Perturbative Computations in Quantum Field Theory

Tobias Huber
Universität Siegen

Universität Siegen

TP1 Theoretical Particle Physics

CPPS Center for Particle Physics Siegen

Collaborative Research Center TRR 257

P3H

Particle Physics Phenomenology after the Higgs Discovery

CPPS retreat Meinerzhagen, February 14th – 16th, 2024

Equations of motion in classical mechanics

- Example: Bead on rotating ring
- Lagrangian function

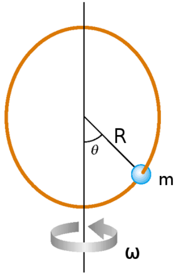
$$L = \frac{m}{2}\left[R^2\,\dot\theta^2 + \omega^2 R^2 \sin^2\theta\right] + m\,g\,R\,\cos\theta$$

- Equation of motion
  (Euler-Lagrange equation):

$$\frac{d}{dt}\frac{\partial L}{\partial\dot\theta} - \frac{\partial L}{\partial\theta} = 0$$

- Yields:

$$\ddot\theta = \omega^2 \sin\theta\cos\theta - \frac{g}{R}\,\sin\theta$$



Equations of motion in field theory

- Example: QED
- Lagrangian density

$$\mathcal{L} = -\frac{1}{4} F^{\mu\nu} F_{\mu\nu} + \bar{\psi}(i\partial\!\!\!/ - m)\psi + e\,\bar{\psi}A\!\!\!/\psi$$

  - ψ: Fermion field (electron, positron)
  - A^μ: Photon field
  - $F^{\mu\nu} = \partial^\mu A^\nu - \partial^\nu A^\mu$: Field strength tensor
  - $e = \sqrt{4\pi\alpha}$: coupling. $\alpha \sim \frac{1}{137}$

- Euler-Lagrange equations:

$$\partial_\mu \frac{\partial \mathcal{L}}{\partial(\partial_\mu \bar{\psi})} - \frac{\partial \mathcal{L}}{\partial \bar{\psi}} = 0$$

$$\partial_\mu \frac{\partial \mathcal{L}}{\partial(\partial_\mu A_\nu)} - \frac{\partial \mathcal{L}}{\partial A_\nu} = 0$$

- Yields:

$$(i\partial\!\!\!/ - m)\psi = -e\,A\!\!\!/\psi$$

$$\partial_\mu F^{\mu\nu} = -e\,\bar{\psi}\gamma^\nu\psi$$

- Coupled, nonlinear equations
  - Cannot be solved in closed form

Perturbative expansion

- Idea: Series expansion in small coupling

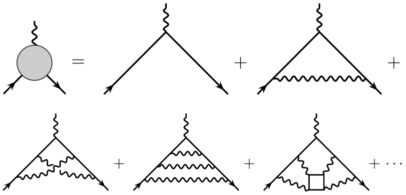
$$\alpha \sim \frac{1}{137} \qquad \alpha_s \sim 0.1 \;-\; 0.2$$

$$\mathcal{A} = \mathcal{O}(1) + \mathcal{O}(\alpha) + \mathcal{O}(\alpha^2) + \ldots$$

- At each order, more Feynman diagrams with more loops
- For each loop, have one unconstrained four-momentum ℓ^μ, to be integrated over with measure

$$\int \frac{d^4\ell}{(2\pi)^4}$$

- $a_e = (g_e - 2)/2$ of electron



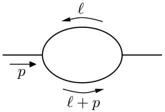
$a_e = 0$	(leading order)
$a_e = \dfrac{\alpha}{2\pi} \approx 0.0011614$	(to $\mathcal{O}(\alpha)$)
$a_e = 0.001159652181643(764)$	(to $\mathcal{O}(\alpha^5)$)
$a_e = 0.00115965218059(13)$	(experiment)

IBP reduction

- A multi-loop amplitude contains easily millions of terms (integrals)
- Not all integrals independents
- Integration-by-parts (IBP) equations [Tkachov'81;Chetyrkin,Tkachov'81]

$$\int \frac{d^d \ell_1}{i\pi^{d/2}} \cdots \int \frac{d^d \ell_L}{i\pi^{d/2}} \, \frac{\partial}{\partial \ell_i^\mu} \, \frac{v_j^\mu}{D_1^{a_1} \cdots D_n^{a_n}} = 0$$

- For the one-loop bubble



$(p^2 \neq 0)$

$\underline{v = \ell:}$

$$0 = (d - a_2 - 2a_1) F(a_1, a_2) - a_2 p^2 F(a_1, a_2 + 1) \\ - a_2 F(a_1 - 1, a_2 + 1)$$

$\underline{v = p:}$

$$0 = (a_1 - a_2) F(a_1, a_2) + a_2 p^2 F(a_1, a_2 + 1) - a_1 p^2 F(a_1 + 1, a_2) \\ + a_2 F(a_1 - 1, a_2 + 1) - a_1 F(a_1 + 1, a_2 - 1)$$

$$F(a_1, a_2) = \int \frac{\mathrm{d}^d \ell}{i\pi^{d/2}} \frac{1}{D_1^{a_1} D_2^{a_2}}\,, \qquad a_i \in \mathbb{Z}$$

$$\begin{aligned} D_1 &= -\ell^2 \\ D_2 &= -(\ell + p)^2 \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} \ell^2 &= -D_1 \\ \ell \cdot p &= \frac{1}{2}(D_1 - D_2 - p^2) \end{aligned}$$

IBP reduction: Laporta's algorithm

- Laporta's algorithm [Laporta'01]
  - Specialize set of IBP equations to a range of integer indices
  - Generate system of linear equations

include symmetries

directly discard scaleless integrals

$$
\begin{aligned}
0 &= F(2,1) - F(1,2) \\
0 &= (d-3)F(1,1) - p^2 F(1,2) \\
0 &= (d-4)F(1,2) - 2p^2 F(1,3) \\
0 &= 2p^2 F(1,3) - p^2 F(2,2) - F(1,2) - F(2,1) \\
0 &= (d-5)F(2,1) - p^2 F(2,2) - F(1,2) \\
&\vdots
\end{aligned}
$$

  - Solve system by Gaussian elimination.

    Express each integral as a linear combination of independent master integrals.

IBP reduction: Laporta's algorithm

- Calcuation of higher loop-orders entails several challenges
  - number of Feynman diagrams
  - size of intermediate expressions
  - complexity of loop integrals (# loops, legs, scales)
- Many sophisticated public and private tools exist
  - AIR [Anastasiou,Lazopoulos'04]
  - Reduze [Studerus'09; Manteuffel,Studerus'12]
  - FIRE [Smirnov et al.'08+]
  - LiteRed [Lee'12+]
  - Kira [Maierhöfer,Usovitsch,Uwer'17; Klappert,Lange,Maierhöfer,Usovitsch'20]
  - Crusher . . . [Marquard,Seidel]

- Mostly based on Laporta's algorithm [Laporta'01]
  - Solves IBP equations for numerical values of indices with Gaussian elimination.
- Several refinements exist, e.g.
  - Parallelization
  - Methods from finite fields
    [v. Manteuffel,Schabinger'14; Smirnov,Chukharev'19]
    [Peraro'16'19; Klappert,Klein,Lange'19'20]
- Drawbacks
  - Compute many more integrals than required
  - Large storage required for results of $10^{\sim 4-6}$ integrals

Algebraic-geometry approach

- New ideas from
  - algebraic geometry [Larsen,Zhang'14; Böhm et al.'18'19]
  - syzygy equations [Kosower et al.'10'18; Schabinger et al.'11'20; Ita'15; Böhm et al.'17]

$$A \cdot B = 0$$

  - Linear algebra
- New approach [Barakat,TH et al. 2020+]
  - Leave propagator powers symbolic
  - Derive so-called normal-form IBP relations

- Normal-form IBP equations

$$R_i \;=\; a_i\, D_i^- - \mathrm{NF}_G\left(a_i D_i^-\right) \;\in\; I_{\mathrm{IBP}}$$

  - e.g. for one-loop bubble

$$\mathrm{NF}_G\left(a_1 D_1^-\right) = \frac{(d-a_1-a_2-1)(d-2a_1-2a_2)}{p^2(d-2a_1-2)}$$

$$\mathrm{NF}_G\left(a_2 D_2^-\right) = \frac{(d-a_1-a_2-1)(d-2a_1-2a_2)}{p^2(d-2a_2-2)}$$

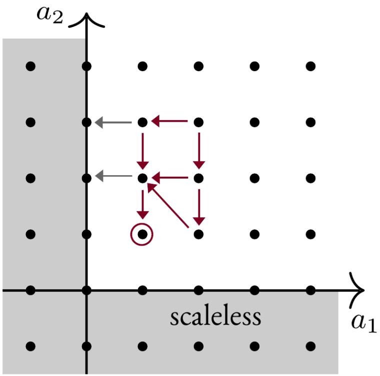
  - Allow for a straightforward reduction
  - All you have to store
  - Easy to implement in Mathematica or FORM
  - Well-suited for parallelization
  - Allows for fast reduction, also of high propagator powers

Laporta vs. Normal-form IBPs

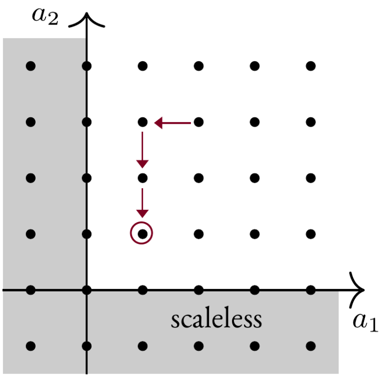
- Compare reduction with Laporta vs. Normal-form IBPs for the one-loop massless bubble

Reduction of $F(3,2)$:

using standard IBPs



using $\widehat{R}_i = \widehat{a}_i \widehat{D}_i^- - \mathrm{NF}_G(\widehat{a}_i \widehat{D}_i^-)$



Computing needs

- Jobs run, depending on complexity, on Laptop, desktop, OMNI, cluster in Karlsruhe
- Need modern CAS (Mathematica, C++, Fermat, ...)
- RAM, up to $\mathcal{O}(10^{2-3}\mathrm{GB})$
- storage space up to $\mathcal{O}(1\mathrm{TB})$
- Fast I/O traffic, read/write on local disk (scratch)
- Parallelizable ($\mathcal{O}(20)$ cores)
- Job runtimes of up to $\mathcal{O}(\mathrm{weeks})$

- Training for students in CAS, programming needed

- Local resources improvable in machines with a lot of RAM, Mathematica licenses, job length

- Not so much in need of GPUs, network