Computational Techniques for the LHC

Thorsten Ohl

Institute for Theoretical Physics and Astrophysics, University of Würzburg

Theory Challenges for LHC Physics
HISS 2015, JINR Dubna
July 20th-30th, 2015

- 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

- 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
  - ...

- 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 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

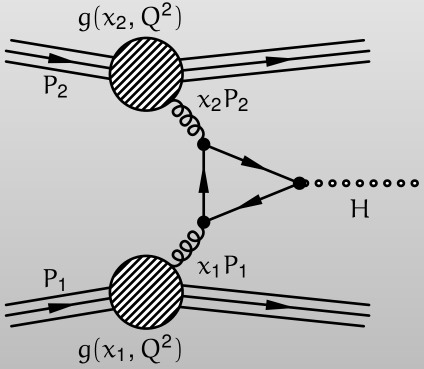
- efficiently and reliably compute scattering probabilities

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

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

- **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!

- 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

- 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
  - universal strongly coupled long distance/low energy phenomena, described by parametrizations

    and/or phenomenological models

- 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

    and/or phenomenological models

- 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

- 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 $D_{i_j}(x_j; \mu)$
    - NB: first lattice results from Mellin moments $\langle p | O^{\mu_1 \cdots \mu_n} | p \rangle$ appeared recently

  and/or phenomenological models

- 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 $D_{i_j}(x_j;\mu)$
  - NB: first lattice results from Mellin moments $\langle p|O^{\mu_1\cdots\mu_n}|p\rangle$ appeared recently

  and/or phenomenological models
  - fragmentation and hadronization

- 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 $D_{i_j}(x_j; \mu)$
    - NB: first lattice results from Mellin moments $\langle p | O^{\mu_1 \cdots \mu_n} | p \rangle$ appeared recently

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

- 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 $D_{i_j}(x_j;\mu)$
    - NB: first lattice results from Mellin moments $\langle p|O^{\mu_1\cdots\mu_n}|p\rangle$ appeared recently

    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!

- 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

- 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, ...)

- 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

- 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).

- 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

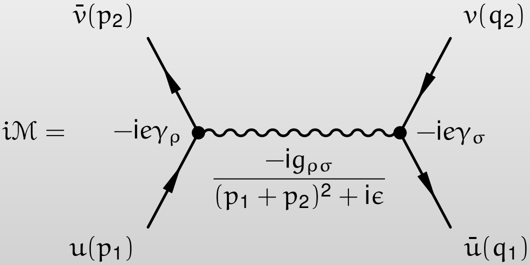
$$\underbrace{\mathbf{R} \times \ldots \times \mathbf{R}}_{\text{masses, couplings, \ldots}} \times \underbrace{V^+ \times \ldots \times V^+}_{\text{4-momenta (forward light cones)}} \times \underbrace{\mathbf{Z} \times \ldots \times \mathbf{Z}}_{\text{helicities, colors}} \to \underbrace{\mathbf{C}}_{\text{amplitude}}$$

    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

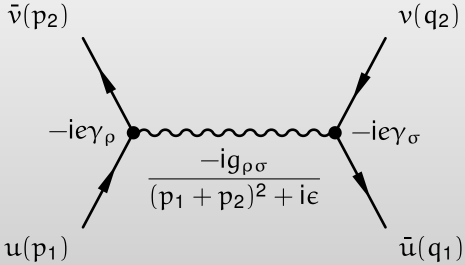
- NB: in some cases only $\mathcal{L} \to \sum |\mathcal{M}(\alpha_1, \ldots; p_1, \ldots; s_1, \ldots \ldots)|^2$ is required. It is often better defined (infrared/collinear cancellations) and sometimes more compact (spin/polarization sums).
- first robust and usable examples in the early 1990s: CompHEP,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

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

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



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

$i\mathcal{M} =$ 

- analytical expression

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

- e. g. command line for O'Mega to compute $e^-e^+ \to \mu^-\mu^+$ in QED

```
$ omega_QED -scatter "e- e+ -> m- m+"
```

- e. g. command line for O'Mega to compute $e^- e^+ \to \mu^- \mu^+$ in QED

  ```
  $ omega_QED -scatter "e- e+ -> m- m+"
  ```

- resulting Fortran95 code

```
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
```

(some additional interface routines suppressed) full disclosure: current O'Mega/WHIZARD version:

```
subroutine calculate_amplitudes (amp, k, mask)
  complex(kind=default), dimension(:,:,:), intent(out) :: amp
  real(kind=default), dimension(0:3,*), intent(in) :: k
  logical, dimension(:), intent(in) :: mask
  integer, dimension(n_prt) :: s
  integer :: h
  p1 = - k(:,1) ! incoming
  p2 = - k(:,2) ! incoming
  p3 =   k(:,3) ! outgoing
  p4 =   k(:,4) ! outgoing
  p12 = p1 + p2
  amp = 0
  do h = 1, n_hel
   if (mask(h)) then
    s = table_spin_states(:,h)
    owf_pos_1 = vbar (mass(11), - p1, s(1))
    owf_ele_2 = u (mass(11), - p2, s(2))
    owf_muo_3 = v (mass(13), p3, s(3))
    owf_amu_4 = ubar (mass(13), p4, s(4))
    call compute_fusions_0001 ()          ! help compiler by breaking code into smaller chunks
    call compute_brakets_0001 ()
    amp(1,h,1) = oks_poseleamumuo         ! compute amplitudes at once
   end if
  end do
end subroutine calculate_amplitudes
subroutine compute_fusions_0001 ()
    owf_gam_12 = pr_feynman(p12, + v_ff(qlep,owf_pos_1,owf_ele_2))
end subroutine compute_fusions_0001
subroutine compute_brakets_0001 ()
    oks_poseleamumuo = 0
    oks_poseleamumuo = oks_poseleamumuo + owf_gam_12*( + v_ff(qlep,owf_amu_4,owf_muo_3))
    oks_poseleamumuo = - oks_poseleamumuo ! 2 vertices, 1 propagators
end subroutine compute_brakets_0001
```

- the usual rules for manual calculations are algorithmic

- ☺ the usual rules for manual calculations are algorithmic
- ∴ can be implemented in a computer program

- ☺ the usual rules for manual calculations are algorithmic
- ∴ can be implemented in a computer program
- ▸ for a given set of Feynman rules and external particles

- the usual rules for manual calculations are algorithmic
- ∴ can be implemented in a computer program
- ▶ for a given set of Feynman rules and external particles
  1. *draw all topologically inequivalent Feynman diagrams connecting the external particles (up to the chosen number of loops)*

☺ the usual rules for manual calculations are algorithmic

∴ can be implemented in a computer program

- for a given set of Feynman rules and external particles
  1. *draw all topologically inequivalent Feynman diagrams connecting the external particles (up to the chosen number of loops)*
  2. *in each diagram, assign a 4-momentum to each internal line according to 4-momentum conservation at each vertex*

- the usual rules for manual calculations are algorithmic
- ∴ can be implemented in a computer program
- for a given set of Feynman rules and external particles
  1. *draw all topologically inequivalent Feynman diagrams connecting the external particles (up to the chosen number of loops)*
  2. *in each diagram, assign a 4-momentum to each internal line according to 4-momentum conservation at each vertex*
  3. *in each diagram, replace each external line with a wave function according to the particle type and momentum*

☺ the usual rules for manual calculations are algorithmic

∴ can be implemented in a computer program

▶ for a given set of Feynman rules and external particles

1. *draw all topologically inequivalent Feynman diagrams connecting the external particles (up to the chosen number of loops)*
2. *in each diagram, assign a 4-momentum to each internal line according to 4-momentum conservation at each vertex*
3. *in each diagram, replace each external line with a wave function according to the particle type and momentum*
4. *in each diagram, replace each internal line with a propagator according to the particle type and momentum*

☺ the usual rules for manual calculations are algorithmic

∴ can be implemented in a computer program

- for a given set of Feynman rules and external particles
  1. *draw all topologically inequivalent Feynman diagrams connecting the external particles (up to the chosen number of loops)*
  2. *in each diagram, assign a 4-momentum to each internal line according to 4-momentum conservation at each vertex*
  3. *in each diagram, replace each external line with a wave function according to the particle type and momentum*
  4. *in each diagram, replace each internal line with a propagator according to the particle type and momentum*
  5. *for each diagram, replace each vertex with a vertex function according to Feynman rules*

☺ the usual rules for manual calculations are algorithmic

∴ can be implemented in a computer program

- for a given set of Feynman rules and external particles
  1. *draw all topologically inequivalent Feynman diagrams connecting the external particles (up to the chosen number of loops)*
  2. *in each diagram, assign a 4-momentum to each internal line according to 4-momentum conservation at each vertex*
  3. *in each diagram, replace each external line with a wave function according to the particle type and momentum*
  4. *in each diagram, replace each internal line with a propagator according to the particle type and momentum*
  5. *for each diagram, replace each vertex with a vertex function according to Feynman rules*
  6. *divide each diagram by the size of its automorphism group*

- ☺ the usual rules for manual calculations are algorithmic
- ∴ can be implemented in a computer program
- ▸ for a given set of Feynman rules and external particles
  1. *draw all topologically inequivalent Feynman diagrams connecting the external particles (up to the chosen number of loops)*
  2. *in each diagram, assign a 4-momentum to each internal line according to 4-momentum conservation at each vertex*
  3. *in each diagram, replace each external line with a wave function according to the particle type and momentum*
  4. *in each diagram, replace each internal line with a propagator according to the particle type and momentum*
  5. *for each diagram, replace each vertex with a vertex function according to Feynman rules*
  6. *divide each diagram by the size of its automorphism group*
  7. *multiply each diagram with a factor of $+1$ or -1 in order to make the sum of the diagrams symmetric/antisymmetric under the exchange of bosons/fermions*

☺ the usual rules for manual calculations are algorithmic

∴ can be implemented in a computer program

- for a given set of Feynman rules and external particles
  1. *draw all topologically inequivalent Feynman diagrams connecting the external particles (up to the chosen number of loops)*
  2. *in each diagram, assign a 4-momentum to each internal line according to 4-momentum conservation at each vertex*
  3. *in each diagram, replace each external line with a wave function according to the particle type and momentum*
  4. *in each diagram, replace each internal line with a propagator according to the particle type and momentum*
  5. *for each diagram, replace each vertex with a vertex function according to Feynman rules*
  6. *divide each diagram by the size of its automorphism group*
  7. *multiply each diagram with a factor of $+1$ or -1 in order to make the sum of the diagrams symmetric/antisymmetric under the exchange of bosons/fermions*
  8. *finally: add them up!*

- NB: familiar trace techniques for

$$\sum_{s_1,\ldots} |\mathcal{M}(\alpha_1,\ldots;p_1,\ldots,;s_1,\ldots\ldots)|^2$$

rarely used, because
  - tedious for polarized scattering (longer traces, due to spin projections)

- NB: familiar trace techniques for

$$\sum_{s_1,\ldots} |\mathcal{M}(\alpha_1,\ldots;p_1,\ldots,;s_1,\ldots\ldots)|^2$$

rarely used, because

- tedious for polarized scattering (longer traces, due to spin projections)
- intermediate expressions grow quadratically with the number of Feynman diagrams

$$\left|\sum_i^{N_D} \mathcal{M}_i\right|^2 = \sum_{i=1}^{N_D}\sum_{j=1}^{N_D} \mathcal{M}_i^* \mathcal{M}_j$$

- NB: familiar trace techniques for

$$\sum_{s_1,\ldots} |\mathcal{M}(\alpha_1,\ldots;p_1,\ldots,;s_1,\ldots\ldots)|^2$$

rarely used, because

  - tedious for polarized scattering (longer traces, due to spin projections)
  - intermediate expressions grow quadratically with the number of Feynman diagrams

$$\left|\sum_i^{N_D} \mathcal{M}_i\right|^2 = \sum_{i=1}^{N_D}\sum_{j=1}^{N_D} \mathcal{M}_i^* \mathcal{M}_j$$

    - number of polarization states grows with a power of the number of particles

- NB: familiar trace techniques for

$$\sum_{s_1,\ldots} |\mathcal{M}(\alpha_1,\ldots;p_1,\ldots,;s_1,\ldots\ldots)|^2$$

rarely used, because

  - tedious for polarized scattering (longer traces, due to spin projections)
  - intermediate expressions grow quadratically with the number of Feynman diagrams

$$\left|\sum_{i}^{N_D} \mathcal{M}_i\right|^2 = \sum_{i=1}^{N_D}\sum_{j=1}^{N_D} \mathcal{M}_i^* \mathcal{M}_j$$

    - number of polarization states grows with a power of the number of particles
    - number of Feynman diagrams grows with a factorial of the number of particles

- NB: familiar trace techniques for

$$\sum_{s_1,\ldots} |\mathcal{M}(\alpha_1,\ldots;p_1,\ldots,;s_1,\ldots\ldots)|^2$$

rarely used, because

- tedious for polarized scattering (longer traces, due to spin projections)
- intermediate expressions grow quadratically with the number of Feynman diagrams

$$\left|\sum_i^{N_D} \mathcal{M}_i\right|^2 = \sum_{i=1}^{N_D}\sum_{j=1}^{N_D} \mathcal{M}_i^*\mathcal{M}_j$$

- number of polarization states grows with a power of the number of particles
- number of Feynman diagrams grows with a factorial of the number of particles

∴ helicity amplitudes win (eventually)!

- 4 of the 8 steps are straightforward translations
  3. *in each diagram, replace each external line with a wave function according to the particle type and momentum*
  4. *in each diagram, replace each internal line with a propagator according to the particle type and momentum*
  5. *for each diagram, replace each vertex with a vertex function according to Feynman rules*
  8. *finally: add them up!*

- 4 of the 8 steps are straightforward translations
    3. *in each diagram, replace each external line with a wave function according to the particle type and momentum*
    4. *in each diagram, replace each internal line with a propagator according to the particle type and momentum*
    5. *for each diagram, replace each vertex with a vertex function according to Feynman rules*
    8. *finally: add them up!*

∴ big boring `case` statements, only problem: striking a balance between legibility (extending and debugging!) and efficiency

- 4 of the 8 steps are straightforward translations
  3. *in each diagram, replace each external line with a wave function according to the particle type and momentum*
  4. *in each diagram, replace each internal line with a propagator according to the particle type and momentum*
  5. *for each diagram, replace each vertex with a vertex function according to Feynman rules*
  8. *finally: add them up!*

∴ big boring `case` statements, only problem: striking a balance between legibility (extending and debugging!) and efficiency

- 1 of the 8 steps is straightforward only for tree diagrams
  2. *in each diagram, assign a 4-momentum to each internal line according to 4-momentum conservation at each vertex*

in tree diagrams all 4-momenta are determined by 4-momentum conservation and have a unique representation as a linear combination of $n-1$ external momenta with coefficients $\{-1, 0, 1\}$.

- 4 of the 8 steps are straightforward translations
  3. *in each diagram, replace each external line with a wave function according to the particle type and momentum*
  4. *in each diagram, replace each internal line with a propagator according to the particle type and momentum*
  5. *for each diagram, replace each vertex with a vertex function according to Feynman rules*
  8. *finally: add them up!*

∴ big boring `case` statements, only problem: striking a balance between legibility (extending and debugging!) and efficiency

- 1 of the 8 steps is straightforward only for tree diagrams
  2. *in each diagram, assign a 4-momentum to each internal line according to 4-momentum conservation at each vertex*

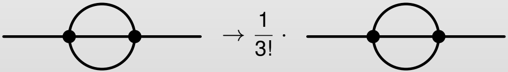
  in tree diagrams all 4-momenta are determined by 4-momentum conservation and have a unique representation as a linear combination of $n-1$ external momenta with coefficients $\{-1, 0, 1\}$.
- in loop diagrams there are undermined loop momenta that must (usually) be integrated analytically: $\int d^4 p_i$

- the 3 remaining steps require non-trivial algorithms

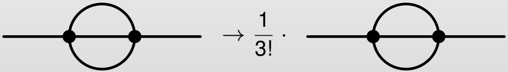
- the 3 remaining steps require non-trivial algorithms
    1. *draw all topologically inequivalent Feynman diagrams connecting the external particles (up to the chosen number of loops)*

- the 3 remaining steps require non-trivial algorithms
  1. *draw all topologically inequivalent Feynman diagrams connecting the external particles (up to the chosen number of loops)*
  6. *divide each diagram by the size of its automorphism group*

- the 3 remaining steps require non-trivial algorithms
  1. *draw all topologically inequivalent Feynman diagrams connecting the external particles (up to the chosen number of loops)*
  6. *divide each diagram by the size of its automorphism group*
     - e. g.

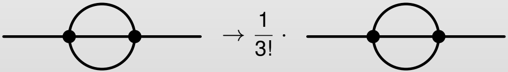


- the 3 remaining steps require non-trivial algorithms
  1. *draw all topologically inequivalent Feynman diagrams connecting the external particles (up to the chosen number of loops)*
  6. *divide each diagram by the size of its automorphism group*
     - e. g.



     - only $\neq 1$ for loop diagrams, no general polynomial algorithm

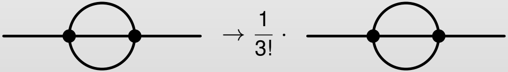
- the 3 remaining steps require non-trivial algorithms
  1. *draw all topologically inequivalent Feynman diagrams connecting the external particles (up to the chosen number of loops)*
  6. *divide each diagram by the size of its automorphism group*
     - e. g.



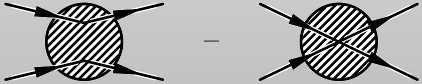
     - only $\neq 1$ for loop diagrams, no general polynomial algorithm
  7. *multiply each diagram with a factor of $+1$ or -1 in order to make the sum of the diagrams symmetric/antisymmetric under the exchange of bosons/fermions*

- the 3 remaining steps require non-trivial algorithms

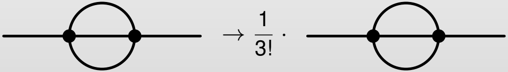
1. *draw all topologically inequivalent Feynman diagrams connecting the external particles (up to the chosen number of loops)*
6. *divide each diagram by the size of its automorphism group*
   - e. g.



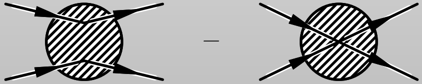
   - only $\neq 1$ for loop diagrams, no general polynomial algorithm
7. *multiply each diagram with a factor of $+1$ or -1 in order to make the sum of the diagrams symmetric/antisymmetric under the exchange of bosons/fermions*
   - if all fermions are Dirac fermions, i. e. their lines can be followed through the diagram, use the sign of the permutation of the endpoints



- the 3 remaining steps require non-trivial algorithms
  1. *draw all topologically inequivalent Feynman diagrams connecting the external particles (up to the chosen number of loops)*
  6. *divide each diagram by the size of its automorphism group*
     - e. g.



     - only $\neq 1$ for loop diagrams, no general polynomial algorithm
  7. *multiply each diagram with a factor of $+1$ or -1 in order to make the sum of the diagrams symmetric/antisymmetric under the exchange of bosons/fermions*
     - if all fermions are Dirac fermions, i. e. their lines can be followed through the diagram, use the sign of the permutation of the endpoints



     - if Majorana fermions are present (MSSM!), use the clever algorithm of [Denner et al. Phys. Lett. **B291**, 278 (1992)]

- Challenge: generate each diagram exactly once

- Challenge: generate each diagram exactly once
- ideal representation: one-to-one correspondence with diagrams

- Challenge: generate each diagram exactly once
- ideal representation: one-to-one correspondence with diagrams

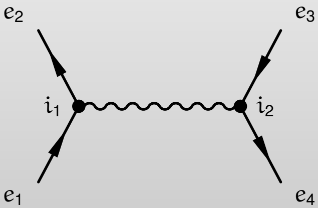
☹ impossible, in general

- Challenge: generate each diagram exactly once
- ideal representation: one-to-one correspondence with diagrams

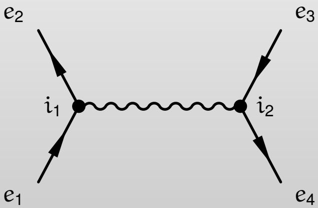
☹ impossible, in general

- choices

- Challenge: generate each diagram exactly once
- ideal representation: one-to-one correspondence with diagrams
- ☹ impossible, in general
- choices
  0. diagram (not machine readable!)



- Challenge: generate each diagram exactly once
- ideal representation: one-to-one correspondence with diagrams
- ☹ impossible, in general
- choices
  0. diagram (not machine readable!)



  1. sets of external and internal vertices and edges

$$\left(\begin{Bmatrix}(e_1, e^-, p_1)\\(e_2, e^+, p_2)\\(e_3, \mu^+, q_2)\\(e_4, \mu^-, q_1)\end{Bmatrix}, \begin{Bmatrix}(i_1, -ie\gamma_\mu)\\(i_2, -ie\gamma_\mu)\end{Bmatrix}, \begin{Bmatrix}(e_1, i_1, \{e^-\})\\(e_2, i_1, \{e^+\})\\(e_3, i_2, \{\mu^-\})\\(e_4, i_2, \{\mu^+\})\\(i_1, i_2, \{\gamma\})\end{Bmatrix}\right)$$

- choices (cont'd)
  2. incidence matrix

$$\left\{\begin{matrix}(e_1, i_1, \{e^-\})\\(e_2, i_1, \{e^+\})\\(e_3, i_2, \{\mu^-\})\\(e_4, i_2, \{\mu^+\})\\(i_1, i_2, \{\gamma\})\end{matrix}\right\} \sim$$

	e_1	e_2	e_3	e_4	i_1	i_2
e_1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\{e^-\}$	$\emptyset$
e_2	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\{e^+\}$	$\emptyset$
e_3	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\{\mu^-\}$
e_4	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\{\mu^+\}$
i_1	$\{e^+\}$	$\{e^-\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{\gamma\}$
i_2	$\emptyset$	$\emptyset$	$\{\mu^+\}$	$\{\mu^-\}$	$\{\gamma\}$	$\emptyset$

- choices (cont'd)
  2. incidence matrix

$$\begin{Bmatrix} (e_1, i_1, \{e^-\}) \\ (e_2, i_1, \{e^+\}) \\ (e_3, i_2, \{\mu^-\}) \\ (e_4, i_2, \{\mu^+\}) \\ (i_1, i_2, \{\gamma\}) \end{Bmatrix} \sim$$

	e_1	e_2	e_3	e_4	i_1	i_2
e_1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\{e^-\}$	$\emptyset$
e_2	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\{e^+\}$	$\emptyset$
e_3	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\{\mu^-\}$
e_4	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\{\mu^+\}$
i_1	$\{e^+\}$	$\{e^-\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{\gamma\}$
i_2	$\emptyset$	$\emptyset$	$\{\mu^+\}$	$\{\mu^-\}$	$\{\gamma\}$	$\emptyset$

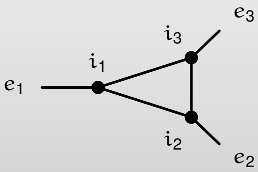
☹ different egde set and incidence matrix for same graph

$$\begin{Bmatrix} (e_1, i_2, \{e^-\}) \\ (e_2, i_2, \{e^+\}) \\ (e_3, i_1, \{\mu^-\}) \\ (e_4, i_1, \{\mu^+\}) \\ (i_1, i_2, \{\gamma\}) \end{Bmatrix} \sim$$

	e_1	e_2	e_3	e_4	i_1	i_2
e_1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\{e^-\}$
e_2	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\{e^+\}$
e_3	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\{\mu^-\}$	$\emptyset$
e_4	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\{\mu^+\}$	$\emptyset$
i_1	$\emptyset$	$\emptyset$	$\{\mu^+\}$	$\{\mu^-\}$	$\emptyset$	$\{\gamma\}$
i_2	$\{e^+\}$	$\{e^-\}$	$\emptyset$	$\emptyset$	$\{\gamma\}$	$\emptyset$

- ambiguity: external vertices distinguished by the momentum, internal vertices named ad-hoc

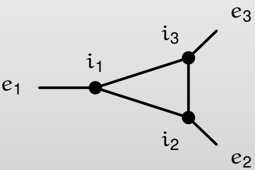
- ambiguity: external vertices distinguished by the momentum, internal vertices named ad-hoc
- solution: canonical ordering of internal vertices, e. g.



$$\sim \left\{ \begin{matrix} (e_1, i_1) \\ (e_2, i_2) \\ (e_3, i_3) \\ (i_1, i_2) \\ (i_2, i_3) \\ (i_3, i_1) \end{matrix} \right\} \sim$$

	e_1	e_2	e_3	i_1	i_2	i_3
e_1				1		
e_2					1	
e_3						1
i_1	1				1	
i_2		1				1
i_3			1	1		

- ambiguity: external vertices distinguished by the momentum, internal vertices named ad-hoc
- solution: canonical ordering of internal vertices, e. g.

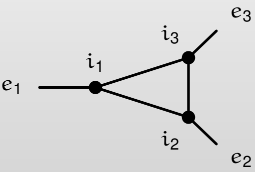


$$\sim \left\{ \begin{matrix} (e_1, i_1) \\ (e_2, i_2) \\ (e_3, i_3) \\ (i_1, i_2) \\ (i_2, i_3) \\ (i_3, i_1) \end{matrix} \right\} \sim$$

	e_1	e_2	e_3	i_1	i_2	i_3
e_1				1		
e_2					1	
e_3						1
i_1	1				1	
i_2		1				1
i_3			1	1		

- rows/columns ordered lexicographically

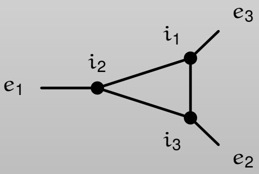
- ambiguity: external vertices distinguished by the momentum, internal vertices named ad-hoc
- solution: canonical ordering of internal vertices, e. g.



$$\sim \begin{Bmatrix} (e_1, i_1) \\ (e_2, i_2) \\ (e_3, i_3) \\ (i_1, i_2) \\ (i_2, i_3) \\ (i_3, i_1) \end{Bmatrix} \sim$$

	e_1	e_2	e_3	i_1	i_2	i_3
e_1				1		
e_2					1	
e_3						1
i_1	1				1	
i_2		1				1
i_3			1	1		

- rows/columns ordered lexicographically, in contrast to



$$\sim \begin{Bmatrix} (e_1, i_2) \\ (e_2, i_3) \\ (e_3, i_1) \\ (i_1, i_2) \\ (i_2, i_3) \\ (i_3, i_1) \end{Bmatrix} \sim$$

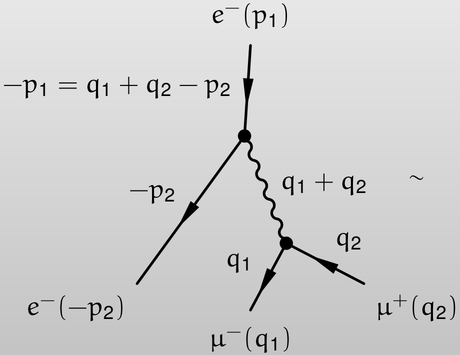
	e_1	e_2	e_3	i_1	i_2	i_3
e_1					1	
e_2						1
e_3				1		
i_1			1		1	
i_2	1					1
i_3		1		1		

- tree diagrams are special: no need to enumerate vertices

- tree diagrams are special: no need to enumerate vertices

∵ all propagators are uniquely specified by their momentum

- tree diagrams are special: no need to enumerate vertices
- ∵ all propagators are uniquely specified by their momentum
- crossing symmetry allows drawing as mathematical tree

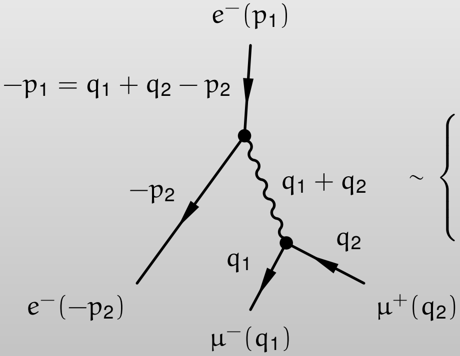


$\sim$

- tree diagrams are special: no need to enumerate vertices

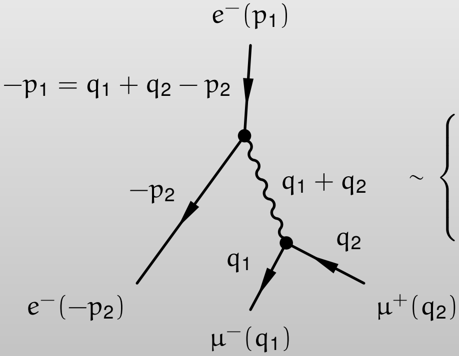
∵ all propagators are uniquely specified by their momentum

- crossing symmetry allows drawing as mathematical tree



$$\sim \left\{ \begin{array}{l} \gamma(q_1+q_2) \mapsto \\ \qquad (\mu^-(q_1), \mu^+(q_2)) \end{array} \right\}$$

- tree diagrams are special: no need to enumerate vertices
- ∵ all propagators are uniquely specified by their momentum
- crossing symmetry allows drawing as mathematical tree

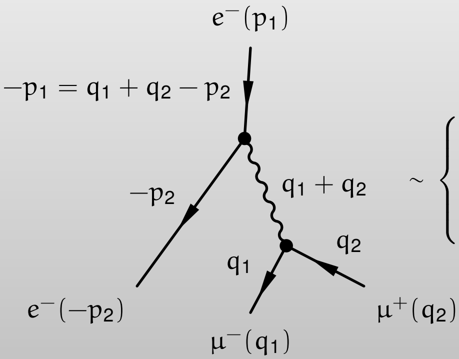


$$\sim \left\{ \begin{array}{l} e^-(q_1+q_2-p_2) \mapsto \\ \qquad (e^-(-p_2), \gamma(q_1+q_2)) \\ \quad \gamma(q_1+q_2) \mapsto \\ \qquad (\mu^-(q_1), \mu^+(q_2)) \end{array} \right\}$$

- tree diagrams are special: no need to enumerate vertices

∵ all propagators are uniquely specified by their momentum

- crossing symmetry allows drawing as mathematical tree



$$\sim \left\{ \begin{array}{l} e^-(q_1 + q_2 - p_2) \mapsto \\ \qquad (e^-(-p_2), \gamma(q_1 + q_2)) \\ \quad \gamma(q_1 + q_2) \mapsto \\ \qquad (\mu^-(q_1), \mu^+(q_2)) \end{array} \right\}$$

- a harmless 2-fold ambiguity from overall momentum conservation can be avoided, if we don't use the momentum at the root, i. e. p_1

- ☹ searching all diagrams directly using the Feynman rules is inefficient, because too many dead ends

- searching all diagrams directly using the Feynman rules is inefficient, because too many dead ends
  - more efficient 2 step approach

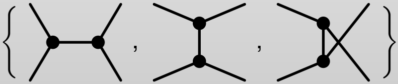
- searching all diagrams directly using the Feynman rules is inefficient, because too many dead ends
  - more efficient 2 step approach
    1. generate all topologies, i. e. discard all quantum numbers (mass, spin, charges, flavor, color) and generate all Feynman diagrams with given number of loops and legs with Lagrangian

$$\mathcal{L}_{\text{topologies}} = \frac{1}{2}\partial_\mu\phi\partial^\mu\phi - \frac{\lambda_3}{3!}\phi^3 - \frac{\lambda_4}{4!}\phi^4 - \ldots$$

- searching all diagrams directly using the Feynman rules is inefficient, because too many dead ends
  - more efficient 2 step approach
    1. generate all topologies, i. e. discard all quantum numbers (mass, spin, charges, flavor, color) and generate all Feynman diagrams with given number of loops and legs with Lagrangian

$$\mathcal{L}_{\text{topologies}} = \frac{1}{2}\partial_\mu\phi\partial^\mu\phi - \frac{\lambda_3}{3!}\phi^3 - \frac{\lambda_4}{4!}\phi^4 - \ldots$$

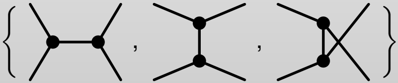
e. g.



- searching all diagrams directly using the Feynman rules is inefficient, because too many dead ends
  - more efficient 2 step approach
    1. generate all topologies, i. e. discard all quantum numbers (mass, spin, charges, flavor, color) and generate all Feynman diagrams with given number of loops and legs with Lagrangian

$$\mathcal{L}_{\text{topologies}} = \frac{1}{2}\partial_\mu\phi\partial^\mu\phi - \frac{\lambda_3}{3!}\phi^3 - \frac{\lambda_4}{4!}\phi^4 - \ldots$$

e. g.

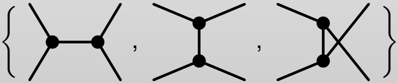


    2. add quantum numbers in all ways compatible with the Feynman rules and the external quantum numbers

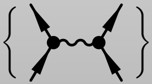
- searching all diagrams directly using the Feynman rules is inefficient, because too many dead ends
  - more efficient 2 step approach
    1. generate all topologies, i. e. discard all quantum numbers (mass, spin, charges, flavor, color) and generate all Feynman diagrams with given number of loops and legs with Lagrangian

$$\mathcal{L}_{\text{topologies}} = \frac{1}{2}\partial_\mu\phi\partial^\mu\phi - \frac{\lambda_3}{3!}\phi^3 - \frac{\lambda_4}{4!}\phi^4 - \ldots$$

e. g.



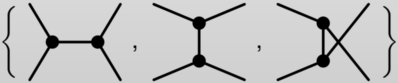
    2. add quantum numbers in all ways compatible with the Feynman rules and the external quantum numbers , e. g.



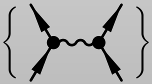
- searching all diagrams directly using the Feynman rules is inefficient, because too many dead ends
  - more efficient 2 step approach
    1. generate all topologies, i. e. discard all quantum numbers (mass, spin, charges, flavor, color) and generate all Feynman diagrams with given number of loops and legs with Lagrangian

$$\mathcal{L}_{\text{topologies}} = \frac{1}{2}\partial_\mu\phi\partial^\mu\phi - \frac{\lambda_3}{3!}\phi^3 - \frac{\lambda_4}{4!}\phi^4 - \ldots$$

e. g.



    2. add quantum numbers in all ways compatible with the Feynman rules and the external quantum numbers , e. g.

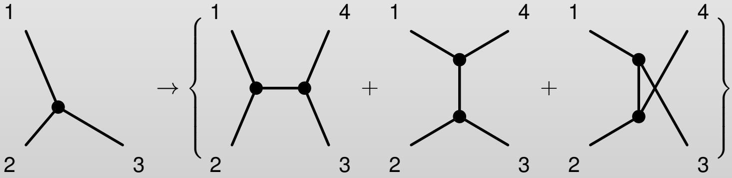


- avoids an enormous number of fruitless attempts

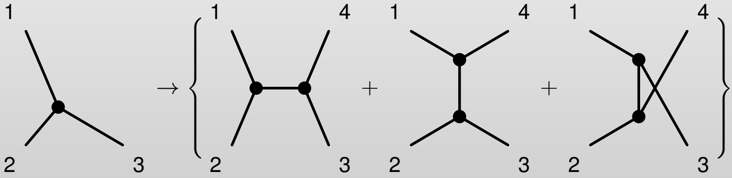
- Algorithm for generating all tree diagram topologies with n external legs

- Algorithm for generating all tree diagram topologies with n external legs
  1. generate all tree diagrams with $n - 1$ external legs (to iterate is human, to recurse is divine . . .)

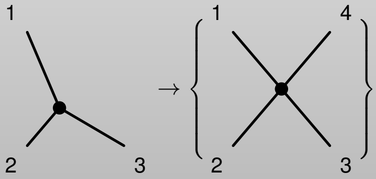
- Algorithm for generating all tree diagram topologies with n external legs
  1. generate all tree diagrams with $n-1$ external legs (to iterate is human, to recurse is divine ...)
  2. insert the nth line once
     - in every propagator, e. g.



- Algorithm for generating all tree diagram topologies with n external legs
  1. generate all tree diagrams with $n-1$ external legs (to iterate is human, to recurse is divine ...)
  2. insert the nth line once
     - in every propagator, e. g.



     - in very vertex of degree less than the maximum, e. g.



- corollary: there are

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

tree topologies with n legs

- corollary: there are

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

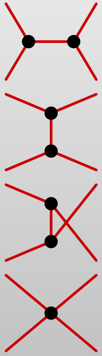
  tree topologies with n legs
- proof:

$$\begin{aligned} F(3) &= 1 \\ F(n) &= (\underbrace{(n-1)}_{\text{external lines}} + \underbrace{(n-4)}_{\text{internal lines}})\cdot F(n-1) \\ &= (2n-5)\cdot F(n-1) = (2n-5)\cdot(2n-7)\cdot F(n-2) = \ldots \end{aligned}$$

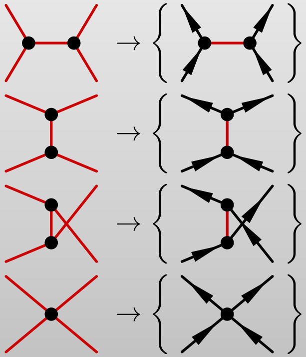
- adding quantum numbers from the outside in

- adding quantum numbers from the outside in
- e. g. $e^- e^+ \to \mu^- \mu^+$

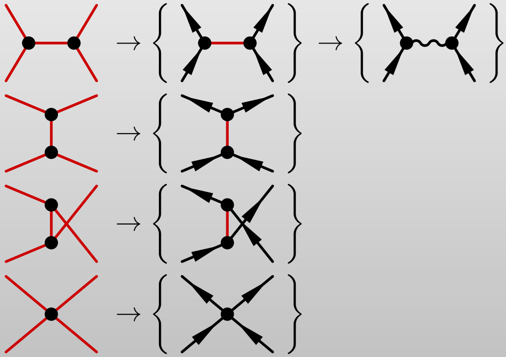
- adding quantum numbers from the outside in
- e. g. $e^- e^+ \to \mu^- \mu^+$



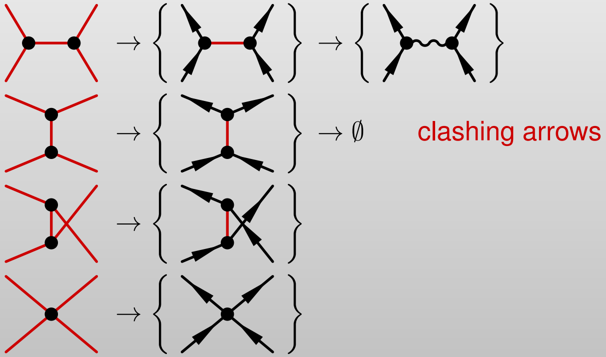
- adding quantum numbers from the outside in
- e. g. $e^- e^+ \to \mu^- \mu^+$



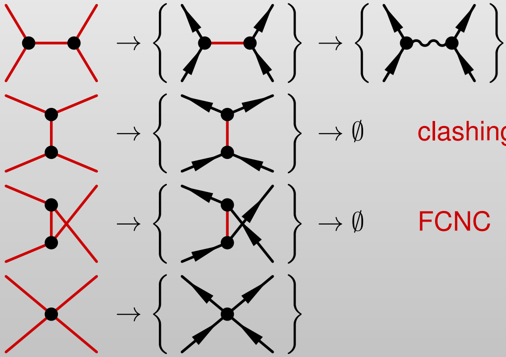
- adding quantum numbers from the outside in
- e. g. $e^- e^+ \to \mu^- \mu^+$



- adding quantum numbers from the outside in
- e. g. $e^- e^+ \to \mu^- \mu^+$



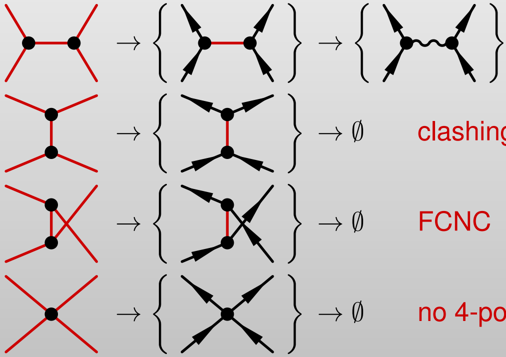
- adding quantum numbers from the outside in
- e. g. $e^- e^+ \to \mu^- \mu^+$



$\to \emptyset$ clashing arrows

$\to \emptyset$ FCNC

- adding quantum numbers from the outside in
- e. g. $e^- e^+ \to \mu^- \mu^+$



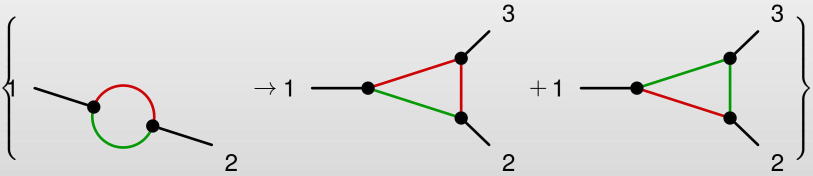
clashing arrows

FCNC

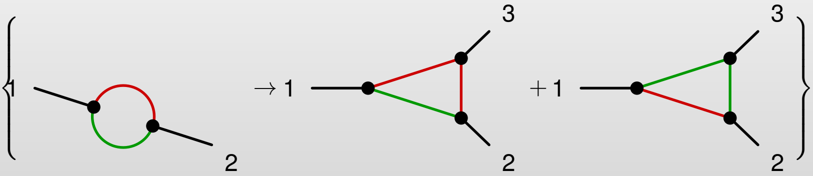
no 4-point vertex

- simple tree level algorithm $n \to n + 1$ doesn't work for loop diagrams because

- simple tree level algorithm $n \to n+1$ doesn't work for loop diagrams because
  - diagrams will appear more than once

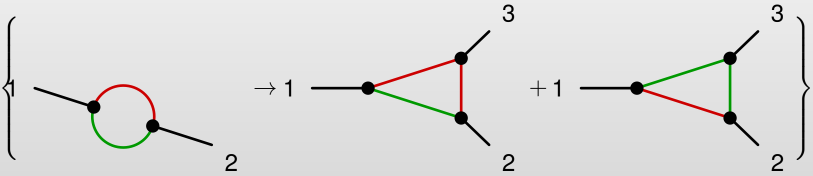


- simple tree level algorithm $n \to n+1$ doesn't work for loop diagrams because
  - diagrams will appear more than once



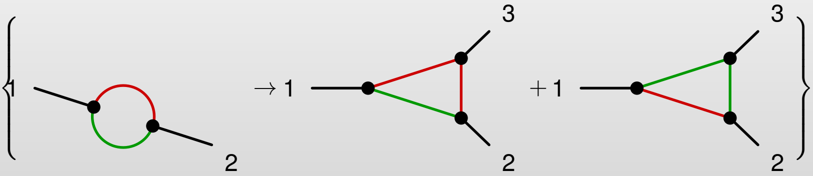
    - easy to spot in this simple example

- simple tree level algorithm $n \to n + 1$ doesn't work for loop diagrams because
  - diagrams will appear more than once



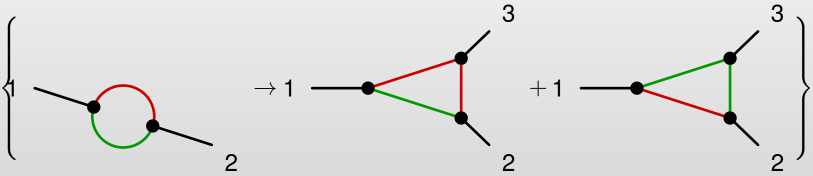
- easy to spot in this simple example
- nontrivial in the general case (e. g. ordered incidence matrices)

- simple tree level algorithm $n \to n+1$ doesn't work for loop diagrams because
  - diagrams will appear more than once



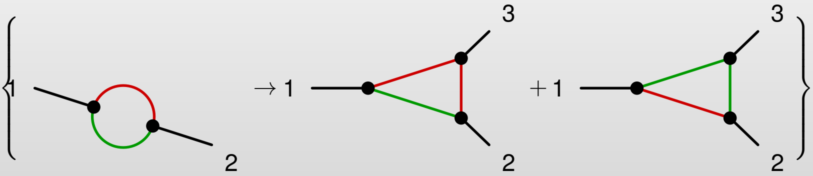
- easy to spot in this simple example
- nontrivial in the general case (e. g. ordered incidence matrices)
- NB: assignment of loop momenta not unique, can not be used to identify diagrams

- simple tree level algorithm $n \to n+1$ doesn't work for loop diagrams because
  - diagrams will appear more than once



  - easy to spot in this simple example
  - nontrivial in the general case (e. g. ordered incidence matrices)
  - NB: assignment of loop momenta not unique, can not be used to identify diagrams
- we need a sufficiently large set of starting topologies with n loops and without external legs (simple)

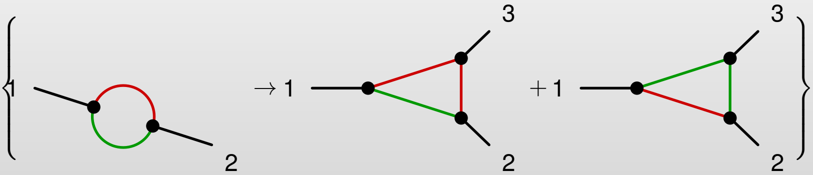
- simple tree level algorithm $n \to n+1$ doesn't work for loop diagrams because
  - diagrams will appear more than once



- easy to spot in this simple example
- nontrivial in the general case (e. g. ordered incidence matrices)
- NB: assignment of loop momenta not unique, can not be used to identify diagrams

- we need a sufficiently large set of starting topologies with n loops and without external legs (simple)
- most efficient, but with badly documented internals, public tool for multi loop diagrams QGRAF [Nogueira, 1991]

- simple tree level algorithm $n \to n+1$ doesn't work for loop diagrams because
  - diagrams will appear more than once



- easy to spot in this simple example
- nontrivial in the general case (e. g. ordered incidence matrices)
- NB: assignment of loop momenta not unique, can not be used to identify diagrams

- we need a sufficiently large set of starting topologies with n loops and without external legs (simple)
- most efficient, but with badly documented internals, public tool for multi loop diagrams `QGRAF` [Nogueira, 1991]
- alternative approach: loop graphs with cut lines as tree graphs in Recola or Openloops [Denner, Pozzorini]

The number of tree Feynman diagrams w/ n legs grows like a factorial, e. g. in ϕ^3-theory: $F(n) = (2n-5)!! = (2n-5)\cdot(2n-7)\cdot\ldots\cdot 3\cdot 1$

n		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		

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	$F(n)$	
4	3	
5	15	
6	105	
7	945	
8	10 395	
9	135 135	
10	2 027 025	
11	34 459 425	
12	654 729 075	
13	13 749 310 575	
14	316 234 143 225	
15	7 905 853 580 625	

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	$F(n)$
4	3
5	15
6	105
7	945
8	10 395
9	135 135
10	2 027 025
11	34 459 425
12	654 729 075
13	13 749 310 575
14	316 234 143 225
15	7 905 853 580 625

☹ computational costs grow beyond all reasonable limits

The number of tree Feynman diagrams w/ n legs grows like a factorial, e. g. in ϕ^3-theory: $F(n) = (2n-5)!! = (2n-5)\cdot(2n-7)\cdot\ldots\cdot 3\cdot 1$

n	$F(n)$
4	3
5	15
6	105
7	945
8	10 395
9	135 135
10	2 027 025
11	34 459 425
12	654 729 075
13	13 749 310 575
14	316 234 143 225
15	7 905 853 580 625

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

The number of tree Feynman diagrams w/ n legs grows like a factorial, e. g. in ϕ^3-theory: $F(n) = (2n-5)!! = (2n-5)\cdot(2n-7)\cdot\ldots\cdot 3\cdot 1$

n	$F(n)$
4	3
5	15
6	105
7	945
8	10 395
9	135 135
10	2 027 025
11	34 459 425
12	654 729 075
13	13 749 310 575
14	316 234 143 225
15	7 905 853 580 625

- ☹ 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 grows like a factorial, e. g. in ϕ^3-theory: $F(n) = (2n-5)!! = (2n-5)\cdot(2n-7)\cdot\ldots\cdot 3\cdot 1$

n	$F(n)$	$P(n)$
4	3	3
5	15	10
6	105	25
7	945	56
8	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

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

The number of tree Feynman diagrams w/ n legs grows like a factorial, e. g. in ϕ^3-theory: $F(n) = (2n-5)!! = (2n-5)\cdot(2n-7)\cdot\ldots\cdot 3\cdot 1$

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

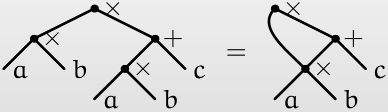
Number of possible momenta in tree diagrams grows only exponentially

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

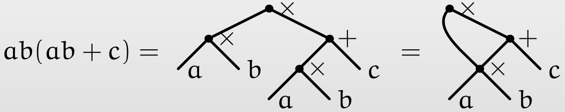
∴ Feynman diagrams redundant for many external particles!

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

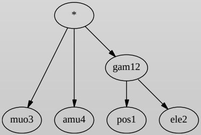
$$ab(ab+c) =$$



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

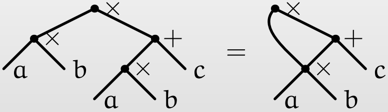


- simplest examples: $e^+e^- \to \mu^+\mu^-$, and
  (only QED)

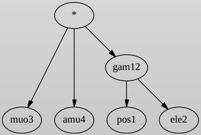


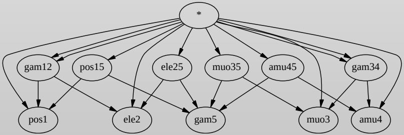
∴ Replace the forest of tree diagrams by the Directed Acyclical Graph (DAG) of the algebraic expression.

$ab(ab+c) =$



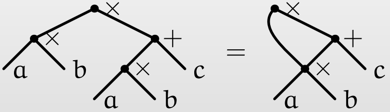
- simplest examples: $e^+e^- \to \mu^+\mu^-$, $e^+e^- \to \mu^+\mu^-\gamma$ and (only QED)



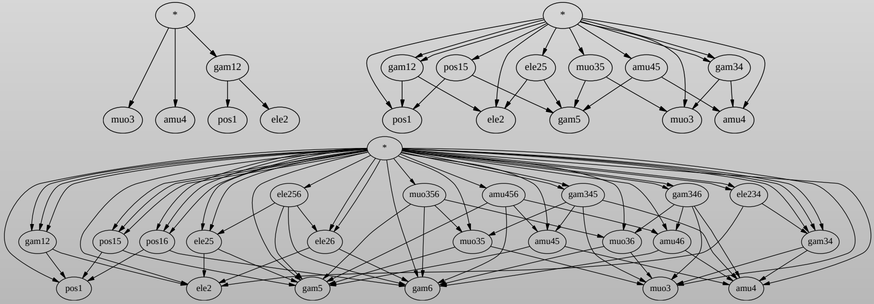


∴ Replace the forest of tree diagrams by the Directed Acyclical Graph (DAG) of the algebraic expression.

$$ab(ab+c) =$$



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



Efficient tree amplitudes

- Berends-Giele Recursion Relations [Berends, Giele]
  - manual calculations

Efficient tree amplitudes

- Berends-Giele Recursion Relations [Berends, Giele]
  - manual calculations
- HELAS [Hagiwara et al.],
  - manual partial common subexpression elimination

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
  - ∴ partial elimination of redundancy

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
  - ∴ 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

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
  - ∴ 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

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
  - ∴ 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

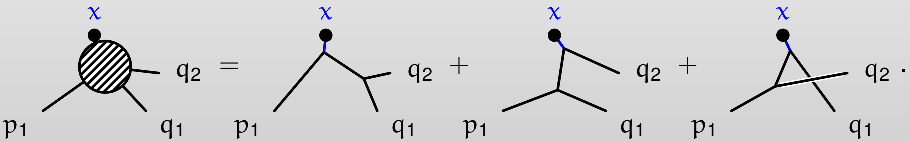
One particle off-shell wave functions (1POWs) are obtained from by applying the LSZ reduction formula to all but one line:

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

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

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

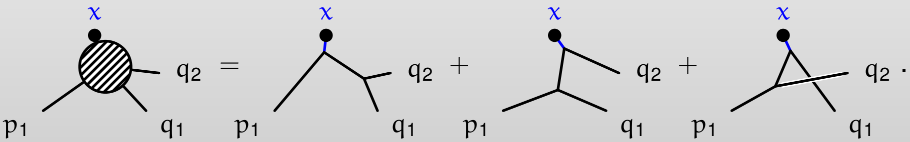
E. g. $\langle \phi(q_1), \phi(q_2); \text{out} | \Phi(x) | \phi(p_1); \text{in} \rangle$ in ϕ^3-theory at tree level



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

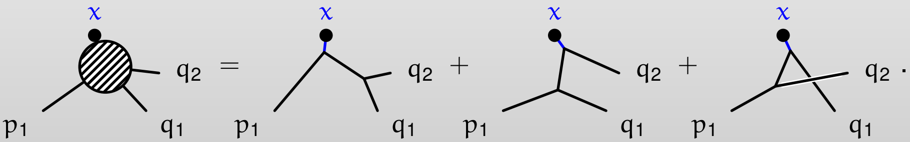


- the set of all 1POWs at tree level grows exponentially and can be constructed recursively from other 1POWs at tree level.

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

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

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



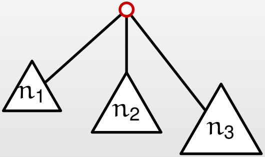
- the set of all 1POWs at tree level grows exponentially and can be constructed recursively from other 1POWs at tree level.

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

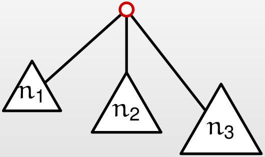
$$T = \sum_{i=1}^{F(n)} D_i = \sum_{k,l,m=1}^{P(n)} K^3_{f_k f_l f_m}(p_k,p_l,p_m) W_{f_k}(p_k) W_{f_l}(p_l) W_{f_m}(p_m)$$

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

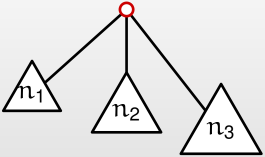


Non-trivial: construction of inequivalent topologies for merging off-shell amplitudes to Feynman diagrams.
Solution: inequivalent partitionings of external momenta for a fixed vertex



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	$1\cdot(1,1,1,1,1,1)+15\cdot(1,1,1,1,2)+40\cdot(1,1,1,3)$
		$+45\cdot(1,1,2,2)+120\cdot(1,2,3)+15\cdot(2,2,2)$

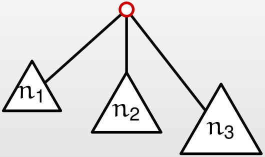
Non-trivial: construction of inequivalent topologies for merging off-shell amplitudes to Feynman diagrams.
Solution: inequivalent partitionings of external momenta for a fixed vertex



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
$F(d_{max}, n) =$ # 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).

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 $n = \mathcal{O}(100)$.

By virtue of they recursive construction, tree level 1POWs form a DAG

By virtue of they recursive construction, tree level 1POWs form a DAG:

∴ find the smallest DAG that corresponds to a given tree (i. e. a sum of Feynman diagrams)

By virtue of they recursive construction, tree level 1POWs form a DAG:

∴ find the smallest DAG that corresponds to a given tree (i. e. a sum of Feynman diagrams).

Systematic four step procedure:

Grow: starting from the external particles, build the tower of all 1POWs up to a given height (the height is always less than the number of external lines) and translate it to the equivalent DAG D.

By virtue of they recursive construction, tree level 1POWs form a DAG:

∴ find the smallest DAG that corresponds to a given tree (i. e. a sum of Feynman diagrams).

Systematic four step procedure:

Grow: starting from the external particles, build the tower of all 1POWs up to a given height (the height is always less than the number of external lines) and translate it to the equivalent DAG D.

Select: from D, determine all possible flavored keystones for the process under consideration and the 1POWs appearing in them.

By virtue of they recursive construction, tree level 1POWs form a DAG:

∴ find the smallest DAG that corresponds to a given tree (i. e. a sum of Feynman diagrams).

Systematic four step procedure:

Grow: starting from the external particles, build the tower of all 1POWs up to a given height (the height is always less than the number of external lines) and translate it to the equivalent DAG D.

Select: from D, determine all possible flavored keystones for the process under consideration and the 1POWs appearing in them.

Harvest: construct a sub-DAG $\mathrm{D}^* \subseteq \mathrm{D}$ consisting only of nodes that contribute to the 1POWs appearing in the flavored keystones.

By virtue of they recursive construction, tree level 1POWs form a DAG:

∴ find the smallest DAG that corresponds to a given tree (i. e. a sum of Feynman diagrams).

Systematic four step procedure:

Grow: starting from the external particles, build the tower of all 1POWs up to a given height (the height is always less than the number of external lines) and translate it to the equivalent DAG D.

Select: from D, determine all possible flavored keystones for the process under consideration and the 1POWs appearing in them.

Harvest: construct a sub-DAG $\mathrm{D}^* \subseteq \mathrm{D}$ consisting only of nodes that contribute to the 1POWs appearing in the flavored keystones.

Calculate: multiply the 1POWs as specified by the keystones and sum the keystones.

By virtue of they recursive construction, tree level 1POWs form a DAG:

∴ find the smallest DAG that corresponds to a given tree (i. e. a sum of Feynman diagrams).

Systematic four step procedure:

Grow: starting from the external particles, build the tower of all 1POWs up to a given height (the height is always less than the number of external lines) and translate it to the equivalent DAG D.

Select: from D, determine all possible flavored keystones for the process under consideration and the 1POWs appearing in them.

Harvest: construct a sub-DAG $\mathrm{D}^* \subseteq \mathrm{D}$ consisting only of nodes that contribute to the 1POWs appearing in the flavored keystones.

Calculate: multiply the 1POWs as specified by the keystones and sum the keystones.

☺ the resulting expression contains no more redundancies!

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
```

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

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

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

```
$ omega_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-
  p12 = p1 + p2
  p13 = p1 + p3
```

```
    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_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))
    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

```
  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_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))
  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 or NLO vertex functions

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

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

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

```
$ omega_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

Remaining problem: phase space integration and simulation

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

Remaining problem: phase space integration and simulation

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

1. non-trivial geometry of multi particle phase space

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

Remaining problem: phase space integration and simulation

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

1. non-trivial geometry of multi particle phase space

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

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

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

Remaining problem: phase space integration and simulation

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

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

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

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 $\mathfrak{p}$.

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 $\mathfrak{p}$.

Conflicting goals for g

1. make $d\mu_g(p)$ simple enough, so that $\mathfrak{p}_g$ can be generated w/ reasonable effort

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 $\mathfrak{p}$.
Conflicting goals for g

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

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:

∵ only a small number of different manifolds $\mathcal{M}$:

- number of particles 2, 3, 4, 5, 6, 7, ...
- massless vs. massive particles

Practical considerations for particle physics:

∵ only a small number of different manifolds $\mathcal{M}$:
- number of particles 2, 3, 4, 5, 6, 7, ...
- massless vs. massive particles

∴ it makes sense to invest manpower into an optimal treatment of the geometry, i. e. $\mathrm{d}\mu$

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

Practical considerations for particle physics:

- $\because$ only a small number of different manifolds $\mathcal{M}$:
  - 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

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 $d\mu_g(x) = g(x)d\mu(x)$?

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 $d\mu_g(x) = g(x)d\mu(x)$?

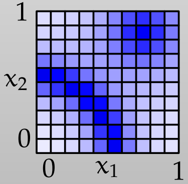
- optimization of expansion coefficients α in $g(x) = \sum_l \alpha_l g_l(x)$ popular, but not exciting for generator generation
  - ∵ selection of g_l requires expert human input
  - ∵ can't deal very efficiently w/ cuts

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 $d\mu_g(x) = g(x)d\mu(x)$?

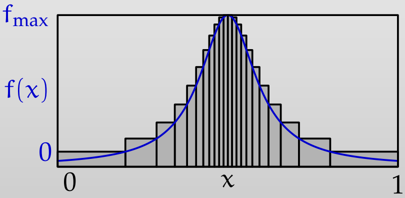
- optimization of expansion coefficients α in $g(x) = \sum_l \alpha_l g_l(x)$ popular, but not exciting for generator generation
  - ∵ selection of g_l requires expert human input
  - ∵ can't deal very efficiently w/ cuts
- fixed grid w/ variable weights



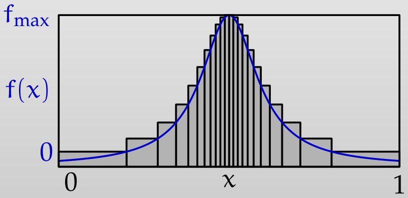
x (i. e. characteristic functions as g_l) not useful at all

- ∵ 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:



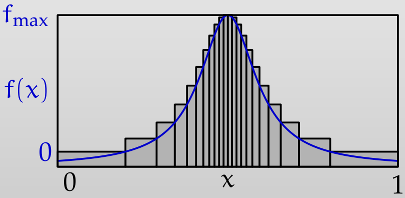
- alternative in one dimension: instead of adjusting weights of fixed bins, adjust density of equal weight bins
  - ∵ globally fixed resolution can nevertheless adjust locally over many orders of magnitude:



iterativeley adjust grid, use estimates to either

- approximate f locally (importance sampling $\Longrightarrow$ event generation)

- 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)

Factorized ansatz

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

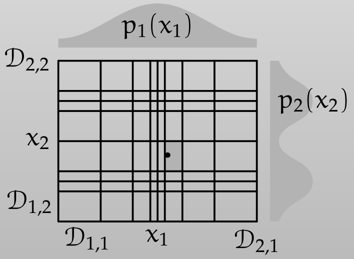
- guarantees hypercubic properties and simple one-dimensional formulae (w/ averaging over other dimensions)
- computational costs grow only linearly w/ the number of dimensions

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:

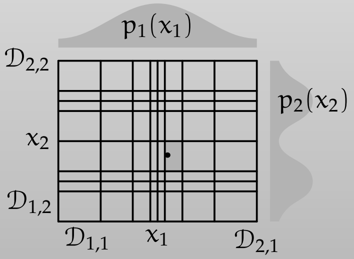


Factorized ansatz

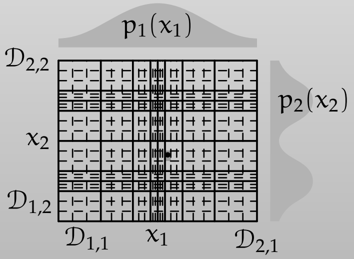
$$g(x) = g_1(x_1)g_2(x_2)\cdots g_n(x_n)$$

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

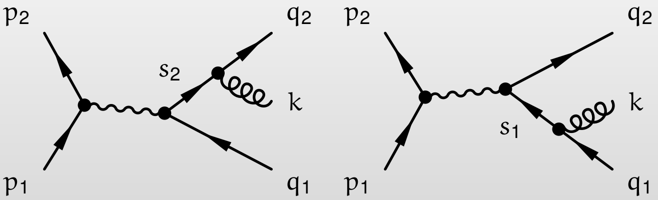
VEGAS grid structure for importance sampling:



for genuinely stratified sampling, used in low dimensions:



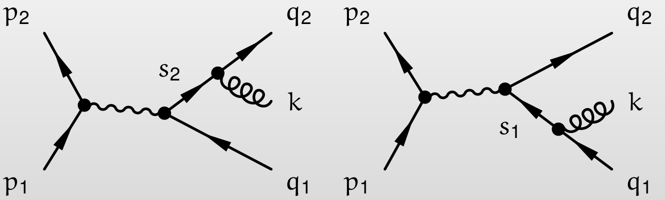
- typically singularities in more than one variable, e. g. $e^+e^- \to q\bar{q}g$



$$f \propto c_1 \frac{1}{s_1 - m_q^2} + c_2 \frac{1}{s_2 - m_q^2}$$

with $s_{1/2} = (q_{1/2} + k)^2$.

- typically singularities in more than one variable, e. g. $e^+e^- \to q\bar{q}g$

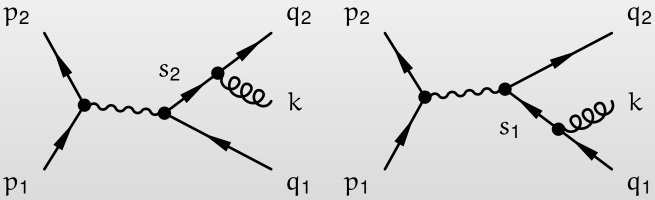


$$f \propto c_1 \frac{1}{s_1 - m_q^2} + c_2 \frac{1}{s_2 - m_q^2}$$

with $s_{1/2} = (q_{1/2} + k)^2$.

- parametrizations with either s_1 or s_2 as parameters easy to find

- typically singularities in more than one variable, e. g. $e^+e^- \to q\bar{q}g$



$$f \propto c_1 \frac{1}{s_1 - m_q^2} + c_2 \frac{1}{s_2 - m_q^2}$$

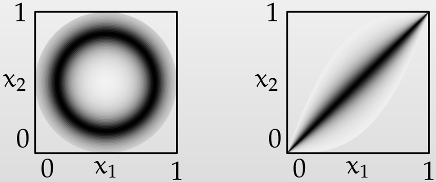
with $s_{1/2} = (q_{1/2} + k)^2$.

- parametrizations with either s_1 or s_2 as parameters easy to find
- ☹ parametrizations with both s_1 and s_2 as parameters contain the Gram determinant as

$$\frac{1}{\sqrt{\Delta_4(p_1, p_2, q_1, q_2)}}$$

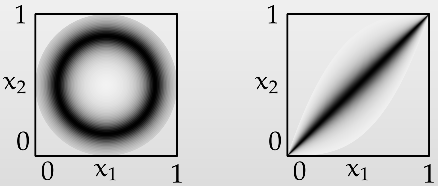
which contributes a non-factorizable singularity itself!

- VEGAS' factorized ansatz handles



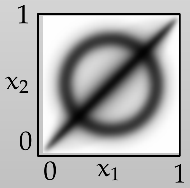
separately after mappings very well.

- VEGAS' factorized ansatz handles



separately after mappings very well.

- ☹ fails for overlapping singularities



which is the common case for more than one diagram

- closer look:

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

  involves two steps:

- closer look:

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

involves two steps:

  - sampling the function f

- closer look:

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

involves two steps:

  - sampling the function f
  - parameterizing M (or $d\mu$)

- closer look:

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

  involves two steps:
  - sampling the function f
  - parameterizing M (or $d\mu$)
- in general, we need more than one chart $\phi : \mathbf{R}^n \to M$ to cover all of M

- closer look:

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

involves two steps:

  - sampling the function f
  - parameterizing M (or $d\mu$)
- in general, we need more than one chart $\phi : \mathbf{R}^n \to M$ to cover all of M
- ☺ global issues rarely a problem for us (sets of measure 0 can be ignored for integration)

- closer look:

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

  involves two steps:
  - sampling the function f
  - parameterizing M (or $d\mu$)
- in general, we need more than one chart $\phi : \mathbf{R}^n \to M$ to cover all of M
  - ☺ global issues rarely a problem for us (sets of measure 0 can be ignored for integration)
  - ☺ differential geometry still offers useful intuition:

- closer look:

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

  involves two steps:
  - sampling the function f
  - parameterizing M (or $d\mu$)
- in general, we need more than one chart $\phi : \mathbf{R}^n \to M$ to cover all of M
  - ☺ global issues rarely a problem for us (sets of measure 0 can be ignored for integration)
  - ☺ differential geometry still offers useful intuition:
- strategy:

  Instead of pasting together locally flat pieces, we can add factorizable pieces

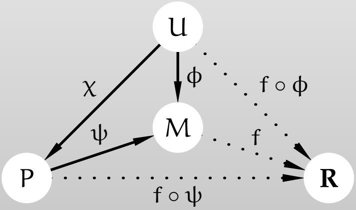
- For convenience, use the same domain for all charts, the open n-dimensional unit hypercube

$$U =]0, 1[^{\otimes n}$$

- For convenience, use the same domain for all charts, the open n-dimensional unit hypercube

$$U =]0,1[^{\otimes n}$$

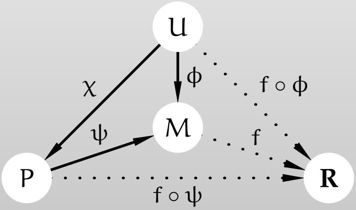
- introduce flat intermediate space $P \in \mathbf{R}^n$ of physical parameters (often dimensionful) and view charts as compositions $\phi = \psi \circ \chi$:



- For convenience, use the same domain for all charts, the open n-dimensional unit hypercube

$$\mathcal{U} =]0,1[^{\otimes n}$$

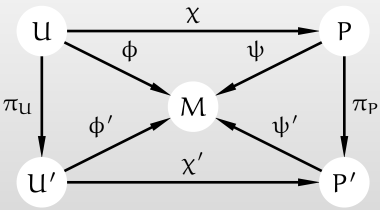
- introduce flat intermediate space $P \in \mathbf{R}^n$ of physical parameters (often dimensionful) and view charts as compositions $\phi = \psi \circ \chi$:



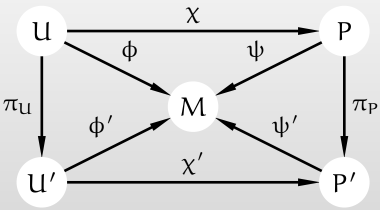
∴ it remains to sample $|\partial\phi/\partial x| \cdot (f \circ \phi)$ on $\mathcal{U}$:

$$I(f) = \int_0^1 d^n x \left|\frac{\partial \phi}{\partial x}\right| f(\phi(x))$$

- There is more than one way to map U onto M

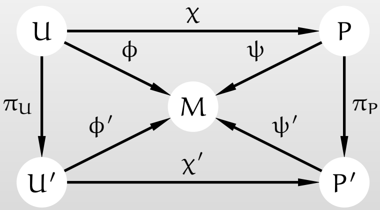


- There is more than one way to map U onto M



☺ we are free to select the most suitable map

- There is more than one way to map U onto M

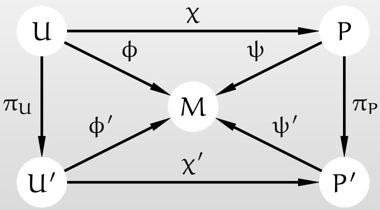


☺ we are free to select the most suitable map

- ideal choice solves partial differential equation

$$|\partial\phi/\partial x| = 1/(f \circ \phi)$$

- There is more than one way to map U onto M



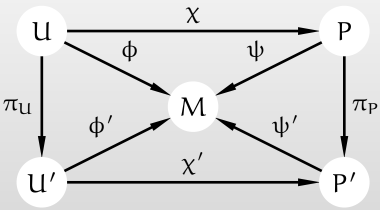
☺ we are free to select the most suitable map

- ideal choice solves partial differential equation

$$|\partial\phi/\partial x| = 1/(f \circ \phi)$$

☹ not practical, since equivalent to analytical evaluation

- There is more than one way to map U onto M



- we are free to select the most suitable map
  - ideal choice solves partial differential equation

$$|\partial\phi/\partial x| = 1/(f \circ \phi)$$

- not practical, since equivalent to analytical evaluation
  - more realistic: require

$$|\partial\phi/\partial x| \cdot (f \circ \phi)$$

to have factorizable singularities, which can be sampled well by VEGAS.

- Consider N_c different charts $\phi_i : U \to M$ and corresponding probability densities $g_i : U \to [0, \infty)$.

- Consider N_c different charts $\phi_i : U \to M$ and corresponding probability densities $g_i : U \to [0, \infty)$.

∴ the function

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

is a probability density $g : M \to [0, \infty)$

$$\int_M d\mu(p)\, g(p) = 1\,,$$

as long as the g_i and α_i are properly normalized

$$\int_0^1 g_i(x) d^n x = 1\,, \quad \sum_{i=1}^{N_c} \alpha_i = 1\,, \quad 0 \leqslant \alpha_i \leqslant 1\,.$$

- Consider N_c different charts $\phi_i : U \to M$ and corresponding probability densities $g_i : U \to [0, \infty)$.

∴ the function

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

is a probability density $g : M \to [0, \infty)$

$$\int_M d\mu(p)\, g(p) = 1\,,$$

as long as the g_i and α_i are properly normalized

$$\int_0^1 g_i(x) d^n x = 1\,, \quad \sum_{i=1}^{N_c} \alpha_i = 1\,, \quad 0 \leqslant \alpha_i \leqslant 1\,.$$

- obviously

$$I(f) = \sum_{i=1}^{N_c} \alpha_i \int_M g_i(\phi_i^{-1}(p)) \left| \frac{\partial \phi_i^{-1}}{\partial p} \right| d\mu(p)\, \frac{f(p)}{g(p)}$$

- pulling back from M to U

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

- pulling back from M to U

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

- corresponding estimate

$$E(f) = \sum_{i=1}^{N_c} \alpha_i \left\langle \frac{f \circ \phi_i}{g \circ \phi_i} \right\rangle_{g_i} \, .$$

- pulling back from M to U

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

- corresponding estimate

$$E(f) = \sum_{i=1}^{N_c} \alpha_i \left\langle \frac{f \circ \phi_i}{g \circ \phi_i} \right\rangle_{g_i} \,.$$

☺ even with factorized g_i, their sum g can approximate nonfactorizable singularities

- pulling back from M to U

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

- corresponding estimate

$$E(f) = \sum_{i=1}^{N_c} \alpha_i \left\langle \frac{f \circ \phi_i}{g \circ \phi_i} \right\rangle_{g_i} \, .$$

- even with factorized g_i, their sum g can approximate nonfactorizable singularities
- geometrically, the maps $\pi_{ij} = \phi_j^{-1} \circ \phi_i : U \to U$ are just the coordinate transformations between the coordinate systems in which the singularities factorize

- Remaining problem: find α_i

- Remaining problem: find α_i
- relative weight of singularities depend on parameters

- Remaining problem: find α_i
- relative weight of singularities depend on parameters

∴ manual tuning not suitable, use numerical procedure

- Remaining problem: find α_i
- relative weight of singularities depend on parameters

∴ manual tuning not suitable, use numerical procedure

- goal: minimize the variance V, i. e. minimize

$$W(f,\alpha) = \int_M g(p) d\mu(p) \left(\frac{f(p)}{g(p)}\right)^2$$

w. r. t. the α_i under the condition $\sum_i \alpha_i = 1$.

- Remaining problem: find α_i
- relative weight of singularities depend on parameters

∴ manual tuning not suitable, use numerical procedure

- goal: minimize the variance V, i. e. minimize

$$W(f,\alpha) = \int_M g(p) d\mu(p) \left(\frac{f(p)}{g(p)}\right)^2$$

w. r. t. the α_i under the condition $\sum_i \alpha_i = 1$.

∴ with a Lagrange multiplier the stationary points are given by

$$\forall i : W_i(f,\alpha) = W(f,\alpha)$$

where

$$W_i(f,\alpha) = -\frac{\partial}{\partial \alpha_i} W(f,\alpha) = \int_0^1 g_i(x) d^n x \left(\frac{f(\phi_i(x))}{g(\phi_i(x))}\right)^2$$

and

$$W(f,\alpha) = \sum_{i=1}^{N_c} \alpha_i W_i(f,\alpha).$$

- Use
$$N_i = \alpha_i N$$
to distribute N sampling points among channels. Then $W_i(f,\alpha)$ is the contribution from channel i to the total variance.

- Use
$$N_i = \alpha_i N$$
to distribute N sampling points among channels. Then $W_i(f, \alpha)$ is the contribution from channel i to the total variance.

∴ Familiar result from stratified sampling: the overall variance is minimized by spreading the variance evenly.

- Use

$$N_i = \alpha_i N$$

to distribute N sampling points among channels. Then $W_i(f,\alpha)$ is the contribution from channel i to the total variance.

∴ Familiar result from stratified sampling: the overall variance is minimized by spreading the variance evenly.

- Estimate $W_i(f,\alpha)$ while sampling $I(f)$

$$V_i(f,\alpha) = \left\langle \left(\frac{f \circ \phi_i}{g \circ \phi_i} \right)^2 \right\rangle_{g_i} .$$

and update the α_i according to

$$\alpha_i \mapsto \alpha_i' = \frac{\alpha_i \left(V_i(f,\alpha)\right)^\beta}{\sum_i \alpha_i \left(V_i(f,\alpha)\right)^\beta}, \quad (\beta > 0)$$

which has the minimal variance as a fixed point.

Recapitulate:

- in general, $g \circ \phi_i$ does not factorize, even if all g_i factorize.

Recapitulate:

- 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.

Recapitulate:

- 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

Recapitulate:

- 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
  - ∵ π_{ij} universal and are calculated automatically by WHIZARD
  - ∴ VEGAS can optimize the g_i for each set of parameters and cuts

Recapitulate:

- 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

Recapitulate:

- 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
  - ∵ π_{ij} universal and are calculated automatically by WHIZARD
  - ∴ VEGAS can optimize the g_i for each set of parameters and cuts

However:

∵ singularity structure determined by Feynman diagrams

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

Recapitulate:

- 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
  - ∵ π_{ij} universal and are calculated automatically by WHIZARD
  - ∴ VEGAS can optimize the g_i for each set of parameters and cuts

However:

∵ singularity structure determined by Feynman diagrams

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

∴ WHIZARD uses heuristics to select the important channels

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

Recapitulate:

- 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
  - ∵ π_{ij} universal and are calculated automatically by WHIZARD
  - ∴ VEGAS can optimize the g_i for each set of parameters and cuts

However:

- ∵ singularity structure determined by Feynman diagrams
- naive application brings the combinatorial explosion in through the back door!
- ∴ WHIZARD uses heuristics to select the important channels
  - s-channel resonances
  - $1/t$-poles for massless particles

  and permutation symmetries to eliminate equivalent channels

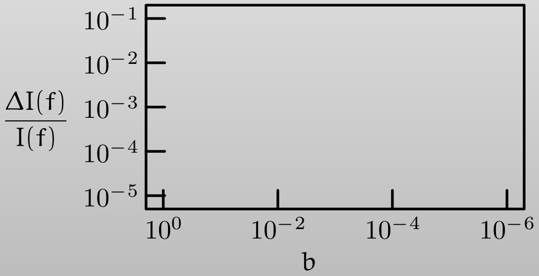
$$f(x) = \frac{b}{144\,\mathrm{atan}(1/2b)} \left(\frac{3\pi\Theta(r_3 < 1)}{r_3^2((r_3 - 1/2)^2 + b^2)} + \frac{2\pi\Theta(r_2 < 1, |x_3| < 1)}{r_2((r_2 - 1/2)^2 + b^2)} + \frac{\Theta(-1 < x_1, x_2, x_3 < 1)}{x_1^2 + b^2} \right)$$

$$f(x) = \frac{b}{144\,\mathrm{atan}(1/2b)} \left(\frac{3\pi\Theta(r_3 < 1)}{r_3^2((r_3 - 1/2)^2 + b^2)} + \frac{2\pi\Theta(r_2 < 1, |x_3| < 1)}{r_2((r_2 - 1/2)^2 + b^2)} + \frac{\Theta(-1 < x_1, x_2, x_3 < 1)}{x_1^2 + b^2} \right)$$

Sampling error for the integral as a function of the width b for comparable computational costs:

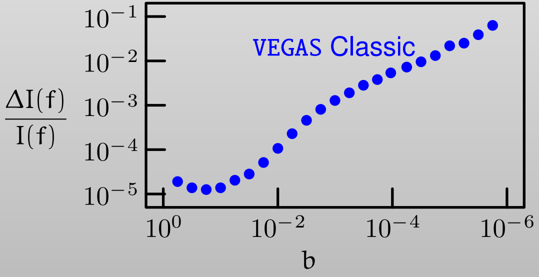
$$f(x) = \frac{b}{144\,\mathrm{atan}(1/2b)} \left(\frac{3\pi\Theta(r_3 < 1)}{r_3^2((r_3 - 1/2)^2 + b^2)} + \frac{2\pi\Theta(r_2 < 1, |x_3| < 1)}{r_2((r_2 - 1/2)^2 + b^2)} + \frac{\Theta(-1 < x_1, x_2, x_3 < 1)}{x_1^2 + b^2} \right)$$

Sampling error for the integral as a function of the width b for comparable computational costs:



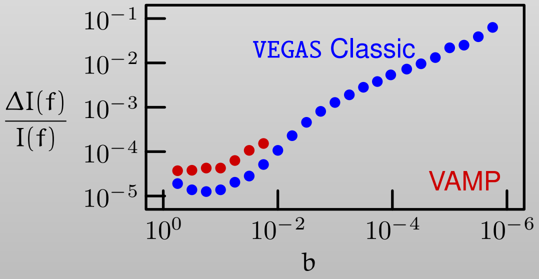
$$f(x) = \frac{b}{144\,\mathrm{atan}(1/2b)} \left(\frac{3\pi\Theta(r_3 < 1)}{r_3^2((r_3 - 1/2)^2 + b^2)} + \frac{2\pi\Theta(r_2 < 1, |x_3| < 1)}{r_2((r_2 - 1/2)^2 + b^2)} + \frac{\Theta(-1 < x_1, x_2, x_3 < 1)}{x_1^2 + b^2} \right)$$

Sampling error for the integral as a function of the width b for comparable computational costs:



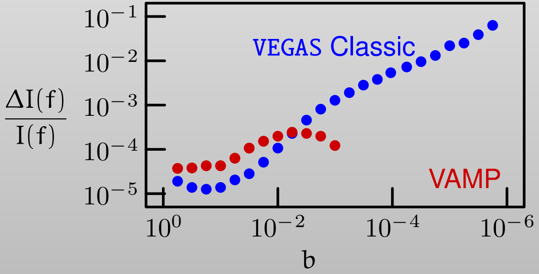
$$f(x) = \frac{b}{144\,\text{atan}(1/2b)}\left(\frac{3\pi\Theta(r_3<1)}{r_3^2((r_3-1/2)^2+b^2)} + \frac{2\pi\Theta(r_2<1,|x_3|<1)}{r_2((r_2-1/2)^2+b^2)} + \frac{\Theta(-1<x_1,x_2,x_3<1)}{x_1^2+b^2}\right)$$

Sampling error for the integral as a function of the width b for comparable computational costs:



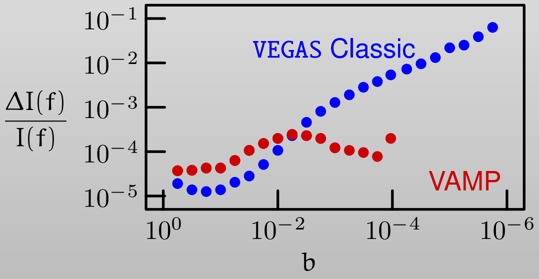
$$f(x) = \frac{b}{144\,\mathrm{atan}(1/2b)} \left(\frac{3\pi\Theta(r_3 < 1)}{r_3^2((r_3 - 1/2)^2 + b^2)} + \frac{2\pi\Theta(r_2 < 1, |x_3| < 1)}{r_2((r_2 - 1/2)^2 + b^2)} + \frac{\Theta(-1 < x_1, x_2, x_3 < 1)}{x_1^2 + b^2} \right)$$

Sampling error for the integral as a function of the width b for comparable computational costs:



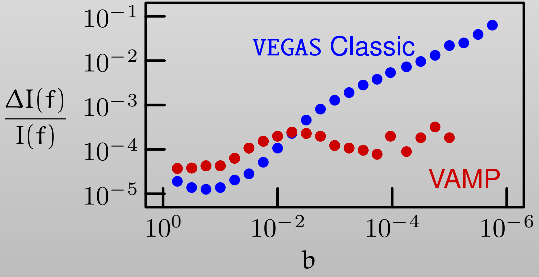
$$f(x) = \frac{b}{144\,\mathrm{atan}(1/2b)}\left(\frac{3\pi\Theta(r_3<1)}{r_3^2((r_3-1/2)^2+b^2)} + \frac{2\pi\Theta(r_2<1,|x_3|<1)}{r_2((r_2-1/2)^2+b^2)} + \frac{\Theta(-1<x_1,x_2,x_3<1)}{x_1^2+b^2}\right)$$

Sampling error for the integral as a function of the width b for comparable computational costs:



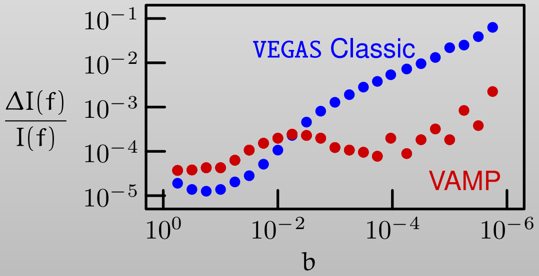
$$f(x) = \frac{b}{144\,\mathrm{atan}(1/2b)} \left(\frac{3\pi\Theta(r_3 < 1)}{r_3^2((r_3 - 1/2)^2 + b^2)} + \frac{2\pi\Theta(r_2 < 1, |x_3| < 1)}{r_2((r_2 - 1/2)^2 + b^2)} + \frac{\Theta(-1 < x_1, x_2, x_3 < 1)}{x_1^2 + b^2} \right)$$

Sampling error for the integral as a function of the width b for comparable computational costs:



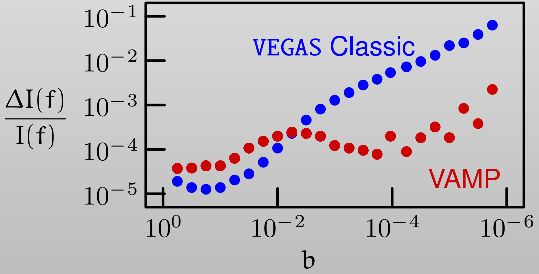
$$f(x) = \frac{b}{144\,\mathrm{atan}(1/2b)} \left(\frac{3\pi\Theta(r_3 < 1)}{r_3^2((r_3 - 1/2)^2 + b^2)} + \frac{2\pi\Theta(r_2 < 1, |x_3| < 1)}{r_2((r_2 - 1/2)^2 + b^2)} + \frac{\Theta(-1 < x_1, x_2, x_3 < 1)}{x_1^2 + b^2} \right)$$

Sampling error for the integral as a function of the width b for comparable computational costs:



$$f(x) = \frac{b}{144\,\mathrm{atan}(1/2b)} \left(\frac{3\pi\Theta(r_3 < 1)}{r_3^2((r_3 - 1/2)^2 + b^2)} + \frac{2\pi\Theta(r_2 < 1, |x_3| < 1)}{r_2((r_2 - 1/2)^2 + b^2)} + \frac{\Theta(-1 < x_1, x_2, x_3 < 1)}{x_1^2 + b^2} \right)$$

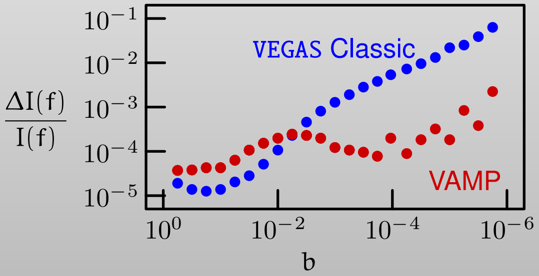
Sampling error for the integral as a function of the width b for comparable computational costs:



- up to two orders of magnitude in precision

$$f(x) = \frac{b}{144\,\mathrm{atan}(1/2b)}\left(\frac{3\pi\Theta(r_3<1)}{r_3^2((r_3-1/2)^2+b^2)} + \frac{2\pi\Theta(r_2<1,|x_3|<1)}{r_2((r_2-1/2)^2+b^2)} + \frac{\Theta(-1<x_1,x_2,x_3<1)}{x_1^2+b^2}\right)$$

Sampling error for the integral as a function of the width b for comparable computational costs:



- up to two orders of magnitude in precision (four orders of magnitude in speed)

WHIZARD Core
Version 2.x

$|\text{in}\rangle$, $|\text{out}\rangle$, cuts

parameters: $e, \alpha_{\text{QCD}}, m_W, \ldots$

physics model: $\mathcal{L}$, Feynman rules

WHIZARD Core *Version 2.x*

$|\text{in}\rangle, |\text{out}\rangle$, cuts

parameters:
$e, \alpha_{\text{QCD}}, m_W, \ldots$

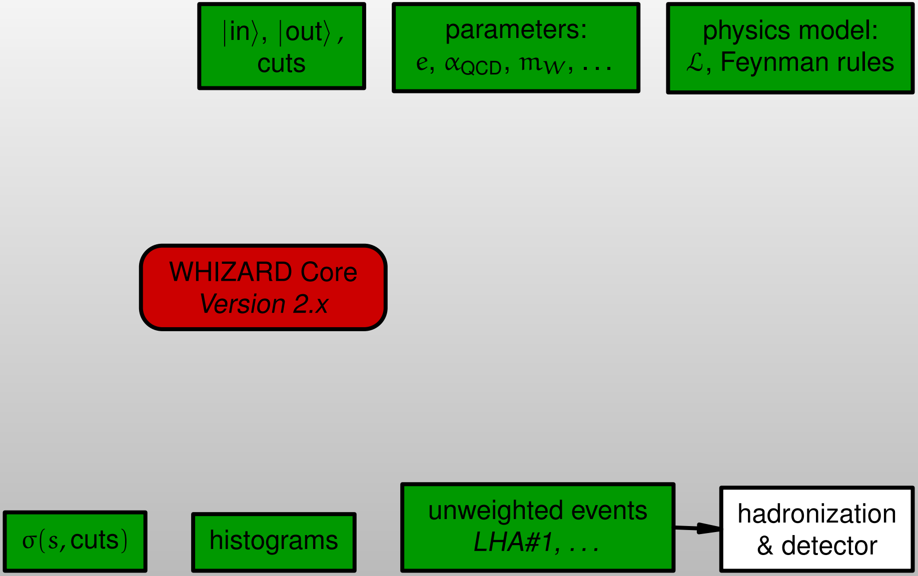
physics model:
$\mathcal{L}$, Feynman rules

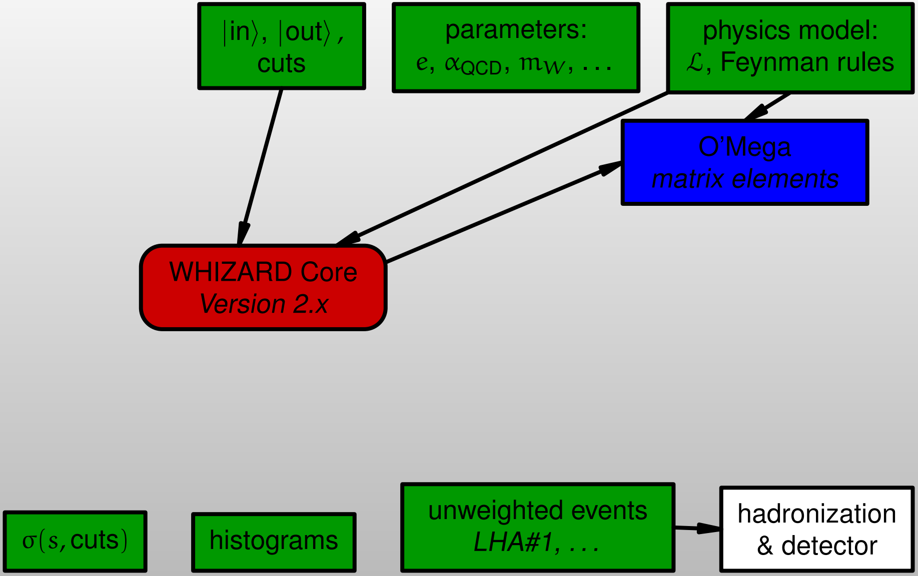
WHIZARD Core
Version 2.x

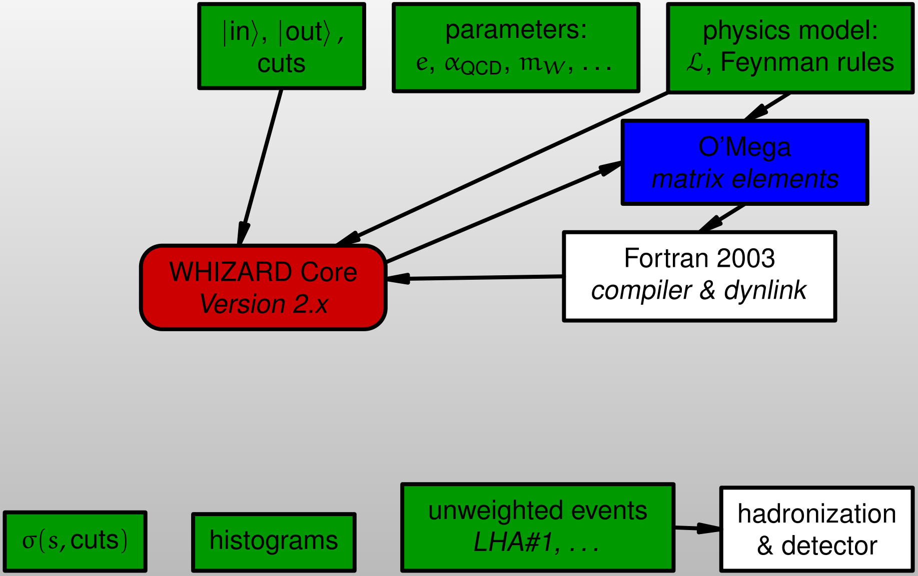
$\sigma(s, \text{cuts})$

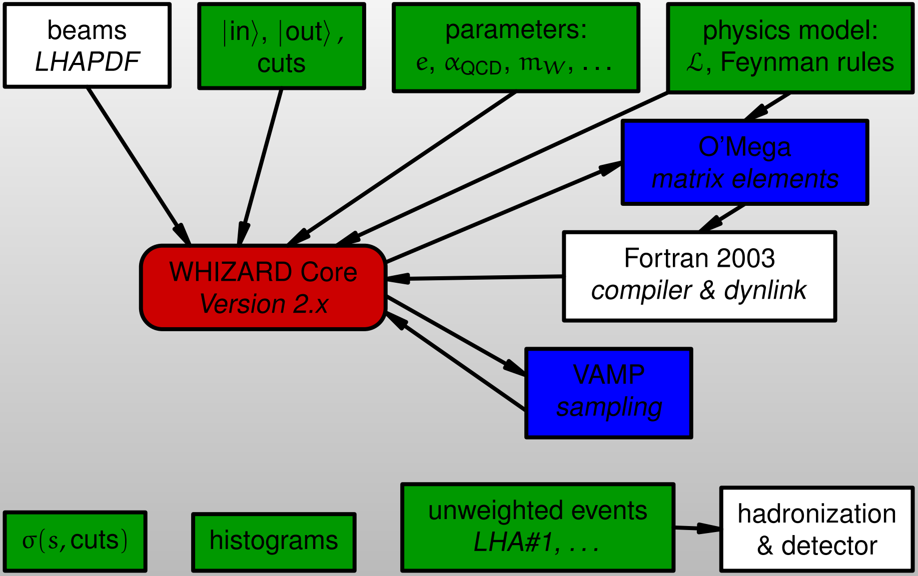
histograms

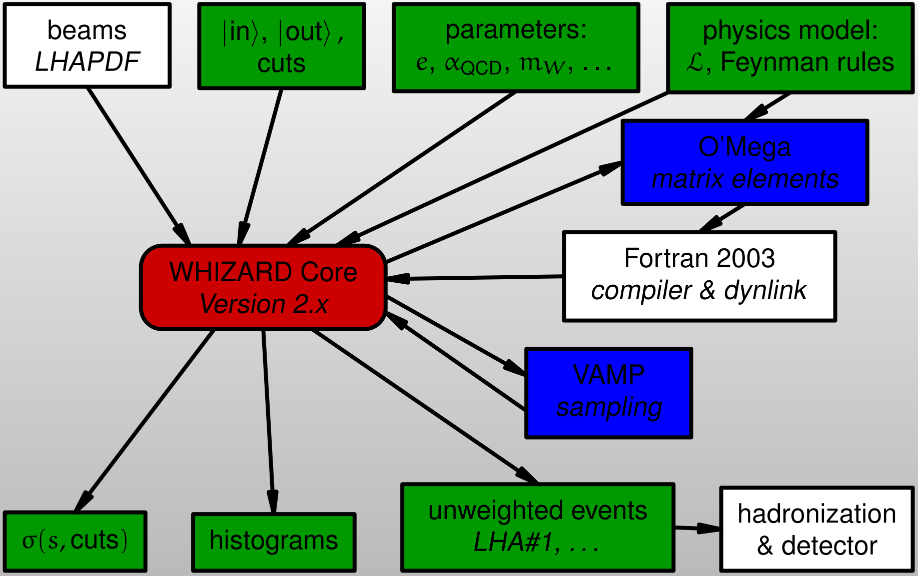
unweighted events
LHA#1, ...


|in⟩, |out⟩, cuts
parameters: e, α_QCD, m_W, . . .
physics model: ℒ, Feynman rules
WHIZARD Core
Version 2.x
σ(s, cuts)
histograms
unweighted events
LHA#1, . . .
hadronization
& detector


|in⟩, |out⟩, cuts
parameters: e, α_QCD, m_W, ...
physics model: ℒ, Feynman rules
O'Mega
matrix elements
WHIZARD Core
Version 2.x
σ(s, cuts)
histograms
unweighted events
LHA#1, ...
hadronization
& detector


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


beams
LHAPDF
|in⟩, |out⟩,
cuts
parameters:
$e, \alpha_{QCD}, m_W, \ldots$
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


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

- available models
  - QED
  - QCD
  - Standard Model

- available models
  - QED
  - QCD
  - Standard Model
  - SM with anomalous top and gauge couplings
  - Supersymmetry: MSSM (cross checked with MadGraph and Sherpa), NMSSM, PSSSM
  - 3-Site Higgsless Model
  - Z'
  - Extra Dimensions, UED
  - Little Higgs: Littlest, Simplest

- available models
  - QED
  - QCD
  - Standard Model
  - SM with anomalous top and gauge couplings
  - Supersymmetry: MSSM (cross checked with MadGraph and Sherpa), NMSSM, PSSSM
  - 3-Site Higgsless Model
  - Z'
  - Extra Dimensions, UED
  - Little Higgs: Littlest, Simplest
  - all FeynRules models

- available models
  - QED
  - QCD
  - Standard Model
  - SM with anomalous top and gauge couplings
  - Supersymmetry: MSSM (cross checked with MadGraph and Sherpa), NMSSM, PSSSM
  - 3-Site Higgsless Model
  - Z'
  - Extra Dimensions, UED
  - Little Higgs: Littlest, Simplest
  - all FeynRules models
  - your own . . .

- available models
  - QED
  - QCD
  - Standard Model
  - SM with anomalous top and gauge couplings
  - Supersymmetry: MSSM (cross checked with MadGraph and Sherpa), NMSSM, PSSSM
  - 3-Site Higgsless Model
  - Z'
  - Extra Dimensions, UED
  - Little Higgs: Littlest, Simplest
  - all FeynRules models
  - your own . . .
- get it from

`http://projects.hepforge.org/whizard/`

Simulate the W^- endpoint distribution

- select the Standard Model

```
model = SM
```

Simulate the W^- endpoint distribution

- select the Standard Model

```
model = SM
```

- set up the parton level processes $q\bar{q} \to \ell\nu_\ell j$

```
alias parton = u:U:d:D:g
alias jet = parton
alias lepton = e1:e2
alias neutrino = n1:N1:n2:N2
process enj = parton, parton => lepton, neutrino, jet
```

Simulate the W^- endpoint distribution

- select the Standard Model

```
model = SM
```

- set up the parton level processes $q\bar{q} \to \ell\nu_\ell j$

```
alias parton = u:U:d:D:g
alias jet = parton
alias lepton = e1:e2
alias neutrino = n1:N1:n2:N2
process enj = parton, parton => lepton, neutrino, jet
```

- call O'Mega, the Fortran compiler and the dynamic linker:

```
compile
```

Simulate the W^- endpoint distribution

- select the Standard Model

```
model = SM
```

- set up the parton level processes $q\bar{q} \to \ell\nu_\ell j$

```
alias parton = u:U:d:D:g
alias jet = parton
alias lepton = e1:e2
alias neutrino = n1:N1:n2:N2
process enj = parton, parton => lepton, neutrino, jet
```

- call O'Mega, the Fortran compiler and the dynamic linker:

```
compile
```

- choose the LHC design energy

```
sqrts = 14 TeV
beams = p, p => lhapdf { $lhapdf_file = "cteq5l.LHgrid" }
```

- define reasonable phase space cuts

```
cuts = all Pt >= 10 GeV [jet:lepton]
```

- define reasonable phase space cuts

```
cuts = all Pt >= 10 GeV [jet:lepton]
```

- integrate the cross section in order to initialize the phase space grids for simulation

```
integrate (enj) { iterations = 5:20000 }
```

- define reasonable phase space cuts

```
cuts = all Pt >= 10 GeV [jet:lepton]
```

- integrate the cross section in order to initialize the phase space grids for simulation

```
integrate (enj) { iterations = 5:20000 }
```

- allocate plots

```
$title = "$W$ Endpoint in $pp\to \ell\bar\nu j$"
$ylabel = "$N_{\textrm{events}}$"
$xlabel = "$p_T^\ell$/GeV"
histogram pt_lepton (0 GeV, 80 GeV, 2 GeV)
analysis =
  record pt_lepton
    (eval Pt [extract index 1 [sort by Pt [lepton]]])
```

- define reasonable phase space cuts

```
cuts = all Pt >= 10 GeV [jet:lepton]
```

- integrate the cross section in order to initialize the phase space grids for simulation

```
integrate (enj) { iterations = 5:20000 }
```

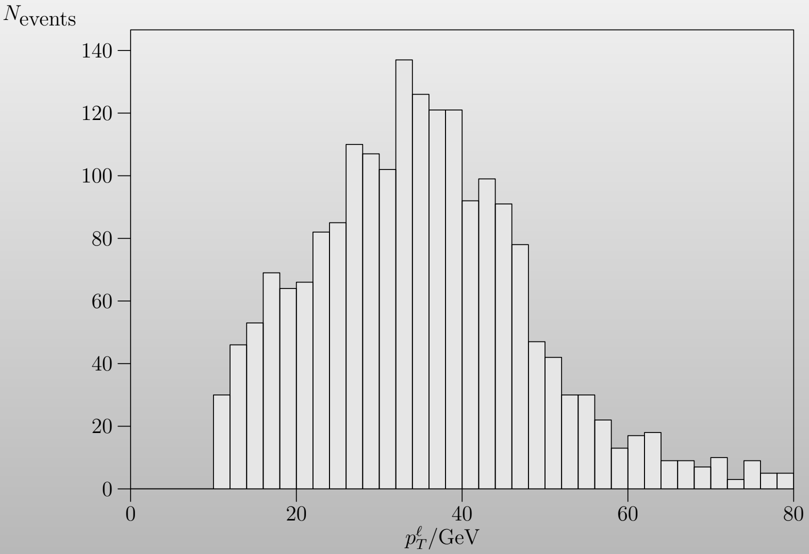
- allocate plots

```
$title = "$W$ Endpoint in $pp\to \ell\bar\nu j$"
$ylabel = "$N_{\textrm{events}}$"
$xlabel = "$p_T^\ell$/GeV"
histogram pt_lepton (0 GeV, 80 GeV, 2 GeV)
analysis =
  record pt_lepton
    (eval Pt [extract index 1 [sort by Pt [lepton]]])
```

- generate 1000 events and write the results

```
simulate (enj) { n_events = 1000 }
write_analysis
```

- Resulting plot



- efficient and complete implementations of the Fortran 2003 standard

- efficient and complete implementations of the Fortran 2003 standard
  - procedure pointers (required for dynamic linking) and onject oriented features only correcty implemented by NAG and gfortran 4.8+

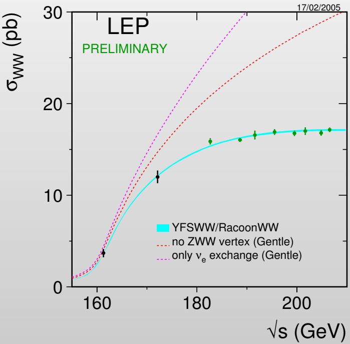
- efficient and complete implementations of the Fortran 2003 standard
  - procedure pointers (required for dynamic linking) and onject oriented features only correcty implemented by NAG and gfortran 4.8+
- use more symmetries to reduce the code size

- efficient and complete implementations of the Fortran 2003 standard
  - procedure pointers (required for dynamic linking) and onject oriented features only correcty implemented by NAG and gfortran 4.8+
- use more symmetries to reduce the code size
  - compiled code for multi jet cross sections at LHC can become larger than a giga[sic!]byte

- efficient and complete implementations of the Fortran 2003 standard
  - procedure pointers (required for dynamic linking) and onject oriented features only correcty implemented by NAG and gfortran 4.8+
- use more symmetries to reduce the code size
  - compiled code for multi jet cross sections at LHC can become larger than a giga[sic!]byte
- allow completely general vertex structures (UFO interface)

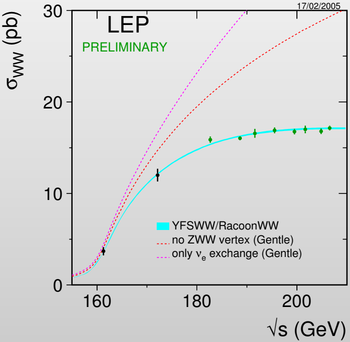
- efficient and complete implementations of the Fortran 2003 standard
  - procedure pointers (required for dynamic linking) and onject oriented features only correcty implemented by NAG and gfortran 4.8+
- use more symmetries to reduce the code size
  - compiled code for multi jet cross sections at LHC can become larger than a giga[sic!]byte
- allow completely general vertex structures (UFO interface)
- loops (proof-of-principle implementations exist, but are not yet completely general and fully automatic)

- LEP2: $e^+e^- \to W^+W^-$

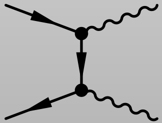


17/02/2005
LEP
PRELIMINARY
σWW (pb)
30
20
10
0
160
180
200
√s (GeV)
YFSWW/RacoonWW
no ZWW vertex (Gentle)
only νe exchange (Gentle)

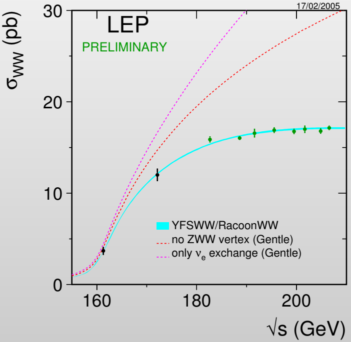
- LEP2: $e^+e^- \to W^+W^-$



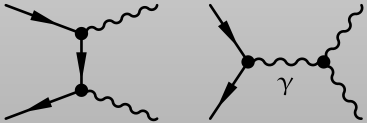
- gauge cancellations



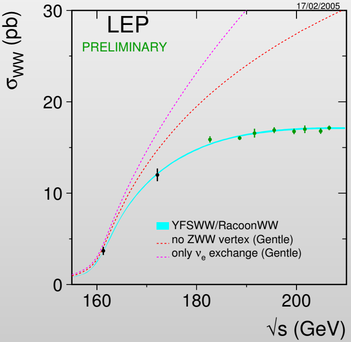
- LEP2: $e^+e^- \to W^+W^-$



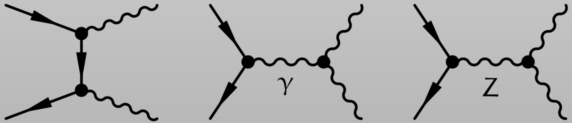
- gauge cancellations



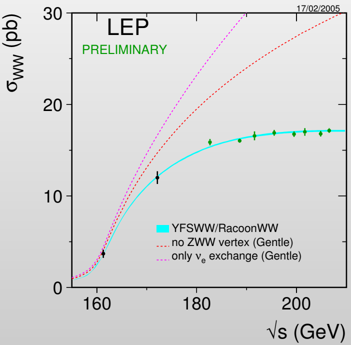
- LEP2: $e^+e^- \to W^+W^-$



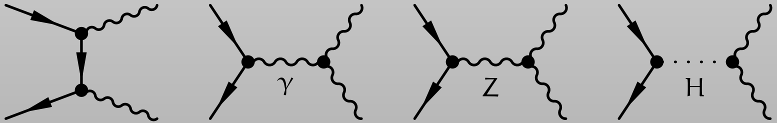
- gauge cancellations



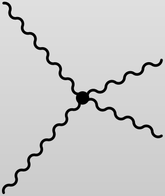
- LEP2: $e^+e^- \to W^+W^-$



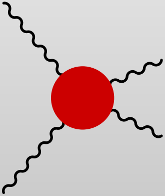
- gauge cancellations



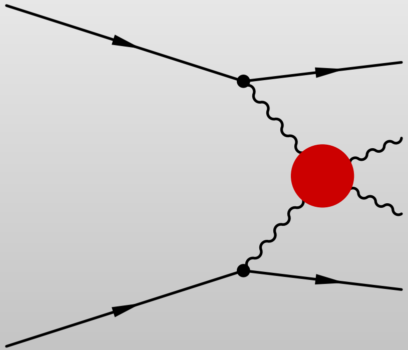
$$VV \to VV$$



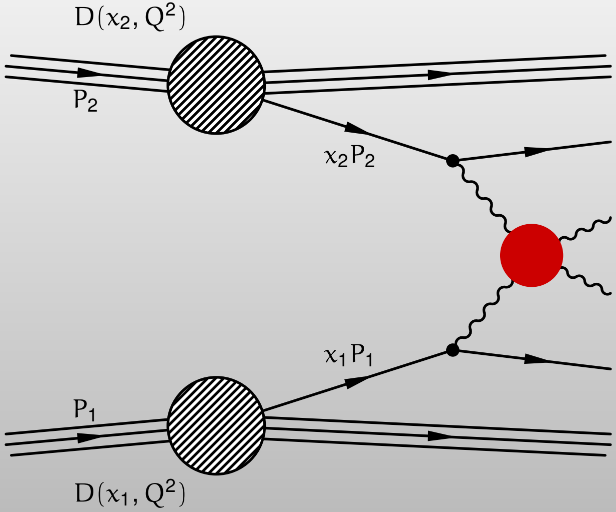
$$VV \to VV$$



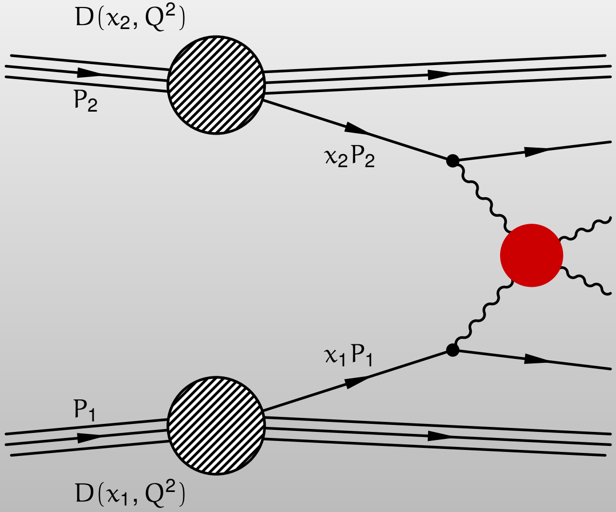
$$jj \to jjVV$$



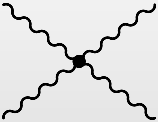
$$pp \to jjVV + X$$



$$pp \to jjVV + X \quad pp \to jj(VV \to 4f) + X$$

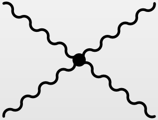


- $V_L V_L \to V_L V_L$



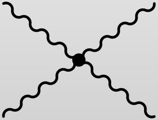
$\propto s^2$

- $V_L V_L \to V_L V_L$

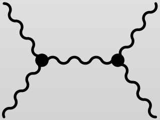


$\propto s^2$

- first set of cancellations



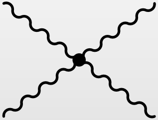
$+$



$\propto s^1$

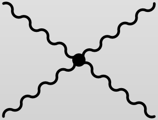
Vector Boson Scattering

- $V_L V_L \to V_L V_L$

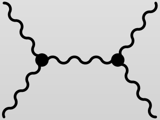


$\propto s^2$

- first set of cancellations

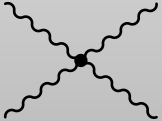


$+$

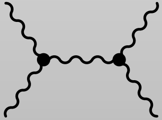


$\propto s^1$

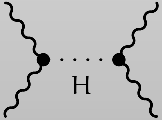
- second set of cancellations



$+$

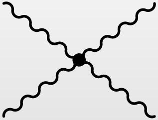


$+$



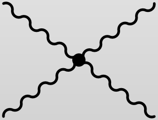
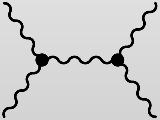
$\propto s^0$

- $V_L V_L \to V_L V_L$

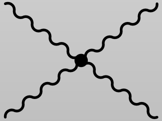
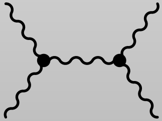
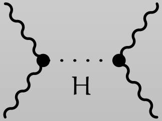


$\propto s^2$

- first set of cancellations

 $+$  $\propto s^1$

- second set of cancellations

 $+$  $+$ 

$\propto s^0$

$\therefore$ small changes in quartic coupling will have big effect on cross section!

- symmetry breaking sector: Higgs and Goldstone bosons

$$\mathbf{H} = \frac{1}{2}(v+h)\Sigma = \frac{1}{2}\begin{pmatrix} v+h-iw^3 & -i\sqrt{2}w^+ \\ -i\sqrt{2}w^- & v+h+iw^3 \end{pmatrix}$$

- symmetry breaking sector: Higgs and Goldstone bosons

$$\mathbf{H} = \frac{1}{2}(v+\mathrm{h})\Sigma = \frac{1}{2}\begin{pmatrix} v+\mathrm{h}-\mathrm{i}w^3 & -\mathrm{i}\sqrt{2}w^+ \\ -\mathrm{i}\sqrt{2}w^- & v+\mathrm{h}+\mathrm{i}w^3 \end{pmatrix}$$

- Standard Model (SM)

$$\mathrm{L}_{\text{min}} = -\frac{1}{2}\,\mathrm{tr}\left[\mathbf{W}_{\mu\nu}\mathbf{W}^{\mu\nu}\right] - \frac{1}{2}\,\mathrm{tr}\left[\mathbf{B}_{\mu\nu}\mathbf{B}^{\mu\nu}\right] + \mathrm{tr}\left[(\mathbf{D}_\mu\mathbf{H})^\dagger\,\mathbf{D}^\mu\mathbf{H}\right] + \mu^2\,\mathrm{tr}\left[\mathbf{H}^\dagger\mathbf{H}\right] - \frac{\lambda}{2}\left(\mathrm{tr}\left[\mathbf{H}^\dagger\mathbf{H}\right]\right)^2$$

- symmetry breaking sector: Higgs and Goldstone bosons

$$\mathbf{H} = \frac{1}{2}(v + h)\Sigma = \frac{1}{2}\begin{pmatrix} v + h - iw^3 & -i\sqrt{2}w^+ \\ -i\sqrt{2}w^- & v + h + iw^3 \end{pmatrix}$$

- Standard Model (SM)

$$\begin{aligned} L_{\text{min}} = & -\frac{1}{2}\operatorname{tr}\left[\mathbf{W}_{\mu\nu}\mathbf{W}^{\mu\nu}\right] - \frac{1}{2}\operatorname{tr}\left[\mathbf{B}_{\mu\nu}\mathbf{B}^{\mu\nu}\right] \\ & + \operatorname{tr}\left[(\mathbf{D}_\mu\mathbf{H})^\dagger \mathbf{D}^\mu\mathbf{H}\right] + \mu^2 \operatorname{tr}\left[\mathbf{H}^\dagger\mathbf{H}\right] - \frac{\lambda}{2}\left(\operatorname{tr}\left[\mathbf{H}^\dagger\mathbf{H}\right]\right)^2 \end{aligned}$$

- dimension-6 and dimension-8 deviations from the SM contributing to quartic interactions (trilinear assumed to be constrained)

$$\begin{aligned} \mathcal{L} = \; & F_{HD} \operatorname{tr}\left[\mathbf{H}^\dagger\mathbf{H} - \frac{v^2}{4}\right] \cdot \operatorname{tr}\left[(\mathbf{D}_\mu\mathbf{H})^\dagger (\mathbf{D}^\mu\mathbf{H})\right] \\ & + F_{S,0} \operatorname{tr}\left[(\mathbf{D}_\mu\mathbf{H})^\dagger \mathbf{D}_\nu\mathbf{H}\right] \cdot \operatorname{tr}\left[(\mathbf{D}^\mu\mathbf{H})^\dagger \mathbf{D}^\nu\mathbf{H}\right] \\ & + F_{S,1} \operatorname{tr}\left[(\mathbf{D}_\mu\mathbf{H})^\dagger \mathbf{D}^\mu\mathbf{H}\right] \cdot \operatorname{tr}\left[(\mathbf{D}_\nu\mathbf{H})^\dagger \mathbf{D}^\nu\mathbf{H}\right] \end{aligned}$$

- power counting:

$$F_{HD} \propto \frac{1}{\Lambda^2}$$
$$F_{S,n} \propto \frac{1}{\Lambda^4}$$

- power counting:

$$F_{HD} \propto \frac{1}{\Lambda^2}$$
$$F_{S,n} \propto \frac{1}{\Lambda^4}$$

∴ expectation

$$\frac{\delta_{HD}\sigma}{\sigma} \propto \frac{s}{\Lambda^2} + \dots$$
$$\frac{\delta_{S,n}\sigma}{\sigma} \propto \left(\frac{s}{\Lambda^2}\right)^2 + \dots$$

- power counting:

$$F_{HD} \propto \frac{1}{\Lambda^2}$$
$$F_{S,n} \propto \frac{1}{\Lambda^4}$$

∴ expectation

$$\frac{\delta_{HD}\sigma}{\sigma} \propto \frac{s}{\Lambda^2} + \ldots$$
$$\frac{\delta_{S,n}\sigma}{\sigma} \propto \left(\frac{s}{\Lambda^2}\right)^2 + \ldots$$

- iff there is a well defined and unique energy scale $\sqrt{s}$

- power counting:

$$F_{HD} \propto \frac{1}{\Lambda^2}$$
$$F_{S,n} \propto \frac{1}{\Lambda^4}$$

∴ expectation

$$\frac{\delta_{HD}\sigma}{\sigma} \propto \frac{s}{\Lambda^2} + \dots$$
$$\frac{\delta_{S,n}\sigma}{\sigma} \propto \left(\frac{s}{\Lambda^2}\right)^2 + \dots$$

- iff there is a well defined and unique energy scale $\sqrt{s}$

☺ at e^+e^- linear collider: accessible $\Lambda > \sqrt{s}$

- power counting:

$$F_{HD} \propto \frac{1}{\Lambda^2}$$
$$F_{S,n} \propto \frac{1}{\Lambda^4}$$

∴ expectation

$$\frac{\delta_{HD}\sigma}{\sigma} \propto \frac{s}{\Lambda^2} + \ldots$$
$$\frac{\delta_{S,n}\sigma}{\sigma} \propto \left(\frac{s}{\Lambda^2}\right)^2 + \ldots$$

- iff there is a well defined and unique energy scale $\sqrt{s}$

☺ at e^+e^- linear collider: accessible $\Lambda > \sqrt{s}$

☹ at LHC: accessible $\Lambda \sim x_1 x_2 s \ll s$, but some events with $x_1 x_2 s \lesssim s$

- power counting:

$$F_{HD} \propto \frac{1}{\Lambda^2}$$
$$F_{S,n} \propto \frac{1}{\Lambda^4}$$

∴ expectation

$$\frac{\delta_{HD}\sigma}{\sigma} \propto \frac{s}{\Lambda^2} + \ldots$$
$$\frac{\delta_{S,n}\sigma}{\sigma} \propto \left(\frac{s}{\Lambda^2}\right)^2 + \ldots$$

- iff there is a well defined and unique energy scale $\sqrt{s}$

☺ at e^+e^- linear collider: accessible $\Lambda > \sqrt{s}$

☹ at LHC: accessible $\Lambda \sim x_1 x_2 s \ll s$, but some events with $x_1 x_2 s \lesssim s$

∴ expansion questionable!!!

- Unitarity of $S = \mathbf{1} + \mathrm{i}\mathrm{T}$ provides nonlinear consistency relations, mixing orders of the expansion in $1/\Lambda$:

$$\mathrm{T}^\dagger \mathrm{T} = \mathrm{i}(\mathrm{T}^\dagger - \mathrm{T})$$

- Unitarity of $S = \mathbf{1} + \mathrm{i}T$ provides nonlinear consistency relations, mixing orders of the expansion in $1/\Lambda$:

$$T^\dagger T = \mathrm{i}(T^\dagger - T)$$

- continuous basis of momentum eigenstates

$$\int d\gamma \, \langle\alpha|T^\dagger|\gamma\rangle \, \langle\gamma|T|\beta\rangle = \mathrm{i}(\langle\alpha|T^\dagger|\beta\rangle - \langle\alpha|T|\beta\rangle)$$

- Unitarity of $S = \mathbf{1} + \mathrm{i}T$ provides nonlinear consistency relations, mixing orders of the expansion in $1/\Lambda$:

$$T^\dagger T = \mathrm{i}(T^\dagger - T)$$

- continuous basis of momentum eigenstates

$$\int d\gamma \, \langle\alpha|T^\dagger|\gamma\rangle \, \langle\gamma|T|\beta\rangle = \mathrm{i}(\langle\alpha|T^\dagger|\beta\rangle - \langle\alpha|T|\beta\rangle)$$

less useful than discrete partial wave amplitudes $a_L(s)$

$$A(s,t) = 32\pi \sum_L a_L(s)(2L+1)P_L(1+2t/s)$$

- Unitarity of $S = \mathbf{1} + \mathrm{i}T$ provides nonlinear consistency relations, mixing orders of the expansion in $1/\Lambda$:

$$T^\dagger T = \mathrm{i}(T^\dagger - T)$$

- continuous basis of momentum eigenstates

$$\int d\gamma \, \langle\alpha|T^\dagger|\gamma\rangle \, \langle\gamma|T|\beta\rangle = \mathrm{i}(\langle\alpha|T^\dagger|\beta\rangle - \langle\alpha|T|\beta\rangle)$$

less useful than discrete partial wave amplitudes $a_L(s)$

$$A(s,t) = 32\pi \sum_L a_L(s)(2L+1)P_L(1+2t/s)$$

which for elastic scattering must lie on the Argand circle

$$\left| a_L(s) - \frac{\mathrm{i}}{2} \right|^2 = \frac{1}{2}$$

of radius $1/2$ centered around $\mathrm{i}/2$

- Unitarity of $S = \mathbf{1} + iT$ provides nonlinear consistency relations, mixing orders of the expansion in $1/\Lambda$:

$$T^\dagger T = i(T^\dagger - T)$$

- continuous basis of momentum eigenstates

$$\int d\gamma \, \langle\alpha|T^\dagger|\gamma\rangle \, \langle\gamma|T|\beta\rangle = i(\langle\alpha|T^\dagger|\beta\rangle - \langle\alpha|T|\beta\rangle)$$

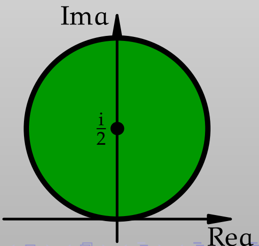
less useful than discrete partial wave amplitudes $a_L(s)$

$$A(s,t) = 32\pi \sum_L a_L(s)(2L+1)P_L(1+2t/s)$$

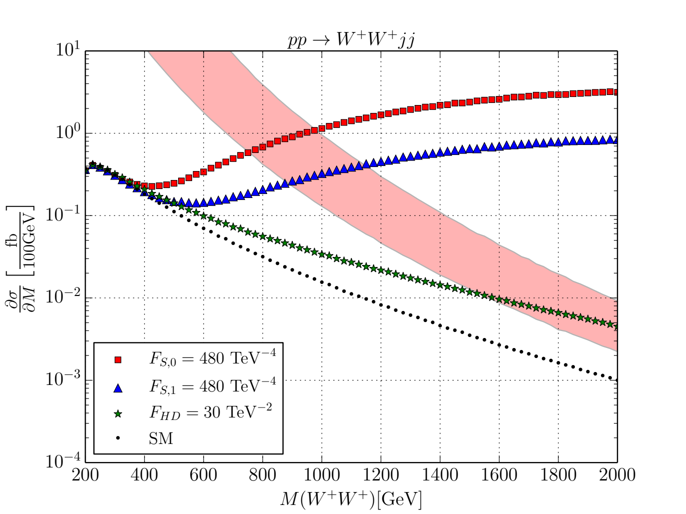
which for elastic scattering must lie on the Argand circle

$$\left| a_L(s) - \frac{i}{2} \right|^2 = \frac{1}{2}$$

of radius $1/2$ centered around $i/2$

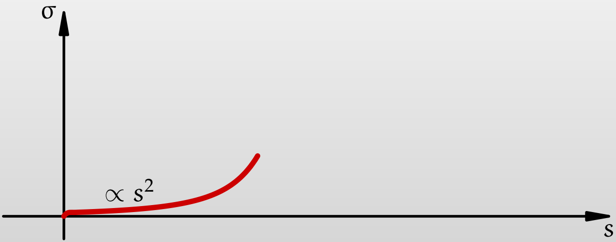


- Vector Boson Scattering at 14 TeV:

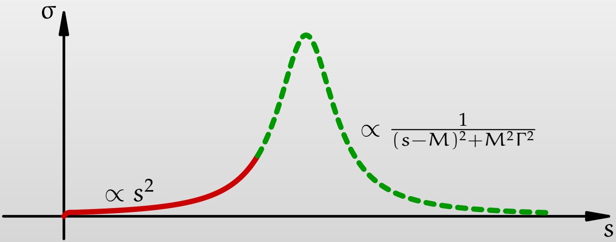


- lower/upper bound from saturation of $(\mathrm{I}, \mathrm{L}) = (2, 0)$ / $(\mathrm{I}, \mathrm{L}) = (2, 0)$ and $(2, 2)$ amplitudes
- cuts: $M_{jj} > 500$ GeV, $\Delta\eta_{jj} > 2.4$, $p_T^j > 20$ GeV, $|\eta_j| > 4.5$.

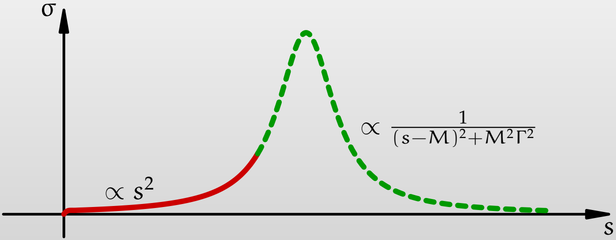
- ad hoc remedy



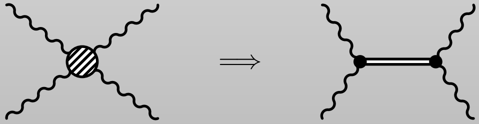
- ad hoc remedy



- ad hoc remedy



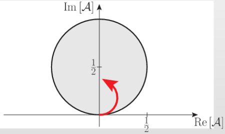
- corresponds to integrating in a new resonance



Possible behavior of scattering amplitudes

- inelastic scattering, opening up of other channels

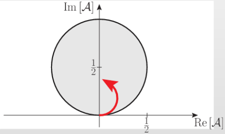
$$1 = |\langle VV|S|VV\rangle|^2 + |\langle X|S|VV\rangle|^2 + \ldots$$



Possible behavior of scattering amplitudes

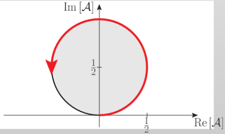
- inelastic scattering, opening up of other channels

$$1 = |\langle VV|S|VV\rangle|^2 + |\langle X|S|VV\rangle|^2 + \ldots$$



- resonance pole

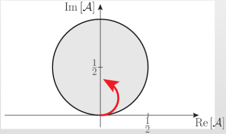
$$\langle VV|S|VV\rangle \propto -\frac{1}{s - M^2 + iM\Gamma} = \frac{1}{M^2} + \frac{s}{M^4} + \ldots$$



Possible behavior of scattering amplitudes

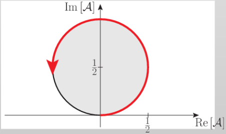
- inelastic scattering, opening up of other channels

$$1 = |\langle VV|S|VV\rangle|^2 + |\langle X|S|VV\rangle|^2 + \ldots$$

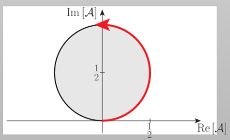


- resonance pole

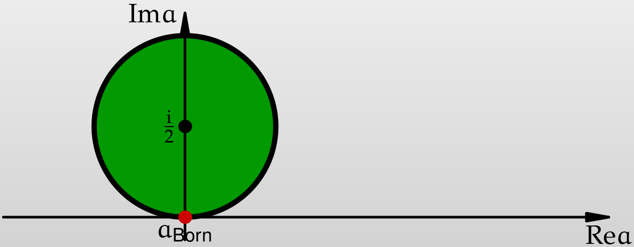
$$\langle VV|S|VV\rangle \propto -\frac{1}{s - M^2 + \mathrm{i}M\Gamma} = \frac{1}{M^2} + \frac{s}{M^4} + \ldots$$



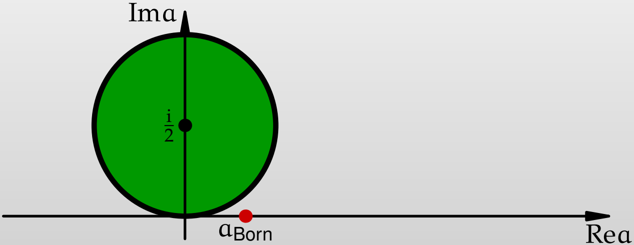
- saturation ("resonance at infinity")



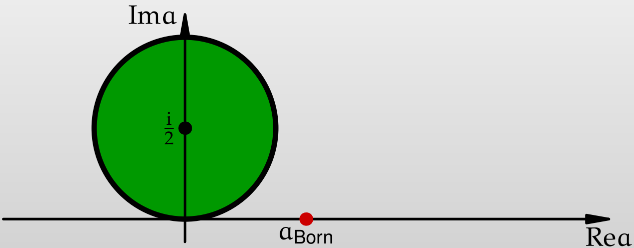
perturbation theory: Im $\langle VV|S|VV\rangle = 0$ at tree level



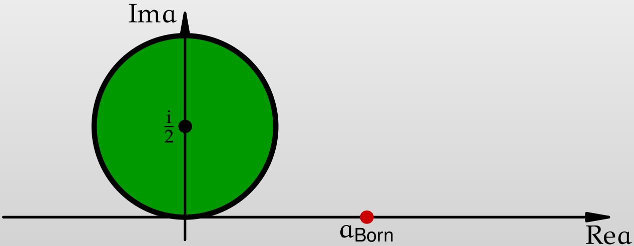
☹ perturbation theory: Im $\langle VV|S|VV\rangle = 0$ at tree level



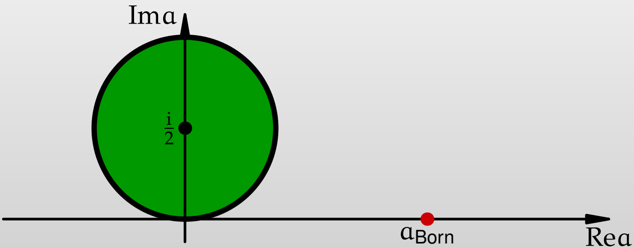
☹ perturbation theory: Im $\langle VV|S|VV\rangle = 0$ at tree level



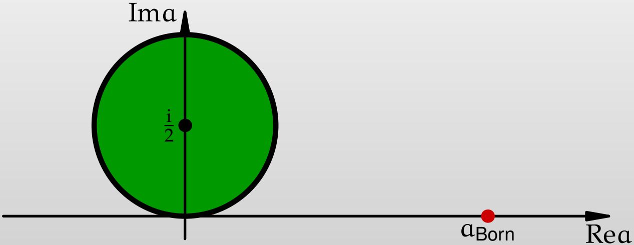
☹ perturbation theory: Im $\langle VV|S|VV\rangle = 0$ at tree level



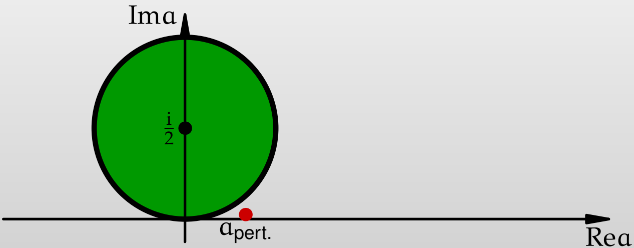
☹ perturbation theory: Im $\langle VV|S|VV\rangle = 0$ at tree level



perturbation theory: Im $\langle VV|S|VV\rangle = 0$ at tree level

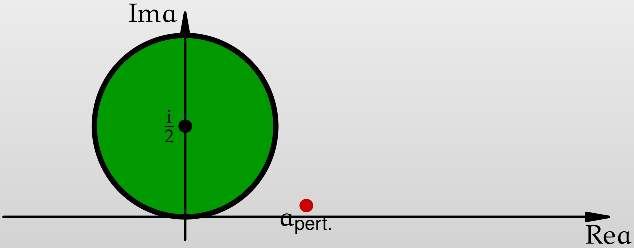


☹ perturbation theory: Im $\langle VV|S|VV\rangle = 0$ at tree level



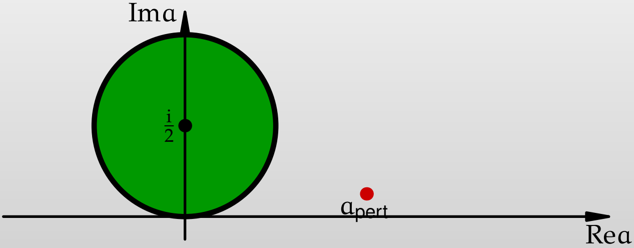
- perturbation theory: Im $\langle VV|S|VV\rangle$ suppressed w.r.t. Re $\langle VV|S|VV\rangle$ at higher orders

☹ perturbation theory: Im $\langle VV|S|VV\rangle = 0$ at tree level



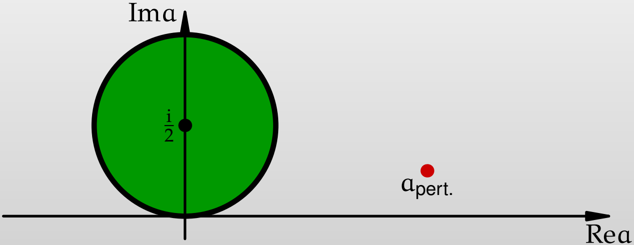
- perturbation theory: Im $\langle VV|S|VV\rangle$ suppressed w.r.t. Re $\langle VV|S|VV\rangle$ at higher orders

perturbation theory: Im $\langle VV|S|VV\rangle = 0$ at tree level



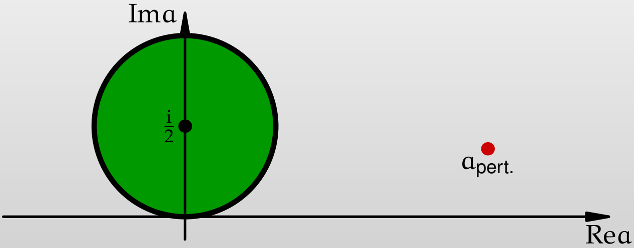
- perturbation theory: Im $\langle VV|S|VV\rangle$ suppressed w.r.t. Re $\langle VV|S|VV\rangle$ at higher orders

perturbation theory: Im $\langle VV|S|VV\rangle = 0$ at tree level



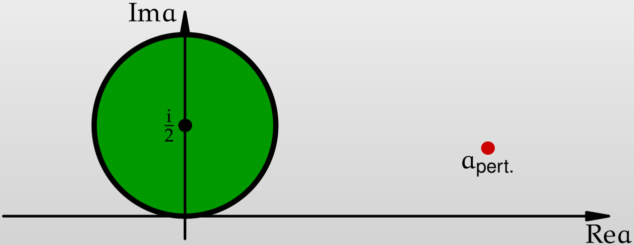
- perturbation theory: Im $\langle VV|S|VV\rangle$ suppressed w.r.t. Re $\langle VV|S|VV\rangle$ at higher orders

☹ perturbation theory: Im $\langle VV|S|VV\rangle = 0$ at tree level



- perturbation theory: Im $\langle VV|S|VV\rangle$ suppressed w.r.t. Re $\langle VV|S|VV\rangle$ at higher orders

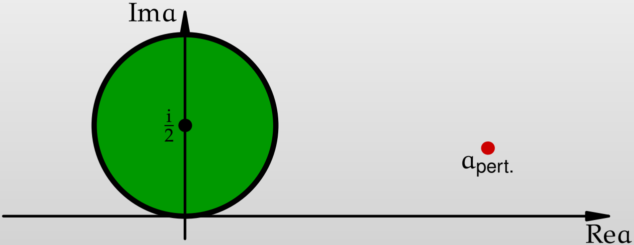
☹ perturbation theory: Im $\langle VV|S|VV\rangle = 0$ at tree level



- perturbation theory: Im $\langle VV|S|VV\rangle$ suppressed w.r.t. Re $\langle VV|S|VV\rangle$ at higher orders

∴ need non-perturbative approach

☹ perturbation theory: Im $\langle VV|S|VV\rangle = 0$ at tree level



- perturbation theory: Im $\langle VV|S|VV\rangle$ suppressed w.r.t. Re $\langle VV|S|VV\rangle$ at higher orders

∴ need non-perturbative approach

or to resum perturbation theory

- Ansatz #1 for manifestly unitary scattering matrix

$$S = \mathrm{e}^{\mathrm{i}A}$$

with $A^\dagger = A$

- Ansatz #1 for manifestly unitary scattering matrix

$$S = \mathrm{e}^{\mathrm{i}A}$$

with $A^\dagger = A$

☹ wraps around the Argand circle forever with growing coupling and energy

- Ansatz #1 for manifestly unitary scattering matrix

$$S = \mathrm{e}^{\mathrm{i}A}$$

with $A^\dagger = A$

☹ wraps around the Argand circle forever with growing coupling and energy

- Ansatz #2 (Cayley transform)

$$S = \frac{\mathbf{1} + \mathrm{i}\mathrm{K}/2}{\mathbf{1} - \mathrm{i}\mathrm{K}/2}$$

with $\mathrm{K}^\dagger = \mathrm{K}$

- Ansatz #1 for manifestly unitary scattering matrix

$$S = \mathrm{e}^{\mathrm{i}A}$$

with $A^\dagger = A$

☹ wraps around the Argand circle forever with growing coupling and energy

- Ansatz #2 (Cayley transform)

$$S = \frac{\mathbf{1} + \mathrm{i}K/2}{\mathbf{1} - \mathrm{i}K/2}$$

with $K^\dagger = K$

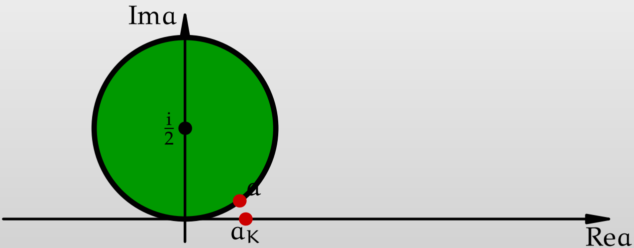
- i. e.

$$T = \frac{K}{\mathbf{1} - \mathrm{i}K/2} = K + \ldots$$

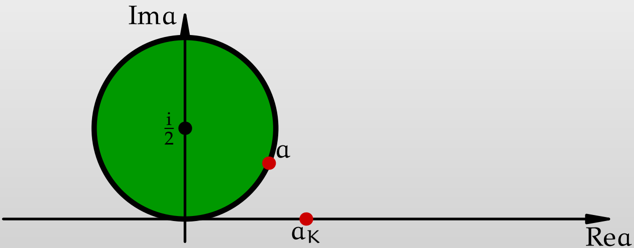
and

$$K = \frac{T}{\mathbf{1} + \mathrm{i}T/2} = T + \ldots.$$

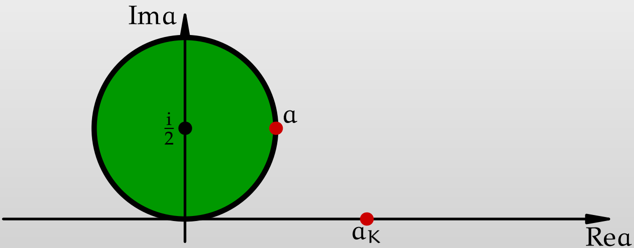
- geometric interpretation for tree level real eigenamplitudes:



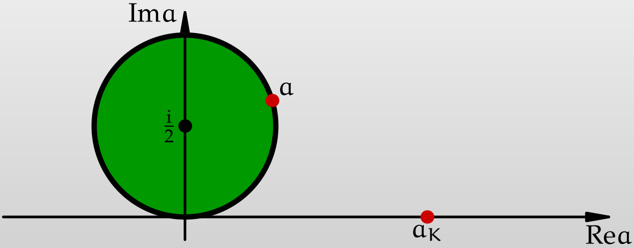
- geometric interpretation for tree level real eigenamplitudes:



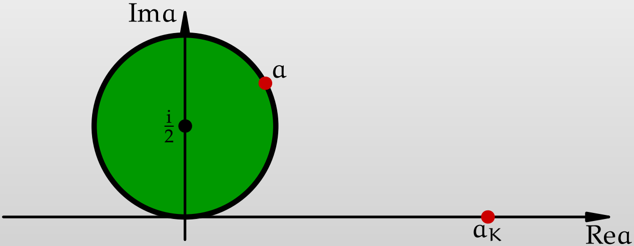
- geometric interpretation for tree level real eigenamplitudes:



- geometric interpretation for tree level real eigenamplitudes:

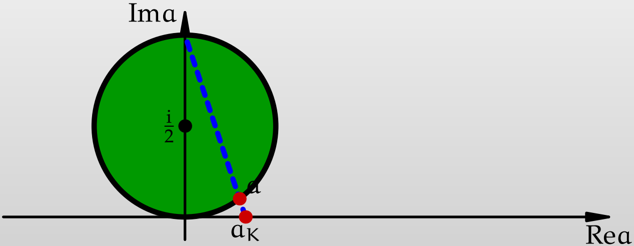


- geometric interpretation for tree level real eigenamplitudes:



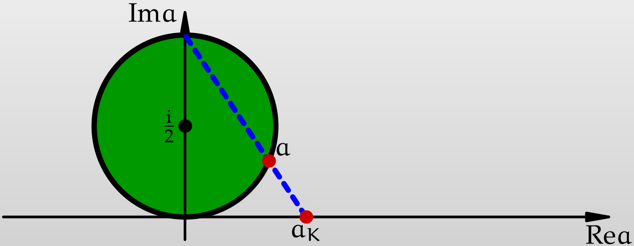
- geometric interpretation for tree level real eigenamplitudes:

stereographic projection



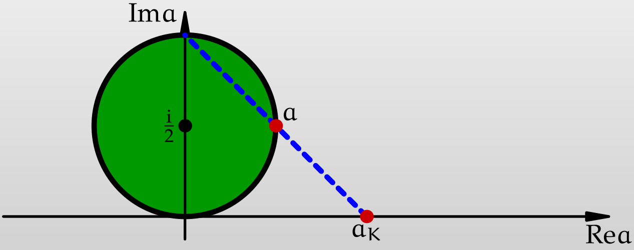
- geometric interpretation for tree level real eigenamplitudes:

stereographic projection



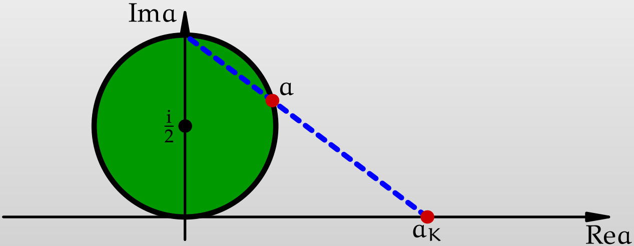
- geometric interpretation for tree level real eigenamplitudes:

stereographic projection



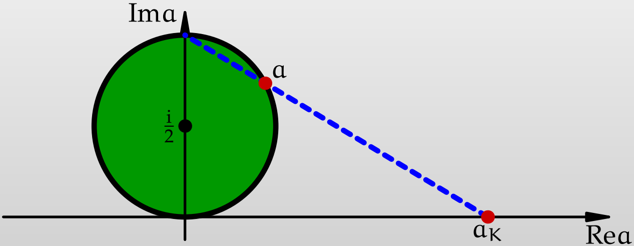
- geometric interpretation for tree level real eigenamplitudes:

stereographic projection



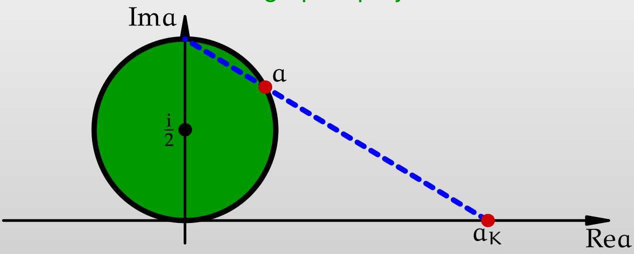
- geometric interpretation for tree level real eigenamplitudes:

stereographic projection



- geometric interpretation for tree level real eigenamplitudes:

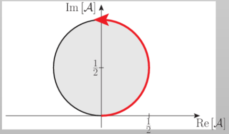
stereographic projection



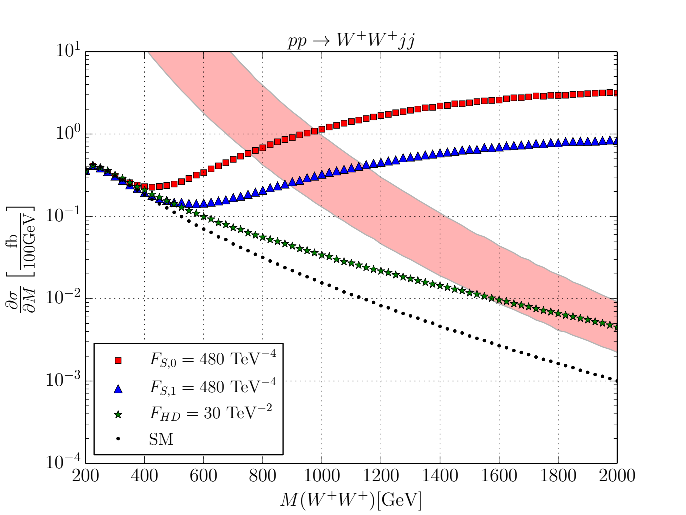
resummation of K-matrix corresponds to Dyson series

$$T = \frac{K}{1 - iK/2}$$

with pole at infinity

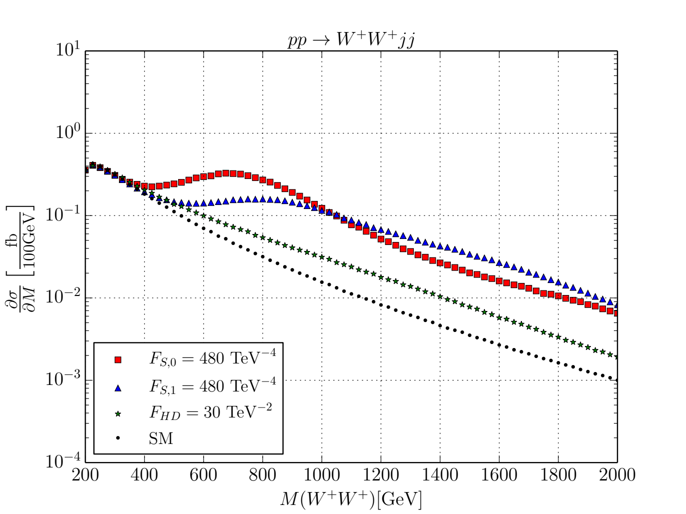


- Vector Boson Scattering at 14 TeV:



- lower/upper bound from saturation of $(\mathrm{I}, \mathrm{L}) = (2, 0)$ / $(\mathrm{I}, \mathrm{L}) = (2, 0)$ and $(2, 2)$ amplitudes
- cuts: $M_{jj} > 500$ GeV, $\Delta\eta_{jj} > 2.4$, $p_T^j > 20$ GeV, $|\eta_j| > 4.5$.

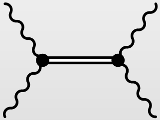
- Vector Boson Scattering at 14 TeV: unitarized



- cuts: $M_{jj} > 500$ GeV, $\Delta\eta_{jj} > 2.4$, $p_T^j > 20$ GeV, $|\eta_j| > 4.5$.

analytical example:

- lowest order $2 \to 2$ scattering with s-channel pole without Dyson resummation

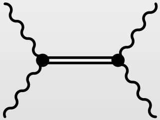


explodes

$$a_K^{(0)}(s) = \frac{\lambda}{s - m^2}$$

analytical example:

- lowest order $2 \to 2$ scattering with s-channel pole without Dyson resummation



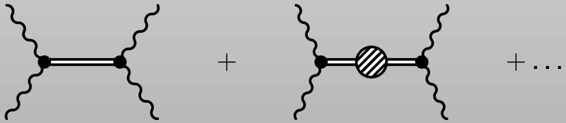
explodes

$$a_K^{(0)}(s) = \frac{\lambda}{s - m^2}$$

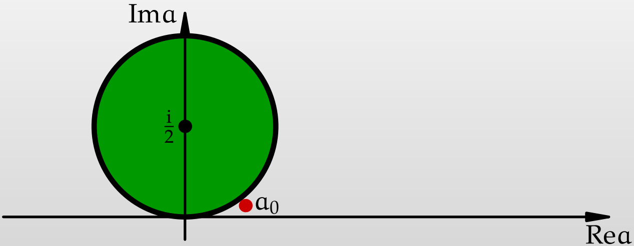
- K matrix resummation gives a width

$$a^{(0)}(s) = \frac{\lambda}{s - m^2 - i\lambda},$$

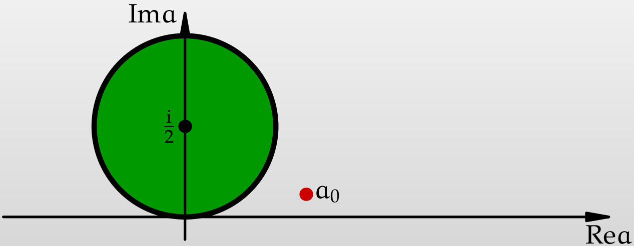
analogous to Dyson resummation



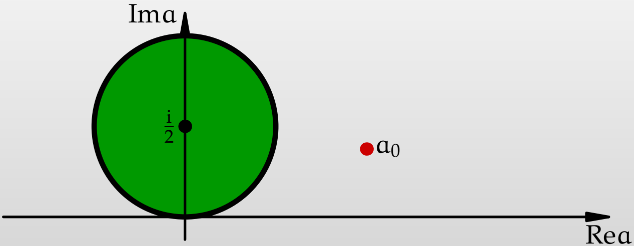
- generalization for higher order eigenamplitudes with imaginary parts [Kilian, TO, Reuter, Sekulla, arXiv:1408.6207]



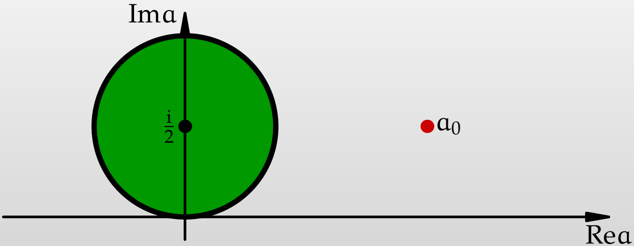
- generalization for higher order eigenamplitudes with imaginary parts [Kilian, TO, Reuter, Sekulla, arXiv:1408.6207]



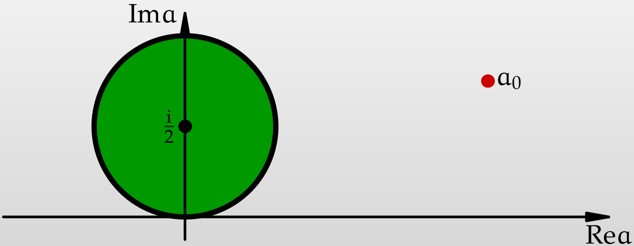
- generalization for higher order eigenamplitudes with imaginary parts [Kilian, TO, Reuter, Sekulla, arXiv:1408.6207]



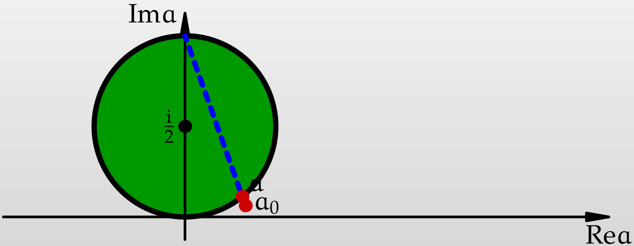
- generalization for higher order eigenamplitudes with imaginary parts [Kilian, TO, Reuter, Sekulla, arXiv:1408.6207]



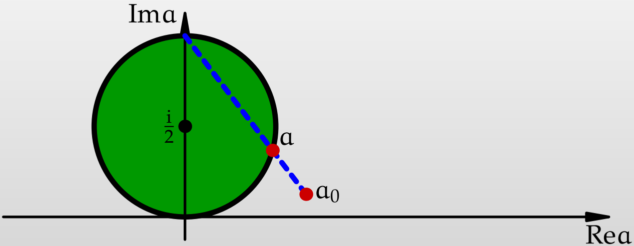
- generalization for higher order eigenamplitudes with imaginary parts [Kilian, TO, Reuter, Sekulla, arXiv:1408.6207]



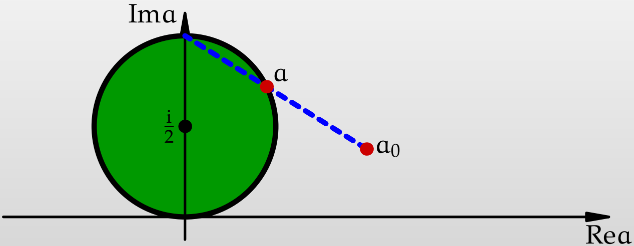
- generalization for higher order eigenamplitudes with imaginary parts [Kilian, TO, Reuter, Sekulla, arXiv:1408.6207]



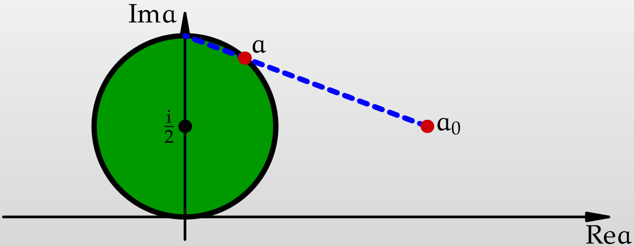
- generalization for higher order eigenamplitudes with imaginary parts [Kilian, TO, Reuter, Sekulla, arXiv:1408.6207]



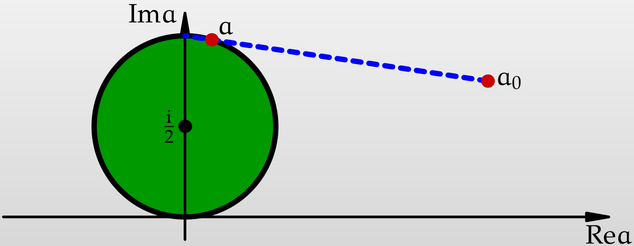
- generalization for higher order eigenamplitudes with imaginary parts [Kilian, TO, Reuter, Sekulla, arXiv:1408.6207]



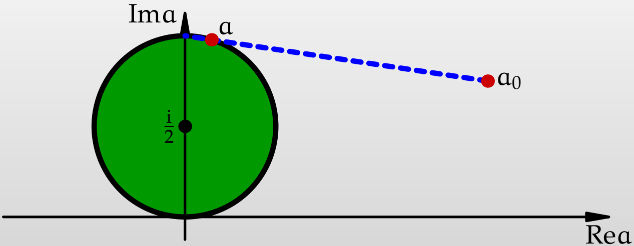
- generalization for higher order eigenamplitudes with imaginary parts [Kilian, TO, Reuter, Sekulla, arXiv:1408.6207]



- generalization for higher order eigenamplitudes with imaginary parts [Kilian, TO, Reuter, Sekulla, arXiv:1408.6207]



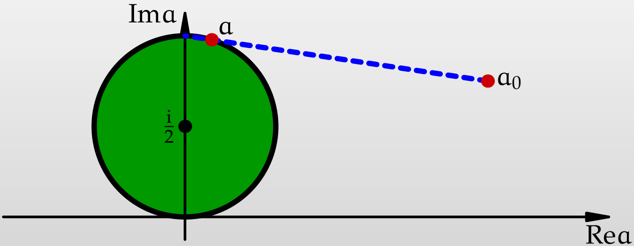
- generalization for higher order eigenamplitudes with imaginary parts [Kilian, TO, Reuter, Sekulla, arXiv:1408.6207]



- corresponding formula

$$a = \begin{cases} \dfrac{\mathrm{Re}\, a_0}{1 - i a_0^*} & \text{if } \mathrm{Im}\, a_0 < i, \\ i & \text{otherwise} \end{cases}$$

- generalization for higher order eigenamplitudes with imaginary parts [Kilian, TO, Reuter, Sekulla, arXiv:1408.6207]



- corresponding formula

$$a = \begin{cases} \dfrac{\mathrm{Re}\,a_0}{1 - \mathrm{i}a_0^*} & \text{if } \mathrm{Im}\,a_0 < \mathrm{i}, \\ \mathrm{i} & \text{otherwise} \end{cases}$$

☺ fixed point property

$$\left| a_0 - \frac{\mathrm{i}}{2} \right|^2 = \frac{1}{2} \Longrightarrow a = a_0$$

- generalization for coupled channels

- generalization for coupled channels
- for normal matrices $T_0^\dagger T_0 = T_0 T_0^\dagger$:

$$T = \frac{\mathrm{Re} T_0}{\mathbf{1} - \frac{i}{2} T_0^\dagger}.$$

- generalization for coupled channels
- for normal matrices $T_0^\dagger T_0 = T_0 T_0^\dagger$:

$$T = \frac{\mathrm{Re}T_0}{\mathbf{1} - \frac{i}{2}T_0^\dagger}.$$

- completely general $T_0^\dagger T_0 \neq T_0 T_0^\dagger$:

$$\begin{aligned} T &= \frac{1}{\sqrt{\mathbf{1} - \frac{1}{2}\mathrm{Im}T_0}}\mathrm{Re}T_0 \frac{1}{\mathbf{1} - \frac{i}{2}T_0^\dagger}\sqrt{\mathbf{1} - \frac{1}{2}\mathrm{Im}T_0} \\ &= \sqrt{\mathbf{1} - \frac{1}{2}\mathrm{Im}T_0}\frac{1}{\mathbf{1} - \frac{i}{2}T_0^\dagger}\mathrm{Re}T_0 \frac{1}{\sqrt{\mathbf{1} - \frac{1}{2}\mathrm{Im}T_0}} \end{aligned}$$

- generalization for coupled channels
- for normal matrices $\mathrm{T}_0^\dagger \mathrm{T}_0 = \mathrm{T}_0 \mathrm{T}_0^\dagger$:

$$\mathrm{T} = \frac{\mathrm{Re}\mathrm{T}_0}{\mathbf{1} - \frac{\mathrm{i}}{2}\mathrm{T}_0^\dagger}.$$

- completely general $\mathrm{T}_0^\dagger \mathrm{T}_0 \neq \mathrm{T}_0 \mathrm{T}_0^\dagger$:

$$\begin{aligned}\mathrm{T} &= \frac{1}{\sqrt{\mathbf{1} - \frac{1}{2}\mathrm{Im}\mathrm{T}_0}} \mathrm{Re}\mathrm{T}_0 \frac{1}{\mathbf{1} - \frac{\mathrm{i}}{2}\mathrm{T}_0^\dagger} \sqrt{\mathbf{1} - \frac{1}{2}\mathrm{Im}\mathrm{T}_0} \\ &= \sqrt{\mathbf{1} - \frac{1}{2}\mathrm{Im}\mathrm{T}_0} \frac{1}{\mathbf{1} - \frac{\mathrm{i}}{2}\mathrm{T}_0^\dagger} \mathrm{Re}\mathrm{T}_0 \frac{1}{\sqrt{\mathbf{1} - \frac{1}{2}\mathrm{Im}\mathrm{T}_0}}\end{aligned}$$

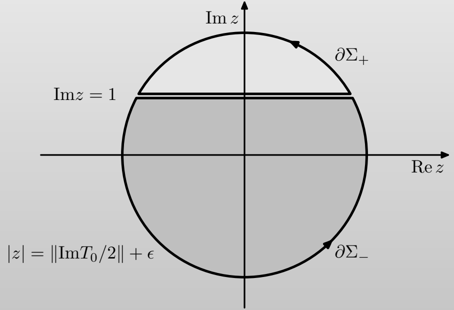
- NB: $\sqrt{A}$ of operator A defined via contour integration

$$\hat{\mathrm{f}}(A) = \int_{\partial\Sigma:\sigma(A)\subseteq\Sigma} \frac{\mathrm{d}z}{2\pi\mathrm{i}} \frac{\mathrm{f}(z)}{z\mathbf{1} - A}$$

- for $\mathrm{Im}T \not< 2$, we can construct projectors

$$P_{A,\Sigma} = \int_{\partial\Sigma} \frac{dz}{2\pi i} \frac{1}{z\mathbf{1} - A}. \tag{1}$$

again via contour integration

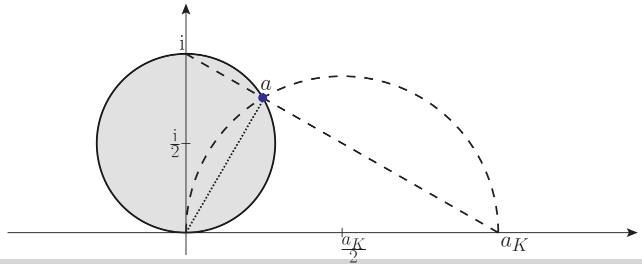


to split into (generalized) eigenspaces

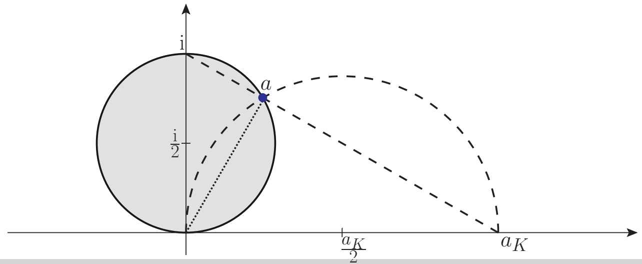
$$\mathbf{1} = P_{\mathrm{Im}T_0/2,\Sigma_+} + P_{\mathrm{Im}T_0/2,\Sigma_-} \tag{2}$$

and prevent “overshooting” even if $T_0^\dagger T_0 \neq T_0 T_0^\dagger$.

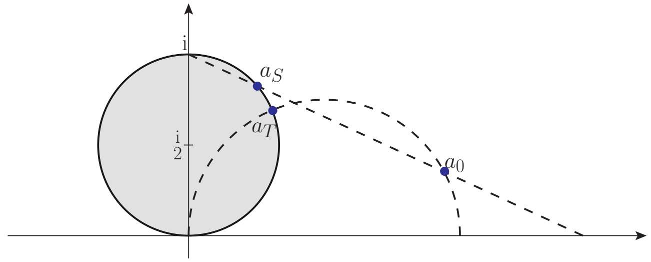
- for $\mathrm{Im}T = 0$ stereographic projection equivalent to semicircle construction (cf. "Thales' circle")



- for $\mathrm{Im}T = 0$ stereographic projection equivalent to semicircle construction (cf. “Thales’ circle”)



- for $\mathrm{Im}T \neq 0$



- automated construction of efficient tree level matrix elements well understood

- automated construction of efficient tree level matrix elements well understood
- topics not covered due to lack of time and/or overlap w/ other lectures

- automated construction of efficient tree level matrix elements well understood
- topics not covered due to lack of time and/or overlap w/ other lectures
  - MHV amplitudes and twistors

Conclusions & Outlook

- automated construction of efficient tree level matrix elements well understood
- topics not covered due to lack of time and/or overlap w/ other lectures
  - MHV amplitudes and twistors
  - computer aided multi loop calculations

- automated construction of efficient tree level matrix elements well understood
- topics not covered due to lack of time and/or overlap w/ other lectures
  - MHV amplitudes and twistors
  - computer aided multi loop calculations
  - fully automated 1 loop calculations: FeynArts, FormCalc, LoopTools, “OPP”

- automated construction of efficient tree level matrix elements well understood
- topics not covered due to lack of time and/or overlap w/ other lectures
  - MHV amplitudes and twistors
  - computer aided multi loop calculations
  - fully automated 1 loop calculations: FeynArts, FormCalc, LoopTools, "OPP"
  - sector decomposition GOLEM

Conclusions & Outlook

- automated construction of efficient tree level matrix elements well understood
- topics not covered due to lack of time and/or overlap w/ other lectures
  - MHV amplitudes and twistors
  - computer aided multi loop calculations
  - fully automated 1 loop calculations: FeynArts, FormCalc, LoopTools, "OPP"
  - sector decomposition GOLEM
  - dipole subtraction MadDipole

- automated construction of efficient tree level matrix elements well understood
- topics not covered due to lack of time and/or overlap w/ other lectures
  - MHV amplitudes and twistors
  - computer aided multi loop calculations
  - fully automated 1 loop calculations: FeynArts, FormCalc, LoopTools, “OPP”
  - sector decomposition GOLEM
  - dipole subtraction MadDipole
  - NLO matching