Galileo Galilei Institute, Florence, Italy

13 September 2023

Updates on Sector Decomposition

Stephen Jones

Assistant Professor, IPPP Durham / Royal Society URF

Durham University

THE ROYAL SOCIETY

Photo by Wolfgang Hasselmann on Unsplash

Based on work with…

Photo by Lisa Biermann

pySecDec 2023: Matthias Kerner, SJ, Gudrun Heinrich, Anton Olsson, Johannes Schlenk, Vitaly Magerya

2305.19768: Numerical Scattering Amplitudes with pySecDec **(= pySecDec v1.6)**

2211.14845: From Landau equations to the Newton polytope
w/ E. Gardi, F. Herzog, Y. Ma, J. Schlenk

2112.09145: Targeting Multi-Loop Integrals with Neural Networks
w/ R. Winterhalder, V. Magerya, E. Villa, M. Kerner, A. Butter, G. Heinrich, T. Plehn

2108.10807: Expansion by regions with pySecDec **(= pySecDec v1.5)**
+ S. Jahn, F. Langer, A. Poldaru, E. Villa

Computing Feynman Integrals

Feynman integrals can be difficult to compute analytically

Various methods to approximate/evaluate them numerically

Numerical differential equations

Series solutions of differential equations (DiffExp, AMFlow, Seasyde)

Mellin-Barnes (MB, Ambre)

Taylor expansion in Feynman parameters (TayInt)

Tropical sampling (Feyntrop)

Numerical Loop-Tree Duality (cLTD, Lotty) →**Talk of Dario**

Sector decomposition (Sector_decomposition, FIESTA, pySecDec)

Sector Decomposition in a Nutshell

$$I \sim \int_{\mathbb{R}^{N+1}_{>0}} [\mathrm{d}\boldsymbol{x}]\, \boldsymbol{x}^{\boldsymbol{\nu}} \frac{[\mathscr{U}(\boldsymbol{x})]^{N-(L+1)D/2}}{[\mathscr{F}(\boldsymbol{x},\mathbf{s}) - i\delta]^{N-LD/2}} \delta(1 - H(\boldsymbol{x}))$$

Singularities

1. UV/IR singularities when some $x \to 0$ simultaneously $\Longrightarrow$ Sector Decomposition
2. Thresholds when $\mathscr{F}$ vanishes inside integration region $\Longrightarrow$ $i\delta$

Sector decomposition

Find a local change of coordinates for each singularity that factorises it (blow-up)

Sector Decomposition in a Nutshell (II)

$$I \sim \int_{\mathbb{R}^N_{>0}} [\mathrm{d}\boldsymbol{x}]\, \boldsymbol{x}^{\boldsymbol{\nu}} \left(c_i\, \boldsymbol{x}^{\mathbf{r}_i}\right)^t$$

$$\mathcal{N}(I) = \mathrm{convHull}(\mathbf{r}_1, \mathbf{r}_2, \ldots) = \bigcap_{f \in F} \left\{ \mathbf{m} \in \mathbb{R}^N \mid \langle \mathbf{m}, \mathbf{n}_f \rangle + a_f \geq 0 \right\}$$

Normal vectors incident to each extremal vertex define a local change of variables*

Kaneko, Ueda 10

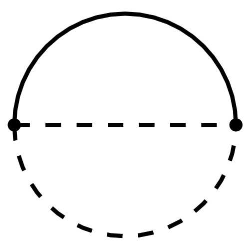
$$x_i = \prod_{f \in S_j} y_f^{\langle \mathbf{n}_f, \mathbf{e}_i \rangle}$$

$$I \sim \sum_{\sigma \in \Delta^T_{\mathcal{N}}} |\sigma| \int_0^1 \left[\mathrm{d}\mathbf{y}_f\right] \underbrace{\prod_{f \in \sigma} y_f^{\langle \mathbf{n}_f, \boldsymbol{\nu} \rangle - t a_f}}_{\text{Singularities}} \Bigg(\underbrace{c_i \prod_{f \in \sigma} y_f^{\langle \mathbf{n}_f, \mathbf{r}_i \rangle + a_f}}_{\text{Finite}} \Bigg)^t$$

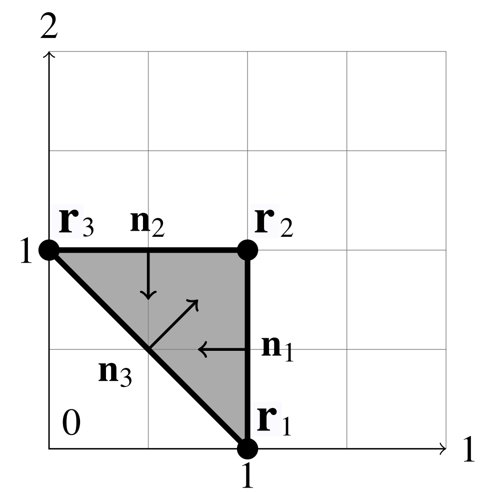
Singularities **Finite**

*If $|S_j| > N$, need triangulation to define variables (simplicial normal cones $\sigma \in \Delta^T_{\mathcal{N}}$)

Sector Decomposition in a Nutshell (III)

$I =$  $= -\Gamma(-1+2\varepsilon)\left(m^2\right)^{1-2\varepsilon} \int_0^\infty \frac{\mathrm{d}x_1 \mathrm{d}x_2}{\left(x_1^1 x_2^0 + x_1^1 x_2^1 + x_1^0 x_2^1\right)^{2-\varepsilon}}.$

$$\mathbf{r}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \mathbf{r}_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mathbf{r}_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$\mathcal{N}(I) =$ 

$$= \quad \mathbf{n}_1 = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \; \mathbf{n}_2 = \begin{pmatrix} 0 \\ -1 \end{pmatrix} \; \mathbf{n}_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$a_1 = 1 \quad a_2 = 1 \quad a_3 = -1$$

For each vertex make the local change of variables

e.g. $\mathbf{r}_1$: $x_1 = y_1^{-1} y_3^1,\ x_2 = y_1^0 y_3^1$, $\mathbf{r}_2 : x_1 = y_1^{-1} y_2^0,\ x_2 = y_1^0 y_2^{-1}$, $\mathbf{r}_3 : x_1 = y_2^0 y_3^1, x_2 = y_2^{-1} y_3^1$

$$I = -\Gamma(-1+2\varepsilon)\left(m^2\right)^{1-2\varepsilon} \int_0^1 \mathrm{d}y_1 \mathrm{d}y_2 \mathrm{d}y_3 \frac{y_1^{-\varepsilon} y_2^{-\varepsilon} y_3^{-1+\varepsilon}}{(y_1+y_2+y_3)^{2-\varepsilon}} \left[\delta(1-y_2)+\delta(1-y_3)+\delta(1-y_1)\right]$$

Schlenk 2016

Challenges and Opportunities

Frontiers

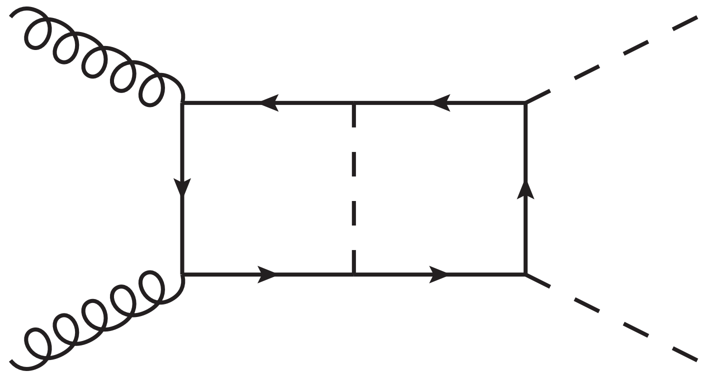
* $2 \to 2$ **@ 2-loop** : fine (e.g. HH, HJ, ZZ, ZH)

  + masses (e.g. EW corrections) - suitable

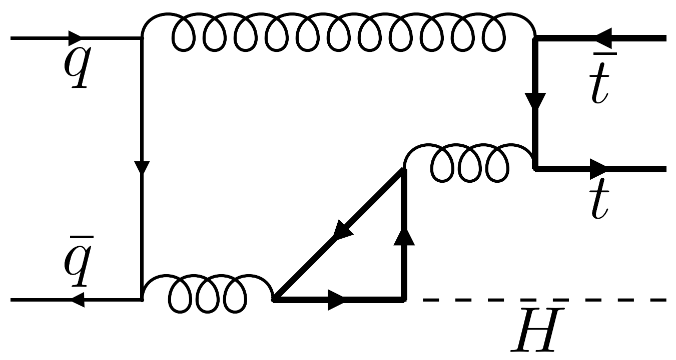
  + large hierarchies (e.g. small m_b , large s, thresholds)

* $2 \to 3$ **@ 2-loop** : challenging (high dim phase-space)

* **3-loop+** : suitable, less explored



WIP: Gudrun Heinrich, SJ, Matthias Kerner, Tom Stone, Augustin Vestner



Opportunities

1. Improvements in algorithm & implementation
2. Smarter numerical integration routines
3. Improved contour deformation
4. Expansions

1. Algorithmic Improvements

Performance Improvements

v1.5: Adaptive sampling of sectors, automatic contour def. adjustment

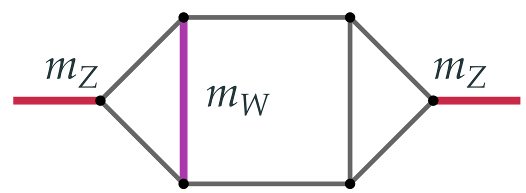
v1.5.6: Optimisations in integrand code

v1.6: New Quasi-Monte Carlo integrator ``Disteval"

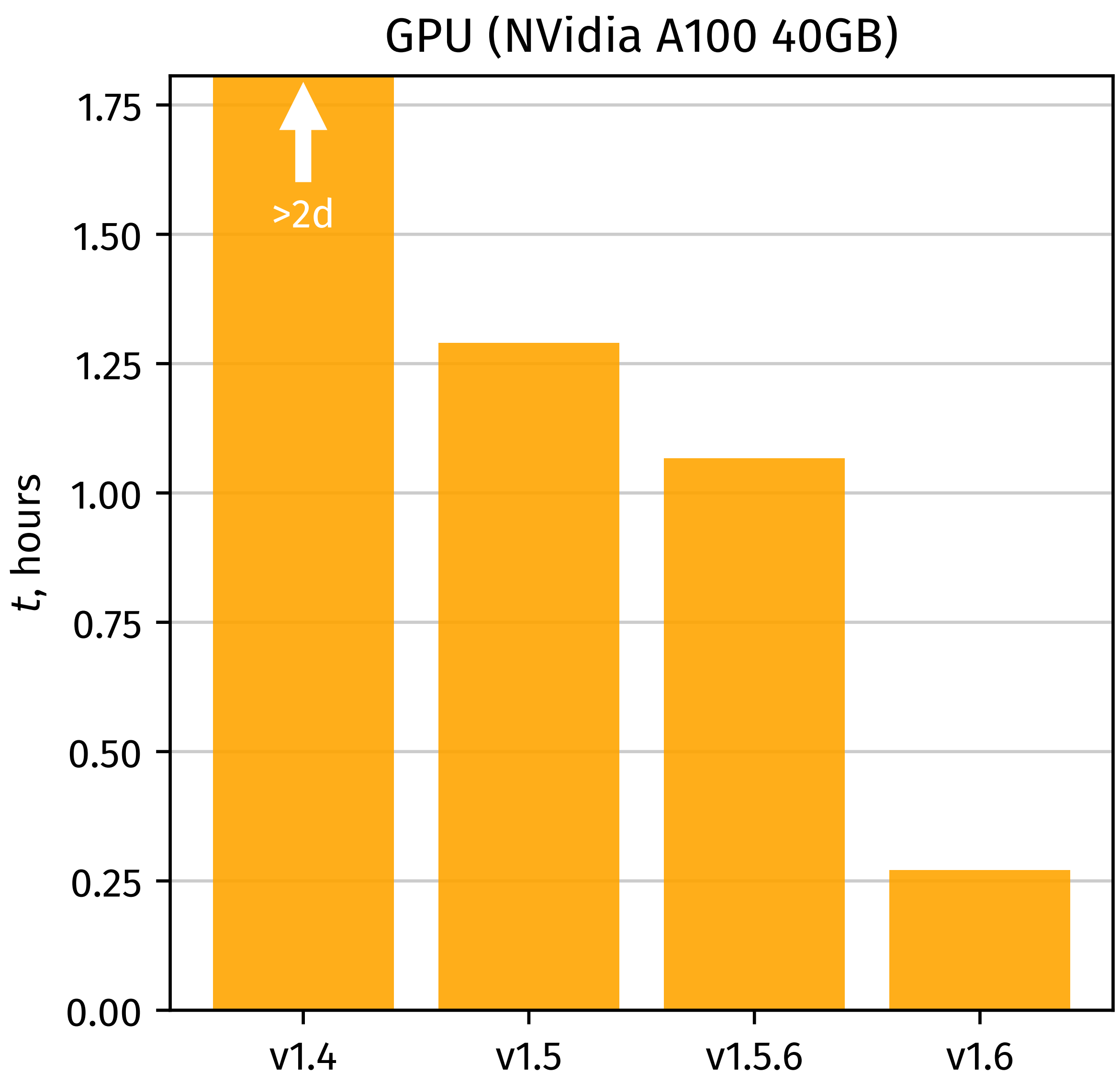
- Faster implementation of old integrator ``IntLib"
- CPU & GPU: fusion of integration/integrand code (less modular arithmetic)
- CPU: better utilisation via SIMD instructions (AVX2, FMA)
- GPU: sum result on GPU, less synchronisation
- Parse amplitude coefficients w/GiNaC (supports e.g. partial fractioned input)
- Workers can run on remote machines (via ssh)

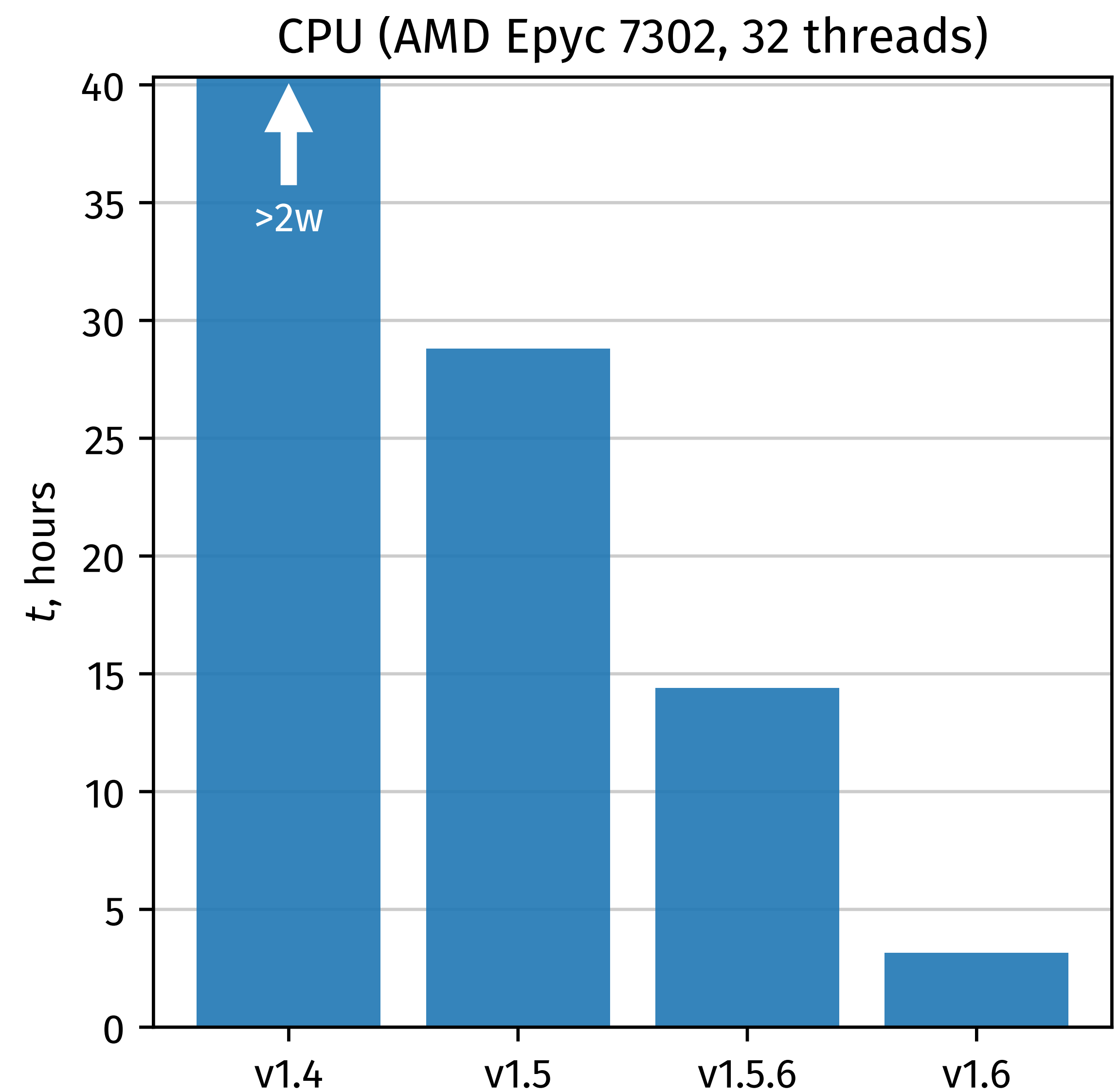
Does it help?

Performance Improvements (II)



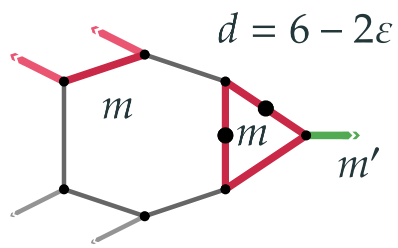
7 digits

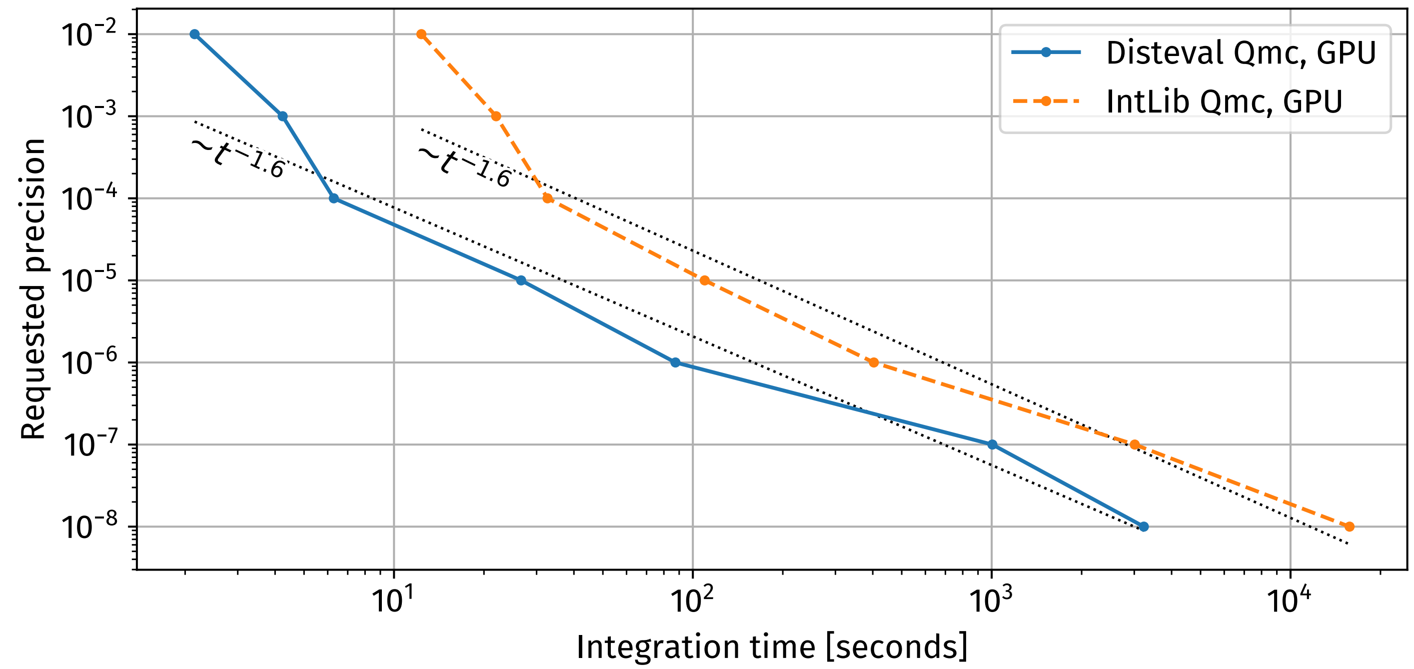




Vitaly Magerya (Radcor 2023)

Profiling (I)





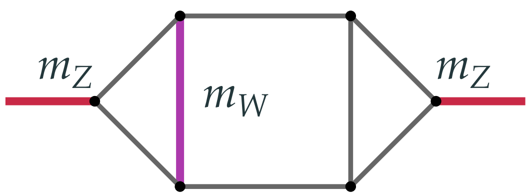
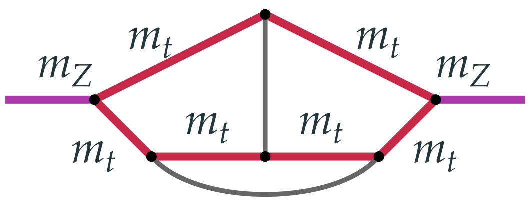
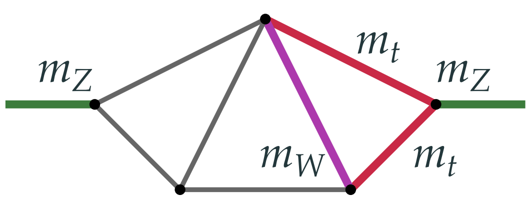
Integrator \ Accuracy		10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
GPU	Disteval	4.2 s	6.3 s	27 s	1.5 m	17 m	54 m
	IntLib	22.0 s	22.0 s	110 s	6.7 m	50 m	263 m
	Speedup	5.2	5.2	4.1	5.6	3.0	4.9
CPU	Disteval	5.1 s	14 s	1.6 m	8.3 m	57 m	4.7 h
	IntLib	20.8 s	86 s	14.2 m	62.2 m	480 m	43.1 h
	Speedup	4.1	6.1	8.7	7.5	8.4	9.2

[GPU: NVidia A100 40GB; CPU: AMD Epyc 7F32 with 32 threads]

Vitaly Magerya (Radcor 2023)

Profiling (II)

pySecDec Disteval *integration times* for 3-loop self-energy integrals:[3]

Diagram \ Relative precision		10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
	GPU	15s	20s	40s	200s	13m	50m
	CPU	10s	50s	400s	4000s	180m	1200m
	GPU	18s	19s	30s	20s	1.2m	2m
	CPU	5s	14s	60s	50s	12m	16m
	GPU	6s	11s	12s	30s	3m	24m
	CPU	5s	10s	50s	800s	60m	800m

[Same diagrams as in Dubovyk, Usovitsch, Grzanka '21]

In short: *seconds to minutes per integral* to achieve practical precision.

[GPU: NVidia A100 40GB; CPU: AMD Epyc 7F32 with 32 threads]

2. Integration: Median Lattice Rules

Quasi-Monte Carlo

Li, Wang, Yan, Zhao 15; de Doncker, Almulihi, Yuasa 17, 18; de Doncker, Almulihi 17;
Kato, de Doncker, Ishikawa, Yuasa 18

$$Q_n^{(k)}[f] \equiv \frac{1}{n}\sum_{i=0}^{n-1} f\left(\left\{\frac{i\mathbf{z}}{n} + \mathbf{\Delta}_k\right\}\right) \qquad I[f] \approx \bar{Q}_{n,m}[f] \equiv \frac{1}{m}\sum_{k=0}^{m-1} Q_n^{(k)}[f],$$

{ } - Fractional part

Δ_k - Random shift vector

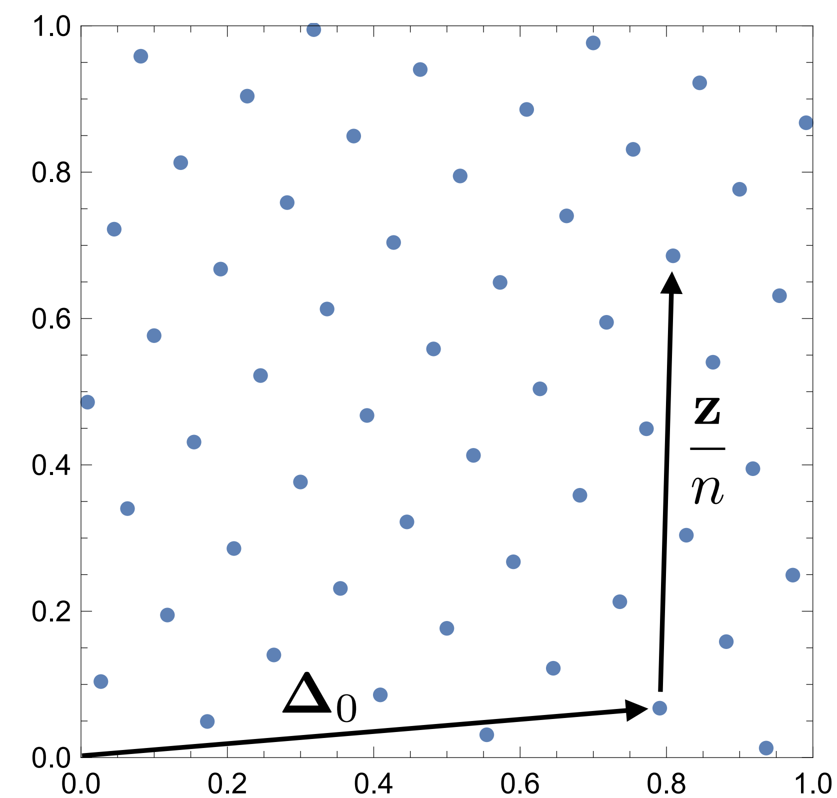
$\mathbf{z}$ - Generating vector

Previously:

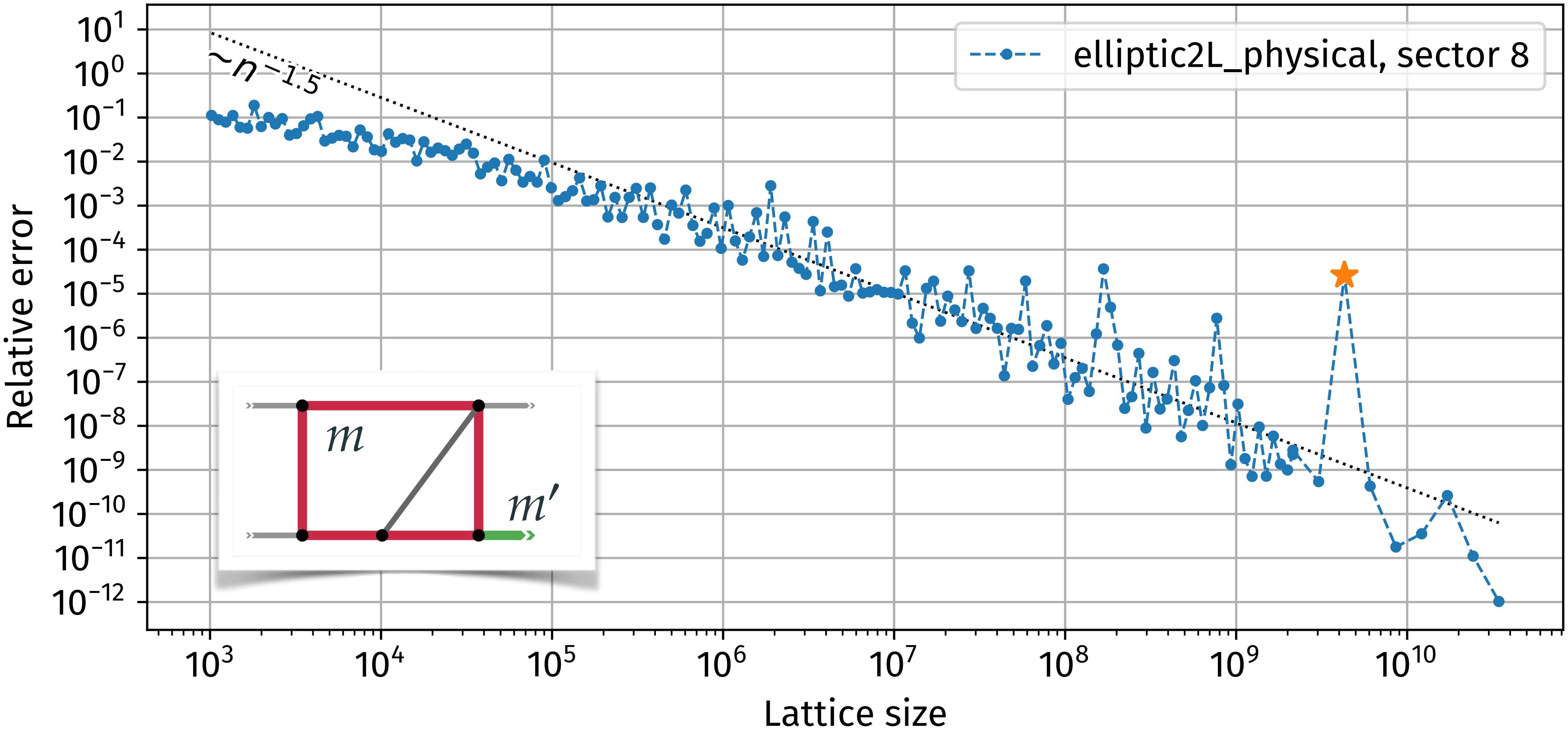
Precompute $\mathbf{z}$ with (CBC) construction
Nuyens, Cools 06

Guarantee error $\sim 1/n^{\alpha}$ if $\delta_x^{(\alpha)} I(\mathbf{x})$ is square-integrable and periodic Dick, Kuo, Sloan 13

CBC needs $\mathcal{O}(n)$ bytes memory $n \lesssim 4.10^{10}$ @ 2TB
Can encounter ``unlucky'' lattices



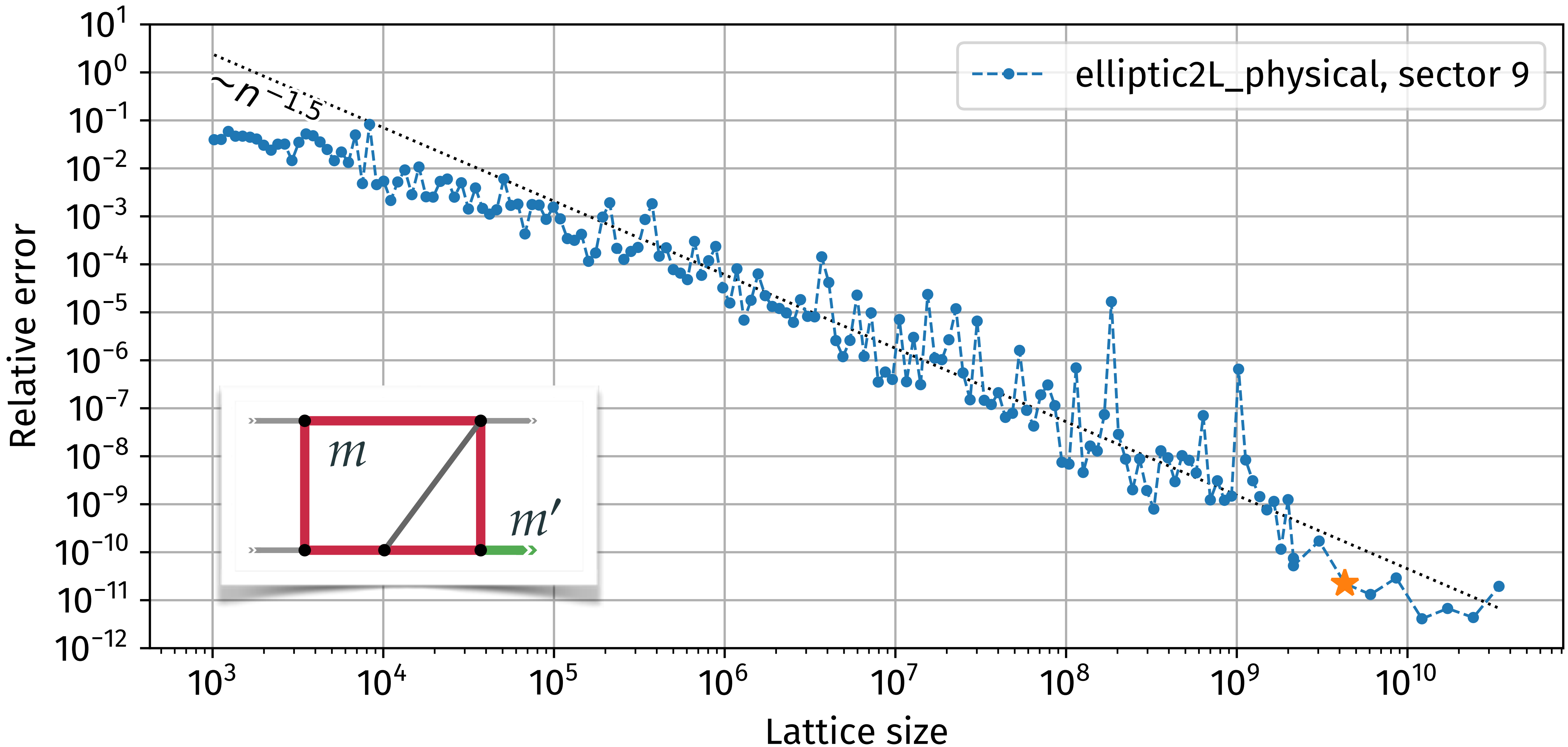
Quasi-Monte Carlo: Unlucky Lattices



Good: Asymptotic error scaling $\sim 1/n^{1.5}$

Bad: Huge drop in precision for some "unlucky" lattices
Not consistent across integrands

Quasi-Monte Carlo: Unlucky Lattices (II)



Good: Asymptotic error scaling $\sim 1/n^{1.5}$

Bad: Huge drop in precision for some "unlucky" lattices

Not consistent across integrands

Median Lattice Rules

Instead:

Compute $\mathbf{z}$ on-the-fly

1. Choose R random $\mathbf{z} \in \mathrm{Uniform}(0; N-1)$
2. Estimate integral on each lattice
3. Choose lattice with median integral value

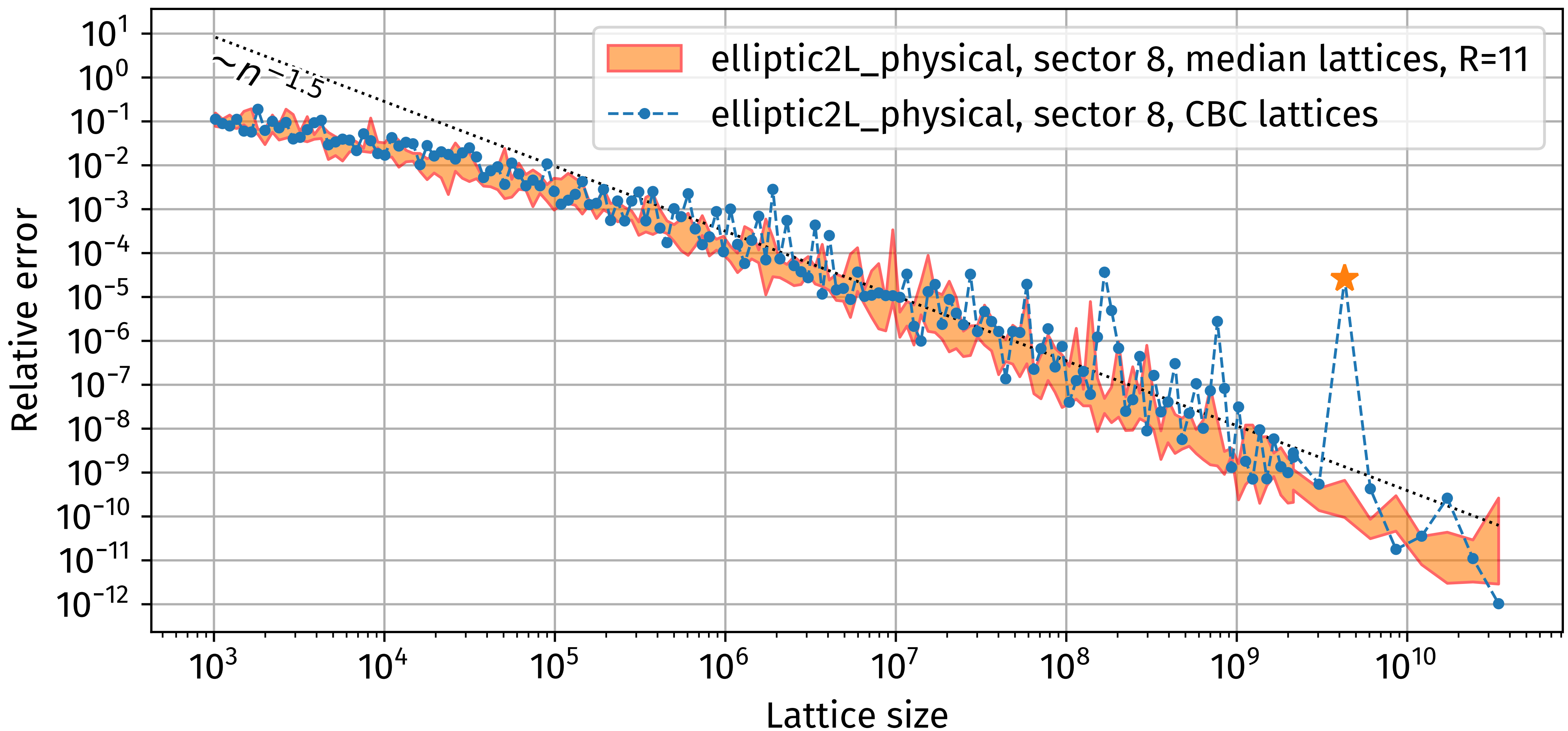
If $\delta_x^{(\alpha)} I(\mathbf{x})$ is square-integrable and periodic

Integration error: $C(\alpha, \varepsilon)/(\rho n)^{\alpha - \epsilon}$

With probability: $1 - \rho^{R+1/2}/4$

$\forall \;\; 0 < \varepsilon \;\&\; 0 < \rho < 1$

Goda, L'Ecuyer 22

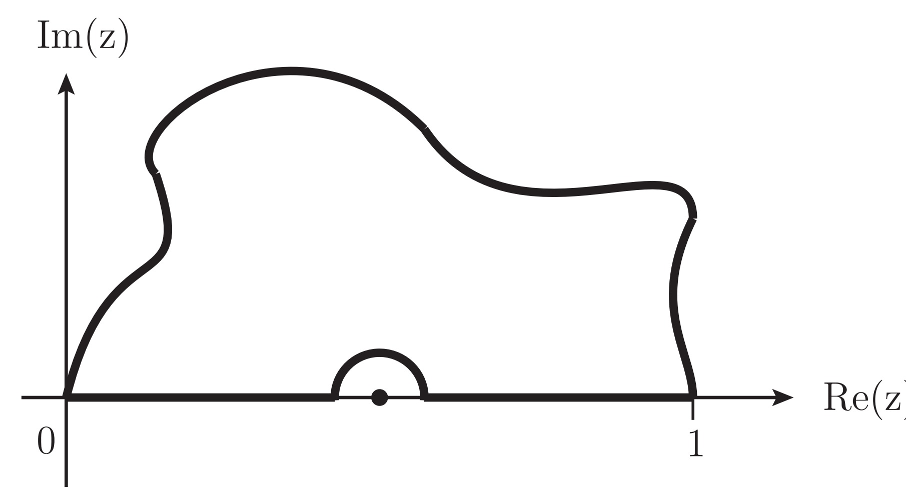


3. Contour Deformation

3. Neural Networks for Contour Deformation

Feynman integral (multi-loop/leg):

$$I \sim \int_0^1 [\mathrm{d}\boldsymbol{x}]\,\boldsymbol{x}^{\nu} \frac{[\mathcal{U}(\boldsymbol{x})]^{N-(L+1)D/2}}{[\mathcal{F}(\boldsymbol{x},\mathbf{s})]^{N-LD/2}}$$



Must deform contour to avoid poles on real axis

Feynman prescription $\mathcal{F} \to \mathcal{F} - i\delta$ tells us how to do this

Expand $\mathcal{F}(z = \boldsymbol{x} - i\boldsymbol{\tau})$ around $\boldsymbol{x}$: $\mathcal{F}(z) = \mathcal{F}(\boldsymbol{x}) - i\sum_j \tau_j \frac{\partial \mathcal{F}(\boldsymbol{x})}{\partial x_j} + \mathcal{O}(\tau^2)$

Old Method

$\tau_j = \lambda_j\, x_j(1 - x_j)\frac{\partial \mathcal{F}(\boldsymbol{x})}{\partial x_j}$ with small constants $\lambda_j > 0$

Soper 99; Binoth, Guillet, Heinrich, Pilon, Schubert 05; Nagy, Soper 06; Anastasiou, Beerli, Daleo 07; Beerli 08; Borowka, Carter, Heinrich 12; Borowka 14;…

New Method

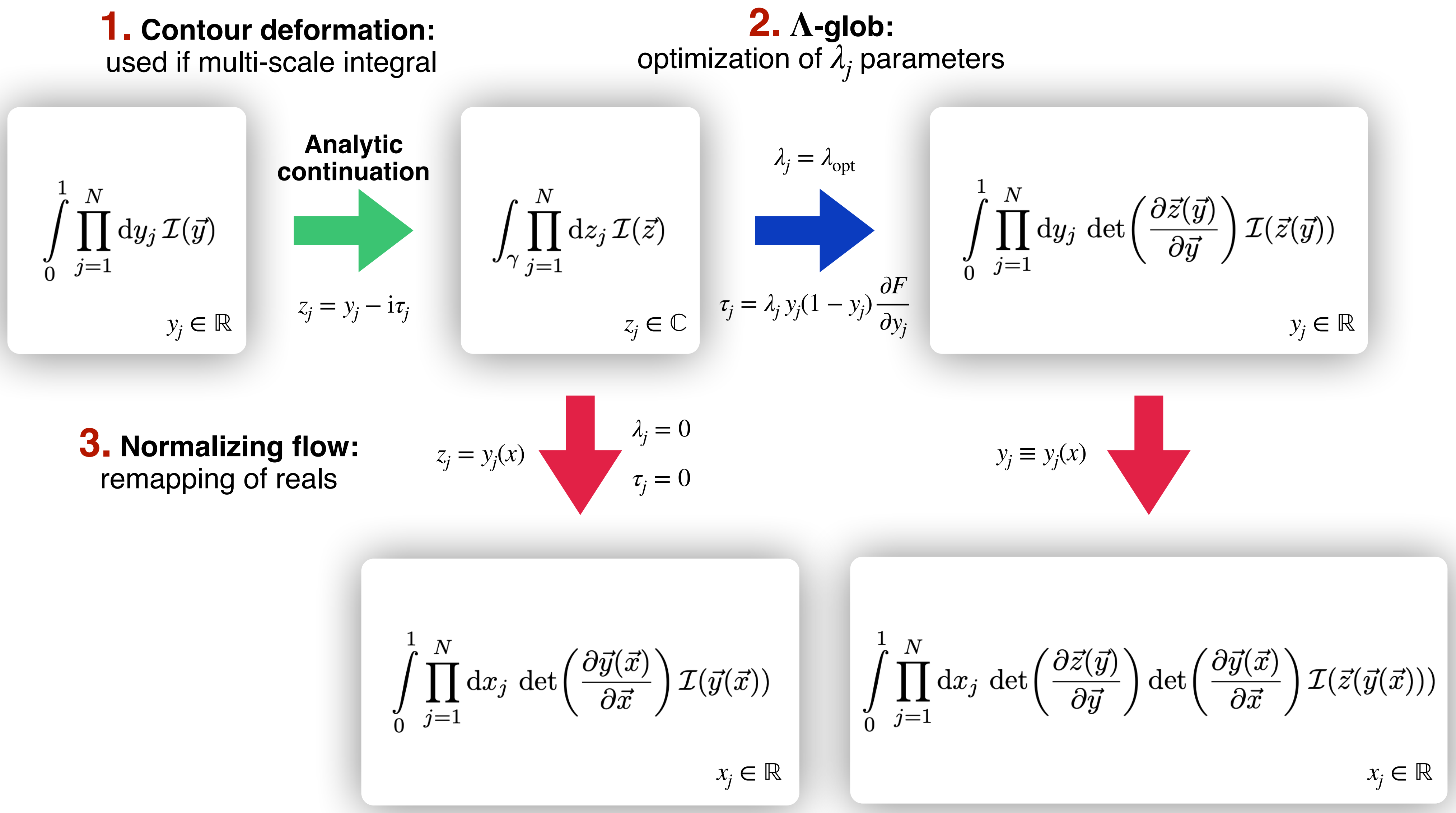
Generalise $\lambda_j \to \lambda_j(\boldsymbol{x})$ and use Neural Network (Normalizing Flows) to pick contour

Winterhalder, Magerya, Villa, SJ, Kerner, Butter, Heinrich, Plehn 22

3. Neural Networks for Contour Deformation (II)

Normalizing Flows consist of a series of (trainable) bijective mappings for which we can efficiently compute the Jacobian

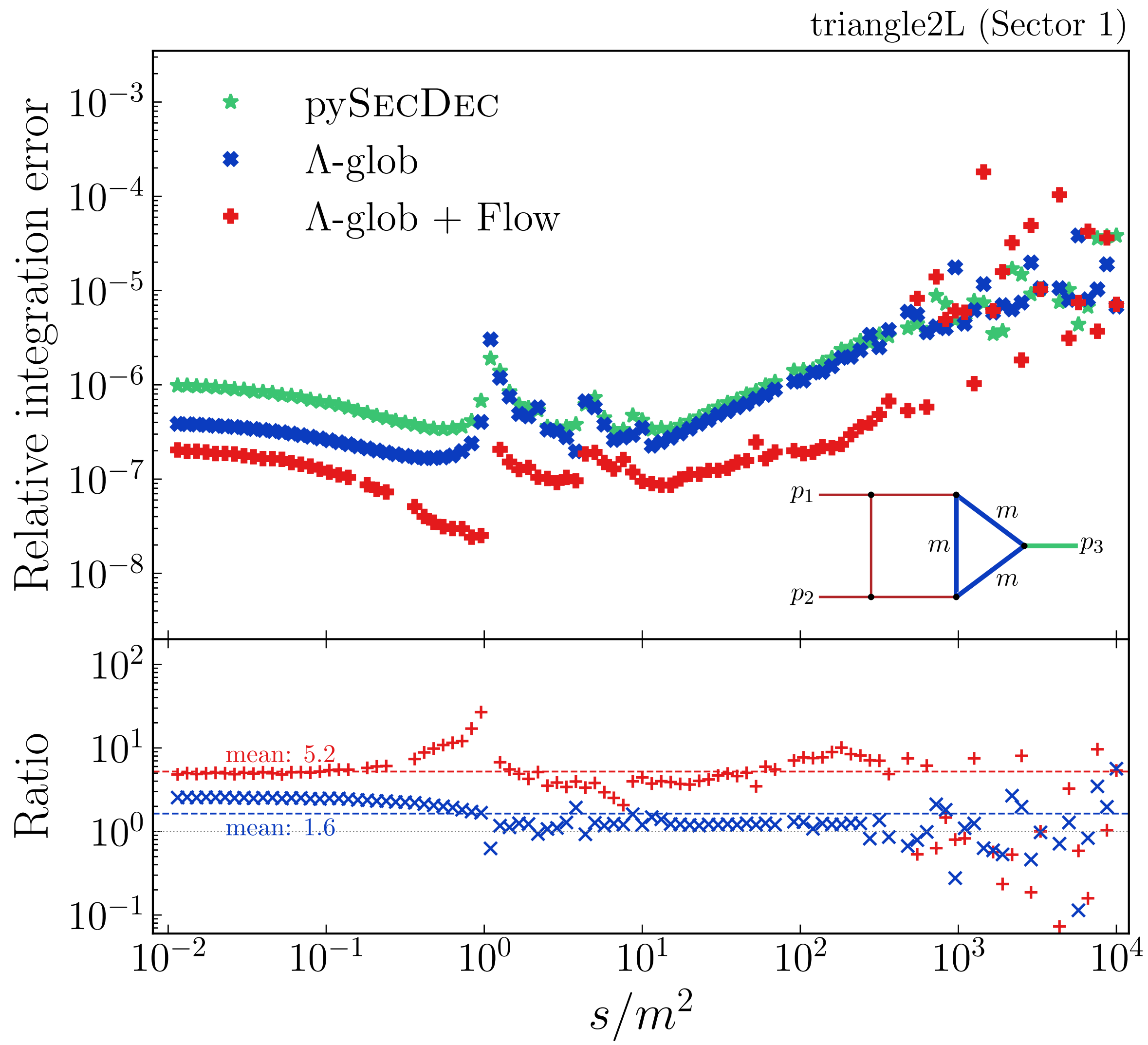
Procedure

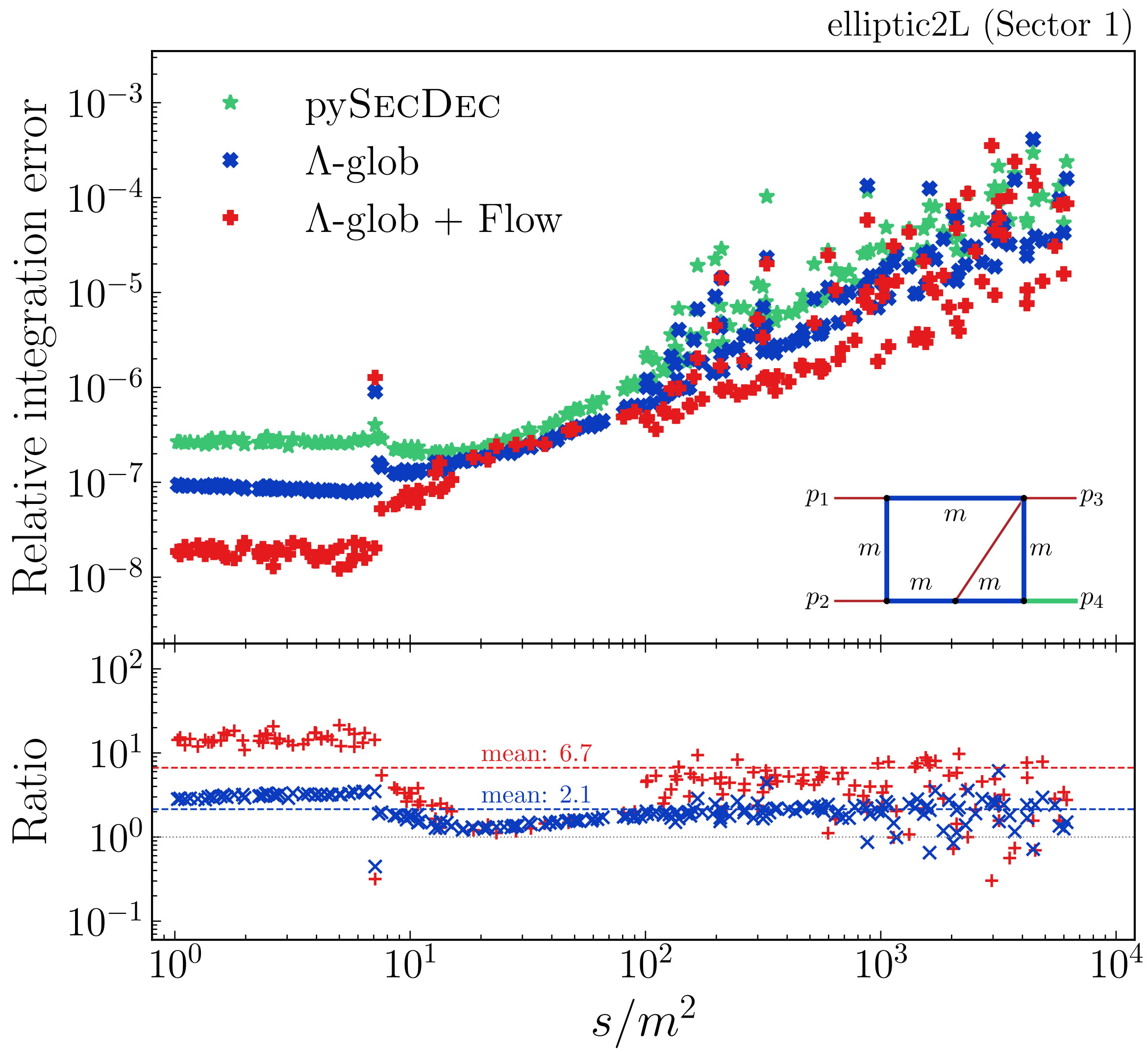


Loss: $L = L_{\mathrm{MC}} + L_{\mathrm{sign}}$ constructed to minimise variance without crossing poles

3. Neural Networks for Contour Deformation (III)

Applied to several 1 & 2-loop Feynman Integrals with multiple masses/thresholds using tensorflow





Proof of principle that Machine Learning can help to find improved contours and reduce variance, still a tradeoff between training time/ integrating time

4. Expansions: Method of Regions

Method of Regions

Consider expanding an integral about some limit:
$p_i^2 \sim \lambda Q^2$, $p_i \cdot p_j \to \lambda Q^2$ or $m^2 \sim \lambda Q^2$ for $\lambda \to 0$

Issue: integration and series expansion do not necessarily commute

Method of Regions

$$I(\mathbf{s}) = \sum_R I^{(R)}(\mathbf{s}) = \sum_R T_{\mathbf{t}}^{(R)} I(\mathbf{s})$$

1. Split integrand up into regions (R)
2. Series expand each region in λ
3. Integrate each expansion over the whole integration domain
4. Discard scaleless integrals (= 0 in dimensional regularisation)
5. Sum over all regions

Smirnov 91; Beneke, Smirnov 97; Smirnov, Rakhmetov 99; Pak, Smirnov 11; Jantzen 2011; …

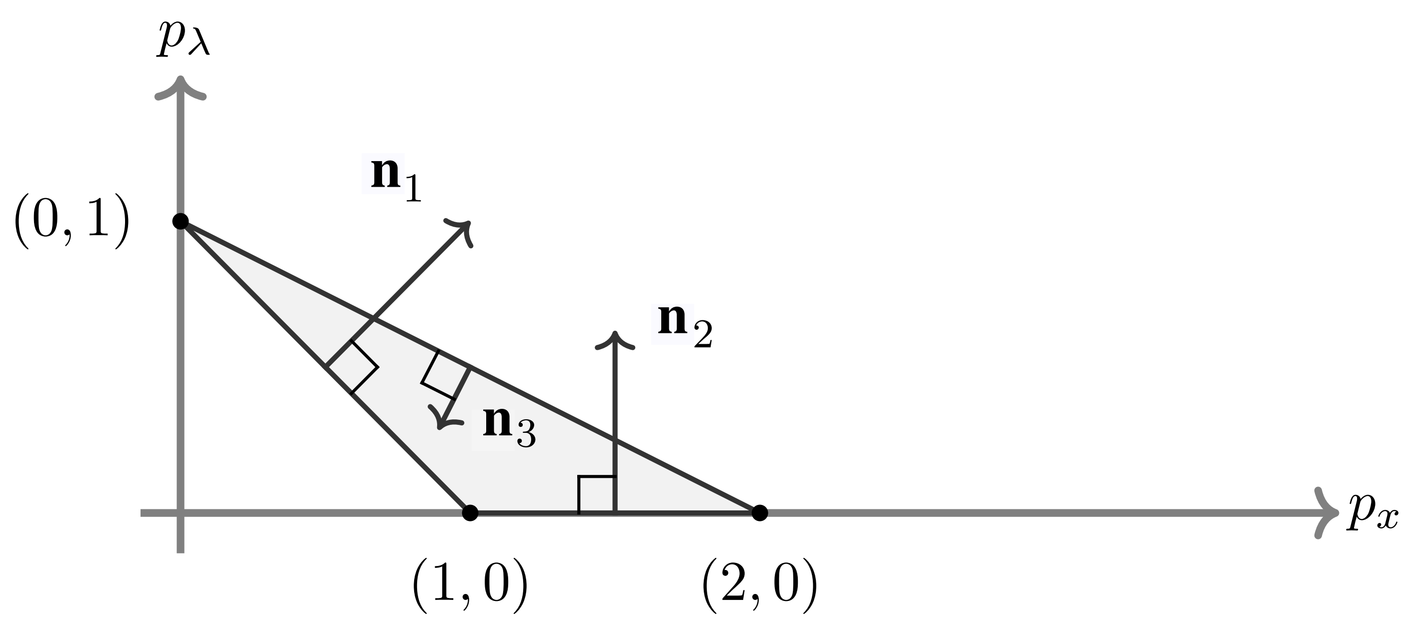
Finding Regions

$$I \sim \int_{\mathbb{R}^N_{>0}} [\mathrm{d}\boldsymbol{x}]\, \boldsymbol{x}^{\boldsymbol{\nu}} \left(c_i\, \boldsymbol{x}^{\mathbf{r}_i}\right)^t \to \int_{\mathbb{R}^N_{>0}} [\mathrm{d}\boldsymbol{x}]\, \boldsymbol{x}^{\boldsymbol{\nu}} \left(c_i\, \boldsymbol{x}^{\mathbf{r}_i} \lambda^{r_{i,N+1}}\right)^t \to \mathcal{N}^{N+1}$$

Normal vectors w/ positive λ component define change of variables $\mathbf{n}_f = (v_1, \ldots, v_N, 1)$

$$\boldsymbol{x} = \boldsymbol{\lambda}^{\mathbf{n}_f}\, \mathbf{y}\,, \qquad \lambda \to \lambda$$

Pak, Smirnov 10; Semenova, A. Smirnov, V. Smirnov 18



$1,2 \in F^+$

$3 \notin F^+$

Original integral I may then be approximated as $I = \sum_{f \in F^+} I^{(f)} + \ldots$

Additional Regulators/ Rapidity Divergences

MoR subdivides $\mathcal{N}(I) \to \{\mathcal{N}(I^R)\} \implies$ new (internal) facets $F^{\text{int.}}$

New facets can introduce spurious singularities not regulated by dim reg

Lee Pomeransky Representation:

$$\mathcal{N}(I^{(R)}) = \bigcap_{f \in F} \left\{ \mathbf{m} \in \mathbb{R}^N \mid \langle \mathbf{m}, \mathbf{n}_f \rangle + a_f \geq 0 \right\}$$

$$I \sim \sum_{\sigma \in \Delta^T_{\mathcal{N}}} |\sigma| \int_{\mathbb{R}^N_{>0}} \left[\mathrm{d}\mathbf{y}_f \right] \prod_{f \in \sigma} y_f^{\langle \mathbf{n}_f, \boldsymbol{\nu} \rangle + \frac{D}{2} a_f} \left(c_i \prod_{f \in \sigma} y_f^{\langle \mathbf{n}_f, \mathbf{r}_i \rangle + a_f} \right)^{-\frac{D}{2}}$$

If $f \in F^{\text{int}}$ have $a_f = 0$ need analytic regulators $\boldsymbol{\nu} \to \boldsymbol{\nu} + \boldsymbol{\delta\nu}$

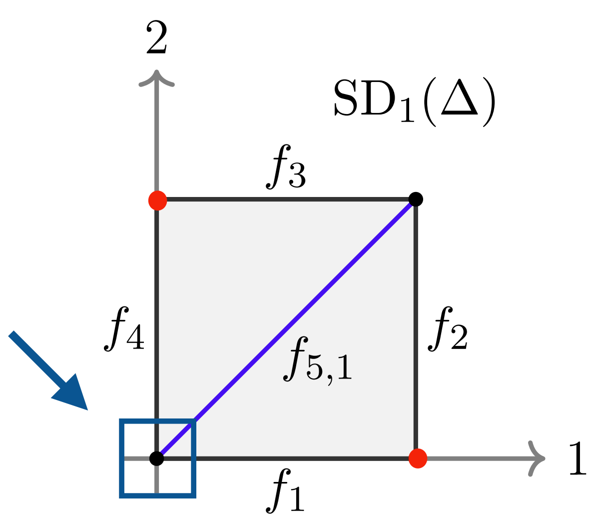
Heinrich, Jahn, SJ, Kerner, Langer, Magerya, Põldaru, Schlenk, Villa 21; Schlenk 16

Additional Regulators/ Rapidity Divergences

Toy Example:

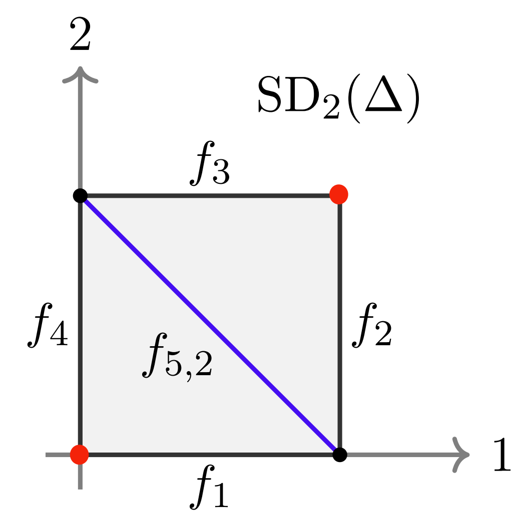
$$P_1(x, \lambda) = 1 + \lambda x_1 + x_1 x_2 + \lambda x_2$$

need analytic regulator



$$P_2(x, \lambda) = \lambda + x_1 + \lambda x_1 x_2 + x_2$$

ok!



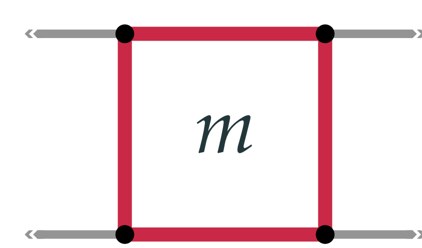
pySecDec can find the constraints on the analytic regulators for you

`extra_regulator_constraints()`:

$$v_2 - v_4 \neq 0, \quad v_1 - v_3 \neq 0$$

`suggested_extra_regulator_exponent()`:

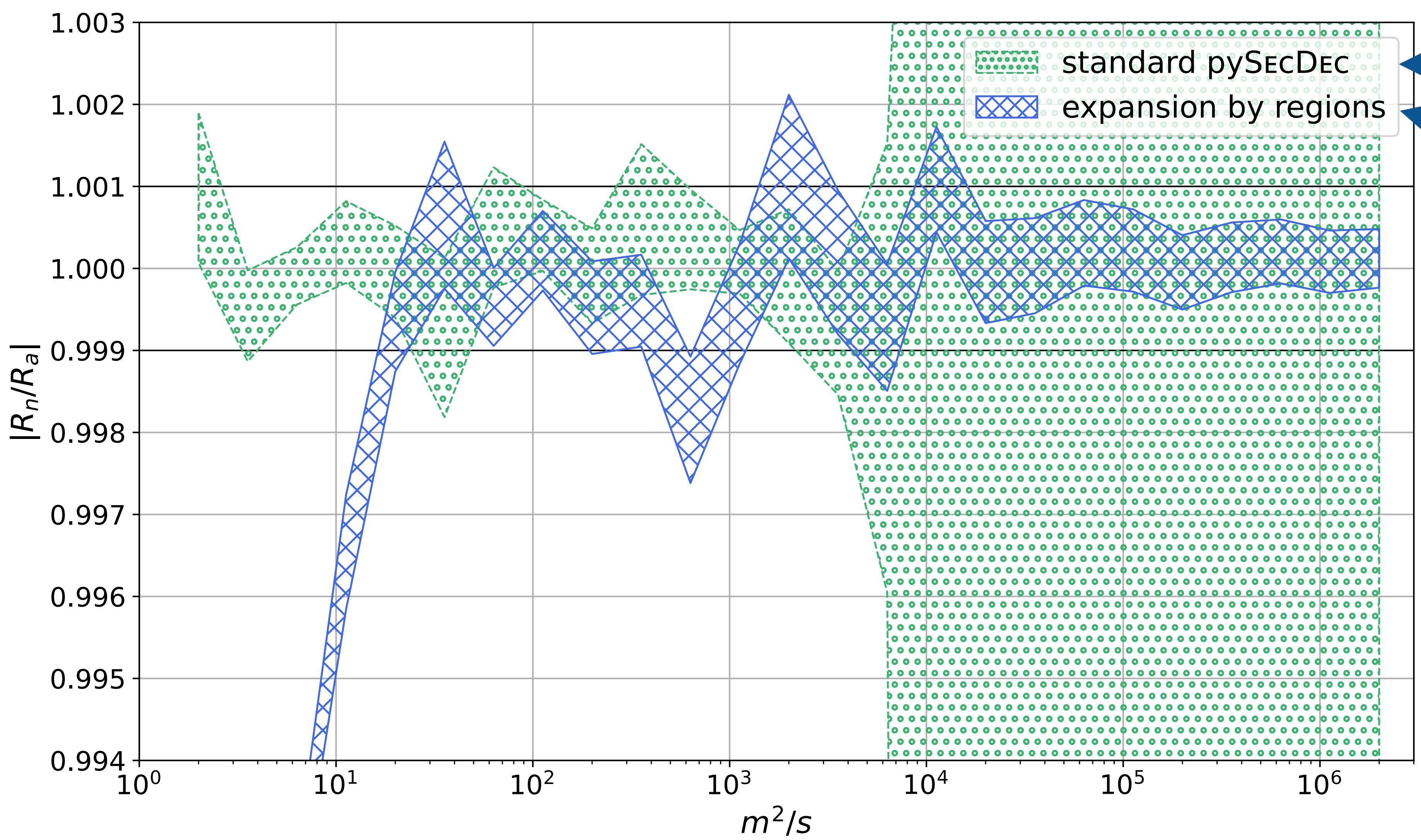
$$\{\delta\nu_1, \delta\nu_2, \delta\nu_3, \delta\nu_4\} = \{0,0,\eta, -\eta\}$$



Small m expansion

Applying Expansion by Regions

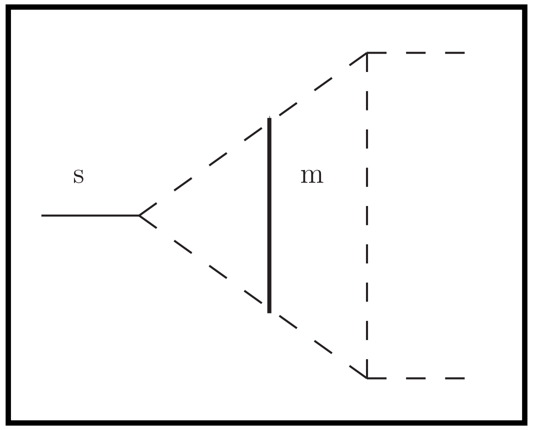
Ratio of the finite $\mathcal{O}(\epsilon^0)$ piece of numerical result R_n to the analytic result R_a



(at large m^2/s)

~1 day

~2 min



For large ratio of scales (m^2/s) the EBR result is **faster** & **easier** to integrate

Building Bridges: LP ↔ Propagator Scaling

Region vectors in momentum space and Lee-Pomeransky space are related, we can see this using Schwinger parameters $\tilde{x}_e$

$$\frac{1}{D_n^{\nu_e}} = \frac{1}{\Gamma(\nu_e)} \int_0^\infty \frac{\mathrm{d}\tilde{x}_e}{\tilde{x}_e}\, \tilde{x}_e^{\nu_e}\, e^{-\tilde{x}_e D_e} \text{, with } x_e \propto \tilde{x}_e$$

$$(D_1^{-1}, \ldots, D_N^{-1}) \sim (\tilde{x}_1, \ldots, \tilde{x}_N) \sim (x_1, \ldots, x_N)$$

Example: 1-loop form factor

$$\begin{aligned}
\text{Hard}: &\quad (D_1^{-1}, D_2^{-1}, D_3^{-1}) \sim (\lambda^0, \lambda^0, \lambda^0), &\quad (x_1, x_2, x_3) \sim (\lambda^0, \lambda^0, \lambda^0)\\
\text{Collinear to } \mathrm{p}_1: &\quad (D_1^{-1}, D_2^{-1}, D_3^{-1}) \sim (\lambda^{-1}, \lambda^0, \lambda^{-1}), &\quad (x_1, x_2, x_3) \sim (\lambda^{-1}, \lambda^0, \lambda^{-1})\\
\text{Collinear to } \mathrm{p}_2: &\quad (D_1^{-1}, D_2^{-1}, D_3^{-1}) \sim (\lambda^0, \lambda^{-1}, \lambda^{-1}), &\quad (x_1, x_2, x_3) \sim (\lambda^0, \lambda^{-1}, \lambda^{-1})\\
\text{Soft}: &\quad (D_1^{-1}, D_2^{-1}, D_3^{-1}) \sim (\lambda^{-1}, \lambda^{-1}, \lambda^{-2}), &\quad (x_1, x_2, x_3) \sim (\lambda^{-1}, \lambda^{-1}, \lambda^{-2})
\end{aligned}$$

Can connect the regions in mom. space with those we determine geometrically

Next step: automatically find (Sudakov decomposed) loop momentum scalings compatible with region vectors WIP w/ Yannick Ulrich

Building Bridges: Landau $\leftrightarrow$ Regions

The **Landau equations** give the necessary conditions for an integral to diverge

$$1)\quad \alpha_e l_e^2(k,p,q) = 0 \qquad \forall e \in G$$

$$2)\quad \frac{\partial}{\partial k_a^\mu}\mathscr{D}(k,p,q;\alpha) = \frac{\partial}{\partial k_a^\mu}\sum_{e\in G}\alpha_e\left(-l_e^2(k,p,q) - i\varepsilon\right) = 0 \qquad \forall a \in \{1,\ldots,L\}$$

Solutions are *pinched surfaces* of the integral where IR divergences may arise

Idea is to explore the *neighbourhood of a pinched surface*, defined by

$$1)\quad \alpha_e l_e^2(k,p,q) \sim \lambda^p \qquad \forall e \in G, \quad \text{with} \quad p \in \{1,2\}$$

$$2)\quad \frac{\partial}{\partial k_a^\mu}\mathscr{D}(k,p,q;\alpha) \lesssim \lambda^{1/2} \qquad \forall a \in \{1,\ldots,L\}$$

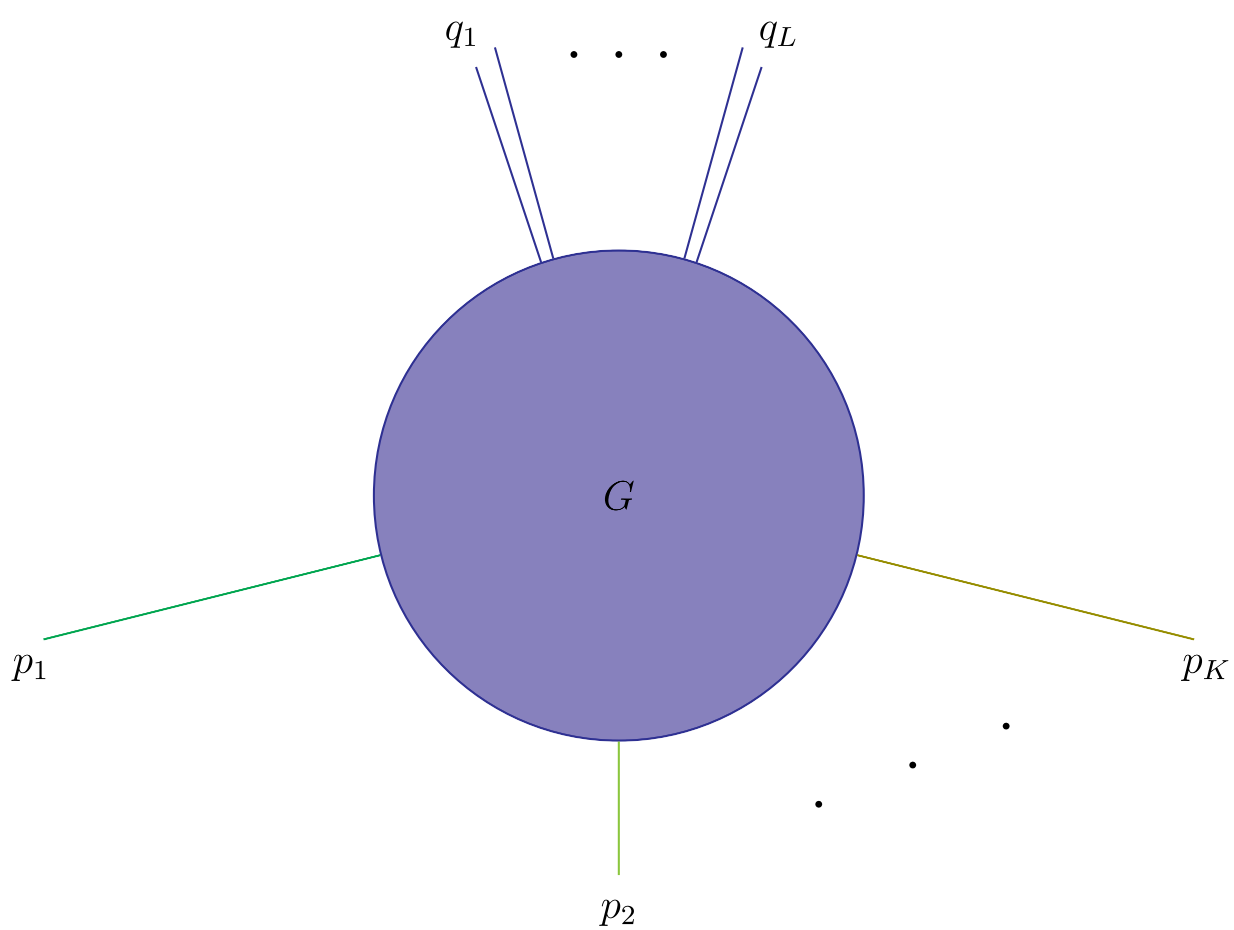
with the goal of further understanding the connection between

Solutions of the Landau equations $\leftrightarrow$ Regions

Gardi, Herzog, Ma, Schlenk 22

On-Shell Expansion

Consider an arbitrary loop, $(K+L)$-leg wide-angle scattering graph



on-shell: $p_i^2 \sim \lambda Q^2 \quad (i = 1, \ldots, K)$,

off-shell: $q_j^2 \sim Q^2 \quad (j = 1, \ldots, L)$,

wide-angle: $p_k \cdot p_l \sim Q^2 \quad (k \neq l)$.

Assuming only hard, collinear & soft modes in momentum space:

What can we say about the regions?

On-Shell Expansion

Using MoR we find:

$$\mathbf{v}_R = (u_{R,1}, u_{R,2}, \ldots, u_{R,N}; 1), \qquad u_{R,e} \in \{0, -1, -2\},$$

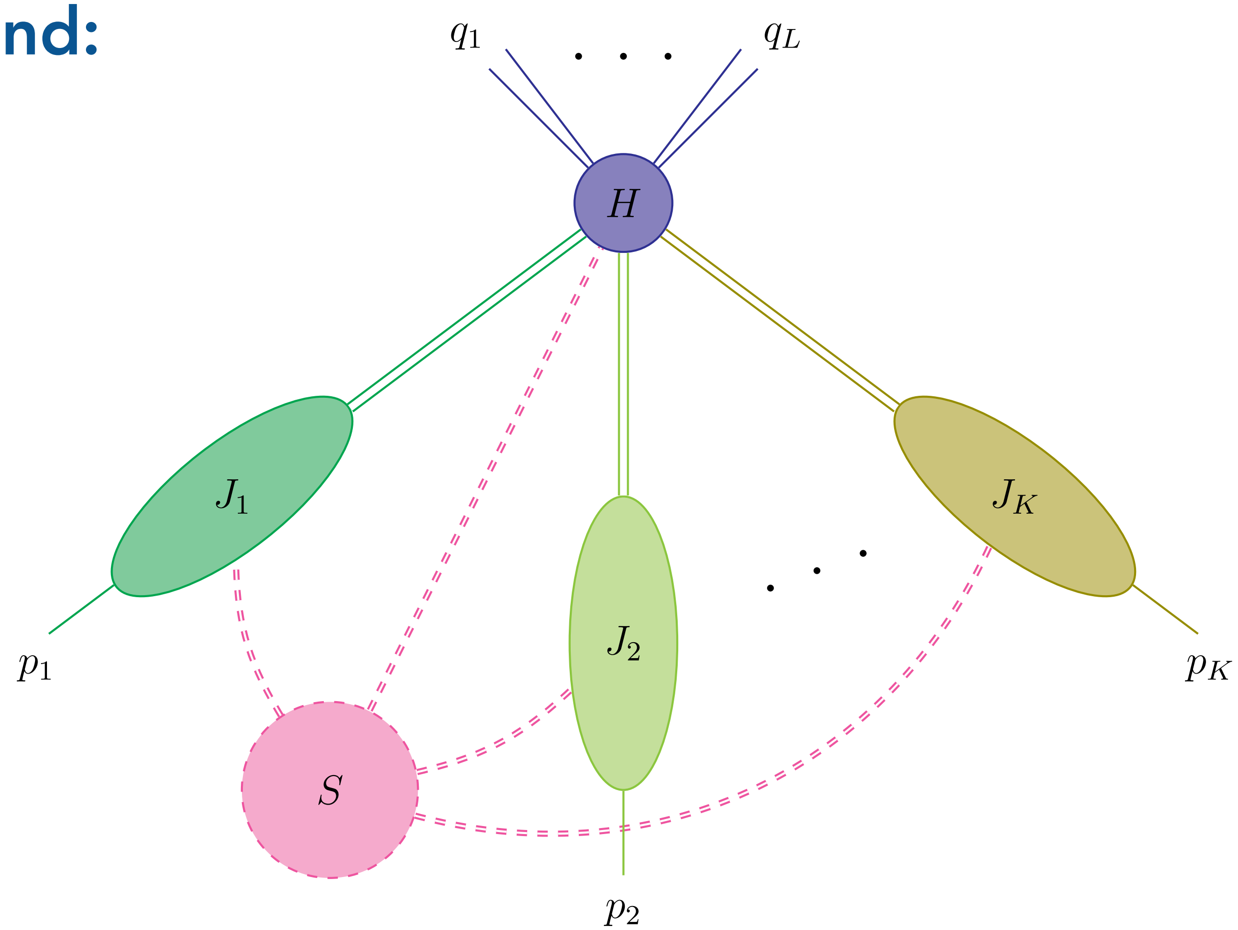
$$u_{R,e} = 0 \quad \leftrightarrow \quad e \in H$$

$$u_{R,e} = -1 \quad \leftrightarrow \quad e \in J \equiv \cup_{i=1}^{K} J_i$$

$$u_{R,e} = -2 \quad \leftrightarrow \quad e \in S$$

Consider possible solutions of the Landau equations, search for the scaleful ones
$\Longrightarrow$ constraints from Landau equations & scalefulness

We find:

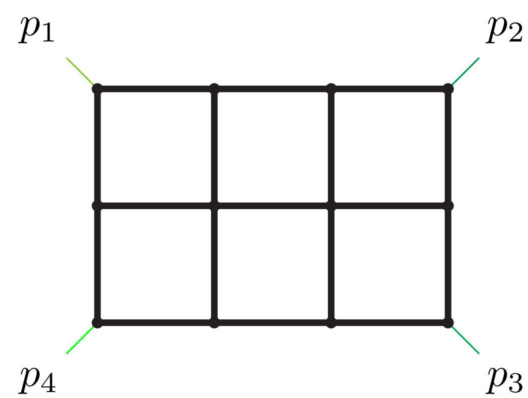


Appears to hold at *any order* in the power expansion (i.e. any order in λ)

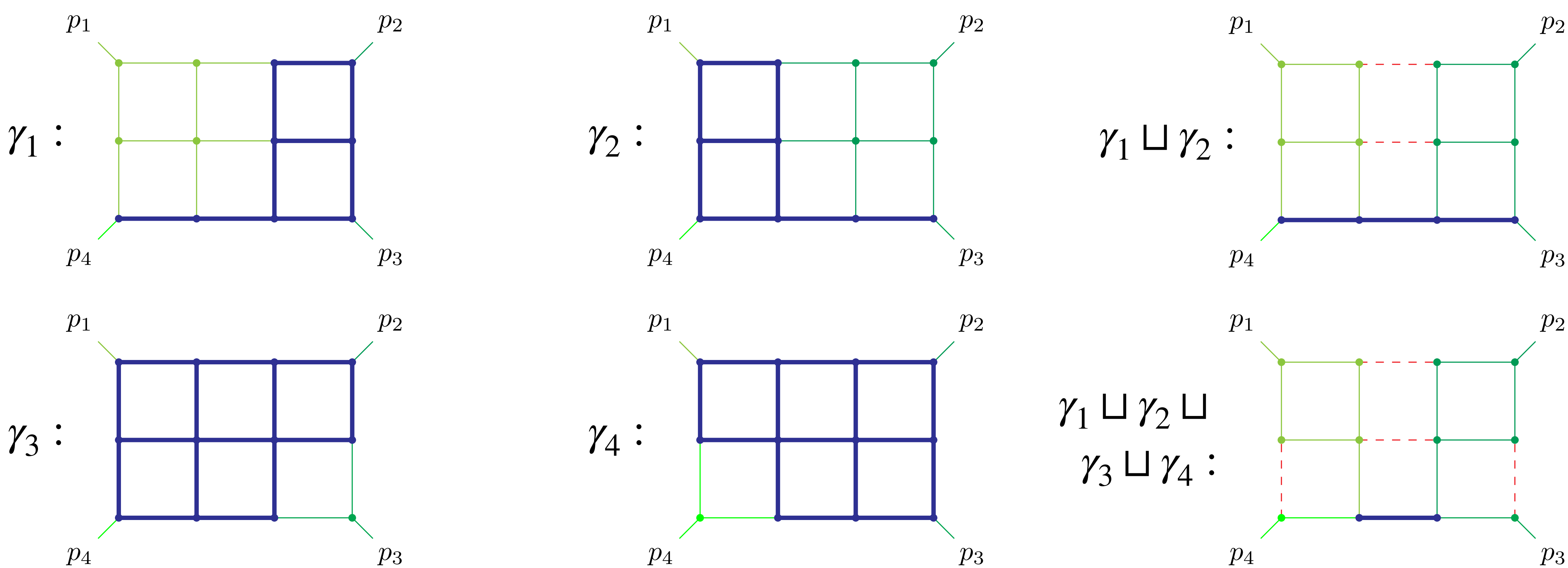
On Shell Expansion: Graphical Method

Can construct graphical method for writing down the region vectors

$G =$



See also:
Arkani-Hamed, Hillman, Mizera 22

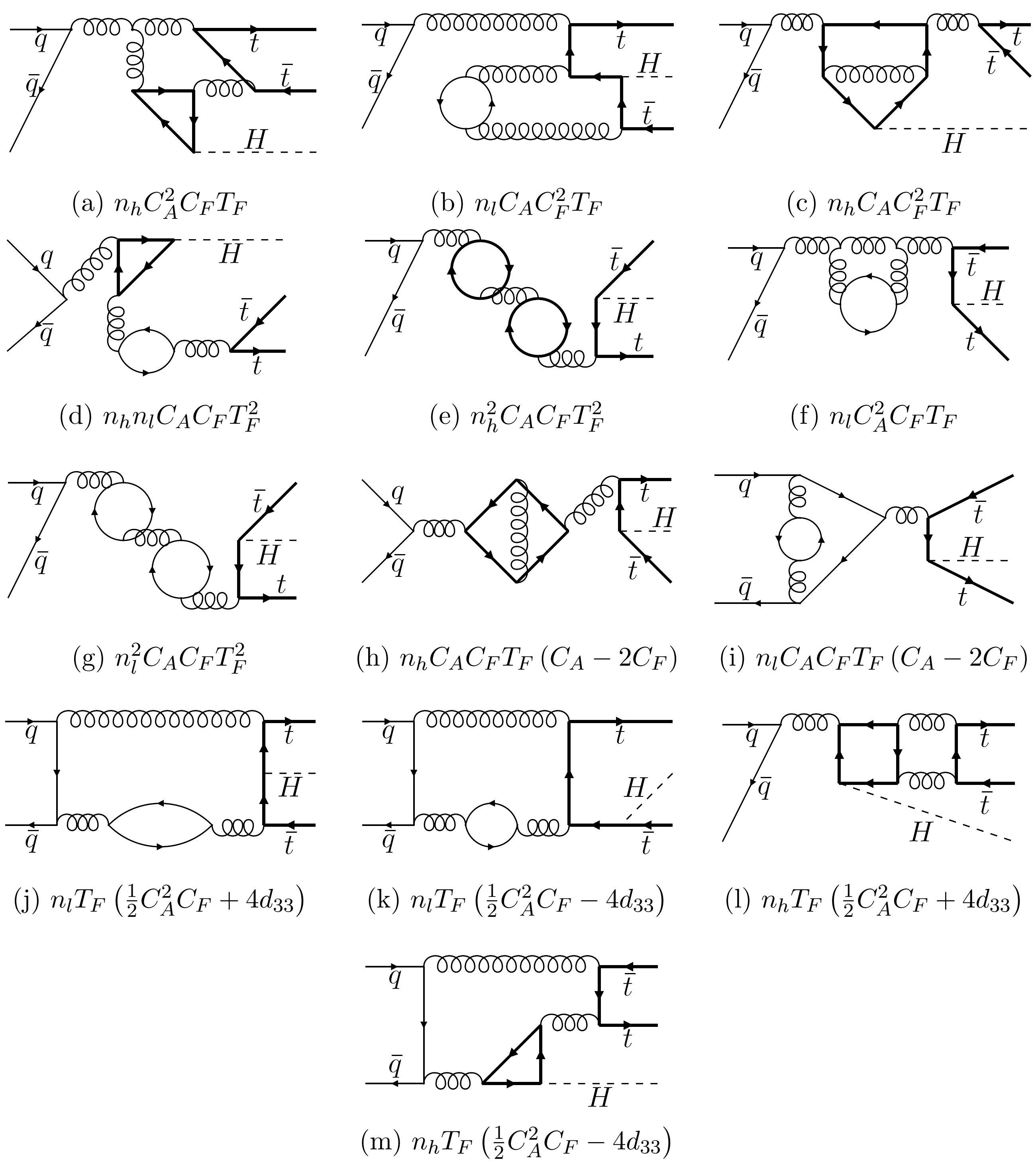


Checked algorithm explicitly for all diagrams in ϕ^3, ϕ^4 with up to 3-legs @ 5-loops & 4-legs @ 4-loops

Work in Progress…

$t\bar{t}H$: Quark Initiated n_f Piece

Compute 2-loop ingredients for $t\bar{t}H$, starting with $q\bar{q}$, n_f pieces



(a) $n_h C_A^2 C_F T_F$ (b) $n_l C_A C_F^2 T_F$ (c) $n_h C_A C_F^2 T_F$

(d) $n_h n_l C_A C_F T_F^2$ (e) $n_h^2 C_A C_F T_F^2$ (f) $n_l C_A^2 C_F T_F$

(g) $n_l^2 C_A C_F T_F^2$ (h) $n_h C_A C_F T_F (C_A - 2C_F)$ (i) $n_l C_A C_F T_F (C_A - 2C_F)$

(j) $n_l T_F \left(\frac{1}{2} C_A^2 C_F + 4 d_{33}\right)$ (k) $n_l T_F \left(\frac{1}{2} C_A^2 C_F - 4 d_{33}\right)$ (l) $n_h T_F \left(\frac{1}{2} C_A^2 C_F + 4 d_{33}\right)$

(m) $n_h T_F \left(\frac{1}{2} C_A^2 C_F - 4 d_{33}\right)$

New challenges

5-point amplitudes depending on 5 kinematic scales + 2 masses

831 master integrals

WIP: **V. Magerya,** G. Heinrich, SJ, M. Kerner, S. Klein, J. Lang, A. Olsson

$t\bar{t}H$: Quark Initiated n_f Piece (II)

Projectors: Born amplitudes
Reduction: On-the-fly numerical reduction for each phase-space point with Ratracer
Magerya 22

Integrals: quasi-finite, d-factorizing, possible with pySecDec, also investigating DiffExp approach

First look at phase-space points…
Target $\epsilon_{\rm rel} = 1 \cdot 10^{-4}$ precision on amplitude

$$
\begin{aligned}
m_H^2 &= \frac{8085251}{15486360} m_t^2, \\
x_{12} &= 10 m_t^2, \\
x_{23} &= -\frac{2571}{620} m_t^2, \\
x_{35} &= \frac{357583}{168330} m_t^2, \\
x_{54} &= \frac{19381}{5704} m_t^2, \\
x_{41} &= -\frac{2734}{465} m_t^2,
\end{aligned}
$$

$d_{33} N_f T_f$	ε^{-2}	$+0.0049204$
	ε^{-1}	$+0.010292 + 0.022622I$
	ε^{0}	$-0.042837 + 0.069854I$
$d_{33} N_{ft} T_f$	ε^{-2}	$+0.0065606$
	ε^{-1}	$+0.020794 + 0.016814I$
	ε^{0}	$+0.039555 + 0.099444I$
$C_a^2 C_f N_f T_f^2$	ε^{-2}	$-0.0056536 - 0.011281I$
	ε^{-1}	$+0.034203 - 0.046061I$
	ε^{0}	$+0.11670 + 0.0067178I$
$C_a C_f^2 N_f T_f^2$	ε^{-3}	$+0.0025559$
	ε^{-2}	$-0.010942 + 0.028986I$
	ε^{-1}	$-0.15880 + 0.030480I$
	ε^{0}	$-0.41022 - 0.17499I$
$C_a^2 C_f N_{ft} T_f^2$	ε^{-2}	$-0.0050392 - 0.015042I$
	ε^{-1}	$+0.0053639 - 0.059042I$
	ε^{0}	$+0.071442 - 0.084413I$
$C_a C_f^2 N_{ft} T_f^2$	ε^{-3}	0.0040894
	ε^{-2}	$-0.0047233 + 0.032887I$
	ε^{-1}	$-0.14753 + 0.12175I$
	ε^{0}	$-0.59562 + 0.11786I$
$C_a C_f N_f^2 T_f^3$	ε^{-2}	$+0.0013631$
	ε^{-1}	$+0.0021910 + 0.0085648I$
	ε^{0}	$-0.023414 + 0.013766I$
$C_a C_f N_f N_{ft} T_f^3$	ε^{-2}	$+0.0027262$
	ε^{-1}	$+0.011458 + 0.012303I$
	ε^{0}	$-0.0010859 + 0.053128I$
$C_a C_f N_{ft}^2 T_f^3$	ε^{-2}	$+0.0013631$
	ε^{-1}	$+0.0092667 + 0.0037380I$
	ε^{0}	$+0.026549 + 0.029970I$

$t\bar{t}H$: Quark Initiated n_f Piece

Poles - checked (subtraction formulae + cross-check)

Finite part - O(16k) points   Chen, Ma, Wang, Yang, Ye 22

```
amp3=(
  +eps^-3*(+2.4812080814589719e-03-2.8947822690179251e-12j)
  +eps^-3*(+1.9029876783836805e-12+1.9981956842417825e-12j)*plusminus
  +eps^-2*(-1.2795194320055608e-02+2.5786079682589620e-02j)
  +eps^-2*(+3.2657872292033090e-11+3.3012784793239365e-11j)*plusminus
  +eps^-1*(-1.4218012895908955e-01-7.6189537659053942e-04j)
  +eps^-1*(+1.8573326634188658e-07+1.7282326416744234e-07j)*plusminus
  +eps^0*(-2.5869321981561438e-01-2.1758799270790402e-01j)
  +eps^0*(+2.2204593845353293e-06+2.3132847824753248e-06j)*plusminus
)
amp3 relative errors by order: 0.00e+00, 0.00e+00, 1.78e-06, 9.49e-06
amp4=(
  +eps^-3*(-2.2776225350185086e-13-5.7714358636920437e-14j)
  +eps^-3*(+1.5245922908967811e-13+1.3428935998390410e-13j)*plusminus
  +eps^-2*(-4.6780134875432436e-03-1.3033417608064988e-02j)
  +eps^-2*(+4.4565990403149097e-09+5.1370665056934937e-09j)*plusminus
  +eps^-1*(+1.1134216059718732e-02-4.4304381365715018e-02j)
  +eps^-1*(+9.5922064468682343e-07+9.4844016477226567e-07j)*plusminus
  +eps^0*(+4.5781345959152238e-02-4.4723941222410774e-02j)
  +eps^0*(+3.0592105076939906e-06+3.1320453916005085e-06j)*plusminus
)
amp4 relative errors by order: 0.00e+00, 4.91e-07, 2.95e-05, 6.84e-05
```

Good point: 3-4 mins

GPU: NVidia A100 40GB

```
amp3=(
  +eps^-3*(+7.4339015599071689e-04-7.9500880445625670e-13j)
  +eps^-3*(+5.7145309613096924e-13+5.9971316345385848e-13j)*plusminus
  +eps^-2*(-3.0981991383820405e-03+9.1682626032127991e-03j)
  +eps^-2*(+9.7650289569317253e-12+9.9189849982950227e-12j)*plusminus
  +eps^-1*(-4.6793708870014772e-02+9.3085610262487999e-03j)
  +eps^-1*(+6.2340661736749192e-07+6.4733201139491059e-07j)*plusminus
  +eps^0*(-1.1711570776049893e-01-4.2728554150957172e-02j)
  +eps^0*(+2.9732556334539358e-06+2.9950056109814405e-06j)*plusminus
)
amp3 relative errors by order: 0.00e+00, 0.00e+00, 1.88e-05, 3.39e-05
amp4=(
  +eps^-3*(-1.2956460461802100e-08+7.8310287578327527e-09j)
  +eps^-3*(+3.3667459966638271e-08+2.6375283024654377e-08j)*plusminus
  +eps^-2*(-1.6697924163405768e-03-4.8625186952290581e-03j)
  +eps^-2*(+8.1602698321304684e-07+6.7055067134358471e-07j)*plusminus
  +eps^-1*(+9.5428369411900554e-04-1.9084852520645498e-02j)
  +eps^-1*(+1.1890510956916082e-04+5.8577385741283674e-04j)*plusminus
  +eps^0*(+3.1867828485252998e-02+4.1227641634472185e-03j)
  +eps^0*(+5.6825877897303881e-03+3.3489604326241845e-02j)*plusminus
)
amp4 relative errors by order: 2.83e+00, 2.05e-04, 3.13e-02, 1.06e+00
```

Bad point: >24 hr

GPU: NVidia A100 40GB

Need to deal with poor performance near thresholds, decide how to sample the PS

Conclusion

Updates

- New``DistEval'' integrator: ~3-5x faster than old ``IntLib''
- Median lattice rules: lattices of unlimited size, smaller fluctuations in error
- Coefficients: accept GiNaC compatible input
- Tools for MoR: including extra regulator construction

Applications

- Various processes at $2 \to 2$ with many masses
- First applications to $2 \to 3$ amplitudes

MoR

- How does the analysis generalise to other types of expansion (e.g. Regge, massive particles, threshold/potential)?
- How should we deal with regions due to cancellation? (e.g. negative c_i)

Thank you for listening!

Backup

Graphical Algorithm

A taste of why this might hold

1) Partition the graph into hard (H), jet (J_i) and soft (S) subgraphs with n_H, n_J, n_S propagators and LP parameters scaling as $\{0, -1, -2\}$

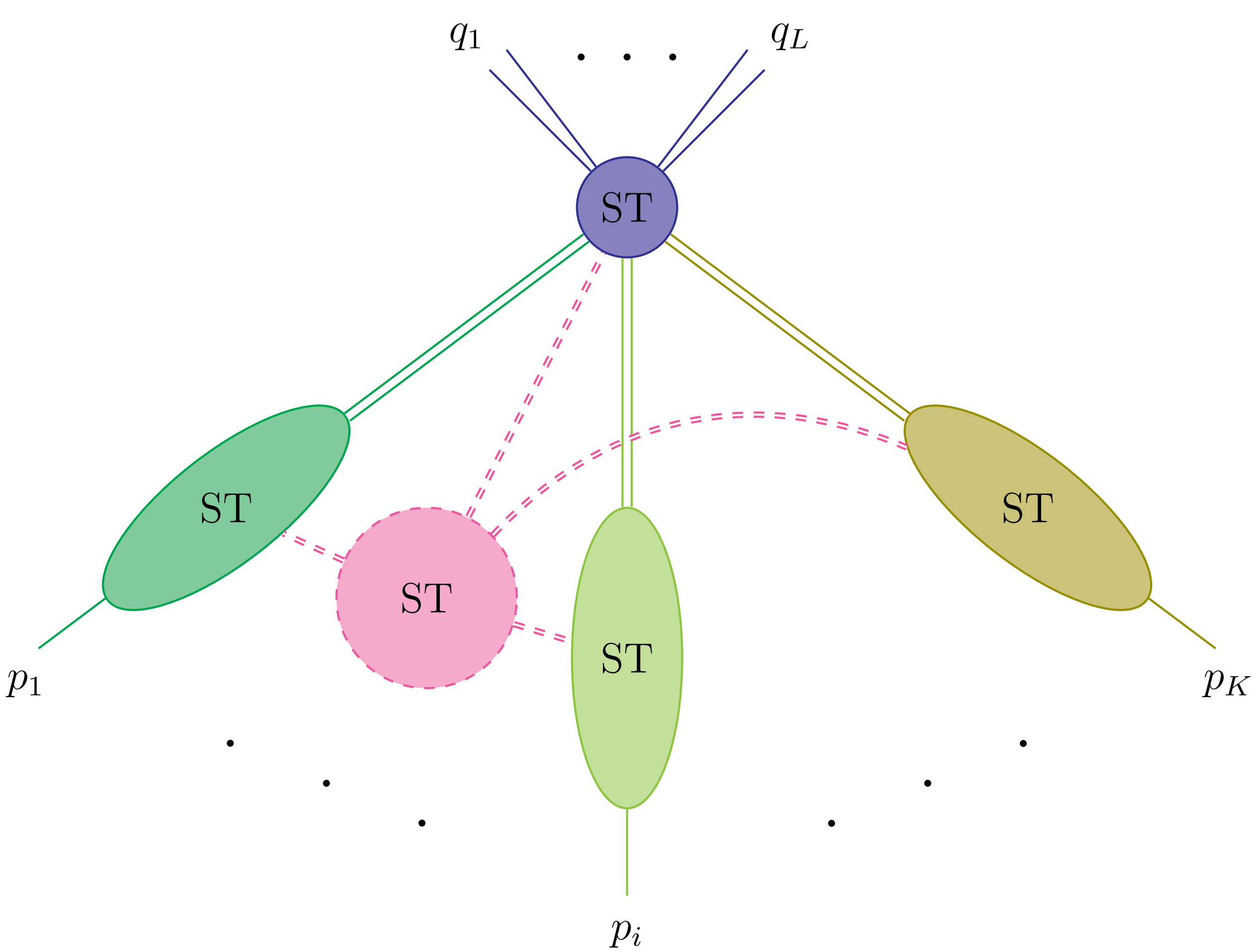
2) Define *contracted subgraphs* ($\widetilde{J}_i$) and ($\widetilde{S}$) by contracting $G \backslash J_i$ or $G \backslash S$ to a point

Possible to show:

$$L(G) = L(H) + \sum_{i=1}^{K} L(\widetilde{J}_i) + L(\widetilde{S})$$

$$n_H \geq L(H), \quad n_S \leq L(\widetilde{S})$$

We want to minimise $\boldsymbol{r} \cdot \boldsymbol{v}_R$
$\Longrightarrow$ small n_H, large n_S



Consider $\mathscr{U}(G)$ (degree $L(G)$):

$$n_H = L(H), \quad n_J = L(\widetilde{J}), \quad n_S = L(\widetilde{S})$$

$$\mathscr{U}^{(R)}(\boldsymbol{x}) = U_H(\boldsymbol{x}^{[H]}) \cdot \Big(\prod_i U_{J_i}(\boldsymbol{x}^{[J_i]})\Big) \cdot U_S(\boldsymbol{x}^{[S]})$$

A taste of why this might hold

Similar (though slightly longer) arguments lead to the following theorem

Theorem 2. *For any region R in the on-shell expansion of a wide-angle scattering graph G, the leading Lee-Pomeransky polynomial takes the form*

$$\mathcal{P}_0^{(R)}(\boldsymbol{x};\boldsymbol{s}) = \mathcal{U}^{(R)}(\boldsymbol{x}) + \mathcal{F}^{(R)}(\boldsymbol{x};\boldsymbol{s}) \tag{3.34}$$

$$\mathcal{F}^{(R)}(\boldsymbol{x};\boldsymbol{s}) = \sum_{i=1}^{K} \mathcal{F}^{(p_i^2,R)}(\boldsymbol{x};\boldsymbol{s}) + \mathcal{F}_{\mathrm{I}}^{(q^2,R)}(\boldsymbol{x};\boldsymbol{s}) + \sum_{i>j=1}^{K} \mathcal{F}_{\mathrm{II}}^{(q_{ij}^2,R)}(\boldsymbol{x};\boldsymbol{s}) \tag{3.35}$$

These polynomials factorise as follows

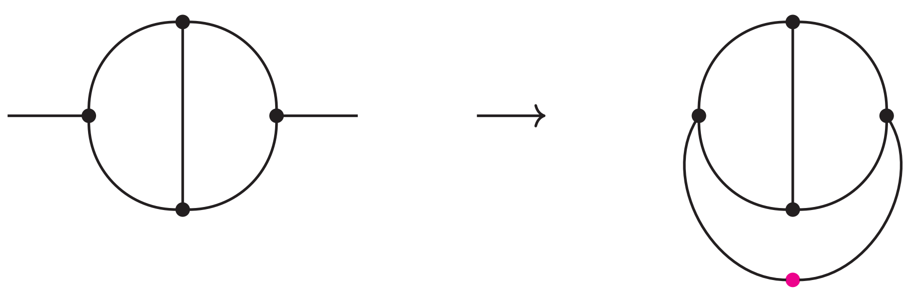
$$\begin{aligned}
\mathcal{U}^{(R)}(\boldsymbol{x}) &= \mathcal{U}_H(\boldsymbol{x}^{[H]}) \cdot \Big(\prod_{i=1}^{K} \mathcal{U}_{J_i}(\boldsymbol{x}^{[J_i]})\Big) \cdot \mathcal{U}_S(\boldsymbol{x}^{[S]})\,,\\
\mathcal{F}^{(p_i^2,R)}(\boldsymbol{x};\boldsymbol{s}) &= \mathcal{U}_H(x^{[H]}) \cdot \mathcal{F}_{J_i}^{(p_i^2)}(\boldsymbol{x}^{[J_i]};\boldsymbol{s}) \cdot \Big(\prod_{j\neq i}^{K} \mathcal{U}_{J_j}(\boldsymbol{x}^{[J_j]})\Big) \cdot \mathcal{U}_S(\boldsymbol{x}^{[S]})\,,\\
\mathcal{F}_{\mathrm{I}}^{(q^2,R)}(\boldsymbol{x};\boldsymbol{s}) &= \mathcal{F}_{H\cup J}^{(q^2)}(\boldsymbol{x}^{[H]},\boldsymbol{x}^{[J]}) \cdot \mathcal{U}_S(\boldsymbol{x}^{[S]})\,,\\
\mathcal{F}_{\mathrm{II}}^{(q_{ij}^2,R)}(\boldsymbol{x};\boldsymbol{s}) &= \mathcal{U}_H(\boldsymbol{x}^{[H]}) \cdot \mathcal{F}_{J_i\cup J_j\cup S}^{(q_{ij}^2)}(\boldsymbol{x}^{[J_i]},\boldsymbol{x}^{[J_j]},\boldsymbol{x}^{[S]}) \cdot \prod_{k\neq i,j} \mathcal{U}_{J_k}(\boldsymbol{x}^{[J_k]})\,.
\end{aligned} \tag{3.36}$$

We find that contributions correspond to solutions of the Landau equations only if some further conditions hold (suggested by our previous figures)

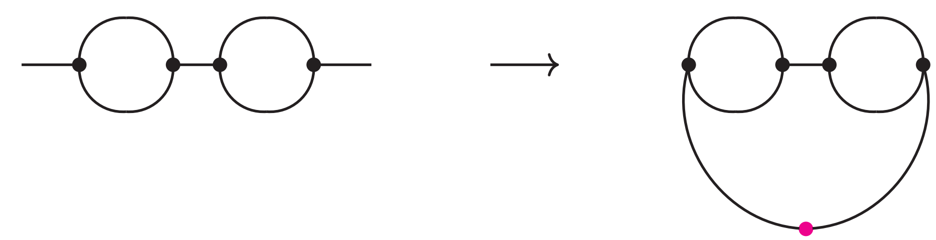
Some Definitions

Motic: components become 1PI after connecting all external lines to a point
Brown 15

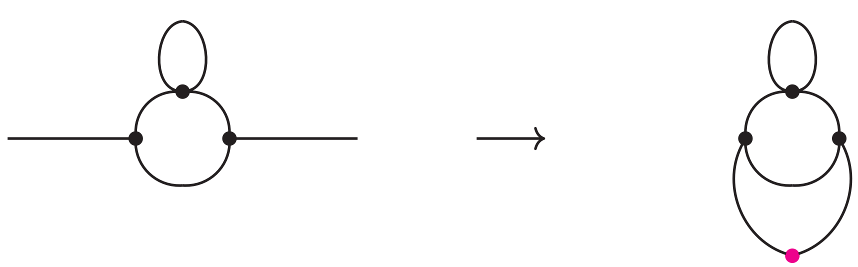
Mojetic: components become 1VI after connecting all external lines to a point
(= motic & scaleful, for massless diagrams)



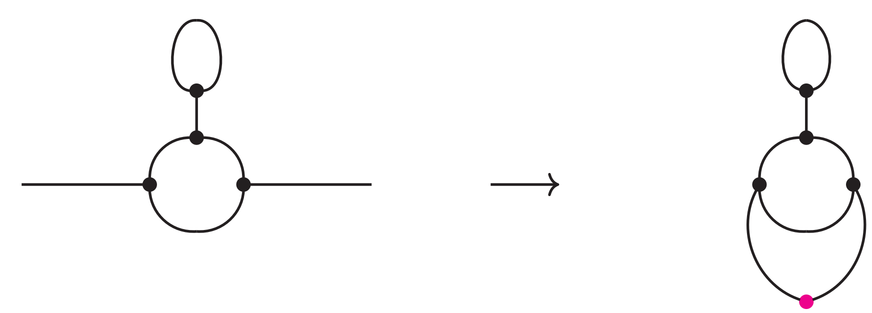
(a) Both motic and mojetic.



(b) Both motic and mojetic.



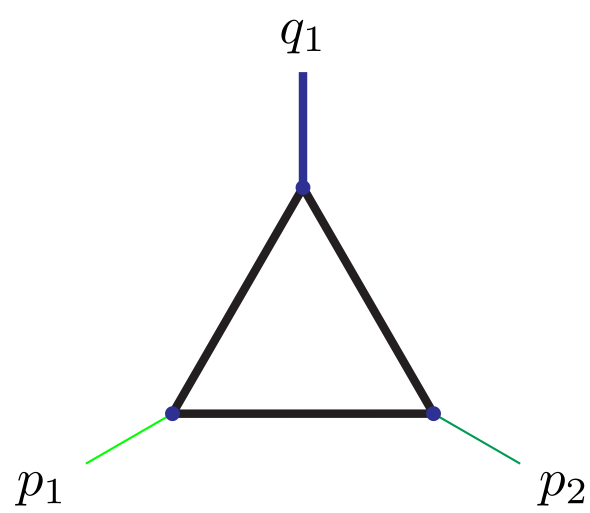
(c) Motic but not mojetic.



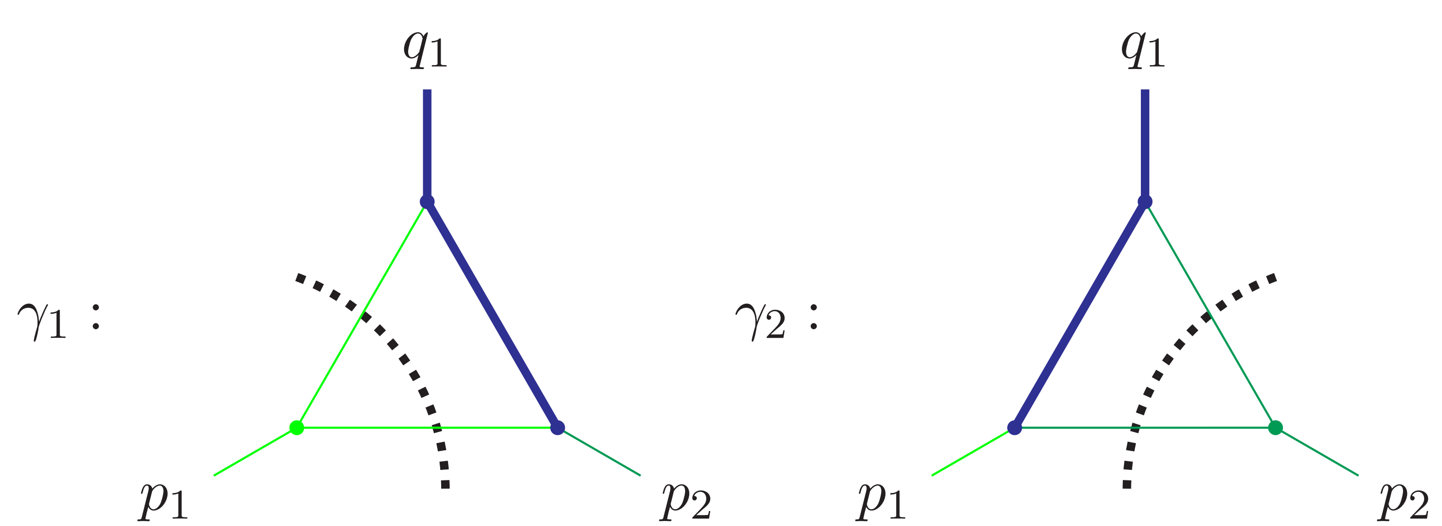
(d) Neither motic nor mojetic.

Graphical Construction Algorithm

$G =$

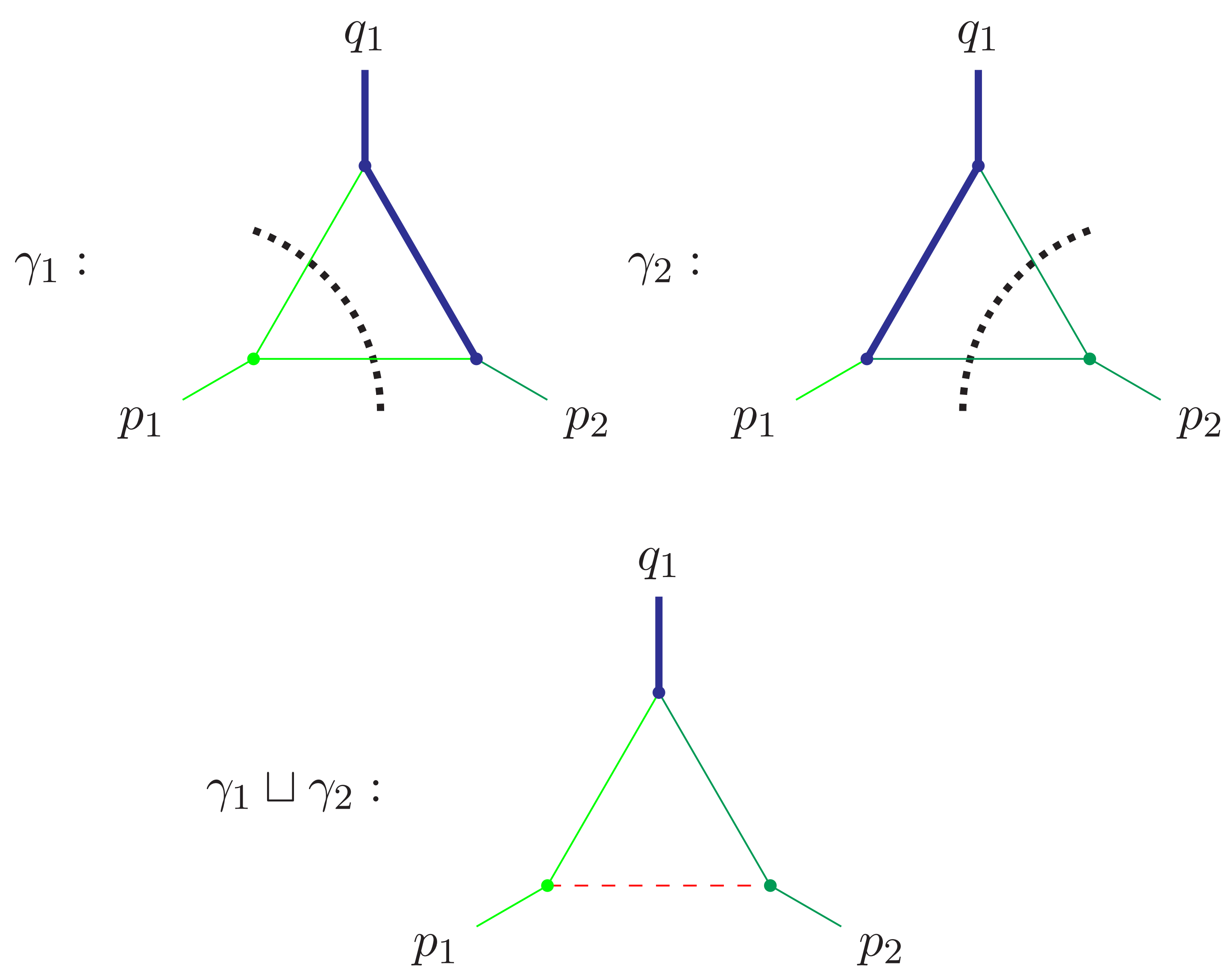


Step 1: For each $i = 1,\ldots,K$, construct the one-external subgraph γ_i in the p_i channel, such that the subgraph $H_i \equiv G\backslash\gamma_i$ is mojetic



Graphical Construction Algorithm

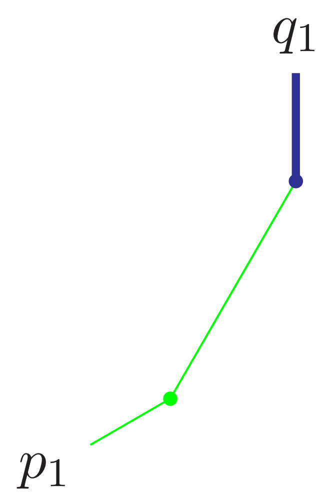
Step 2: Consider all possible sets $\{\gamma_1, \ldots, \gamma_K\}$.
If an edge has been assigned to two or more γ_i, it belongs to the soft subgraph S; if it has been assigned to exactly one γ_i, it belongs to the jet subgraph J_i; if it has not been assigned to any γ_i, it belongs to H.



Graphical Construction Algorithm

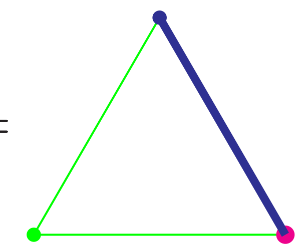
Step 3: Check that result obeys: (i) each jet subgraph J_i is connected; (ii) each hard subgraph H is connected; (iii) each of the K subgraphs $H \cup J \backslash J_i$ $(i = 1, \ldots, K)$ is mojetic. The region is ruled out if any of these conditions are not satisfied.

$H \cup J \setminus J_2 =$



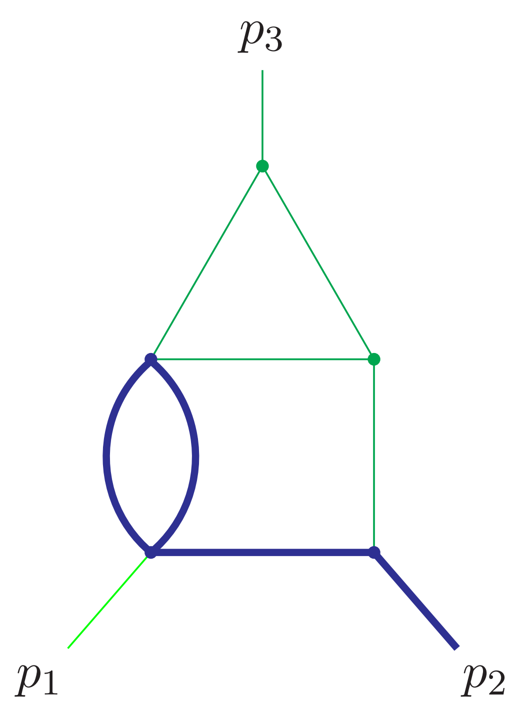
,

$H \cup J \setminus J_2|_c =$

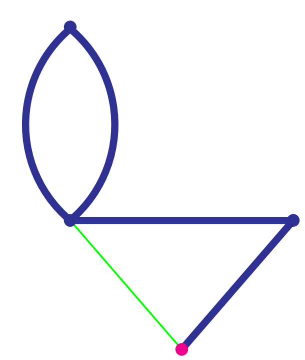


Example - Failing criterion (iii): not mojetic

J_3 :



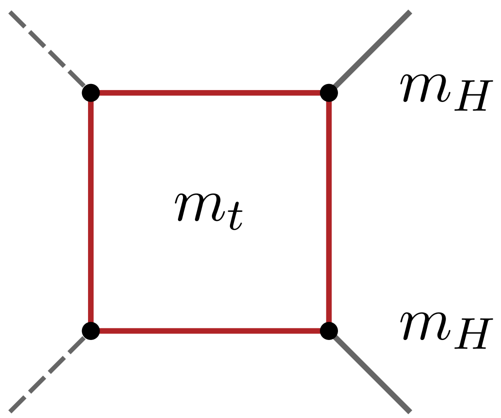
$H_3 \cup J_1|_c =$



Expansion by Regions

pySecDec: EBR Box Example

Example: 1-loop massive box expanded for small $m_t^2 \ll s, |t|$



Requires the use of analytic regulators

Can regulate spurious singularities by adjusting propagators powers

$$G_4 = \mu^{2\epsilon} \int_{-\infty}^{\infty} \frac{d^D k}{i\pi^{D/2}} \frac{1}{[k^2 - m_t^2]^{\delta_1} [(k+p_1)^2 - m_t^2]^{\delta_2} [(k+p_1+p_2)^2 - m_t^2]^{\delta_3} [(k-p_4)^2 - m_t^2]^{\delta_4}}$$

Can keep $\delta_1, \ldots, \delta_4$ symbolic or $\delta_1 = 1 + n_1/2, \delta_2 = 1 + n_1/3, \ldots$ and take $n_1 \to 0^+$

Output region vectors:

$\mathbf{v}_1 = (0,0,0,0,1)$

$\mathbf{v}_2 = (-1,-1,0,0,1)$

$\mathbf{v}_3 = (0,0,-1,-1,1)$

$\mathbf{v}_4 = (-1,0,0,-1,1)$

$\mathbf{v}_5 = (0,-1,-1,0,1)$

Result: s = 4.0, t = -2.82843, $m_t^2 = 0.1, m_h^2 = 0$)

$$I = -1.30718 \pm 2.7 \cdot 10^{-6} + (1.85618 \pm 3.0 \cdot 10^{-6})\, i$$

$$+ \mathcal{O}\left(\epsilon, n_1, \frac{m_t^2}{s}, \frac{m_t^2}{t}\right)$$

Geometric Method: Negative Coefficients

Issue 2: What happens if we have negative coefficients $c_i < 0$? ← **not handled by pySecDec (yet!)**

Consider a 1-loop massive bubble at threshold $y = m^2 - q^2/4 \to 0$

$$\mathcal{F} = q^2/4(x_1 - x_2)^2 + y(x_1 + x_2)^2$$

Can split integral into two subdomains $x_1 \leq x_2$ and $x_2 \leq x_1$ then remap

$$\begin{matrix} x_1 = x_1'/2 \\ x_2 = x_2' + x_1'/2 \end{matrix} : \quad \mathcal{F} \to \frac{q^2}{4} x_2'^2 + y(x'_1 + x'_2)^2 \quad \text{(for first domain)}$$

Various tools attempt to find such re-mappings:

FIESTA Jantzen, A. Smirnov, V. Smirnov 12

Check all pairs of variables (x_1, x_2) which are part of monomials of opposite sign

For each pair, try to build linear combination x_1' s.t negative monomial vanishes

Repeat until all negative monomials vanish **or** warn user

ASPIRE Ananthanarayan, Pal, Ramanan, Sarkar 18; B. Ananthanarayan, Das, Sarkar 20

Consider Gröbner basis of $\{\mathcal{F}, \partial\mathcal{F}/x_1, \partial\mathcal{F}/x_2, \ldots\}$ (i.e. $\mathcal{F}$ and Landau equations)

Eliminate negative monomials with linear transformations $x_1 \to a x_1', x_2 \to x_2' + a x_1'$

Geometric Method: Determining the Regions (VI)

Rewrite our polynomial as: $P(\mathbf{x}) = Q(\mathbf{x}) + R(\mathbf{x})$

With $Q(\mathbf{x})$ defined such that it contains all of the lowest order terms in t

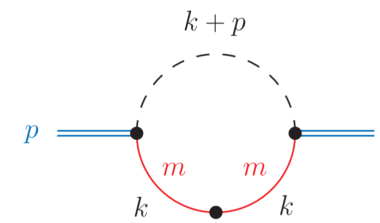
Then, binomial expansion of

$P(\mathbf{x})^m = Q(\mathbf{x})^m \left(1 + \frac{R(\mathbf{x})}{Q(\mathbf{x})}\right)^m$ converges for $\mathbf{x} = t^{\mathbf{u}}$ if $R(\mathbf{x})/Q(\mathbf{x}) < 1$

Some observations:

- An expansion with region vector $\mathbf{v}$ converges at a point $\mathbf{u}$ if the lowest order terms along the direction $\mathbf{v}$ contain the lowest order terms along the direction $\mathbf{u}$
- For any direction $\mathbf{u}$ the vertices with the smallest $< \mathbf{p}_i, \mathbf{u} >$ must be part of some facet F of the polytope
- Since $u_{N+1} > 0$, the lowest order terms for any $\mathbf{u}$ must lie on a facet whose inwards pointing normal vector has a positive $(N+1)$-th component, let us call the set of such facets F^+

Transform the expression for the full integral:



$$F = \int\limits_{k\in D_h} \mathrm{D}k\, I + \int\limits_{k\in D_s} \mathrm{D}k\, I = \sum_i \int\limits_{k\in D_h} \mathrm{D}k\, T_i^{(h)} I + \sum_j \int\limits_{k\in D_s} \mathrm{D}k\, T_j^{(s)} I$$

$$= \sum_i \Bigg(\int\limits_{k\in\mathbb{R}^d} \mathrm{D}k\, T_i^{(h)} I - \sum_j \int\limits_{k\in D_s} \mathrm{D}k\, T_j^{(s)} T_i^{(h)} I \Bigg) + \sum_j \Bigg(\int\limits_{k\in\mathbb{R}^d} \mathrm{D}k\, T_j^{(s)} I - \sum_i \int\limits_{k\in D_h} \mathrm{D}k\, T_i^{(h)} T_j^{(s)} I \Bigg)$$

The expansions commute: $\boxed{T_i^{(h)} T_j^{(s)} I = T_j^{(s)} T_i^{(h)} I \equiv T_{i,j}^{(h,s)} I}$

$\Rightarrow$ **Identity:** $$F = \underbrace{\sum_i \int \mathrm{D}k\, T_i^{(h)} I}_{\boldsymbol{F^{(h)}}} + \underbrace{\sum_j \int \mathrm{D}k\, T_j^{(s)} I}_{\boldsymbol{F^{(s)}}} - \underbrace{\sum_{i,j} \int \mathrm{D}k\, T_{i,j}^{(h,s)} I}_{\boldsymbol{F^{(h,s)}}}$$

All terms are integrated over the whole integration domain $\mathbb{R}^d$ as prescribed for the expansion by regions $\Rightarrow$ location of boundary Λ between D_h, D_s is irrelevant.

Slide from: Bernd Jantzen, High Precision for Hard Processes (HP2) 2012

The general formalism (details)

Identities as in the examples are generally valid, under some conditions.

Consider

- a (multiple) integral $F = \int \mathrm{D}k\, I$ over the domain D (e.g. $D = \mathbb{R}^d$),
- a set of N regions $R = \{x_1, \ldots, x_N\}$,
- for each region $x \in R$ an expansion $T^{(x)} = \sum_j T_j^{(x)}$ which converges absolutely in the domain $D_x \subset D$.

Conditions

- $\bigcup_{x \in R} D_x = D \qquad [D_x \cap D_{x'} = \emptyset \; \forall x \neq x']\,.$
- Some of the expansions commute with each other.
  Let $R_\mathsf{c} = \{x_1, \ldots, x_{N_\mathsf{c}}\}$ and $R_\mathsf{nc} = \{x_{N_\mathsf{c}+1}, \ldots, x_N\}$ with $1 \leq N_\mathsf{c} \leq N$.
  Then: $T^{(x)} T^{(x')} = T^{(x')} T^{(x)} \equiv T^{(x,x')} \; \forall x \in R_\mathsf{c}\,,\; x' \in R\,.$
- Every pair of non-commuting expansions is invariant under some expansion from R_c:
  $\forall x_1', x_2' \in R_\mathsf{nc},\; x_1' \neq x_2',\; \exists x \in R_\mathsf{c} : \; T^{(x)} T^{(x_2')} T^{(x_1')} = T^{(x_2')} T^{(x_1')}\,.$
- $\exists$ regularization for singularities, e.g. dimensional (+ analytic) regularization.
  $\hookrightarrow$ All expanded integrals and series expansions in the formalism are well-defined.

Slide from: Bernd Jantzen, High Precision for Hard Processes (HP2) 2012

The general formalism (2)

Under these conditions, the following **identity** holds: $\left[F^{(x,\ldots)} \equiv \sum_{j,\ldots} \int \mathrm{D}k\, T^{(x,\ldots)}_{j,\ldots} I\right]$

$$F = \sum_{x\,\in\,R} F^{(x)} - \sum^{\langle R_\mathrm{c}+1\rangle}_{\{x_1',x_2'\}\,\subset\,R} F^{(x_1',x_2')} + \ldots - (-1)^n \sum^{\langle R_\mathrm{c}+1\rangle}_{\{x_1',\ldots,x_n'\}\,\subset\,R} F^{(x_1',\ldots,x_n')} + \ldots + (-1)^{N_\mathrm{c}} \sum_{x'\,\in\,R_\mathrm{nc}} F^{(x',x_1,\ldots,x_{N_\mathrm{c}})}$$

where the sums run over subsets $\{x_1',\ldots\}$ containing at most one region from R_nc.

Comments

- This identity is exact when the expansions are summed to all orders. ✓
  Leading-order approximation for $F \rightsquigarrow$ dropping higher-order terms.
- It is independent of the regularization (dim. reg., analytic reg., cut-off, infinitesimal masses/off-shellness, ...) as long as all individual terms are well-defined.
- Usually regions & regularization are chosen such that multiple expansions $F^{(x_1',\ldots,x_n')}$ $(n \geq 2)$ are scaleless and vanish.
  [✓ if each $F_0^{(x)}$ is a *homogeneous* function of the expansion parameter with *unique scaling*.]
- If $\exists\, F^{(x_1',x_2',\ldots)} \neq 0 \rightsquigarrow$ relevant overlap contributions ($\rightarrow$ "zero-bin subtractions").
  They appear e.g. when avoiding analytic regularization in SCET. e.g. Manohar, Stewart '06; Chiu, Fuhrer, Hoang, Kelley, Manohar '09; ...

Slide from: Bernd Jantzen, High Precision for Hard Processes (HP2) 2012

Quasi Monte-Carlo

Periodising Transforms

Lattice rules work especially well for continuous, smooth and periodic functions

Functions can be periodized by a suitable change of variables: $\mathbf{x} = \phi(\mathbf{u})$

$$I[f] \equiv \int_{[0,1]^d} \mathrm{d}\mathbf{x}\; f(\mathbf{x}) = \int_{[0,1]^d} \mathrm{d}\mathbf{u}\; \omega_d(\mathbf{u}) f(\phi(\mathbf{u}))$$

$$\phi(\mathbf{u}) = (\phi(u_1), \ldots, \phi(u_d)), \qquad \omega_d(\mathbf{u}) = \prod_{j=1}^{d} \omega(u_j) \qquad \text{and} \qquad \omega(u) = \phi'(u)$$

Korobov transform: $\omega(u) = 6u(1-u), \quad \phi(u) = 3u^2 - 2u^3$

Sidi transform: $\omega(u) = \pi/2 \sin(\pi u), \quad \phi(u) = 1/2(1 - \cos \pi t)$

Baker transform: $\phi(u) = 1 - |2u - 1|$