WHiZard & O'Mega

Thorsten Ohl

Institute for Theoretical Physics and Astrophysics
University of Würzburg

LoopFest V
SLAC, June 20th, 2006

- preaching to the choir:

- preaching to the choir:
  - ∵ . . . even if we had enough graduate students and postdocs, we should not waste them on repetitive “assembly line” calculations

- preaching to the choir:
  - ∵ . . . even if we had enough graduate students and postdocs, we should not waste them on repetitive “assembly line” calculations
  - ∴ formalize the calculations so that the repetitive part can be delegated to patient computers. Ideally:

$$\left\{\begin{array}{r}\text{Lagrangian, parameters}\\ \text{final state, cuts}\end{array}\right\} \Longrightarrow \text{unweighted event generator}$$

- preaching to the choir:
  - $\because$ ... even if we had enough graduate students and postdocs, we should not waste them on repetitive “assembly line” calculations
  - $\therefore$ formalize the calculations so that the repetitive part can be delegated to patient computers. Ideally:

$$\left\{\begin{array}{r}\text{Lagrangian, parameters}\\ \text{final state, cuts}\end{array}\right\} \Longrightarrow \text{unweighted event generator}$$

- ingredients:

- preaching to the choir:
  - $\because$ . . . even if we had enough graduate students and postdocs, we should not waste them on repetitive “assembly line” calculations
  - $\therefore$ formalize the calculations so that the repetitive part can be delegated to patient computers. Ideally:

$$\left\{\begin{array}{r}\text{Lagrangian, parameters}\\ \text{final state, cuts}\end{array}\right\} \Longrightarrow \text{unweighted event generator}$$

- ingredients:
  - matrix element generation

- preaching to the choir:
  - $\because$ … even if we had enough graduate students and postdocs, we should not waste them on repetitive "assembly line" calculations
  - $\therefore$ formalize the calculations so that the repetitive part can be delegated to patient computers. Ideally:

$$\left\{\begin{array}{r}\text{Lagrangian, parameters}\\ \text{final state, cuts}\end{array}\right\} \Longrightarrow \text{unweighted event generator}$$

- ingredients:
  - matrix element generation
  - parameter handling

- preaching to the choir:
  - $\because$ . . . even if we had enough graduate students and postdocs, we should not waste them on repetitive "assembly line" calculations
  - $\therefore$ formalize the calculations so that the repetitive part can be delegated to patient computers. Ideally:

$$\left\{\begin{array}{r}\text{Lagrangian, parameters}\\ \text{final state, cuts}\end{array}\right\} \Longrightarrow \text{unweighted event generator}$$

- ingredients:
  - matrix element generation
  - parameter handling
  - phase space parametrization

- preaching to the choir:
  - $\because$ ... even if we had enough graduate students and postdocs, we should not waste them on repetitive "assembly line" calculations
  - $\therefore$ formalize the calculations so that the repetitive part can be delegated to patient computers. Ideally:

$$\left\{\begin{array}{r}\text{Lagrangian, parameters}\\ \text{final state, cuts}\end{array}\right\} \Longrightarrow \text{unweighted event generator}$$

- ingredients:
  - matrix element generation
  - parameter handling
  - phase space parametrization
  - integration and sampling

- preaching to the choir:
  - $\because$ ... even if we had enough graduate students and postdocs, we should not waste them on repetitive "assembly line" calculations
  - $\therefore$ formalize the calculations so that the repetitive part can be delegated to patient computers. Ideally:

$$\left\{\begin{array}{r}\text{Lagrangian, parameters}\\ \text{final state, cuts}\end{array}\right\} \Longrightarrow \text{unweighted event generator}$$

- ingredients:
  - matrix element generation
  - parameter handling
  - phase space parametrization
  - integration and sampling
  - cuts and (simple) analysis

- preaching to the choir:
  - $\because$ ... even if we had enough graduate students and postdocs, we should not waste them on repetitive “assembly line” calculations
  - $\therefore$ formalize the calculations so that the repetitive part can be delegated to patient computers. Ideally:

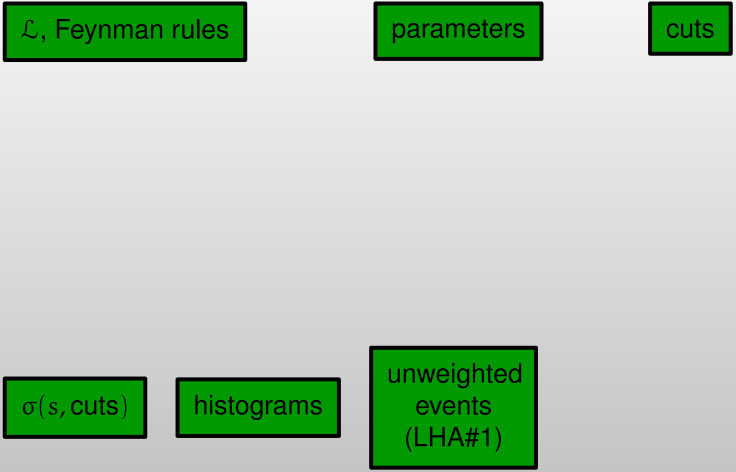
$$\left\{\begin{array}{r}\text{Lagrangian, parameters}\\ \text{final state, cuts}\end{array}\right\} \Longrightarrow \text{unweighted event generator}$$

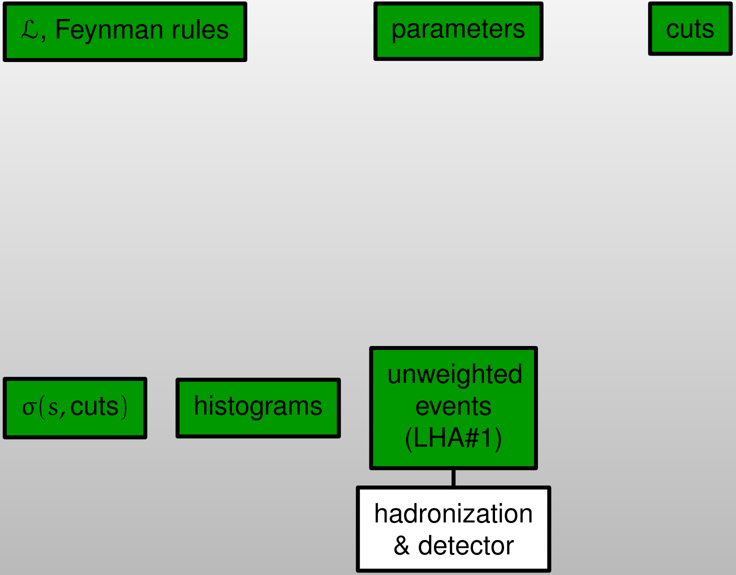
- ingredients:
  - matrix element generation
  - parameter handling
  - phase space parametrization
  - integration and sampling
  - cuts and (simple) analysis
  - interfacing to the rest of the world

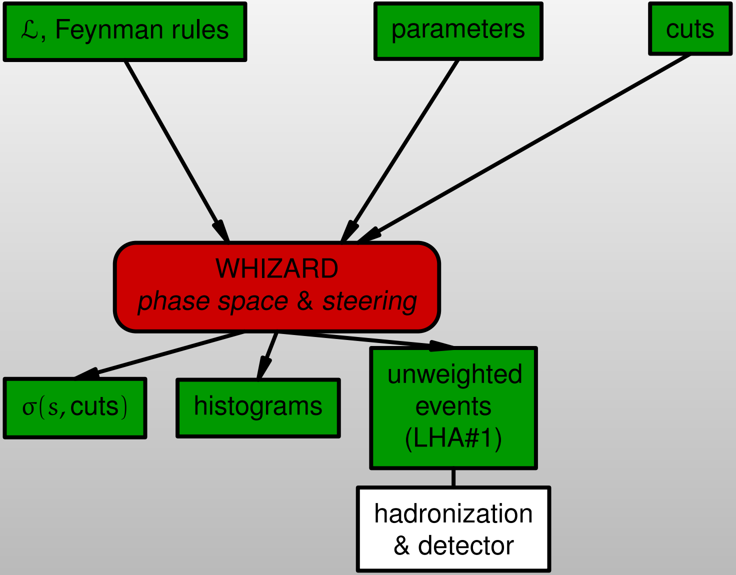
$\mathcal{L}$, Feynman rules

parameters

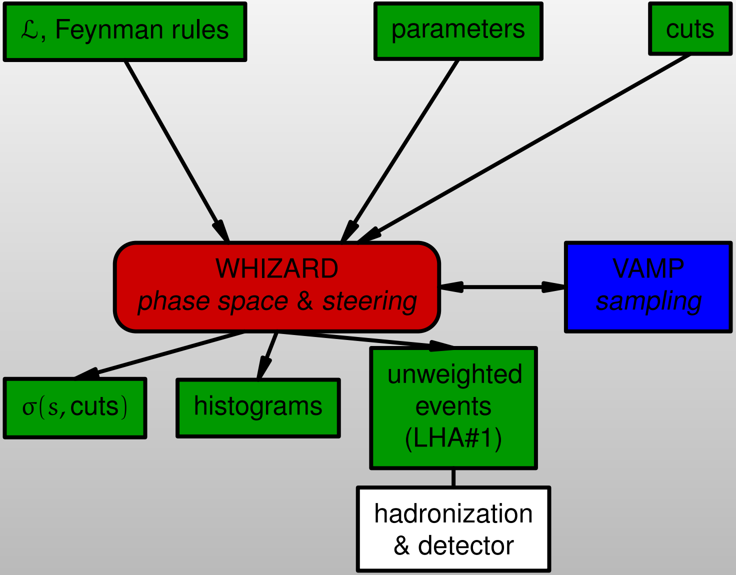
cuts


$\mathcal{L}$, Feynman rules
parameters
cuts
$\sigma(s, \text{cuts})$
histograms
unweighted events (LHA#1)


$\mathcal{L}$, Feynman rules
parameters
cuts
$\sigma(s, \text{cuts})$
histograms
unweighted events (LHA#1)
hadronization & detector

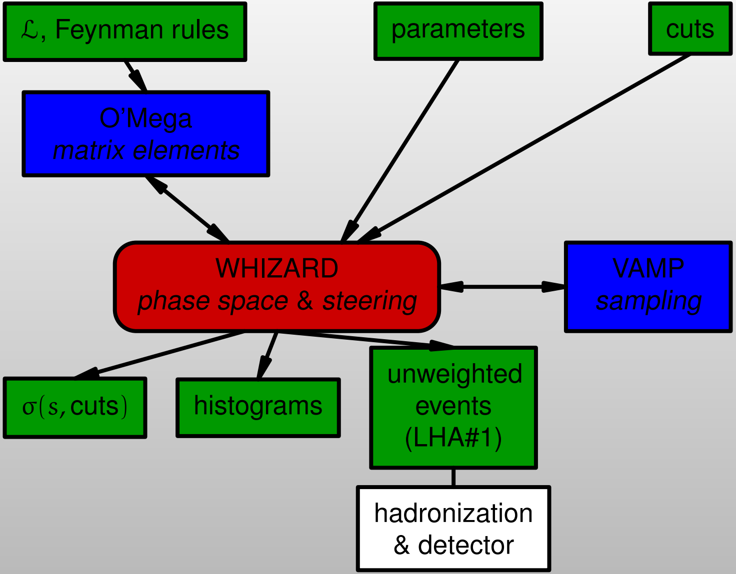

$\mathcal{L}$, Feynman rules
parameters
cuts
WHIZARD
phase space & steering
$\sigma(s, \text{cuts})$
histograms
unweighted events (LHA#1)
hadronization & detector

Kilian



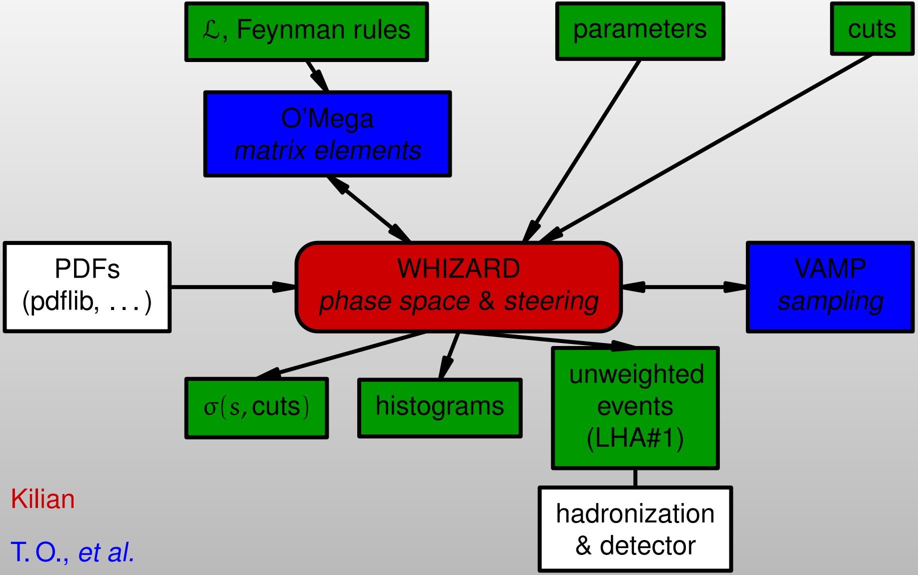
Kilian

T. O., *et al.*



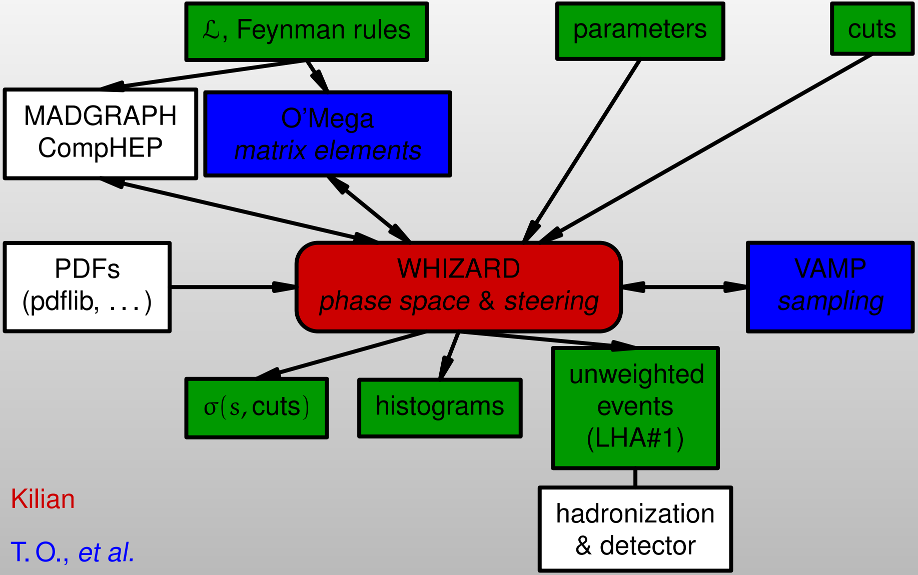
Kilian

T. O., *et al.*



Kilian

T. O., *et al.*



Kilian

T. O., *et al.*

The number of tree Feynman diagrams w/ n legs in vanilla ϕ^3-theory grows like a factorial

$$F(n) = (2n-5)!! = (2n-5)\cdot(2n-7)\cdot\ldots\cdot3\cdot1$$

n		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		

The number of tree Feynman diagrams w/ n legs in vanilla ϕ^3-theory grows like a factorial

$$F(n) = (2n-5)!! = (2n-5)\cdot(2n-7)\cdot\ldots\cdot 3\cdot 1$$

n	$F(n)$	
4	3	
5	15	
6	105	
7	945	
8	10395	
9	135135	
10	2027025	
11	34459425	
12	654729075	
13	13749310575	
14	316234143225	

The number of tree Feynman diagrams w/ n legs in vanilla ϕ^3-theory grows like a factorial

$$F(n) = (2n-5)!! = (2n-5) \cdot (2n-7) \cdot \ldots \cdot 3 \cdot 1$$

n	$F(n)$	
4	3	
5	15	
6	105	
7	945	
8	10395	
9	135135	
10	2027025	
11	34459425	
12	654729075	
13	13749310575	
14	316234143225	

☹ computational costs grow beyond all reasonable limits

The number of tree Feynman diagrams w/ n legs in vanilla ϕ^3-theory grows like a factorial

$$F(n) = (2n-5)!! = (2n-5) \cdot (2n-7) \cdot \ldots \cdot 3 \cdot 1$$

n	$F(n)$	
4	3	
5	15	
6	105	
7	945	
8	10395	
9	135135	
10	2027025	
11	34459425	
12	654729075	
13	13749310575	
14	316234143225	

- ☹ computational costs grow beyond all reasonable limits
- ☹ gauge cancellations cause loss of precision

The number of tree Feynman diagrams w/ n legs in vanilla ϕ^3-theory grows like a factorial

$$F(n) = (2n-5)!! = (2n-5) \cdot (2n-7) \cdot \ldots \cdot 3 \cdot 1$$

n	$F(n)$	
4	3	
5	15	
6	105	
7	945	
8	10395	
9	135135	
10	2027025	
11	34459425	
12	654729075	
13	13749310575	
14	316234143225	

- ☹ computational costs grow beyond all reasonable limits
- ☹ gauge cancellations cause loss of precision

Number of possible momenta in tree diagrams grows only exponentially

$$P(n) = \frac{2^n - 2}{2} - n = 2^{n-1} - n - 1$$

The number of tree Feynman diagrams w/ n legs in vanilla ϕ^3-theory grows like a factorial

$$F(n) = (2n-5)!! = (2n-5)\cdot(2n-7)\cdot\ldots\cdot 3\cdot 1$$

n	$F(n)$	$P(n)$
4	3	3
5	15	10
6	105	25
7	945	56
8	10395	119
9	135135	246
10	2027025	501
11	34459425	1012
12	654729075	2035
13	13749310575	4082
14	316234143225	8177

- computational costs grow beyond all reasonable limits
- gauge cancellations cause loss of precision

Number of possible momenta in tree diagrams grows only exponentially

$$P(n) = \frac{2^n-2}{2} - n = 2^{n-1} - n - 1$$

The number of tree Feynman diagrams w/ n legs in vanilla ϕ^3-theory grows like a factorial

$$F(n) = (2n-5)!! = (2n-5)\cdot(2n-7)\cdot\ldots\cdot 3\cdot 1$$

n	$F(n)$	$P(n)$
4	3	3
5	15	10
6	105	25
7	945	56
8	10395	119
9	135135	246
10	2027025	501
11	34459425	1012
12	654729075	2035
13	13749310575	4082
14	316234143225	8177

- ☹ computational costs grow beyond all reasonable limits
- ☹ gauge cancellations cause loss of precision

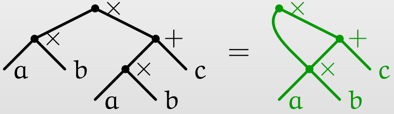
Number of possible momenta in tree diagrams grows only exponentially

$$P(n) = \frac{2^n-2}{2} - n = 2^{n-1} - n - 1$$

∴ Feynman diagrams redundant for many external particles!

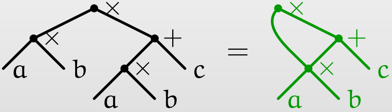
∴ Replace the forest of tree diagrams by the Directed Acyclical Graph (DAG) of the algebraic expression.

$$ab(ab+c) =$$

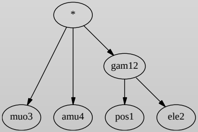


∴ Replace the forest of tree diagrams by the Directed Acyclical Graph (DAG) of the algebraic expression.

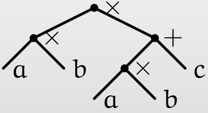
$$ab(ab+c) =$$

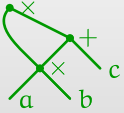


- simplest examples: $e^+e^- \to \mu^+\mu^-$, and (only QED)

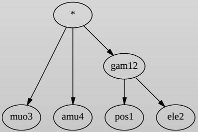


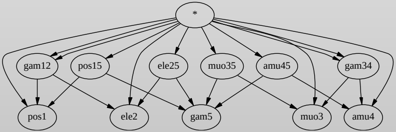
∴ Replace the forest of tree diagrams by the Directed Acyclical Graph (DAG) of the algebraic expression.

$ab(ab+c) =$ 

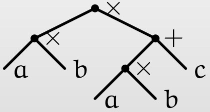
$=$ 

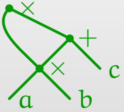
- simplest examples: $e^+e^- \to \mu^+\mu^-$, $e^+e^- \to \mu^+\mu^-\gamma$ and (only QED)



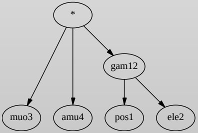


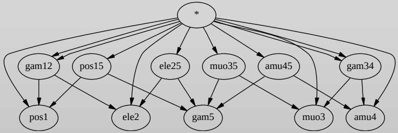
∴ Replace the forest of tree diagrams by the Directed Acyclical Graph (DAG) of the algebraic expression.

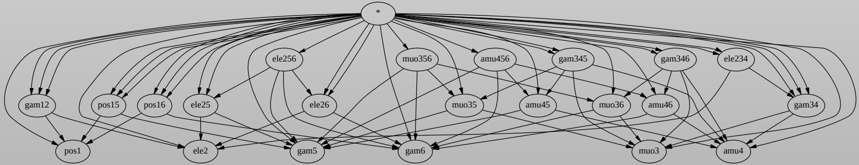
$ab(ab+c) =$ 

$=$ 

- simplest examples: $e^+e^- \to \mu^+\mu^-$, $e^+e^- \to \mu^+\mu^-\gamma$ and $e^+e^- \to \mu^+\mu^-\gamma\gamma$ (only QED)







Prior Art:

- HELAS / Madgraph [Hagiwara et al. / Stelzer et al.], AMEGIC++ [Krauss et al.]:
  - some common subexpression elimination
  - ∴ partial elimination of redundancy

Prior Art:

- HELAS / Madgraph [Hagiwara et al. / Stelzer et al.], AMEGIC++ [Krauss et al.]:
  - some common subexpression elimination
  - ∴ partial elimination of redundancy
- ALPHA [Caravaglios & Moretti]:
  - tree level scattering amplitude is Legendre transform of Lagragian
  - ☺ can be performed numerically, using only $P^*(n)$ independent variables

Prior Art:

- HELAS / Madgraph [Hagiwara et al. / Stelzer et al.], AMEGIC++ [Krauss et al.]:
  - some common subexpression elimination
  - ∴ partial elimination of redundancy
- ALPHA [Caravaglios & Moretti]:
  - tree level scattering amplitude is Legendre transform of Lagragian
  - ☺ can be performed numerically, using only $P^*(n)$ independent variables
- HELAC [Kanaki & Papadopoulos]:
  - ALPHA algorithm can be reformulated as recursive numerical solution of Schwinger-Dyson equations

Prior Art:

- HELAS / Madgraph [Hagiwara et al. / Stelzer et al.], AMEGIC++ [Krauss et al.]:
  - some common subexpression elimination
  - ∴ partial elimination of redundancy
- ALPHA [Caravaglios & Moretti]:
  - tree level scattering amplitude is Legendre transform of Lagragian
  - ☺ can be performed numerically, using only $P^*(n)$ independent variables
- HELAC [Kanaki & Papadopoulos]:
  - ALPHA algorithm can be reformulated as recursive numerical solution of Schwinger-Dyson equations

Shortcomings:

Prior Art:

- HELAS / Madgraph [Hagiwara et al. / Stelzer et al.], AMEGIC++ [Krauss et al.]:
  - some common subexpression elimination
  - ∴ partial elimination of redundancy
- ALPHA [Caravaglios & Moretti]:
  - tree level scattering amplitude is Legendre transform of Lagragian
  - ☺ can be performed numerically, using only $P^*(n)$ independent variables
- HELAC [Kanaki & Papadopoulos]:
  - ALPHA algorithm can be reformulated as recursive numerical solution of Schwinger-Dyson equations

Shortcomings:

- ALPHA and HELAC strictly numerical
  - ☹ perturbative renormalization machinery unavailable

Prior Art:

- HELAS / Madgraph [Hagiwara et al. / Stelzer et al.], AMEGIC++ [Krauss et al.]:
  - some common subexpression elimination
  - ∴ partial elimination of redundancy
- ALPHA [Caravaglios & Moretti]:
  - tree level scattering amplitude is Legendre transform of Lagragian
  - ☺ can be performed numerically, using only $P^*(n)$ independent variables
- HELAC [Kanaki & Papadopoulos]:
  - ALPHA algorithm can be reformulated as recursive numerical solution of Schwinger-Dyson equations

Shortcomings:

- ALPHA and HELAC strictly numerical
  - ☹ perturbative renormalization machinery unavailable
- hardcoded Lagragrians
  - ☹ difficult to extent by users

Prior Art:

- HELAS / Madgraph [Hagiwara et al. / Stelzer et al.], AMEGIC++ [Krauss et al.]:
  - some common subexpression elimination
  - ∴ partial elimination of redundancy
- ALPHA [Caravaglios & Moretti]:
  - tree level scattering amplitude is Legendre transform of Lagragian
  - ☺ can be performed numerically, using only $P^*(n)$ independent variables
- HELAC [Kanaki & Papadopoulos]:
  - ALPHA algorithm can be reformulated as recursive numerical solution of Schwinger-Dyson equations

Shortcomings:

- ALPHA and HELAC strictly numerical
  - ☹ perturbative renormalization machinery unavailable
- hardcoded Lagragrians
  - ☹ difficult to extent by users (sometimes rather ☺...)

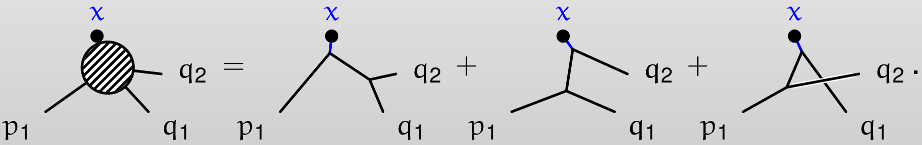
One particle off-shell wave functions (1POWs) are obtained from by applying the LSZ reduction formula to all but one line:

$$W(x; p_1, \ldots, p_n; q_1, \ldots, q_m) = \langle \phi(q_1), \ldots, \phi(q_m); \text{out} | \Phi(x) | \phi(p_1), \ldots, \phi(p_n); \text{in} \rangle \; .$$

One particle off-shell wave functions (1POWs) are obtained from by applying the LSZ reduction formula to all but one line:

$$W(x; p_1, \ldots, p_n; q_1, \ldots, q_m) = \langle \phi(q_1), \ldots, \phi(q_m); \text{out} | \Phi(x) | \phi(p_1), \ldots, \phi(p_n); \text{in} \rangle \,.$$

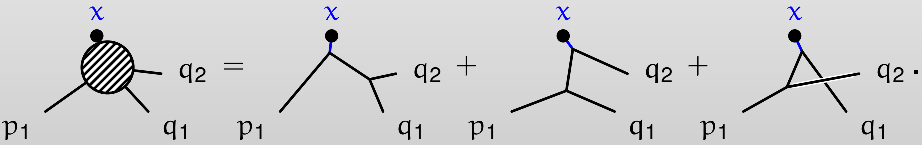
E. g. $\langle \phi(q_1), \phi(q_2); \text{out} | \Phi(x) | \phi(p_1); \text{in} \rangle$ in ϕ^3-theory at tree level



One particle off-shell wave functions (1POWs) are obtained from by applying the LSZ reduction formula to all but one line:

$$W(x;p_1,\ldots,p_n;q_1,\ldots,q_m) = \langle\phi(q_1),\ldots,\phi(q_m);\text{out}|\Phi(x)|\phi(p_1),\ldots,\phi(p_n);\text{in}\rangle\ .$$

E. g. $\langle\phi(q_1),\phi(q_2);\text{out}|\Phi(x)|\phi(p_1);\text{in}\rangle$ in ϕ^3-theory at tree level

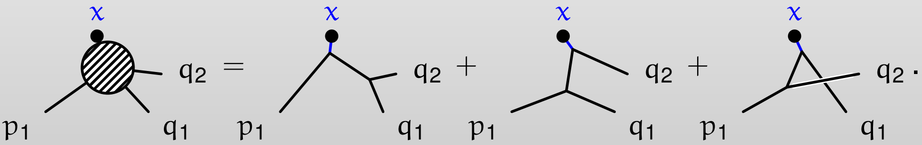


- the set of all 1POWs at tree level grows exponentially and can be constructed recursively from other 1POWs at tree level.

One particle off-shell wave functions (1POWs) are obtained from by applying the LSZ reduction formula to all but one line:

$$W(x;p_1,\ldots,p_n;q_1,\ldots,q_m) = \langle \phi(q_1),\ldots,\phi(q_m);\mathrm{out}|\Phi(x)|\phi(p_1),\ldots,\phi(p_n);\mathrm{in}\rangle \,.$$

E. g. $\langle \phi(q_1),\phi(q_2);\mathrm{out}|\Phi(x)|\phi(p_1);\mathrm{in}\rangle$ in ϕ^3-theory at tree level



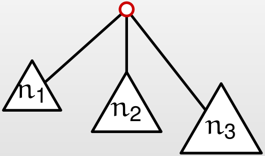
- the set of all 1POWs at tree level grows exponentially and can be constructed recursively from other 1POWs at tree level.

There exists a well defined set of keystones K that allow to express the sum of Feynman diagrams through 1POWs:

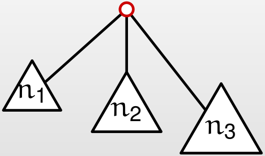
$$T = \sum_{i=1}^{F(n)} D_i = \sum_{k,l,m=1}^{P(n)} K^3_{f_k f_l f_m}(p_k,p_l,p_m) W_{f_k}(p_k) W_{f_l}(p_l) W_{f_m}(p_m)$$

Non-trivial: construction of inequivalent topologies for merging off-shell amplitudes to Feynman diagrams.

Non-trivial: construction of inequivalent topologies for merging off-shell amplitudes to Feynman diagrams. Solution: inequivalent partitionings of external momenta for a fixed vertex

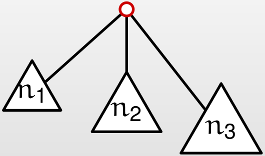


Non-trivial: construction of inequivalent topologies for merging off-shell amplitudes to Feynman diagrams. Solution: inequivalent partitionings of external momenta for a fixed vertex



n	$\sum$	$\sum$
4	4	$1 \cdot (1,1,1,1) + 3 \cdot (1,1,2)$
5	26	$1 \cdot (1,1,1,1,1) + 10 \cdot (1,1,1,2) + 15 \cdot (1,2,2)$
6	236	$1 \cdot (1,1,1,1,1,1) + 15 \cdot (1,1,1,1,2) + 40 \cdot (1,1,1,3)$ $+ 45 \cdot (1,1,2,2) + 120 \cdot (1,2,3) + 15 \cdot (2,2,2)$

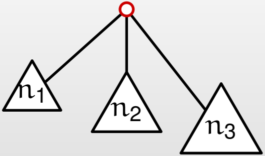
Non-trivial: construction of inequivalent topologies for merging off-shell amplitudes to Feynman diagrams. Solution: inequivalent partitionings of external momenta for a fixed vertex



n	$\sum$	$\sum$
4	4	$1 \cdot (1,1,1,1) + 3 \cdot (1,1,2)$
5	26	$1 \cdot (1,1,1,1,1) + 10 \cdot (1,1,1,2) + 15 \cdot (1,2,2)$
6	236	$1 \cdot (1,1,1,1,1,1) + 15 \cdot (1,1,1,1,2) + 40 \cdot (1,1,1,3)$
		$+ 45 \cdot (1,1,2,2) + 120 \cdot (1,2,3) + 15 \cdot (2,2,2)$

Subtlety: some partitions for an even number of external lines are degenerate

Non-trivial: construction of inequivalent topologies for merging off-shell amplitudes to Feynman diagrams. Solution: inequivalent partitionings of external momenta for a fixed vertex



n	$\sum$	$\sum$
4	4	$1 \cdot (1,1,1,1) + 3 \cdot (1,1,2)$
5	26	$1 \cdot (1,1,1,1,1) + 10 \cdot (1,1,1,2) + 15 \cdot (1,2,2)$
6	236	$1 \cdot (1,1,1,1,1,1) + 15 \cdot (1,1,1,1,2) + 40 \cdot (1,1,1,3)$
		$+ 45 \cdot (1,1,2,2) + 120 \cdot (1,2,3) + 15 \cdot (2,2,2)$

Subtlety: some partitions for an even number of external lines are degenerate, e. g. $(1,1,1,3)$ and $(1,2,3)$ contain the same diagram

and representatives must be chosen consistently.

Non trivial cross check from self-consistency of counting
$F(d_{\text{max}}, n) = \#$ of Feynman diagrams with n external legs in unflavored

$$\mathcal{L} = \frac{1}{2}\partial_\mu\phi\partial^\mu\phi - \frac{m^2}{2}\phi^2 - \sum_{d=3}^{d_{\text{max}}} \frac{\lambda_d}{d!}\phi^d$$

theory. In a partition $N_{d,n} = \{n_1, n_2, \ldots, n_d\}$ with $n = n_1 + n_2 + \cdots + n_d$, there are

$$\tilde{F}(d_{\text{max}}, N_{d,n}) = \frac{1}{(1 + \delta_{n_d, n_1+n_2+\cdots+n_{d-1}})} \frac{n!}{|\mathcal{S}(N_{d,n})|} \prod_{i=1}^{d} \frac{F(d_{\text{max}}, n_i + 1)}{n_i!}$$

Feynman diagrams ($|\mathcal{S}(N)|$ the size of the symmetric group of N).

Non trivial cross check from self-consistency of counting
$F(d_{\max}, n)$ = # of Feynman diagrams with n external legs in unflavored

$$\mathcal{L} = \frac{1}{2}\partial_\mu\phi\partial^\mu\phi - \frac{m^2}{2}\phi^2 - \sum_{d=3}^{d_{\max}} \frac{\lambda_d}{d!}\phi^d$$

theory. In a partition $N_{d,n} = \{n_1, n_2, \ldots, n_d\}$ with
$n = n_1 + n_2 + \cdots + n_d$, there are

$$\tilde{F}(d_{\max}, N_{d,n}) = \frac{1}{(1 + \delta_{n_d, n_1+n_2+\cdots+n_{d-1}})} \frac{n!}{|\mathcal{S}(N_{d,n})|} \prod_{i=1}^{d} \frac{F(d_{\max}, n_i + 1)}{n_i!}$$

Feynman diagrams ($|\mathcal{S}(N)|$ the size of the symmetric group of N).

$$F(d_{\max}, n) = \sum_{d=3}^{d_{\max}} \sum_{\substack{N=\{n_1,n_2,\ldots,n_d\} \\ n_1+n_2+\cdots+n_d=n \\ 1\leqslant n_1\leqslant n_2\leqslant\cdots\leqslant n_d\leqslant\lfloor n/2\rfloor}} \tilde{F}(d_{\max}, N)$$

can be be checked numerically up to $n = \mathcal{O}(100)$.

Even for vector particles, the 1POWs are 'almost' physical objects and satisfy simple Ward Identities in unbroken gauge theories

$$\frac{\partial}{\partial x_\mu} \langle \text{out} | A_\mu(x) | \text{in} \rangle_{\text{amp.}} = 0$$

Even for vector particles, the 1POWs are 'almost' physical objects and satisfy simple Ward Identities in unbroken gauge theories

$$\frac{\partial}{\partial x_\mu} \langle \text{out} | A_\mu(x) | \text{in} \rangle_{\text{amp.}} = 0$$

and in spontaneously gauge theories in R_ξ-gauge

$$\frac{\partial}{\partial x_\mu} \langle \text{out} | W_\mu(x) | \text{in} \rangle_{\text{amp.}} = \xi_W m_W \langle \text{out} | \phi_W(x) | \text{in} \rangle_{\text{amp.}} .$$

Even for vector particles, the 1POWs are 'almost' physical objects and satisfy simple Ward Identities in unbroken gauge theories

$$\frac{\partial}{\partial x_\mu} \langle \text{out}|A_\mu(x)|\text{in}\rangle_{\text{amp.}} = 0$$

and in spontaneously gauge theories in R_ξ-gauge

$$\frac{\partial}{\partial x_\mu} \langle \text{out}|W_\mu(x)|\text{in}\rangle_{\text{amp.}} = \xi_W m_W \langle \text{out}|\phi_W(x)|\text{in}\rangle_{\text{amp.}} .$$

☺ code for matrix elements can optionally be instrumented to check these Ward identities, testing the consistency a particular model and the numerical stability of expressions.

Even for vector particles, the 1POWs are 'almost' physical objects and satisfy simple Ward Identities in unbroken gauge theories

$$\frac{\partial}{\partial x_\mu} \langle \text{out}|A_\mu(x)|\text{in}\rangle_{\text{amp.}} = 0$$

and in spontaneously gauge theories in R_ξ-gauge

$$\frac{\partial}{\partial x_\mu} \langle \text{out}|W_\mu(x)|\text{in}\rangle_{\text{amp.}} = \xi_W m_W \langle \text{out}|\phi_W(x)|\text{in}\rangle_{\text{amp.}} \,.$$

☺ code for matrix elements can optionally be instrumented to check these Ward identities, testing the consistency a particular model and the numerical stability of expressions.

Amplitudes can be continued off-shell:

- Slavnov-Taylor Identities can be checked numerically by adding operator insertions implementing BRS transformations.

Slightly simplified `Model.T` signature that all models must implement:

```
module type Model.T =
  sig
    type flavor (* all quantum numbers *)
    val flavor_symbol : flavor -> string
    val conjugate : flavor -> flavor (* antiparticles *)
    val lorentz : flavor -> Coupling.lorentz (* spin *)
    val fermion : flavor -> int (* fermion, boson, antifermion *)
    val width : flavor -> Coupling.width (* scheme, not value! *)
    type gauge (* parametrized gauges *)
    val gauge_symbol : gauge -> string
    val propagator : flavor -> gauge Coupling.propagator
    type constant (* coupling constants *)
    val constant_symbol : constant -> string
    val fuse2 : flavor -> flavor ->
     (flavor * constant Coupling.t) list (* A_μ(p_12) ← gψ̄(p_1)γ_μψ(p_2) *)
    val fuse3 : flavor -> flavor -> flavor ->
     (flavor * constant Coupling.t) list (* φ(p_123) ← gφ(p_1)φ(p_2)φ(p_3) *)
    val fuse : flavor list -> (flavor * constant Coupling.t) list
  end
```

Supported models in WHiZard 1.51 / O'Mega 0.11

- SM
  - ☺ robust since many years

Supported models in WHiZard 1.51 / O'Mega 0.11

- SM
  - ☺ robust since many years
    - unitarity gauge and R_ξ gauges
    - width treatments: constant, time-like, "fudge" scheme, complex mass [C. Schwinn], ...

Supported models in WHiZard 1.51 / O'Mega 0.11

- SM
  - ☺ robust since many years
    - unitarity gauge and R_ξ gauges
    - width treatments: constant, time-like, "fudge" scheme, complex mass [C. Schwinn], ...
- complete MSSM (using SLHA input parameters)
  - ☺ "ready for prime time" since 2005 (see below)

Supported models in WHiZard 1.51 / O'Mega 0.11

- SM
  - ☺ robust since many years
    - unitarity gauge and R_ξ gauges
    - width treatments: constant, time-like, "fudge" scheme, complex mass [C. Schwinn], ...
- complete MSSM (using SLHA input parameters)
  - ☺ "ready for prime time" since 2005 (see below)
    - some NMSSM

Supported models in WHiZard 1.51 / O'Mega 0.11

- SM
  - ☺ robust since many years
  - unitarity gauge and R_ξ gauges
  - width treatments: constant, time-like, "fudge" scheme, complex mass [C. Schwinn], ...
- complete MSSM (using SLHA input parameters)
  - ☺ "ready for prime time" since 2005 (see below)
  - some NMSSM
- BSM models remain constantly evolving (as is the nature of BSM physics ...)
  - "anomalous couplings": robust since TESLA TDR
  - various little higgs models available
  - extra dimension models available
  - feel free to DIY ...

credit for many recent implementations and applications: [J. Reuter]

- comprehensive comparison of three independent complete MSSM implementations
  - AMEGIC++ / SHERPA
  - MADGRAPH / HELAS
  - WHiZard / O'Mega

- comprehensive comparison of three independent complete MSSM implementations
  - AMEGIC++ / SHERPA
  - MADGRAPH / HELAS
  - WHiZard / O'Mega
- different representations of the MSSM Lagrangrian
  - e. g. signs of neutralino masses

- comprehensive comparison of three independent complete MSSM implementations
  - AMEGIC++ / SHERPA
  - MADGRAPH / HELAS
  - WHiZard / O'Mega
- different representations of the MSSM Lagrangrian
  - e. g. signs of neutralino masses
- different algorithms in different programming languages
  - C++ (AMEGIC++) vs. Fortran90 (MADGRAPH) vs. O'Caml (O'Mega)

- comprehensive comparison of three independent complete MSSM implementations
  - AMEGIC++ / SHERPA
  - MADGRAPH / HELAS
  - WHiZard / O'Mega
- different representations of the MSSM Lagrangrian
  - e. g. signs of neutralino masses
- different algorithms in different programming languages
  - C++ (AMEGIC++) vs. Fortran90 (MADGRAPH) vs. O'Caml (O'Mega)

☺ same results!

- comprehensive comparison of three independent complete MSSM implementations
  - AMEGIC++ / SHERPA
  - MADGRAPH / HELAS
  - WHiZard / O'Mega
- different representations of the MSSM Lagrangrian
  - e. g. signs of neutralino masses
- different algorithms in different programming languages
  - C++ (AMEGIC++) vs. Fortran90 (MADGRAPH) vs. O'Caml (O'Mega)

☺ same results!

- … after some (surprisingly little) debugging …

- comprehensive comparison of three independent complete MSSM implementations
  - AMEGIC++ / SHERPA
  - MADGRAPH / HELAS
  - WHiZard / O'Mega
- different representations of the MSSM Lagrangrian
  - e. g. signs of neutralino masses
- different algorithms in different programming languages
  - C++ (AMEGIC++) vs. Fortran90 (MADGRAPH) vs. O'Caml (O'Mega)

☺ same results!

- . . . after some (surprisingly little) debugging . . .
- read all about it in [Hagiwara, Kilian, Krauss, T. O., Plehn, Rainwater, Reuter, Schumann, PRD73 (2006) 055005]

- comprehensive comparison of three independent complete MSSM implementations
  - AMEGIC++ / SHERPA
  - MADGRAPH / HELAS
  - WHiZard / O'Mega
- different representations of the MSSM Lagrangrian
  - e. g. signs of neutralino masses
- different algorithms in different programming languages
  - C++ (AMEGIC++) vs. Fortran90 (MADGRAPH) vs. O'Caml (O'Mega)

☺ same results!

- … after some (surprisingly little) debugging …
- read all about it in [Hagiwara, Kilian, Krauss, T. O., Plehn, Rainwater, Reuter, Schumann, PRD73 (2006) 055005]
  - ☺ contains comprehensive list of $2 \to 2$ cross sections for testing your own Feynman rules

$W^- Z \to X^-$						
Final state	MADGRAPH / HELAS		WHiZard / O'Mega		AMEGIC++ / SHERPA	
	0.5 TeV	2 TeV	0.5 TeV	2 TeV	0.5 TeV	2 TeV
$\tilde{e}_L \tilde{\nu}_e^*$						
$\tilde{\mu}_L \tilde{\nu}_\mu^*$						
$\tilde{\tau}_1 \tilde{\nu}_\tau^*$						
$\tilde{\tau}_2 \tilde{\nu}_\tau^*$						
$\tilde{d}_L \tilde{u}_L^*$	—		—		—	
$\tilde{s}_L \tilde{c}_L^*$	—		—		—	
$\tilde{b}_1 \tilde{t}_1^*$	—		—		—	
$\tilde{b}_2 \tilde{t}_2^*$	—		—		—	
$\tilde{b}_1 \tilde{t}_2^*$	—		—		—	
$\tilde{b}_2 \tilde{t}_1^*$	—		—		—	
$\tilde{\chi}_1^0 \tilde{\chi}_1^-$						
$\tilde{\chi}_2^0 \tilde{\chi}_1^-$						
$\tilde{\chi}_3^0 \tilde{\chi}_1^-$	—		—		—	
$\tilde{\chi}_4^0 \tilde{\chi}_1^-$	—		—		—	
$\tilde{\chi}_1^0 \tilde{\chi}_2^-$						
$\tilde{\chi}_2^0 \tilde{\chi}_2^-$	—		—		—	
$\tilde{\chi}_3^0 \tilde{\chi}_2^-$	—		—		—	
$\tilde{\chi}_4^0 \tilde{\chi}_2^-$	—		—		—	
$h^0 H^-$	—		—		—	
$H^0 H^-$	—		—		—	
$A^0 H^-$	—		—		—	
$W^- h^0$						
$W^- H^0$						
$W^- A^0$						
$Z H^-$						

$W^- Z \to X^-$						
Final state	MADGRAPH / HELAS		WHiZard / O'Mega		AMEGIC++ / SHERPA	
	0.5 TeV	2 TeV	0.5 TeV	2 TeV	0.5 TeV	2 TeV
$\tilde{e}_L \tilde{\nu}_e^*$			96.639(2)			
$\tilde{\mu}_L \tilde{\nu}_\mu^*$			96.638(2)			
$\tilde{\tau}_1 \tilde{\nu}_\tau^*$			14.952(1)			
$\tilde{\tau}_2 \tilde{\nu}_\tau^*$			85.875(2)			
$\tilde{d}_L \tilde{u}_L^*$	—		—		—	
$\tilde{s}_L \tilde{c}_L^*$	—		—		—	
$\tilde{b}_1 \tilde{t}_1^*$	—		—		—	
$\tilde{b}_2 \tilde{t}_2^*$	—		—		—	
$\tilde{b}_1 \tilde{t}_2^*$	—		—		—	
$\tilde{b}_2 \tilde{t}_1^*$	—		—		—	
$\tilde{\chi}_1^0 \tilde{\chi}_1^-$			61.626(3)			
$\tilde{\chi}_2^0 \tilde{\chi}_1^-$			2.8350(3)e3			
$\tilde{\chi}_3^0 \tilde{\chi}_1^-$	—		—		—	
$\tilde{\chi}_4^0 \tilde{\chi}_1^-$	—		—		—	
$\tilde{\chi}_1^0 \tilde{\chi}_2^-$			11.7619(7)			
$\tilde{\chi}_2^0 \tilde{\chi}_2^-$	—		—		—	
$\tilde{\chi}_3^0 \tilde{\chi}_2^-$	—		—		—	
$\tilde{\chi}_4^0 \tilde{\chi}_2^-$	—		—		—	
$h^0 H^-$	—		—		—	
$H^0 H^-$	—		—		—	
$A^0 H^-$	—		—		—	
$W^- h^0$			7.6213(6)e4			
$W^- H^0$			4.2446(2)			
$W^- A^0$			1.07037(1)			
$Z H^-$			0.17723(2)			

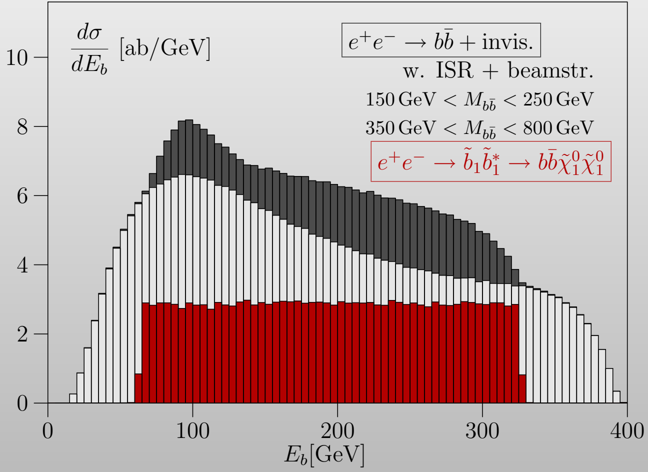
$W^- Z \to X^-$						
Final state	MADGRAPH / HELAS		WHiZard / O'Mega		AMEGIC++ / SHERPA	
	0.5 TeV	2 TeV	0.5 TeV	2 TeV	0.5 TeV	2 TeV
$\tilde{e}_L \tilde{\nu}_e^*$	96.635(6)		96.639(2)		96.632(5)	
$\tilde{\mu}_L \tilde{\nu}_\mu^*$	96.635(6)		96.638(2)		96.631(5)	
$\tilde{\tau}_1 \tilde{\nu}_\tau^*$	14.9542(8)		14.952(1)		14.953(1)	
$\tilde{\tau}_2 \tilde{\nu}_\tau^*$	85.875(5)		85.875(2)		85.870(4)	
$\tilde{d}_L \tilde{u}_L^*$	—		—		—	
$\tilde{s}_L \tilde{c}_L^*$	—		—		—	
$\tilde{b}_1 \tilde{t}_1^*$	—		—		—	
$\tilde{b}_2 \tilde{t}_2^*$	—		—		—	
$\tilde{b}_1 \tilde{t}_2^*$	—		—		—	
$\tilde{b}_2 \tilde{t}_1^*$	—		—		—	
$\tilde{\chi}_1^0 \tilde{\chi}_1^-$	61.634(6)		61.626(3)		61.633(3)	
$\tilde{\chi}_2^0 \tilde{\chi}_1^-$	2.8355(7)e3		2.8350(3)e3		2.8356(2)e3	
$\tilde{\chi}_3^0 \tilde{\chi}_1^-$	—		—		—	
$\tilde{\chi}_4^0 \tilde{\chi}_1^-$	—		—		—	
$\tilde{\chi}_1^0 \tilde{\chi}_2^-$	11.7607(3)		11.7619(7)		11.7602(6)	
$\tilde{\chi}_2^0 \tilde{\chi}_2^-$	—		—		—	
$\tilde{\chi}_3^0 \tilde{\chi}_2^-$	—		—		—	
$\tilde{\chi}_4^0 \tilde{\chi}_2^-$	—		—		—	
$h^0 H^-$	—		—		—	
$H^0 H^-$	—		—		—	
$A^0 H^-$	—		—		—	
$W^- h^0$	7.620(3)e4		7.6213(6)e4		7.6209(4)e4	
$W^- H^0$	4.2446(2)		4.2446(2)		4.2445(2)	
$W^- A^0$	1.07034(3)		1.07037(1)		1.07017(6)	
$Z H^-$	0.177241(1)		0.17723(2)		0.17714(4)	

$W^- Z \to X^-$						
Final state	MADGRAPH / HELAS		WHiZard / O'Mega		AMEGIC++ / SHERPA	
	0.5 TeV	2 TeV	0.5 TeV	2 TeV	0.5 TeV	2 TeV
$\tilde{e}_L \tilde{\nu}_e^*$	96.635(6)		96.639(2)	15.728(2)	96.632(5)	
$\tilde{\mu}_L \tilde{\nu}_\mu^*$	96.635(6)		96.638(2)	15.727(2)	96.631(5)	
$\tilde{\tau}_1 \tilde{\nu}_\tau^*$	14.9542(8)		14.952(1)	1.4268(2)	14.953(1)	
$\tilde{\tau}_2 \tilde{\nu}_\tau^*$	85.875(5)		85.875(2)	14.478(2)	85.870(4)	
$\tilde{d}_L \tilde{u}_L^*$	—		—	24.220(1)	—	
$\tilde{s}_L \tilde{c}_L^*$	—		—	24.221(1)	—	
$\tilde{b}_1 \tilde{t}_1^*$	—		—	40.676(4)	—	
$\tilde{b}_2 \tilde{t}_2^*$	—		—	8.3706(7)	—	
$\tilde{b}_1 \tilde{t}_2^*$	—		—	63.592(6)	—	
$\tilde{b}_2 \tilde{t}_1^*$	—		—	3.9236(5)	—	
$\tilde{\chi}_1^0 \tilde{\chi}_1^-$	61.634(6)		61.626(3)	16.389(1)	61.633(3)	
$\tilde{\chi}_2^0 \tilde{\chi}_1^-$	2.8355(7)e3		2.8350(3)e3	668.1(1)	2.8356(2)e3	
$\tilde{\chi}_3^0 \tilde{\chi}_1^-$	—		—	278.53(1)	—	
$\tilde{\chi}_4^0 \tilde{\chi}_1^-$	—		—	270.97(2)	—	
$\tilde{\chi}_1^0 \tilde{\chi}_2^-$	11.7607(3)		11.7619(7)	12.380(1)	11.7602(6)	
$\tilde{\chi}_2^0 \tilde{\chi}_2^-$	—		—	218.38(2)	—	
$\tilde{\chi}_3^0 \tilde{\chi}_2^-$	—		—	76.494(5)	—	
$\tilde{\chi}_4^0 \tilde{\chi}_2^-$	—		—	97.693(7)	—	
$h^0 H^-$	—		—	4.4399(5)e-3	—	
$H^0 H^-$	—		—	6.1592(2)	—	
$A^0 H^-$	—		—	5.9726(5)	—	
$W^- h^0$	7.620(3)e4		7.6213(6)e4	8.289(2)e4	7.6209(4)e4	
$W^- H^0$	4.2446(2)		4.2446(2)	15.783(3)	4.2445(2)	
$W^- A^0$	1.07034(3)		1.07037(1)	0.24815(7)	1.07017(6)	
$Z H^-$	0.177241(1)		0.17723(2)	0.25403(7)	0.17714(4)	

$W^- Z \to X^-$						
Final state	MADGRAPH / HELAS		WHiZard / O'Mega		AMEGIC++ / SHERPA	
	0.5 TeV	2 TeV	0.5 TeV	2 TeV	0.5 TeV	2 TeV
$\tilde{e}_L \tilde{\nu}_e^*$	96.635(6)	15.726(1)	96.639(2)	15.728(2)	96.632(5)	15.7249(8)
$\tilde{\mu}_L \tilde{\nu}_\mu^*$	96.635(6)	15.726(1)	96.638(2)	15.727(2)	96.631(5)	15.7264(8)
$\tilde{\tau}_1 \tilde{\nu}_\tau^*$	14.9542(8)	1.427(1)	14.952(1)	1.4268(2)	14.953(1)	1.42747(7)
$\tilde{\tau}_2 \tilde{\nu}_\tau^*$	85.875(5)	14.479(1)	85.875(2)	14.478(2)	85.870(4)	14.4780(7)
$\tilde{d}_L \tilde{u}_L^*$	—	24.220(3)	—	24.220(1)	—	24.219(1)
$\tilde{s}_L \tilde{c}_L^*$	—	24.220(3)	—	24.221(1)	—	24.220(1)
$\tilde{b}_1 \tilde{t}_1^*$	—	40.676(2)	—	40.676(4)	—	40.677(2)
$\tilde{b}_2 \tilde{t}_2^*$	—	8.3717(5)	—	8.3706(7)	—	8.3722(4)
$\tilde{b}_1 \tilde{t}_2^*$	—	63.596(3)	—	63.592(6)	—	63.591(3)
$\tilde{b}_2 \tilde{t}_1^*$	—	3.9242(2)	—	3.9236(5)	—	3.9244(2)
$\tilde{\chi}_1^0 \tilde{\chi}_1^-$	61.634(6)	16.389(5)	61.626(3)	16.389(1)	61.633(3)	16.391(1)
$\tilde{\chi}_2^0 \tilde{\chi}_1^-$	2.8355(7)e3	668.2(4)	2.8350(3)e3	668.1(1)	2.8356(2)e3	668.34(3)
$\tilde{\chi}_3^0 \tilde{\chi}_1^-$	—	278.5(1)	—	278.53(1)	—	278.58(2)
$\tilde{\chi}_4^0 \tilde{\chi}_1^-$	—	270.9(1)	—	270.97(2)	—	271.02(2)
$\tilde{\chi}_1^0 \tilde{\chi}_2^-$	11.7607(3)	12.379(4)	11.7619(7)	12.380(1)	11.7602(6)	12.380(1)
$\tilde{\chi}_2^0 \tilde{\chi}_2^-$	—	218.3(1)	—	218.38(2)	—	218.40(1)
$\tilde{\chi}_3^0 \tilde{\chi}_2^-$	—	76.50(3)	—	76.494(5)	—	76.497(4)
$\tilde{\chi}_4^0 \tilde{\chi}_2^-$	—	97.70(4)	—	97.693(7)	—	97.693(4)
$h^0 H^-$	—	4.439(6)e-3	—	4.4399(5)e-3	—	4.4395(2)e-3
$H^0 H^-$	—	6.1592(6)	—	6.1592(2)	—	6.1589(3)
$A^0 H^-$	—	5.9728(6)	—	5.9726(5)	—	5.9723(3)
$W^- h^0$	7.620(3)e4	8.29(1)e4	7.6213(6)e4	8.289(2)e4	7.6209(4)e4	8.2909(4)e4
$W^- H^0$	4.2446(2)	15.78(2)	4.2446(2)	15.783(3)	4.2445(2)	15.7848(8)
$W^- A^0$	1.07034(3)	0.24799(1)	1.07037(1)	0.24815(7)	1.07017(6)	0.24801(1)
$Z H^-$	0.177241(1)	0.25405(1)	0.17723(2)	0.25403(7)	0.17714(4)	0.25404(1)

- ILC example: sbottom pair production

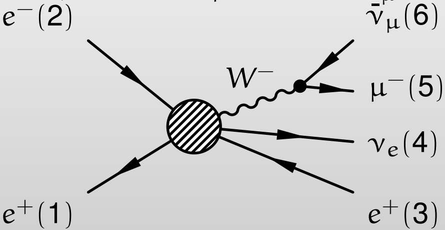
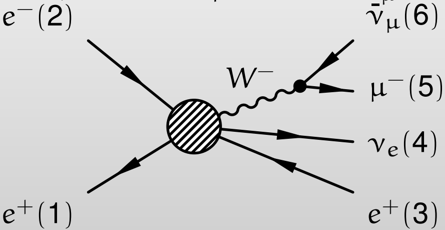
- ILC example: sbottom pair production
- complete irreducible background, ISR and beamstrahlung



Cascade decays are supported

- ```
  $ f90_SM -scatter "e+ e- -> e+ nue mu- numubar" \
           -cascade "3 + 4 ~ W-"
  ```

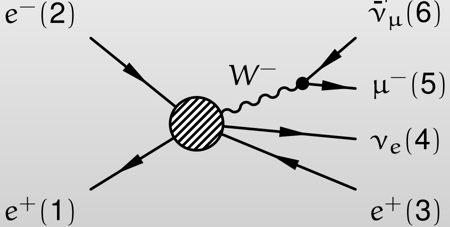
  select only diagrams, where the μ^- and the $\bar{\nu}_\mu$ derive from a $W^\pm$


  ```

Cascade decays are supported

- ```
  $ f90_SM -scatter "e+ e- -> e+ nue mu- numubar" \
            -cascade "3 + 4 ~ W-"
  ```

  select only diagrams, where the μ^- and the $\bar{\nu}_\mu$ derive from a $W^\pm$



☹ not necessarily fully gauge invariant, but correct spin correlations from compact expressions
  ```

Cascade decays are supported

```
$ f90_SM -scatter "e+ e- -> e+ nue mu- numubar" \
              -cascade "3 + 4 ~ W-"
```

select only diagrams, where the μ^- and the $\bar{\nu}_\mu$ derive from a $W^\pm$



☹ not necessarily fully gauge invariant, but correct spin correlations from compact expressions

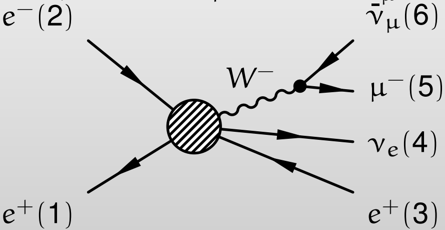
```
$ f90_SM -scatter "e+ e- -> e+ nue mu- numubar" \
              -cascade "3 + 4 = W-"
```

the same, but in addition force the $W^\pm$ on-shell

Cascade decays are supported

```
$ f90_SM -scatter "e+ e- -> e+ nue mu- numubar" \
         -cascade "3 + 4 ~ W-"
```

select only diagrams, where the μ^- and the $\bar{\nu}_\mu$ derive from a $W^\pm$



☹ not necessarily fully gauge invariant, but correct spin correlations from compact expressions

```
$ f90_SM -scatter "e+ e- -> e+ nue mu- numubar" \
         -cascade "3 + 4 = W-"
```

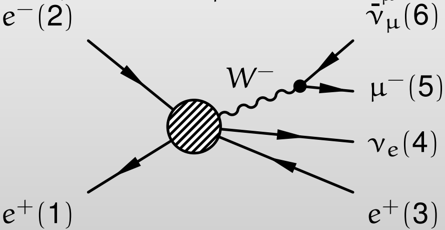
the same, but in addition force the $W^\pm$ on-shell

☺ gauge invariant

Cascade decays are supported

```
$ f90_SM -scatter "e+ e- -> e+ nue mu- numubar" \
         -cascade "3 + 4 ~ W-"
```

select only diagrams, where the μ^- and the $\bar{\nu}_\mu$ derive from a $W^\pm$



☹ not necessarily fully gauge invariant, but correct spin correlations from compact expressions

```
$ f90_SM -scatter "e+ e- -> e+ nue mu- numubar" \
         -cascade "3 + 4 = W-"
```

the same, but in addition force the $W^\pm$ on-shell

☺ gauge invariant

- both can be combined arbitrarily with boolean operators && and ||

- O’Mega / WHiZard does general colored amplitudes

- O'Mega / WHiZard does general colored amplitudes
- finally . . .

- O'Mega / WHiZard does general colored amplitudes
- finally ...
  - color flow representation

$$\psi \rightarrow \psi^i$$

$$\psi \rightarrow \psi^{ij}, \psi^0$$

- O'Mega / WHiZard does general colored amplitudes
- finally ...
  - color flow representation

$$\psi \to \psi^i$$
$$\phi \to \phi^{ij}, \phi^0$$

- can be implemented by enlarging the flavor quantum number and using special Feynman rules [à la Stelzer/Willenbrock]

$$|\text{flavor}\rangle \to |\text{flavor}\rangle \otimes |\text{color}\rangle$$

- O'Mega / WHiZard does general colored amplitudes
- finally ...
  - color flow representation

$$\psi \to \psi^i$$
$$\psi \to \psi^{ij}, \psi^0$$

- can be implemented by enlarging the flavor quantum number and using special Feynman rules [à la Stelzer/Willenbrock]

$$|\text{flavor}\rangle \to |\text{flavor}\rangle \otimes |\text{color}\rangle$$

  - currently complete "proof of concept" implementation in Whizard

- O'Mega / WHiZard does general colored amplitudes
- finally ...
  - color flow representation

$$\psi \to \psi^i$$
$$\psi \to \psi^{ij}, \psi^0$$

- can be implemented by enlarging the flavor quantum number and using special Feynman rules [à la Stelzer/Willenbrock]

$$|\text{flavor}\rangle \to |\text{flavor}\rangle \otimes |\text{color}\rangle$$

  - currently complete "proof of concept" implementation in Whizard
- generation of matrix elements can be time-consuming

- O'Mega / WHiZard does general colored amplitudes
- finally …
  - color flow representation

$$\psi \to \psi^i$$
$$\psi \to \psi^{ij}, \psi^0$$

- can be implemented by enlarging the flavor quantum number and using special Feynman rules [à la Stelzer/Willenbrock]

$$|\text{flavor}\rangle \to |\text{flavor}\rangle \otimes |\text{color}\rangle$$

  - currently complete "proof of concept" implementation in Whizard
- generation of matrix elements can be time-consuming
  - fast implementation in O'Mega on the way: general "`Colorize`" functor mapping of any uncolored model to a $SU(N)$ colorized model

Forthcoming native model file format:

- particle declarations

```
particle e- e+ : spin=1/2, fermion, pdg=11, tex="e^-", tex.anti="e^+"
particle nue nuebar : spin=1/2, fermion, pdg=12, tex="\nu_e", tex.anti="\bar\nu_e"
particle A : spin=1, boson, pdg=22, tex="\gamma"
particle Z : spin=1, boson, pdg=23, tex="Z"
particle W+ W- : spin=1, boson, pdg=24, tex="W^+", tex.anti="W^-"

coupling e
coupling g
coupling gv
coupling ga
```

Forthcoming native model file format:

- particle declarations

```
particle e- e+ : spin=1/2, fermion, pdg=11, tex="e^-", tex.anti="e^+"
particle nue nuebar : spin=1/2, fermion, pdg=12, tex="\nu_e", tex.anti="\bar\nu_e"
particle A : spin=1, boson, pdg=22, tex="\gamma"
particle Z : spin=1, boson, pdg=23, tex="Z"
particle W+ W- : spin=1, boson, pdg=24, tex="W^+", tex.anti="W^-"

coupling e
coupling g
coupling gv
coupling ga
```

- gauge currents $e\bar{e}\not{A}e$, $\bar{e}\not{Z}(g_V - g_A\gamma_5)e$ and $g\bar{e}\not{W}(1-\gamma_5)\nu_e$

```
vertex e+, A, e- :   { e * <1|V.e2|3> }
vertex e+, Z, e- :   { gv * <1|V.e2|3> - ga * <1|A.e2|3> }
vertex e+, W-, nue : { g * <1|(V-A).e2|3> }
```

Forthcoming native model file format:

- particle declarations

```
particle e- e+ : spin=1/2, fermion, pdg=11, tex="e^-", tex.anti="e^+"
particle nue nuebar : spin=1/2, fermion, pdg=12, tex="\nu_e", tex.anti="\bar\nu_e"
particle A : spin=1, boson, pdg=22, tex="\gamma"
particle Z : spin=1, boson, pdg=23, tex="Z"
particle W+ W- : spin=1, boson, pdg=24, tex="W^+", tex.anti="W^-"

coupling e
coupling g
coupling gv
coupling ga
```

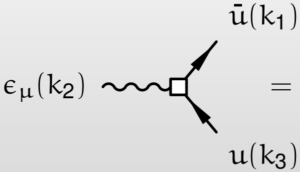
- gauge currents $e\bar{e}\not{A}e$, $\bar{e}\not{Z}(g_V - g_A\gamma_5)e$ and $g\bar{e}\not{W}(1-\gamma_5)\nu_e$

```
vertex e+, A, e- :   { e * <1|V.e2|3> }
vertex e+, Z, e- :   { gv * <1|V.e2|3> - ga * <1|A.e2|3> }
vertex e+, W-, nue : { g * <1|(V-A).e2|3> }
```

- triple gauge couplings

```
vertex W+, A, W- :   { e * ((k1 - k2).e3*e1.e2
                          + (k2 - k3).e1*e2.e3 + (k3 - k1).e2*e3.e1) }
vertex W+, Z, W- :   { g * ((k1 - k2).e3*e1.e2
                          + (k2 - k3).e1*e2.e3 + (k3 - k1).e2*e3.e1) }
```

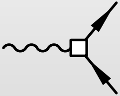
- noncommutative QED [à la Seiberg/Witten, Witten]



$$= ig \cdot \frac{i}{2} \left[(k\theta)_\mu \not{p} + (\theta p)_\mu \not{k} - (k\theta p)\gamma_\mu \right]$$

```
vertex e+, A, e- :   {   e * k2.[mu1]*[mu2].k3*<1|V.e2|3>
                       - e * k2.[mu1]*[mu2].e2*<1|V.k3|3>
                       - e * e2.[mu1]*[mu2].k3*<1|V.k2|3> }
```

- noncommutative QED [à la Seiberg/Witten, Witten]



$$= ig \cdot \frac{i}{2}\left[(k\theta)_\mu \not{p} + (\theta p)_\mu \not{k} - (k\theta p)\gamma_\mu\right]$$

```
vertex e+, A, e- :   {   e * k2.[mu1]*[mu2].k3*<1|V.e2|3>
                       - e * k2.[mu1]*[mu2].e2*<1|V.k3|3>
                       - e * e2.[mu1]*[mu2].k3*<1|V.k2|3> }
```

- Tradeoff: convenience vs. complexity
  - ☹ full programming language harder to learn, but more appropriate than model file format for defining complicated models with many couplings, e. g. the MSSM, from scratch
  - ☺ physics motivated model file format simplifies small extensions
    - additional particles
    - new vertices

$e^+e^- \to \tilde{\chi}_1^+\tilde{\chi}_1^-$:

```
$ f90_MSSM -scatter "e+ e- -> ch1+ ch1-"
```

$e^+e^- \to \tilde\chi_1^+\tilde\chi_1^-$:

```
$ f90_MSSM -scatter "e+ e- -> ch1+ ch1-"
```

```
pure function l1bl1cp1cm1 (k, s) result (amp)
  real(kind=omega_prec), dimension(0:,:), intent(in) :: k
  integer, dimension(:), intent(in) :: s
  complex(kind=omega_prec) :: amp
  type(momentum) :: p1, p2, p3, p4
  type(bispinor) :: cp1_4, l1_2
  type(bispinor) :: cm1_3, l1b_1
  complex(kind=omega_prec) :: snc1_13
  type(vector) :: a_12, z_12
  type(momentum) :: p12, p13
  p1 = - k(:,1) ! incoming e+
  p2 = - k(:,2) ! incoming e-
  p3 =   k(:,3) ! outgoing ch1+
  p4 =   k(:,4) ! outgoing ch1-
  l1b_1 = u (mass(11), - p1, s(1))
  l1_2 = u (mass(11), - p2, s(2))
  cm1_3 = v (mass(69), p3, s(3))
  cp1_4 = v (mass(69), p4, s(4))
```

```
  p12 = p1 + p2
  a_12 = pr_feynman(p12, + v_ff(qlep,l1b_1,l1_2))
  z_12 = pr_unitarity(p12,mass(23),wd_tl(p12,width(23)), &
     + va_ff(gnclep(1),gnclep(2),l1b_1,l1_2))
  p13 = p1 + p3
  snc1_13 = pr_phi(p13,mass(54),wd_tl(p13,width(54)), &
     + sr_ff(g_yuk_ch1_sn1_1_c,l1b_1,cm1_3))
  amp = 0
  amp = amp + snc1_13*( - sl_ff(g_yuk_ch1_sn1_1,l1_2,cp1_4))
  amp = amp + z_12*( + va_ff(-gczc_1_1(1),-gczc_1_1(2),cm1_3,cp1_4))
  amp = amp + a_12*( + v_ff(qchar,cm1_3,cp1_4))
  amp = - amp ! 2 vertices, 1 propagators
end function l1bl1cp1cm1
```

9 fusions, 3 propagators, 3 diagrams

```
    p12 = p1 + p2
    a_12 = pr_feynman(p12, + v_ff(qlep,l1b_1,l1_2))
    z_12 = pr_unitarity(p12,mass(23),wd_tl(p12,width(23)), &
       + va_ff(gnclep(1),gnclep(2),l1b_1,l1_2))
    p13 = p1 + p3
    snc1_13 = pr_phi(p13,mass(54),wd_tl(p13,width(54)), &
       + sr_ff(g_yuk_ch1_sn1_1_c,l1b_1,cm1_3))
    amp = 0
    amp = amp + snc1_13*( - sl_ff(g_yuk_ch1_sn1_1,l1_2,cp1_4))
    amp = amp + z_12*( + va_ff(-gczc_1_1(1),-gczc_1_1(2),cm1_3,cp1_4))
    amp = amp + a_12*( + v_ff(qchar,cm1_3,cp1_4))
    amp = - amp ! 2 vertices, 1 propagators
  end function l1bl1cp1cm1
```

9 fusions, 3 propagators, 3 diagrams

- ☺ readable code, can edited for exotic models (see below) or NLO vertex functions (see below)

Adding a photon $e^+e^- \to \tilde{\chi}_1^+\tilde{\chi}_1^-\gamma$:

```
$ f90_MSSM -scatter "e+ e- -> ch1+ ch1- A"
```

Adding a photon $e^+e^- \to \tilde\chi_1^+\tilde\chi_1^-\gamma$:

```
$ f90_MSSM -scatter "e+ e- -> ch1+ ch1- A"
```

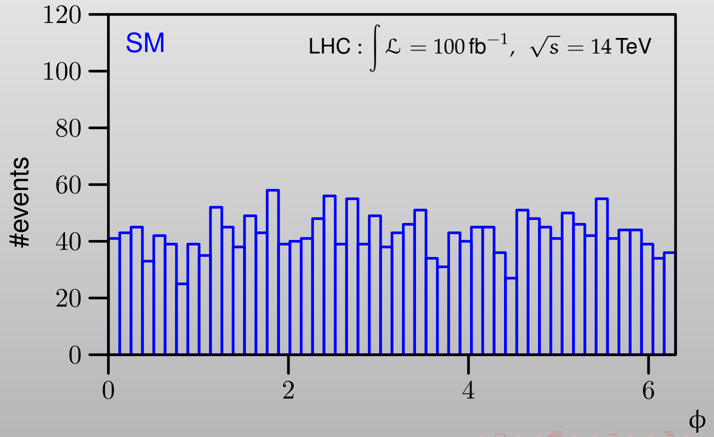
```
pure function l1bl1cp1cm1a (k, s) result (amp)
  real(kind=omega_prec), dimension(0:,:), intent(in) :: k
  integer, dimension(:), intent(in) :: s
  complex(kind=omega_prec) :: amp
  type(momentum) :: p1, p2, p3, p4, p5
  type(bispinor) :: cp1_4, l1_2
  type(bispinor) :: cm1_3, l1b_1
  type(vector) :: a_5
  complex(kind=omega_prec) :: sn1_24, snc1_13
  type(bispinor) :: cp1_45, l1_25
  type(bispinor) :: cm1_35, l1b_15
  type(vector) :: a_34, a_12, z_34, z_12
  type(momentum) :: p12, p13, p15, p24, p25, p34, p35, p45
  p1 = - k(:,1) ! incoming e+
  p2 = - k(:,2) ! incoming e-
  p3 =   k(:,3) ! outgoing ch1+
  p4 =   k(:,4) ! outgoing ch1-
  p5 =   k(:,5) ! outgoing A
  l1b_1 = u (mass(11), - p1, s(1))
  l1_2 = u (mass(11), - p2, s(2))
  cm1_3 = v (mass(69), p3, s(3))
  cp1_4 = v (mass(69), p4, s(4))
  a_5 = conjg (eps (mass(22), p5, s(5)))
  p12 = p1 + p2
  a_12 = pr_feynman(p12, + v_ff(qlep,l1b_1,l1_2))
  z_12 = pr_unitarity(p12,mass(23),wd_tl(p12,width(23)), &
     + va_ff(gnclep(1),gnclep(2),l1b_1,l1_2))
```

```
  p13 = p1 + p3
  snc1_13 = pr_phi(p13,mass(54),wd_tl(p13,width(54)), &
     + sr_ff(g_yuk_ch1_sn1_1_c,l1b_1,cm1_3))
  p24 = p2 + p4
  sn1_24 = pr_phi(p24,mass(54),wd_tl(p24,width(54)), &
     + sl_ff(g_yuk_ch1_sn1_1,l1_2,cp1_4))
  p34 = p3 + p4
  a_34 = pr_feynman(p34, + v_ff(qchar,cm1_3,cp1_4))
  z_34 = pr_unitarity(p34,mass(23),wd_tl(p34,width(23)), &
     + va_ff(-gczc_1_1(1),-gczc_1_1(2),cm1_3,cp1_4))
  p15 = p1 + p5
  l1b_15 = pr_psi(p15,mass(11),wd_tl(p15,width(11)), + f_vf(-qlep,a_5,l1b_1))
  p25 = p2 + p5
  l1_25 = pr_psi(p25,mass(11),wd_tl(p25,width(11)), + f_vf(qlep,a_5,l1_2))
  p35 = p3 + p5
  cm1_35 = pr_psi(p35,mass(69),wd_tl(p35,width(69)), &
     + f_vf(-qchar,a_5,cm1_3))
  p45 = p4 + p5
  cp1_45 = pr_psi(p45,mass(69),wd_tl(p45,width(69)), + f_vf(qchar,a_5,cp1_4))
  amp = 0
  amp = amp + sn1_24*( + sr_ff(g_yuk_ch1_sn1_1_c,cm1_35,l1b_1))
  amp = amp + snc1_13*( - sl_ff(g_yuk_ch1_sn1_1,l1_25,cp1_4) &
     + sl_ff(g_yuk_ch1_sn1_1,cp1_45,l1_2))
  amp = amp + l1_25*( - f_vf(-qlep,a_34,l1b_1) &
     - f_vaf(-(gnclep(1)),gnclep(2),z_34,l1b_1))
  amp = amp + l1b_15*( - f_srf(g_yuk_ch1_sn1_1_c,sn1_24,cm1_3) &
     + f_vf(qlep,a_34,l1_2) + f_vaf(gnclep(1),gnclep(2),z_34,l1_2))
  amp = amp + z_12*( - va_ff(-(-gczc_1_1(1)),-gczc_1_1(2),cp1_45,cm1_3) &
     + va_ff(-gczc_1_1(1),-gczc_1_1(2),cm1_35,cp1_4))
  amp = amp + a_12*( - v_ff(-qchar,cp1_45,cm1_3) + v_ff(qchar,cm1_35,cp1_4))
end function l1bl1cp1cm1a
```

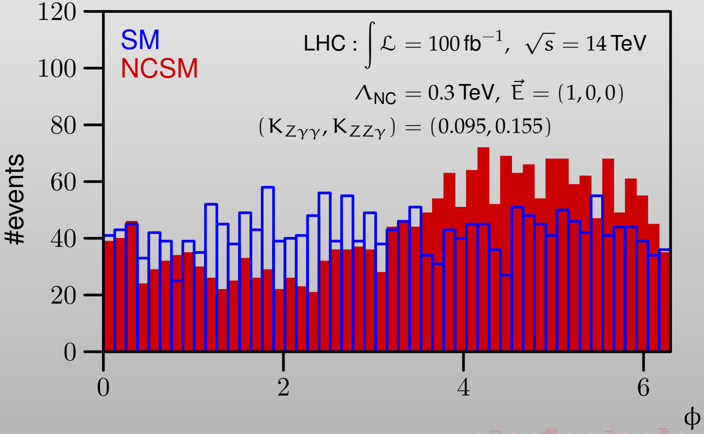
28 fusions, 10 propagators, 12 diagrams

- using omegalib and WHiZard [Alboteanu/T. O.,Rückl, 2005]

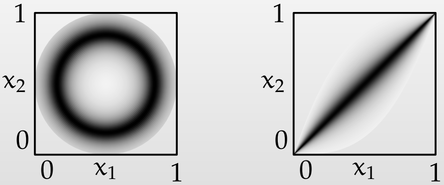
- using omegalib and WHiZard [Alboteanu/T. O.,Rückl, 2005]
- standard acceptance cuts and 85 GeV $< m_{\ell^+\ell^-} <$ 97 GeV, 200 GeV $< m_{\ell^+\ell^-\gamma} <$ 1 TeV, $0 < \cos\theta^*_\gamma < 0.9$, $\cos\theta_Z > 0$ and $\cos\theta_\gamma > 0$ (favoring $\bar{q}q$ over $q\bar{q}$!)



- using omegalib and WHiZard [Alboteanu/T. O.,Rückl, 2005]
- standard acceptance cuts and 85 GeV $< m_{\ell^+\ell^-} <$ 97 GeV, 200 GeV $< m_{\ell^+\ell^-\gamma} <$ 1 TeV, 0 $< \cos\theta^*_\gamma <$ 0.9, $\cos\theta_Z > 0$ and $\cos\theta_\gamma > 0$ (favoring $\bar{q}q$ over $q\bar{q}$!)

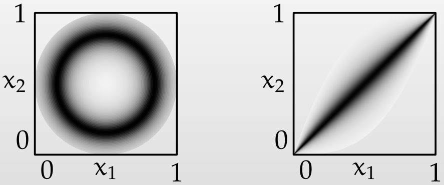


VEGAS' factorized ansatz handles



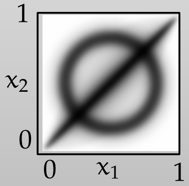
separately after mappings

VEGAS' factorized ansatz handles



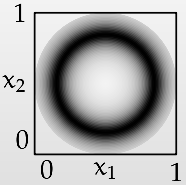
separately after mappings

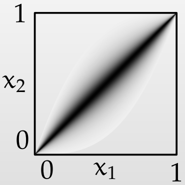
- ☹ fails for overlapping sing'ies



which is the common case (for more than one Feynman diagram)

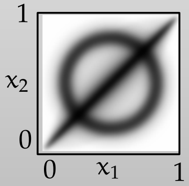
VEGAS' factorized ansatz handles





separately after mappings

fails for overlapping sing'ies



which is the common case (for more than one Feynman diagram)

∴ adaptive multichannel approach

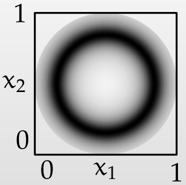
$$I(f) = \int_M d\mu(p)\, f(p)$$

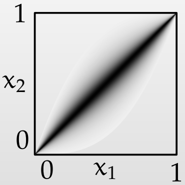
$$I(f) = \sum_{i=1}^{N_c} \alpha_i \int_0^1 g_i(x) d^n x \, \frac{f(\phi_i(x))}{g(\phi_i(x))}$$

with

$$g = \sum_{i=1}^{N_c} \alpha_i \cdot (g_i \circ \phi_i^{-1}) \left| \frac{\partial \phi_i^{-1}}{\partial p} \right|$$

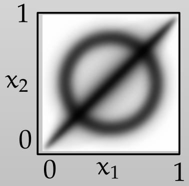
VEGAS' factorized ansatz handles





separately after mappings

- ☹ fails for overlapping sing'ies



which is the common case (for more than one Feynman diagram)

∴ adaptive multichannel approach

$$I(f) = \int_M d\mu(p)\, f(p)$$

$$I(f) = \sum_{i=1}^{N_c} \alpha_i \int_0^1 g_i(x) d^n x \, \frac{f(\phi_i(x))}{g(\phi_i(x))}$$

with

$$g = \sum_{i=1}^{N_c} \alpha_i \cdot (g_i \circ \phi_i^{-1}) \left| \frac{\partial \phi_i^{-1}}{\partial p} \right|$$

- ☺ works with factorized g_i adapted by VEGAS and α_i adapted by variance reduction.

- WHiZard calls O'Mega to generate matrix elements according to `conf/whizard.prc`

```
model   SM
uuza    u,U                        Z,A            omega
ppza    u:d:s:U:D:S,u:d:s:U:D:S   Z,A            omega
uulla   u,U                        e1:e2,E1:E2,A  omega
pplla   u:d:s:U:D:S,u:d:s:U:D:S   e1:e2,E1:E2,A  omega
```

(CompHEP notation for historical reasons . . .)

- WHiZard calls O'Mega to generate matrix elements according to `conf/whizard.prc`

```
model   SM
uuza    u,U                          Z,A              omega
ppza    u:d:s:U:D:S,u:d:s:U:D:S      Z,A              omega
uulla   u,U                          e1:e2,E1:E2,A    omega
pplla   u:d:s:U:D:S,u:d:s:U:D:S      e1:e2,E1:E2,A    omega
```

  (CompHEP notation for historical reasons ...)

- generated Fortran90 code for matrix elements is compiled and linked with support libraries (PDFs, vamp, etc.)

- WHiZard calls O'Mega to generate matrix elements according to `conf/whizard.prc`

```
model   SM
uuza    u,U                       Z,A              omega
ppza    u:d:s:U:D:S,u:d:s:U:D:S   Z,A              omega
uulla   u,U                       e1:e2,E1:E2,A    omega
pplla   u:d:s:U:D:S,u:d:s:U:D:S   e1:e2,E1:E2,A    omega
```

  (CompHEP notation for historical reasons . . .)
- generated Fortran90 code for matrix elements is compiled and linked with support libraries (PDFs, vamp, etc.)
- generates unweighted events according to `results/whizard.in`

```
&process_input
process_id = "pplla"
sqrts = 14000
luminosity = 0
polarized_beams = F
structured_beams = T
/
```

  etc.

- event samples are generated according to phase space cuts

```
process pplla
  cut E of 1 within 10 9999
  cut E of 2 within 10 9999
  cut E of 4 within 10 9999
  cut THETA(DEG) of 8 1 within 5 175
  cut THETA(DEG) of 8 2 within 5 175
  cut THETA(DEG) of 8 4 within 5 175
```

- event samples are generated according to phase space cuts

```
process pplla
  cut E of 1 within 10 9999
  cut E of 2 within 10 9999
  cut E of 4 within 10 9999
  cut THETA(DEG) of 8 1 within 5 175
  cut THETA(DEG) of 8 2 within 5 175
  cut THETA(DEG) of 8 4 within 5 175
```

- simple analysis can be performed by filling histograms

```
process pplla
  histogram M of 3  within 0 600 nbin 60
  histogram PT of 1 within 0 200 nbin 50
  histogram PT of 2 within 0 200 nbin 50
  histogram PT of 4 within 0 200 nbin 50
```

(will be generalized in the next version)

- no intention to become a full stand-alone replacement for Pythia and Herwig (unlike Sherpa)

- no intention to become a full stand-alone replacement for Pythia and Herwig (unlike Sherpa)
- instead: interfaces to external
  - QCD fragmentation
  - decay tables
  - underlying events
  - etc.

- no intention to become a full stand-alone replacement for Pythia and Herwig (unlike Sherpa)
- instead: interfaces to external
  - QCD fragmentation
  - decay tables
  - underlying events
  - etc.
- focus: reliable, efficient and convenient tool for
  - SM
  - MSSM (and NMSSM)
  - BSM from anomalous couplings to “far out” models

- no intention to become a full stand-alone replacement for Pythia and Herwig (unlike Sherpa)
- instead: interfaces to external
  - QCD fragmentation
  - decay tables
  - underlying events
  - etc.
- focus: reliable, efficient and convenient tool for
  - SM
  - MSSM (and NMSSM)
  - BSM from anomalous couplings to “far out” models
- semiautomatic support for loops. Proof of concept: exponentiated NLO results for $e^+e^- \to \tilde{\chi}^+\tilde{\chi}^-$ [T. Robens, SUSY06]

- no intention to become a full stand-alone replacement for Pythia and Herwig (unlike Sherpa)
- instead: interfaces to external
  - QCD fragmentation
  - decay tables
  - underlying events
  - etc.
- focus: reliable, efficient and convenient tool for
  - SM
  - MSSM (and NMSSM)
  - BSM from anomalous couplings to “far out” models
- semiautomatic support for loops. Proof of concept: exponentiated NLO results for $e^+e^- \to \tilde{\chi}^+\tilde{\chi}^-$ [T. Robens, SUSY06]
- parallelization build into vamp, not enabled yet ...