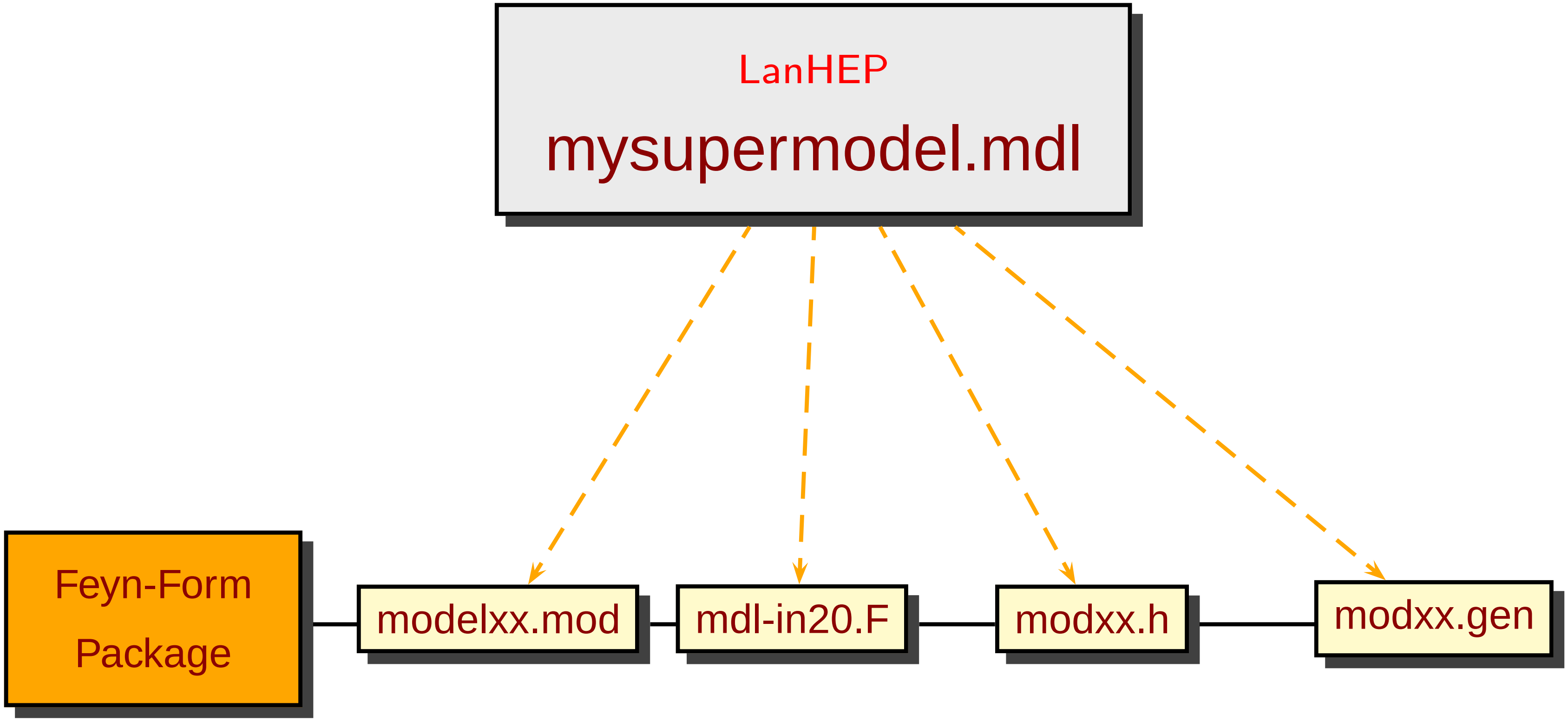
partclsxx.mdl

varxx.mdl

funcxx.mdl

lgrng.mdl

LanHEP to MEG



LanHEP

mysupermodel.mdl

Feyn-Form

Package

modelxx.mod

mdl-in20.F

modxx.h

modxx.gen

QED simple

$$\mathcal{L}_{QED} = -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} + \bar{e}\gamma^{\mu}(i\partial_{\mu} + g_e A_{\mu})e - m\bar{e}e, \quad \mathcal{L}_{GF} = -\frac{1}{2}(\partial_{\mu}A^{\mu})^2.$$

QED simple

$$\mathcal{L}_{QED} = -\frac{1}{4} F_{\mu\nu} F^{\mu\nu} + \bar{e}\gamma^{\mu}(i\partial_{\mu} + g_e A_{\mu})e - m\bar{e}e, \quad \mathcal{L}_{GF} = -\frac{1}{2}(\partial_{\mu} A^{\mu})^2.$$

```
model QED/1.
parameter ee=0.31333:'elementary electric charge'.
spinor e1/E1:(electron, mass me=0.000511).
vector A/A:(photon).
let F^mu^nu=deriv^mu*A^nu-deriv^nu*A^mu.
lterm -1/4*(F^mu^nu)**2 - 1/2*(deriv^mu*A^mu)**2.
lterm E1*(i*gamma*deriv+me)*e1.
lterm ee*E1*gamma*A*e1.
```

QED simple

$$\mathcal{L}_{QED} = -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} + \bar{e}\gamma^{\mu}(i\partial_{\mu} + g_e A_{\mu})e - m\bar{e}e, \quad \mathcal{L}_{GF} = -\frac{1}{2}(\partial_{\mu}A^{\mu})^2.$$

```
model QED/1.
parameter ee=0.31333:’elementary electric charge’.
spinor e1/E1:(electron, mass me=0.000511).
vector A/A:(photon).
let F^mu^nu=deriv^mu*A^nu-deriv^nu*A^mu.
lterm -1/4*(F^mu^nu)**2 - 1/2*(deriv^mu*A^mu)**2.
lterm E1*(i*gamma*deriv+me)*e1.
lterm ee*E1*gamma*A*e1.
```

```
lterm ee*E1^a*gamma^a^b^mu*A^mu*e1^b.
```

QCD on paper

$$L_{YM} = -\frac{1}{4} F^{a\mu\nu} F^{a}_{\mu\nu},$$

where

$$F^{a}_{\mu\nu} = \partial_\mu G^a_\nu - \partial_\nu G^a_\mu - g_s f^{abc} G^b_\mu G^c_\nu,$$

$$L_F = \bar{q}_i \gamma^\mu \partial_\mu q_i + g_s \lambda^a_{ij} \bar{q}_i \gamma^\mu q_j G^c_\mu,$$

where λ^a_{ij} are Gell-Mann matrices.

$$L_{GF+Gh} = -\frac{1}{2} (\partial_\mu G^\mu_a)^2 + i g_s f^{abc} \bar{c}^a G^b_\mu \partial^\mu c^c,$$

$(c, \bar{c})$ ghost fields.

QCD in LanHEP

```
model QCD/2.
parameter gg=1.117:'Strong coupling'.
spinor q/Q:(quark, mass mq=0.01, color c3).
vector G/G:(gluon, color c8, gauge).
let F^mu^nu^a = deriv^nu*G^mu^a - deriv^mu*G^nu^a -
    gg*f_SU3^a^b^c*G^mu^b*G^nu^c.
lterm -F**2/4-(deriv*G)**2/2.
lterm Q*(i*gamma*deriv+mq)*q.
lterm i*gg*f_SU3*ccghost(G)*G*deriv*ghost(G).
lterm gg*Q*gamma*lambda*G*q.
```

LanHEP Input file: qcd.mdl

```
model QCD/2.

parameter       gg= 1.13 : 'Strong coupling'.

vector  G/G: (gluon, color c8, gauge). spinor  q:(quark, color c3,
mass Mq=0.02). lterm  i*gg*f_SU3*ccghost(G)*G*deriv*ghost(G).
lterm Q*gamma*(i*deriv + gg*lambda*G)*q. lterm -F**2/4  where
        F=deriv^mu*G^nu^a-deriv^nu*G^mu^a+i*gg*f_SU3^a^b^c*G^mu^b*G^nu^c
```

QCD, Feynman rules from LanHEP, compHEP/CalcHEP format

Fields in the vertex	Variational derivative of Lagrangian by fields
$G_{\mu p}$ $\bar{\eta}^G_q$ $\eta^G{}_r$	$-g_s p_3^\mu f_{pqr}$
$\bar{q}_{ap}$ q_{bq} $G_{\mu r}$	$g_s \gamma^\mu_{ab} \lambda^r_{pq}$
$G_{\mu p}$ $G_{\nu q}$ $G_{\rho r}$	$g_s f_{pqr} (p_3^\nu g^{\mu\rho} - p_2^\rho g^{\mu\nu} - p_3^\mu g^{\nu\rho} + p_1^\rho g^{\mu\nu} + p_2^\mu g^{\nu\rho} - p_1^\nu g^{\mu\rho})$
$G_{\mu p}$ $G_{\nu q}$ $G_{\rho r}$ $G_{\sigma s}$	$g_s^2 (g^{\mu\rho} g^{\nu\sigma} f_{pqt} f_{rst} - g^{\mu\sigma} g^{\nu\rho} f_{pqt} f_{rst} + g^{\mu\nu} g^{\rho\sigma} f_{prt} f_{qst}$ $+ g^{\mu\nu} g^{\rho\sigma} f_{pst} f_{qrt} - g^{\mu\sigma} g^{\nu\rho} f_{prt} f_{qst} - g^{\mu\rho} g^{\nu\sigma} f_{pst} f_{qrt})$

Let’s play: LanHEP Input file, qedscal.mdl

We introduce a complex/charged scalar field ϕ

```
model qedscal/20.
parameter ee = 0.3133: ’Electric charge’.
vector A/A:photon.
let F^mu^nu=deriv^mu*A^nu-deriv^nu*A^mu.
spinor e1:(electron, mass me=0.000511).
scalar phi/PHI:(scalar, mass mphi=100).
lterm ee*E1*gamma*A*e1.
let Dphi^mu = (deriv^mu+i*ee*A^mu)*phi.
let DPHI^mu = (deriv^mu-i*ee*A^mu)*PHI.
lterm DPHI*Dphi.
```

LanHEP output CalcHEP/(CompHEP): partclsxx.mdl

```
qedscal
 Particles
 Full  name  | P | aP| number |2*spin| mass |width |color|aux|>  LaTeX(A
photon       |A  |A  |      22|2     |0     |0     |1    |   |A
electron     |e1 |E1 |      11|1     |me    |0     |1    |   |e1
scalar       |phi|PHI|       0|0     |mphi  |0     |1    |   |phi
```

LanHEP output CalcHEP/CompHEP, variables: varxx.mdl

```
qedscal
 Variables
 Name | Value       |>  Comment                                              <|
ee    |0.3133       |Electric charge
me    |0.000511     |mass of electron
mphi  |100          |mass of scalar
```

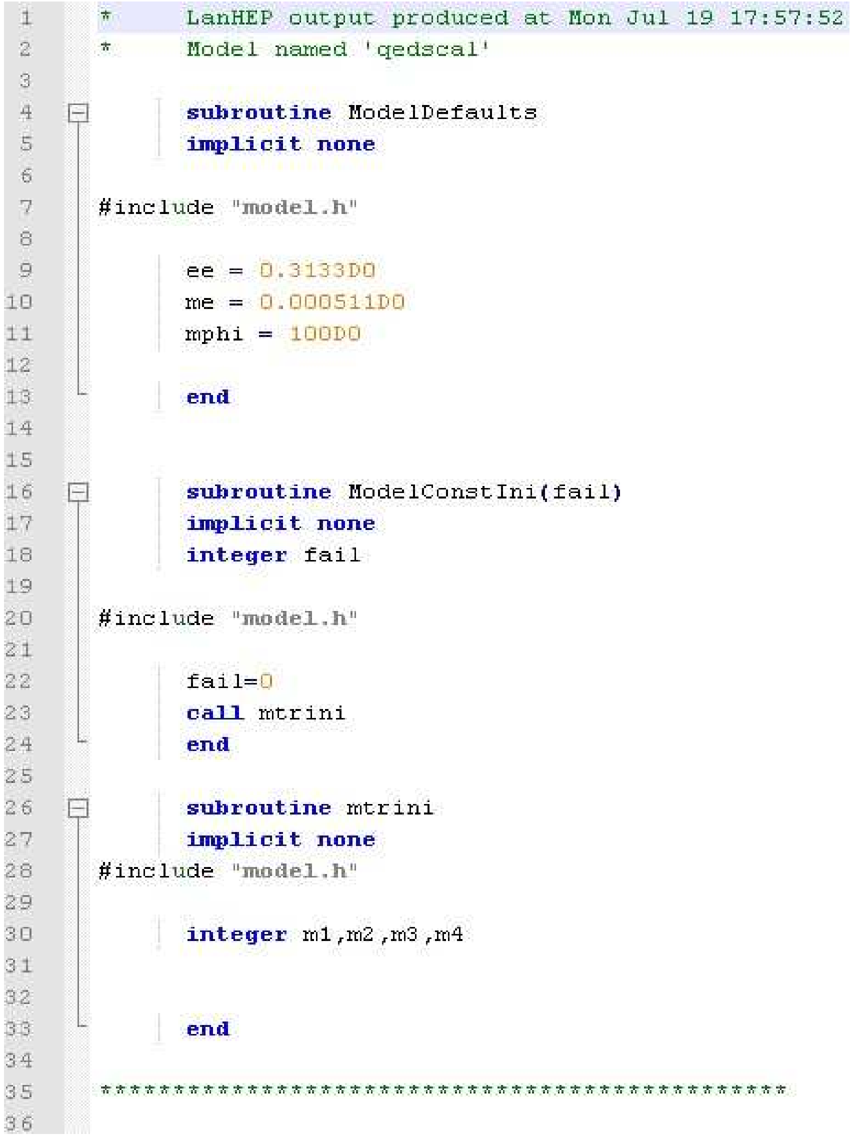
LanHEP output CalcHEP/CompHEP, constraints: funcxx.mdl

```
qedscal
 Constraints
  Name |> Expression     <&> Comment           <|
```

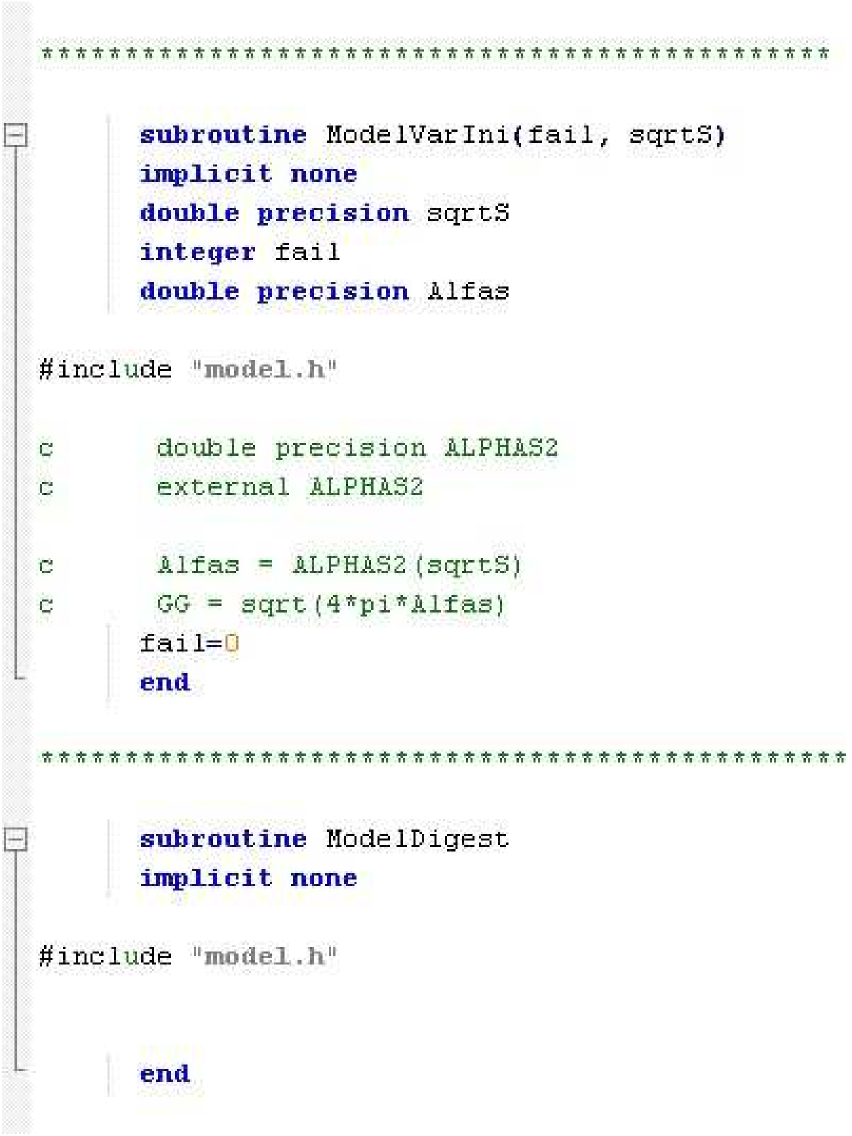
LanHEP output CalcHEP/CompHEP: Feynman rules lgrngxx.mdl

```
qedscal
 Lagrangian
P1    |P2    |P3    |P4    |>   Factor  <|> dLagrangian/ dA(p1) dA(p2) dA(p3
A     |PHI   |phi   |      |ee              |m1.p2-m1.p3
E1    |e1    |A     |      |ee              |G(m3)
A     |A     |PHI   |phi   |2*ee^2          |m1.m2
```

LanHEP output Feyn package: mdl ini20.F



```
*       LanHEP output produced at Mon Jul 19 17:57:52
*       Model named 'qedscal'

      subroutine ModelDefaults
      implicit none

#include "model.h"

      ee = 0.3133D0
      me = 0.000511D0
      mphi = 100D0

      end


      subroutine ModelConstIni(fail)
      implicit none
      integer fail

#include "model.h"

      fail=0
      call mtrini
      end

      subroutine mtrini
      implicit none
#include "model.h"

      integer m1,m2,m3,m4


      end

***********************************************

```

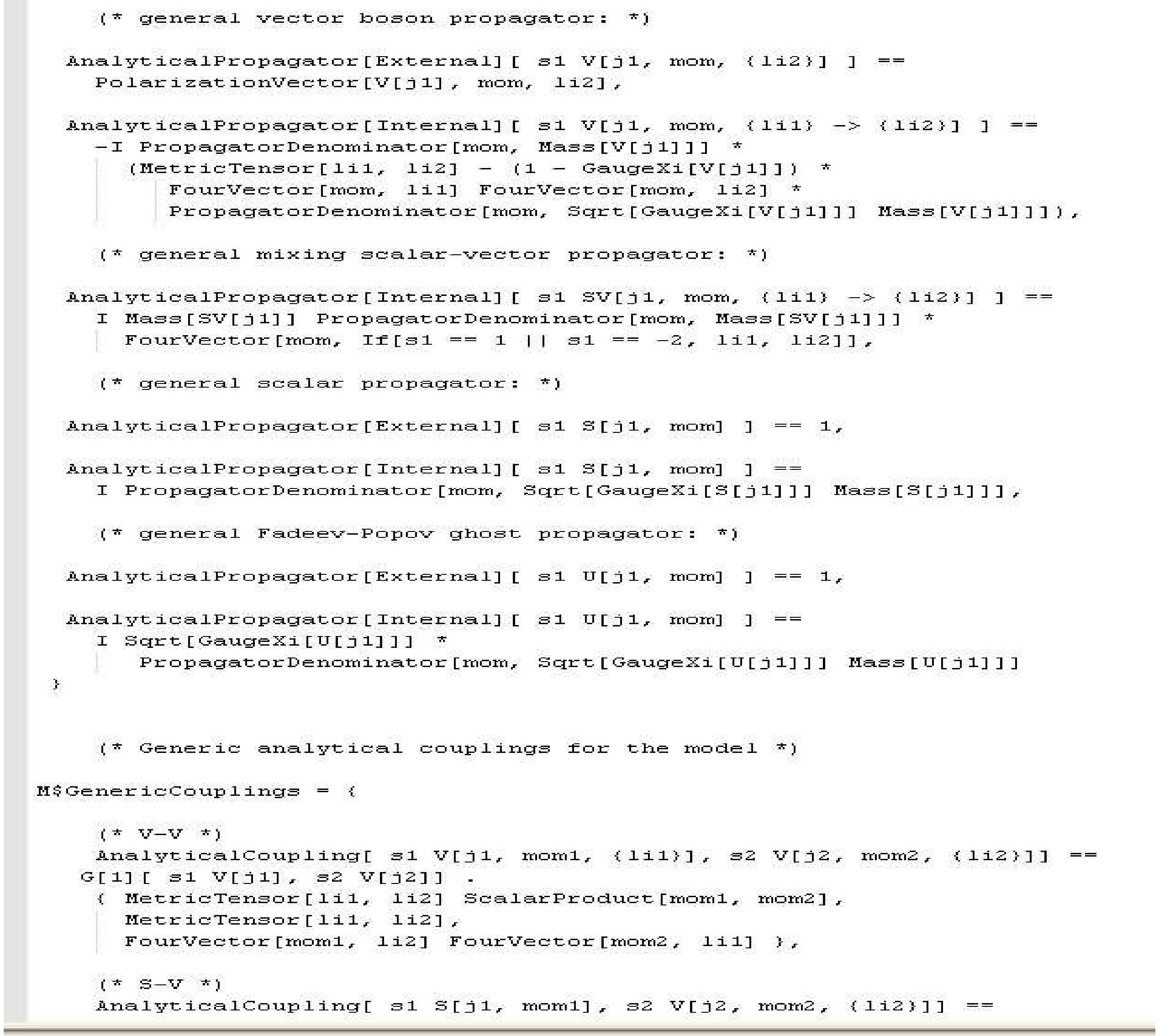


```
********************************************


      subroutine ModelVarIni(fail, sqrtS)
      implicit none
      double precision sqrtS
      integer fail
      double precision Alfas

#include "model.h"

c       double precision ALPHAS2
c       external ALPHAS2

c       Alfas = ALPHAS2(sqrtS)
c       GG = sqrt(4*pi*Alfas)
      fail=0
      end

********************************************

      subroutine ModelDigest
      implicit none

#include "model.h"


      end
```

LanHEP output Feyn package: mdl ini20.F

```
*      LanHEP output produced at Mon Jul 19 17:57:52 2010
*      Model named 'qedscal'

       subroutine ModelDefaults
       implicit none

#include "model.h"

       ee = 0.3133D0
       me = 0.000511D0
       mphi = 100D0

       end


       subroutine ModelConstIni(fail)
       implicit none
       integer fail

#include "model.h"
```

LanHEP output Feyn package: Generic file: modelxx.gen

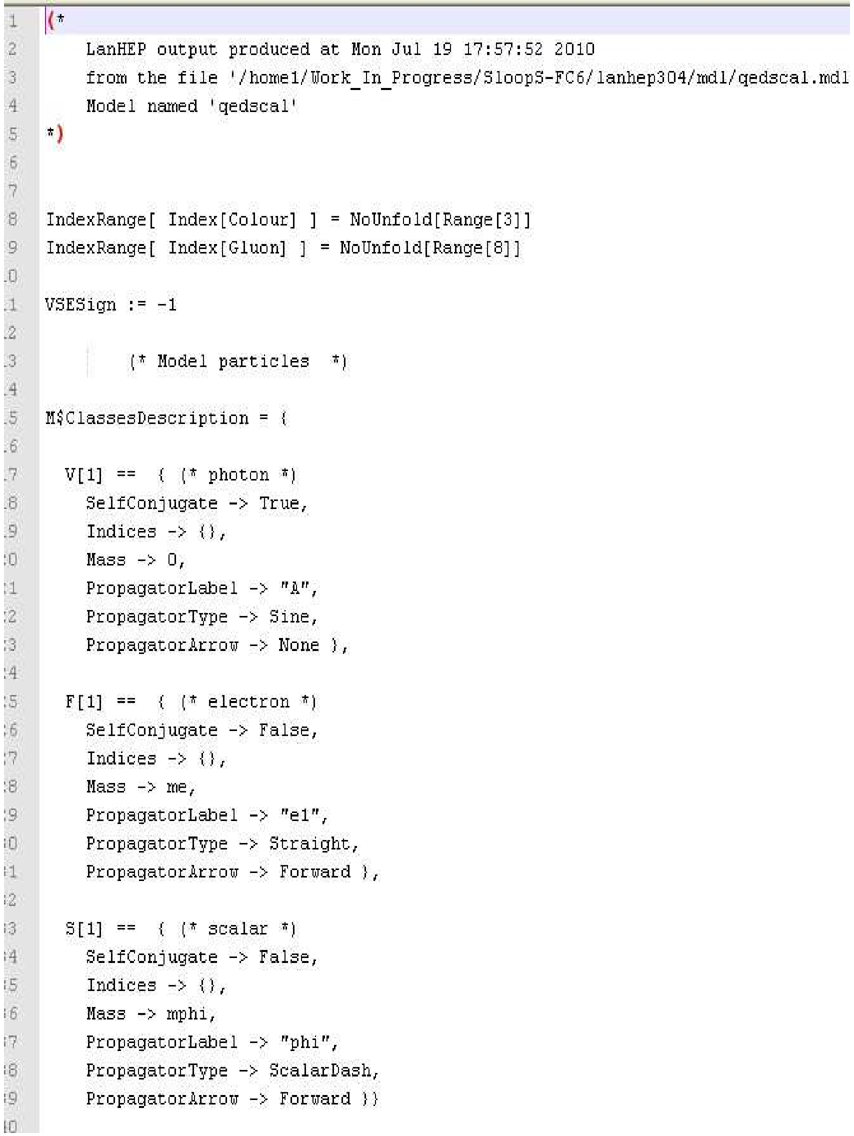


```
    (* general vector boson propagator: *)

  AnalyticalPropagator[External][ s1 V[j1, mom, {li2}] ] ==
    PolarizationVector[V[j1], mom, li2],

  AnalyticalPropagator[Internal][ s1 V[j1, mom, {li1} -> {li2}] ] ==
    -I PropagatorDenominator[mom, Mass[V[j1]]] *
      (MetricTensor[li1, li2] - (1 - GaugeXi[V[j1]]) *
         FourVector[mom, li1] FourVector[mom, li2] *
         PropagatorDenominator[mom, Sqrt[GaugeXi[V[j1]]] Mass[V[j1]]]),

    (* general mixing scalar-vector propagator: *)

  AnalyticalPropagator[Internal][ s1 SV[j1, mom, {li1} -> {li2}] ] ==
    I Mass[SV[j1]] PropagatorDenominator[mom, Mass[SV[j1]]] *
      FourVector[mom, If[s1 == 1 || s1 == -2, li1, li2]],

    (* general scalar propagator: *)

  AnalyticalPropagator[External][ s1 S[j1, mom] ] == 1,

  AnalyticalPropagator[Internal][ s1 S[j1, mom] ] ==
    I PropagatorDenominator[mom, Sqrt[GaugeXi[S[j1]]] Mass[S[j1]]],

    (* general Fadeev-Popov ghost propagator: *)

  AnalyticalPropagator[External][ s1 U[j1, mom] ] == 1,

  AnalyticalPropagator[Internal][ s1 U[j1, mom] ] ==
    I Sqrt[GaugeXi[U[j1]]] *
       PropagatorDenominator[mom, Sqrt[GaugeXi[U[j1]]] Mass[U[j1]]]
 }


    (* Generic analytical couplings for the model *)

M$GenericCouplings = {

    (* V-V *)
    AnalyticalCoupling[ s1 V[j1, mom1, {li1}], s2 V[j2, mom2, {li2}]] ==
   G[1][ s1 V[j1], s2 V[j2]] .
    { MetricTensor[li1, li2] ScalarProduct[mom1, mom2],
      MetricTensor[li1, li2],
      FourVector[mom1, li2] FourVector[mom2, li1] },

    (* S-V *)
    AnalyticalCoupling[ s1 S[j1, mom1], s2 V[j2, mom2, {li2}]] ==
```

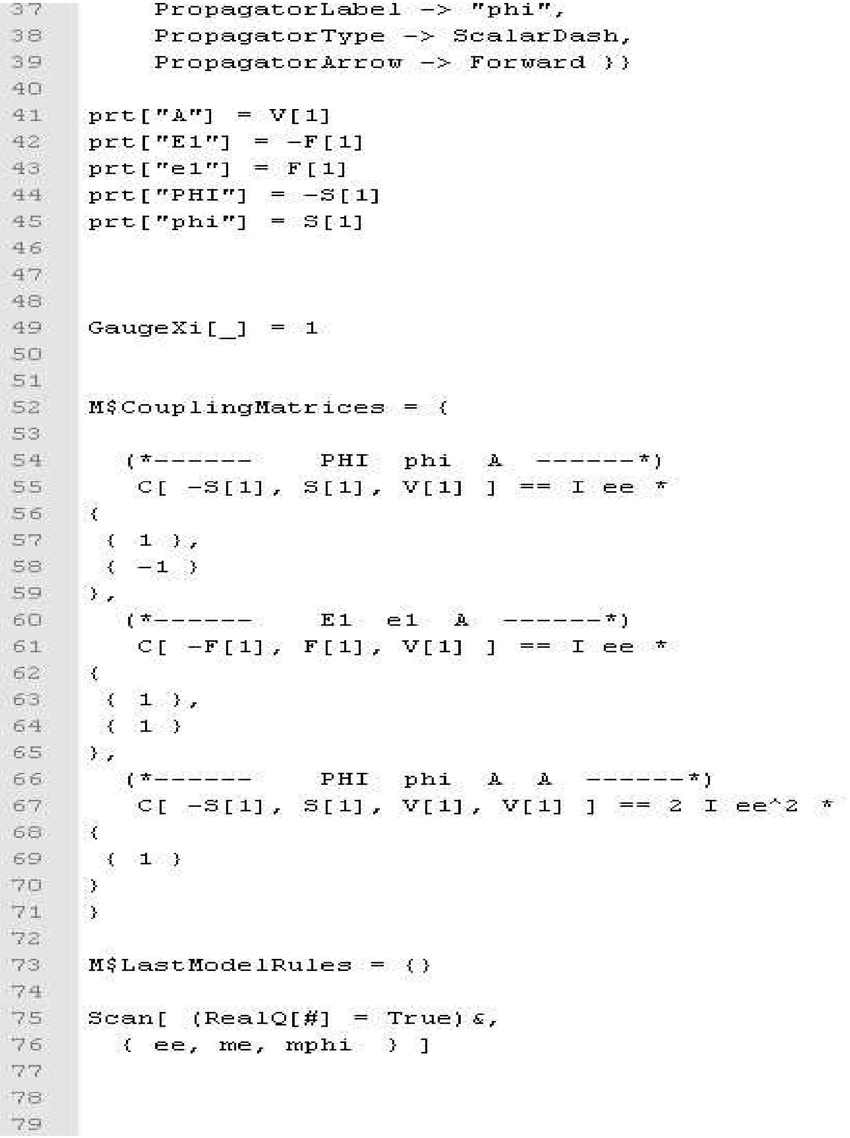
LanHEP output Feyn Generic file: modelxx.gen

```
M$GenericPropagators = {

    (* general fermion propagator: *)

  AnalyticalPropagator[External][ s1 F[j1, mom] ] ==
    NonCommutative[ SpinorType[j1][-mom, Mass[F[j1]]] ],

    (* Remarks:
       Fermionic propagators have (like all others, too) their
       momentum flowing from left to right. The fermion flow (for
       Dirac fermions: fermion number flow) is from right to left.
       If the fermion inside the propagator has no sign (i.e. fermion
       number flow is opposite to fermion flow or fermion is self
       conjugate) we just use the internal propagator S(-p).
       If the fermion has a sign, we have to use the Feynman rule S(p)
       according to the Majorana paper. However, this rule is given
       for a momentum flowing against the fermion flow so, again, we
       end up with S(-p). *)

  AnalyticalPropagator[Internal][ s1 F[j1, mom] ] ==
```

LanHEP output Feyn package: modelxx.mod



```
(*
    LanHEP output produced at Mon Jul 19 17:57:52 2010
    from the file '/home1/Work_In_Progress/SloopS-FC6/lanhep304/mdl/qedscal.mdl
    Model named 'qedscal'
*)


IndexRange[ Index[Colour] ] = NoUnfold[Range[3]]
IndexRange[ Index[Gluon] ] = NoUnfold[Range[8]]

VSESign := -1

        (* Model particles  *)

M$ClassesDescription = {

  V[1] ==  { (* photon *)
    SelfConjugate -> True,
    Indices -> {},
    Mass -> 0,
    PropagatorLabel -> "A",
    PropagatorType -> Sine,
    PropagatorArrow -> None },

  F[1] ==  { (* electron *)
    SelfConjugate -> False,
    Indices -> {},
    Mass -> me,
    PropagatorLabel -> "e1",
    PropagatorType -> Straight,
    PropagatorArrow -> Forward },

  S[1] ==  { (* scalar *)
    SelfConjugate -> False,
    Indices -> {},
    Mass -> mphi,
    PropagatorLabel -> "phi",
    PropagatorType -> ScalarDash,
    PropagatorArrow -> Forward }}

```



```
     PropagatorLabel -> "phi",
     PropagatorType -> ScalarDash,
     PropagatorArrow -> Forward }}

prt["A"]  = V[1]
prt["E1"]  = -F[1]
prt["e1"]  = F[1]
prt["PHI"]  = -S[1]
prt["phi"]  = S[1]



GaugeXi[_]  = 1


M$CouplingMatrices  = {

   (*------   PHI  phi  A  ------*)
    C[ -S[1], S[1], V[1] ] == I ee  *
{
 { 1 },
 { -1 }
},
   (*------   E1  e1  A  ------*)
    C[ -F[1], F[1], V[1] ] == I ee  *
{
 { 1 },
 { 1 }
},
   (*------   PHI  phi  A  A  ------*)
    C[ -S[1], S[1], V[1], V[1] ] == 2 I ee^2  *
{
 { 1 }
}
}

M$LastModelRules = {}

Scan[ (RealQ[#] = True)&,
  { ee, me, mphi  } ]



```

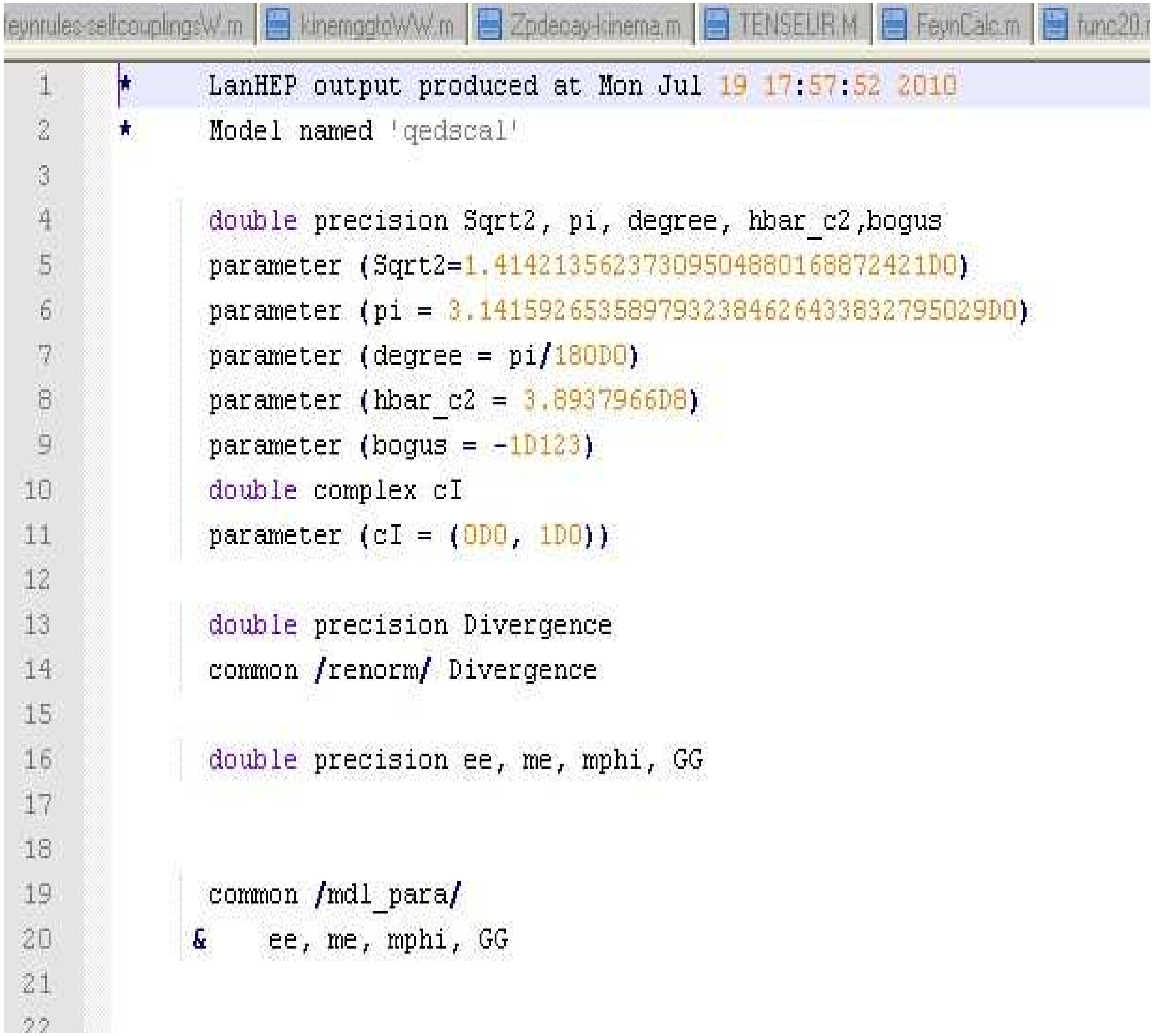
LanHEP output: modelxx.mod part 1

```
(*
    LanHEP output produced at Mon Jul 19 17:57:52 2010
    from the file '/home1/Work_In_Progress/SloopS-FC6/lanhep304/mdl/qedsc
    Model named 'qedscal'
*)


IndexRange[ Index[Colour] ] = NoUnfold[Range[3]] IndexRange[
Index[Gluon] ] = NoUnfold[Range[8]]

VSESign := -1

        (* Model particles  *)

M$ClassesDescription = {

  V[1] ==  { (* photon *)
    SelfConjugate -> True,
    Indices -> {},
    Mass -> 0,
```

LanHEP output: modelxx.mod part 2

```
GaugeXi[_] = 1


M$CouplingMatrices = {

  (*------    PHI  phi  A  ------*)
   C[ -S[1], S[1], V[1] ] == I ee *
{
 { 1 },
 { -1 }
},
  (*------    E1  e1  A  ------*)
   C[ -F[1], F[1], V[1] ] == I ee *
{
 { 1 },
 { 1 }
},
  (*------    PHI  phi  A  A  ------*)
   C[ -S[1], S[1], V[1], V[1] ] == 2 I ee^2 *
{
```

LanHEP output Feyn package: modelxx.mod

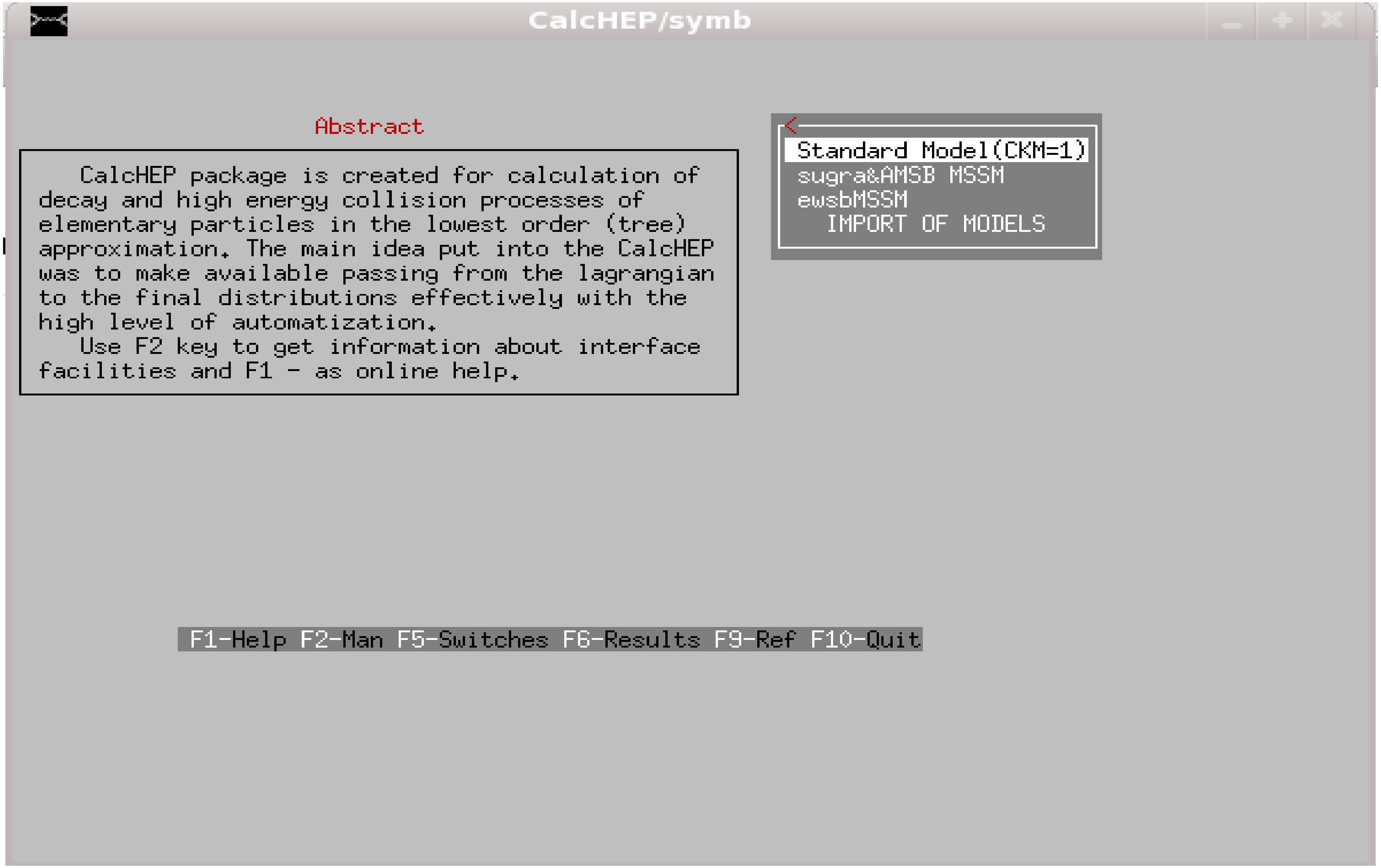


```
*      LanHEP output produced at Mon Jul 19 17:57:52 2010
*      Model named 'qedscal'

      double precision Sqrt2, pi, degree, hbar_c2,bogus
      parameter (Sqrt2=1.41421356237309504880168872421D0)
      parameter (pi = 3.1415926535897932384626433832795029D0)
      parameter (degree = pi/180D0)
      parameter (hbar_c2 = 3.8937966D8)
      parameter (bogus = -1D123)
      double complex cI
      parameter (cI = (0D0, 1D0))

      double precision Divergence
      common /renorm/ Divergence

      double precision ee, me, mphi, GG


      common /mdl_para/
     &    ee, me, mphi, GG
```

LanHEP output Feyn package: modelxx.h part 2

```
*      LanHEP output produced at Mon Jul 19 17:57:52 2010
*      Model named 'qedscal'

      double precision Sqrt2, pi, degree, hbar_c2,bogus
      parameter (Sqrt2=1.41421356237309504880168872421D0)
      parameter (pi = 3.1415926535897932384626433832795029D0)
      parameter (degree = pi/180D0)
      parameter (hbar_c2 = 3.8937966D8)
      parameter (bogus = -1D123)
      double complex cI
      parameter (cI = (0D0, 1D0))

      double precision Divergence
      common /renorm/ Divergence

      double precision ee, me, mphi, GG


      common /mdl_para/
     &    ee, me, mphi, GG
```

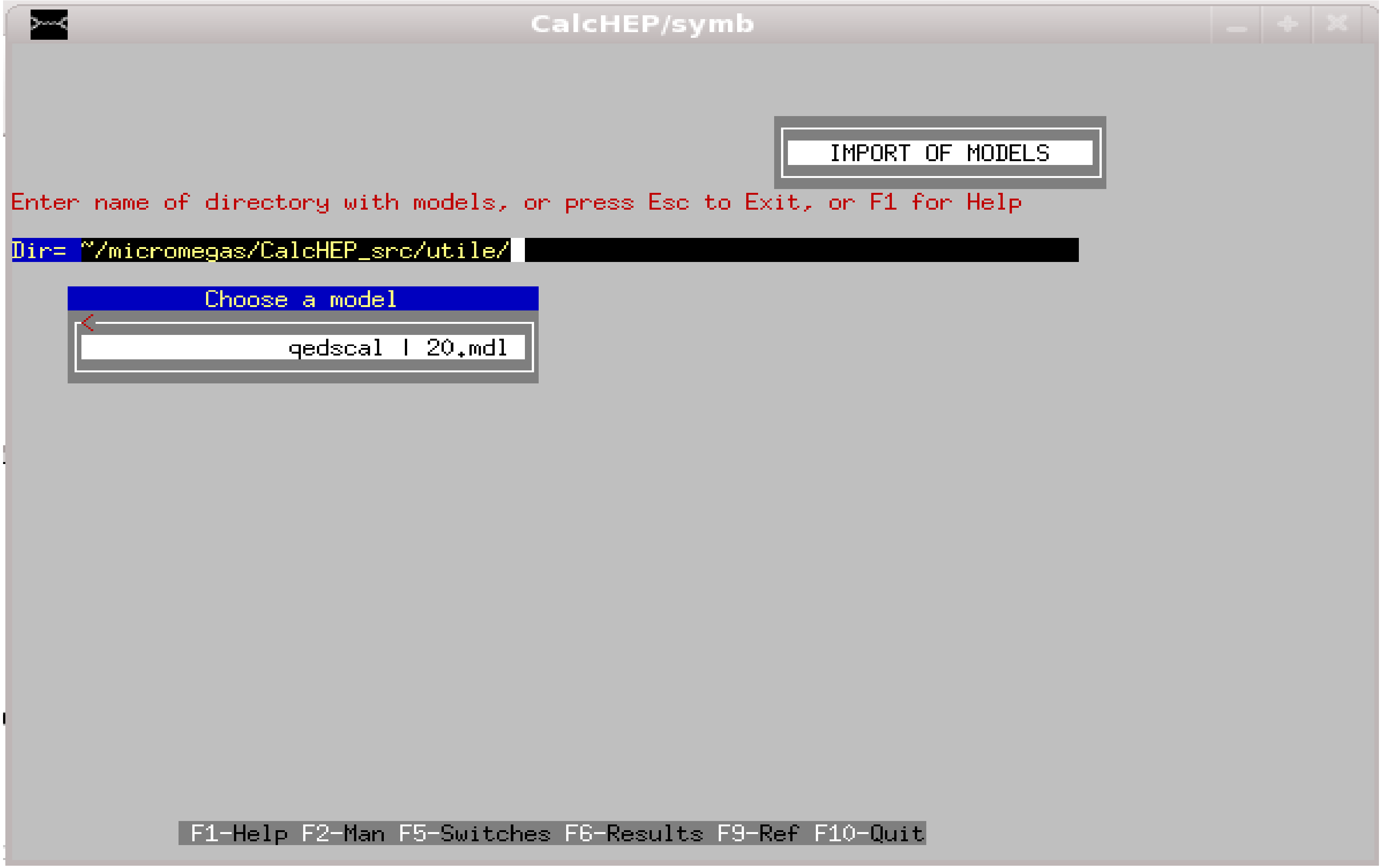
Using our newly implemented model in CalcHEP 1.



Abstract

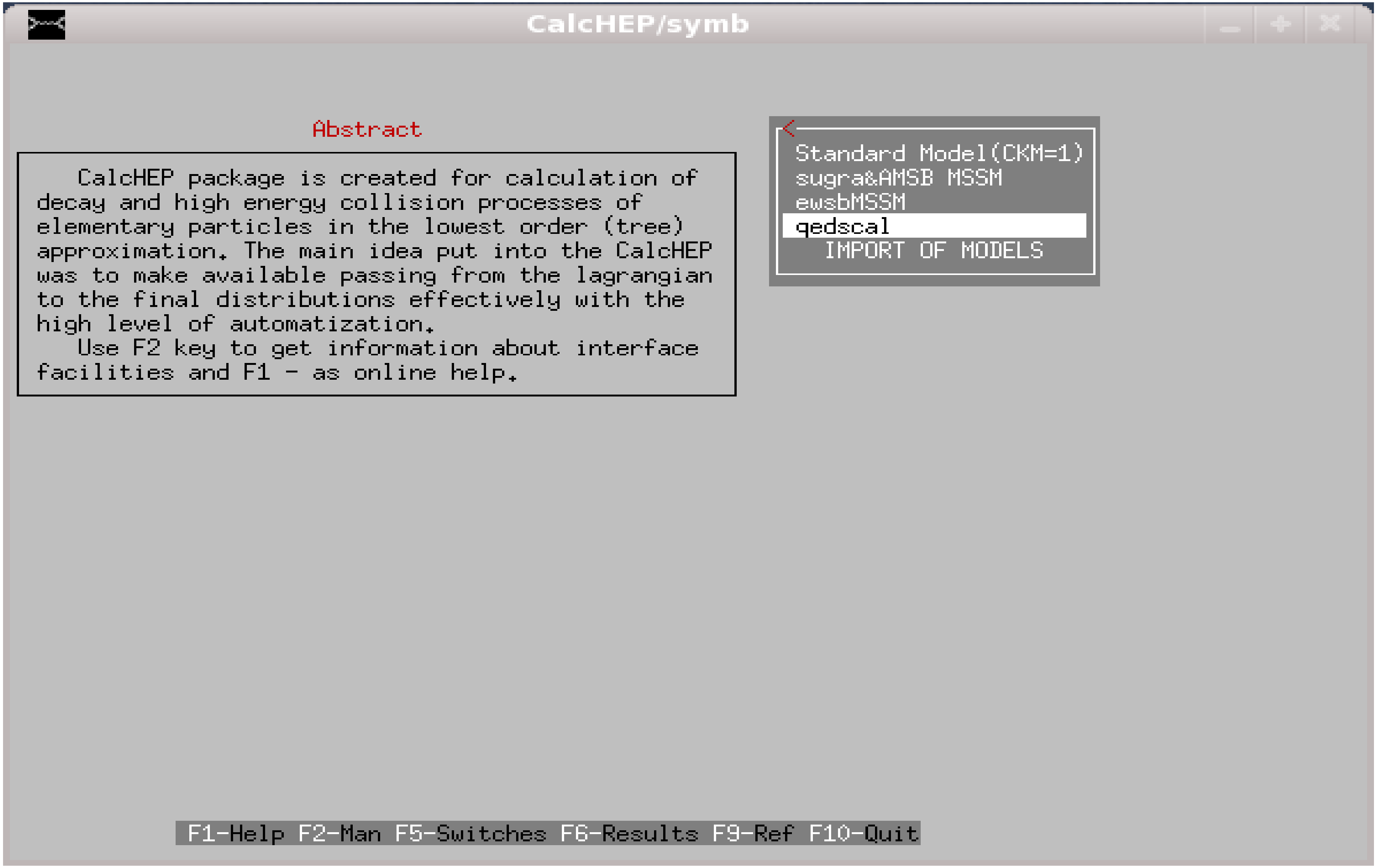
CalcHEP package is created for calculation of decay and high energy collision processes of elementary particles in the lowest order (tree) approximation. The main idea put into the CalcHEP was to make available passing from the lagrangian to the final distributions effectively with the high level of automatization.
Use F2 key to get information about interface facilities and F1 - as online help.

Using our newly implemented model in CalcHEP 2.



```
IMPORT OF MODELS

Enter name of directory with models, or press Esc to Exit, or F1 for Help

Dir= ~/micromegas/CalcHEP_src/utile/

Choose a model
       qedscal | 20.mdl

F1-Help F2-Man F5-Switches F6-Results F9-Ref F10-Quit
```

Using our newly implemented model in CalcHEP 3.

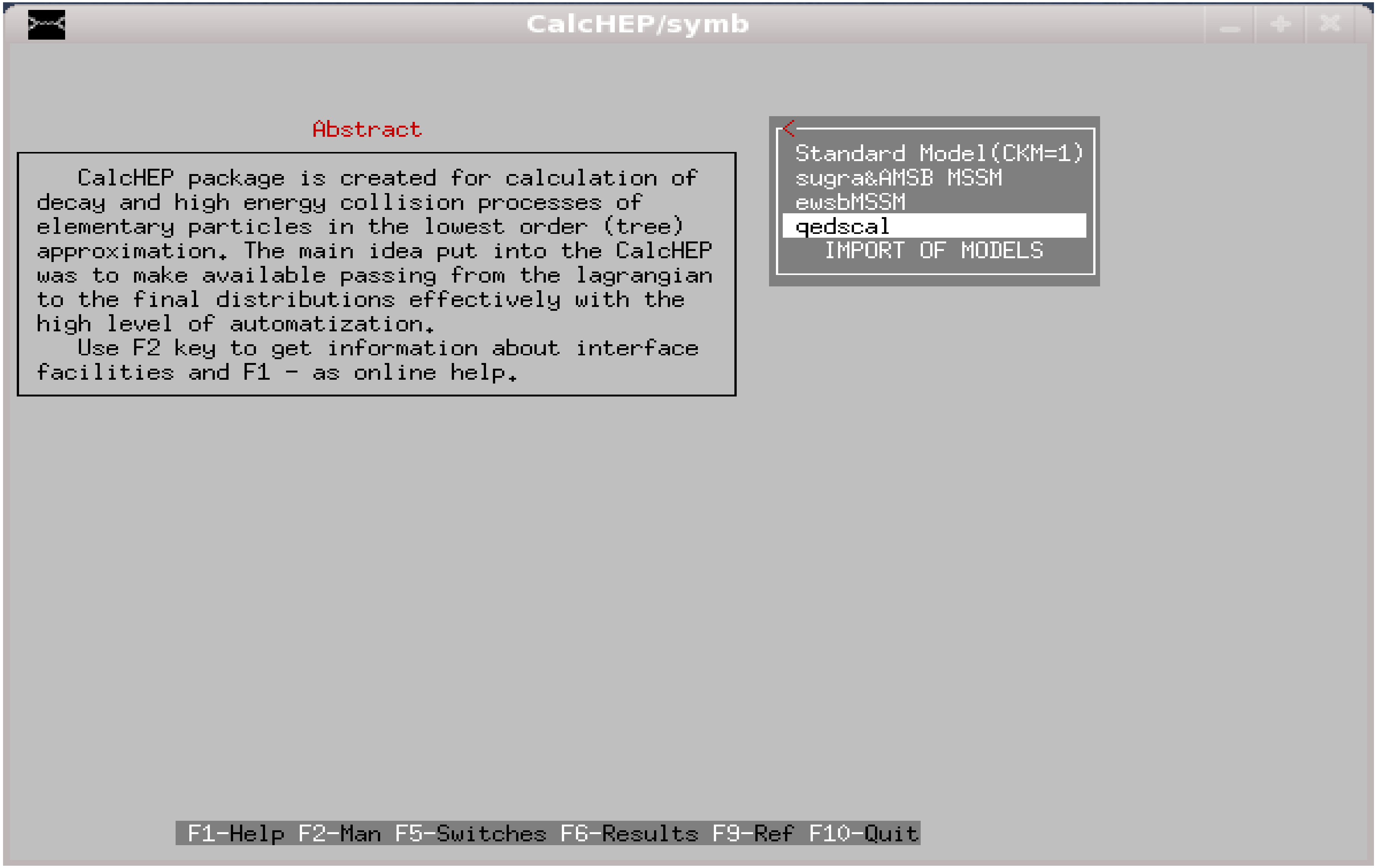


Abstract

CalcHEP package is created for calculation of decay and high energy collision processes of elementary particles in the lowest order (tree) approximation. The main idea put into the CalcHEP was to make available passing from the lagrangian to the final distributions effectively with the high level of automatization.

Use F2 key to get information about interface facilities and F1 - as online help.

Using our newly implemented model in CalcHEP 3.

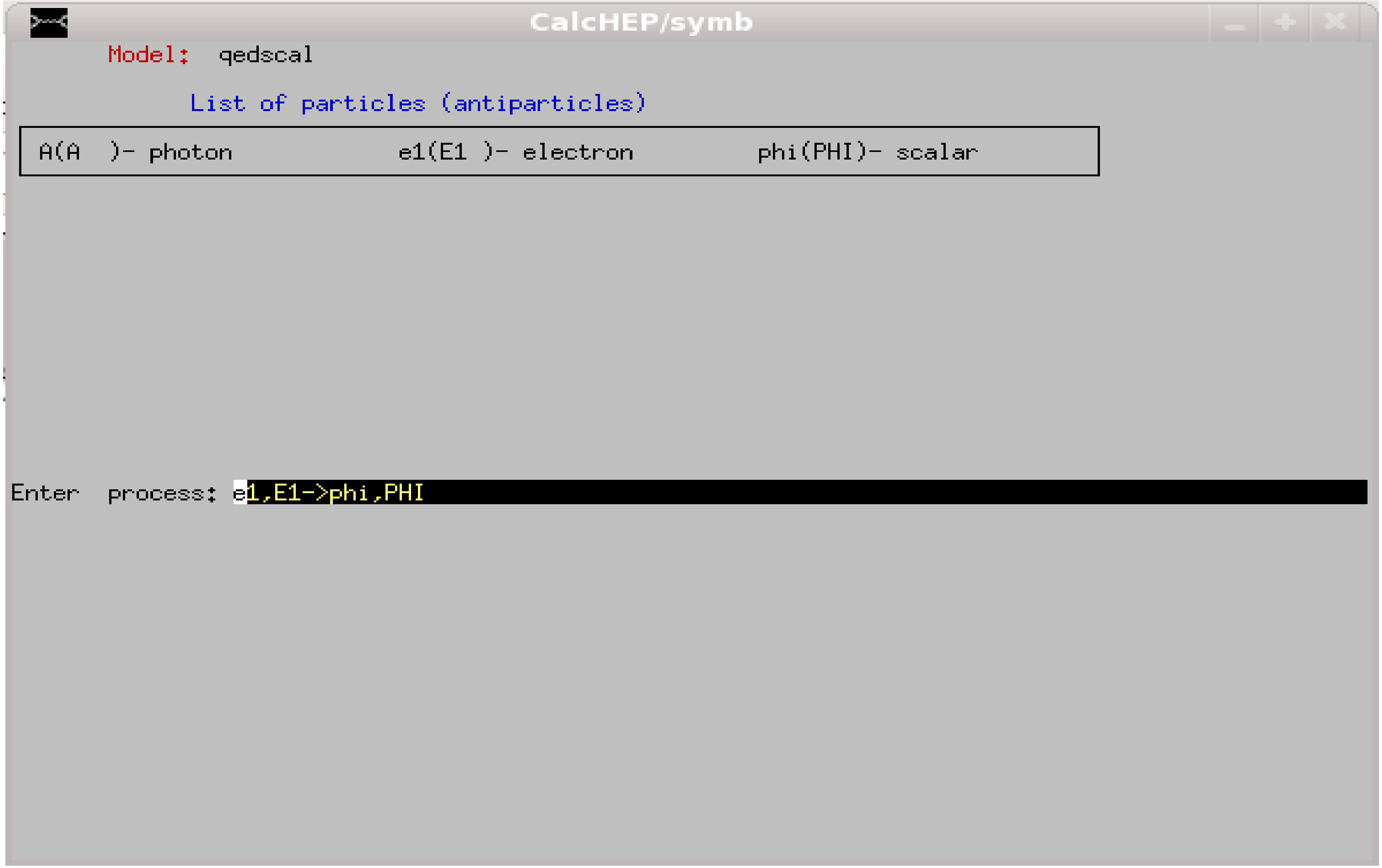


Abstract

CalcHEP package is created for calculation of decay and high energy collision processes of elementary particles in the lowest order (tree) approximation. The main idea put into the CalcHEP was to make available passing from the lagrangian to the final distributions effectively with the high level of automatization.

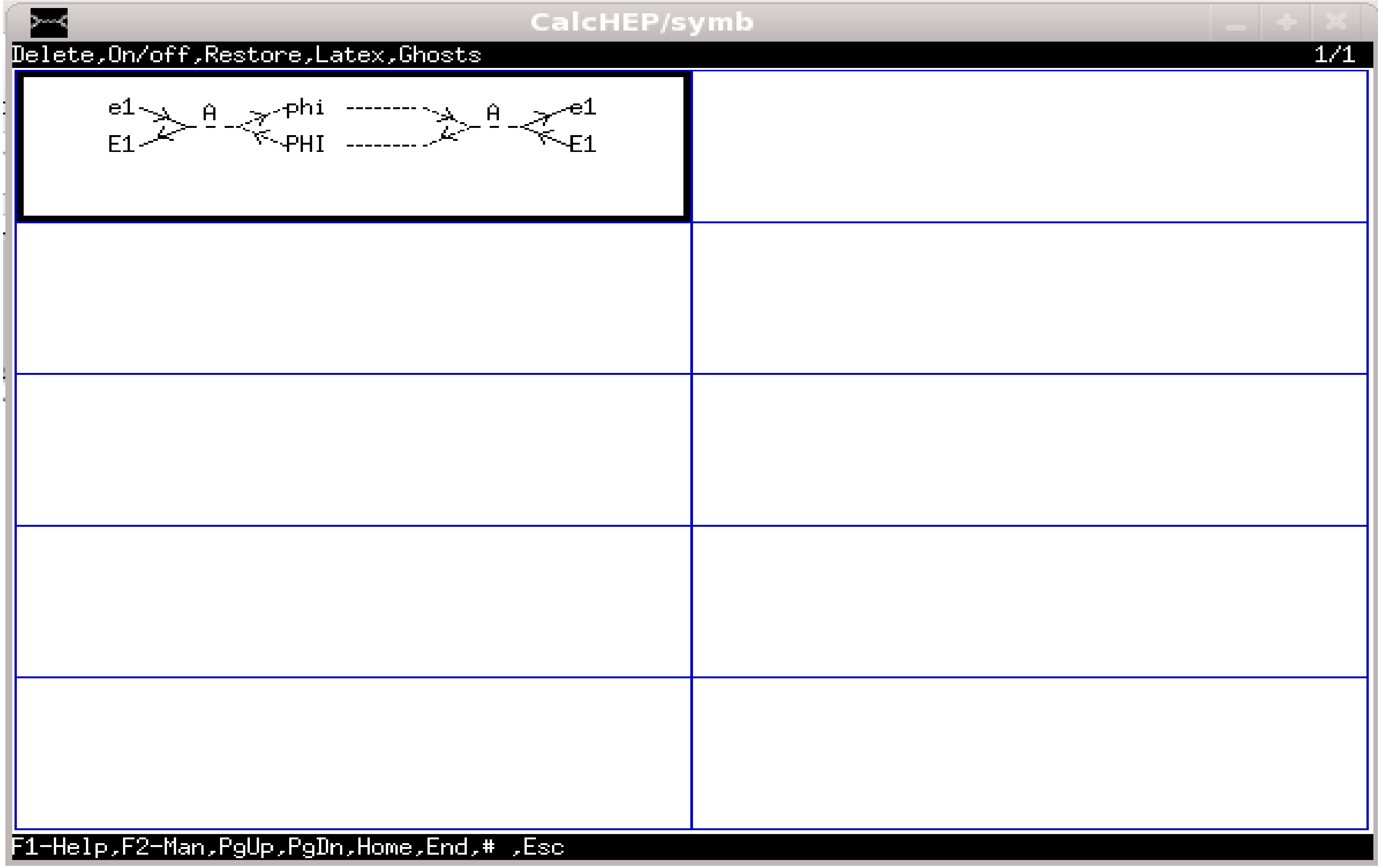
Use F2 key to get information about interface facilities and F1 - as online help.

Using our newly implemented model in CalcHEP 4.

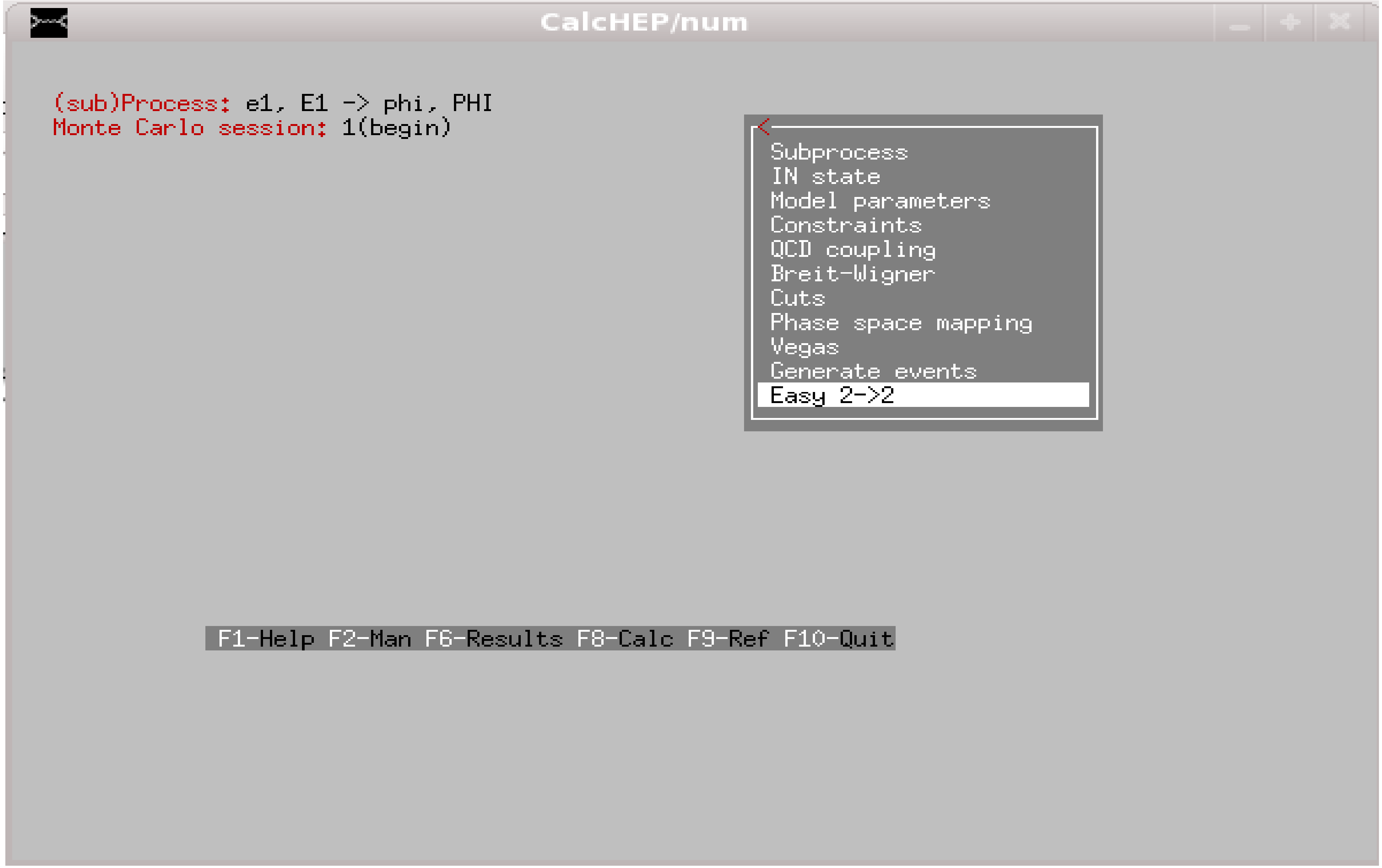


```
Model:   qedscal

        List of particles (antiparticles)
A(A  )- photon          e1(E1 )- electron         phi(PHI)- scalar

Enter  process: e1,E1->phi,PHI
```

Using our newly implemented model in CalcHEP 5.

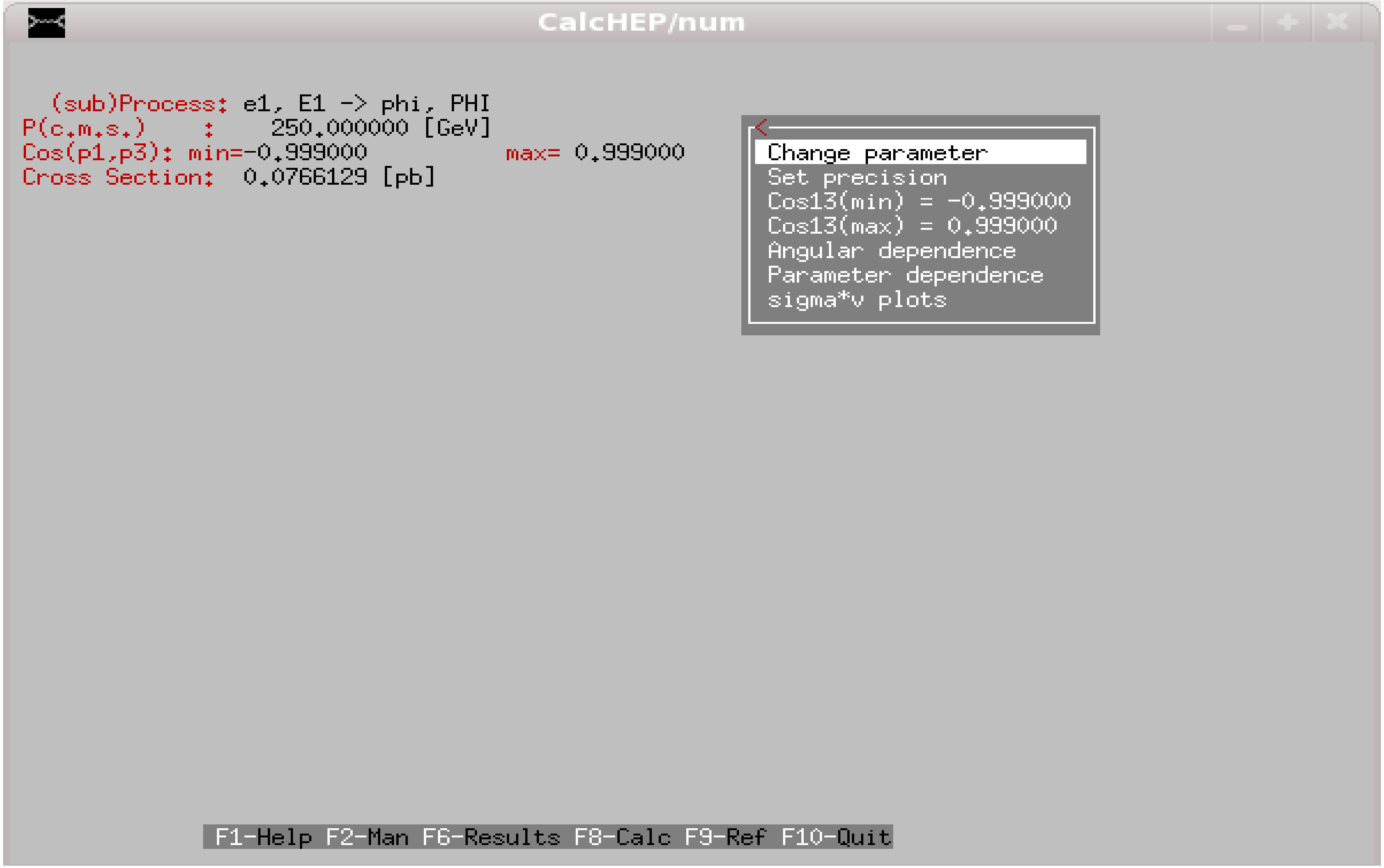


Using our newly implemented model in CalcHEP 6.



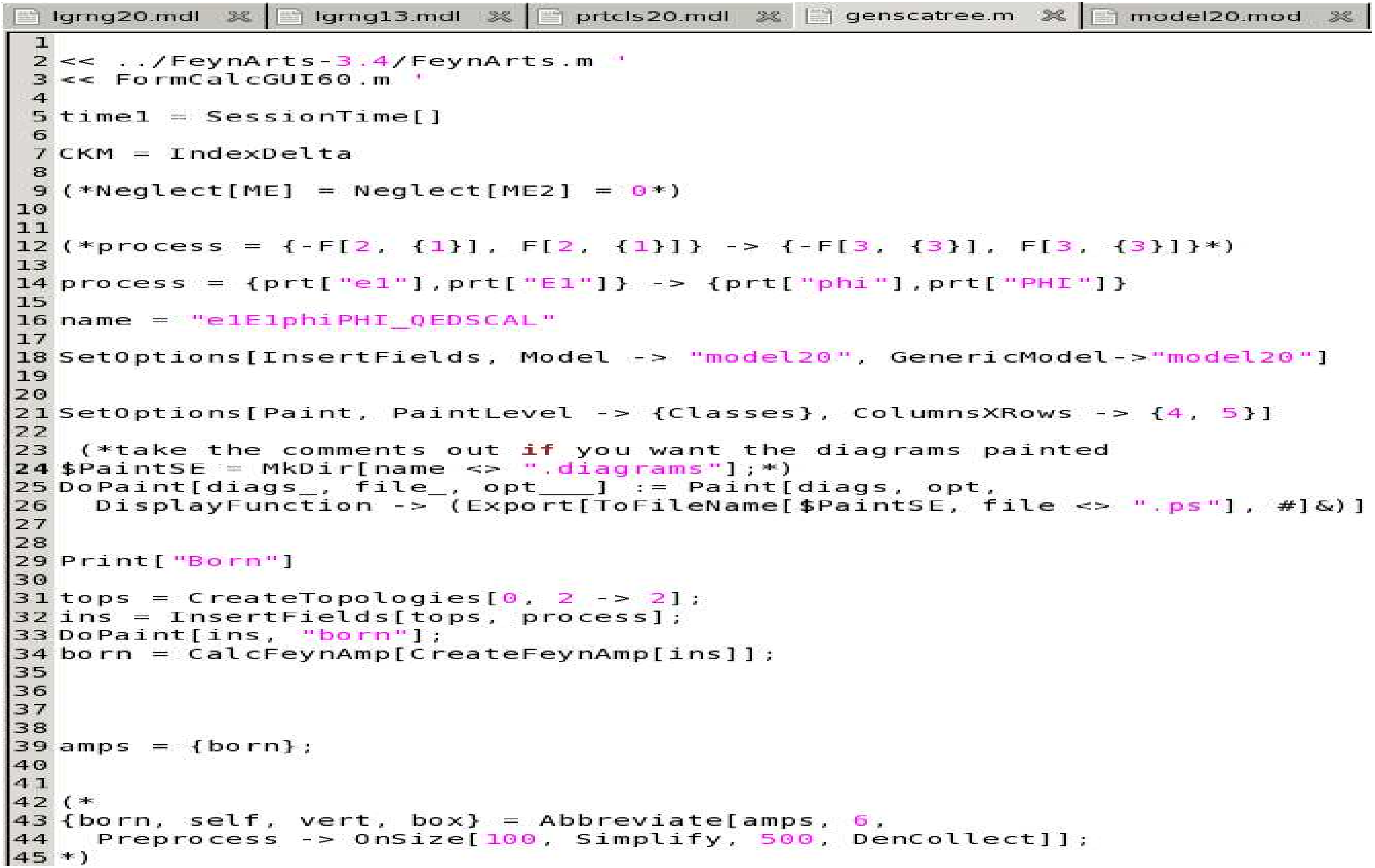
```
(sub)Process: e1, E1 -> phi, PHI
Monte Carlo session: 1(begin)
```

Using our newly implemented model in CalcHEP 7.



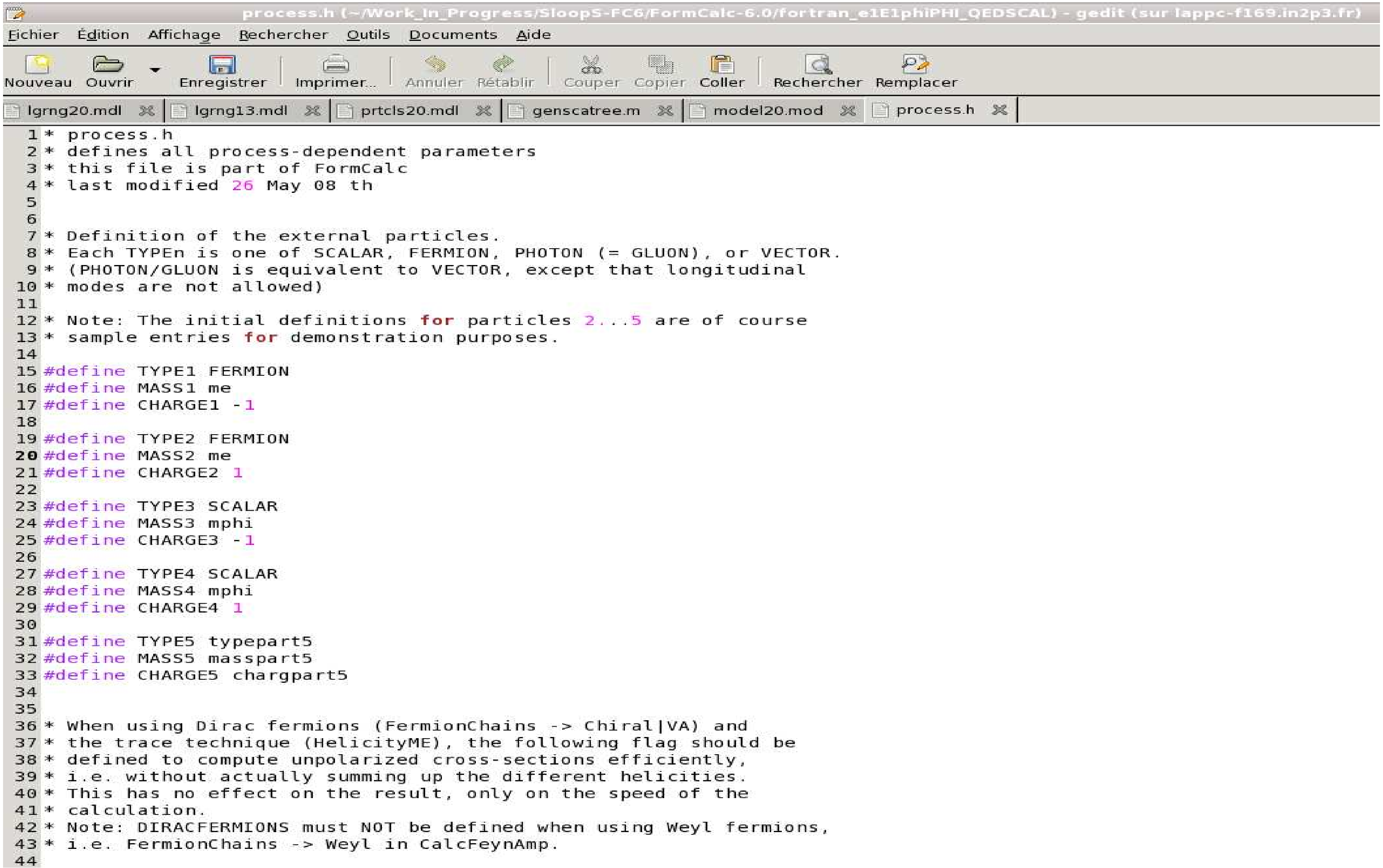
```
  (sub)Process: e1, E1 -> phi, PHI
P(c.m.s.)    :    250.000000 [GeV]
Cos(p1,p3): min=-0.999000          max= 0.999000
Cross Section:  0.0766129 [pb]
```

Using our newly implemented model in Feyn-Form Package 1.



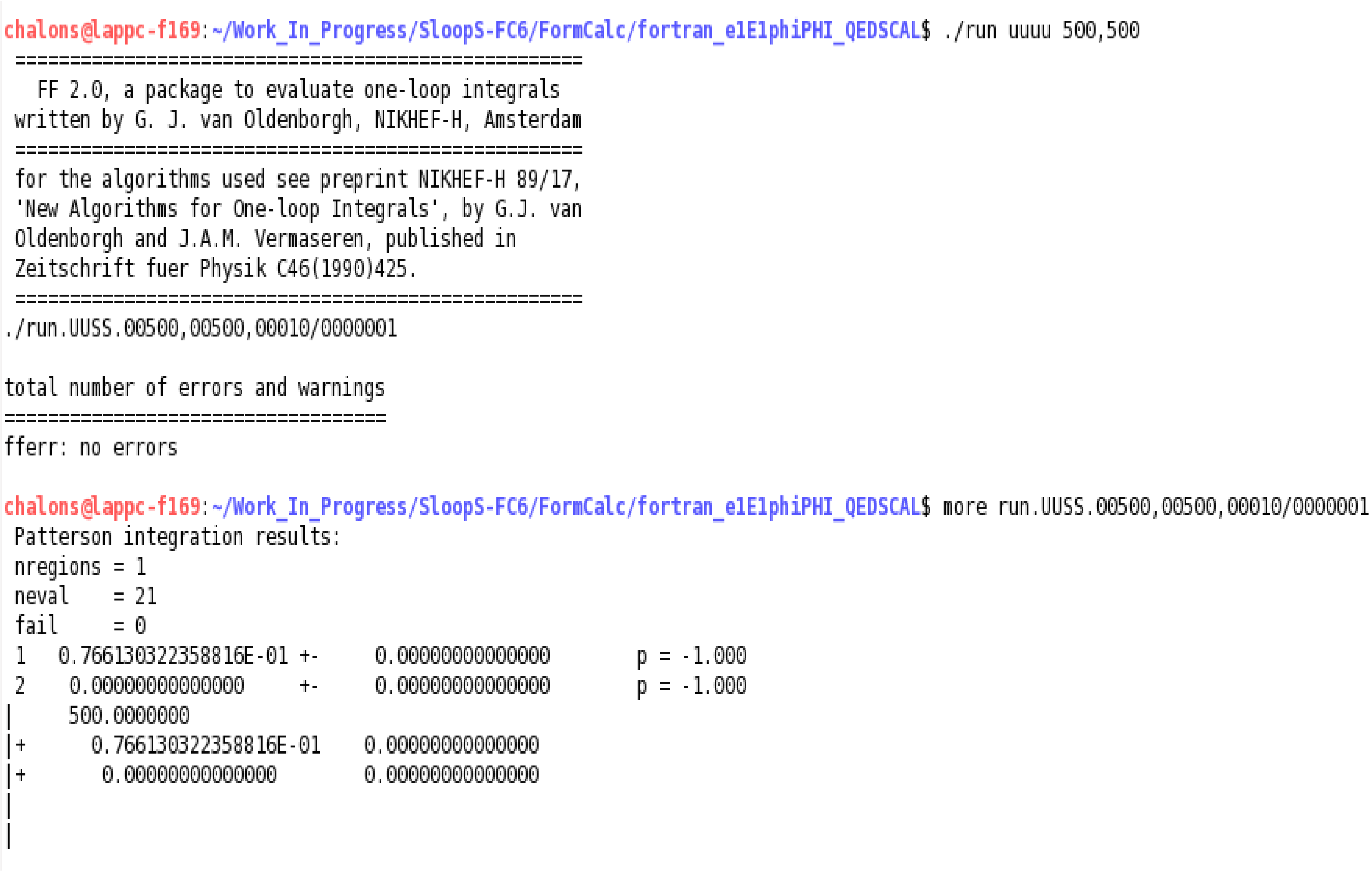
```

<< ../FeynArts-3.4/FeynArts.m '
<< FormCalcGUI60.m '

time1 = SessionTime[]

CKM = IndexDelta

(*Neglect[ME] = Neglect[ME2] = 0*)


(*process = {-F[2, {1}], F[2, {1}]} -> {-F[3, {3}], F[3, {3}]}*)

process = {prt["e1"],prt["E1"]} -> {prt["phi"],prt["PHI"]}

name = "e1E1phiPHI_QEDSCAL"

SetOptions[InsertFields, Model -> "model20", GenericModel->"model20"]


SetOptions[Paint, PaintLevel -> {Classes}, ColumnsXRows -> {4, 5}]

 (*take the comments out if you want the diagrams painted
$PaintSE = MkDir[name <> ".diagrams"];*)
DoPaint[diags_, file_, opt___] := Paint[diags, opt,
  DisplayFunction -> (Export[ToFileName[$PaintSE, file <> ".ps"], #]&)]


Print["Born"]

tops = CreateTopologies[0, 2 -> 2];
ins = InsertFields[tops, process];
DoPaint[ins, "born"];
born = CalcFeynAmp[CreateFeynAmp[ins]];




amps = {born};


(*
{born, self, vert, box} = Abbreviate[amps, 6,
  Preprocess -> OnSize[100, Simplify, 500, DenCollect]];
*)
```

Using our newly implemented model in Feyn-Form Package 2.



```
* process.h
* defines all process-dependent parameters
* this file is part of FormCalc
* last modified 26 May 08 th


* Definition of the external particles.
* Each TYPEn is one of SCALAR, FERMION, PHOTON (= GLUON), or VECTOR.
* (PHOTON/GLUON is equivalent to VECTOR, except that longitudinal
* modes are not allowed)

* Note: The initial definitions for particles 2...5 are of course
* sample entries for demonstration purposes.

#define TYPE1 FERMION
#define MASS1 me
#define CHARGE1 -1

#define TYPE2 FERMION
#define MASS2 me
#define CHARGE2 1

#define TYPE3 SCALAR
#define MASS3 mphi
#define CHARGE3 -1

#define TYPE4 SCALAR
#define MASS4 mphi
#define CHARGE4 1

#define TYPE5 typepart5
#define MASS5 masspart5
#define CHARGE5 chargpart5


* When using Dirac fermions (FermionChains -> Chiral|VA) and
* the trace technique (HelicityME), the following flag should be
* defined to compute unpolarized cross-sections efficiently,
* i.e. without actually summing up the different helicities.
* This has no effect on the result, only on the speed of the
* calculation.
* Note: DIRACFERMIONS must NOT be defined when using Weyl fermions,
* i.e. FermionChains -> Weyl in CalcFeynAmp.

```

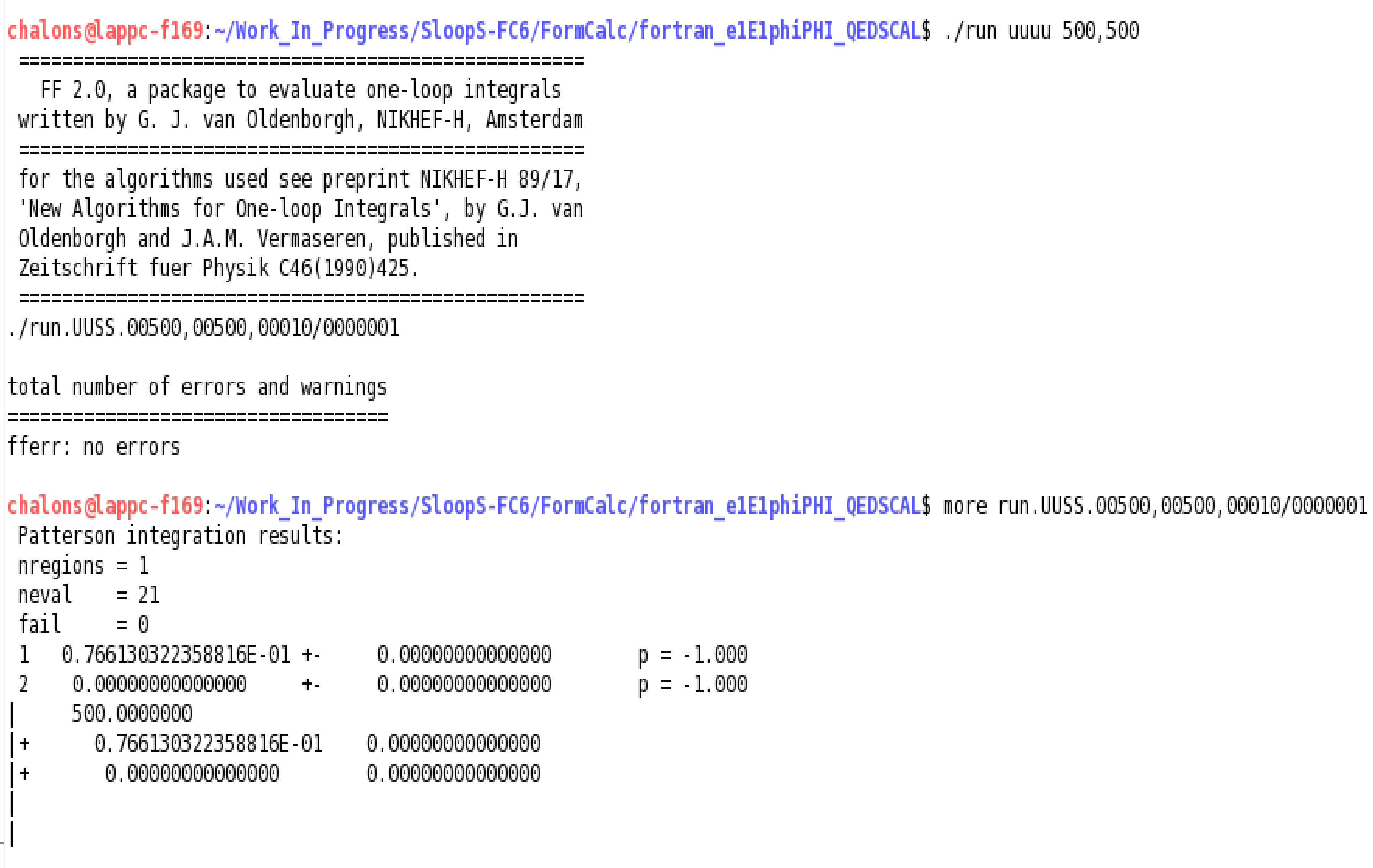
Using our newly implemented model in Feyn-Form Package 3.



```
chalons@lappc-f169:~/Work_In_Progress/SloopS-FC6/FormCalc/fortran_e1E1phiPHI_QEDSCAL$ ./run uuuu 500,500
 ====================================================
   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.
 ====================================================
./run.UUSS.00500,00500,00010/0000001

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

chalons@lappc-f169:~/Work_In_Progress/SloopS-FC6/FormCalc/fortran_e1E1phiPHI_QEDSCAL$ more run.UUSS.00500,00500,00010/0000001
 Patterson integration results:
 nregions = 1
 neval    = 21
 fail     = 0
 1   0.766130322358816E-01 +-     0.000000000000000       p = -1.000
 2    0.000000000000000     +-     0.000000000000000       p = -1.000
|     500.0000000
|+      0.766130322358816E-01    0.000000000000000
|+       0.000000000000000        0.000000000000000
|
|
```

Using our newly implemented model in Feyn-Form Package 3.

result agrees with CalcHEP



```
chalons@lappc-f169:~/Work_In_Progress/SloopS-FC6/FormCalc/fortran_e1E1phiPHI_QEDSCAL$ ./run uuuu 500,500
 ====================================================
   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.
 ====================================================
./run.UUSS.00500,00500,00010/0000001

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

chalons@lappc-f169:~/Work_In_Progress/SloopS-FC6/FormCalc/fortran_e1E1phiPHI_QEDSCAL$ more run.UUSS.00500,00500,00010/0000001
 Patterson integration results:
 nregions = 1
 neval    = 21
 fail     = 0
 1   0.766130322358816E-01 +-     0.000000000000000        p = -1.000
 2    0.000000000000000    +-     0.000000000000000        p = -1.000
|     500.0000000
|+      0.766130322358816E-01    0.000000000000000
|+       0.000000000000000       0.000000000000000
|
|
```

LanHEP, CalcHEP, micrOMEGAs: implementing a model for Dark Matter

- A particle that could qualify as a DM candidate need to be neutral and stable.

LanHEP, CalcHEP, micrOMEGAs: implementing a model for Dark Matter

- A particle that could qualify as a DM candidate need to be neutral and stable.
- One way for it to be stable is that it has to have a new quantum number that is odd as compared to the SM which would have this quantum number even.

LanHEP, CalcHEP, micrOMEGAs: implementing a model for Dark Matter

- A particle that could qualify as a DM candidate need to be neutral and stable.
- One way for it to be stable is that it has to have a new quantum number that is odd as compared to the SM which would have this quantum number even.
- Then the DM particle could qualify if it is the lightest among the odd particles

LanHEP, CalcHEP, micrOMEGAs: implementing a model for Dark Matter

- A particle that could qualify as a DM candidate need to be neutral and stable.
- One way for it to be stable is that it has to have a new quantum number that is odd as compared to the SM which would have this quantum number even.
- Then the DM particle could qualify if it is the lightest among the odd particles
- simple model: add both a charged scalar $s, S \longrightarrow phi, Phi$ and a heavy neutrino n, N to our QED model

LanHEP, CalcHEP, micrOMEGAs: implementing a model for Dark Matter

- A particle that could qualify as a DM candidate need to be neutral and stable.
- One way for it to be stable is that it has to have a new quantum number that is odd as compared to the SM which would have this quantum number even.
- Then the DM particle could qualify if it is the lightest among the odd particles
- simple model: add both a charged scalar $s, S \longrightarrow phi, Phi$ and a heavy neutrino n, N to our QED model
- n will couple through a magnetic moment coupling to the photon (example of a higher dim operator)

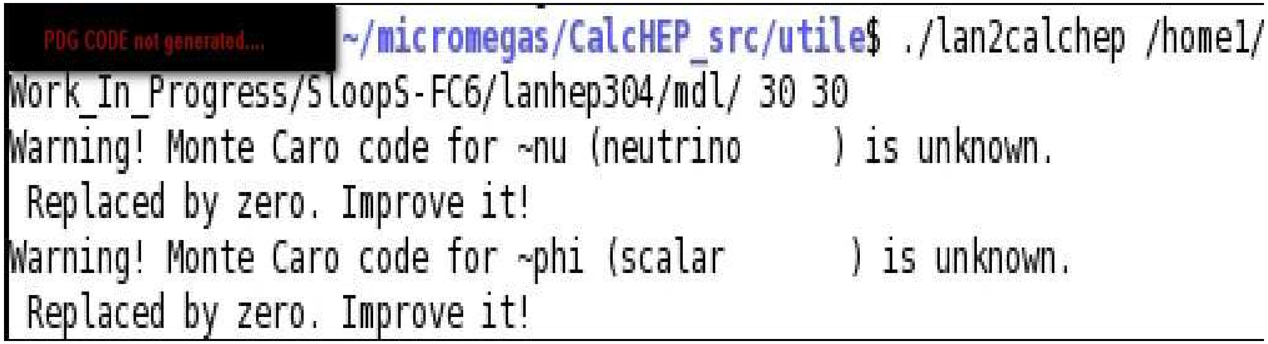
LanHEP, CalcHEP, micrOMEGAs: implementing a model for Dark Matter

- A particle that could qualify as a DM candidate need to be neutral and stable.
- One way for it to be stable is that it has to have a new quantum number that is odd as compared to the SM which would have this quantum number even.
- Then the DM particle could qualify if it is the lightest among the odd particles
- simple model: add both a charged scalar $s, S \longrightarrow phi, Phi$ and a heavy neutrino n, N to our QED model
- n will couple through a magnetic moment coupling to the photon (example of a higher dim operator)
- both n, s are odd, we will label them as $\tilde{n}, \tilde{s}$

LanHEP, CalcHEP, micrOMEGAs: implementing a model for Dark Matter

```
model qedscalneutrino/31.
parameter ee = 0.3133: ’Electric charge’.
parameter  mag = 0.01: ’magnetic momentum’.
vector A/A:photon.
let F^mu^nu=deriv^mu*A^nu-deriv^nu*A^mu.
spinor e1:(electron, mass me=0.000511).

spinor ~nu/ ~Nu :(neutrino, mass mnu=90, width wnu=0).
scalar ~s/ ~S:(scalar, mass mS=100, width wS=0).

lterm ee*E1*gamma*A*e1.

let Dphi^mu = (deriv^mu+i*ee*A^mu)*’~s’.
let DPHI^mu = (deriv^mu-i*ee*A^mu)*’~S’.

lterm DPHI*Dphi.

lterm  i*mag/2*’~Nu’*(gamma^mu*gamma^nu-gamma^nu*gamma^mu)*’~nu’*F^mu^nu

CheckHerm.
```

Lanhep pdg code for calchep.



PDG CODE not generated....

```
~/micromegas/CalcHEP_src/utile$ ./lan2calchep /home1/
Work_In_Progress/SloopS-FC6/lanhep304/mdl/ 30 30
Warning! Monte Caro code for ~nu (neutrino      ) is unknown.
 Replaced by zero. Improve it!
Warning! Monte Caro code for ~phi (scalar        ) is unknown.
 Replaced by zero. Improve it!
```

LanHEP output, Lagrangian

```
qedscalneutrino
 Lagrangian
P1   |P2   |P3   |P4   |>   Factor  <|> dLagrangian/ dA(p1) dA(p2) dA(p3
A    |~S   |~s   |     |ee           |m1.p2-m1.p3
E1   |e1   |A    |     |ee           |G(m3)
~Nu  |~nu  |A    |     |mag          |G(p3)*G(m3)-G(m3)*G(p3)
A    |A    |~S   |~s   |2*ee^2       |m1.m2
```

LanHEP output, variables

```
qedscalneutrino
 Variables
  Name | Value        |>  Comment                  <|
 ee    |0.3133        |Electric charge
 mag   |0.01          |magnetic momentum
 me    |0.000511      |mass of electron
 mnu   |90            |mass of neutrino
 wnu   |0             |width of neutrino
 mS    |100           |mass of scalar
 wS    |0             |width of scalar
```

Results of micrOMEGAs, large magnetic moment

```
=== MASSES OF ODD SECTOR: ===

Masses of odd sector Particles: ~nu  : mnu   =    90.0 || ~s   :
mS    =   100.0 ||

==== Physical Constraints: =====

==== Calculation of relic density =====
Dark Matter candidate is ~nu Xf=3.40e+01 Omega=5.83e-06

Channels which contribute to 1/(omega) more than 1%.
Relative contributions in % are displayed
 97% ~nu ~Nu -> A A
  3% ~nu ~Nu -> e1 E1

==== Indirect detection =======
1.38E-02 gamma with E > 1.00E-01 are generated at one collision
gamma flux for fi=0.00E+00[rad] is 3.00E-06[ph/cm^2/s/sr]

==== Calculation of WIMP-nucleons amplitudes  =====
```

Results of micrOMEGAs, co-annihilation!

```
mS  100. mnu 99.9. mag 0.0001.
=== MASSES OF ODD SECTOR: ===

Masses of odd sector Particles:
~nu  : mnu   =    99.9  ||
~s   : mS    =   100.0  ||

==== Physical Constraints: =====

==== Calculation of relic density =====
Dark Matter candidate is ~nu Xf=2.46e+01 Omega=1.22e-01

Channels which contribute to 1/(omega) more than 1%.
Relative contributions in % are displayed
 95% ~s ~S -> A A
  4% ~nu ~Nu -> e1 E1
  1% ~s ~S -> e1 E1

==== Indirect detection =======
4.78E-01 gamma with E > 1.00E-01 are generated at one collision
gamma flux for fi=0.00E+00[rad] is 2.49E-10[ph/cm^2/s/sr]
```

Part of the SM input file

Use of multiplets (doublets)

$$\Phi = \begin{pmatrix} -iW_f^+ \\ (\frac{2M_W}{es_W} + H + iZ_f)/\sqrt{2} \end{pmatrix},$$

Part of the SM input file

Use of multiplets (doublets)

$$\Phi = \begin{pmatrix} -iW_f^+ \\ (\frac{2M_W}{es_W} + H + iZ_f)/\sqrt{2} \end{pmatrix},$$

Option gauge in the declaration of gauge fields allows to use `gsb(Z)` and `gsb('W+')` for the goldstone bosons.

Part of the SM input file

```
model Higgs/1.
parameter EE = 0.31333 :  'Electromagnetic coupling constant',
      SW = 0.4740 :  'sin of the Weinberg angle (PDG-94)',
      CW = Sqrt(1-SW**2) :  'cos of the Weinberg angle'.
let g=EE/SW, g1=EE/CW.

vector A/A: (photon, gauge),
      Z/Z:('Z boson', mass MZ = 91.187, gauge),
      'W+'/'W-':  ('W boson', mass MW = MZ*CW, gauge).
scalar H/H:(Higgs, mass MH = 200, width wH = 1.461).

let B = -SW*Z+CW*A.
let W = {'W+', CW*Z+SW*A, 'W-'}.
let phi = { -i*gsb('W+'), (vev(2*MW/EE*SW)+H+i*gsb(Z))/Sqrt2 },
      Phi = anti(phi).

lterm -2*lambda*(phi*anti(phi)-v**2/2)**2 where
          lambda=(g*MH/MW)**2/16, v=2*MW*SW/EE.
let D^a^b^mu =(deriv^mu+i*g1/2*B^mu)*delta(2)^a^b
          +i*g/2*taupm^a^b^c*W^mu^c,
      Dc^a^b^mu=(deriv^mu-i*g1/2*B^mu)*delta(2)^a^b
          -i*g/2*taupm^a^b^c*anti(W)^mu^c.
lterm D^a^b^mu*phi^b*Dc^a^c^mu*Phi^c.
```

Bosonic sector of the SM, Feynman rules

Fields in the vertex	Variational derivative of Lagrangian by fields
$A_\mu \quad W^+{}_\nu \quad W_F^-$	$ieM_W g^{\mu\nu}$
$A_\mu \quad W_F^+ \quad W^-{}_\nu$	$-ieM_W g^{\mu\nu}$
$A_\mu \quad W_F^+ \quad W_F^-$	$-e(p_2^\mu - p_3^\mu)$
$H \quad W^+{}_\mu \quad W^-{}_\nu$	$\frac{eM_W}{s_w} g^{\mu\nu}$
$H \quad W^+{}_\mu \quad W_F^-$	$-\frac{1}{2}\frac{ie}{s_w}(p_1^\mu - p_3^\mu)$
$H \quad W_F^+ \quad W^-{}_\mu$	$\frac{1}{2}\frac{ie}{s_w}(p_2^\mu - p_1^\mu)$
$H \quad Z_\mu \quad Z_\nu$	$\frac{eM_W}{{c_w}^2 s_w} g^{\mu\nu}$
$H \quad Z_\mu \quad Z_F$	$-\frac{1}{2}\frac{ie}{c_w s_w}(p_1^\mu - p_3^\mu)$
$W^+{}_\mu \quad W_F^- \quad Z_\nu$	$-\frac{ieM_W s_w}{c_w} g^{\mu\nu}$
$W^+{}_\mu \quad W_F^- \quad Z_F$	$\frac{1}{2}\frac{e}{s_w}(p_3^\mu - p_2^\mu)$
$W_F^+ \quad W^-{}_\mu \quad Z_\nu$	$\frac{ieM_W s_w}{c_w} g^{\mu\nu}$
$W_F^+ \quad W^-{}_\mu \quad Z_F$	$\frac{1}{2}\frac{e}{s_w}(p_1^\mu - p_3^\mu)$
$W_F^+ \quad W_F^- \quad Z_\mu$	$-\frac{1}{2}\frac{(1-2{s_w}^2)e}{c_w s_w}(p_1^\mu - p_2^\mu)$
$A_\mu \quad A_\nu \quad W_F^+ \quad W_F^-$	$2e^2 g^{\mu\nu}$
$A_\mu \quad H \quad W^+{}_\nu \quad W_F^-$	$\frac{1}{2}\frac{ie^2}{s_w} g^{\mu\nu}$
$A_\mu \quad H \quad W_F^+ \quad W^-{}_\nu$	$-\frac{1}{2}\frac{ie^2}{s_w} g^{\mu\nu}$
$A_\mu \quad W^+{}_\nu \quad W_F^- \quad Z_F$	$-\frac{1}{2}\frac{e^2}{s_w} g^{\mu\nu}$
$A_\mu \quad W_F^+ \quad W^-{}_\nu \quad Z_F$	$-\frac{1}{2}\frac{e^2}{s_w} g^{\mu\nu}$
$A_\mu \quad W_F^+ \quad W_F^- \quad Z_\nu$	$\frac{(1-2{s_w}^2)e^2}{c_w s_w} g^{\mu\nu}$
$H \quad H \quad W^+{}_\mu \quad W^-{}_\nu$	$\frac{1}{2}\frac{e^2}{{s_w}^2} g^{\mu\nu}$
$H \quad H \quad Z_\mu \quad Z_\nu$	$\frac{1}{2}\frac{e^2}{{c_w}^2 {s_w}^2} g^{\mu\nu}$
$H \quad W^+{}_\mu \quad W_F^- \quad Z_\nu$	$-\frac{1}{2}\frac{ie^2}{c_w} g^{\mu\nu}$
$H \quad W_F^+ \quad W^-{}_\mu \quad Z_\nu$	$\frac{1}{2}\frac{ie^2}{c_w} g^{\mu\nu}$
$W^+{}_\mu \quad W_F^+ \quad W^-{}_\nu \quad W_F^-$	$\frac{1}{2}\frac{e^2}{{s_w}^2} g^{\mu\nu}$
$W^+{}_\mu \quad W^-{}_\nu \quad Z_F \quad Z_F$	$\frac{1}{2}\frac{e^2}{{s_w}^2} g^{\mu\nu}$
$W^+{}_\mu \quad W_F^- \quad Z_\nu \quad Z_F$	$\frac{1}{2}\frac{e^2}{c_w} g^{\mu\nu}$
$W_F^+ \quad W^-{}_\mu \quad Z_\nu \quad Z_F$	$\frac{1}{2}\frac{e^2}{c_w} g^{\mu\nu}$
$W_F^+ \quad W_F^- \quad Z_\mu \quad Z_\nu$	$\frac{1}{2}\frac{(1-2{s_w}^2)^2 e^2}{{c_w}^2 {s_w}^2} g^{\mu\nu}$
$Z_\mu \quad Z_\nu \quad Z_F \quad Z_F$	$\frac{1}{2}\frac{e^2}{{c_w}^2 {s_w}^2} g^{\mu\nu}$

Weyl and two component fermion vs Dirac

$$\psi = \begin{pmatrix} \xi \\ \bar{\eta} \end{pmatrix}, \quad \psi^c = \begin{pmatrix} \eta \\ \bar{\xi} \end{pmatrix}, \quad \bar{\psi} = \begin{pmatrix} \eta \\ \bar{\xi} \end{pmatrix}^T, \quad \bar{\psi}^c = \begin{pmatrix} \xi \\ \bar{\eta} \end{pmatrix}^T.$$

If the user has declared a spinor particle p (with antiparticle P), the LanHEP notation for its components is:

$\xi \longrightarrow$ `up(p)`

$\bar{\eta} \longrightarrow$ `down(p)`

$\eta \longrightarrow$ `up(cc(p))` *or* `up(P)`

$\bar{\xi} \longrightarrow$ `down(cc(p))` *or* `down(P)`

Weyl and two component fermion vs Dirac

$$\psi = \begin{pmatrix} \xi \\ \bar\eta \end{pmatrix}, \qquad \psi^c = \begin{pmatrix} \eta \\ \bar\xi \end{pmatrix}, \qquad \bar\psi = \begin{pmatrix} \eta \\ \bar\xi \end{pmatrix}^T, \qquad \bar\psi^c = \begin{pmatrix} \xi \\ \bar\eta \end{pmatrix}^T.$$

$\eta_1\xi_2 = \bar\psi_1 P_L \psi_2$	`up(P1)*up(p2)`	$\longrightarrow$	`P1*(1-gamma5)/2*p2`
$\bar\eta_1\bar\xi_2 = \bar\psi_1 P_R \psi_2$	`down(P1)*down(p2)`	$\longrightarrow$	`P1*(1+gamma5)/2*p2`
$\xi_1\xi_2 = \bar\psi_1^c P_L \psi_2$	`up(p1)*up(p2)`	$\longrightarrow$	`cc(p1)*(1-gamma5)/2*p2`
$\bar\xi_1\bar\xi_2 = \bar\psi_1 P_R \psi_2^c$	`down(P1)*down(P2)`	$\longrightarrow$	`P1*(1+gamma5)/2*cc(P2)`
$\xi_1\eta_2 = \bar\psi_1^c P_L \psi_2^c$	`up(p1)*up(P2)`	$\longrightarrow$	`cc(p1)*(1-gamma5)/2*cc(`
$\bar\xi_1\bar\eta_2 = \bar\psi_1^c P_R \psi_2^c$	`down(p1)*down(P2)`	$\longrightarrow$	`cc(p1)*(1+gamma5)/2*cc(`
$\eta_1\eta_2 = \bar\psi_1 P_L \psi_2^c$	`up(P1)*up(P2)`	$\longrightarrow$	`P1*(1-gamma5)/2*cc(P2)`
$\bar\eta_1\bar\eta_2 = \bar\psi_1^c P_R \psi_2$	`down(p1)*down(p2)`	$\longrightarrow$	`cc(p1)*(1+gamma5)/2*p2`
$\bar\xi_1\sigma^\mu\xi_2 = \bar\psi_1\gamma^\mu P_L\psi_2$	`down(P1)*sigma*up(p2)`	$\longrightarrow$	`P1*gamma*(1-gamma5)/2*p`
$\bar\eta_1\sigma^\mu\eta_2 = \bar\psi_1^c\gamma^\mu P_L\psi_2^c$	`down(p1)*sigma*up(P2)`	$\longrightarrow$	`cc(p1)*gamma*(1-gamma5)`

Superpotential $\mathbf{W}$

$$\mathbf{W} = eps_{ij}(\mu H^1_i H^2_j + M_l^{IJ} H^1_i L^I_j R^J + M_d^{IJ} H^1_i Q^I_j D^J + M_u^{IJ} H^2_i Q^I_j U^J)$$

(H_i, L, Q, R, U, D defined as doublets and singlets, here in terms of scalar part.)

```
keep_lets W.
let W=eps*(mu*H1*H2+ml*H1*L*R+md*H1*Q*D+mu*H2*Q*U).
```

Yukawa interactions

$$-\frac{1}{2}\left(\frac{\partial^2 W}{\partial A_i \partial A_j}\Psi_i\Psi_j + h.c.\right)$$

Ψ_i fermionic partners of A_i

```
lterm - df(W,H1,H2)*fH1*fH2 - ...  + AddHermConj.
```

$F_i^* F_i$ terms, $F_i = \partial W / \partial A_i$

```
lterm - df(W,H1)*df(Wc,H1c) - ....
```

or even shorter

```
lterm - dfdfc(W,H1) - ....
```

Running the codes

run lanHEP to generate the output files (compHEP)

```
cd ~/lanhep304/mdl/
./lhep qedscalneutrino.mdl
```

run convert output from compHEP to calcHEP

```
cd ~/micromegas/CalcHEP_src/utile/
./lan2calchep ~/lanhep304/mdl/ 30 1
```

running micrOMEGAs

```
cd ~/micromegas
```

create new project with new model

```
./newProject DMheavyneut
cd DMheavyneut
```

import the model (into CalcHEP)

```
mv ~/micromegas/CalcHEP_src/utile/*1.mdl work/models/.
```

compile and execute

```
gmake main=main.c
./main data.par
```

Lanhep at one-loop

New gauge structures, novel gauge fixing

Interface with FeynArts/FormCacl/LoopTools

MSSM example

N. Baro, FB, G. Chalons, G. Drieu La Rochelle, S. Hao, Ninh Le Duc, A. Semenov, (D. Temes)

- Need for an automatic tool for susy calculations, for Colliders and Dark Matter, On-Shell scheme
- handles large numbers of diagrams both for tree-level
- and loop level
- able to compute loop diagrams at $v = 0$: dark matter, LSP, move at galactic velocities, $v = 10^{-3}$
- ability to check results: UV and IR finiteness but also gauge parameter independence for example
- ability to include different models easily and switch between different renormalisation schemes
- Used for SM one-loop multi-leg: new powerful loop libraries (with Ninh Le Duc)

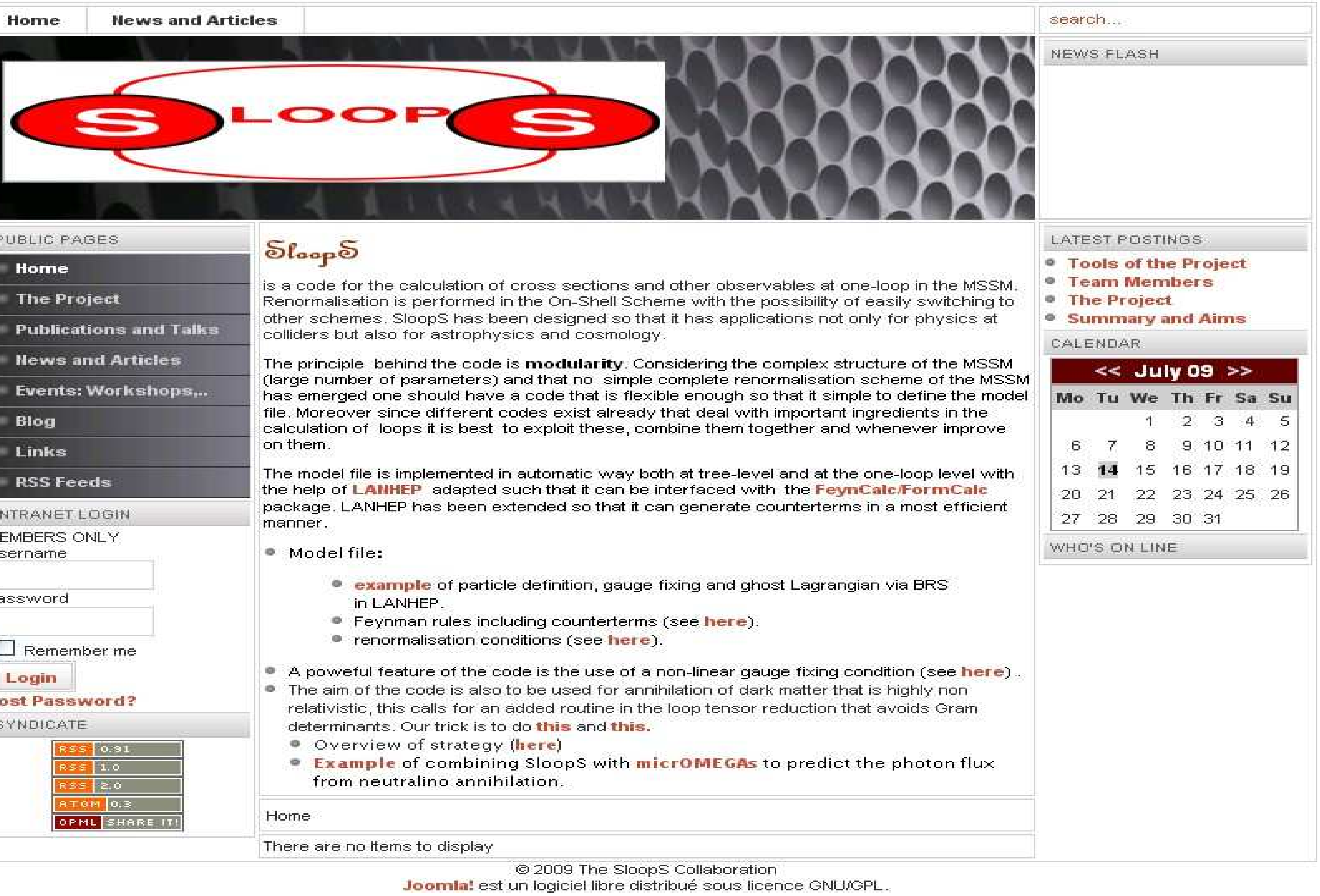
LanHEP application to one-loop calculation: SloopS project

- *FeynArts* and *FormCalc* are used for matrix element calculation: *FeynArts* model format output implemented in LanHEP.
- Shifts in fields and parameters to produce counterterms by LanHEP:

  ```
  infinitesimal dMHsq, dMZsq, dMWsq,dZAA, dZAZ, dZZA, dZZZ, dZW,
  dZH. infinitesimal   dEE= -(dZAA - SW/CW*dZZA)/2. transform
  A->A*(1+dZAA/2)+dZAZ*Z/2, Z->Z*(1+dZZZ/2)+dZZA*A/2,
        'W+'->'W+'*(1+dZW/2),'W-'->'W-'*(1+dZW/2), H->H*(1+dZH/2).
  ```

  Different normalization schemes can be used, easy to switch between different RS
- Non-linear gauge fixing (see later)

SloopS



Mo	Tu	We	Th	Fr	Sa	Su
		1	2	3	4	5
6	7	8	9	10	11	12
13	**14**	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

SloopS

is a code for the calculation of cross sections and other observables at one-loop in the MSSM. Renormalisation is performed in the On-Shell Scheme with the possibility of easily switching to other schemes. SloopS has been designed so that it has applications not only for physics at colliders but also for astrophysics and cosmology.

The principle behind the code is **modularity**. Considering the complex structure of the MSSM (large number of parameters) and that no simple complete renormalisation scheme of the MSSM has emerged one should have a code that is flexible enough so that it simple to define the model file. Moreover since different codes exist already that deal with important ingredients in the calculation of loops it is best to exploit these, combine them together and whenever improve on them.

The model file is implemented in automatic way both at tree-level and at the one-loop level with the help of **LANHEP** adapted such that it can be interfaced with the **FeynCalc/FormCalc** package. LANHEP has been extended so that it can generate counterterms in a most efficient manner.

- Model file:
    - **example** of particle definition, gauge fixing and ghost Lagrangian via BRS in LANHEP.
    - Feynman rules including counterterms (see **here**).
    - renormalisation conditions (see **here**).
- A poweful feature of the code is the use of a non-linear gauge fixing condition (see **here**) .
- The aim of the code is also to be used for annihilation of dark matter that is highly non relativistic, this calls for an added routine in the loop tensor reduction that avoids Gram determinants. Our trick is to do **this** and **this.**
    - Overview of strategy (**here**)
    - **Example** of combining SloopS with **micrOMEGAs** to predict the photon flux from neutralino annihilation.

Home

There are no Items to display

© 2009 The SloopS Collaboration

Joomla! est un logiciel libre distribué sous licence GNU/GPL.

Strategy: Exploiting and interfacing modules from different codes

Lagrangian of the model defined in LanHEP

- particle content
- interaction terms
- shifts in fields and parameters
- ghost terms constructed by BRST

↓ ↓

Generic Model
-kinematical structures

Classes Model
-Feynman rules, including CT

⇓

Evaluation via FeynArts-FormCalc

LoopTools modified!!
tensor reduction inappropriate for small relative velocities
(Zero Gram determinants)

⇑

Renormalisation scheme

- definition of renorm. const. in the classes model

Non-Linear gauge-fixing constraints, gauge parameter dependence checks

From the Lagrangian to the Feynman Rules

```
vector
    A/A: (photon, gauge),
    Z/Z:(’Z boson’, mass MZ = 91.1875, gauge),
    ’W+’/’W-’: (’W boson’, mass MW = MZ*CW, gauge).
scalar  H/H:(Higgs, mass MH = 115).
```

```
transform A->A*(1+dZAA/2)+dZAZ*Z/2, Z->Z*(1+dZZZ/2)+dZZA*A/2,
    ’W+’->’W+’*(1+dZW/2),’W-’->’W-’*(1+dZW/2).
transform H->H*(1+dZH/2), ’Z.f’->’Z.f’*(1+dZZf/2),
    ’W+.f’->’W+.f’*(1+dZWf/2),’W-.f’->’W-.f’*(1+dZWf/2).

let pp = { -i*’W+.f’,  (vev(2*MW/EE*SW)+H+i*’Z.f’)/Sqrt2 },
PP=anti(pp).
```

```
lterm -2*lambda*(pp*anti(pp)-v**2/2)**2
     where
    lambda=(EE*MH/MW/SW)**2/16, v=2*MW*SW/EE .

let Dpp^mu^a = (deriv^mu+i*g1/2*B0^mu)*pp^a +
     i*g/2*taupm^a^b^c*WW^mu^c*pp^b.
let DPP^mu^a = (deriv^mu-i*g1/2*B0^mu)*PP^a
    -i*g/2*taupm^a^b^c*{’W-’^mu,W3^mu,’W+’^mu}^c*PP^b.
lterm DPP*Dpp.
```

Gauge fixing and BRS transformation

```
let G_Z = deriv*Z+(MW/CW+EE/SW/CW/2*nle*H)*’Z.f’.

lterm -G_A**2/2 - G_Wp*G_Wm - G_Z**2/2.

lterm -’Z.C’*brst(G_Z).
```

From the Lagrangian to the Feynman Rules

```
vector
    A/A: (photon, gauge),
    Z/Z:('Z boson', mass MZ = 91.1875, gauge),
    'W+'/'W-': ('W boson', mass MW = MZ*CW, gauge).
scalar  H/H:(Higgs, mass MH = 115).



transform A->A*(1+dZAA/2)+dZAZ*Z/2, Z->Z*(1+dZZZ/2)+dZZA*A/2,
    'W+'->'W+'*(1+dZW/2),'W-'->'W-'*(1+dZW/2).
transform H->H*(1+dZH/2), 'Z.f'->'Z.f'*(1+dZZf/2),
    'W+.f'->'W+.f'*(1+dZWf/2),'W-.f'->'W-.f'*(1+dZWf/2).

let pp = { -i*'W+.f',  (vev(2*MW/EE*SW)+H+i*'Z.f')/Sqrt2 },
PP=anti(pp).



lterm -2*lambda*(pp*anti(pp)-v**2/2)**2
     where
    lambda=(EE*MH/MW/SW)**2/16, v=2*MW*SW/EE .

let Dpp^mu^a = (deriv^mu+i*g1/2*B0^mu)*pp^a +
     i*g/2*taupm^a^b^c*WW^mu^c*pp^b.
let DPP^mu^a = (deriv^mu-i*g1/2*B0^mu)*PP^a
    -i*g/2*taupm^a^b^c*{'W-'^mu,W3^mu,'W+'^mu}^c*PP^b.
lterm DPP*Dpp.
```

Gauge fixing and BRS transformation

```
let G_Z = deriv*Z+(MW/CW+EE/SW/CW/2*nle*H)*'Z.f'.

lterm -G_A**2/2 - G_Wp*G_Wm - G_Z**2/2.

lterm -'Z.C'*brst(G_Z).
```

Output of Feynman Rules with Counterterms !!

```
M$CouplingMatrices = {

   (*------     H   H   ------*)
    C[ S[3], S[3] ] == - I *
{
 { 0 , dZH },
 { 0 , MH^2 dZH + dMHsq }
},
   (*------     W+.f   W-.f   ------*)
    C[ S[2], -S[2] ] == - I *
{
 { 0 , dZWf },
 { 0, 0 }
}, (*------     A   Z   ------*)
    C[ V[1], V[2] ] == 1/2 I / CW^2 MW^2 *
{
 { 0, 0 },
 { 0 , dZZA },
 { 0, 0 }
},


(*------     H   H   H   ------*)
    C[ S[3], S[3], S[3] ] == -3/4 I EE / MW / SW *
{
 { 2 MH^2 , 3 MH^2 dZH -2 MH^2 / SW dSW - MH^2 / MW^2 dMWsq
},
   (*------     H   W+.f   W-.f   ------*)
    C[ S[3], S[2], -S[2] ] == -1/4 I EE / MW / SW *
{
 { 2 MH^2 , MH^2 dZH + 2 MH^2 dZWf -2 MH^2 / SW dSW - MH^2 /
},


   (*------     W-.C   A.c   W+   ------*)
    C[ -U[3], U[1], V[3] ] == - I EE *
{
 { 1 },
 { - nla }
},
```

From the Lagrangian to the Feynman Rules

```
vector
    A/A: (photon, gauge),
    Z/Z:('Z boson', mass MZ = 91.1875, gauge),
    'W+'/'W-': ('W boson', mass MW = MZ*CW, gauge).
scalar  H/H:(Higgs, mass MH = 115).



transform A->A*(1+dZAA/2)+dZAZ*Z/2, Z->Z*(1+dZZZ/2)+dZZA*A/2,
    'W+'->'W+'*(1+dZW/2),'W-'->'W-'*(1+dZW/2).
transform H->H*(1+dZH/2), 'Z.f'->'Z.f'*(1+dZZf/2),
    'W+.f'->'W+.f'*(1+dZWf/2),'W-.f'->'W-.f'*(1+dZWf/2).

let pp = { -i*'W+.f',  (vev(2*MW/EE*SW)+H+i*'Z.f')/Sqrt2 },
PP=anti(pp).



lterm -2*lambda*(pp*anti(pp)-v**2/2)**2
     where
    lambda=(EE*MH/MW/SW)**2/16, v=2*MW*SW/EE .

let Dpp^mu^a = (deriv^mu+i*g1/2*B0^mu)*pp^a +
     i*g/2*taupm^a^b^c*WW^mu^c*pp^b.
let DPP^mu^a = (deriv^mu-i*g1/2*B0^mu)*PP^a
    -i*g/2*taupm^a^b^c*{'W-'^mu,W3^mu,'W+'^mu}^c*PP^b.
lterm DPP*Dpp.
```

Gauge fixing and BRS transformation

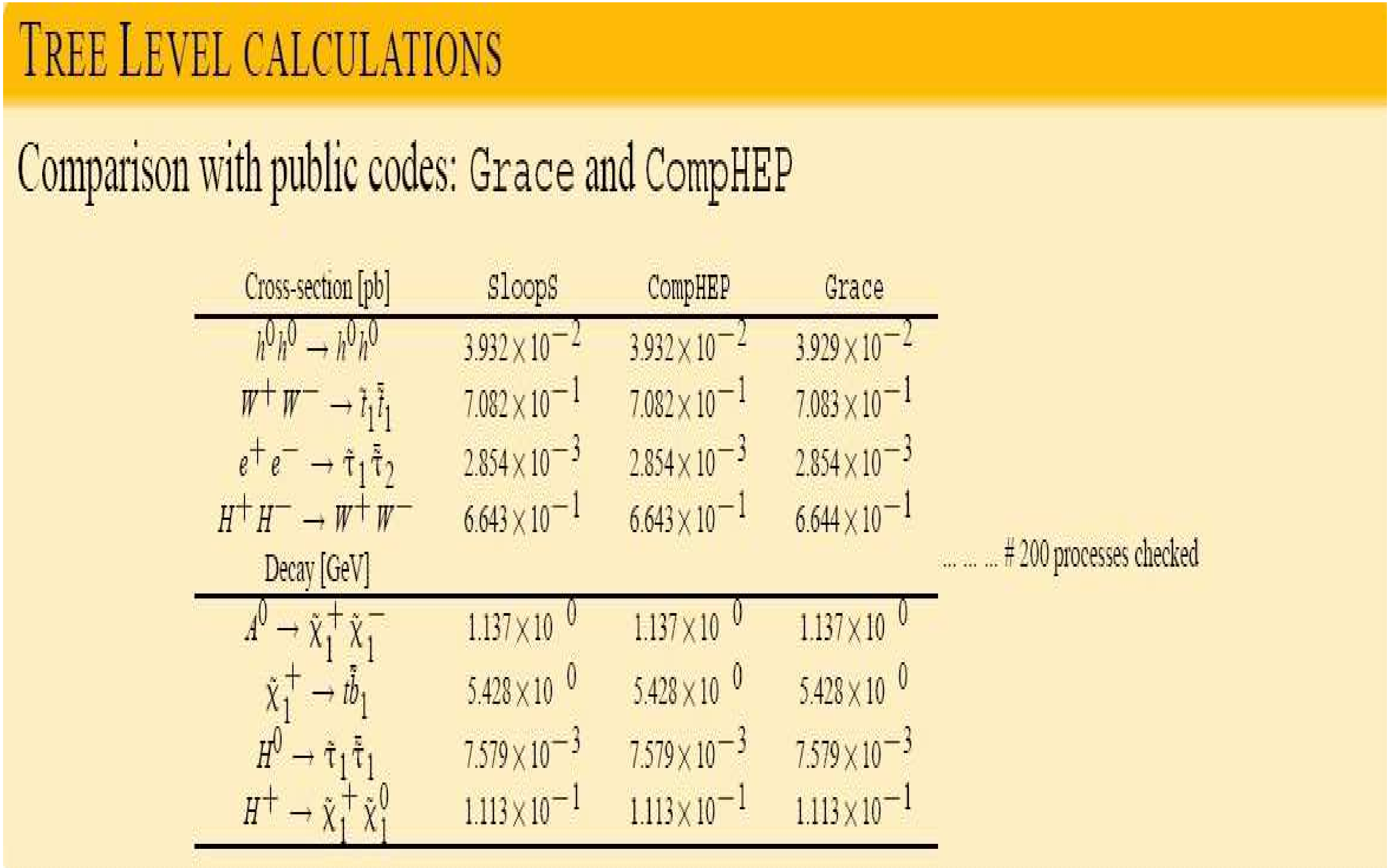
```
let G_Z = deriv*Z+(MW/CW+EE/SW/CW/2*nle*H)*'Z.f'.

lterm -G_A**2/2 - G_Wp*G_Wm - G_Z**2/2.

lterm -'Z.C'*brst(G_Z).
```

```
RenConst[ dMHsq ] := ReTilde[SelfEnergy[prt["H"] -> prt["H"], MH]]
RenConst[ dZH ] := -ReTilde[DSelfEnergy[prt["H"] -> prt["H"], MH]]
RenConst[ dZZf ] := -ReTilde[DSelfEnergy[prt["Z.f"] -> prt["Z.f"],
MZ]] RenConst[ dZWf ] := -ReTilde[DSelfEnergy[prt["W+.f"] ->
prt["W+.f"], MW]]
```

Output of Feynman Rules with Counterterms !!

```
M$CouplingMatrices = {

  (*------    H   H   ------*)
   C[ S[3], S[3] ] == - I *
{
 { 0 , dZH },
 { 0 , MH^2 dZH + dMHsq }
},
  (*------    W+.f  W-.f  ------*)
   C[ S[2], -S[2] ] == - I *
{
 { 0 , dZWf },
 { 0, 0 }
}, (*------    A   Z   ------*)
   C[ V[1], V[2] ] == 1/2 I / CW^2 MW^2 *
{
 { 0, 0 },
 { 0 , dZZA },
 { 0, 0 }
},


(*------    H   H   H   ------*)
   C[ S[3], S[3], S[3] ] == -3/4 I EE / MW / SW *
{
 { 2 MH^2 , 3 MH^2 dZH -2 MH^2 / SW dSW - MH^2 / MW^2 dMWsq
},
  (*------    H  W+.f  W-.f  ------*)
   C[ S[3], S[2], -S[2] ] == -1/4 I EE / MW / SW *
{
 { 2 MH^2 , MH^2 dZH + 2 MH^2 dZWf -2 MH^2 / SW dSW - MH^2 /
},


  (*------    W-.C  A.c  W+  ------*)
   C[ -U[3], U[1], V[3] ] == - I EE *
{
 { 1 },
 { - nla }
},
```

SloopS for SUSY at tree-level



TREE LEVEL CALCULATIONS

Comparison with public codes: Grace and CompHEP

Cross-section [pb]	SloopS	CompHEP	Grace
$h^0 h^0 \to h^0 h^0$	3.932×10^{-2}	3.932×10^{-2}	3.929×10^{-2}
$W^+ W^- \to \tilde{t}_1 \bar{\tilde{t}}_1$	7.082×10^{-1}	7.082×10^{-1}	7.083×10^{-1}
$e^+ e^- \to \tilde{\tau}_1 \bar{\tilde{\tau}}_2$	2.854×10^{-3}	2.854×10^{-3}	2.854×10^{-3}
$H^+ H^- \to W^+ W^-$	6.643×10^{-1}	6.643×10^{-1}	6.644×10^{-1}
Decay [GeV]			
$A^0 \to \tilde{\chi}_1^+ \tilde{\chi}_1^-$	1.137×10^{0}	1.137×10^{0}	1.137×10^{0}
$\tilde{\chi}_1^+ \to t \bar{\tilde{b}}_1$	5.428×10^{0}	5.428×10^{0}	5.428×10^{0}
$H^0 \to \tilde{\tau}_1 \bar{\tilde{\tau}}_1$	7.579×10^{-3}	7.579×10^{-3}	7.579×10^{-3}
$H^+ \to \tilde{\chi}_1^+ \tilde{\chi}_1^0$	1.113×10^{-1}	1.113×10^{-1}	1.113×10^{-1}

... ... # 200 processes checked

Non-linear gauge implementation

$$\begin{aligned}\mathcal{L}_{GF} &= -\frac{1}{\xi_W}|\partial.W^+ + \xi_W \frac{g}{2} v G^+|^2 \\ &\quad -\frac{1}{2\xi_Z}(\partial.Z + \xi_Z \frac{g}{2c_W} v + G^0)^2 - \frac{1}{2\xi_\gamma}(\partial.A)^2\end{aligned}$$

This only affects the propagators. Usually calculations done with $\xi = 1$, otherwise large expressions, higher rank tensors, unphysical thresholds,..

$$\frac{1}{k^2 - M_W^2}\left(g_{\mu\nu} - (1 - \xi_W)\frac{k_\mu k_\nu}{k^2 - \xi_W M_W^2}\right)$$

how to have $\xi = 1$ and still check for gauge parameter independence?

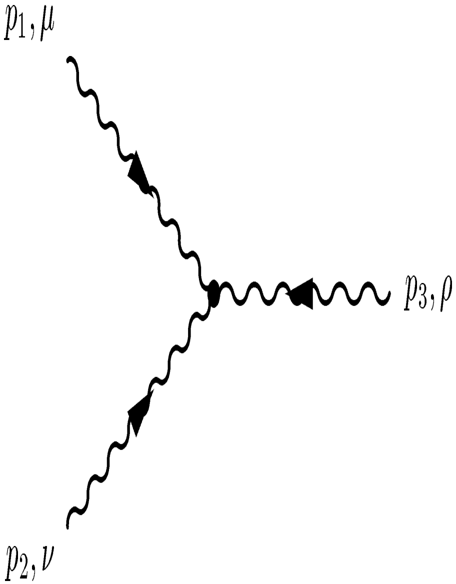
Non-linear gauge implementation

$$\begin{aligned}\mathcal{L}_{GF} \quad = \quad & -\frac{1}{\xi_W}\left|(\partial_\mu - ie\tilde{\alpha}A_\mu - igc_W\tilde{\beta}Z_\mu)W^{\mu\,+} + \xi_W\frac{g}{2}(v+\tilde{\delta}h+\tilde{\omega}H+i\tilde{\rho}A^0+i\tilde{\kappa}G^0)G^+\right|^2 \\ & -\frac{1}{2\xi_Z}(\partial.Z+\xi_Z\frac{g}{2c_W}(v+\tilde{\epsilon}h+\tilde{\gamma}H)G^0)^2-\frac{1}{2\xi_\gamma}(\partial.A)^2\end{aligned}$$

- quite a handful of gauge parameters, but with $\xi_i = 1$, no "unphysical threshold", no higher rank tensors, gauge parameter dependence in gauge/Goldstone/ghosts vertices.
- more important: no need for higher (than the minimal set)for higher rank tensors and tedious algebraic manipulations

Non-linear gauge implementation

$$\mathcal{L}_{GF} = -\frac{1}{\xi_W}|(\partial_\mu - ie\tilde{\alpha}A_\mu - igc_W\tilde{\beta}Z_\mu)W^{\mu\,+} + \xi_W\frac{g}{2}(v+\tilde{\delta}h+\tilde{\omega}H+i\tilde{\rho}A^0+i\tilde{\kappa}G^0)G^+|^2$$
$$-\frac{1}{2\xi_Z}(\partial.Z+\xi_Z\frac{g}{2c_W}(v+\tilde{\epsilon}h+\tilde{\gamma}H)G^0)^2 - \frac{1}{2\xi_\gamma}(\partial.A)^2$$



p_1 (μ)	p_2 (ν)	p_3 (ρ)	
W^-	W^+	A	$e\Big[g^{\mu\nu}(p_1-p_2)^\rho$ $+(1+\tilde{\alpha}/\xi_W)(p_3^\nu g^{\mu\rho}-p_3^\mu g^{\nu\rho})$ $+(1-\tilde{\alpha}/\xi_W)(p_2^\mu g^{\nu\rho}-p_1^\nu g^{\mu\rho})\Big]$
W^-	W^+	Z	$e\frac{c_W}{s_W}\Big[g^{\mu\nu}(p_1-p_2)^\rho$ $+(1+\tilde{\beta}/\xi_W)(p_3^\nu g^{\mu\rho}-p_3^\mu g^{\nu\rho})$ $+(1-\tilde{\beta}/\xi_W)(p_2^\mu g^{\nu\rho}-p_1^\nu g^{\mu\rho})\Big]$

not your usual VVV gauge vertex!

SloopS: SUSY renormalisation at one-loop

- Default: on-shell, GI, renormalisation in **ALL** sectors

SloopS: SUSY renormalisation at one-loop

- Default: on-shell, GI, renormalisation in **ALL** sectors
- possibility to switch to other schemes easily (DRbar,..)

SloopS: SUSY renormalisation at one-loop

- Default: on-shell, GI, renormalisation in **ALL** sectors
- possibility to switch to other schemes easily (DRbar,..)
- Mixing and rotations: no renormalisation of diagonalisation matrices! ditto gauge-fixing

SloopS: SUSY renormalisation at one-loop

- Default: on-shell, GI, renormalisation in **ALL** sectors
- possibility to switch to other schemes easily (DRbar,..)
- Mixing and rotations: no renormalisation of diagonalisation matrices! ditto gauge-fixing
- Wave-function renormalisation to get rid of all mixing between *physical fields* when on-shell.

SloopS: SUSY renormalisation at one-loop

- Default: on-shell, GI, renormalisation in **ALL** sectors
- possibility to switch to other schemes easily (DRbar,..)
- Mixing and rotations: no renormalisation of diagonalisation matrices! ditto gauge-fixing
- Wave-function renormalisation to get rid of all mixing between *physical fields* when on-shell.
- Issues with definition of $\tan\beta$, many defs not gauge invariant!

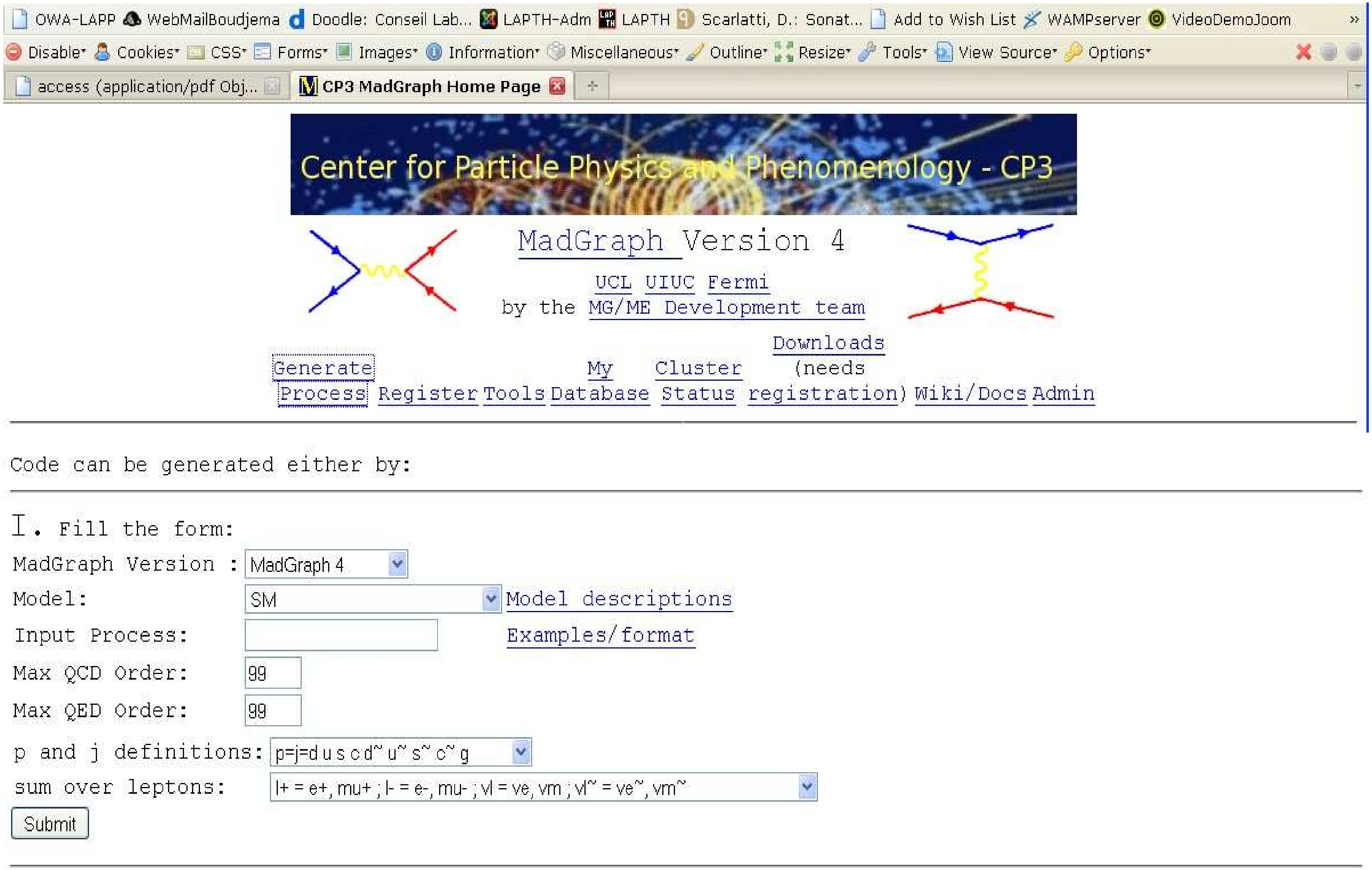
SloopS: SUSY renormalisation at one-loop

- Default: on-shell, GI, renormalisation in **ALL** sectors
- possibility to switch to other schemes easily (DRbar,..)
- Mixing and rotations: no renormalisation of diagonalisation matrices! ditto gauge-fixing
- Wave-function renormalisation to get rid of all mixing between *physical fields* when on-shell.
- Issues with definition of $\tan\beta$, many defs not gauge invariant!
- Same for mixing angle in the sfermion sector.

SloopS: SUSY renormalisation at one-loop

- Default: on-shell, GI, renormalisation in **ALL** sectors
- possibility to switch to other schemes easily (DRbar,..)
- Mixing and rotations: no renormalisation of diagonalisation matrices! ditto gauge-fixing
- Wave-function renormalisation to get rid of all mixing between *physical fields* when on-shell.
- Issues with definition of $\tan\beta$, many defs not gauge invariant!
- Same for mixing angle in the sfermion sector.
- Good scale dependence of ren. csts.

Madgraph Web Interface, you've had a tutorial already



Center for Particle Physics and Phenomenology - CP3

MadGraph Version 4

UCL UIUC Fermi
by the MG/ME Development team

Generate Process | Register | Tools | My Database | Cluster Status | Downloads (needs registration) | Wiki/Docs | Admin

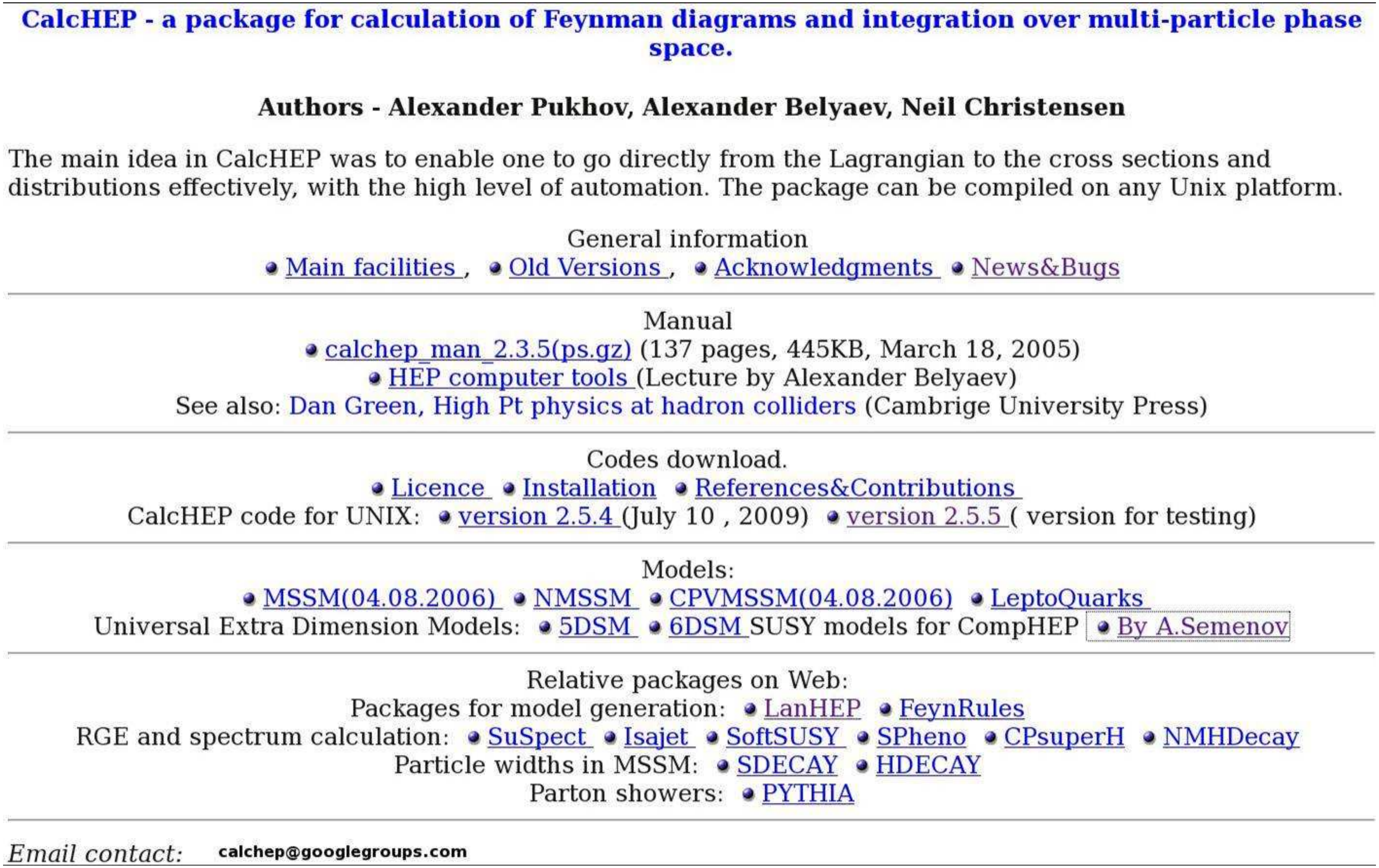
Code can be generated either by:

I. Fill the form:

MadGraph Version :	MadGraph 4	
Model:	SM	Model descriptions
Input Process:		Examples/format
Max QCD Order:	99	
Max QED Order:	99	
p and j definitions:	p=j=d u s c d~ u~ s~ c~ g	
sum over leptons:	l+ = e+, mu+ ; l- = e-, mu- ; vl = ve, vm ; vl~ = ve~, vm~	

Submit

CalcHEP webpage



CalcHEP - a package for calculation of Feynman diagrams and integration over multi-particle phase space.

Authors - Alexander Pukhov, Alexander Belyaev, Neil Christensen

The main idea in CalcHEP was to enable one to go directly from the Lagrangian to the cross sections and distributions effectively, with the high level of automation. The package can be compiled on any Unix platform.

General information
● Main facilities , ● Old Versions , ● Acknowledgments ● News&Bugs

Manual
● calchep_man_2.3.5(ps.gz) (137 pages, 445KB, March 18, 2005)
● HEP computer tools (Lecture by Alexander Belyaev)
See also: Dan Green, High Pt physics at hadron colliders (Cambrige University Press)

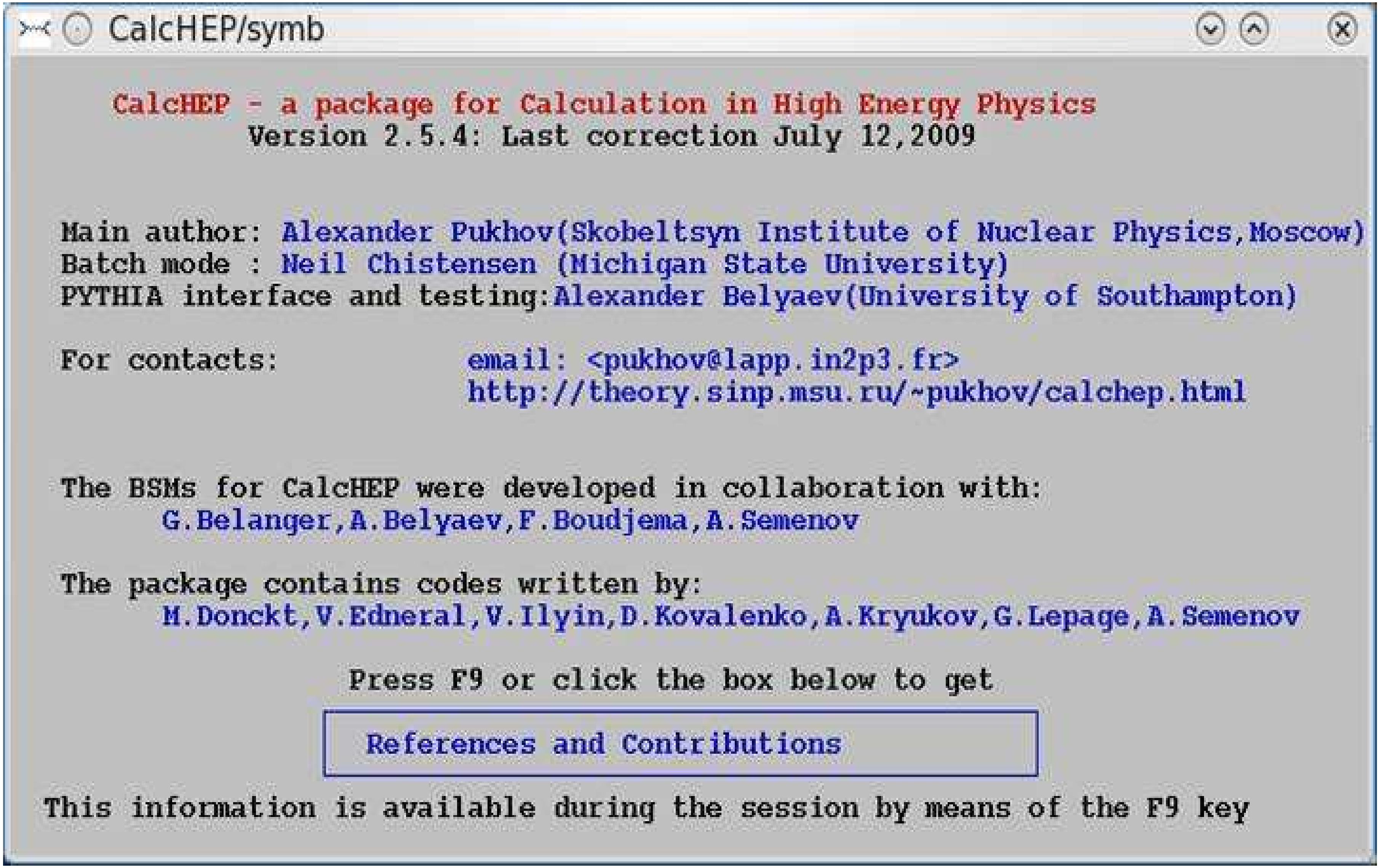
Codes download.
● Licence ● Installation ● References&Contributions
CalcHEP code for UNIX: ● version 2.5.4 (July 10 , 2009) ● version 2.5.5 (version for testing)

Models:
● MSSM(04.08.2006) ● NMSSM ● CPVMSSM(04.08.2006) ● LeptoQuarks
Universal Extra Dimension Models: ● 5DSM ● 6DSM SUSY models for CompHEP ● By A.Semenov

Relative packages on Web:
Packages for model generation: ● LanHEP ● FeynRules
RGE and spectrum calculation: ● SuSpect ● Isajet ● SoftSUSY ● SPheno ● CPsuperH ● NMHDecay
Particle widths in MSSM: ● SDECAY ● HDECAY
Parton showers: ● PYTHIA

Email contact: **calchep@googlegroups.com**

CalcHEP start page



CalcHEP - a package for Calculation in High Energy Physics
Version 2.5.4: Last correction July 12,2009

Main author: Alexander Pukhov(Skobeltsyn Institute of Nuclear Physics,Moscow)
Batch mode : Neil Chistensen (Michigan State University)
PYTHIA interface and testing:Alexander Belyaev(University of Southampton)

For contacts: email: <pukhov@lapp.in2p3.fr>
http://theory.sinp.msu.ru/~pukhov/calchep.html

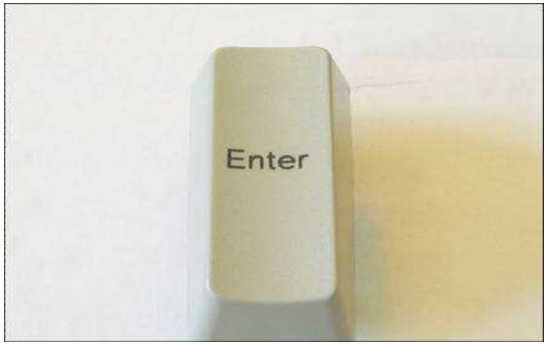
The BSMs for CalcHEP were developed in collaboration with:
G.Belanger,A.Belyaev,F.Boudjema,A.Semenov

The package contains codes written by:
M.Donckt,V.Edneral,V.Ilyin,D.Kovalenko,A.Kryukov,G.Lepage,A.Semenov

Press F9 or click the box below to get

This information is available during the session by means of the F9 key

In interactive mode you only need to know/press 3 keys

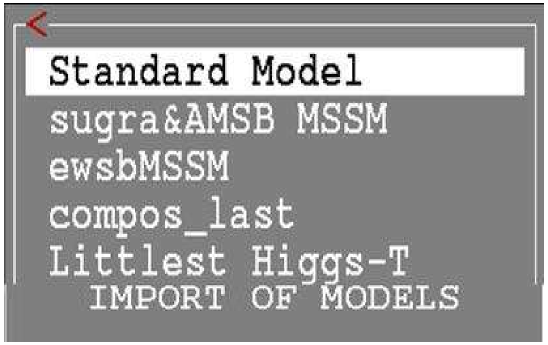


Enter menu selection
(forward)

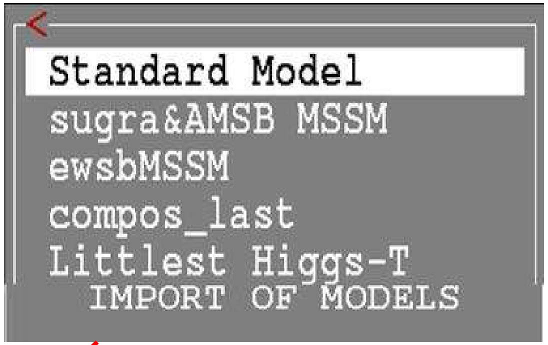
Exit menu selection
(back)

Help !
(or details of menu choice)

Work flow

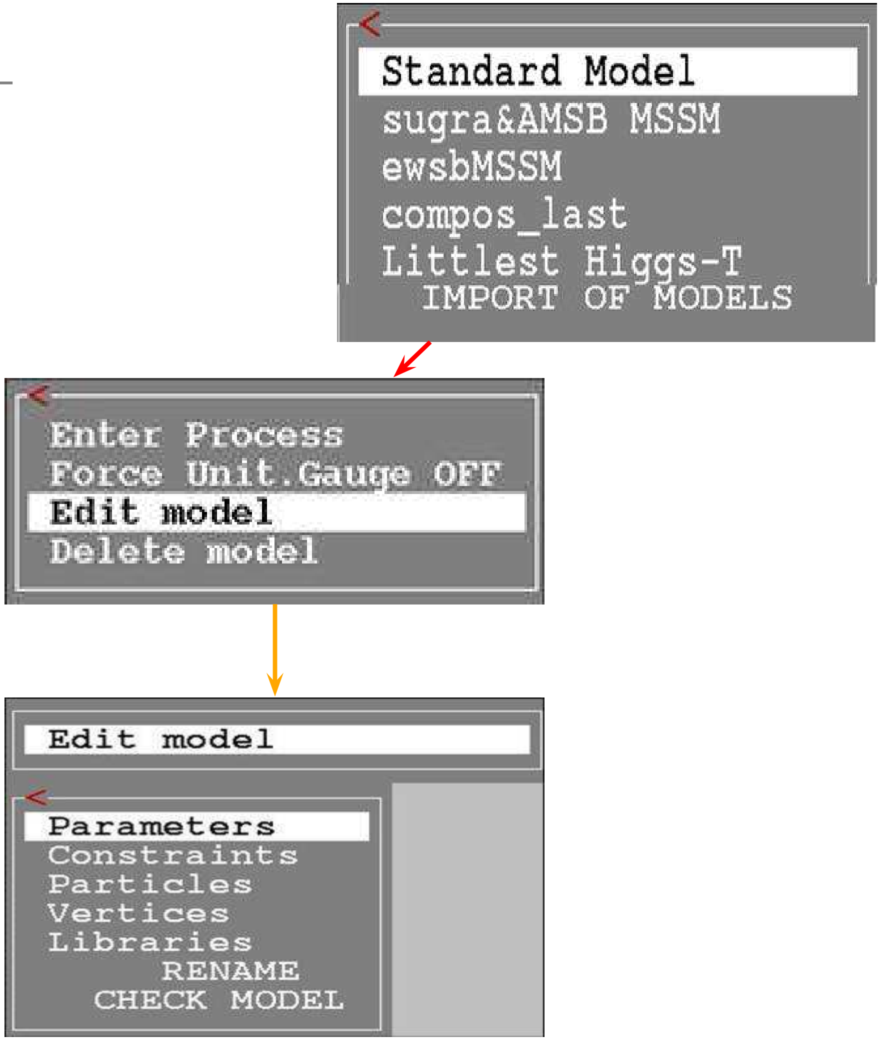

Standard Model
sugra&AMSB MSSM
ewsbMSSM
compos_last
Littlest Higgs-T
IMPORT OF MODELS

Work flow

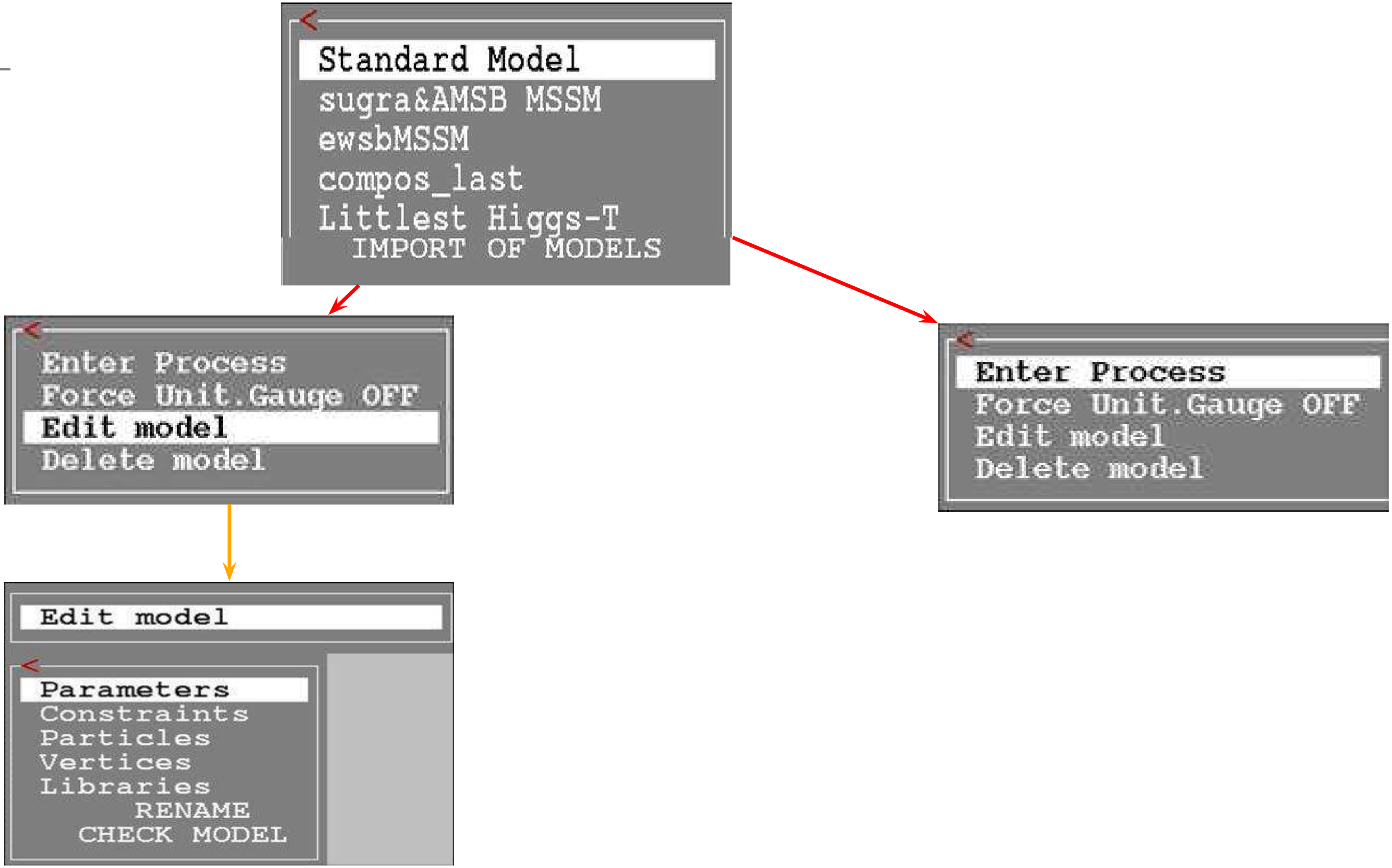

Standard Model
sugra&AMSB MSSM
ewsbMSSM
compos_last
Littlest Higgs-T
IMPORT OF MODELS


Enter Process
Force Unit.Gauge OFF
Edit model
Delete model

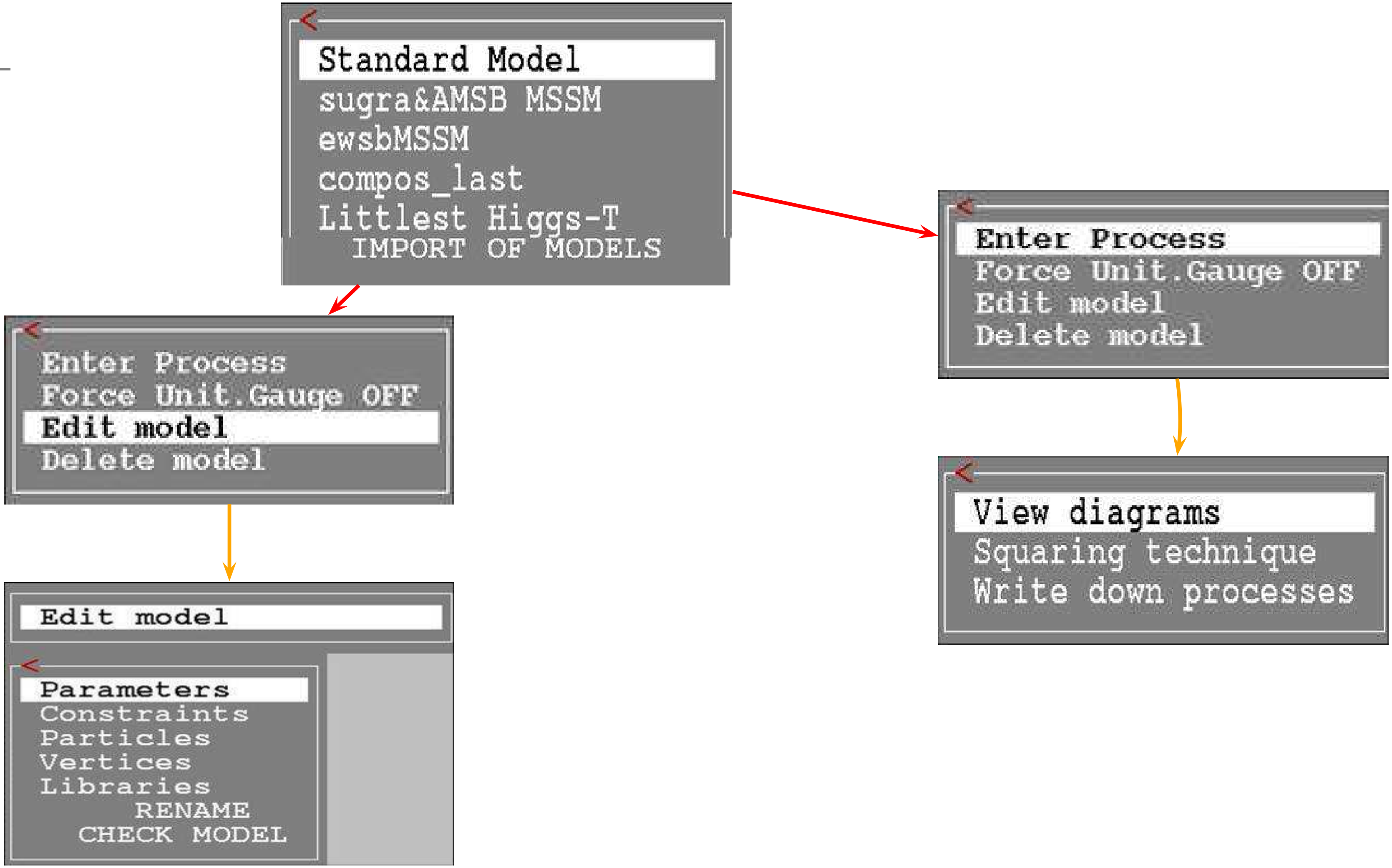
Work flow


Standard Model
sugra&AMSB MSSM
ewsbMSSM
compos_last
Littlest Higgs-T
IMPORT OF MODELS
Enter Process
Force Unit.Gauge OFF
Edit model
Delete model
Edit model
Parameters
Constraints
Particles
Vertices
Libraries
RENAME
CHECK MODEL

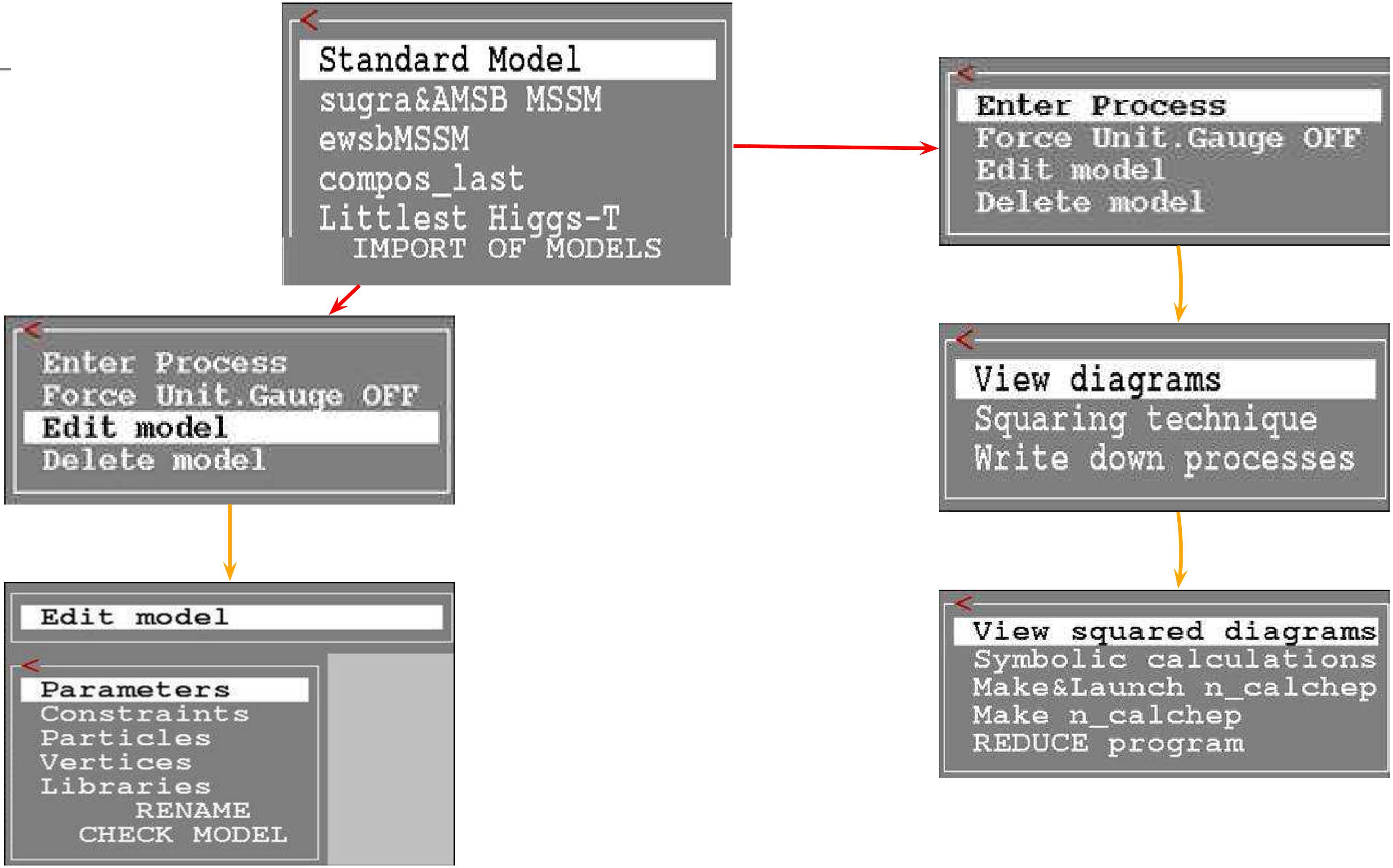
Work flow


Standard Model
sugra&AMSB MSSM
ewsbMSSM
compos_last
Littlest Higgs-T
IMPORT OF MODELS
Enter Process
Force Unit.Gauge OFF
Edit model
Delete model
Enter Process
Force Unit.Gauge OFF
Edit model
Delete model
Edit model
Parameters
Constraints
Particles
Vertices
Libraries
RENAME
CHECK MODEL

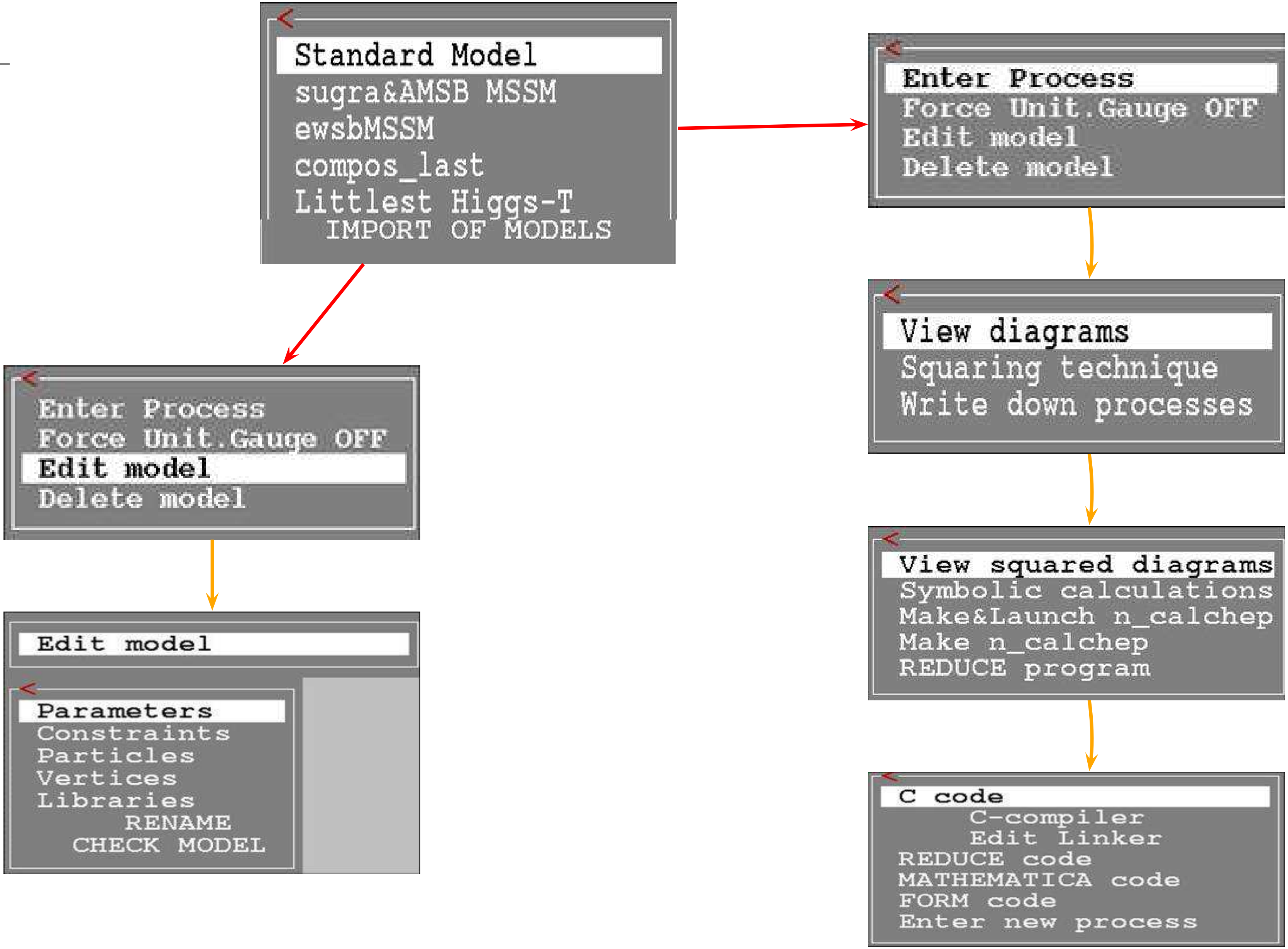
Work flow

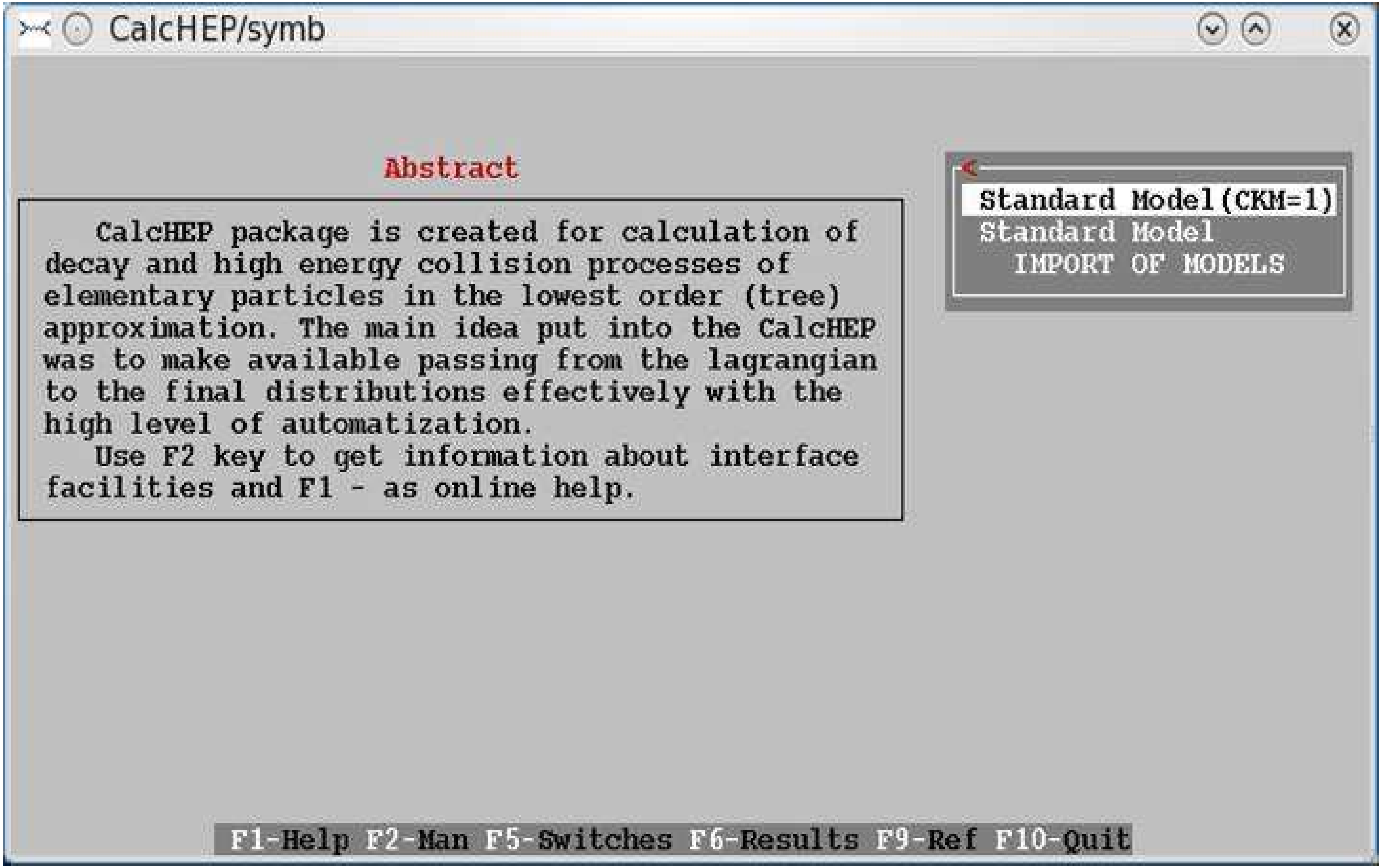

Standard Model
sugra&AMSB MSSM
ewsbMSSM
compos_last
Littlest Higgs-T
IMPORT OF MODELS
Enter Process
Force Unit.Gauge OFF
Edit model
Delete model
Enter Process
Force Unit.Gauge OFF
Edit model
Delete model
View diagrams
Squaring technique
Write down processes
Edit model
Parameters
Constraints
Particles
Vertices
Libraries
RENAME
CHECK MODEL

Work flow


Standard Model
sugra&AMSB MSSM
ewsbMSSM
compos_last
Littlest Higgs-T
IMPORT OF MODELS
Enter Process
Force Unit.Gauge OFF
Edit model
Delete model
Enter Process
Force Unit.Gauge OFF
Edit model
Delete model
View diagrams
Squaring technique
Write down processes
Edit model
Parameters
Constraints
Particles
Vertices
Libraries
RENAME
CHECK MODEL
View squared diagrams
Symbolic calculations
Make&Launch n_calchep
Make n_calchep
REDUCE program

Work flow

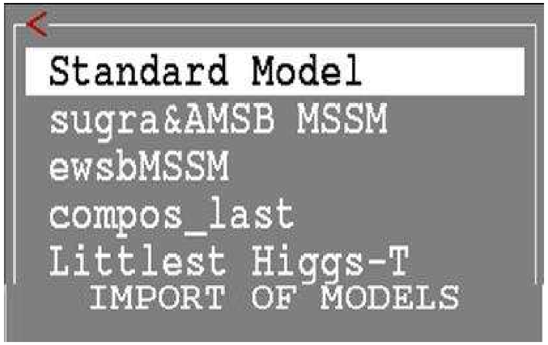

Standard Model
sugra&AMSB MSSM
ewsbMSSM
compos_last
Littlest Higgs-T
IMPORT OF MODELS
Enter Process
Force Unit.Gauge OFF
Edit model
Delete model
Edit model
Parameters
Constraints
Particles
Vertices
Libraries
RENAME
CHECK MODEL
Enter Process
Force Unit.Gauge OFF
Edit model
Delete model
View diagrams
Squaring technique
Write down processes
View squared diagrams
Symbolic calculations
Make&Launch n_calchep
Make n_calchep
REDUCE program
C code
C-compiler
Edit Linker
REDUCE code
MATHEMATICA code
FORM code
Enter new process

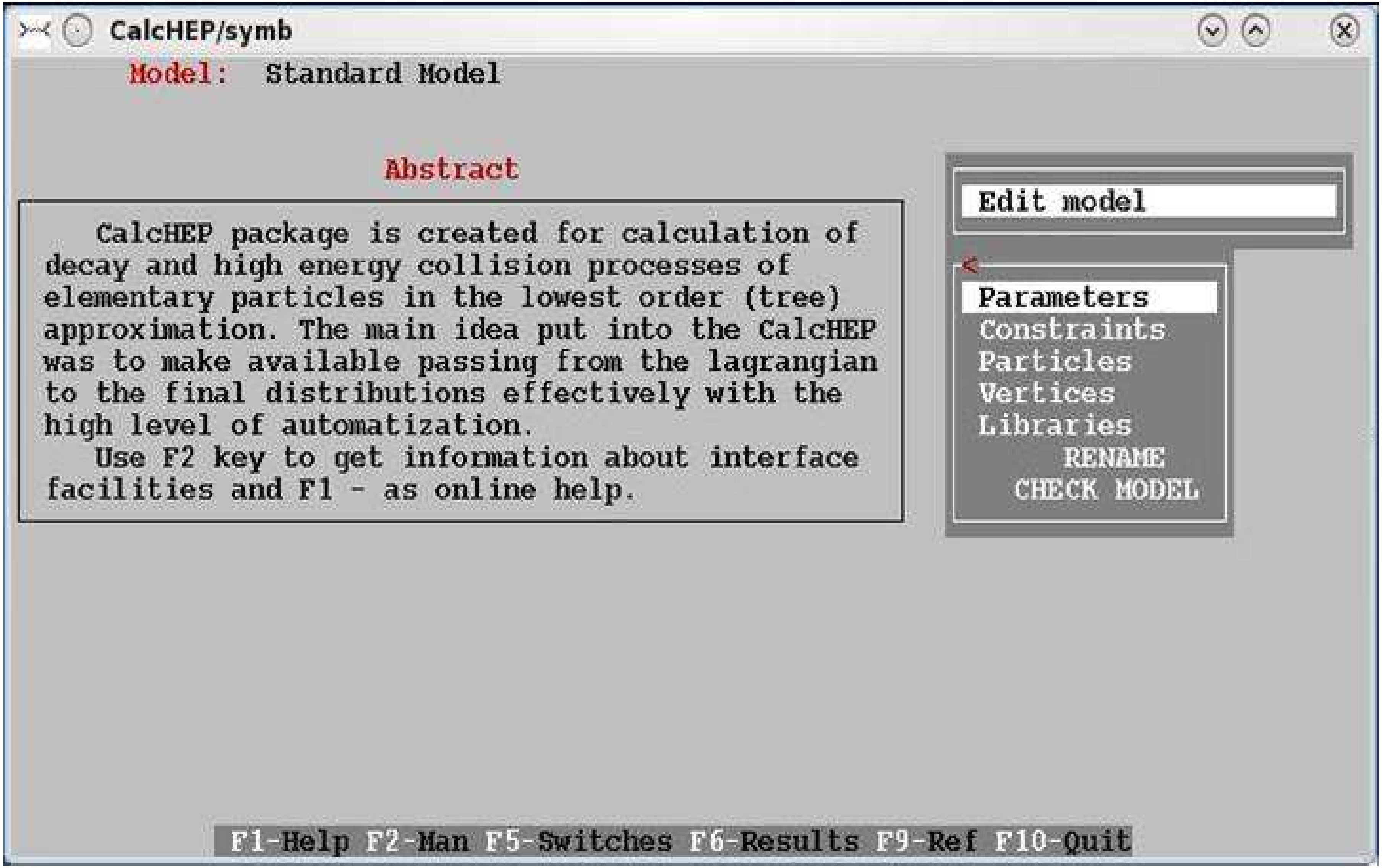


Abstract

CalcHEP package is created for calculation of decay and high energy collision processes of elementary particles in the lowest order (tree) approximation. The main idea put into the CalcHEP was to make available passing from the lagrangian to the final distributions effectively with the high level of automatization.

Use F2 key to get information about interface facilities and F1 - as online help.


Standard Model
sugra&AMSB MSSM
ewsbMSSM
compos_last
Littlest Higgs-T
IMPORT OF MODELS

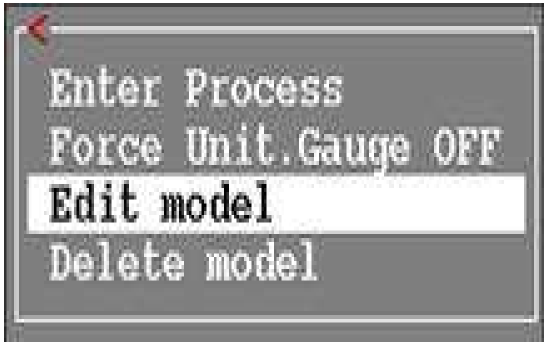


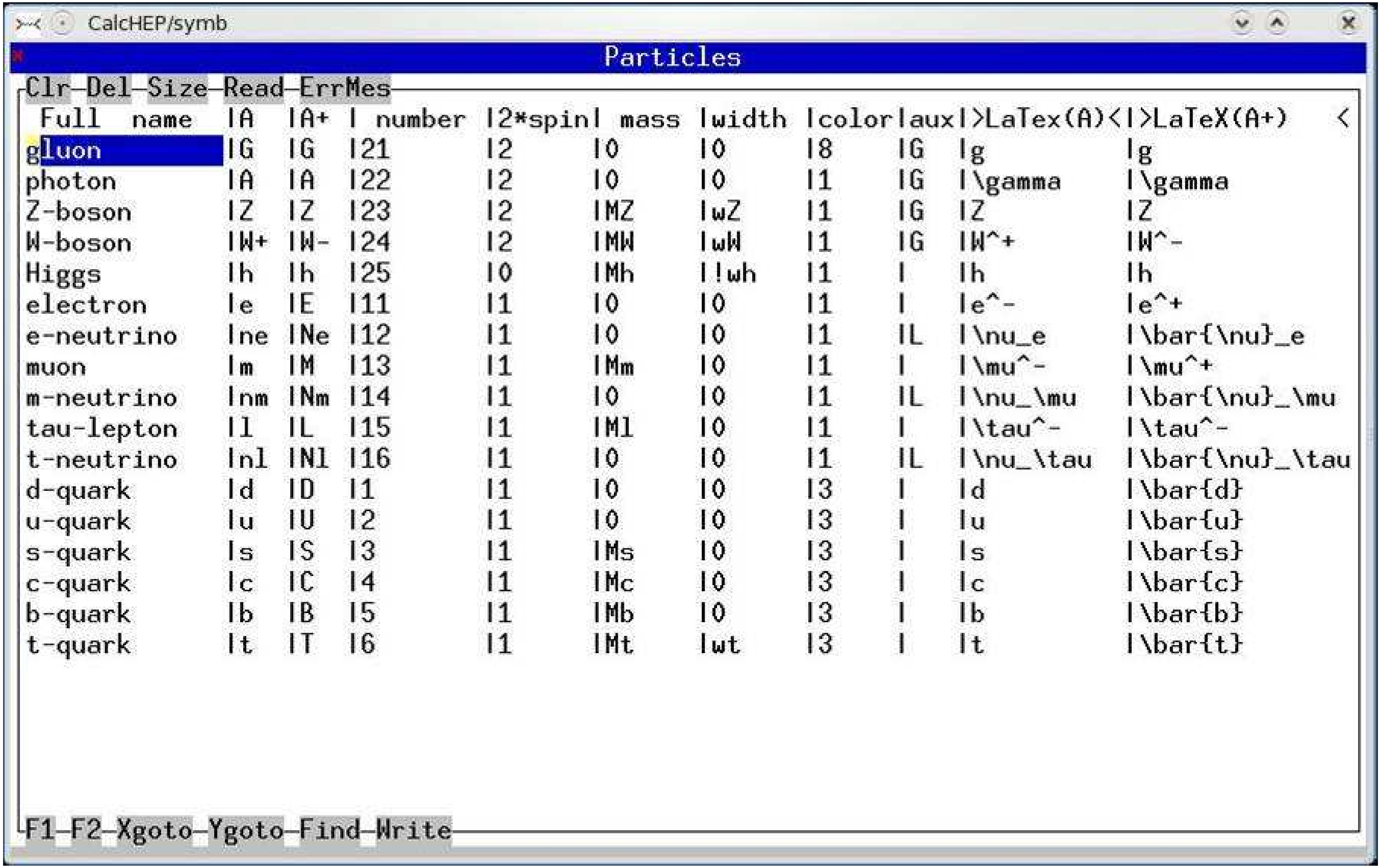
Model: Standard Model

Abstract

CalcHEP package is created for calculation of decay and high energy collision processes of elementary particles in the lowest order (tree) approximation. The main idea put into the CalcHEP was to make available passing from the lagrangian to the final distributions effectively with the high level of automatization.

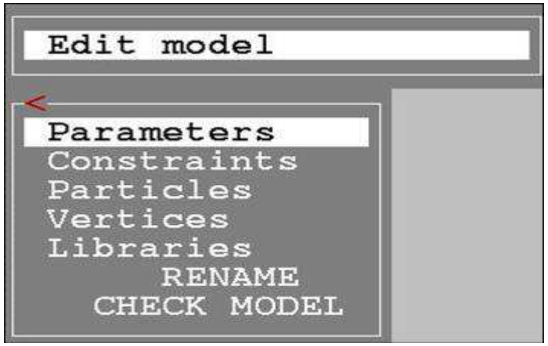
Use F2 key to get information about interface facilities and F1 - as online help.

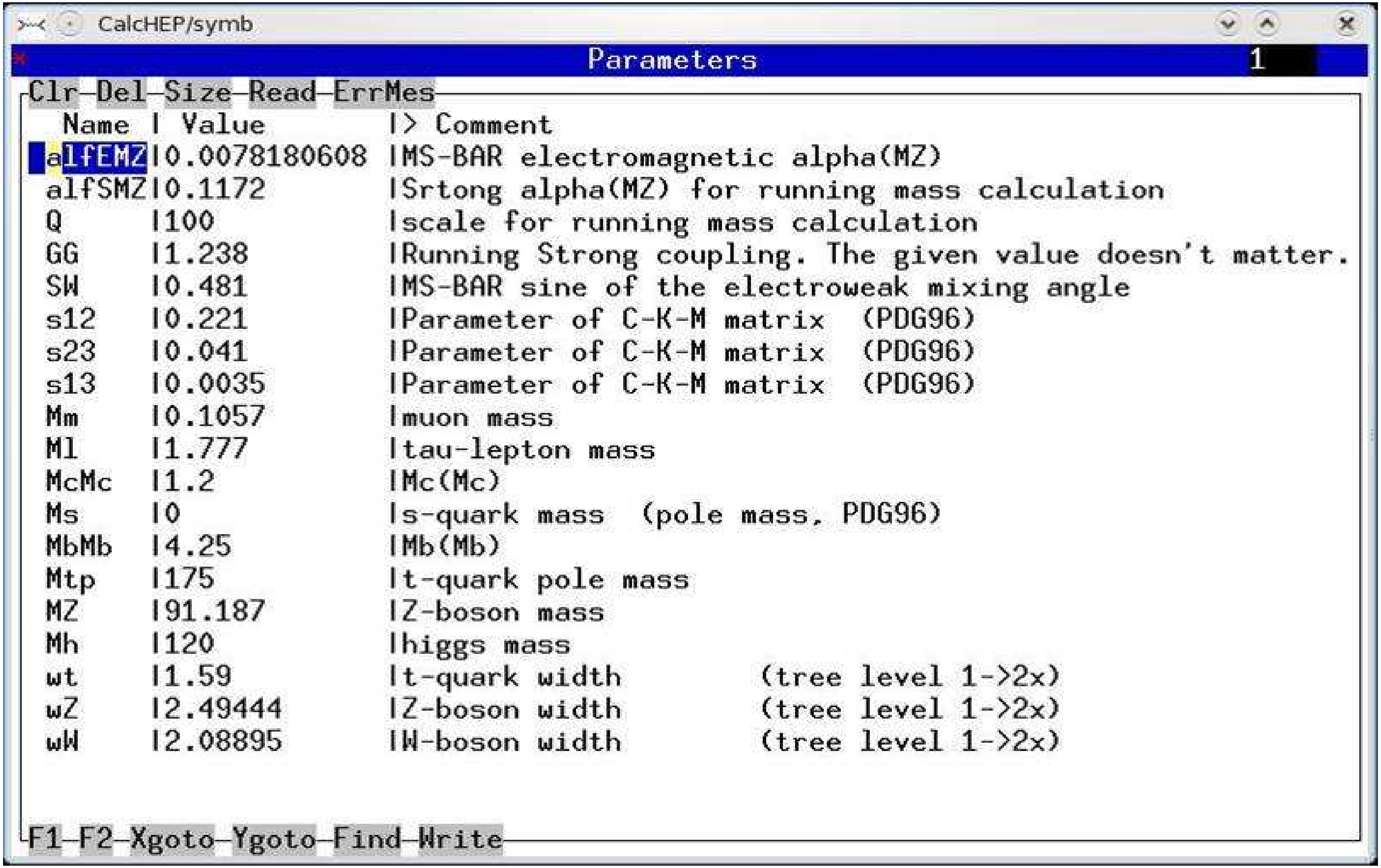

Enter Process
Force Unit.Gauge OFF
Edit model
Delete model



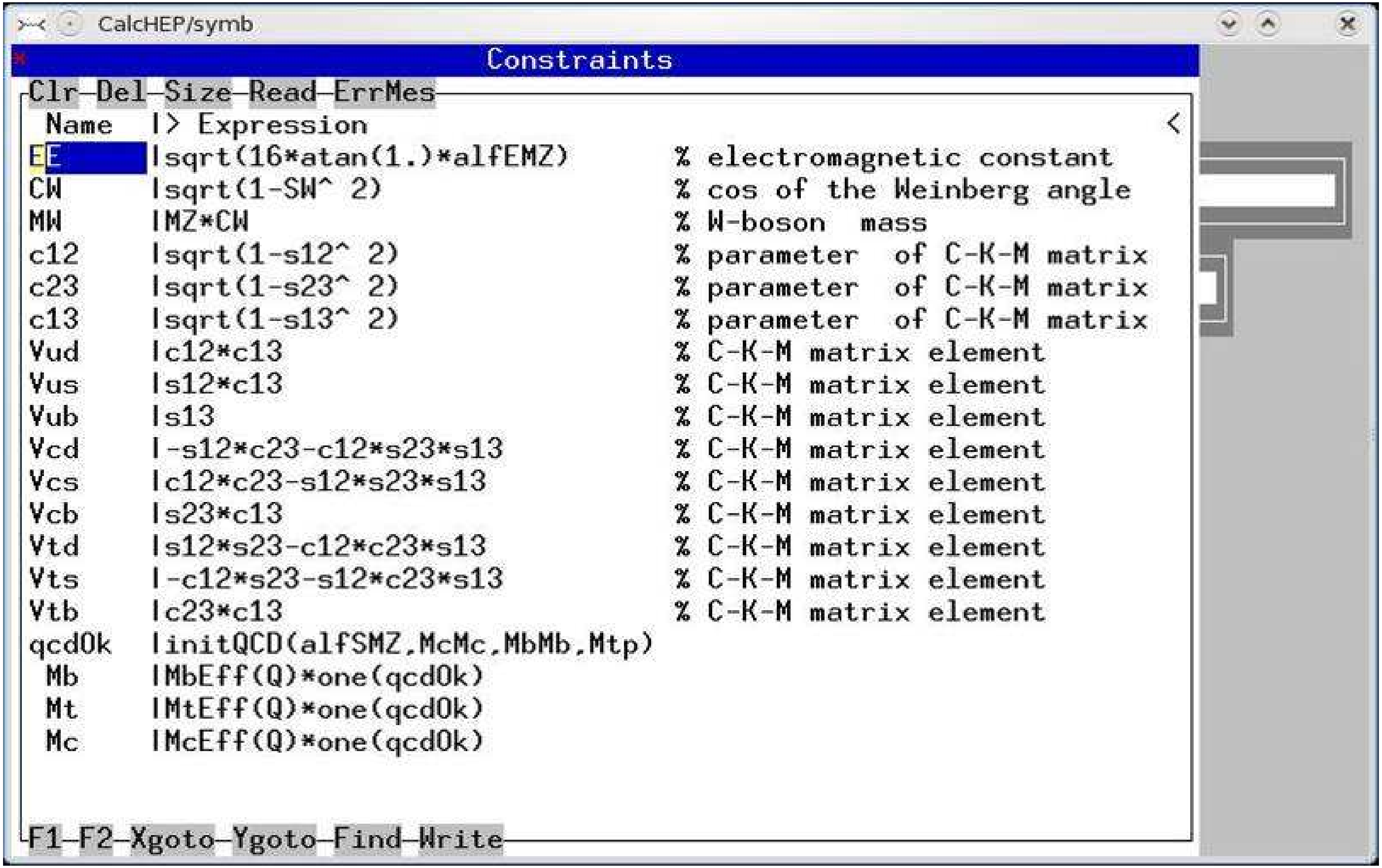
CalcHEP/symb

```
                                   Particles
Clr-Del-Size-Read-ErrMes
 Full    name  |A   |A+  | number |2*spin| mass |width |color|aux|>LaTex(A)<|>LaTeX(A+)   <
gluon          |G   |G   |21      |2     |0     |0     |8    |G  |g         |g
photon         |A   |A   |22      |2     |0     |0     |1    |G  |\gamma    |\gamma
Z-boson        |Z   |Z   |23      |2     |MZ    |wZ    |1    |G  |Z         |Z
W-boson        |W+  |W-  |24      |2     |MW    |wW    |1    |G  |W^+       |W^-
Higgs          |h   |h   |25      |0     |Mh    |!wh   |1    |   |h         |h
electron       |e   |E   |11      |1     |0     |0     |1    |   |e^-       |e^+
e-neutrino     |ne  |Ne  |12      |1     |0     |0     |1    |L  |\nu_e     |\bar{\nu}_e
muon           |m   |M   |13      |1     |Mm    |0     |1    |   |\mu^-     |\mu^+
m-neutrino     |nm  |Nm  |14      |1     |0     |0     |1    |L  |\nu_\mu   |\bar{\nu}_\mu
tau-lepton     |l   |L   |15      |1     |Ml    |0     |1    |   |\tau^-    |\tau^-
t-neutrino     |nl  |Nl  |16      |1     |0     |0     |1    |L  |\nu_\tau  |\bar{\nu}_\tau
d-quark        |d   |D   |1       |1     |0     |0     |3    |   |d         |\bar{d}
u-quark        |u   |U   |2       |1     |0     |0     |3    |   |u         |\bar{u}
s-quark        |s   |S   |3       |1     |Ms    |0     |3    |   |s         |\bar{s}
c-quark        |c   |C   |4       |1     |Mc    |0     |3    |   |c         |\bar{c}
b-quark        |b   |B   |5       |1     |Mb    |0     |3    |   |b         |\bar{b}
t-quark        |t   |T   |6       |1     |Mt    |wt    |3    |   |t         |\bar{t}
F1-F2-Xgoto-Ygoto-Find-Write
```

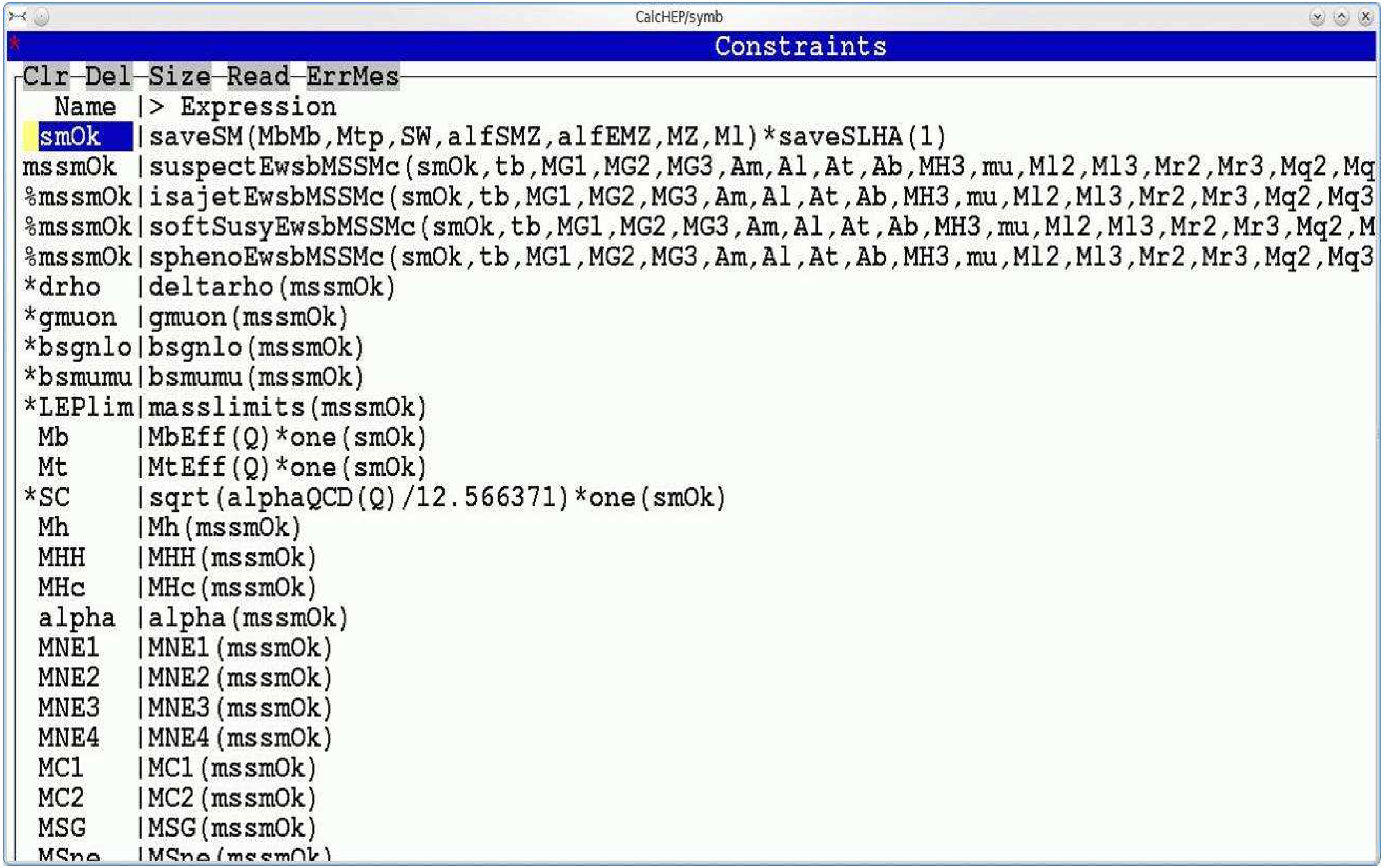

Edit model
<
Parameters
Constraints
Particles
Vertices
Libraries
RENAME
CHECK MODEL



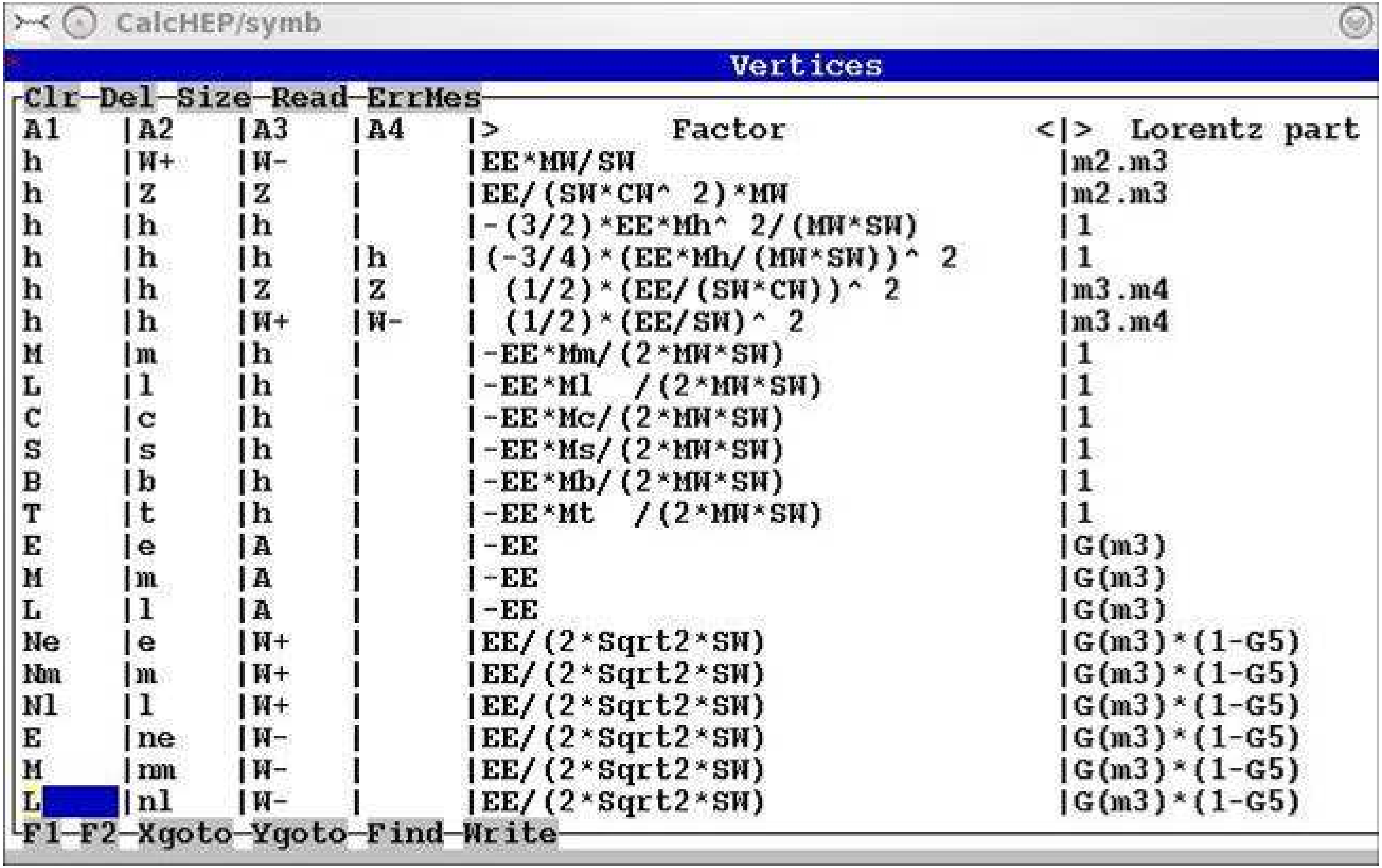
```
  Name  | Value        |> Comment
 alfEMZ|0.0078180608 |MS-BAR electromagnetic alpha(MZ)
 alfSMZ|0.1172       |Srtong alpha(MZ) for running mass calculation
 Q     |100          |scale for running mass calculation
 GG    |1.238        |Running Strong coupling. The given value doesn't matter.
 SW    |0.481        |MS-BAR sine of the electroweak mixing angle
 s12   |0.221        |Parameter of C-K-M matrix  (PDG96)
 s23   |0.041        |Parameter of C-K-M matrix  (PDG96)
 s13   |0.0035       |Parameter of C-K-M matrix  (PDG96)
 Mm    |0.1057       |muon mass
 Ml    |1.777        |tau-lepton mass
 McMc  |1.2          |Mc(Mc)
 Ms    |0            |s-quark mass  (pole mass, PDG96)
 MbMb  |4.25         |Mb(Mb)
 Mtp   |175          |t-quark pole mass
 MZ    |91.187       |Z-boson mass
 Mh    |120          |higgs mass
 wt    |1.59         |t-quark width         (tree level 1->2x)
 wZ    |2.49444      |Z-boson width         (tree level 1->2x)
 wW    |2.08895      |W-boson width         (tree level 1->2x)
```



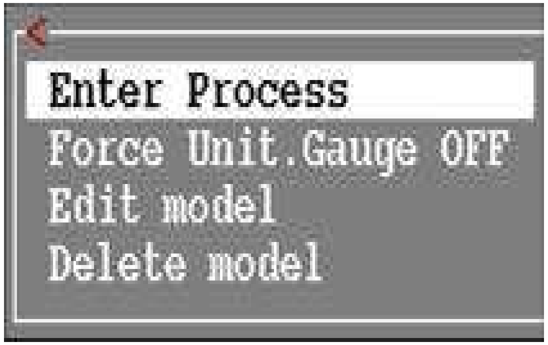
```
 Name   |> Expression
EE      |sqrt(16*atan(1.)*alfEMZ)         % electromagnetic constant
CW      |sqrt(1-SW^ 2)                    % cos of the Weinberg angle
MW      |MZ*CW                            % W-boson  mass
c12     |sqrt(1-s12^ 2)                   % parameter  of C-K-M matrix
c23     |sqrt(1-s23^ 2)                   % parameter  of C-K-M matrix
c13     |sqrt(1-s13^ 2)                   % parameter  of C-K-M matrix
Vud     |c12*c13                          % C-K-M matrix element
Vus     |s12*c13                          % C-K-M matrix element
Vub     |s13                              % C-K-M matrix element
Vcd     |-s12*c23-c12*s23*s13             % C-K-M matrix element
Vcs     |c12*c23-s12*s23*s13              % C-K-M matrix element
Vcb     |s23*c13                          % C-K-M matrix element
Vtd     |s12*s23-c12*c23*s13              % C-K-M matrix element
Vts     |-c12*s23-s12*c23*s13             % C-K-M matrix element
Vtb     |c23*c13                          % C-K-M matrix element
qcdOk   |initQCD(alfSMZ,McMc,MbMb,Mtp)
 Mb     |MbEff(Q)*one(qcdOk)
 Mt     |MtEff(Q)*one(qcdOk)
 Mc     |McEff(Q)*one(qcdOk)
```

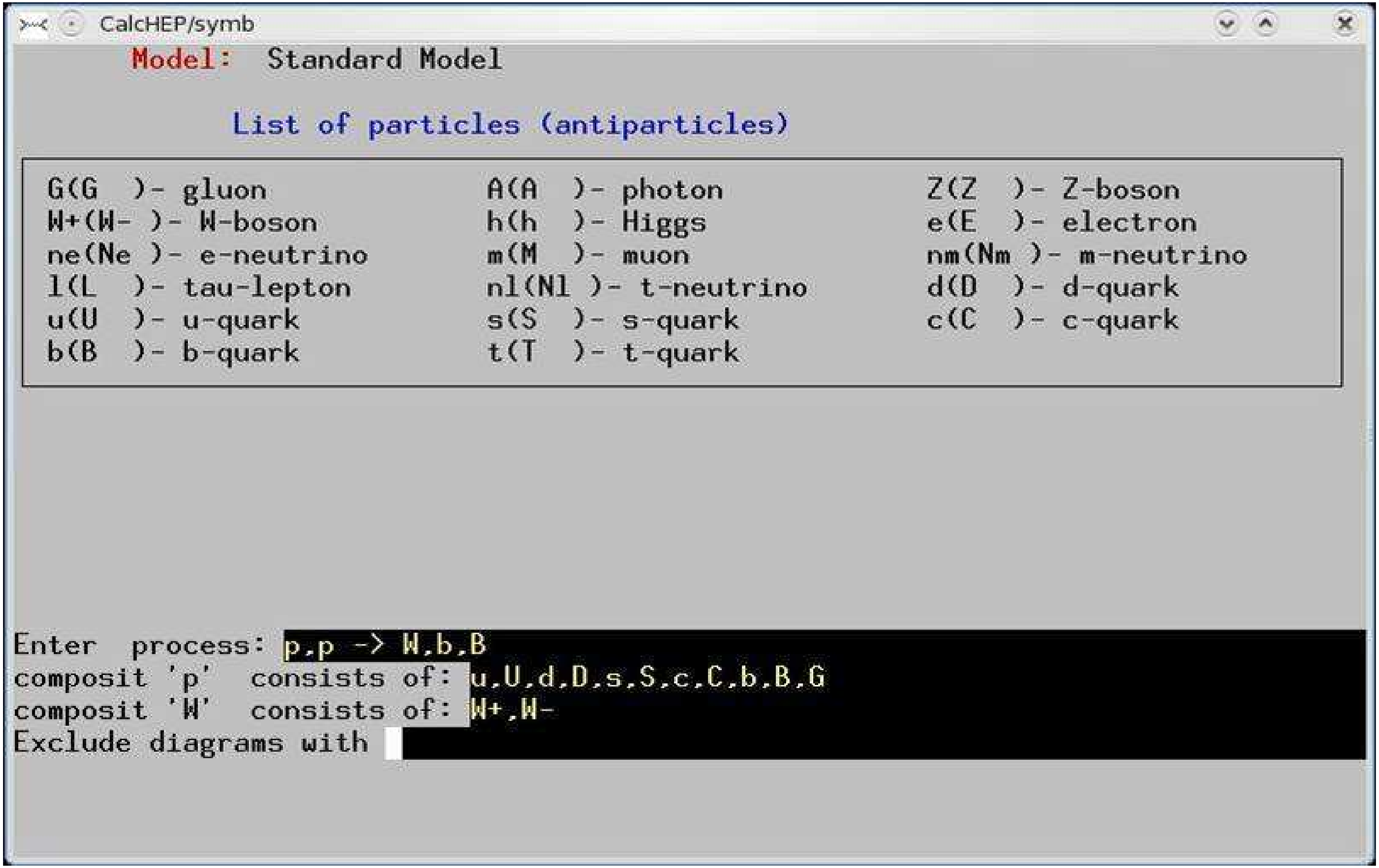


```
                         Constraints
 Name  |> Expression
smOk   |saveSM(MbMb,Mtp,SW,alfSMZ,alfEMZ,MZ,Ml)*saveSLHA(1)
mssmOk |suspectEwsbMSSMc(smOk,tb,MG1,MG2,MG3,Am,Al,At,Ab,MH3,mu,Ml2,Ml3,Mr2,Mr3,Mq2,Mq
%mssmOk|isajetEwsbMSSMc(smOk,tb,MG1,MG2,MG3,Am,Al,At,Ab,MH3,mu,Ml2,Ml3,Mr2,Mr3,Mq2,Mq3
%mssmOk|softSusyEwsbMSSMc(smOk,tb,MG1,MG2,MG3,Am,Al,At,Ab,MH3,mu,Ml2,Ml3,Mr2,Mr3,Mq2,M
%mssmOk|sphenoEwsbMSSMc(smOk,tb,MG1,MG2,MG3,Am,Al,At,Ab,MH3,mu,Ml2,Ml3,Mr2,Mr3,Mq2,Mq3
*drho  |deltarho(mssmOk)
*gmuon |gmuon(mssmOk)
*bsgnlo|bsgnlo(mssmOk)
*bsmumu|bsmumu(mssmOk)
*LEPlim|masslimits(mssmOk)
 Mb    |MbEff(Q)*one(smOk)
 Mt    |MtEff(Q)*one(smOk)
*SC    |sqrt(alphaQCD(Q)/12.566371)*one(smOk)
 Mh    |Mh(mssmOk)
 MHH   |MHH(mssmOk)
 MHc   |MHc(mssmOk)
 alpha |alpha(mssmOk)
 MNE1  |MNE1(mssmOk)
 MNE2  |MNE2(mssmOk)
 MNE3  |MNE3(mssmOk)
 MNE4  |MNE4(mssmOk)
 MC1   |MC1(mssmOk)
 MC2   |MC2(mssmOk)
 MSG   |MSG(mssmOk)
 MSne  |MSne(mssmOk)
```

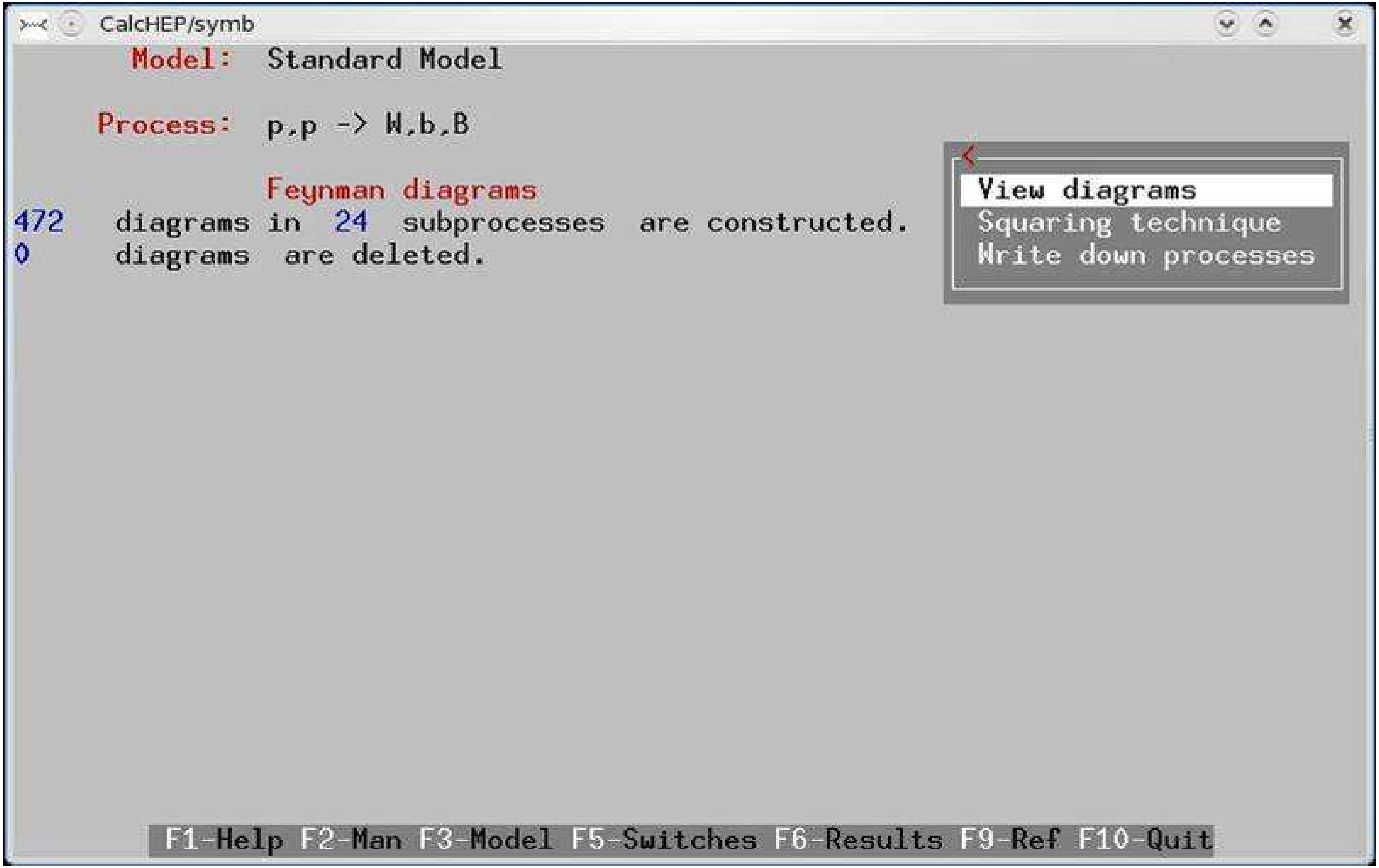


```
A1     |A2    |A3    |A4    |>           Factor                <|> Lorentz part
h      |W+    |W-    |      |EE*MW/SW                            |m2.m3
h      |Z     |Z     |      |EE/(SW*CW^ 2)*MW                    |m2.m3
h      |h     |h     |      |-(3/2)*EE*Mh^ 2/(MW*SW)             |1
h      |h     |h     |h     |(-3/4)*(EE*Mh/(MW*SW))^ 2           |1
h      |h     |Z     |Z     | (1/2)*(EE/(SW*CW))^ 2              |m3.m4
h      |h     |W+    |W-    | (1/2)*(EE/SW)^ 2                   |m3.m4
M      |m     |h     |      |-EE*Mm/(2*MW*SW)                    |1
L      |l     |h     |      |-EE*Ml  /(2*MW*SW)                  |1
C      |c     |h     |      |-EE*Mc/(2*MW*SW)                    |1
S      |s     |h     |      |-EE*Ms/(2*MW*SW)                    |1
B      |b     |h     |      |-EE*Mb/(2*MW*SW)                    |1
T      |t     |h     |      |-EE*Mt  /(2*MW*SW)                  |1
E      |e     |A     |      |-EE                                 |G(m3)
M      |m     |A     |      |-EE                                 |G(m3)
L      |l     |A     |      |-EE                                 |G(m3)
Ne     |e     |W+    |      |EE/(2*Sqrt2*SW)                     |G(m3)*(1-G5)
Nm     |m     |W+    |      |EE/(2*Sqrt2*SW)                     |G(m3)*(1-G5)
Nl     |l     |W+    |      |EE/(2*Sqrt2*SW)                     |G(m3)*(1-G5)
E      |ne    |W-    |      |EE/(2*Sqrt2*SW)                     |G(m3)*(1-G5)
M      |nm    |W-    |      |EE/(2*Sqrt2*SW)                     |G(m3)*(1-G5)
L      |nl    |W-    |      |EE/(2*Sqrt2*SW)                     |G(m3)*(1-G5)
```

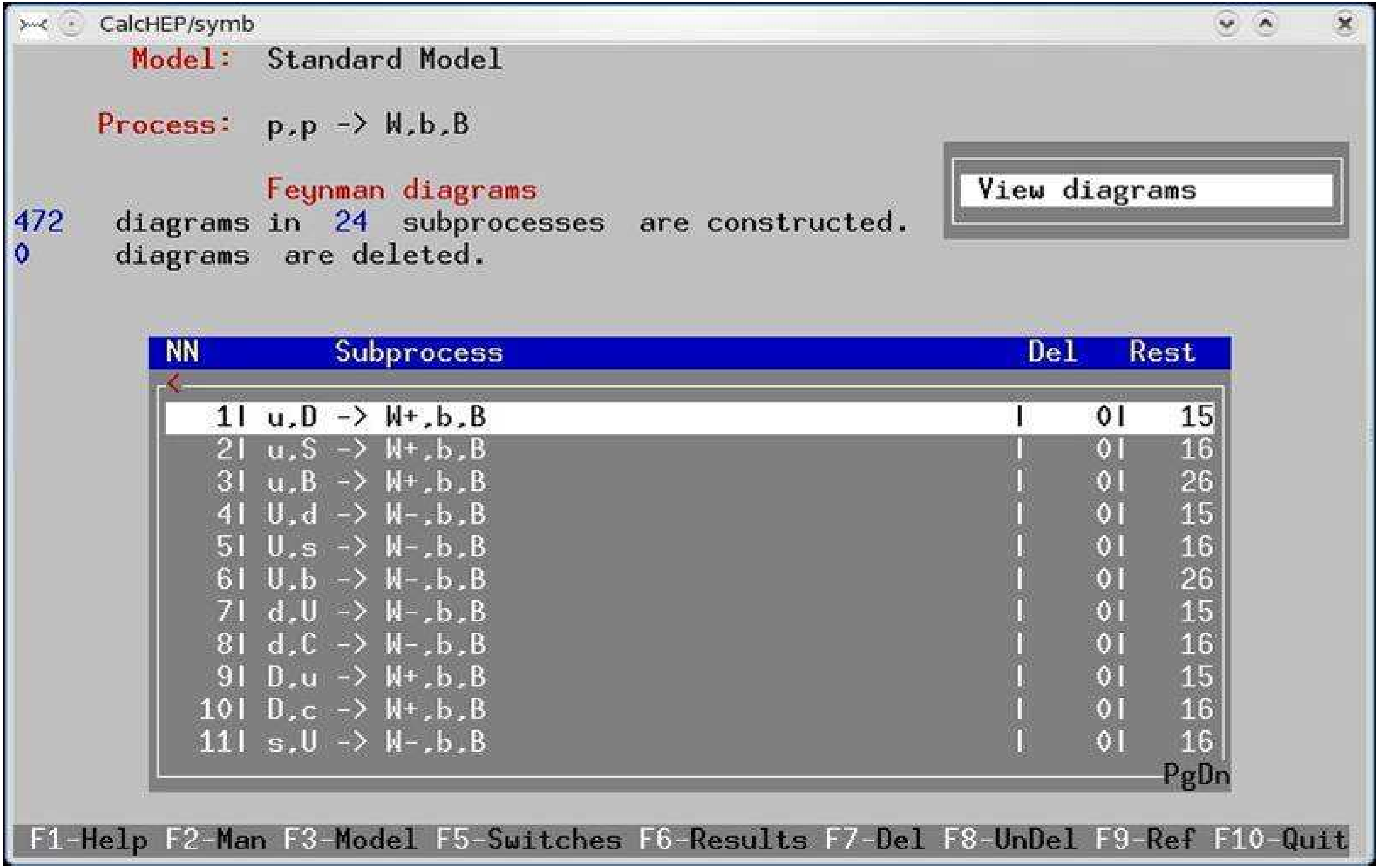

Enter Process
Force Unit.Gauge OFF
Edit model
Delete model



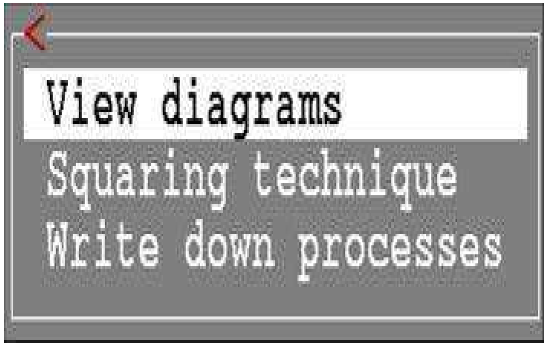
```
      Model:  Standard Model

            List of particles (antiparticles)

  G(G  )- gluon          A(A  )- photon        Z(Z  )- Z-boson
  W+(W- )- W-boson       h(h  )- Higgs         e(E  )- electron
  ne(Ne )- e-neutrino    m(M  )- muon          nm(Nm )- m-neutrino
  l(L  )- tau-lepton     nl(Nl )- t-neutrino   d(D  )- d-quark
  u(U  )- u-quark        s(S  )- s-quark       c(C  )- c-quark
  b(B  )- b-quark        t(T  )- t-quark

Enter  process: p,p -> W,b,B
composit 'p'  consists of: u,U,d,D,s,S,c,C,b,B,G
composit 'W'  consists of: W+,W-
Exclude diagrams with
```

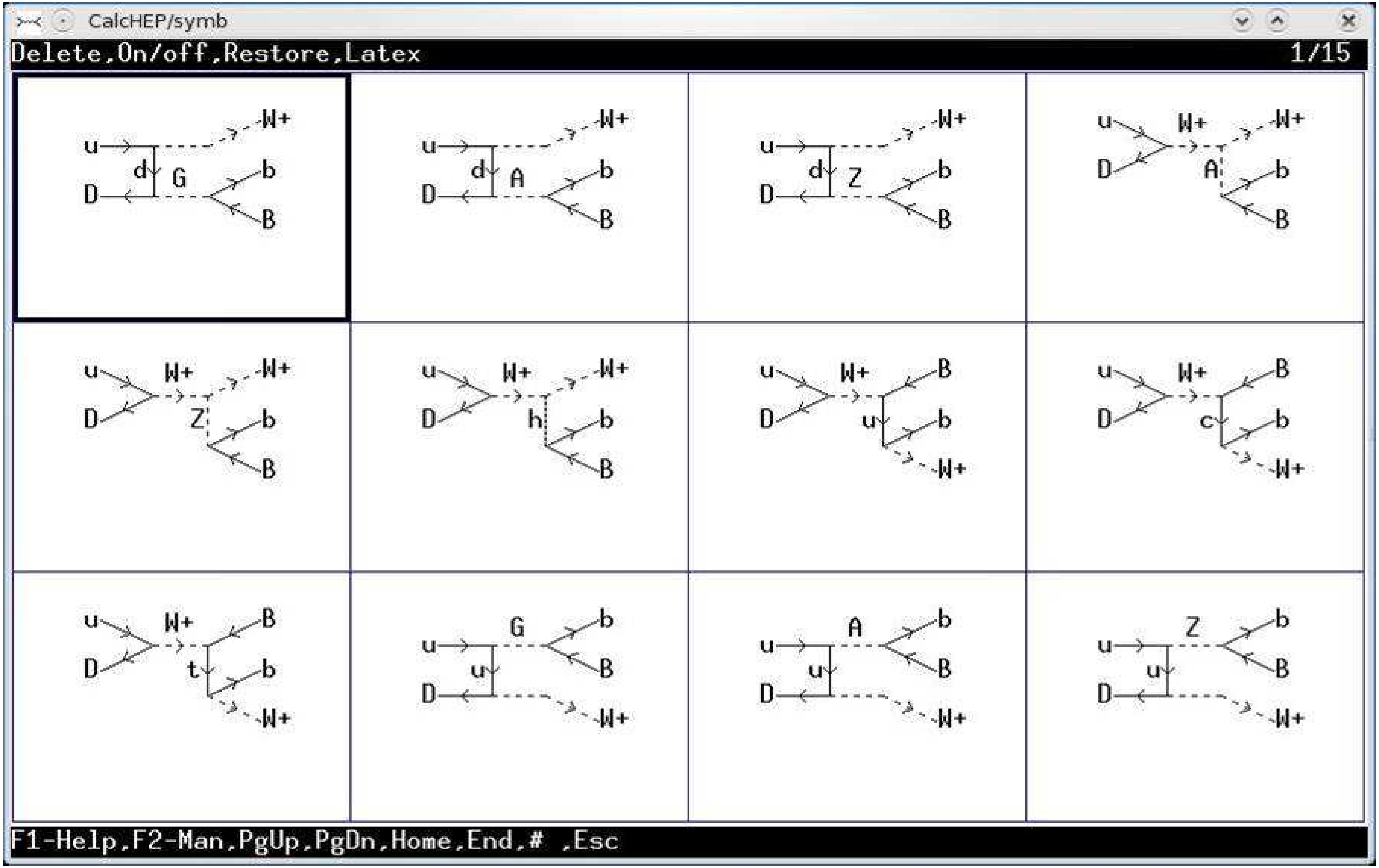


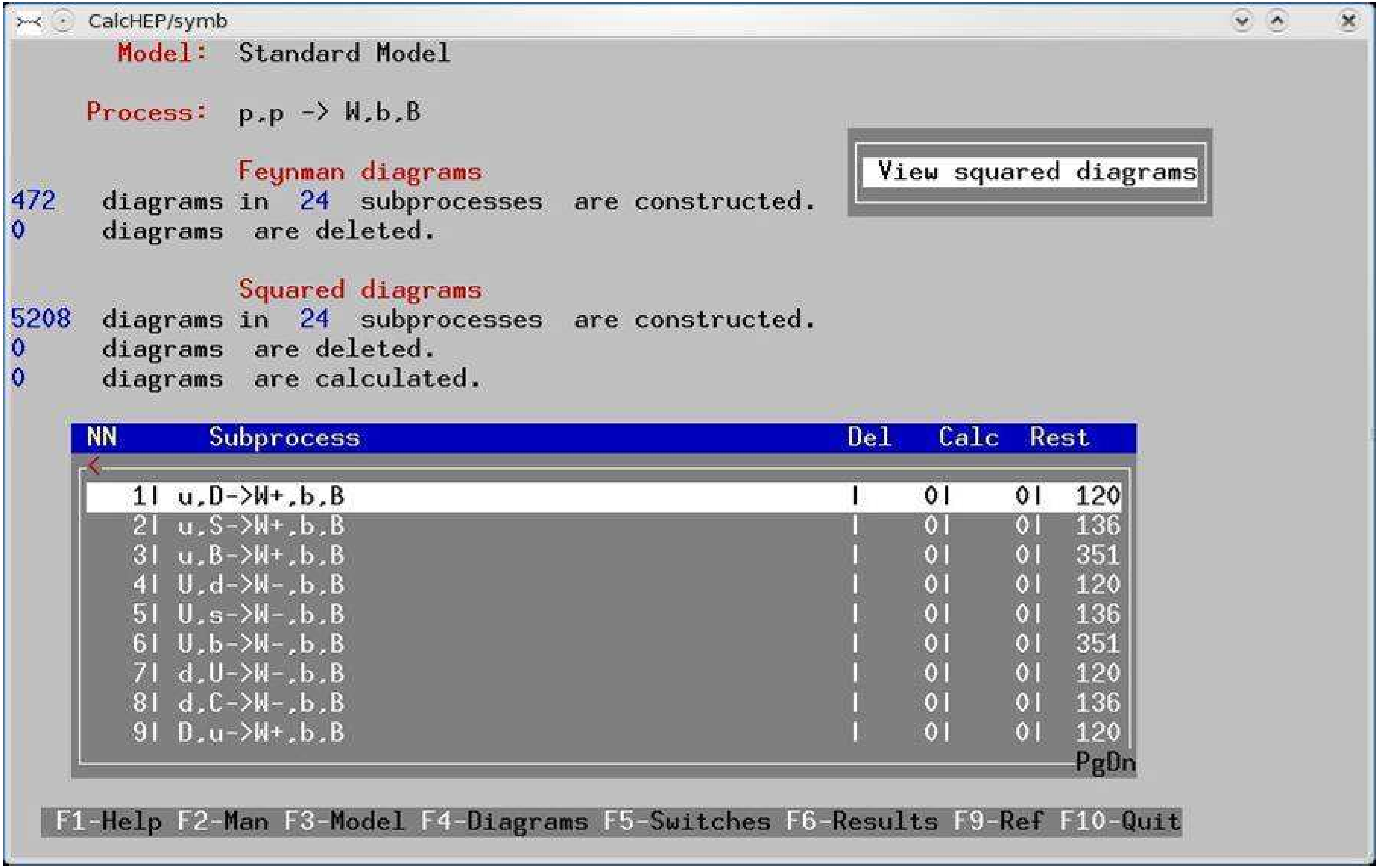
```
     Model:  Standard Model

   Process:  p,p -> W,b,B

             Feynman diagrams
472   diagrams in  24  subprocesses  are constructed.
0     diagrams  are deleted.
```



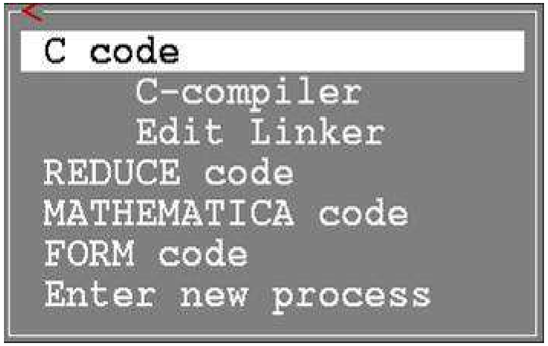
```
     Model:  Standard Model

   Process:  p,p -> W,b,B

             Feynman diagrams
472   diagrams in  24  subprocesses  are constructed.
0     diagrams  are deleted.

        NN           Subprocess                         Del    Rest
         <
           1| u,D -> W+,b,B                          |     0|    15
           2| u,S -> W+,b,B                          |     0|    16
           3| u,B -> W+,b,B                          |     0|    26
           4| U,d -> W-,b,B                          |     0|    15
           5| U,s -> W-,b,B                          |     0|    16
           6| U,b -> W-,b,B                          |     0|    26
           7| d,U -> W-,b,B                          |     0|    15
           8| d,C -> W-,b,B                          |     0|    16
           9| D,u -> W+,b,B                          |     0|    15
          10| D,c -> W+,b,B                          |     0|    16
          11| s,U -> W-,b,B                          |     0|    16
                                                               PgDn
```

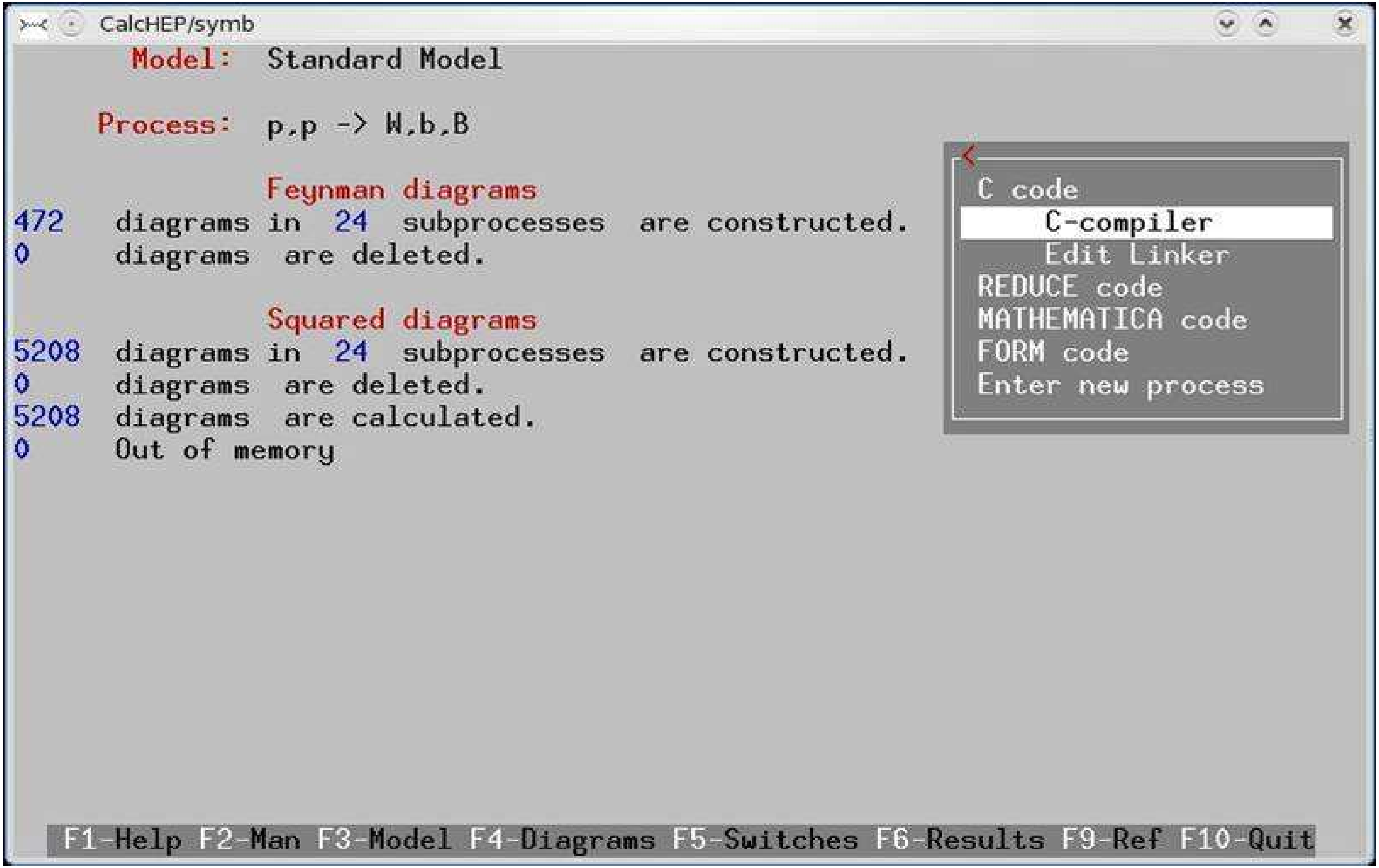

View diagrams
Squaring technique
Write down processes


CalcHEP/symb
Delete,On/off,Restore,Latex
1/15
u D d G W+ b B
u D d A W+ b B
u D d Z W+ b B
u D W+ A W+ b B
u D W+ Z W+ b B
u D W+ h W+ b B
u D W+ u B b W+
u D W+ c B b W+
u D W+ t B b W+
u D u G b B W+
u D u A b B W+
u D u Z b B W+
F1-Help,F2-Man,PgUp,PgDn,Home,End,# ,Esc

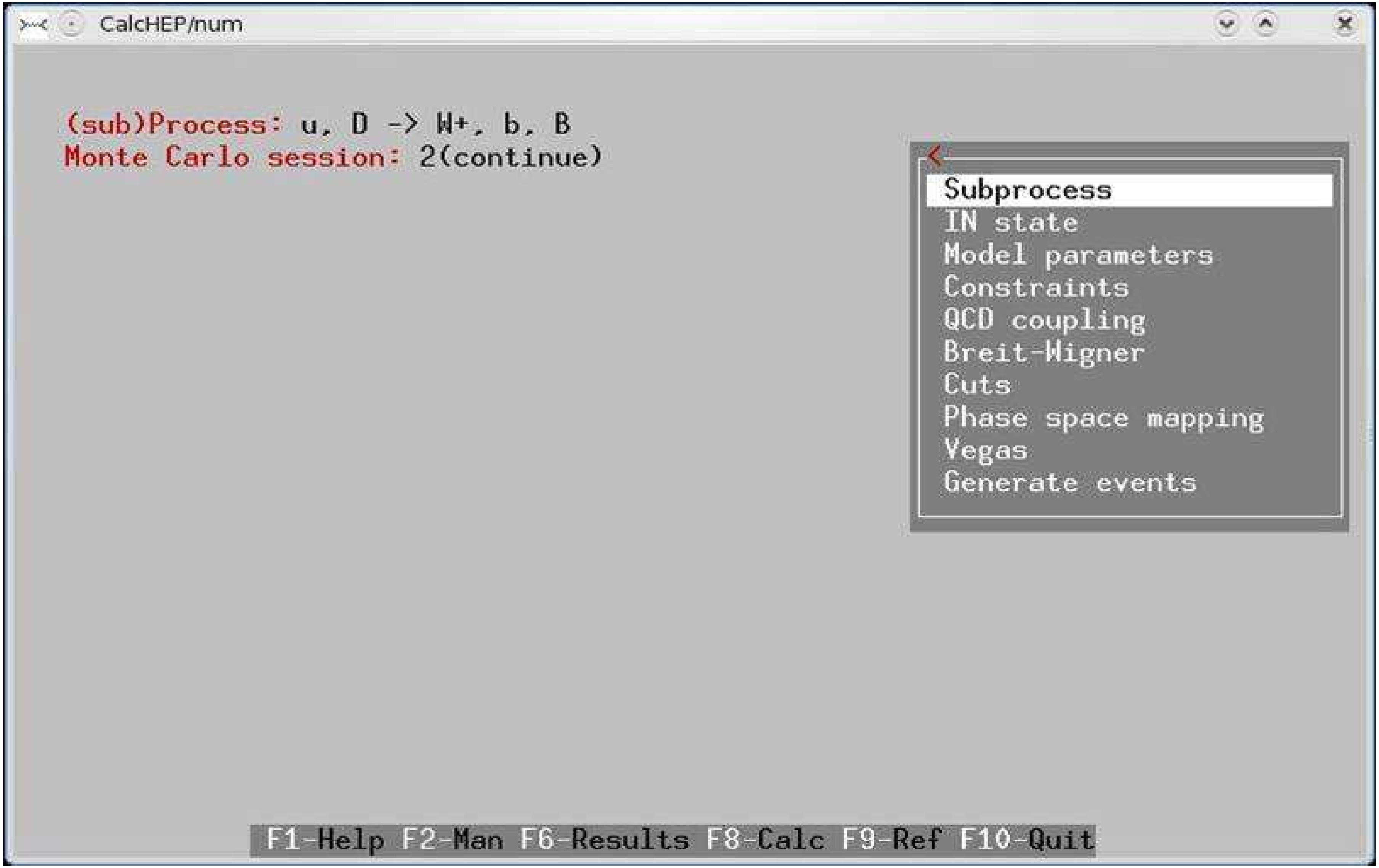


```
     Model:   Standard Model

   Process:   p,p -> W,b,B

              Feynman diagrams
472   diagrams in  24  subprocesses  are constructed.
0     diagrams  are deleted.

              Squared diagrams
5208  diagrams in  24  subprocesses  are constructed.
0     diagrams  are deleted.
0     diagrams  are calculated.

  NN      Subprocess                             Del    Calc   Rest
     1| u,D->W+,b,B                             |     0|     0|  120
     2| u,S->W+,b,B                             |     0|     0|  136
     3| u,B->W+,b,B                             |     0|     0|  351
     4| U,d->W-,b,B                             |     0|     0|  120
     5| U,s->W-,b,B                             |     0|     0|  136
     6| U,b->W-,b,B                             |     0|     0|  351
     7| d,U->W-,b,B                             |     0|     0|  120
     8| d,C->W-,b,B                             |     0|     0|  136
     9| D,u->W+,b,B                             |     0|     0|  120
                                                                PgDn
```

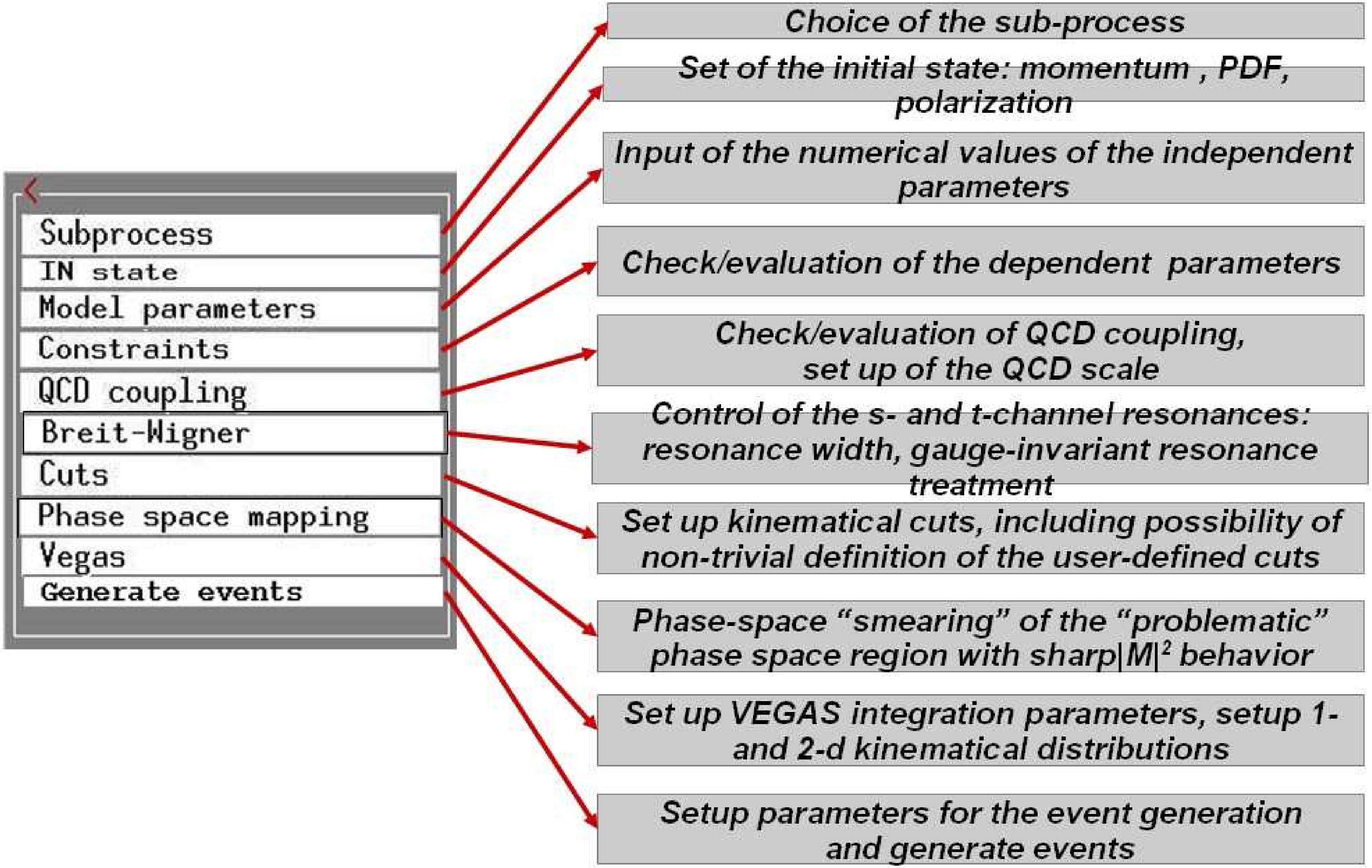

C code
C-compiler
Edit Linker
REDUCE code
MATHEMATICA code
FORM code
Enter new process



```
      Model:  Standard Model

    Process:  p,p -> W,b,B

              Feynman diagrams
472    diagrams in  24  subprocesses  are constructed.
0      diagrams  are deleted.

              Squared diagrams
5208   diagrams in  24  subprocesses  are constructed.
0      diagrams  are deleted.
5208   diagrams  are calculated.
0      Out of memory
```



```
(sub)Process: u, D -> W+, b, B
Monte Carlo session: 2(continue)
```



Choice of the sub-process

Set of the initial state: momentum , PDF, polarization

Input of the numerical values of the independent parameters

Check/evaluation of the dependent parameters

Check/evaluation of QCD coupling, set up of the QCD scale

Control of the s- and t-channel resonances: resonance width, gauge-invariant resonance treatment

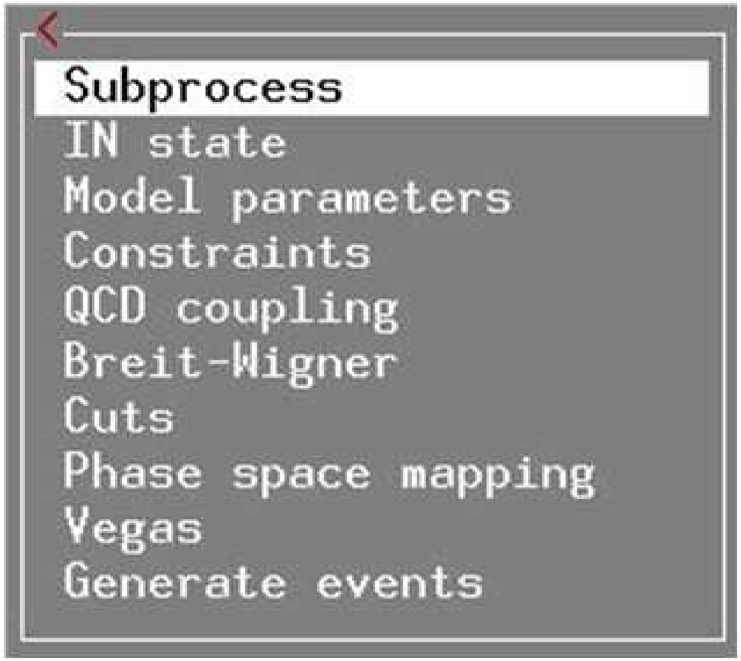
Set up kinematical cuts, including possibility of non-trivial definition of the user-defined cuts

Phase-space “smearing” of the “problematic” phase space region with sharp$|M|^2$ behavior

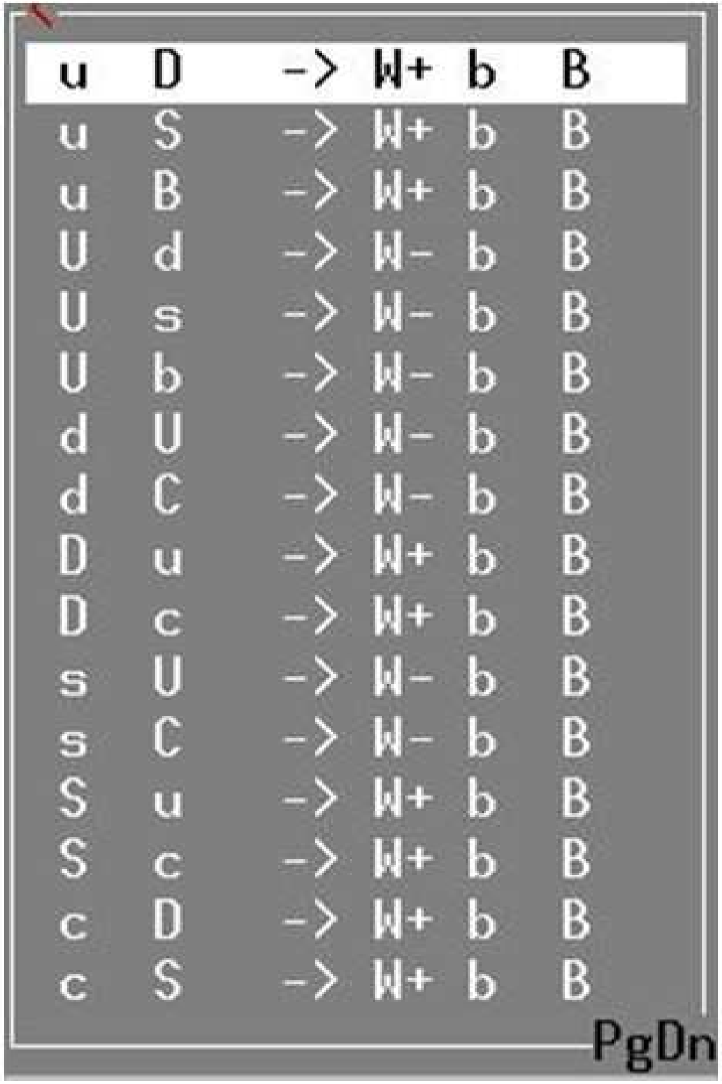
Set up VEGAS integration parameters, setup 1- and 2-d kinematical distributions

Setup parameters for the event generation and generate events

List of sub-processes

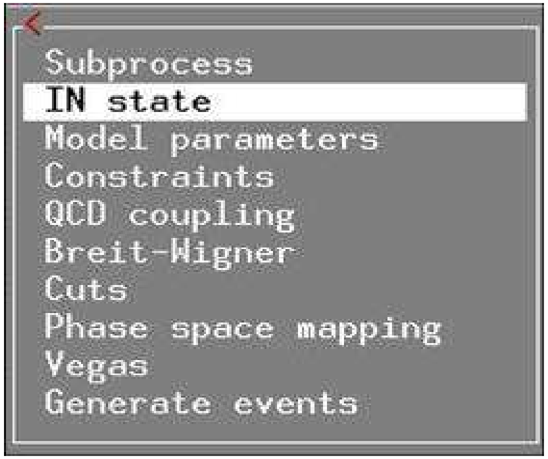


```
Subprocess
IN state
Model parameters
Constraints
QCD coupling
Breit-Wigner
Cuts
Phase space mapping
Vegas
Generate events
```



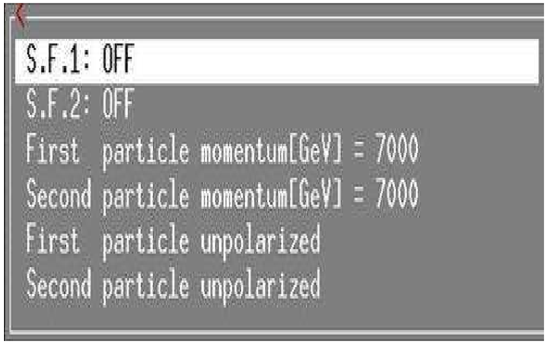
```
u  D   -> W+ b  B
u  S   -> W+ b  B
u  B   -> W+ b  B
U  d   -> W- b  B
U  s   -> W- b  B
U  b   -> W- b  B
d  U   -> W- b  B
d  C   -> W- b  B
D  u   -> W+ b  B
D  c   -> W+ b  B
s  U   -> W- b  B
s  C   -> W- b  B
S  u   -> W+ b  B
S  c   -> W+ b  B
c  D   -> W+ b  B
c  S   -> W+ b  B
                 PgDn
```

IN-STATE, Structure functions



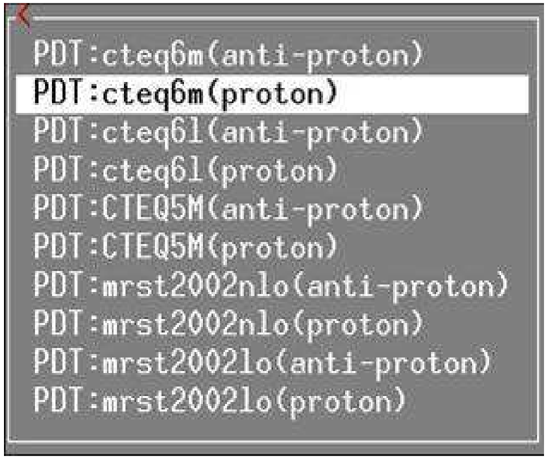
Subprocess
IN state
Model parameters
Constraints
QCD coupling
Breit-Wigner
Cuts
Phase space mapping
Vegas
Generate events

IN-STATE, Structure functions



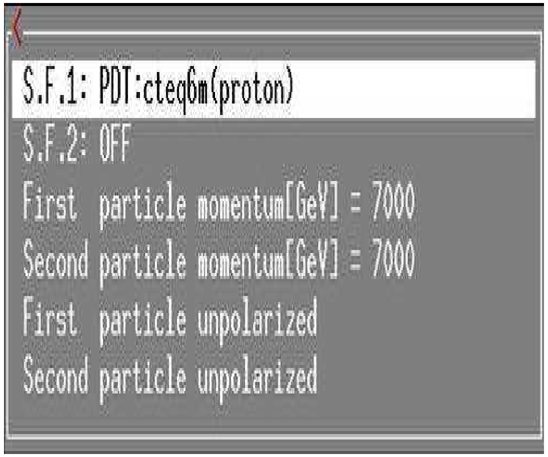
```
S.F.1: OFF
S.F.2: OFF
First  particle momentum[GeV] = 7000
Second particle momentum[GeV] = 7000
First  particle unpolarized
Second particle unpolarized
```

IN-STATE, Structure functions



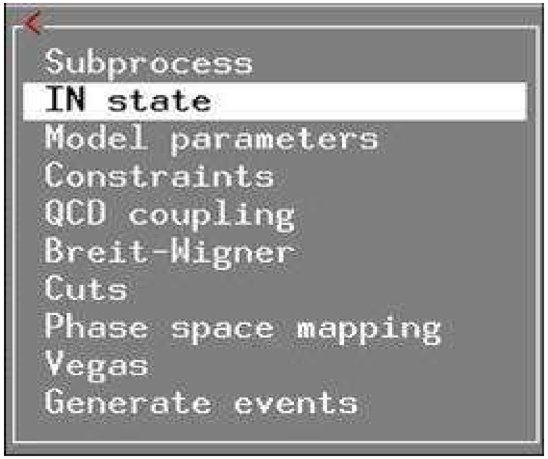
```
PDT:cteq6m(anti-proton)
PDT:cteq6m(proton)
PDT:cteq6l(anti-proton)
PDT:cteq6l(proton)
PDT:CTEQ5M(anti-proton)
PDT:CTEQ5M(proton)
PDT:mrst2002nlo(anti-proton)
PDT:mrst2002nlo(proton)
PDT:mrst2002lo(anti-proton)
PDT:mrst2002lo(proton)
```

IN-STATE, Structure functions

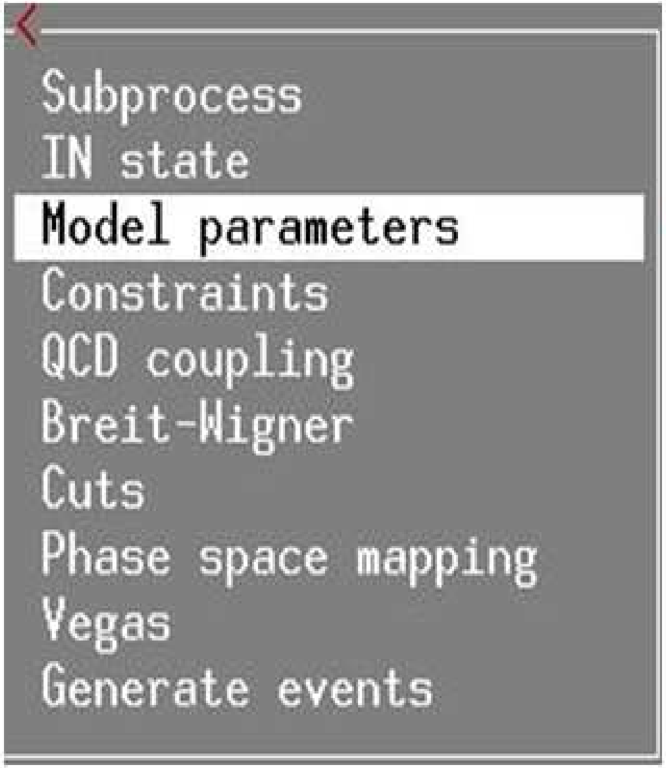


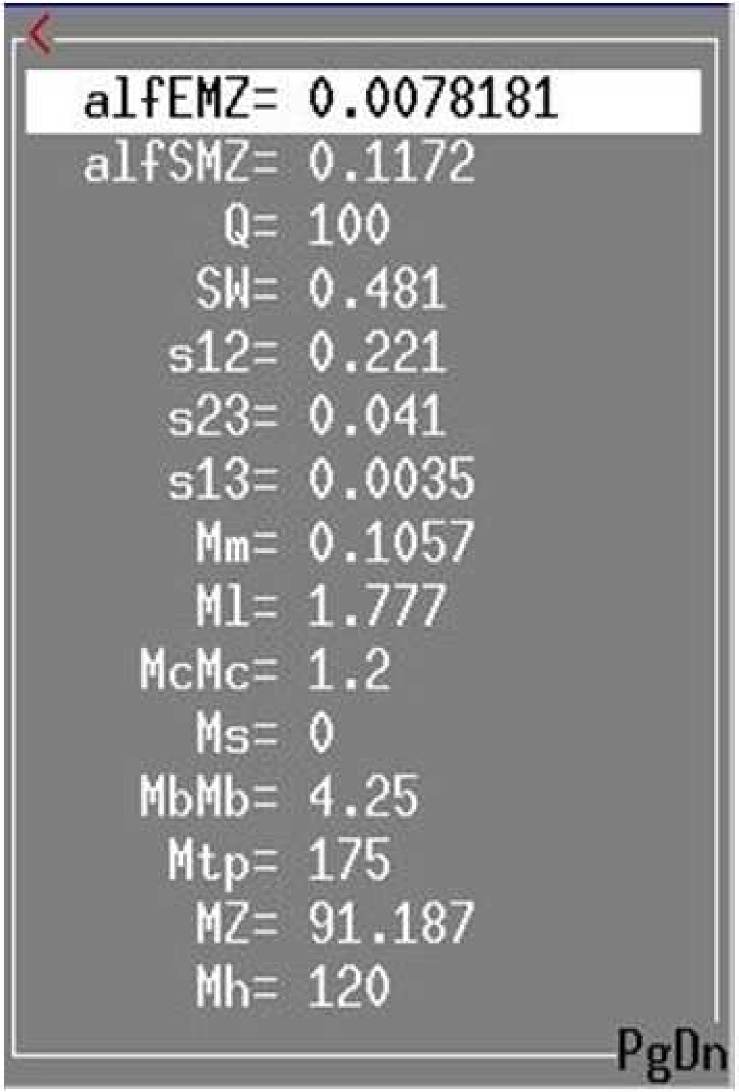
```
S.F.1: PDT:cteq6m(proton)
S.F.2: OFF
First  particle momentum[GeV] = 7000
Second particle momentum[GeV] = 7000
First  particle unpolarized
Second particle unpolarized
```

back to menu


Subprocess
IN state
Model parameters
Constraints
QCD coupling
Breit-Wigner
Cuts
Phase space mapping
Vegas
Generate events

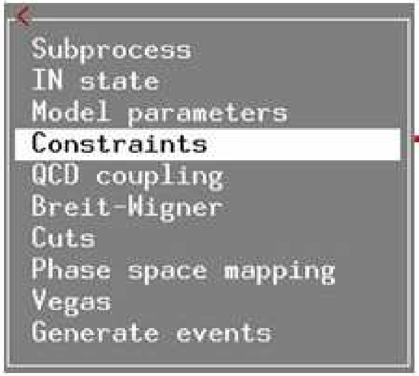
Model Parameters

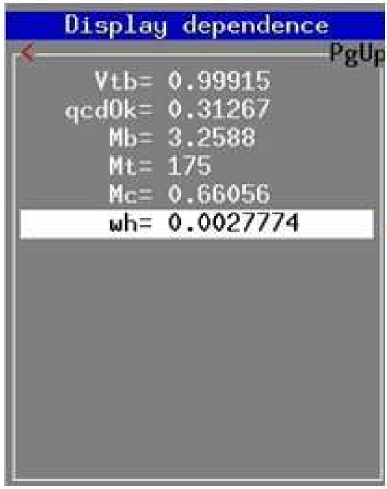




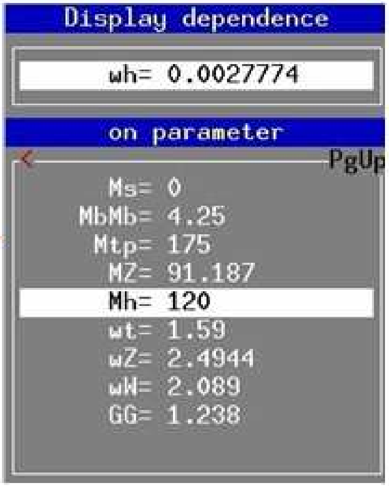
```
alfEMZ= 0.0078181
alfSMZ= 0.1172
     Q= 100
    SW= 0.481
   s12= 0.221
   s23= 0.041
   s13= 0.0035
    Mm= 0.1057
    Ml= 1.777
  McMc= 1.2
    Ms= 0
  MbMb= 4.25
   Mtp= 175
    MZ= 91.187
    Mh= 120
```

Constraints, (Higgs width on the fly)



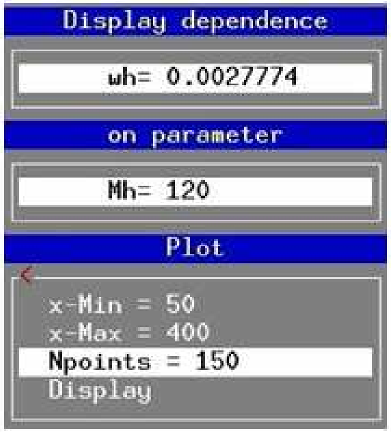


Vtb= 0.99915
qcd0k= 0.31267
Mb= 3.2588
Mt= 175
Mc= 0.66056
wh= 0.0027774



wh= 0.0027774

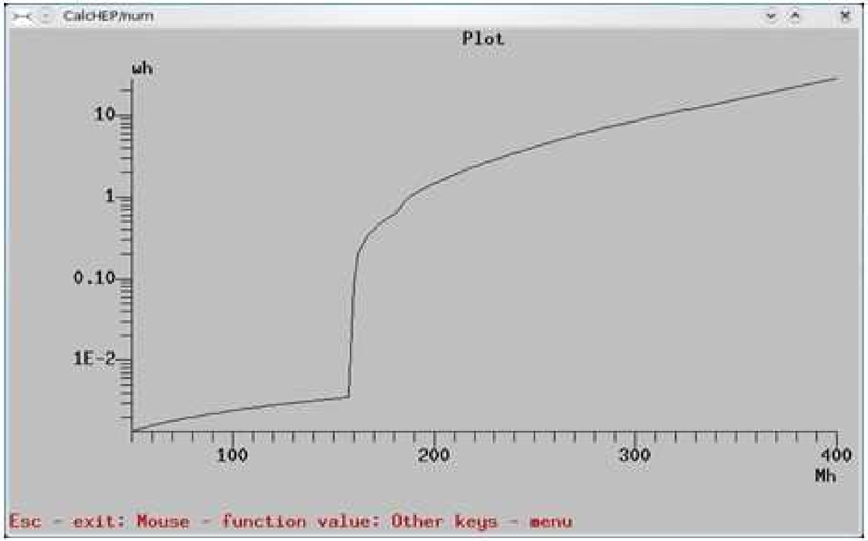
Ms= 0
MbMb= 4.25
Mtp= 175
MZ= 91.187
Mh= 120
wt= 1.59
wZ= 2.4944
wW= 2.089
GG= 1.238



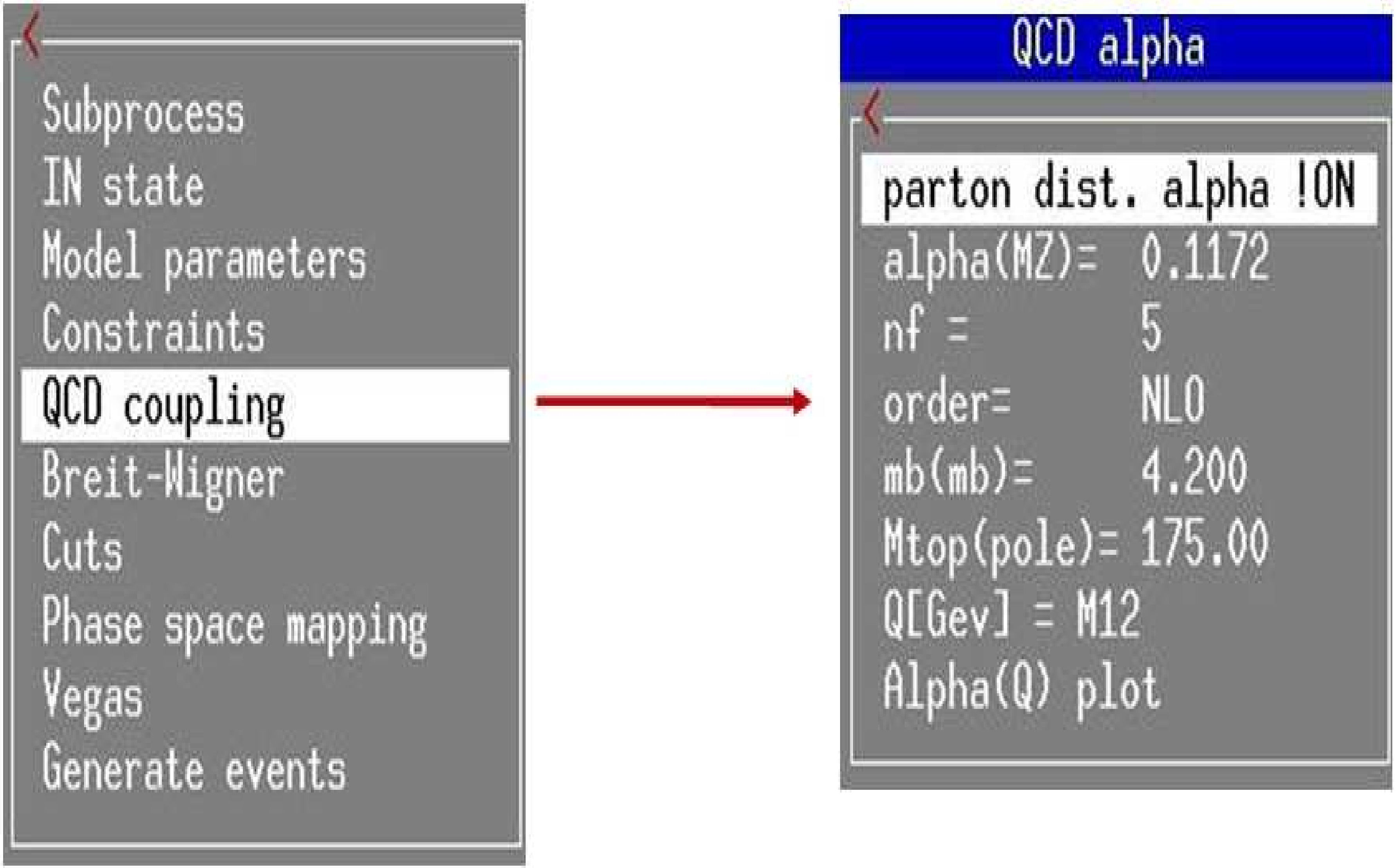
wh= 0.0027774

Mh= 120

x-Min = 50
x-Max = 400
Npoints = 150

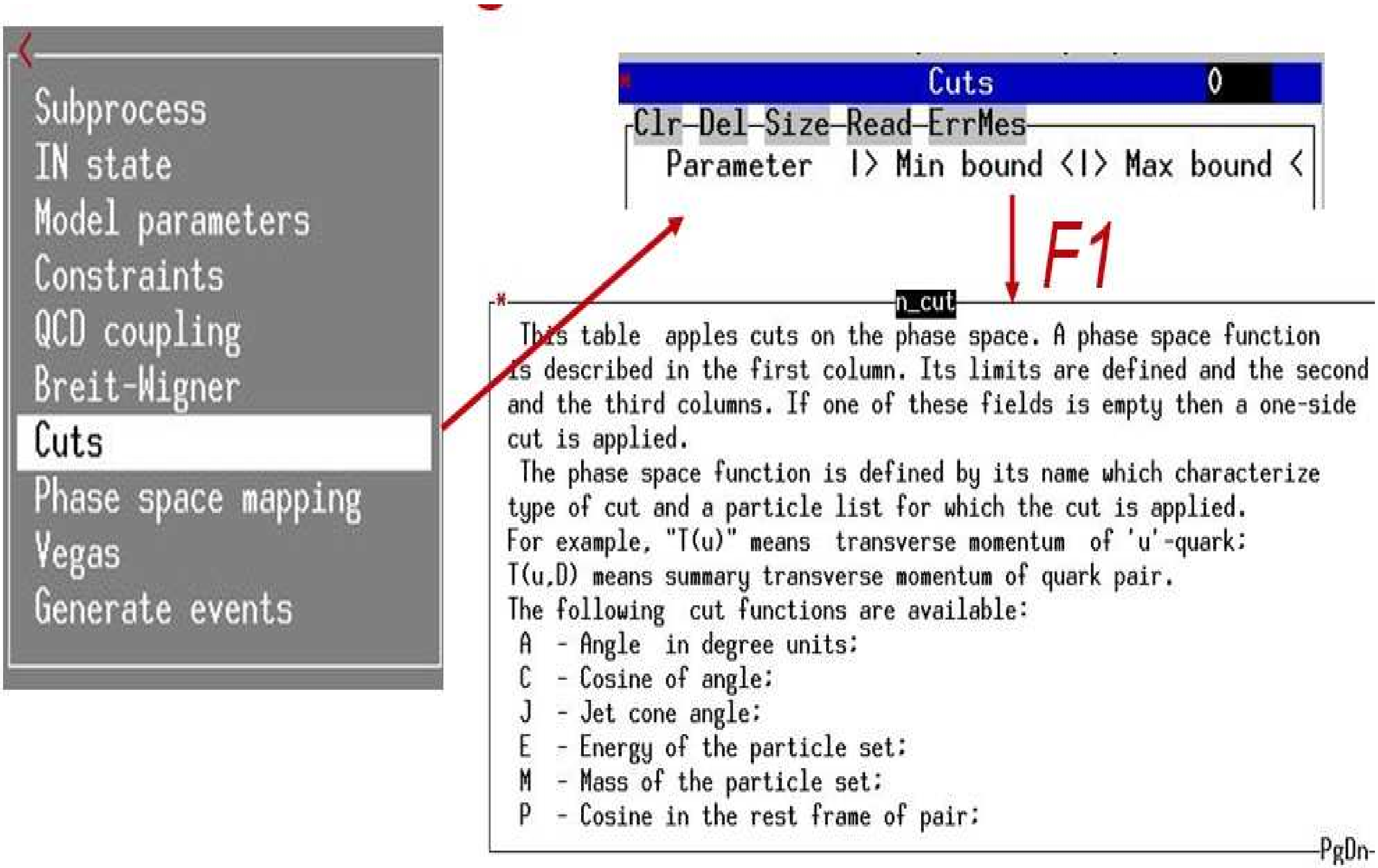


$\alpha_s(Q^2)$



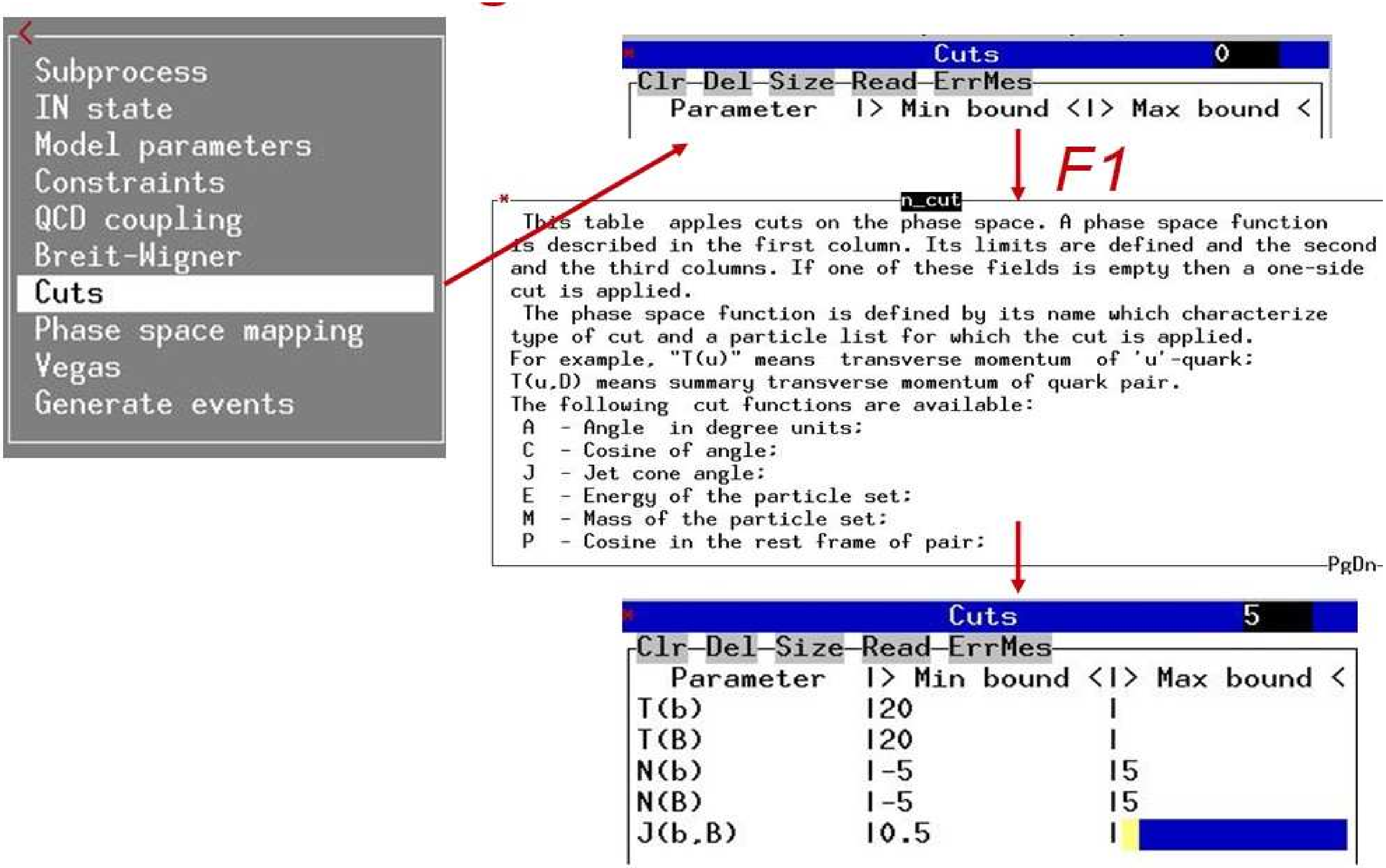
```
parton dist. alpha !ON
alpha(MZ)=  0.1172
nf =        5
order=      NLO
mb(mb)=     4.200
Mtop(pole)= 175.00
Q[Gev] = M12
Alpha(Q) plot
```

Setting cuts before integration



```
 This table  apples cuts on the phase space. A phase space function
is described in the first column. Its limits are defined and the second
and the third columns. If one of these fields is empty then a one-side
cut is applied.
 The phase space function is defined by its name which characterize
type of cut and a particle list for which the cut is applied.
For example, "T(u)" means  transverse momentum  of 'u'-quark;
T(u,D) means summary transverse momentum of quark pair.
The following  cut functions are available:
 A  - Angle  in degree units;
 C  - Cosine of angle;
 J  - Jet cone angle;
 E  - Energy of the particle set;
 M  - Mass of the particle set;
 P  - Cosine in the rest frame of pair;
```

Setting cuts before integration

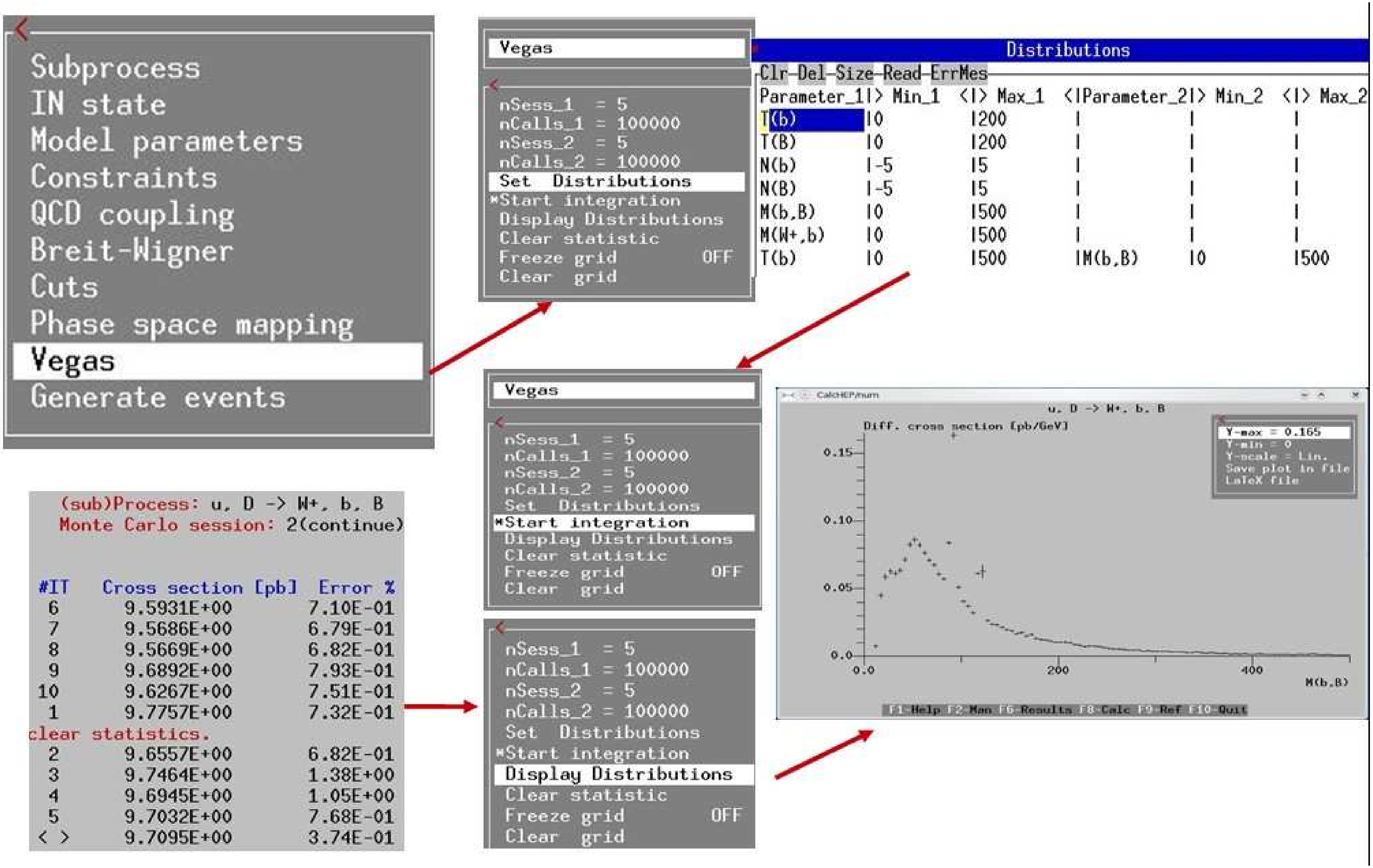


```
                       Cuts                          0
Clr-Del-Size-Read-ErrMes
   Parameter    |> Min bound <|> Max bound <
```

```
                                   n_cut
 This table  apples cuts on the phase space. A phase space function
is described in the first column. Its limits are defined and the second
and the third columns. If one of these fields is empty then a one-side
cut is applied.
 The phase space function is defined by its name which characterize
type of cut and a particle list for which the cut is applied.
For example, "T(u)" means  transverse momentum  of 'u'-quark:
T(u,D) means summary transverse momentum of quark pair.
The following  cut functions are available:
 A  - Angle  in degree units:
 C  - Cosine of angle:
 J  - Jet cone angle:
 E  - Energy of the particle set:
 M  - Mass of the particle set:
 P  - Cosine in the rest frame of pair:
                                                            PgDn
```

```
                       Cuts                          5
Clr-Del-Size-Read-ErrMes
   Parameter    |> Min bound <|> Max bound <
T(b)            |20           |
T(B)            |20           |
N(b)            |-5           |5
N(B)            |-5           |5
J(b,B)          |0.5          |
```

MC Integration, distributions



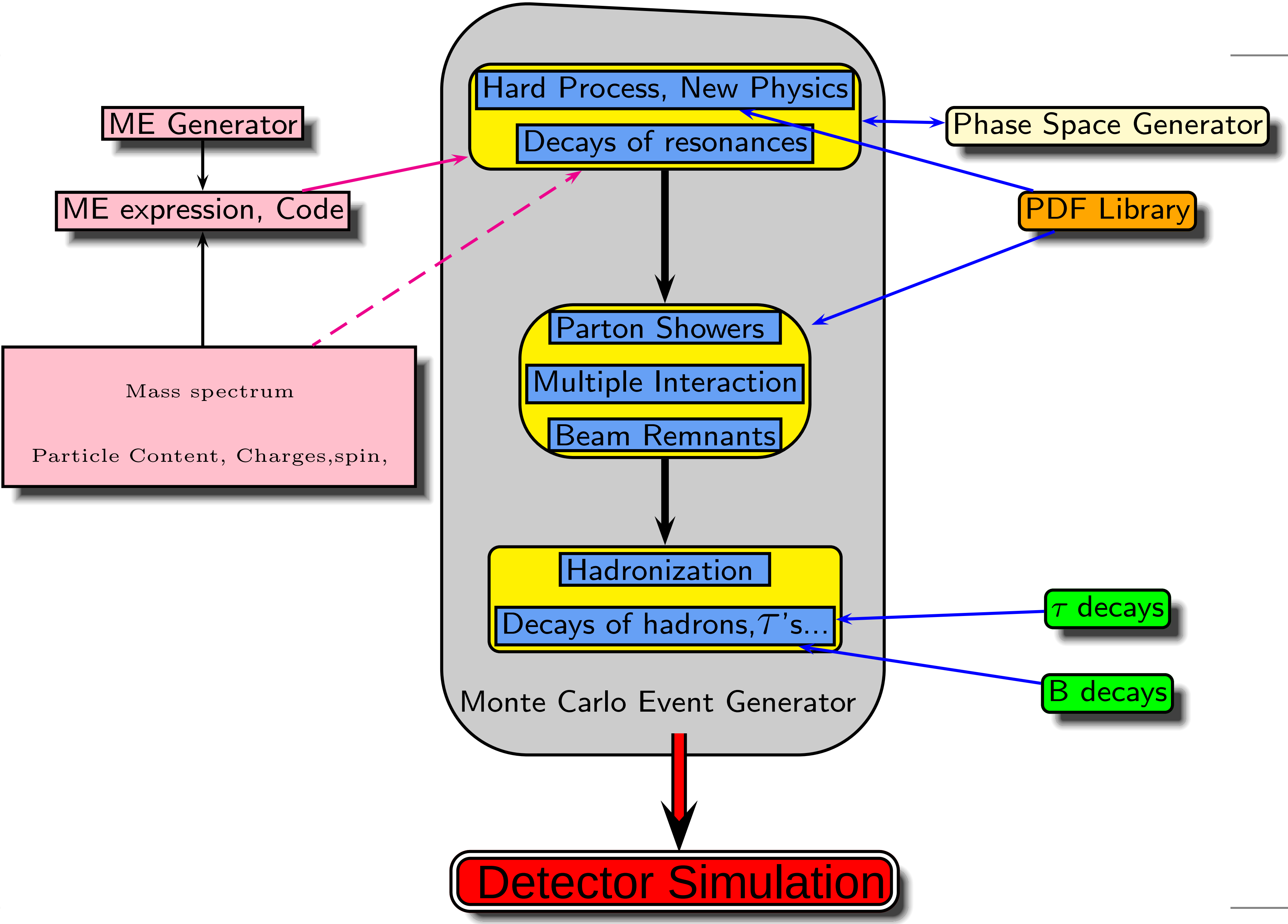
```
Subprocess
IN state
Model parameters
Constraints
QCD coupling
Breit-Wigner
Cuts
Phase space mapping
Vegas
Generate events
```

```
Vegas

nSess_1   = 5
nCalls_1 = 100000
nSess_2   = 5
nCalls_2 = 100000
Set   Distributions
*Start integration
Display Distributions
Clear statistic
Freeze grid           OFF
Clear  grid
```

```
                     Distributions
Clr-Del-Size-Read-ErrMes
Parameter_1|> Min_1   <|> Max_1   <|Parameter_2|> Min_2   <|> Max_2
T(b)         |0         |200       |            |          |
T(B)         |0         |200       |            |          |
N(b)         |-5        |5         |            |          |
N(B)         |-5        |5         |            |          |
M(b,B)       |0         |500       |            |          |
M(W+,b)      |0         |500       |            |          |
T(b)         |0         |500       |M(b,B)      |0         |500
```

```
(sub)Process: u, D -> W+, b, B
Monte Carlo session: 2(continue)

#IT   Cross section [pb]   Error %
 6      9.5931E+00       7.10E-01
 7      9.5686E+00       6.79E-01
 8      9.5669E+00       6.82E-01
 9      9.6892E+00       7.93E-01
10      9.6267E+00       7.51E-01
 1      9.7757E+00       7.32E-01
clear statistics.
 2      9.6557E+00       6.82E-01
 3      9.7464E+00       1.38E+00
 4      9.6945E+00       1.05E+00
 5      9.7032E+00       7.68E-01
< >     9.7095E+00       3.74E-01
```

Putting all together



ME Generator

ME expression, Code

Mass spectrum

Particle Content, Charges,spin,

Hard Process, New Physics

Decays of resonances

Parton Showers

Multiple Interaction

Beam Remnants

Hadronization

Decays of hadrons,τ's...

Monte Carlo Event Generator

Phase Space Generator

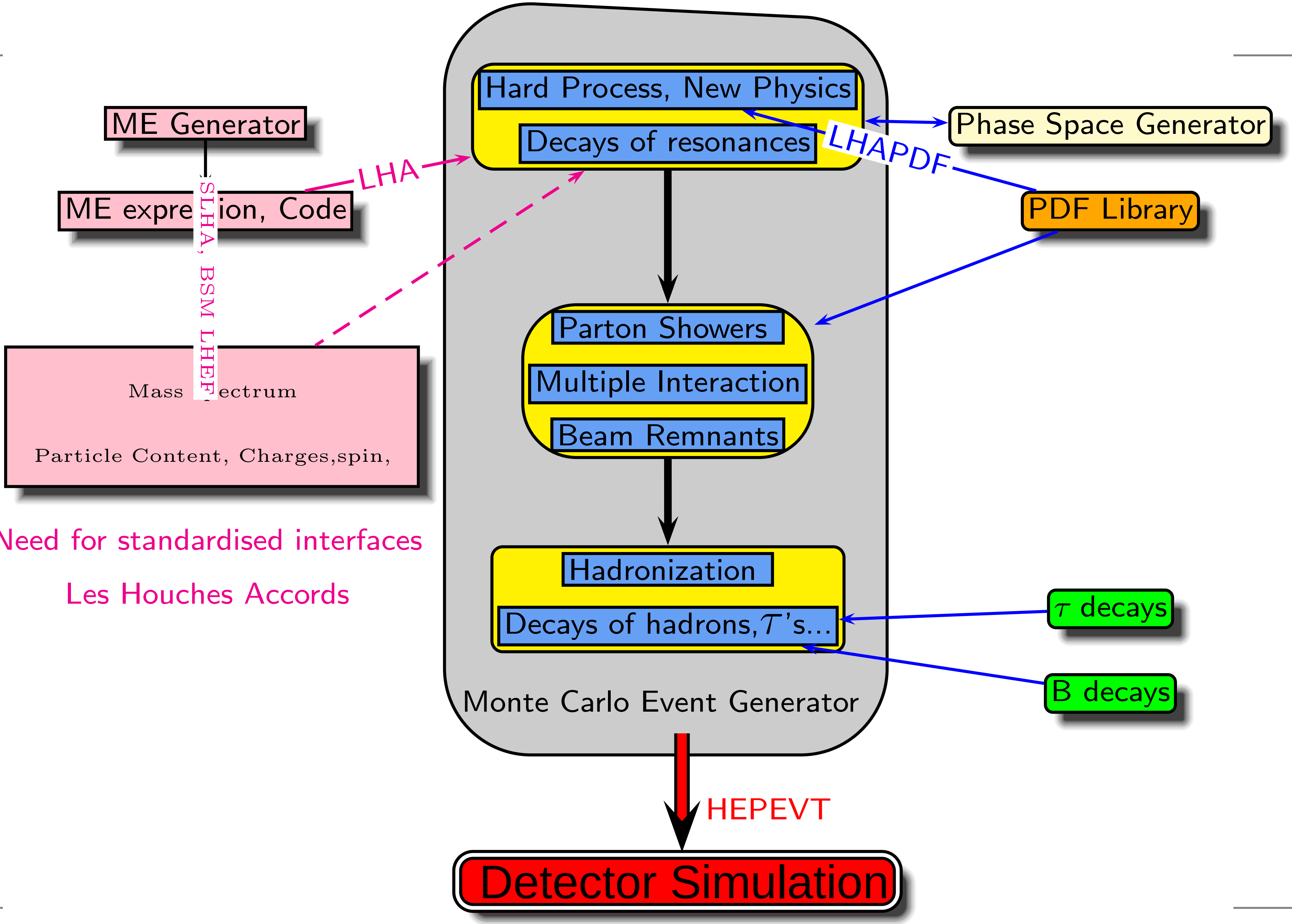
PDF Library

τ decays

B decays

Detector Simulation

Putting all together, Les Houches Accords



ME Generator

ME expre ion, Code

SLHA, BSM LHEF

Mass ectrum

Particle Content, Charges,spin,

LHA

Hard Process, New Physics

Decays of resonances

Phase Space Generator

LHAPDF

PDF Library

Parton Showers

Multiple Interaction

Beam Remnants

Hadronization

Decays of hadrons,τ's...

τ decays

B decays

Monte Carlo Event Generator

HEPEVT

Detector Simulation

Need for standardised interfaces

Les Houches Accords

Les Houches Accord: Examples

Pass on the names in a standard manner: PDG code.

Code	Name	Code	Name	Code	Name	Code	Name
1	d	11	e^-	21	g		
2	u	12	ν_e	22	γ	35	H^0
3	s	13	μ^-	23	Z^0	36	A^0
4	c	14	ν_μ	24	W^+	37	H^+
5	b	15	τ^-	25	h^0		
6	t	16	ν_τ			39	G (graviton)

SM fundamental particle codes (+ extended Higgs sector)

Les Houches Accord: Examples

Pass on the names in a standard manner: PDG code.

Code	Name	Code	Name	Code	Name	Code	Name
1	d	11	e^-	21	g		
2	u	12	ν_e	22	γ	35	H^0
3	s	13	μ^-	23	Z^0	36	A^0
4	c	14	ν_μ	24	W^+	37	H^+
5	b	15	τ^-	25	h^0		
6	t	16	ν_τ			39	G (graviton)

SM fundamental particle codes (+ extended Higgs sector)

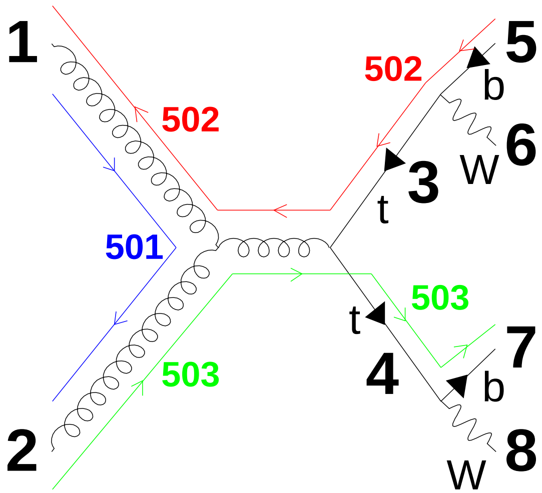
For New Physics particles, create new code names (1000000+), SLHA2

Scalar Quarks

FLV	No	Yes	No	No	Yes	Yes	NMSSM
RPV	No	No	Yes	No	Yes	No	
CPV	No	No	No	Yes	No	Yes	
1000001	$\tilde{d}_L$	$\tilde{d}_1$	$\tilde{d}_1$	$\tilde{d}_L$	$\tilde{d}_1$	$\tilde{d}_1$	$\tilde{d}_L$
1000002	$\tilde{u}_L$	$\tilde{u}_1$	$\tilde{u}_1$	$\tilde{u}_L$	$\tilde{u}_1$	$\tilde{u}_1$	$\tilde{u}_L$
1000003	$\tilde{s}_L$	$\tilde{d}_2$	$\tilde{d}_2$	$\tilde{s}_L$	$\tilde{d}_2$	$\tilde{d}_2$	$\tilde{s}_L$
1000004	$\tilde{c}_L$	$\tilde{u}_2$	$\tilde{u}_2$	$\tilde{c}_L$	$\tilde{u}_2$	$\tilde{u}_2$	$\tilde{c}_L$
1000005	$\tilde{b}_1$	$\tilde{d}_3$	$\tilde{d}_3$	$\tilde{b}_1$	$\tilde{d}_3$	$\tilde{d}_3$	$\tilde{b}_1$
1000006	$\tilde{t}_1$	$\tilde{u}_3$	$\tilde{u}_3$	$\tilde{t}_1$	$\tilde{u}_3$	$\tilde{u}_3$	$\tilde{t}_1$
2000001	$\tilde{d}_R$	$\tilde{d}_4$	$\tilde{d}_4$	$\tilde{d}_R$	$\tilde{d}_4$	$\tilde{d}_4$	$\tilde{d}_R$
2000002	$\tilde{u}_R$	$\tilde{u}_4$	$\tilde{u}_4$	$\tilde{u}_R$	$\tilde{u}_4$	$\tilde{u}_4$	$\tilde{u}_R$
2000003	$\tilde{s}_R$	$\tilde{d}_5$	$\tilde{d}_5$	$\tilde{s}_R$	$\tilde{d}_5$	$\tilde{d}_5$	$\tilde{s}_R$
2000004	$\tilde{c}_R$	$\tilde{u}_5$	$\tilde{u}_5$	$\tilde{c}_R$	$\tilde{u}_5$	$\tilde{u}_5$	$\tilde{c}_R$
2000005	$\tilde{b}_2$	$\tilde{d}_6$	$\tilde{d}_6$	$\tilde{b}_2$	$\tilde{d}_6$	$\tilde{d}_6$	$\tilde{b}_2$
2000006	$\tilde{t}_2$	$\tilde{u}_6$	$\tilde{u}_6$	$\tilde{t}_2$	$\tilde{u}_6$	$\tilde{u}_6$	$\tilde{t}_2$

Particle codes and corresponding labels for squarks.
The labels in the first column correspond to the current PDG nomenclature.

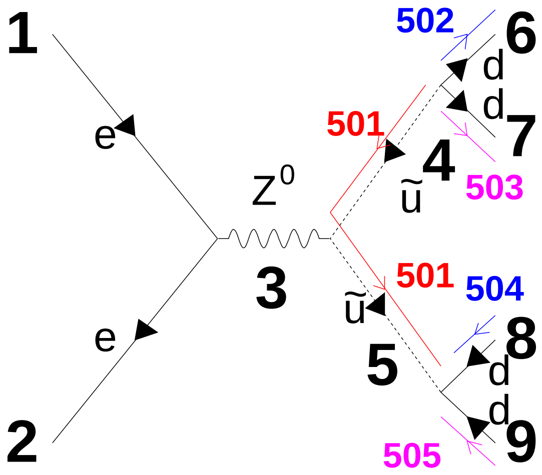
Les Houches Accord: Examples



I	ISTUP(I)	IDUP(I)	MOTHUP(1,I)	MOTHUP(2,I)	ICOLUP(1,I)	ICOLUP(2,I)
1	–1	21 (g)	0	0	501	502
2	–1	21 (g)	0	0	503	501
3	+2	–6 ($\bar{t}$)	1	2	0	502
4	+2	6 (t)	1	2	503	0
5	+1	–5 ($\bar{b}$)	3	3	0	502
6	+1	–24 (W^-)	3	3	0	0
7	+1	5 (b)	4	4	503	0
8	+1	24 (W^+)	4	4	0	0

Pass the colour information, essential for parton shower and hadronisation, apart of course from the kinematics

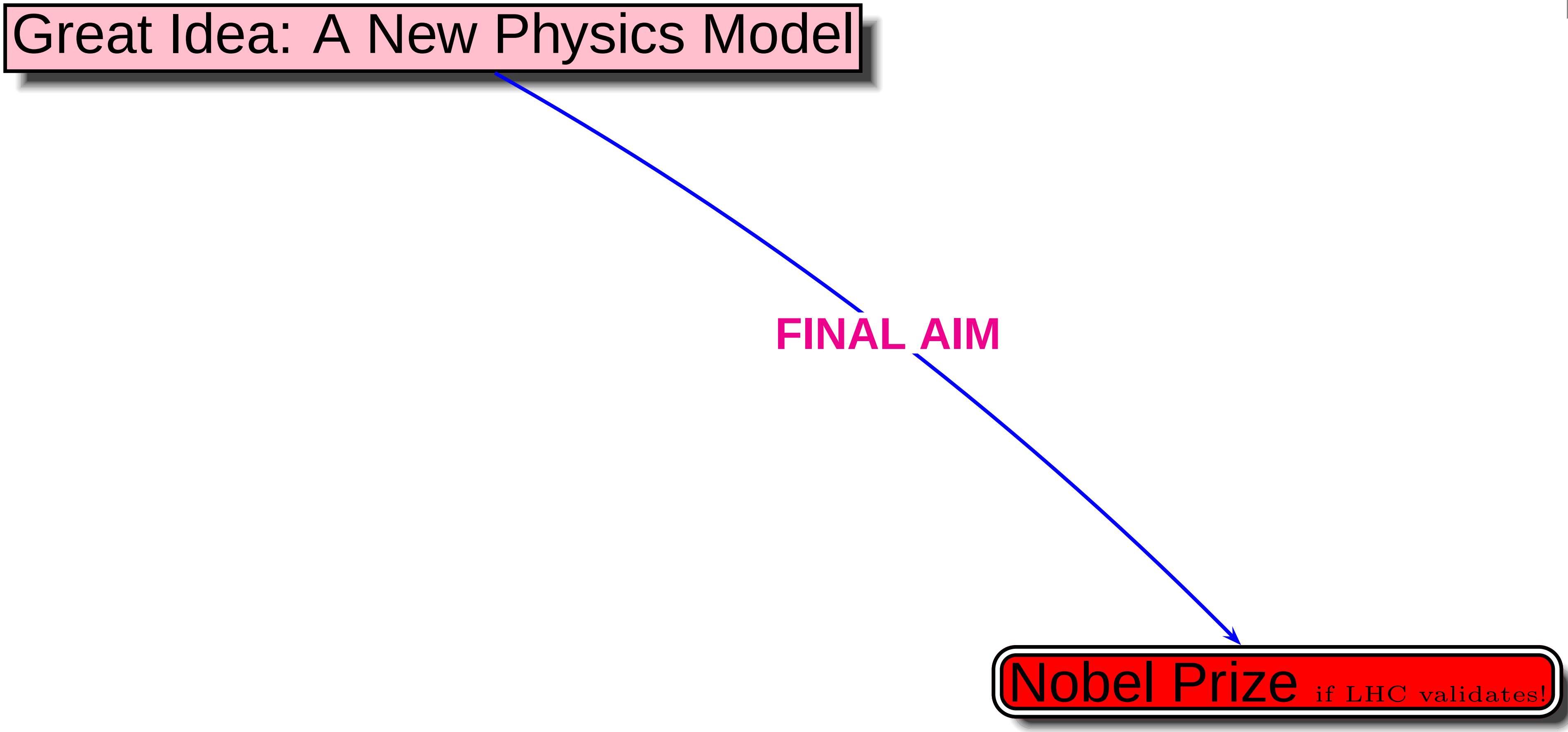
Les Houches Accord: Examples



I	ISTUP(I)	IDUP(I)	MOTHUP(1,I)	MOTHUP(2,I)	ICOLUP(1,I)	ICOLUP(
1	–1	11 (e^-)	0	0	0	0
2	–1	–11 (e^+)	0	0	0	0
3	+2	23 (Z^0)	1	2	0	0
4	+2	–1000002 ($\tilde{\bar{u}}$)	3	3	0	501
5	+2	1000002 ($\tilde{u}$)	3	3	501	0
6	+1	1 (d)	4	4	502	0
7	+1	1 (d)	4	4	503	0
8	+1	–1 ($\bar{d}$)	5	5	0	504
9	+1	–1 ($\bar{d}$)	5	5	0	505

Baryon number violation, SUSY RPV

Nobel Dreams



Too many Codes?

yes. but we have a modular structure, let codes talk to each other and let's implement quickly and efficiently

from Konstantin Matchev

Experimentalist's complaint: This model is very nice,

but do you have an event generator for it? is it in Pythia?

not that many MC developers

On the other hand, too many model builders

$$N_{\text{model builders}} \gg N_{MCdev.} \longrightarrow$$

$$N_{existing\ models} \gg N_{implemented\ model}$$

even worse

$$dN_{existing\ models}/dt \gg dN_{implemented\ model}/dt$$

SLHA 1 and 2

SLHA (1): MSSM, SLHA2 (CPV,RPV,NMSSM) Most of the authors have adopted it

- Signs (μ,..), factors of $\sqrt{2}$
- Mixing angles conventions
- Eigenstates decomposition
- Renormalisation schemes/scales !!!
- Effective field content (sparticles integrated out or not)
- SLHA2 more of a headache, but we got there eventually

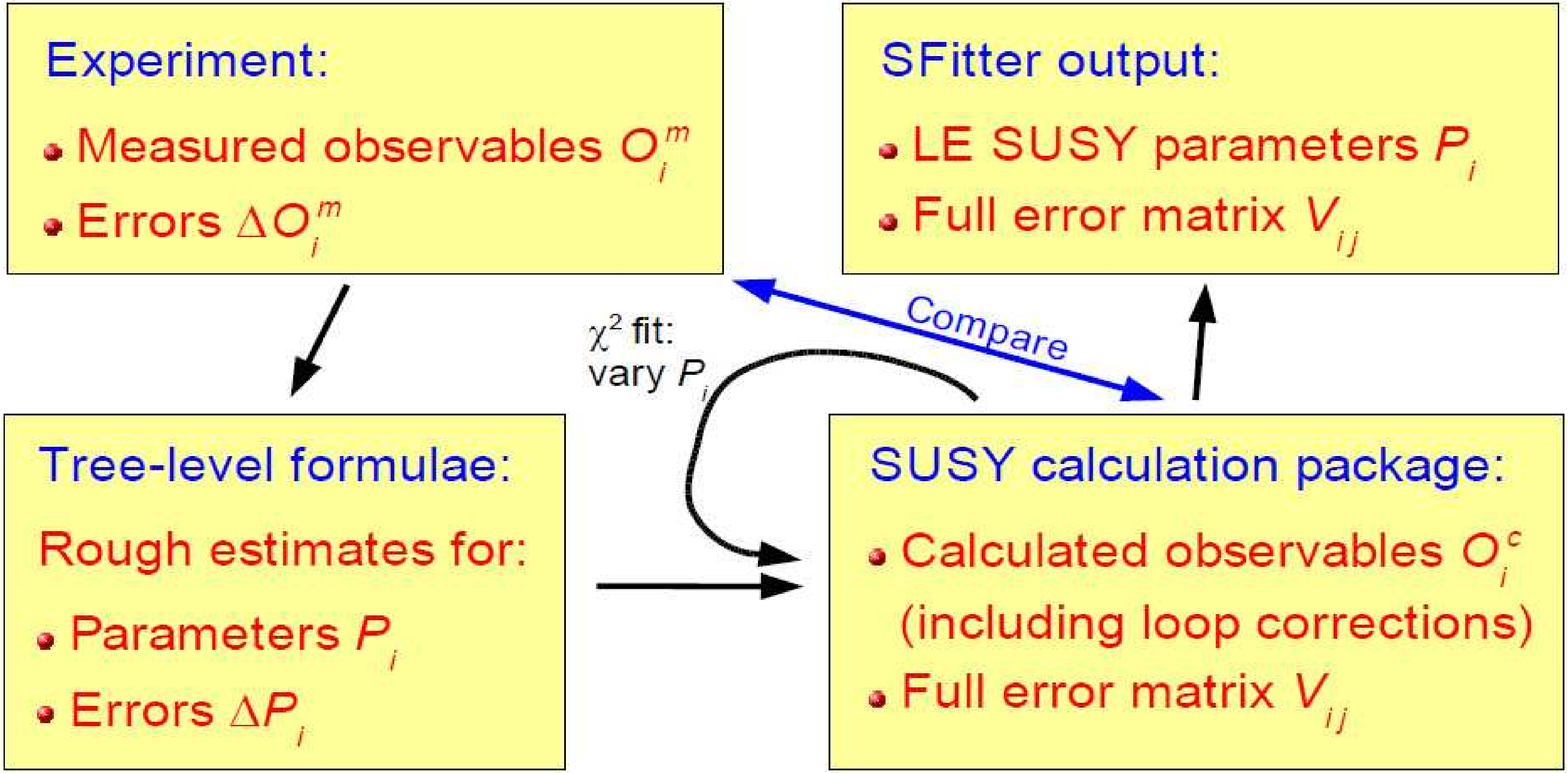
BSM LHEF

This is a mix of

- SLHA2 and (model parameters)

  introduce new SLHA like blocks QNUMBERS for each BSM particle containing PDG code, spin, elec. charge, colour rep., particle/antiparticle
- and LHEF2 (xml format) for event files

Codes for Fit”s: Fittino, SFITTER

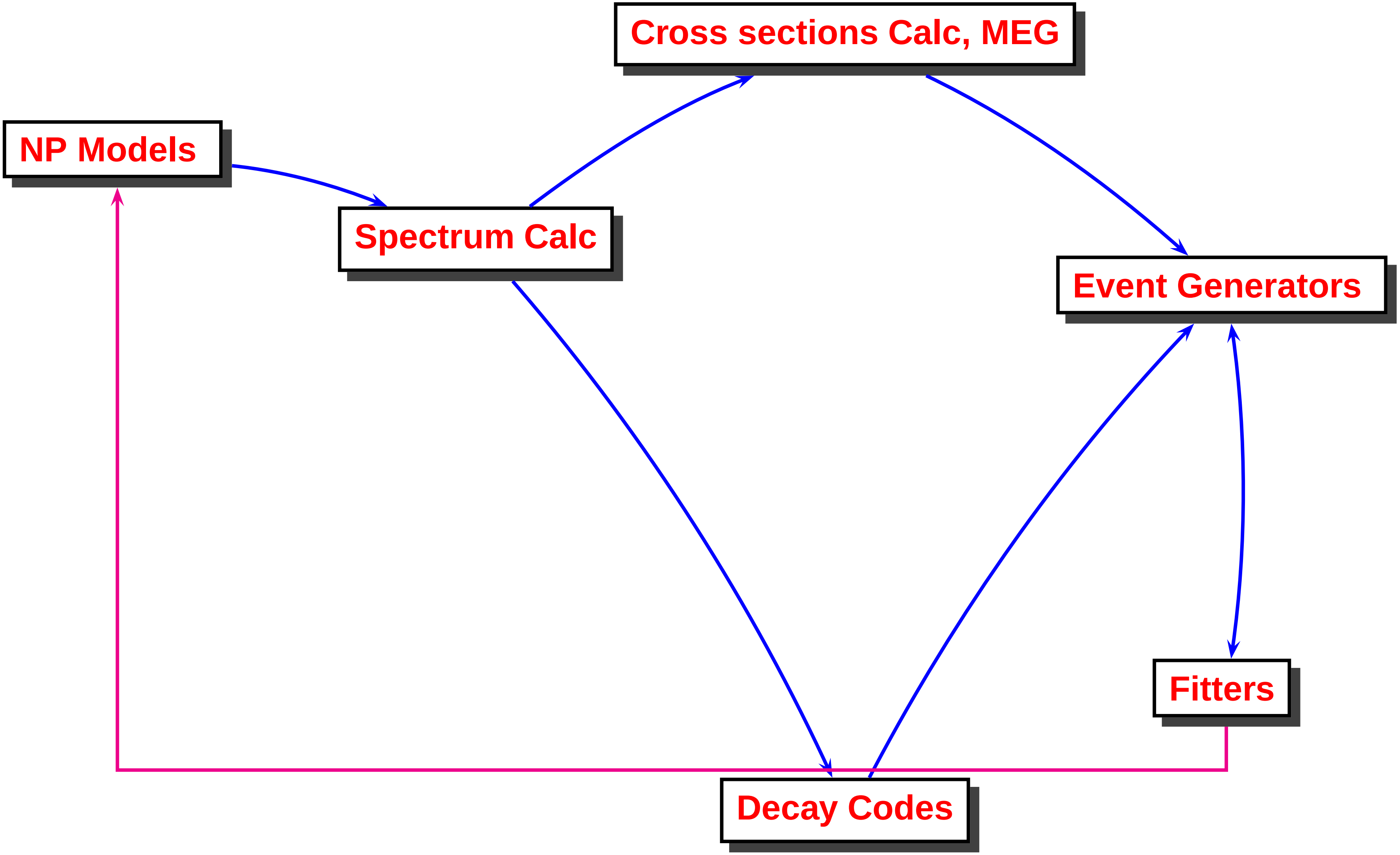


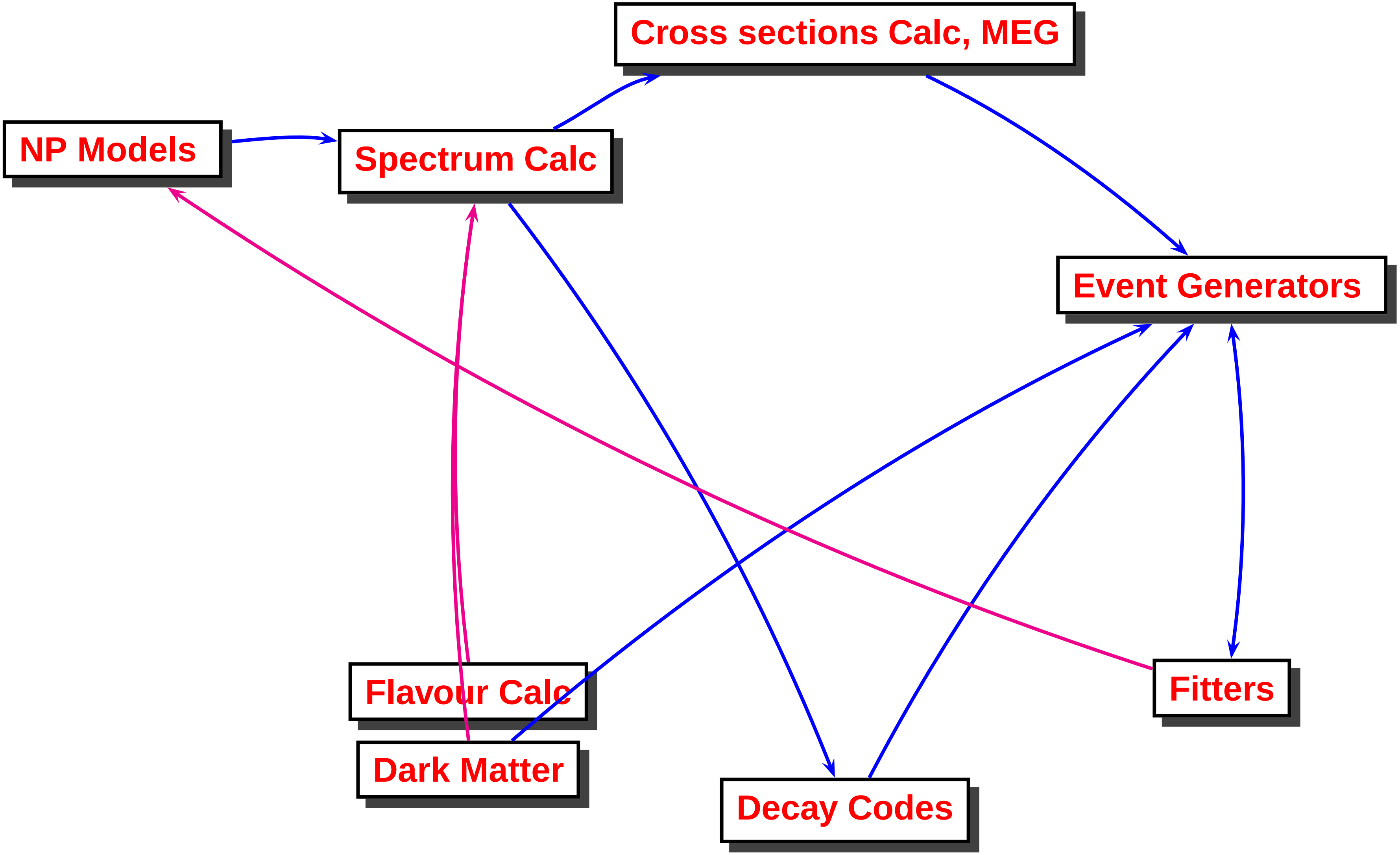
tree-level vs “1-loop level”, fits after background subtraction?

NP Models

FINAL AIM

Event Generators


Cross sections Calc, MEG
NP Models
Spectrum Calc
Event Generators
Fitters
Decay Codes



Cross sections Calc, MEG

NP Models

Spectrum Calc

Event Generators

Flavour Calc

Fitters

Dark Matter

Decay Codes

Cross sections Calc, MEG

NP Models

Spectrum Calc

- SUSY
  - MSSM
  - mSUGRA
  - GMSB, AMSB
  - NMSSM
  - RPV, CPV,...

Event Generators

- TeXColour
- Extra-dim
- Little Higgs
- $f^\star, V'$
- Black Holes (!)

Flavour Calc

Fitters

Dark Matter

Decay Codes

Cross sections Calc, MEG

NP Models

- SUSY
  - MSSM
  - mSUGRA
  - GMSB, AMSB
  - NMSSM
  - RPV, CPV,...
- TeXColour
- Extra-dim
- Little Higgs
- $f^{\star}, V'$
- Black Holes (!)

Spectrum Calc

- FeynHiggs
- NMHDECAY$\star$
- RGE Codes
  - Isasusy
  - SoftSusy
  - Spheno
  - Suspect

Event Generators

Flavour Calc

Dark Matter

Decay Codes

Fitters

NP Models

- SUSY
  - MSSM
  - mSUGRA
  - GMSB, AMSB
  - NMSSM
  - RPV, CPV,...
- TeXColour
- Extra-dim
- Little Higgs
- $f^\star, V'$
- Black Holes (!)

Spectrum Calc

- `FeynHiggs`
- `NMHDECAY`$\star$
- RGE Codes
  - `Isasusy`
  - `SoftSusy`
  - `Spheno`
  - `Suspect`

Flavour Calc

- $(g-2)_\mu$
- $b \to s\gamma$
- $B_s \to \mu^+\mu^-$
- Asym, $\Delta M, \cdots$

Dark Matter

Cross sections Calc, MEG

Event Generators

Fitters

Decay Codes

NP Models

- SUSY
  - MSSM
  - mSUGRA
  - GMSB, AMSB
  - NMSSM
  - RPV, CPV,...
- TeXColour
- Extra-dim
- Little Higgs
- $f^\star, V'$
- Black Holes (!)

Spectrum Calc

- `FeynHiggs`
- `NMHDECAY`$\star$
- RGE Codes
  - `Isasusy`
  - `SoftSusy`
  - `Spheno`
  - `Suspect`

Flavour Calc

Dedicated Codes

- `SusyBSG`
- `SuperIso`

Dark Matter

Cross sections Calc, MEG

- Tree-level,any
  - `CalcHEP, CompHEP`
  - `GRACE, FORMCalc`
  - `Madgraph`
  - `SHERPA/Amegic++`
  - `Whizard/O'Mega`
- 1-loop dedicated
  - `AF's SLEPTONS`
  - `Prospino, hprod`
- 1-loop/General
  - `GRACE-SUSY`
  - `FormCalc,SloopS`

Event Generators

Fitters

Decay Codes

NP Models

- SUSY
  - MSSM
  - mSUGRA
  - GMSB, AMSB
  - NMSSM
  - RPV, CPV,...
- TeXColour
- Extra-dim
- Little Higgs
- $f^\star, V'$
- Black Holes (!)

Spectrum Calc

- `FeynHiggs`
- `NMHDECAY`$\star$
- RGE Codes
  - `Isasusy`
  - `SoftSusy`
  - `Spheno`
  - `Suspect`

Flavour Calc

Dark Matter

Cross sections Calc, MEG

- Tree-level,any
  - `CalcHEP, CompHEP`
  - `GRACE, FORMCalc`
  - `Madgraph`
  - `SHERPA/Amegic++`
  - `Whizard/O'Mega`
- 1-loop dedicated
  - `AF's SLEPTONS`
  - `Prospino, hprod`
- 1-loop/General
  - `GRACE-SUSY`
  - `FormCalc,SloopS`

Event Generators

Fitters

Decay Codes

- `BRIDGE`
- `HDECAY`
- `NMHDECAY`$\star$
- `SDECAY`

NP Models

- SUSY
  - MSSM
  - mSUGRA
  - GMSB, AMSB
  - NMSSM
  - RPV, CPV,...
- TeXColour
- Extra-dim
- Little Higgs
- $f^\star, V'$
- Black Holes (!)

Spectrum Calc

- `FeynHiggs`
- `NMHDECAY⋆`
- RGE Codes
  - `Isasusy`
  - `SoftSusy`
  - `Spheno`
  - `Suspect`

Flavour Calc

Dark Matter

- `SIsoRelic`
- `micrOMEGAs`
  - `SloopS⋆`
- `DARKSUSY`
- `IsaRED/RES`

Cross sections Calc, MEG

- Tree-level,any
  - `CalcHEP, CompHEP`
  - `GRACE, FORMCalc`
  - `Madgraph`
  - `SHERPA/Amegic++`
  - `Whizard/O'Mega`
- 1-loop dedicated
  - `AF's SLEPTONS`
  - `Prospino, hprod`
- 1-loop/General
  - `GRACE-SUSY`
  - `FormCalc,SloopS`

Event Generators

Fitters

Decay Codes

- `BRIDGE`
- `HDECAY`
- `NMHDECAY⋆`
- `SDECAY`

NP Models

- SUSY
  MSSM
  mSUGRA
  GMSB, AMSB
  NMSSM
  RPV, CPV,...
- TeXColour
- Extra-dim
- Little Higgs
- $f^{\star}, V'$
- Black Holes (!)

Spectrum Calc

- `FeynHiggs`
- `NMHDECAY`$\star$
- RGE Codes
  `Isasusy`
  `SoftSusy`
  `Spheno`
  `Suspect`

Flavour Calc

Dark Matter

- `SIsoRelic`
- `micrOMEGAs`
  `SloopS`$\star$
- `DARKSUSY`
- `IsaRED/RES`

Cross sections Calc, MEG

- Tree-level,any
  `CalcHEP, CompHEP`
  `GRACE, FORMCalc`
  `Madgraph`
  `SHERPA/Amegic++`
  `Whizard/O'Mega`
- 1-loop dedicated
  `AF's SLEPTONS`
  `Prospino, hprod`
- 1-loop/General
  `GRACE-SUSY`
  `FormCalc,SloopS`

Decay Codes

- `BRIDGE`
- `HDECAY`
- `NMHDECAY`$\star$
- `SDECAY`

Event Generators

- `[Isajet]`
- `Herwig++`
- `Pythia`
- `Sherpa`

Fitters

NP Models

- SUSY
  - MSSM
  - mSUGRA
  - GMSB, AMSB
  - NMSSM
  - RPV, CPV,...
- TeXColour
- Extra-dim
- Little Higgs
- $f^\star, V'$
- Black Holes (!)

Spectrum Calc

- FeynHiggs
- NMHDECAY⋆
- RGE Codes
  - Isasusy
  - SoftSusy
  - Spheno
  - Suspect

Flavour Calc

Dark Matter

- SIsoRelic
- micrOMEGAs
  - SloopS⋆
- DARKSUSY
- IsaRED/RES

Cross sections Calc, MEG

- Tree-level,any
  - CalcHEP, CompHEP
  - GRACE, FORMCalc
  - Madgraph
  - SHERPA/Amegic++
  - Whizard/O’Mega
- 1-loop dedicated
  - AF’s SLEPTONS
  - Prospino, hprod
- 1-loop/General
  - GRACE-SUSY
  - FormCalc,SloopS

Decay Codes

- BRIDGE
- HDECAY
- NMHDECAY⋆
- SDECAY

Event Generators

- [Isajet]
- Herwig++
- Pythia
- Sherpa

Fitters

- Fittino
- SFitter
- SuperBayes
- HiggsBounds
- MasterCode !

NP Models

- SUSY
  - MSSM
  - mSUGRA
  - GMSB, AMSB
  - NMSSM
  - RPV, CPV,...
- TeXColour
- Extra-dim
- Little Higgs
- $f^\star, V'$
- Black Holes (!)

Spectrum Calc

- FeynHiggs
- NMHDECAY⋆
- RGE Codes
  - Isasusy
  - SoftSusy
  - Spheno
  - Suspect

Flavour Calc

Dark Matter

- SIsoRelic
- micrOMEGAs
  - SloopS⋆
- DARKSUSY
- IsaRED/RES

Cross sections Calc, MEG

- Tree-level,any
  - CalcHEP, CompHEP
  - GRACE, FORMCalc
  - Madgraph
  - SHERPA/Amegic++
  - Whizard/O'Mega
- 1-loop dedicated
  - AF's SLEPTONS
  - Prospino, hprod
- 1-loop/General
  - GRACE-SUSY
  - FormCalc,SloopS

Decay Codes

- BRIDGE
- HDECAY
- NMHDECAY⋆
- SDECAY

Event Generators

- [Isajet]
- Herwig++
- Pythia
- Sherpa

Fitters

- Fittino
- SFitter
- SuperBayes
- HiggsBounds
- MasterCode !

Black Holes

CatFish, Charybdis, TrueNoir

NP Models

- SUSY
  - MSSM
  - mSUGRA
  - GMSB, AMSB
  - NMSSM
  - RPV, CPV,...
- TeXColour
- Extra-dim
- Little Higgs
- $f^\star, V'$
- Black Holes (!)

$\mathcal{L}$

Feynman rules

(Eff. Pot.)

Spectrum Calc

- `FeynHiggs`
- `NMHDECAY`$\star$
- RGE Codes
  `Isasusy`
  `SoftSusy`
  `Spheno`

  `Suspect`

Flavour Calc

Dark Matter

- `SIsoRelic`
- `micrOMEGAs`
  `SloopS`$\star$
- `DARKSUSY`
- `IsaRED/RES`

Cross sections Calc, MEG

- Tree-level,any
  `CalcHEP, CompHEP`
  `GRACE, FORMCalc`
  `Madgraph`
  `SHERPA/Amegic++`
  `Whizard/O'Mega`
- 1-loop dedicated
  `AF's SLEPTONS`
  `Prospino, hprod`
- 1-loop/General
  `GRACE-SUSY`

  `FormCalc,SloopS`

Decay Codes

- `BRIDGE`
- `HDECAY`
- `NMHDECAY`$\star$
- `SDECAY`

Event Generators

- `[Isajet]`
- `Herwig++`
- `Pythia`
- `Sherpa`

Fitters

- `Fittino`
- `SFitter`
- `SuperBayes`
- `HiggsBounds`
- `MasterCode!`

Black Holes

CatFish, Charybdis,

TrueNoir

NP Models

- SUSY
  - MSSM
  - mSUGRA
  - GMSB, AMSB
  - NMSSM
  - RPV, CPV,...
- TeXColour
- Extra-dim
- Little Higgs
- $f^{\star}, V'$
- Black Holes (!)

$\mathcal{L}$

manual

Feynman rules

(Eff. Pot.)

Spectrum Calc

- FeynHiggs
- NMHDECAY⋆
- RGE Codes
  - Isasusy
  - SoftSusy
  - Spheno
  - Suspect

Flavour Calc

Dark Matter

- SIsoRelic
- micrOMEGAs
  - SloopS⋆
- DARKSUSY
- IsaRED/RES

manual

Cross sections Calc, MEG

- Tree-level,any
  - CalcHEP, CompHEP
  - GRACE, FORMCalc
  - Madgraph
  - SHERPA/Amegic++
  - Whizard/O'Mega
- 1-loop dedicated
  - AF's SLEPTONS
  - Prospino, hprod
- 1-loop/General
  - GRACE-SUSY
  - FormCalc,SloopS

Decay Codes

- BRIDGE
- HDECAY
- NMHDECAY⋆
- SDECAY

Event Generators

- [Isajet]
- Herwig++
- Pythia
- Sherpa

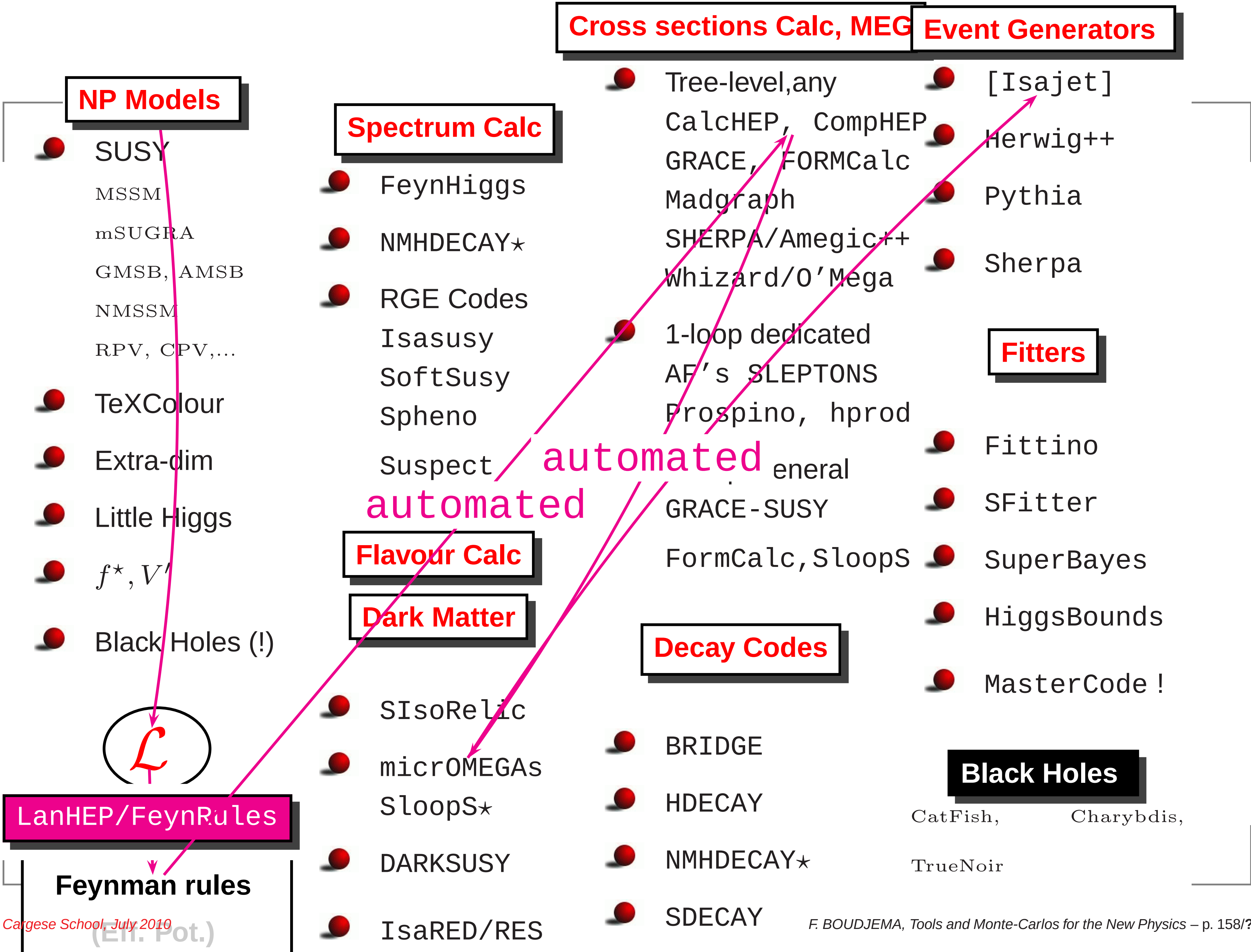
Fitters

- Fittino
- SFitter
- SuperBayes
- HiggsBounds
- MasterCode!

Black Holes

CatFish, Charybdis,

TrueNoir



NP Models

- SUSY
  - MSSM
  - mSUGRA
  - GMSB, AMSB
  - NMSSM
  - RPV, CPV,...
- TeXColour
- Extra-dim
- Little Higgs
- $f^\star, V'$
- Black Holes (!)

$\mathcal{L}$

LanHEP/FeynRules

Feynman rules

(Eff. Pot.)

Spectrum Calc

- FeynHiggs
- NMHDECAY⋆
- RGE Codes
  Isasusy
  SoftSusy
  Spheno
  Suspect

automated

Flavour Calc

Dark Matter

- SIsoRelic
- micrOMEGAs
  SloopS⋆
- DARKSUSY
- IsaRED/RES

Cross sections Calc, MEG

- Tree-level,any
  CalcHEP, CompHEP
  GRACE, FORMCalc
  Madgraph
  SHERPA/Amegic++
  Whizard/O'Mega
- 1-loop dedicated
  AF's SLEPTONS
  Prospino, hprod

automated eneral
GRACE-SUSY
FormCalc,SloopS

Decay Codes

- BRIDGE
- HDECAY
- NMHDECAY⋆
- SDECAY

Event Generators

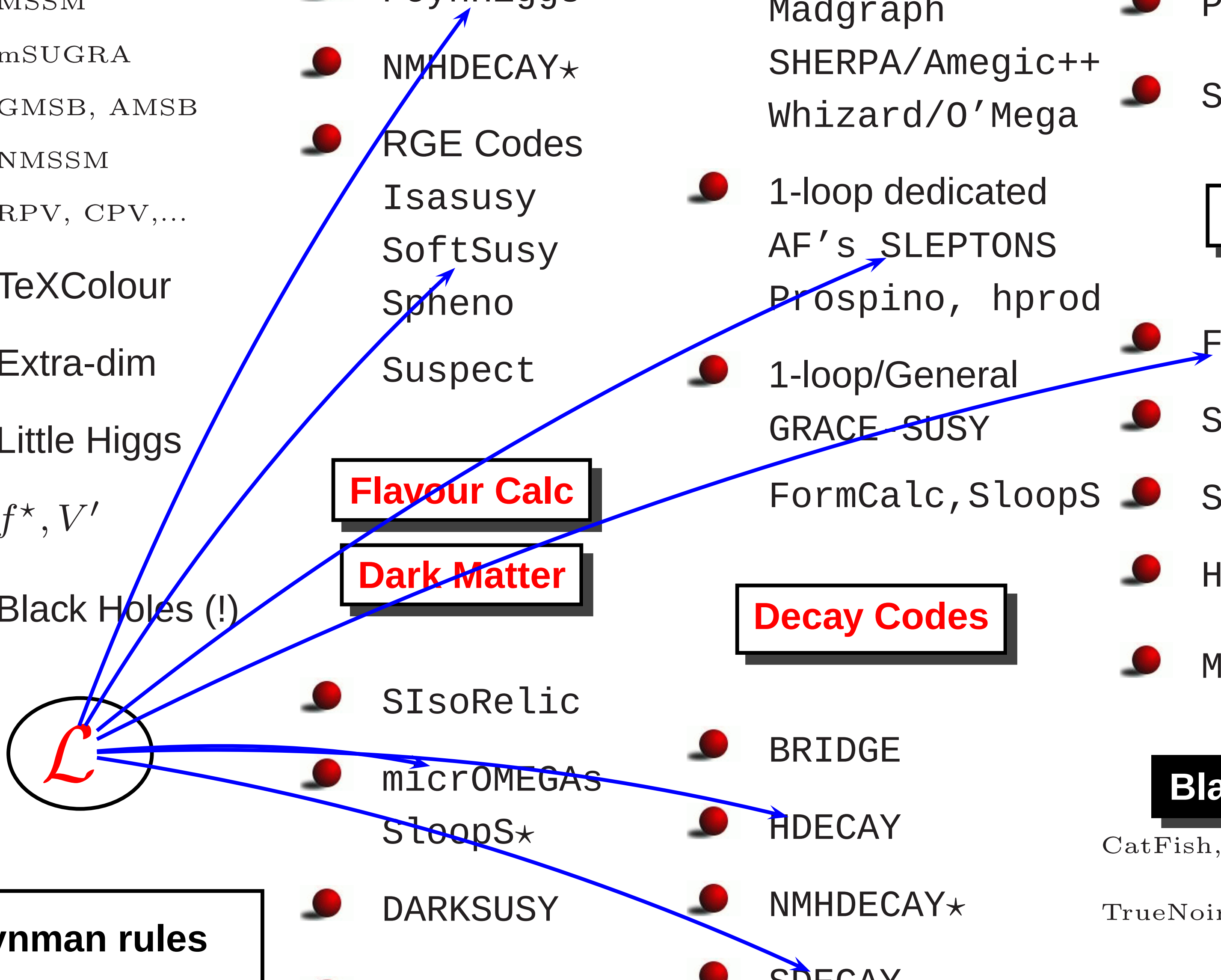
- [Isajet]
- Herwig++
- Pythia
- Sherpa

Fitters

- Fittino
- SFitter
- SuperBayes
- HiggsBounds
- MasterCode !

Black Holes

CatFish, Charybdis, TrueNoir



NP Models

- SUSY
  - MSSM
  - mSUGRA
  - GMSB, AMSB
  - NMSSM
  - RPV, CPV,...
- TeXColour
- Extra-dim
- Little Higgs
- $f^\star, V'$
- Black Holes (!)

Spectrum Calc

- `FeynHiggs`
- `NMHDECAY`$\star$
- RGE Codes
  `Isasusy`
  `SoftSusy`
  `Spheno`
  `Suspect`

Flavour Calc

Dark Matter

- `SIsoRelic`
- `micrOMEGAs`
  `SloopS`$\star$
- `DARKSUSY`
- `IsaRED/RES`

Cross sections Calc, MEG

- Tree-level,any
  `CalcHEP, CompHEP`
  `GRACE, FORMCalc`
  `Madgraph`
  `SHERPA/Amegic++`
  `Whizard/O'Mega`
- 1-loop dedicated
  `AF's SLEPTONS`
  `Prospino, hprod`
- 1-loop/General
  `GRACE-SUSY`
  `FormCalc,SloopS`

Decay Codes

- `BRIDGE`
- `HDECAY`
- `NMHDECAY`$\star$
- `SDECAY`

Event Generators

- `[Isajet]`
- `Herwig++`
- `Pythia`
- `Sherpa`

Fitters

- `Fittino`
- `SFitter`
- `SuperBayes`
- `HiggsBounds`
- `MasterCode!`

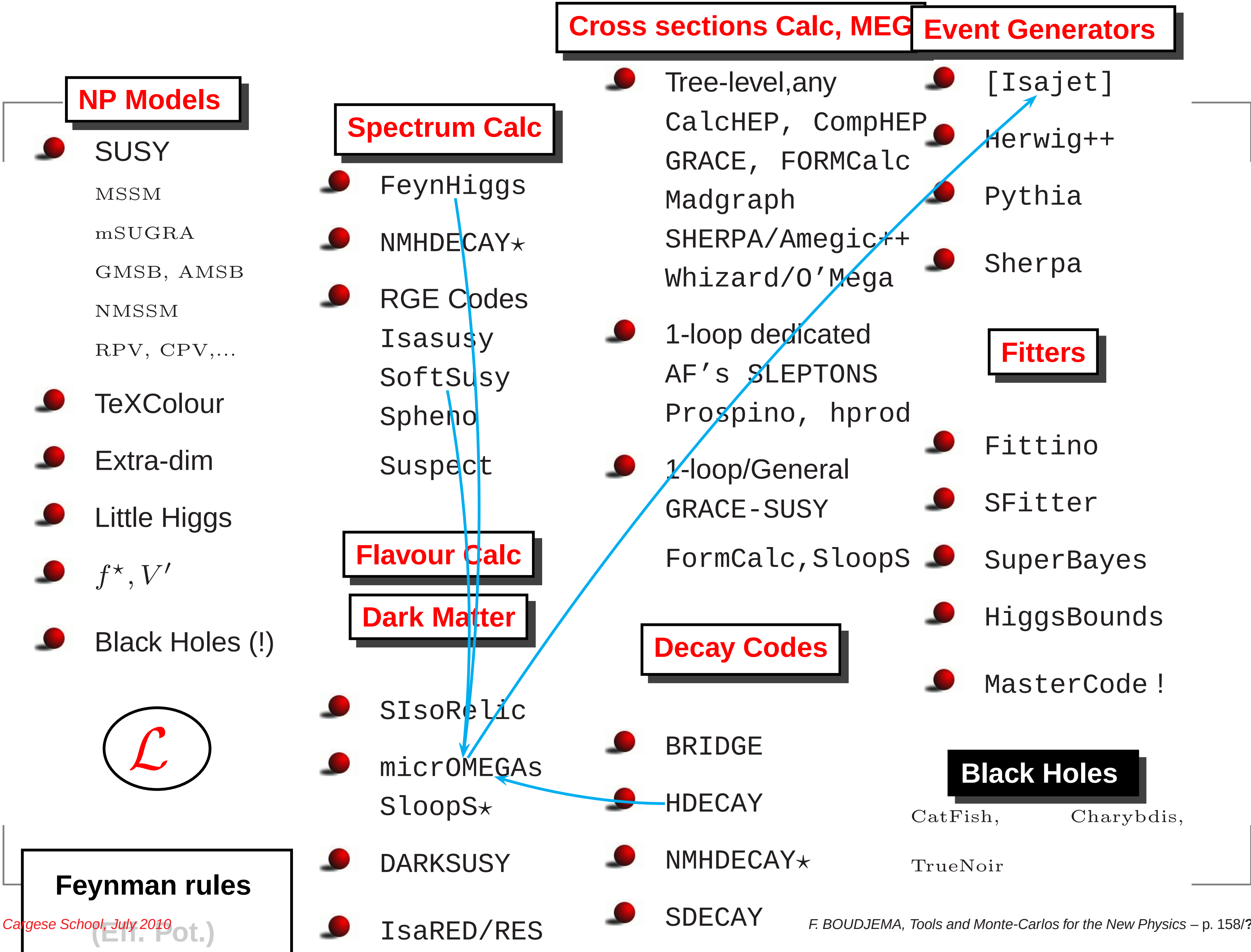
Black Holes

CatFish, Charybdis,

TrueNoir

Feynman rules

(Eff. Pot.)



NP Models

- SUSY
  - MSSM
  - mSUGRA
  - GMSB, AMSB
  - NMSSM
  - RPV, CPV,...
- TeXColour
- Extra-dim
- Little Higgs
- $f^\star, V'$
- Black Holes (!)

$\mathcal{L}$

Feynman rules

(Eff. Pot.)

Spectrum Calc

- `FeynHiggs`
- `NMHDECAY⋆`
- RGE Codes
  `Isasusy`
  `SoftSusy`
  `Spheno`
  `Suspect`

Flavour Calc

Dark Matter

- `SIsoRelic`
- `micrOMEGAs`
  `SloopS⋆`
- `DARKSUSY`
- `IsaRED/RES`

Cross sections Calc, MEG

- Tree-level,any
  `CalcHEP, CompHEP`
  `GRACE, FORMCalc`
  `Madgraph`
  `SHERPA/Amegic++`
  `Whizard/O'Mega`
- 1-loop dedicated
  `AF's SLEPTONS`
  `Prospino, hprod`
- 1-loop/General
  `GRACE-SUSY`
  `FormCalc,SloopS`

Decay Codes

- `BRIDGE`
- `HDECAY`
- `NMHDECAY⋆`
- `SDECAY`

Event Generators

- `[Isajet]`
- `Herwig++`
- `Pythia`
- `Sherpa`

Fitters

- `Fittino`
- `SFitter`
- `SuperBayes`
- `HiggsBounds`
- `MasterCode!`

Black Holes

CatFish, Charybdis,

TrueNoir

NP Models

- SUSY
  - MSSM
  - mSUGRA
  - GMSB, AMSB
  - NMSSM
  - RPV, CPV,...
- TeXColour
- Extra-dim
- Little Higgs
- $f^{\star}, V'$
- Black Holes (!)

$\mathcal{L}$

Feynman rules

(Eff. Pot.)

Spectrum Calc

- `FeynHiggs`
- `NMHDECAY`$\star$
- RGE Codes
  - `Isasusy`
  - `SoftSusy`
  - `Spheno`
  - `Suspect`

Flavour Calc

Dark Matter

- `SIsoRelic`
- `micrOMEGAs`
  `SloopS`$\star$
- `DARKSUSY`
- `IsaRED/RES`

Cross sections Calc, MEG

- Tree-level,any
  - `CalcHEP, CompHEP`
  - `GRACE, FORMCalc`
  - `Madgraph`
  - `SHERPA/Amegic++`
  - `Whizard/O'Mega`
- 1-loop dedicated
  - `AF's SLEPTONS`
  - `Prospino, hprod`
- 1-loop/General
  - `GRACE-SUSY`
  - `FormCalc,SloopS`

Decay Codes

- `BRIDGE`
- `HDECAY`
- `NMHDECAY`$\star$
- `SDECAY`

Event Generators

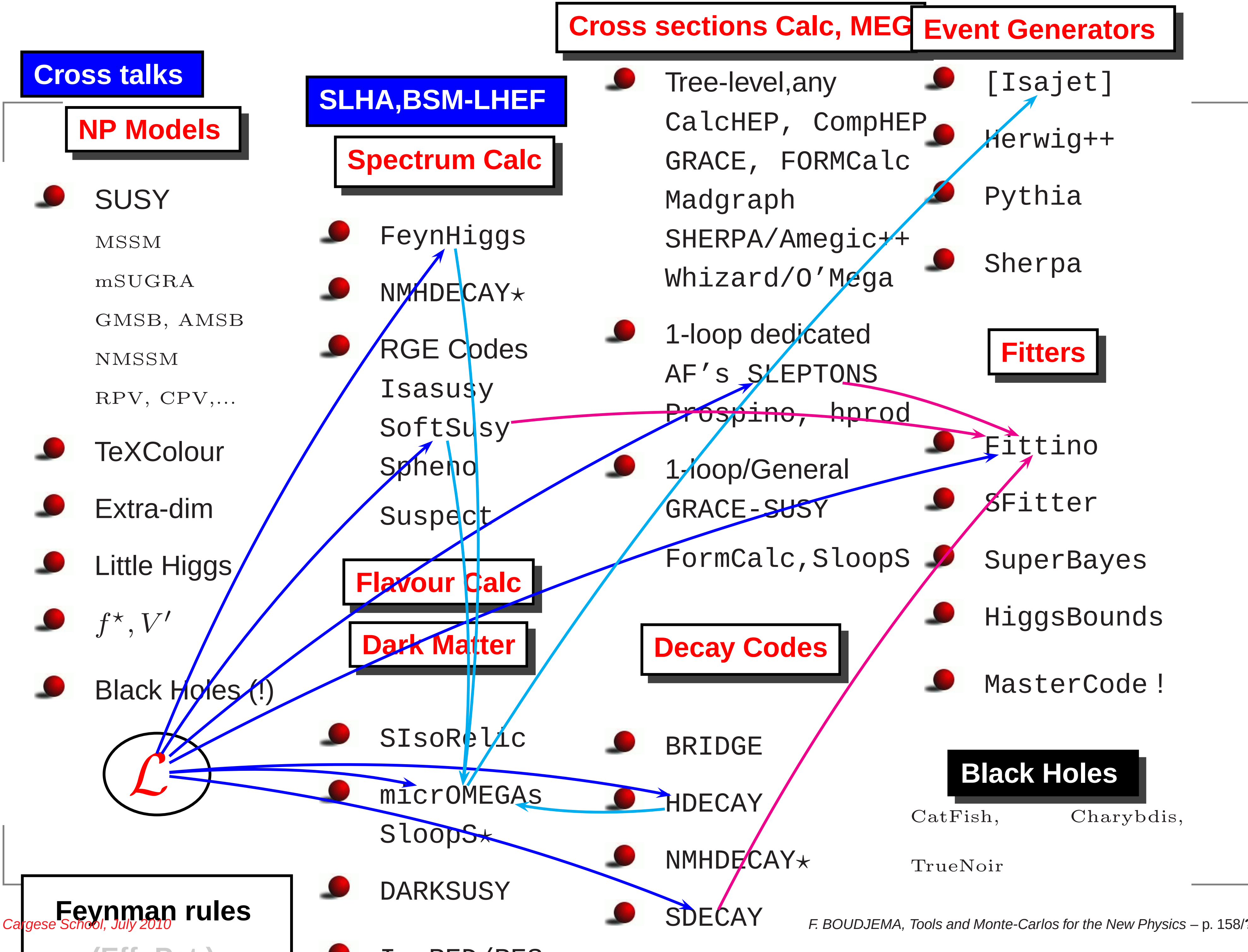
- `[Isajet]`
- `Herwig++`
- `Pythia`
- `Sherpa`

Fitters

- `Fittino`
- `SFitter`
- `SuperBayes`
- `HiggsBounds`
- `MasterCode!`

Black Holes

CatFish, Charybdis, TrueNoir



Cross talks

SLHA,BSM-LHEF

NP Models

- SUSY
  - MSSM
  - mSUGRA
  - GMSB, AMSB
  - NMSSM
  - RPV, CPV,...
- TeXColour
- Extra-dim
- Little Higgs
- $f^\star, V'$
- Black Holes (!)

Feynman rules

Spectrum Calc

- `FeynHiggs`
- `NMHDECAY⋆`
- RGE Codes
  `Isasusy`
  `SoftSusy`
  `Spheno`
  `Suspect`

Flavour Calc

Dark Matter

- `SIsoRelic`
- `micrOMEGAs`
  `SloopS⋆`
- `DARKSUSY`

Cross sections Calc, MEG

- Tree-level,any
  `CalcHEP, CompHEP`
  `GRACE, FORMCalc`
  `Madgraph`
  `SHERPA/Amegic++`
  `Whizard/O'Mega`
- 1-loop dedicated
  `AF's SLEPTONS`
  `Prospino, hprod`
- 1-loop/General
  `GRACE-SUSY`
  `FormCalc,SloopS`

Decay Codes

- `BRIDGE`
- `HDECAY`
- `NMHDECAY⋆`
- `SDECAY`

Event Generators

- `[Isajet]`
- `Herwig++`
- `Pythia`
- `Sherpa`

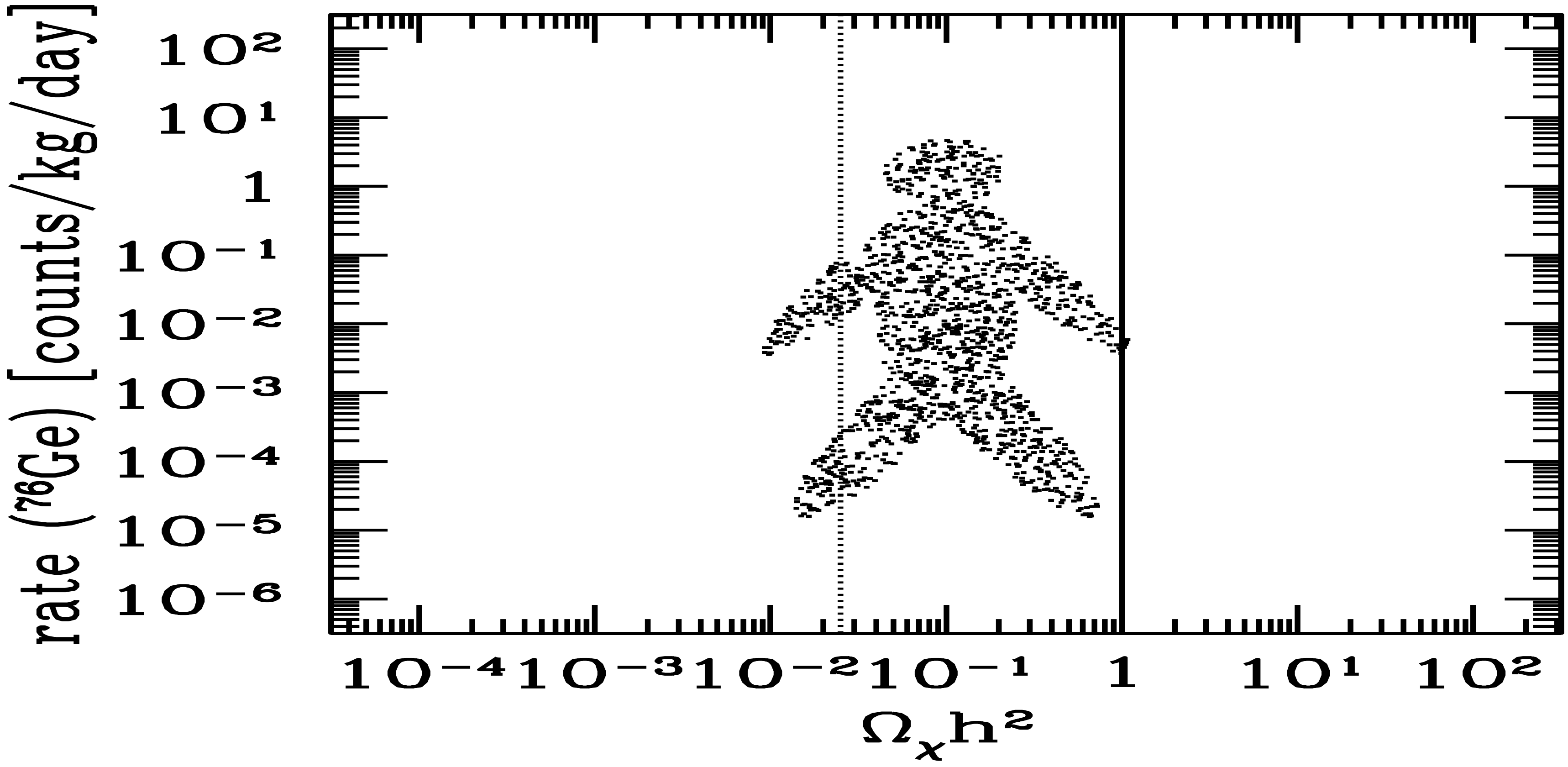
Fitters

- `Fittino`
- `SFitter`
- `SuperBayes`
- `HiggsBounds`
- `MasterCode!`

Black Holes

CatFish, Charybdis, TrueNoir

Weather forecasts, MCMC priors and posteriors



NLO and better SM (and BSM) Tools

- Most of what was discussed was based on Feynman graphs
- I have not said much about MC at NLO
- NLO is essential
- Intense activity these last few years in NLO multi-leg
- many new techniques, string inspired to SYM/Wilson Loops and integrability connection though most of it conformal
- plans for $2 \to 3, 4$ and benchmark cross sections

Yuri Dokhsitzer: "virtual SUSY is helping QCD (*twistor techniques!*), QCD will pay back discovering *"real"* SUSY

BSM Tools Repository

- http://www.ippp.dur.ac.uk/montecarlo/BSM

  Please submit your code or get a code from there

  otherwise google the codes I have described

  If you contribute a code make it SLHA/LHEF compliant, if SLHA exists for the model please give a description of the code: what physics there is inside not just how to run it!!

  at the moment about 50 BSM tools listed so far...
- other repositories, e.g. http://mcelrath.org/Notes/Software (see also open directory project)
- For codes that do the same things (or supposed to do the same thing) Comparison page like what is done with RGE (see Sabine Kraml's page)

Organise round-tables involving model builders, calculation theorists, experimentalists

- More work on New Physics which is not SUSY
- for some SUSY models, probably need “background tools”: contact with SM/QCD tools
- experimentalists need to speak up and ask what is needed most urgently : priority list (similar to what has been done for SM in Les Houches)
- how should codes be interfaced and written: modules, C++, SLHA,LHEF
- go to the Monte Carlo Schools and or the SUSY-BSM tools

Progress/Conclusions

- A lot of progress and a lot of tools
- more and more on modularity and exchange of modules
- much easier now to contribute a new model
- Flexibility is the key
  - [-] Need to be ready to implement a model quickly
  - [-] Check output with different ME Calc./MC/MG
- This is now possible, while earlier even parameters of simple models hard wired, model implementation needed experts
- Now many tools automatize the different steps and as long as
  - [-] particles has spin ≤ 2
  - [-] Standard couplings through known Lorentz structures, this precludes higher order operators in some MEG but ALOHA is on the way and CalcHEP/CompHEP are ready...
  - [-] decay chain does not end up in higher order or unusual colour representations

End

That's all folks

do go through the notes

and send typos to

boudjema@lapp.in2p3.fr