On two-loop amplitudes for $t\bar{t}H$ production

Vitaly Magerya
With B. Agarwal, G. Heinrich, S.P. Jones, M. Kerner, S.Y. Klein, J. Lang, A. Olsson

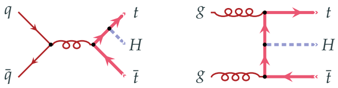
Institute for Theoretical Physics

Annual Meeting of the CRC TRR 257,
March 11, 2024, Karlsruhe

$t\bar{t}H$ production at the LHC

CRC project **B1b**, “Precision top-quark physics at the LHC”:

b) *“NNLO QCD predictions for $pp \to t\bar{t}H$ decays”.*



First observation at LHC reported in 2018. [CMS ’18, ’18, ’20, ’20, ’22; ATLAS ’18, ’20, ’23]

Current results, based on data from LHC Run 2 (2015–2018):

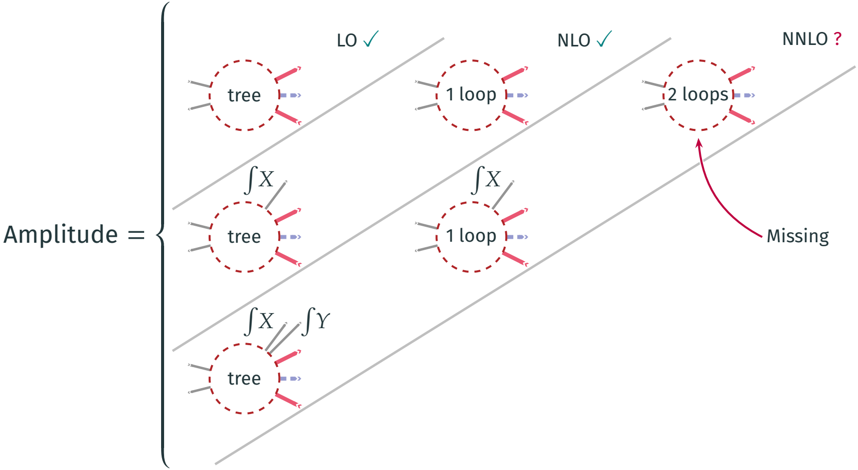
	$\sigma_{t\bar{t}H}/\sigma_{t\bar{t}H,\text{SM}}$			$\mathscr{L}$
ATLAS	0.92	$^{+0.19}_{-0.19}$(stat)	$^{+0.17}_{-0.13}$(syst)	$139\,\text{fb}^{-1}$
CMS	1.43	$^{+0.33}_{-0.31}$(stat)	$^{+0.21}_{-0.15}$(syst)	$137\,\text{fb}^{-1}$

HL-LHC will have $\mathscr{L} \sim 3000\,\text{fb}^{-1}$, reducing *statistical* uncertainty by 4x–5x.

To reduce *systematic* uncertainty: *NNLO calculation is needed*.

[HL-LHC ’19; Les Houches ’21; Snowmass ’22]

Parts of a NNLO calculation



Big missing part for NNLO: *two-loop virtual amplitudes*.

Theory results for $t\bar{t}H$ production

NLO:

- NLO QCD [Beenakker, Dittmaier, Krämer, Plümper, Spira, Zerwas ’01]
  [Reina, Dawson ’01]
  [Reina, Dawson, Wackeroth ’01]
  [Beenakker, Dittmaier, Krämer, Plümper, Spira, Zerwas ’02]
  [Dawson, Jackson, Orr, Reina, Wackeroth ’03]
- NLO QCD, parton shower [Frederix, Frixione, Hirschi, Maltoni, Pittau, Torrielli ’11]
  [Garzelli, Kardos, Papadopoulos, Trocsanyi ’11]
  [Hartanto, Jager, Reina, Wackeroth ’15]
- NLO QCD+EW [Frixione, Hirschi, Pagani, Shao, and Zaro ’14]
- NLO QCD+EW, NWA [Zhang, Ma, Zhang, Chen, Guo ’14]
  [Frixione, Hirschi, Pagani, Shao, and Zaro ’15]
- NLO QCD, off-shell [Denner, Feger ’15]
  [Stremmer, Worek ’21]
  [Denner, Lang, Pellen ’20]
  [Bevilacqua, Bi, Hartanto, Kraus, Lupattelli, Worek ’22]

Theory results for $t\bar{t}H$ production, II

NLO, contd.:

- NLO+NLL QCD [Kulesza, Motyka, Stebel, Theeuwes '15]
  [Ju, Yang '19]
- NLO+NNLL QCD [Broggio, Ferroglia, Pecjak, Signer, Yang '15]
  [Broggio, Ferroglia, Pecjak, Yang '16]
  [Kulesza, Motyka, Stebel, Theeuwes '17]
- NLO QCD+SMEFT [Maltoni, Vryonidou, Zhang '16]
- NLO QCD+EW, off-shell [Denner, Lang, Pellen, Uccirati '16]
- NLO+NNLL QCD+EW [Broggio, Ferroglia, Frederix, Pagani, Pecjak, Tsinikos '19]
- NLO QCD to $\mathcal{O}(\varepsilon^2)$ [Buccioni, Kreer, Liu, Tancredi '23]
- $t \to H$ fragmentation functions at $\mathcal{O}(y_t^2 \alpha_s)$
  [Brancaccio, Czakon, Generet, Krämer '21]

Theory results for $t\bar{t}H$ production, III

NNLO:

- NNLO QCD, flavour off-diagonal [Catani, Fabre, Grazzini, Kallweit '21]
- NNLO QCD total cross-section, soft Higgs
  [Catani, Devoto, Grazzini, Kallweit, Mazzitelli, Savoini '22]
- Two-loop QCD virtual amplitude, IR poles [Chen, Ma, Wang, Yang, Ye '22]
- Two-loop QCD master integrals, N_f-part, leading N_c
  [Cordero, Figueiredo, Kraus, Page, Reina '23]
- Two-loop QCD virtual amplitude, high-energy boosted limit
  [Wang, Xia, Yang, Ye '24]

This talk:

- *Two-loop QCD virtual amplitude, $q\bar{q}$ channel, N_f-part*
  [Agarwal, Heinrich, Jones, Kerner, Klein, Lang, V.M., Olsson '24]

The amplitude

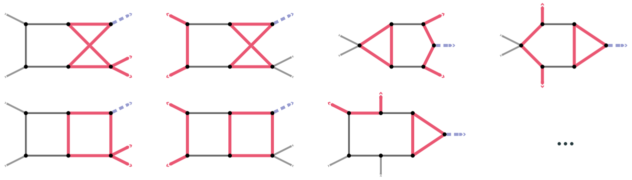
Model: QCD with a scalar H, n_l light (massless) quarks, n_h heavy (top) quarks.
Amplitude of $q\bar{q} \to t\bar{t}H$ projected onto Born, and decomposed in α_s as

$$\langle \mathrm{AMP} \,|\, \mathrm{AMP}_{\mathrm{tree}} \rangle = \mathscr{A} + \left(\frac{\alpha_s}{2\pi}\right) \mathscr{B} + \left(\frac{\alpha_s}{2\pi}\right)^2 \mathscr{C} \,.$$

As a proof-of-concept: only parts proportional to *n_l or n_h* in $\mathscr{C}$ for now.

Why is the calculation complicated?

1. IBP reduction of the amplitude to master integrals is too complicated to be computed symbolically.
   * 5 legs and 2 masses $(m_t, m_H) \Rightarrow$ 7 scales (6 scaleless variables).
2. Massive two-loop integrals contributing to $\mathscr{C}$ are not known analytically.



Calculation method

1. Generate Feynman diagrams. [QGRAF]
   - $\Rightarrow$ 249 non-zero diagrams.
2. Insert Feynman rules, apply the projector $|\,\text{AMP}_{\text{tree}}\rangle$.
3. Sum over the spinor and color tensors. [FORM; COLOR.H]
   - $\Rightarrow$ ~20000 scalar integrals;
   - $\Rightarrow$ 9 structures: $\{n_h|n_l\}\, C_A C_F N_c$, $\{n_h|n_l\}\, C_F^2 N_c$, $\{n_h|n_l\}\, d_{33}$, $\{n_h|n_l\}^2\, C_F N_c$.
4. Resolve integal symmetries, construct integral families. [FEYNSON]
   - $\Rightarrow$ 43 families (28 up to external leg permutation).
   - $\Rightarrow$ 831 master integrals in total.
5. Optimize the selection of master integrals.
   - * Criteria: quasi-finite, d-factorizing, fast to evaluate with pySECDEC.
   - * Allow increasing denominator powers up to 5 dots, and adding dimensional shifts (integrals in $d = 6 - 2\varepsilon$ and $d = 8 - 2\varepsilon$).
6. Generate IBP relations, dimensional recurrence relations. [KIRA]

...

(All of the items here use Alibrary).

Calculation method, II

7. Precompute (“trace”) the IBP solution for each family with *Rational Tracer*.
8. Precompile the *pySecDec* integration library for the amplitude pieces (each color structure is a separate weighted sum of master integrals).
9. *For each point* in the phase space:
   - 9.1 Solve IBP relations using the precomputed trace. [Ratracer; FireFly]
     - ⁎ Each variable set to a rational number.
   - 9.2 Evaluate the amplitudes as weighted sums of masters. [pySecDec]
     - ⁎ The weights are taken from the IBP solution.
   - 9.3 Apply renormalization and pole subtraction.
     [Ferroglia, Neubert, Pecjak, Yang ’09; Bärnreuther, Czakon, Fiedler ’13]
   - 9.4 Save the result.

Solving IBP with Rational Tracer

Linear equation system solving with *Ratracer*: [V.M. '22]

- Write down a system of equations (i.e. in a text file).
- Solve the system *once* using modular artihmetics with variables set to integers, and *record every arithmetic operation* into a file (a *"trace"*).
- Optimize the trace (constant propagation, dead code elimination, etc).
- Use the usual modular function reconstruction methods (i.e. FireFly) by *replaying the trace* many times (with different inputs).

$\Longrightarrow$ Around 10x faster black-box evaluation than Kira on average.

Additional trick:

- Take a trace and *expand it in ε*, save it as a new trace.
  - $\Longrightarrow$ Get the ε expansion of the IBP coefficients directly as the trace outputs.
  - $\Longrightarrow$ Eliminate ε from the list of variables.

$\Longrightarrow$ Overall 3x-4x performance gain for this calculation.

In our case (all variable set to numbers, ε eliminated via expansion):

- No function reconstruction needed! Each output is a rational number.
- Under 2 CPU minutes per point (scales well with threads).
  - Down from $\sim$1 hour on 16 cores with Kira+FireFly!

Amplitude evaluation with pySecDec

pySecDec: library for numerically evaluating Feynman integrals via *sector decomposition* and *Monte Carlo integration*. [Heinrich et al '23, '21, '18, '17]

- Takes a specification for *weighted sum of integrals* (i.e. amplitudes), produces an integration library.
  - One sum per color structure.
  - Integrals sampled adaptively to reach the requested precision of the sums.
  - The 831 masters decomposed into ∼18000 sectors (∼28000 integrals).
- Around 4x-5x speedup in version 1.6 with the new integrator "disteval".
- Integration time to get 0.3% precision for this calculation on a GPU:
  - from *5 minutes in the bulk* of the phase-space,
  - to ∞ near boundaries (e.g. high-energy region) due to growing cancellations between and inside the integrals.

Particularly large cancellations between sunrise and snow-cone integrals in the high energy region:

- The IBP coefficients become $\mathcal{O}(10^{20})$, while the amplitude is $\mathcal{O}(10^{-3})$.
- Knowing the integrals at full double precision (16 digits) is not enough!

⟹ Make pySecDec use *double-double* (32 digits) for integrals that need it.

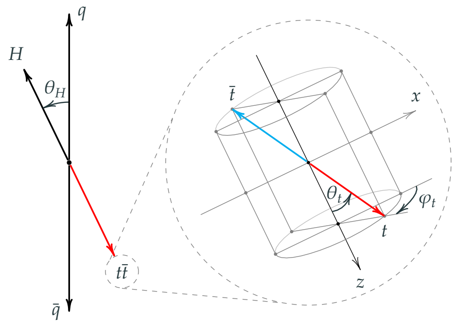
  - Around 20x performance hit on GPU, but 20-digit precision for snow cones reachable.

Phase-space parameters

To parameterize the $q\bar{q} \to t\bar{t}H$ phase-space, instead of

$$s = \left(p_q + p_{\bar{q}}\right)^2 \in \left[\,(2m_t + m_H)^2\,;\,\infty\right],$$
$$s_{t\bar{t}} = \left(p_t + p_{\bar{t}}\right)^2 \in \left[\,(2m_t)^2\,;\,\left(\sqrt{s} - m_H\right)^2 - (2m_t)^2\,\right],$$

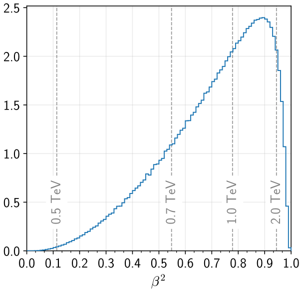
introducing:

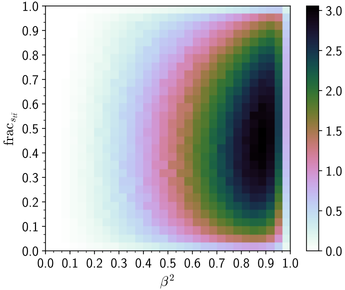


$$\beta^2 \equiv 1 - \frac{s_{min}}{s} \in [0;1],$$
$$\mathrm{frac}_{s_{t\bar{t}}} \equiv \frac{s_{t\bar{t}} - s_{t\bar{t},min}}{s_{t\bar{t},max} - s_{t\bar{t},min}} \in [0;1],$$
$$\theta_H \in [0;\pi],$$
$$\theta_t \in [0;\pi],$$
$$\varphi_t \in [0;2\pi].$$

Which parts of the phase-space are relevant?

Event density at the LHC according to the tree-level amplitude:





To cover 90% of events: $\beta^2 \in [0.34, 0.95]$, that is $\sqrt{s} \in [580\ \text{GeV}, 2.1\ \text{TeV}]$.

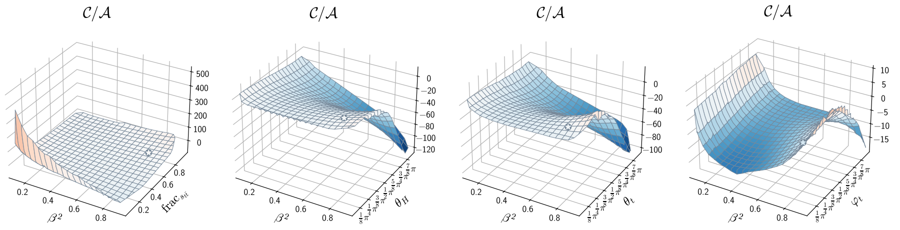
* * *

Example results as two-dimensional slices around the center point of:

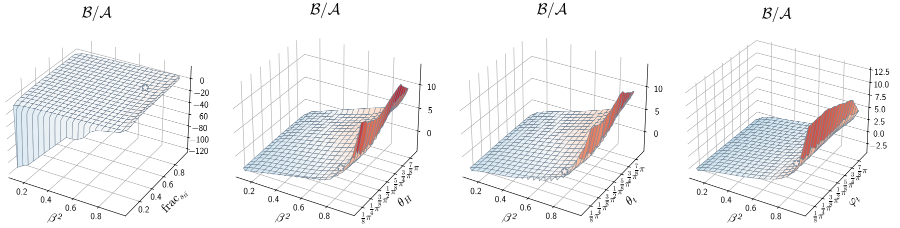
$$\beta^2 = 0.8, \qquad \mathrm{frac}_{s_{t\bar{t}}} = 0.7,$$
$$\cos\theta_H = 0.8, \qquad \cos\theta_t = 0.9, \qquad \cos\varphi_t = 0.7,$$
$$m_H^2 = 12/23\, m_t^2, \qquad \mu = s/2.$$

Resulting slices in β^2 and $\mathrm{frac}_{s_{t\bar{t}}}$, θ_H, θ_t, φ_t

N_f part of the two-loop amplitude (*our result*):

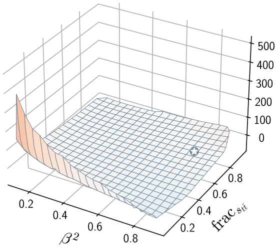


One-loop amplitude (*already known*):

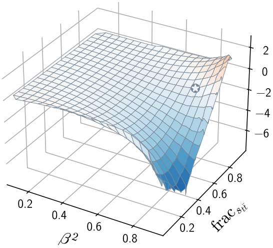


Resulting slices in β^2 and $\mathrm{frac}_{s_{t\bar{t}}}$

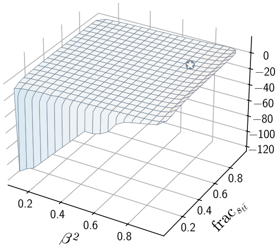
$\mathcal{C}/\mathcal{A}$



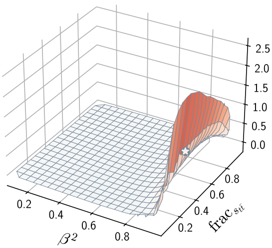
$\mathcal{C} \times (\text{phase-space density}) \times 10^3$



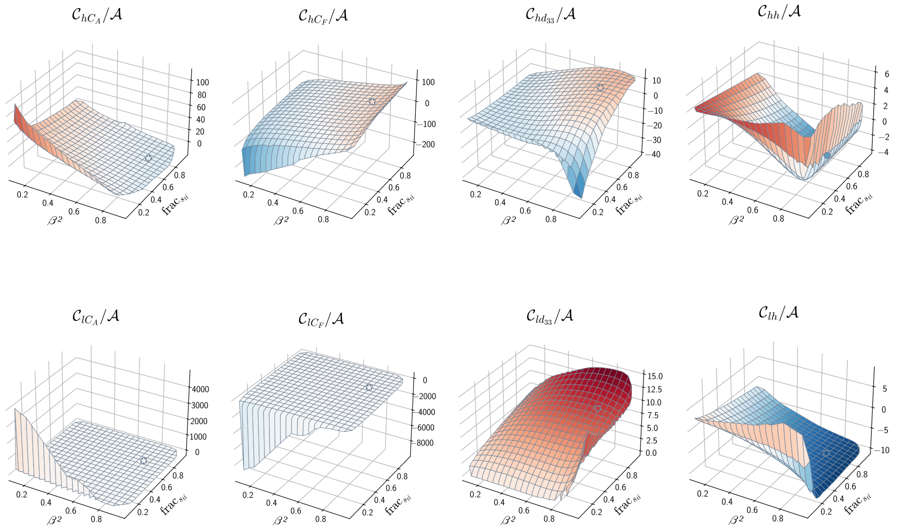
$\mathcal{B}/\mathcal{A}$



$\mathcal{B} \times (\text{phase-space density}) \times 10^3$



Resulting slices in β^2 and $\mathrm{frac}_{s_{t\bar{t}}}$ by color factor



Summary & Outlook

Done:

- N_f-part of the two-loop virtual amplitude for $q\bar{q} \to t\bar{t}H$.
- Subprojects:
  - Peformace and precision improvements in pySecDec.
  - Faster IBP solving with Ratracer.
  - Amplitude generation with Alibrary.

In progress:

- The rest of the two-loop virtual amplitude for $q\bar{q} \to t\bar{t}H$.
- Interpolation for the results.

Todo:

- Full two-loop virtual amplitude for $gg \to t\bar{t}H$.
- Phenomenological applications.

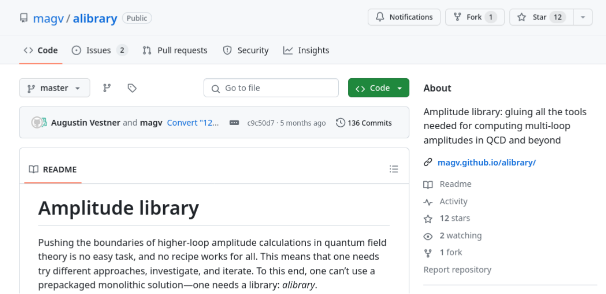
Backup slides

Amplitude library

Most of this work is powered by Alibrary (*“amplitude library”*). It provides functions and interfaces to tools for multiloop calculations in Mathematica:

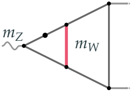
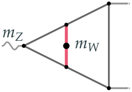
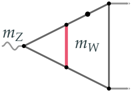
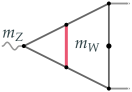
- Diagram generation and visualization (Qgraf, Graphviz, TikZ).
- Feynman rule insertion.
- Tensor trace summation (Form, Color.h).
- Integral symmetries, IBP families (Feynson).
- Export to/from IBP solvers (Kira, Fire+LiteRed).
- Export to/from pySecDec.

github.com/magv/alibrary



On the choice of master integrals

Integration time of similarly looking integrals to 10^{-3} precision:[1]

	orders	t, s		orders	t, s
	$\varepsilon^{-3} \dots \varepsilon^{0}$	27		$\varepsilon^{-2} \dots \varepsilon^{0}$	57
	$\varepsilon^{-2} \dots \varepsilon^{0}$	1230		$\varepsilon^{-2} \dots \varepsilon^{0}$	>9000

Takeaway: for best performance, test the integration speed and adjust the selection of the master integrals.

[1] pySecDec 1.5.3, NVidia A100 GPU.