WHIZARD
Introduction & Tutorial

Thorsten Ohl

http://physik.uni-wuerzburg.de/ohl

Institute for Theoretical Physics and Astrophysics
Würzburg University

German-Egyptian School of Particle Physics
Center for Theoretical Physics
Zewail City of Science and Technology
February 24-28, 2013

Contents

Theoretical Challenges

Matrix Elements

Adaptive Monte Carlo

Usage

Exercise

- efficiently and reliably compute scattering probabilities

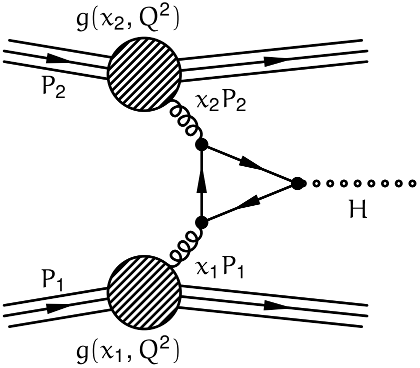
$$|\langle q_1, q_2, \ldots; \text{out}|T|p_1, p_2; \text{in}\rangle|^2$$

with lots of gauge cancellations among contributions

- for a multitude of physics models with qualitatively different particle content and interactions
    - standard model
    - supersymmetric extensions of the SM
    - SM with anomalous couplings
    - SM with extended gauge sector
    - SM with strongly interacting gauge bosons
    - additional space dimensions
    - ...
- such that their parameter space can be scanned and compared with experimental observations

- efficiently sample the multi particle phase space
    - scattering probabilities typically have many overlapping narrow peaks and integrable boundary singularities

- **Caveat:** not everything can be calculated in perturbation theory when hadrons (i. e. strongly interacting particles) are in play
- e. g. Higgs production at the LHC depends not only on the $gg \to H$ cross section, but also on the composition of the protons:



- asymptotic freedom and factorization allow to separate

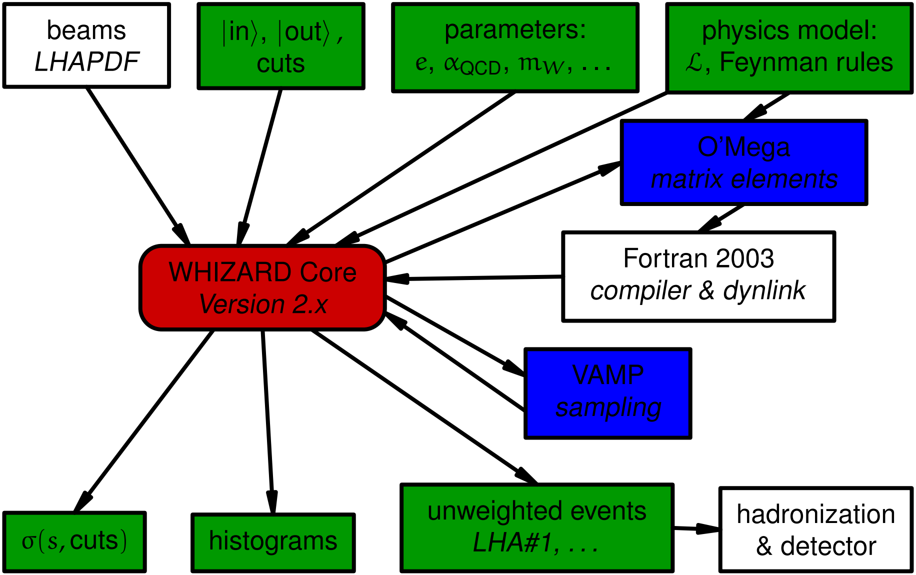
$$\sigma(s) = \sum_{i_1 i_2} \int dx_1 dx_2 \, D_{i_1}(x_1, \mu) D_{i_2}(x_2, \mu) \hat{\sigma}(x_1 x_2 s; \mu)$$

independent of the scale μ!

  - weakly coupled short distance/high energy phenomena, calculable in perturbation theory
    - hard scattering cross sections $\hat{\sigma}(\hat{s}; \mu)$
  - universal strongly coupled long distance/low energy phenomena, described by parametrizations
    - parton distributions $D_{i_j}(x_j; \mu)$

    and/or phenomenological models
    - fragmentation and hadronization
- a series of Les Houches Accords defines interfaces implementing this separation

∴ studies of new physics can concentrate on the hard interactions!


beams
LHAPDF
$|in\rangle$, $|out\rangle$, cuts
parameters:
e, α_{QCD}, m_W, ...
physics model:
$\mathcal{L}$, Feynman rules
O'Mega
matrix elements
WHIZARD Core
Version 2.x
Fortran 2003
compiler & dynlink
VAMP
sampling
$\sigma(s, \text{cuts})$
histograms
unweighted events
LHA#1, ...
hadronization
& detector

- desired: a computer program implementing the function

$$(\mathcal{L}, \{\text{incoming}\}, \{\text{outgoing}\}) \mapsto \mathcal{M}(\alpha_1, \ldots; p_1, \ldots; s_1, \ldots)$$

where

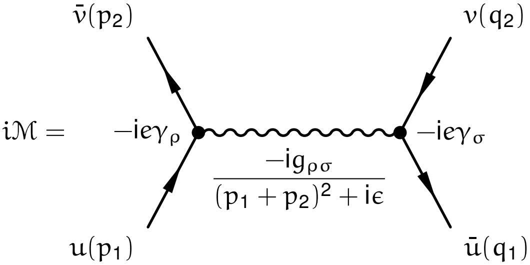
  - $\mathcal{L}$: Lagrangian (or Feynman rules) of a model (SM, MSSM, ...)
  - $\mathcal{M}(\alpha_1, \ldots; p_1, \ldots; s_1, \ldots)$: a function

$$\underbrace{\mathbf{R} \times \ldots \times \mathbf{R}}_{\text{masses, couplings, ...}} \times \underbrace{V^+ \times \ldots \times V^+}_{\text{4-momenta (forward light cones)}} \times \underbrace{\mathbf{Z} \times \ldots \times \mathbf{Z}}_{\text{helicities, colors}} \to \underbrace{\mathbf{C}}_{\text{amplitude}}$$

  in a form that can be evaluated numerically, typically as C, C++ or Fortran code in that can be compiled and linked to Monte Carlo phase space integrators and generators

- NB: in some cases only $\mathcal{L} \to \sum |\mathcal{M}(\alpha_1, \ldots; p_1, \ldots; s_1, \ldots\ldots)|^2$ is required. It is often better defined (infrared/collinear cancellations) and sometimes more compact (spin/polarization sums).
- first robust and usable examples in the early 1990s: CompHEP, FeynArts, Grace, MadGraph, ...

- for simplicity: $e^+e^- \to \mu^+\mu^-$ at PETRA (i. e. QED, mostly)
- just one Feynman diagram



- analytical expression

$$i\mathcal{M} = \bar{v}(p_2)(-ie\gamma^\rho)u(p_1)\frac{-ig_{\rho\sigma}}{(p_1+p_2)^2+i\epsilon}\bar{u}(q_1)(-ie\gamma^\sigma)v(q_2)$$
$$= ie^2\frac{1}{s}\left[\bar{v}(p_2)\gamma_\rho u(p_1)\right]\left[\bar{u}(q_1)\gamma^\rho v(q_2)\right]$$

- corresponing Fortran95 code
  (using a library for vector and spinor products and states)

```
pure function eleposmuoamu (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(spinor) :: muo_4, ele_1
  type(conjspinor) :: amu_3, pos_2
  type(vector) :: gam_12
  type(momentum) :: p12
  p1 = - k(:,1) ! incoming e-
  p2 = - k(:,2) ! incoming e+
  p3 =   k(:,3) ! outgoing m-
  p4 =   k(:,4) ! outgoing m+
  ele_1 = u (mass(11), - p1, s(1))                    ! u_{s_1}(k_1)
  pos_2 = vbar (mass(11), - p2, s(2))                 ! \bar{v}_{s_2}(k_2)
  amu_3 = ubar (mass(13), p3, s(3))                   ! \bar{u}_{s_3}(k_3)
  muo_4 = v (mass(13), p4, s(4))                      ! v_{s_4}(k_4)
  p12 = p1 + p2
  gam_12 = pr_feynman(p12, + v_ff(qlep,pos_2,ele_1)) ! (1/s) e\bar{v}(k_2)\gamma_\mu u(k_1)
  amp = 0
  amp = amp + gam_12*( + v_ff(qlep,amu_3,muo_4))     ! (1/s) e\bar{v}(k_2)\gamma_\mu u(k_1) e\bar{u}(k_3)\gamma^\mu v(k_4)
  amp = - amp ! 2 vertices, 1 propagators
end function eleposmuoamu
```

- the usual rules for manual calculations are algorithmic

∴ can be implemented in a computer program

The number of tree Feynman diagrams w/ n legs grows like a factorial, e. g. in ϕ^3-theory: $F(n) = (2n-5)!! = (2n-5)\cdot(2n-7)\cdot\ldots\cdot3\cdot1$

n	$F(n)$	$P(n)$
4	3	3
5	15	10
6	105	25
7	945	56
8	10 395	119
9	135 135	246
10	2 027 025	501
11	34 459 425	1 012
12	654 729 075	2 035
13	13 749 310 575	4 082
14	316 234 143 225	8 177
15	7 905 853 580 625	16 368

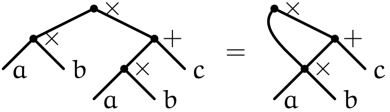
- computational costs grow beyond all reasonable limits
- gauge cancellations cause loss of precision

Number of possible momenta in tree diagrams grows only exponentially

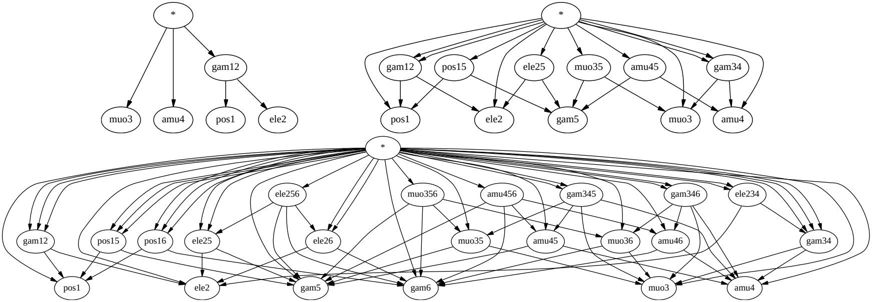
$$P(n) = \frac{2^n - 2}{2} - n = 2^{n-1} - n - 1$$

∴ Feynman diagrams redundant for many external particles!

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

$$ab(ab+c) =$$ 

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



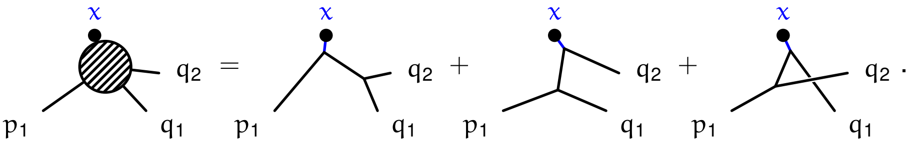
Efficient tree amplitudes

- Berends-Giele Recursion Relations [Berends, Giele]
  - manual calculations
- HELAS [Hagiwara et al.],
  - manual partial common subexpression elimination
- Madgraph [Stelzer et al.], AMEGIC++, COMIX [Krauss et al.]:
  - partial common subexpression elimination
  - $\therefore$ 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 [Papadopoulos et al.]:
  - ALPHA algorithm can be reformulated as recursive numerical solution of Schwinger-Dyson equations
- O'Mega [TO et al.]:
  - systematic elimination of all redundancies
  - symbolic, generation of compilable code

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.

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

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

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.

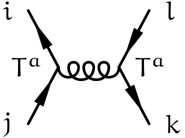
Amplitudes can be continued off-shell:

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

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

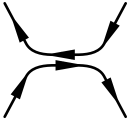
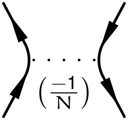
```
module type Model.T =
  sig
    type flavor (* all quantum numbers *)
    val flavor_symbol : flavor -> string
    val conjugate : flavor -> flavor (* antiparticles *)
    val lorentz : flavor -> Coupling.lorentz (* spin *)
    val fermion : flavor -> int (* fermion, boson, antifermion *)
    val 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 (* Aμ(p12) ← gψ̄(p1)γμψ(p2) *)
    val fuse3 : flavor -> flavor -> flavor ->
     (flavor * constant Coupling.t) list (* φ(p123) ← gφ(p1)φ(p2)φ(p3) *)
    val fuse : flavor list -> (flavor * constant Coupling.t) list
  end
```

- For interfacing to parton shower and hadronization programs, we need the amplitudes for all possible color flows or color connections. E. g. in $q\bar{q} \to q\bar{q}$



$$\Rightarrow \quad \sum_a (T^a)_i^j (T^a)_k^l = \frac{1}{2}\left(\delta_i^l \delta_k^j - \frac{1}{N}\delta_i^j \delta_k^l\right)$$

- This can be expressed by two diagrams, one for gluon and one for phantom exchange

$\frac{1}{\sqrt{2}}\delta_i^{l'}\delta_{k'}^j$  $\frac{1}{\sqrt{2}}\delta_{l'}^l\delta_k^{k'} \; + \; \frac{-1}{\sqrt{2}}\delta_i^j$ 

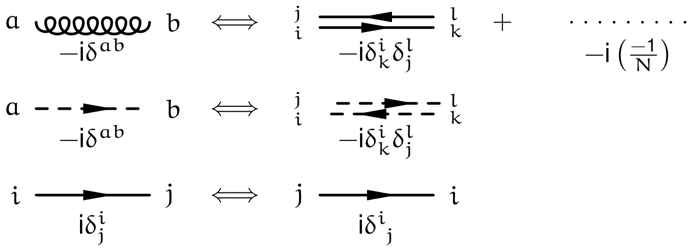
$\frac{-1}{\sqrt{2}}\delta_k^l$

- the sum over all colors can be written as a sum over all color flows

$$\sum_{IJ} N_C^{\lambda(J,J)} A_I A_J^*$$

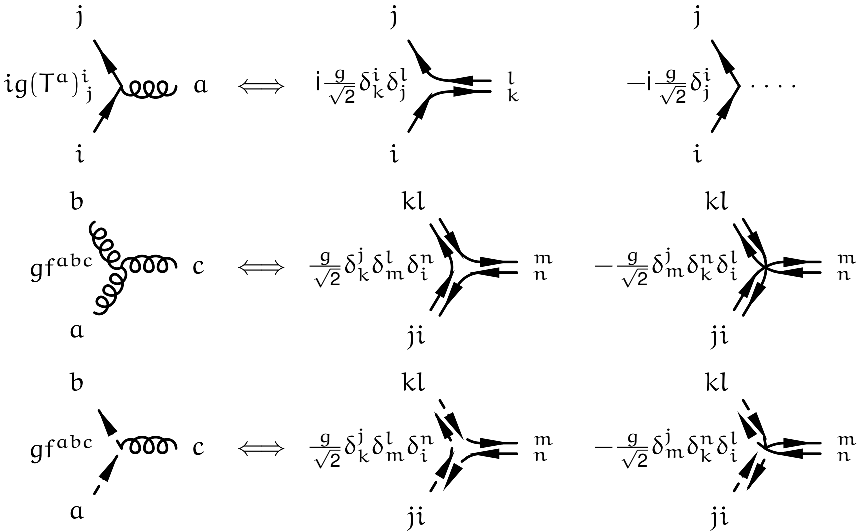
with $\lambda(J,J)$ the number of closed color loops in $A_I A_J^*$.

- one can use the completeness relation $2T^a \otimes T^a = \delta \tilde{\otimes} \delta - \delta \otimes \delta$ repeatedly to compute all the color flow amplitudes
- instead, one can formulate equivalent Feynman rules that give the color flow amplitudes directly
- Propagators

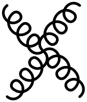


- The price to pay in the introduction of phantom particles that subtract the trace part of the gluons
- NB: they're not required for the Faddeev-Popov ghosts, because the trace of the gluons behaves as if it was abelian.

- Cubic vertices:

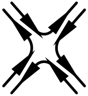


- Quartic vertices:

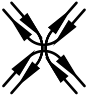


$$g^2(f^{abe}f^{cde} + f^{ace}f^{dbe} + f^{ade}f^{bce})$$

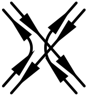
$\Longleftrightarrow$



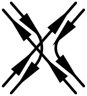
$\frac{g^2}{2}\delta^j_k\delta^l_m\delta^n_p\delta^o_i$



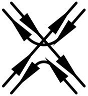
$\frac{g^2}{2}\delta^j_p\delta^l_i\delta^n_k\delta^o_m$



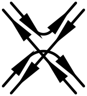
$\frac{g^2}{2}\delta^j_k\delta^l_p\delta^n_i\delta^o_m$



$\frac{g^2}{2}\delta^j_m\delta^l_i\delta^n_p\delta^o_k$



$\frac{g^2}{2}\delta^j_m\delta^l_p\delta^n_k\delta^o_i$



$\frac{g^2}{2}\delta^j_p\delta^l_m\delta^n_i\delta^o_k$

Example from supersymmetry: electroproduction of chargino pairs with bremsstrahlung, i. e. $e^+e^- \to \tilde{\chi}_1^+\tilde{\chi}_1^-\gamma$:

```
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

- readable code, can edited for exotic models or NLO vertex functions

Remaining problem:

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

1. non-trivial geometry of multi particle phase space

$$d\mu(p) = \delta^4(\textstyle\sum_n k_n - P) \prod_n d^4k_n\, \delta(k_n^2 - m_n^2)$$

2. ill-behaved function, i. e. squared matrix element w/ kinematical cuts

$$f(p) = |T(k_1, k_2, \ldots)|^2 \cdot C(k_1, k_2, \ldots)$$

Choose a (pseudo-)random sequence $p_g = \{p_1, p_2, \ldots, p_N\}$ distributed according to $d\mu_g(p) = g(p) d\mu(p)$, then an estimator of $I(f)$ is

$$E(f) = \left\langle \frac{f}{g} \right\rangle_g = \frac{1}{N} \sum_{i=1}^{N} \frac{f(p_i)}{g(p_i)}$$

The sampling error is estimated by the square root of the variance

$$V(f,g) = \frac{1}{N-1}\left(\left\langle\left(\frac{f}{g}\right)^2\right\rangle_g - \left\langle\frac{f}{g}\right\rangle_g^2\right)$$

which depends on g, even after averaging over p.
Conflicting goals for g

1. make $d\mu_g(p)$ simple enough, so that p_g can be generated w/ reasonable effort
2. choose g to minimize $V(f,g)$: importance sampling or stratified sampling

- for multi particle phase space, $d\mu(p)$ is very intricate and the generation of p_g is not trivial even for $g(p) = 1/\mathrm{Vol}(M)$.
- ∵ RAMBO: elegant trick only for $m_n = 0$ and g constant
- ∵ parametrizations $]0,1[^{\otimes \dim(M)} \to M$: require compensation of bad Jacobians

Practical considerations for particle physics:

- $\because$ only a small number of different manifolds M:
  - number of particles 2, 3, 4, 5, 6, 7, . . .
  - massless vs. massive particles
- $\therefore$ it makes sense to invest manpower into an optimal treatment of the geometry, i. e. $d\mu$
- $\because$ f changes w/
  - physics model du jour
  - physical parameters in the model
  - changing external cuts that can affect singular regions
- $\therefore$ automatic and computer aided adaptive approaches, i. e. numerical optimizations, are appropriate

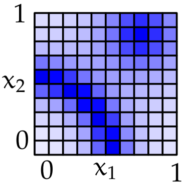
For over a quarter century, Peter Lepage's VEGAS has been the workhorse for adaptive Monte Carlo in particle physics.

For simplicity

$$x \in M =]0,1[^{\otimes n}\ ,\quad d\mu(x) = dx_1 \wedge dx_2 \wedge \ldots \wedge dx_n$$

How can we implement efficiently a variable weight g in $d\mu_g(x) = g(x)d\mu(x)$?

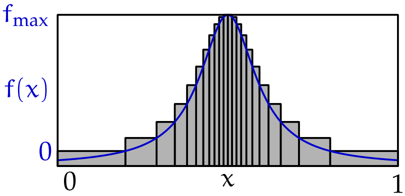
- optimization of expansion coefficients α in $g(x) = \sum_l \alpha_l g_l(x)$ popular, but not exciting for generator generation
  - ∵ selection of g_l requires expert human input
  - ∵ can't deal very efficiently w/ cuts
- fixed grid w/ variable weights



x (i. e. characteristic functions as g_l) not useful at all

- ∵ locally fixed resolution can not adapt to the typical power law singularities over orders of magnitude

- alternative in one dimension: instead of adjusting weights of fixed bins, adjust density of equal weight bins
  - ∵ globally fixed resolution can nevertheless adjust locally over many orders of magnitude:



iteratively adjust grid, use estimates to either

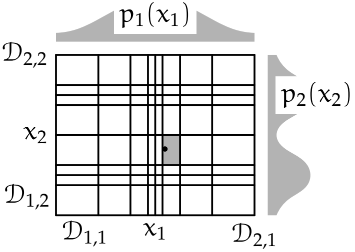
- approximate f locally (importance sampling $\Longrightarrow$ event generation)
- or equidistribute the variance (stratified sampling $\Longrightarrow$ high precision integration).

Factorized ansatz

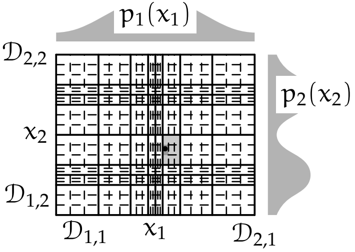
$$g(x) = g_1(x_1)g_2(x_2)\cdots g_n(x_n)$$

- guarantees hypercubic properties and simple one-dimensional formulae (w/ averaging over other dimensions)
- computational costs grow only linearly w/ the number of dimensions

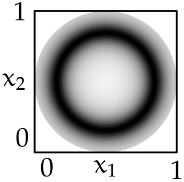
VEGAS grid structure for importance sampling:

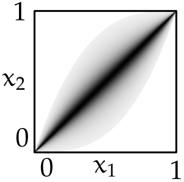


for genuinely stratified sampling, used in low dimensions:



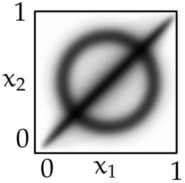
VEGAS' factorized ansatz handles





separately after mappings.

- fails for overlapping singularities



which is the common case
(for more than one 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.

- in general, $g \circ \phi_i$ does not factorize, even if all g_i factorize.
- $\pi_{ij} = \phi_j^{-1} \circ \phi_i$: coordinate transformations among coordinate systems in which different singularities factorize.
- pure geometry: economical studies of dependence on cuts and parameters
  - $\because$ π_{ij} universal and are calculated automatically by WHIZARD
  - $\therefore$ VEGAS can optimize the g_i for each set of parameters and cuts

However:

$\because$ singularity structure determined by Feynman diagrams

- naive application brings the combinatorial explosion in through the back door!

$\therefore$ WHIZARD uses heuristics to select the important channels

  - s-channel resonances
  - $1/t$-poles for massless particles

and permutation symmetries to eliminate equivalent channels

- In Monte Carlo integration of integrals of observables f on phasespace weighted with the cross section

$$\Sigma(f) = \int d\sigma(\Phi)\, f(\Phi) = \int d\Phi\, \frac{d\sigma}{d\Phi}(\Phi)\, f(\Phi)$$

it suffices to generate weighted phase space configurations

$$\mathcal{W} = \{(\phi_i, w_i)\}_{i\in\mathbf{N}}$$

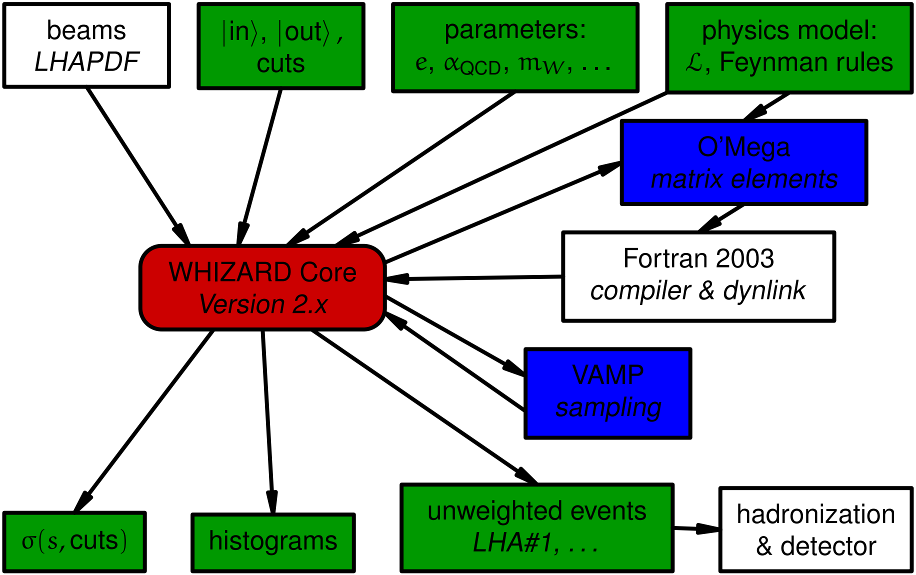
such that the integral is approximated by the weighted sum

$$\Sigma(f) \approx \sum_{i\in\mathbf{N}} w_i f(\phi_i)\,.$$

- In the case of wildly varying cross sections, this is often much simpler than generating generate unweighted phase space configurations $\mathcal{U} = \{\phi_i\}_{i\in\mathbf{N}}$ with

$$\Sigma(f) \approx \sum_{i\in\mathbf{N}} f(\phi_i)\,,$$

- ... by they are required for realistic detector simulations.


beams
LHAPDF
|in⟩, |out⟩ ,
cuts
parameters:
e, αQCD, mW, . . .
physics model:
ℒ, Feynman rules
O'Mega
matrix elements
WHIZARD Core
Version 2.x
Fortran 2003
compiler & dynlink
VAMP
sampling
σ(s, cuts)
histograms
unweighted events
LHA#1, . . .
hadronization
& detector

- the names of particles and couplings depend on the model and can be looked up in the `share/whizard/models` directory, e. g. in

  `/home/HEP/toolbox-1.1.7/whizard/share/whizard/models/SM.mdl`

  for the standard model
- input parameters

```
parameter GF     = 1.16639E-5   # Fermi constant
parameter mZ     = 91.1882      # Z-boson mass
parameter mW     = 80.419       # W-boson mass
parameter mH     = 125          # Higgs mass
parameter alphas = 0.1178       # Strong coupling constant (Z point)
parameter me     = 0.000510997  # electron mass
...
```

- derived parameters

```
derived v     = 1 / sqrt (sqrt (2.) * GF)      # v (Higgs vev)
derived cw    = mW / mZ                         # cos(theta-W)
...
```

- particles

```
particle D_QUARK 1  parton
  spin 1/2  charge -1/3   isospin -1/2  color 3
  name d down
  anti dbar D "d~"
  tex_anti "\bar{d}"
```

- WHIZARD input `xsect_eemm.sin`

```
model = SM
process mumu = e1, E1 => e2, E2
compile
sqrts = 90 GeV
beams = e1, E1
integrate (mumu) { iterations = 2:1000, 3:5000 }
```

- run WHIZARD

```
$ /home/HEP/toolbox-1.1.7/whizard/bin/whizard xsect_eemm.sin
```

- console output

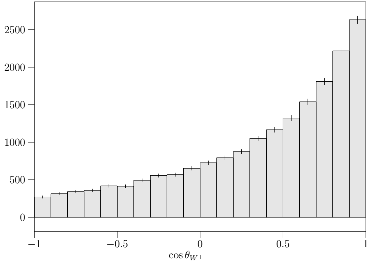
```
| Integrating process 'mumu':
|=============================================================================|
| It      Calls  Integral[fb]  Error[fb]   Err[%]    Acc  Eff[%]   Chi2 N[It] |
|=============================================================================|
   1       1000  1.0605313E+06  8.45E+02    0.08    0.03*  60.96
   2       1000  1.0597762E+06  5.35E+02    0.05    0.02*  61.25
|-----------------------------------------------------------------------------|
   2       2000  1.0599922E+06  4.52E+02    0.04    0.02   61.25    0.57   2
|-----------------------------------------------------------------------------|
   3       5000  1.0601102E+06  1.34E+02    0.01    0.01*  61.12
   4       5000  1.0599916E+06  9.36E+01    0.01    0.01*  78.78
   5       5000  1.0598832E+06  7.58E+01    0.01    0.01*  67.98
|-----------------------------------------------------------------------------|
   5      15000  1.0599559E+06  5.39E+01    0.01    0.01   67.98    1.19   3
|-----------------------------------------------------------------------------|
|=============================================================================|
   5      15000  1.0599559E+06  5.39E+01    0.01    0.01   67.98    1.19   3
```

- WHIZARD input

```
model = SM
alias parton = u:U:d:D:g
process tt = parton, parton => b, Wp, B, Wm
compile
sqrts = 8 TeV
beams = p, p => pdf_builtin
cuts = all Pt > 10 GeV [b:Wp:B:Wm]
integrate (tt) { iterations = 2:10000, 5:20000 }
```

- console output

```
[1/5] gl gl -> b W+ bbar W- ... allowed.
[2/5] d dbar -> b W+ bbar W- ... allowed.
[3/5] dbar d -> b W+ bbar W- ... allowed.
[4/5] u ubar -> b W+ bbar W- ... allowed.
[5/5] ubar u -> b W+ bbar W- ... allowed.
|=============================================================================|
| It      Calls  Integral[fb]  Error[fb]   Err[%]    Acc  Eff[%]   Chi2 N[It] |
|=============================================================================|
   1      10000  9.9611255E+04  2.89E+04   29.06   29.06*   2.02
   2      10000  1.1805759E+05  1.65E+04   13.95   13.95*   2.63
|-----------------------------------------------------------------------------|
   2      20000  1.1354494E+05  1.43E+04   12.61   17.83    2.63    0.31   2
|-----------------------------------------------------------------------------|
   3      20000  1.1757383E+05  1.64E+04   13.91   19.67    1.45
   4      20000  1.2032373E+05  8.96E+03    7.45   10.53*   2.32
   5      20000  1.1948028E+05  5.69E+03    4.76    6.73*   3.09
   6      20000  1.2279117E+05  4.20E+03    3.42    4.84*   3.86
   7      20000  1.2270293E+05  3.64E+03    2.97    4.20*   3.42
|-----------------------------------------------------------------------------|
   7     100000  1.2190186E+05  2.36E+03    1.94    6.13    3.42    0.09   5
|-----------------------------------------------------------------------------|
|=============================================================================|
   7     100000  1.2190186E+05  2.36E+03    1.94    6.13    3.42    0.09   5
|=============================================================================|
|    time estimate for generating 10000 unweighted events: 14m:18s
```

- ```
  model = SM
  process ww = e1, E1 => Wp, Wm
  compile
  $x_label = "$\cos\theta_{W^+}$"
  ?draw_errors = true
  histogram costh (-1, 1, 0.1)
  sqrts = 180 GeV
  beams = e1, E1
  luminosity = 1000 / 1 pbarn
  integrate (ww) { iterations = 2:1000, 5:50000 }
  simulate (ww) { analysis = record costh (eval cos (Theta) [Wm]) }
  compile_analysis { $out_file = "ww.dat" }
  ```
- 
  ```

$pp \to gg \to \mu^- \bar{\nu}_e u \bar{d}$ as a Les Houches Event File (LHEF):

```
model = SM
process hWW = g, g => e2, N2, u, D
compile
sqrts = 14 TeV
beams = p, p => pdf_builtin
sample_format = lhef
simulate (hWW) { n_events = 1 }
```


```
<LesHouchesEvents version="1.0">
<header>
  <generator_name>WHIZARD</generator_name>
  <generator_version>2.1.1</generator_version>
</header>
<init>
        2212        2212   7000.0000000000000        7000.0000000000000               -1          -1          -1          -1
   420.46515665414375        249.25905165127301        1.0000000000000000                1
</init>
<event>
           6           1   1.0000000000000000        136.31643354892731       -1.0000000000000000        0.11780000000000000
          21          -1           0           0         501         502   0.0000000000000000        0.0000000000000000        64.59551
          21          -1           0           0         503         501   0.0000000000000000        0.0000000000000000       -71.91741
          13           1           1           2           0           0   16.068546922717953       -15.231242240310792        28.41895
         -14           1           1           2           0           0  -35.486359371670360        14.938719772529632       -38.73337
           2           1           1           2         503           0   14.687109514917077        13.162760335459950       -13.89070
          -1           1           1           2           0         502   4.7307029340353255       -12.870237867678787        16.88323
</event>
</LesHouchesEvents>
```

$pp \to gg \to \mu^- \bar{\nu}_e u \bar{d}$ as a HepMC file:

```
model = SM
process hWW = g, g => e2, N2, u, D
compile
sqrts = 14 TeV
beams = p, p => pdf_builtin
sample_format = hepmc
simulate (hWW) { n_events = 1 }
```

```
HepMC::Version 2.06.09
HepMC::IO_GenEvent-START_EVENT_LISTING
E 1 -1 1.3631643354892731e+02 1.1780000000000000e-01 -1.0000000000000000e+00 1 0 3 10001 10002 0 4 1.0000000000000000e+00 7.08284436471
N 4 "0" "1" "2" "3"
U GEV MM
C 4.2046515665414375e+02 2.4925905165127301e+02
V -1 0 0 0 0 0 1 2 0
P 10001 2212 0 0 7.0000000000000000e+03 7.0000000000000000e+03 0 4 0 0 -1 0
P 10003 21 0 0 6.4595518178633455e+01 6.4595518178633455e+01 0 3 0 0 -3 2 1 1 2 2
P 10005 93 0 0 6.9354044818213670e+03 6.9354044818213670e+03 0 3 0 0 0 2 1 2 2 1
V -2 0 0 0 0 0 1 2 0
P 10002 2212 0 0 -7.0000000000000000e+03 7.0000000000000000e+03 0 4 0 0 -2 0
P 10004 21 0 0 -7.1917412304487158e+01 7.1917412304487158e+01 0 3 0 0 -3 2 1 3 2 1
P 10006 93 0 0 -6.9280825876955132e+03 6.9280825876955132e+03 0 3 0 0 0 2 1 1 2 3
V -3 0 0 0 0 0 0 4 0
P 10007 13 1.6068546922717953e+01 -1.5231242240310792e+01 2.8418956865688791e+01 3.6025507816263499e+01 1.0565838899979969e-01 1 0 0 0
P 10008 -14 -3.5486359371670360e+01 1.4938719772529632e+01 -3.8733375441153328e+01 5.4614296873280757e+01 0 1 0 0 0 0
P 10009 2 1.4687109514917077e+01 1.3162760335459950e+01 -1.3890707675246947e+01 2.4123043035054472e+01 4.7683715820312500e-07 1 0 0 0 1
P 10010 -1 4.7307029340353255e+00 -1.2870237867678787e+01 1.6883232124857773e+01 2.1750082758521934e+01 4.1295309247228556e-07 1 0 0 0
HepMC::IO_GenEvent-END_EVENT_LISTING
```

```
model = SM
process hWW = g, g => e2, N2, u, D
compile
sqrts = 14 TeV
beams = p, p => lhapdf
sample_format = hepmc, lhef
integrate (hWW) { iterations = 1:1000 }
?ps_fsr_active = true
?ps_isr_active = true
?hadronization_active = true
?ps_use_PYTHIA_shower = true
simulate (hWW) { n_events = 1 }
```

```
<event>
     310       1  1.0000000000000000      103.17887552082534      -1.0000000000000000      0.11780000000000000
      21      -1       0       0     501     502   0.0000000000000000     0.0000000000000000     22.60643
      21      -1       0       0     502     503   0.0000000000000000     0.0000000000000000    -117.7306
      93       3       0       0     502     501   0.0000000000000000     0.0000000000000000     6977.393
      93       3       0       0     503     502   0.0000000000000000     0.0000000000000000    -6882.269
     -14       1       1       2       0       0  -27.571846966921186     11.001180372709559    -41.40772
      13       1       1       2       0       0   8.2564499233431601    -44.666087683248804    -32.04015
      22       1       1       2       0       0 -0.60053536107337491    -4.8109348517593622    -4.105068
     211       1       1       2       0       0  0.18025914014147792    0.23261429271929662    0.4834233
    -211       1       1       2       0       0 -0.83369544321947398    0.25818955197077087     1.536586
    2112       1       1       2       0       0 -0.34678766523372612     2.0518746878398826    0.9660215
   -2212       1       1       2       0       0  -7.7893814158812913E-002  1.0446949824101728   2.571061
    2212       1       1       2       0       0  0.46265670377713941     2.0358472993730437     3.421707
   -2212       1       1       2       0       0  0.71417103301686868     1.6232688240042823     2.751320
     211       1       1       2       0       0   2.7762381422163100     3.4505228032902853     9.001631
    -211       1       1       2       0       0  0.65565726609788855    0.75407218229487294     1.674591
     211       1       1       2       0       0   1.1650307289886463    0.58079731055743034     4.041057
    -211       1       1       2       0       0  0.10194216370164123    -4.8888582506378710E-002  7.609617
    -321       1       1       2       0       0   1.2505517529312573    0.24270629084564452    -1.790215
...
```

- Compute the cross section for single top production

$$pp \to tb$$

in the standard model at the Tevatron and at the 14 TeV LHC.
- Plot the rapidity distribution of the produced top quark.
- Include the two-particle decay of the top quark.
- Include the irreducible backgrounds.
- Estimate the influence of bottom-quark parton distribution functions.