Analytic Regression and the Semi-numerical Landau Boostrap

Symbology@15

Dec 17, 2025

Matthew Schwartz

Harvard University

Institute for Artificial Intelligence
and Fundamental Interactions (IAIFI)

Based on

“Applications of the Landau Bootstrap” 2410.02424

“Constraints on sequential discontinuities from the geometry of on-shell spaces” 2211.07633

- By H. Hannesdottir, A. McLeod, MDS, C. Vergu

“Analytic Regression of Feynman Integrals from High-Precision Numerical Sampling”, 2507.17815

- By O. Barrera, A. Dersy, R. Husain, MDS and X. Zhang

Outline

1. Landau bootstrap

Easy ↓ Hard

Constraints	# Coeffs
All Symbols	20736
Integrability	6993
Galois symmetry	861
Physical branch cuts	161
Genealogical constraints	28
α-positive thresholds	6

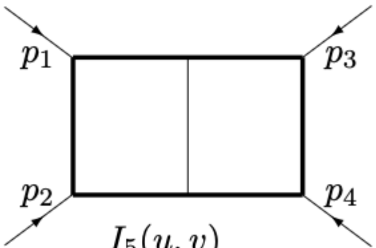
Hard ↑ Easy

2. Analytic regression with lattice reduction

Bootstrapping integrals

Q: Can we bootstrap Feynman integrals?

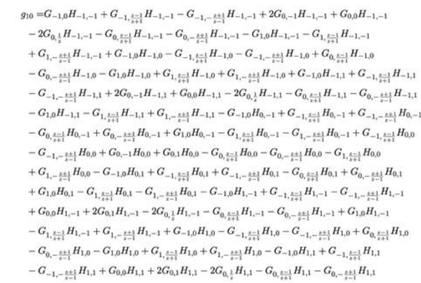
Rules: Don't integrate!



$I_5(u,v)$

$$I_5(u,v) = \int \frac{d^D k_1}{(2\pi)^D} \int \frac{d^D k_2}{(2\pi)^D} \frac{1}{[k_1^2+m^2]\,[k_2^2+m^2]\,[(k_1-p_1)^2+m^2]\,[(k_2-p_3)^2+m^2]\,[(k_1-k_2)^2]}$$

- Computed by Caron-Huot and Henn 1404.2922



1. Parametrize a finite basis

$$I_5(s,t,u,m) = \sum_{j=1}^{\text{finite}} c_j f_j(s,t,u,m)$$

- determined by singularities

2. Landau bootstrap

apply enough constraints to uniquely fix all c_j

3. Analytic regression

fit the c_j numerically

Landau Equations

$$I_G(p) = (n_{\rm int}-1)! \int_0^\infty \prod_{e\in E_{\rm int}(G)} {\rm d}\alpha_e \int \prod_{c\in \widehat{C}(G)} {\rm d}^d k_c \frac{1}{(\ell + i\varepsilon)^{n_{\rm int}}} \delta\Big(1-\sum_{e\in E_{\rm int}(G)} \alpha_e\Big)$$

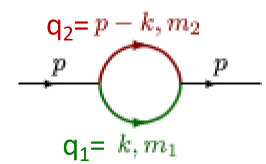
A necessary condition for a singularity is that the *integrand* is singular (ℓ=0)

$$\boxed{\ell = \sum_{e\in E_{\rm int}(G)} \alpha_e (q_e^2 - m_e^2) \;=0}$$

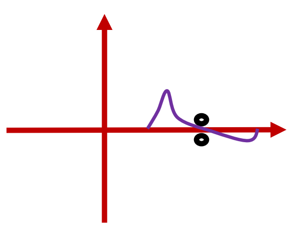
- every internal line is either on-shell (q^2=m^2) or α=0 or both

A necessary condition for a singularity of the *integral* is that poles pinch the contour

for each loop k_c: $\sum_{e\in E_{\rm int}(G^\kappa)} \alpha_e \frac{\partial}{\partial k_c}(q_e^2 - m_e^2) = 0\,.$



Double pole:



integration contour pinched between poles

- since q_e are linear in k_c

$$\Rightarrow \boxed{\sum_{e\,{\rm in\,loop}} \pm\alpha_e q_e^\mu = 0}$$

Landau loop equations

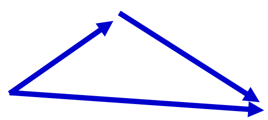
Coleman-Norton interpretation

Landau equations

$$\ell = \sum_{e\in E_{\mathrm{int}}(G)} \alpha_e(q_e^2 - m_e^2) = 0$$

$$\sum_{e\,\mathrm{in\,loop}} \pm\alpha_e q_e^\mu = 0$$

4-momenta add up to zero after rescaling by α



[Coleman and Norton 1965]

Landau diagram is interpreted as space-time diagram

- momenta are on-shell (classical)
- α_e are the proper times for propagation

More physically: singularities due to classically allowed processes

- similar to optical thoerem

Pham interpretation

Landau equations

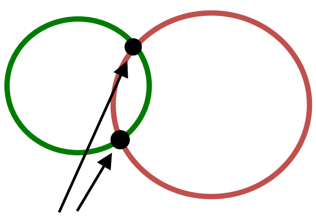
$$\ell = \sum_{e \in E_{\text{int}}(G)} \alpha_e (q_e^2 - m_e^2) = 0$$

$$\sum_{e\,\text{in}\,\text{loop}} \pm \alpha_e q_e^\mu = 0$$

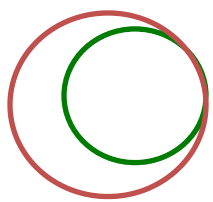
normal vectors of on-shell constraints $q^2=m^2$ are linearly dependent

on-shell constraints (Euclidean d=2)

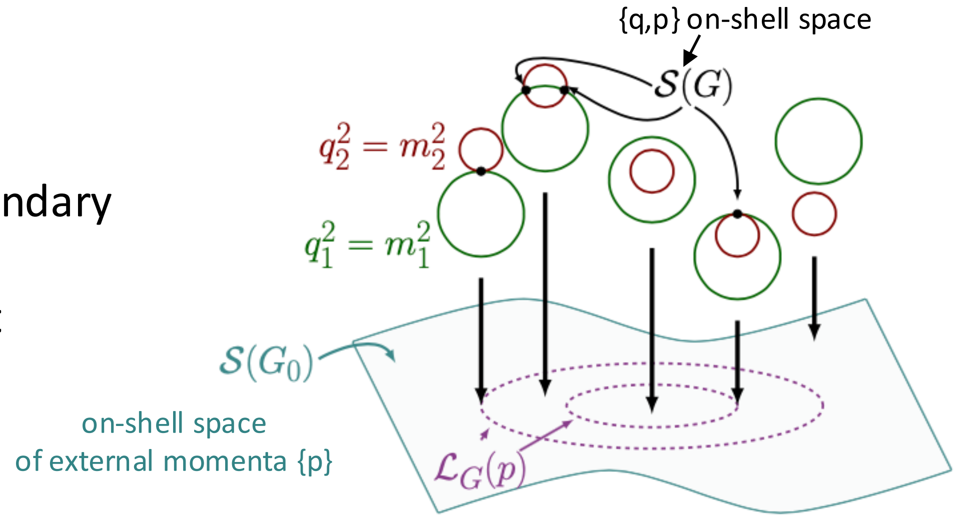
$$q_x^2 + q_y^2 = m_e^2$$



intersection satisfies both on-shell constraints



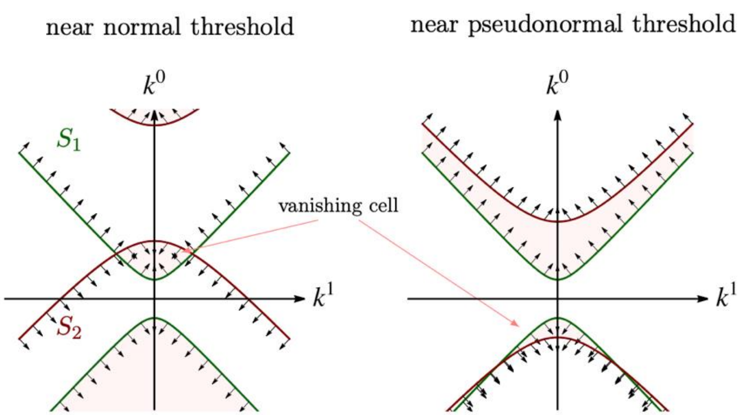
tangent on boundary of space where circles intersect



Landau variety is
the boundary of the projection map

Different kinds of singularities

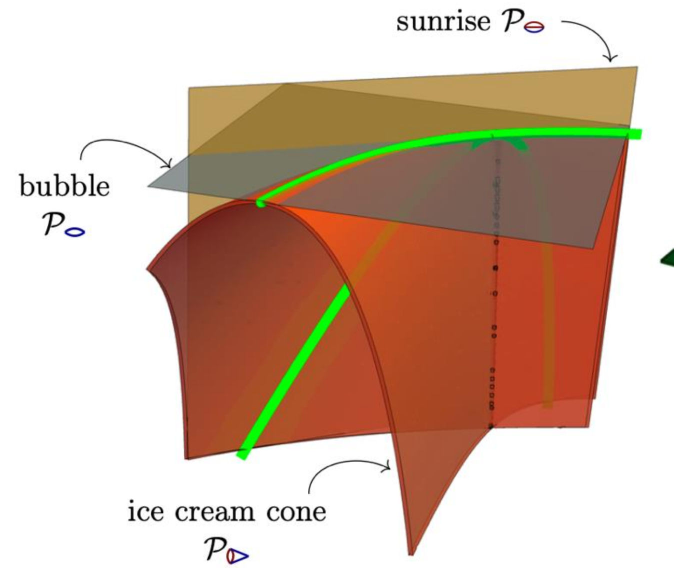
Simple pinches



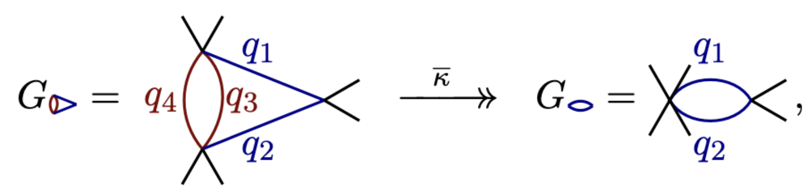
- hypersufaces meet transversely
- e.g. physical thresholds

McLeod, Hannesdottir, MDS, Vergu
arXiv:2211.07633

Non-simple pinches



- hypersufaces meet tangentially
  - e.g. sunrise in the ice-cream cone



- Permanent pinches (e.g. IR divergences)
- Pinches at infinity

Solving the Landau equations

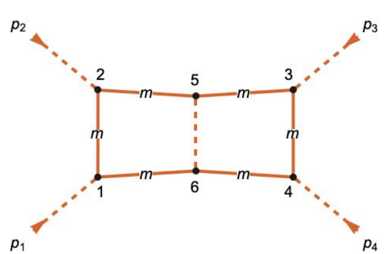
Lots of ways to solve the Landau equations

- Solve them by hand (e.g. Eden et al 1950)
- HyperInt (Panzer 2014)
- PLD (Fevola, Mizera, Telen 2013)
- BaikovLetter (Jiang et al 2024
- Recursive approach (Caron-Huot, M. Correia and M. Giroux 2024)
- Numerical implementation for any diagram (Correia, Giroux, Mizera 2024) SOFIA

Input

```
diag =
 {{{{1, 2}, m1}, {{2, 5}, m2}, {{3, 5}, m3}, {{3, 4}, m4}, {{4, 6}, m5},
     {{1, 6}, m6}, {{5, 6}, m7}}, {{1, M1}, {2, M2}, {3, M3}, {4, M4}}} //.
   m7 :> 0 //. m_ :> m //. M_ :> 0
FeynmanPlot[diag]

{{{{1, 2}, m}, {{2, 5}, m}, {{3, 5}, m}, {{3, 4}, m}, {{4, 6}, m},
  {{1, 6}, m}, {{5, 6}, 0}}, {{1, 0}, {2, 0}, {3, 0}, {4, 0}}}
```



Output

```
candidateSingularities =
  SOFIA[diag, SolverBound → Infinity];
% // TableForm

Total runtime= 2.86441

mm
s12
s23
s12 + s23
mm - s12
mm - s23
4 mm - s12
4 mm - s23
4 mm s12 + 4 mm s23 - s12 s23
mm^2 s12 - 2 mm s12 s23 - 4 mm s23^2 + s12 s23^2
mm s12^2 + 4 mm s12 s23 - s12^2 s23 + 4 mm s23^2 - s12 s23^2
```

Finds all singularities (simple & non-simple)

Finite basis

Alphabet = {letters}

- A letter L can appear in the symbol
- Integral can **only** have singularities when L=0

$$\mathcal{S}[\mathcal{I}] = L_{i_1} \otimes \ldots \otimes L_{i_n}$$

Landau's original paper determined if singularities were logarithmic or square-root

- Not so easy to use, since singularities may appear multiple times on multiple sheets
- Still true that singuarites are always either logarithmic or square root

singularities

{s,t,s-t}

alphabet

$$\left\{ s, t, s-t, \frac{\sqrt{s-t} - \sqrt{s}}{\sqrt{s-t} + \sqrt{s}} \right\}$$

$$\left\{ s=0, \frac{1}{s}=0 \right\}$$

logarithmic

s

$$\left\{ s=0, \frac{1}{t}=0, \quad s=t, t, \frac{1}{s} \right\}$$

logarithmic square root

- Galois symmetry
  - invariance under i ->-i

  or $\sqrt{\cdot} \to -\sqrt{\cdot}$
- $\sqrt{s-t}$ not a valid letter

Finite basis

Location of Landau singularities

$$s = 4m^2, \quad s \to \infty,$$
$$t = 4m^2, \quad t \to \infty,$$
$$m^2 = 0,$$
$$s = 0, \qquad t = 0, \qquad m^2 \to \infty\,,$$
$$s + t = 0, \qquad st - 4m^2 s - 4m^2 t = 0\,.$$

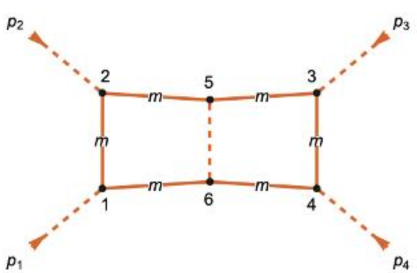
symbol alphabet

$$\tilde{A} = \Big\{u, v, 1+u, 1+v, u+v, 1+u+v, \frac{\beta_u - 1}{\beta_u + 1}, \frac{\beta_v - 1}{\beta_v + 1}, \frac{\beta_{uv} - 1}{\beta_{uv} + 1}, \frac{\beta_{uv} - \beta_u}{\beta_{uv} + \beta_u}, \frac{\beta_{uv} - \beta_v}{\beta_{uv} + \beta_v}, \frac{\beta_{uv} - \beta_u \beta_v}{\beta_{uv} + \beta_u \beta_v}\Big\}$$

$$\beta_u = \sqrt{1+u},\ \beta_v = \sqrt{1+v}\ ;\beta_{uv} = \sqrt{1+u+v}.$$

- SOFIA can also produce the alphabet (may be larger than needed)

- Length of symbol is <= 2 x loop order



2 loop outer-mass double box

- 12 letters
- 12^4 = 20,736 symbol entries

$$\mathcal{S}[\mathcal{I}] = \sum c_{\{i\}} L_{i_1} \otimes L_{i_2} \otimes L_{i_3} \otimes L_{i_4}$$

Finite basis!

Landau Boostrap

$$\mathcal{S}[\mathcal{I}] = \sum c_{\{i\}} L_{i_1} \otimes L_{i_2} \otimes L_{i_3} \otimes L_{i_4}$$

How can we fix the coefficients?

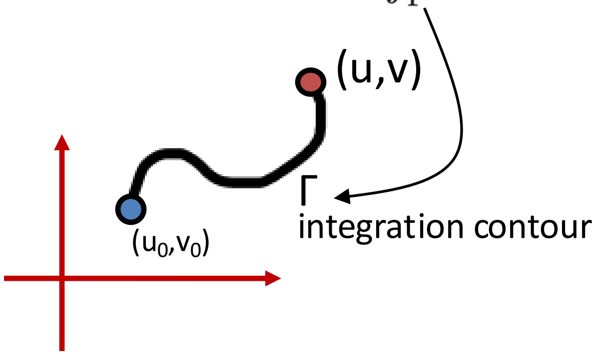
- Integrability
- Galois symmetry
- α-positivity
- First-entry conditions
- Last-entry conditions
- Geneological constraints
  - Sequential discontinuities
  - Cluster adjacency conditions
- Regions analysis
  - Soft, collinear, Regge limits
  - Often don't work diagram-by-diagram
- Direct calculation

Sequences of four letters	20736
Integrable weight-four symbols	6993
Galois symmetry	861
Physical logarithmic branch cuts	161
Genealogical constraints	28
Only algebraic α-positive thresholds	6
Threshold expansion in t	1

Integrability

- Every interated dlog integral has a symbol

$$f(u,v;\Gamma) = c_{i_1,i_2,\dots,i_n} \int_\Gamma d\ln L_{i_1} \circ \cdots \circ d\ln L_{i_n}\,. \longrightarrow \sum c_{i_1,i_2,\dots,i_n}\, L_{i_1} \otimes \cdots \otimes L_{i_n}$$



- Not every symbol corresponds to a function

- For f(u,v,Γ) to be a function, must be independent of local path deformations

$$\Rightarrow \quad [\partial_u, \partial_v] f = 0$$

- Derivatives only act on the last entry of the symbol (end of integration contour)

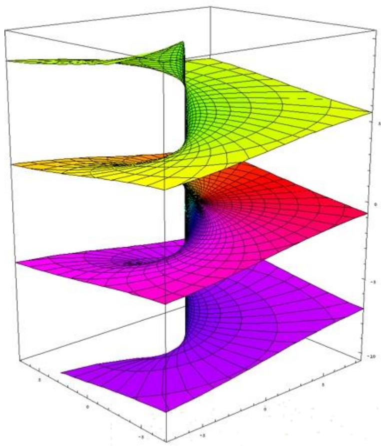
$$\partial_u[S \otimes K \otimes L] = (\partial_u \ln L)[S \otimes K]$$

$$\partial_v\partial_u[S \otimes K \otimes L] = (\partial_v\partial_u \ln L)[S \otimes K] + (\partial_u \ln L)(\partial_v \ln K) S$$

$$[\partial_u, \partial_v][S \otimes K \otimes L] = \underbrace{[(\partial_u \ln L)(\partial_v \ln K) - (\partial_v \ln K)(\partial_u \ln L)]}_{\text{must vanish}} S$$

must vanish
(integrability condition)

α positivity



- Symbol encodes all branch points, even on unphysical sheets

$$I_G(p) = (n_{\text{int}}-1)! \int_0^\infty \prod_{e\in E_{\text{int}}(G)} \mathrm{d}\alpha_e \int \prod_{c\in\widehat{C}(G)} \mathrm{d}^d k_c \frac{1}{(\ell+i\varepsilon)^{n_{\text{int}}}} \delta\Big(1-\sum_{e\in E_{\text{int}}(G)} \alpha_e\Big)$$

actual Feynman integral on physical sheet is over positive α

- Discontinuities/monodromies act on first entry of the symbol

$$I = \int \omega_1 \int \omega_2 \ldots \int \omega_n$$
$$dI = \omega_1 \int \omega_2 \ldots \int \omega_n$$

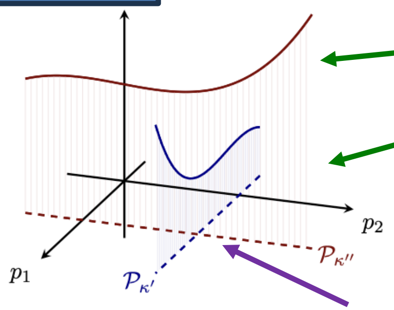
- Singularity for physical momenta (physical sheet) ⟷ singularity of first entry
  - See if α > 0 in solutions to Landau equations: constrain first symbol entries

Genealogical constraints

Which symbol entries can be adjacent?

- Adjacent means sequential discontinuities are possible

Pham approach



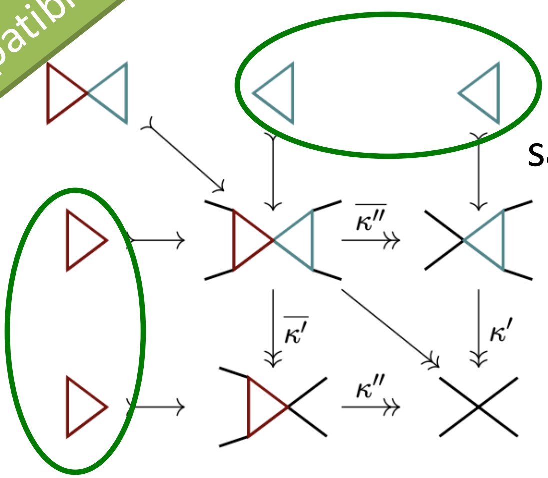
on-shell surfaces may not intersect in *internal momenta*

- vanishing cell from first monodromy doesn't intersect integration contour of second
- sequential monodromy vanishes

$$\left(\mathbb{1}-\mathscr{M}_{\mathcal{P}_{\kappa'}}\right)\left(\mathbb{1}-\mathscr{M}_{\mathcal{P}_{\kappa''}}\right)I_G(p)=0$$

Singular surfaces intersect transversaly in external momenta

compatible

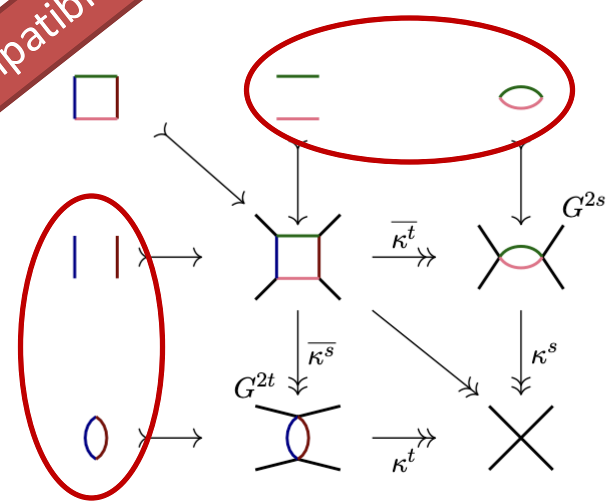


same

same

incompatible

different



different

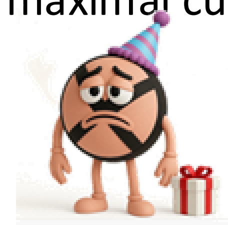
- e.g. Steinmann relations

Outer-mass double box

Landau boostrap

Constraints	# Coeffs
All Symbols	20736
Integrability	6993
Galois symmetry	861
Physical branch cuts	161
Genealogical constraints	28
α-positive thresholds	6

- Impose as many constraints as you can
- Work from easy generic stuff (integrability) to hard
- Also need rational prefactor
  - SOFIA can compute prefactor from maximal cut
  - Not part of the symbol
- Final steps may involve computing integral in a limit

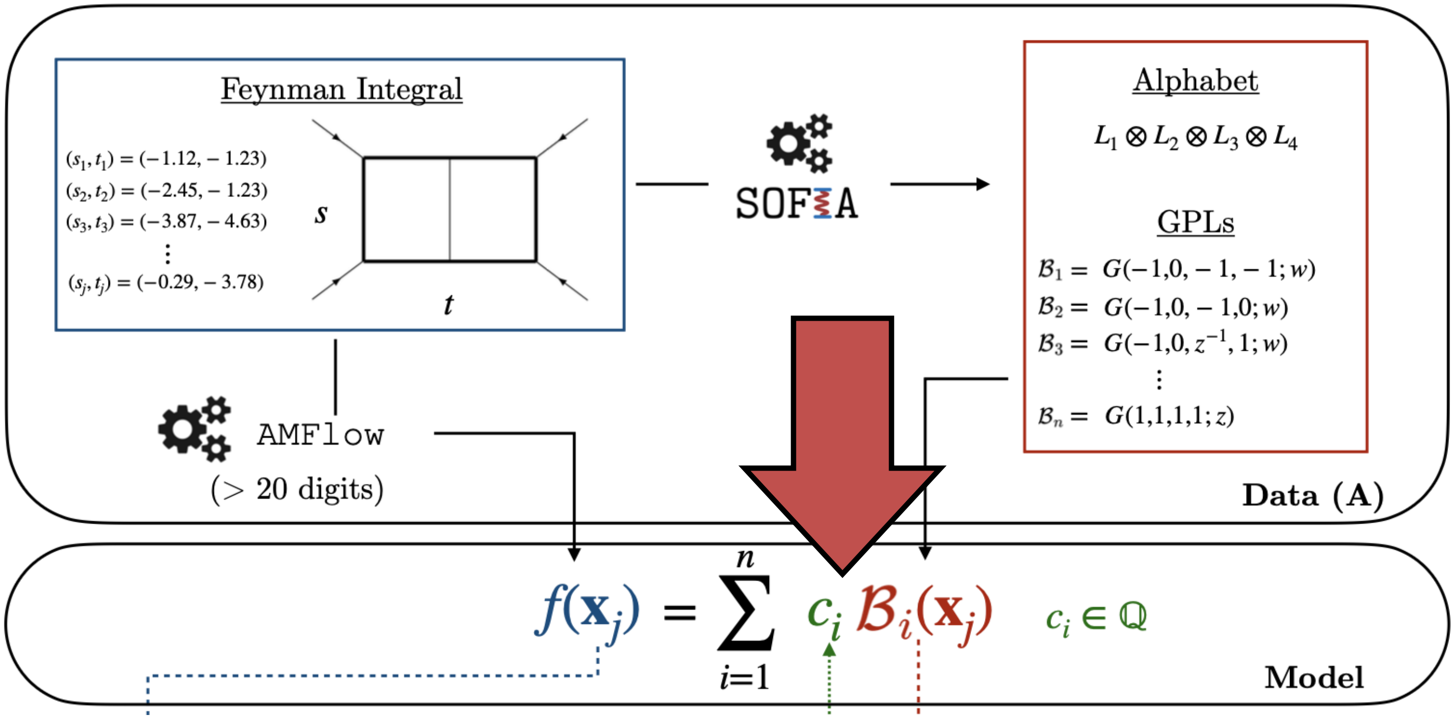


- Landau bootstrap works!

$$\mathcal{S}(\tilde{\mathcal{I}}_{\text{dbox}}) = -L_6 \otimes \frac{L_1}{L_3} \otimes L_6 \otimes L_9 - L_6 \otimes \frac{L_1}{L_3} \otimes L_9 \otimes L_6 + L_6 \otimes L_6 \otimes \frac{L_1 L_2}{L_3 L_5} \otimes L_9 + L_6 \otimes L_9 \otimes \frac{L_2}{L_5} \otimes L_6 + L_6 \otimes L_6 \otimes L_8 \otimes L_6 + L_6 \otimes L_9 \otimes L_8 \otimes L_9 + L_7 \otimes L_{10} \otimes \frac{L_2}{L_5} \otimes L_6 + L_7 \otimes L_{10} \otimes L_8 \otimes L_9 + L_7 \otimes L_7 \otimes \frac{L_1}{L_5} \otimes L_9 + L_7 \otimes L_7 \otimes L_8 \otimes L_6 \,.$$

- Agrees with Caron-Huot & Henn ‘14

Numerical bootstrap



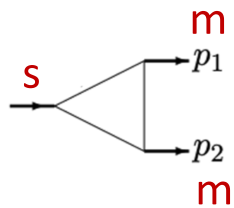
Start with finite basis

can we just fit the c_j numerically?

Requirements

1. Evaluate f(x) numerically to high precision
2. Evaluate B(x) to high precision
3. Solve the equation without losing precision

1-loop triangle

exact answer: $T =$ 

$= \frac{x}{m^2(1-x^2)}\left(-\frac{\pi^2}{3} + 4G(0,-1,x) - 2G(0,0;x)\right)$

one invariant $x = \frac{1-\sqrt{1+\frac{4m^2}{s}}}{1+\sqrt{1+\frac{4m^2}{s}}}$

Alphabet (from Sofia)

3 letters

$$A = \{L\} = \{m, s, 4m^2 - s\} \longleftrightarrow \{x, 1+x, 1-x\}$$

(Sofia computes)

Total runtime= 0.042703

$$-\frac{1}{16\,\pi^4\,\sqrt{(4\,m^2 - s)\,s}}$$

rational prefactor

Basis is

$$T(x) = \sum_{i=1}^{17} c_i \mathcal{B}_i(x) = P(x) \times [c_{ij} L_i \otimes L_j + (a_j + b_j \pi) L_j + d_1 \pi^2 + d_2 \zeta_3]$$

9 terms weight 2

6 terms weight 1

2 numbers

17 constants

- Need T(x) and $B_i(x)$ to high precision

Numerical Feynman Integration

- AMFlow: numerical evaluation for Feynman loop integrals to high precision

Integration-by-Parts (IBP) [Chetyrkin, Tkachov, 1981]

Rational functions — Master integrals (basis)

$$\mathcal{I}(p_i, m_e) = \int \prod_{j=1}^{L} \mathrm{d}^D k_j \frac{N(p_i, k_j)}{\prod_{e=1}^{E}(q_e^2 - m_e^2 + i\varepsilon)} \Rightarrow \mathcal{I} = \sum_a R_a(p_i \cdot p_j, m_e)\, \mathrm{MI}_a$$

Auxiliary mass flow method (AMFLow) [Liu, Ma, 1801.10523; 2107.01864; 2201.11669; 2201.11637]

1. Introduce an auxiliary mass η to some of the propagator denominators
2. Set up closed differential equations w.r.t η using IBPs
3. Solve the differential equations numerically with boundary conditions $\eta \to \infty$

Timing on 1-loop triangle

- each phase space point takes 5-10 CPU-min for 30 signficiant digits

Numerical basis functions

$$T(x) = \sum_{i=1}^{17} c_i \mathcal{B}_i(x) = P(x) \times [c_{ij} L_i \otimes L_j + (a_j + b_j \pi) L_j + d_1 \pi^2 + d_2 \zeta_3]$$

- Rational prefactor evaluation is instant
- Weight-2 symbols can be integrated into closed form expressions
    - FiberSymbol in Polylogtools can do this
    - Numerical evaluation is very fast

Integrals of highter weight symbols are not always known

- Can always integrate numerically along a path

$$\tilde{B} = \sum_{i=1}^{|\tilde{A}|} c_{i_1,i_2,i_3,i_4} \int_0^1 d\log L_{i_4}(\lambda_4) \int_0^{\lambda_4} d\log L_{i_3}(\lambda_3) \int_0^{\lambda_3} d\log L_{i_2}(\lambda_2) \int_0^{\lambda_2} d\log L_{i_1}(\lambda_1)\,,$$

- Individual terms may be path dependent, but final result is integrable
- Can do the first and last integral ananlytically
    - Reduce weight by 2: speeds up integration tremendously
- Sometimes analytic integrals can be so complicated that it is faster to do the integral numerically
    - More work needed on efficient numerical evaluation

Matrix inversion

$$T(x) = \sum_{i=1}^{17} c_i \mathcal{B}_i(x)$$

- Evaluate both sides of this equation at 17 points
- Solve 17 linear equations for coefficients
  - i.e. invert the matrix

$$M_{ij} = \mathcal{B}_i(\mathbf{x}_j) \Rightarrow c_i = (M^{-1})_{\mathrm{ij}} \cdot T(x_j)$$

Problems:

1. No way to impose that c_j are rational
   - Cannot every resolve functions that differ by irrational constants

$$2\,[\ln x] + 3\,[\pi \ln x] = (1+\pi)[\ln x] + \left(2 + \frac{1}{\pi}\right)[\pi \ln x]$$

   - Even with infinite precision, will not find the exact analytic answer

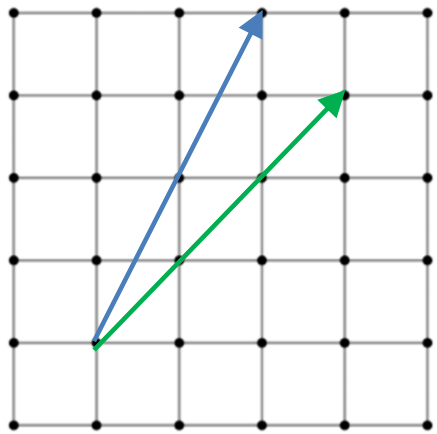
2. Matrix inversion loses precision very fast
   - Controlled by condition number

$$\kappa(M) = \|M\| \|M^{-1}\|\,.$$

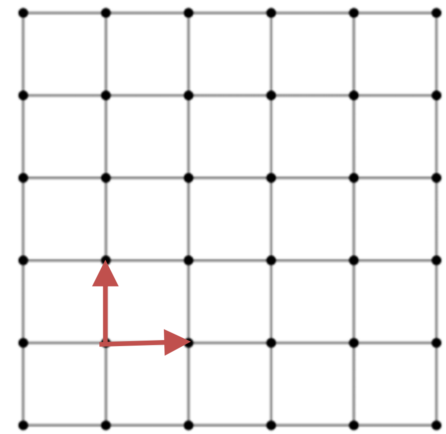
   - Generally our matrices will be ill-conditioned
     (they come from smooth functions)

Lattice reduction

vectors span a lattice



$$\vec{u}_1^2 + \vec{u}_2^2 = 24$$



$$\vec{v}_1^2 + \vec{v}_2^2 = 2$$

- Lattice reduction finds another set of vectors for same lattice
- Can minimize some norm (length of lattice vectors)
- NP-Hard problem: no polynomial-time algorithm for truly best solution
- Efficient algorithms exist to find what is almost always the minimum

Lattice reduction

Rational number coefficients can be fit for numbers using lattice reduction

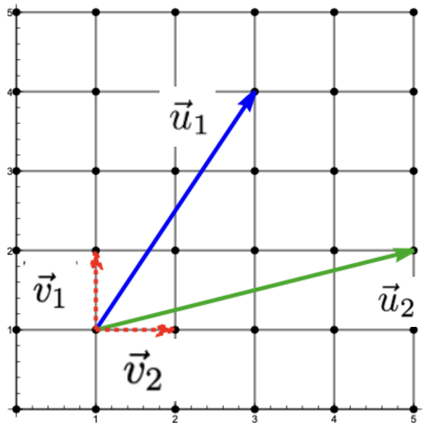
$$f = -\int_0^1 du \int_0^1 dv \frac{\log(1-uv) + v\log(1-u)}{uv} = \frac{\pi^2}{6} + \zeta_3 \,. = c_1\pi^2 + c_2\zeta_3 = \vec{c}\cdot\vec{v}$$

f=2.847 ⟷ Assume 4 digits known ⟷ π^2=9.870 ζ_3=1.202

Multiply by 10^3 and put into a matrix

$$\begin{pmatrix} \vec{u}_1 \\ \vec{u}_2 \\ \vec{u}_3 \end{pmatrix} = \begin{pmatrix} 10^3 f & 1 & & \\ 10^3\pi^2 & & 1 & \\ 10^3\zeta_3 & & & 1 \end{pmatrix} = \begin{pmatrix} 2847 & 1 & 0 & 0 \\ 9870 & 0 & 1 & 0 \\ 1202 & 0 & 0 & 1 \end{pmatrix} \xrightarrow[\text{lattice reduction}]{} \begin{pmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vec{v}_3 \end{pmatrix} = \begin{pmatrix} 0 & -6 & 1 & 6 \\ 3 & -11 & 5 & -15 \\ 62 & 4 & -2 & 7 \end{pmatrix}$$

vectors span a lattice



- Lattices are the same so v in the span of u

$$\vec{v}_1 = -6\vec{u}_1 + \vec{u}_2 + 6\vec{u}_3.$$

First component

$$0 = 10^3 \times (-6f + \pi^2 + 6\zeta^3) \quad ✓$$

Precision requirements

Rational number coefficients can be fit for numbers lattice reduction

$$f = -\int_0^1 du \int_0^1 dv \frac{\log(1-uv) + v\log(1-u)}{uv} = \frac{\pi^2}{6} + \zeta_3 = c_1\pi^2 + c_2\zeta_3 = \vec{c}\cdot\vec{v}$$

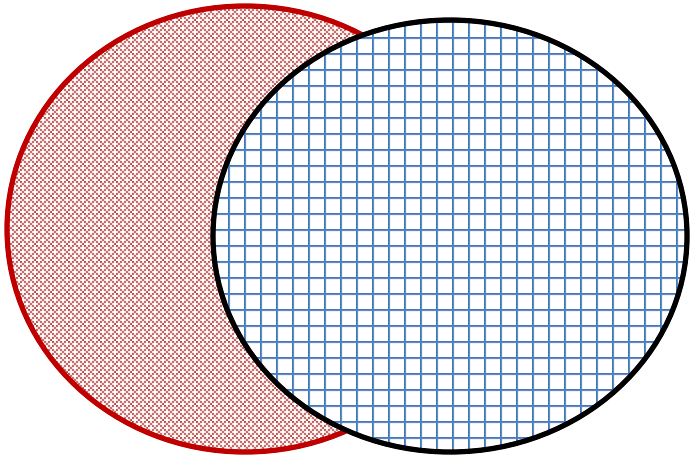
$$\vec{v} = (\pi^2, \zeta_3)$$

Q: how many digits of f are required to fit rational c_1 and c_2?

assume size of c's

$$c_1 \sim c_2 \lesssim 10^R$$

- Multiple solutions imples that $\vec{c}\cdot\vec{v} = 0$
- Multiply by GCD so c_1 and c_2 are integers
  - There are $(10^R)^n = 10^{2R}$ vectors (c_1,c_2)
- Assume d digits of precision on v
  - $\vec{c}\cdot\vec{v}$ produce 10^{2R} d digit numbers
- There are only 10^d d-digit numbers in all
- Need precision $d > nR$ to fit pure numbers
  - Information content must be sufficient



Precision requirements

For fitting *functions* we can sample at multiple points

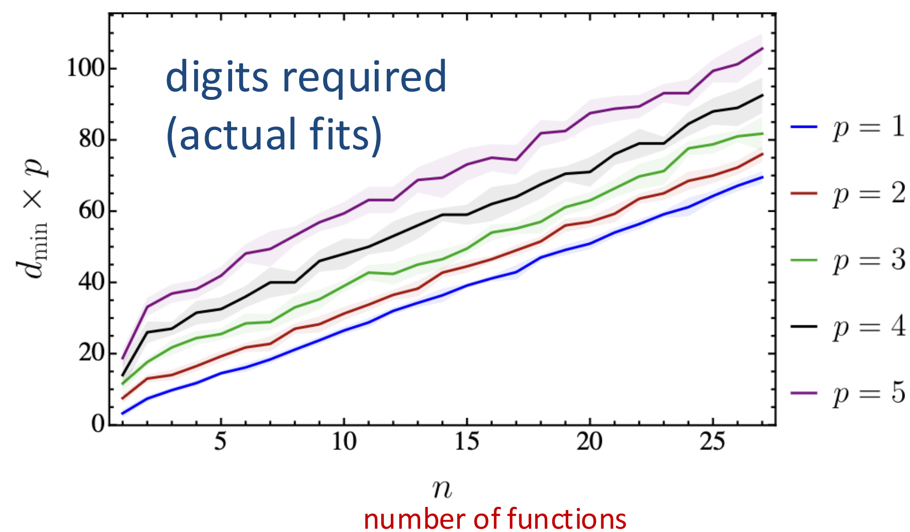
$$f(x) = G(0,1;x) - G(1,-1;x) \qquad x_1 = 4/10, x_2 = 9/10$$

$$\mathcal{B}(x) = \{G(1,0;x), G(0,1;x), G(0,-1;x), G(1,-1;x)\}$$

$$M = \text{round}\; 10^s \left(\begin{array}{c|c} f(\mathbf{x_1}) \cdots f(\mathbf{x_p}) & \\ \mathcal{B}_1(\mathbf{x_1}) \cdots \mathcal{B}_1(\mathbf{x_p}) & 10^{-s}\, \mathbb{I}_{n+1} \\ \vdots & \\ \mathcal{B}_n(\mathbf{x_1}) \cdots \mathcal{B}_n(\mathbf{x_p}) & \end{array} \right) = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \\ \mathbf{v}_4 \\ \mathbf{v}_5 \end{pmatrix} = \begin{pmatrix} -35 & -24 & 1 & 0 & 0 & 0 & 0 \\ 92 & 154 & 0 & 1 & 0 & 0 & 0 \\ -45 & -129 & 0 & 0 & 1 & 0 & 0 \\ 36 & 75 & 0 & 0 & 0 & 1 & 0 \\ -10 & -106 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

- With p points and d digits
  - net digits of information is p × d
- Expected digits needed

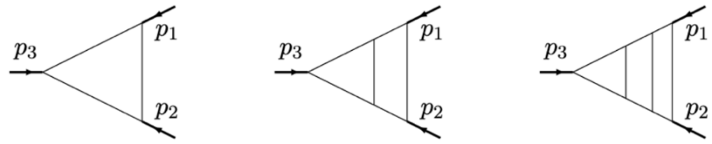
digits of precision req'd → $d \lesssim \frac{nR}{p}$ ← # basis functions (n), size of integers (R), number of points (p)



We can trade off digits of precision for number of points
- Will work even if only a few digits of precision are available!

Example 1: Triangles

Triangle ladder diagrams



$$z\bar{z} = p_2^2/p_1^2,$$
$$(1-z)(1-\bar{z}) = p_3^2/p_1^2$$

exact results known

$$T_1(z) = \frac{1}{z-\bar{z}}\left[2\mathrm{Li}_2(z) - \mathrm{Li}_2(\bar{z}) + \log(z\bar{z})\log\left(\frac{1-z}{1-\bar{z}}\right)\right],$$

$$T_2(z) = \frac{1}{(1-z)(1-\bar{z})(z-\bar{z})}\Big[6\mathrm{Li}_4(z) - 6\mathrm{Li}_4(\bar{z}) - 3\log(z\bar{z})\left(\mathrm{Li}_3(z) - \mathrm{Li}_3(\bar{z})\right)$$
$$+\frac{1}{2}\log^2(z\bar{z})\left(\mathrm{Li}_2(z) - \mathrm{Li}_2(\bar{z})\right)\Big],$$

$$T_3(z) = \frac{1}{(1-z)^2(1-\bar{z})^2(z-\bar{z})}\Big[20\mathrm{Li}_6(z) - 20\mathrm{Li}_6(\bar{z}) - 10\log(z\bar{z})\left(\mathrm{Li}_5(z) - \mathrm{Li}_5(\bar{z})\right)$$
$$+\log^2(z\bar{z})\left(\mathrm{Li}_4(z) - \mathrm{Li}_4(\bar{z})\right) - \frac{1}{6}\log^3(z\bar{z})\left(\mathrm{Li}_3(z) - \mathrm{Li}_3(\bar{z})\right)\Big]$$

Full alphabet (from SOFIA)

$$A_{1,2} = \left\{z\bar{z},\ (1-z)(1-\bar{z}),\ z-\bar{z},\ \frac{\bar{z}}{z},\ \frac{1-z}{1-\bar{z}}\right\}$$

Simplified alphabet (quicker for testing)

$$A_3^\star = \left\{z\bar{z},\ (1-z)(1-\bar{z}),\ \frac{\bar{z}}{z},\ \frac{1-z}{1-\bar{z}}\right\}$$

Allow all possibilities up to weight 6

Weight-0: 1

Weight-1: $G(a_1,x),\ \pi$

Weight-2: $G(a_1,a_2,x),\ \pi\times G(a_1,x),\ \zeta_2$

Weight-3: $G(a_1,a_2,a_3,x),\ \pi\times G(a_1,a_2,x),\ \zeta_2\times G(a_1,x),\ \zeta_3$

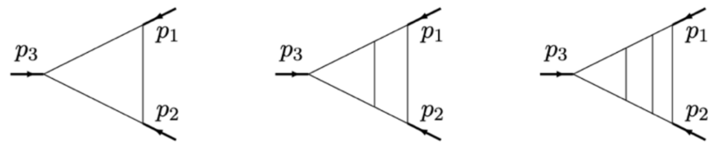
Weight-4: $G(a_1,a_2,a_3,a_4,x),\ \pi\times G(a_1,a_2,a_3,x),\ \zeta_2\times G(a_1,a_2,x),\ \pi^3\times G(a_1,x),$
$\zeta_3\times G(a_1,x),\ \zeta_4$

Weight-5: $G(a_1,a_2,a_3,a_4,a_5,x),\ \pi\times G(a_1,a_2,a_3,a_4,x),\ \zeta_2\times G(a_1,a_2,a_3,x),$
$\pi^3\times G(a_1,a_2,x),\ \zeta_3\times G(a_1,a_2,x),\ \zeta_4\times G(a_1,x),\ \zeta_5,\ \zeta_2\times\zeta_3$

Weight-6: $G(a_1,a_2,a_3,a_4,a_5,a_6,x),\ \pi\times G(a_1,a_2,a_3,a_4,a_5,x),\ \zeta_2\times G(a_1,a_2,a_3,a_4,x),$
$\pi^3\times G(a_1,a_2,a_3,x),\ \zeta_3\times G(a_1,a_2,a_3,x),\ \zeta_4\times G(a_1,a_2,x),\ \zeta_5\times G(a_1,x),$
$\zeta_2\zeta_3\times G(a_1,x),\ \pi^5\times G(a_1,x),\ \zeta_6,\ \zeta_3^2$

Example 1: Triangles

Triangle ladder diagrams



Pick random points in (unphysical) Euclidean region $0 < z < \bar{z} < 1$

- Makes basis functions real
- Imaginary numbers are fine, just technically complicated python implemntation

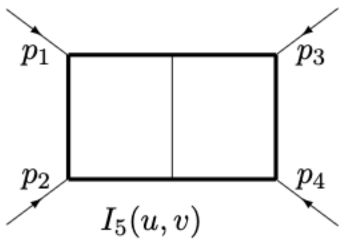
Diagram	AMFLOW point time	Transcendental weights	# points sampled	Basis size	Reduction time	
One-loop	15.6 CPU-min	≤ 2	5	full(32) simplified(26) uniform(25)	<1s <1s <1s	3 digits
Two-loop	1.1 CPU-h	≤ 4	100 100 60	full(488) simplified(393) uniform(366)	9.6 min 10.7 min 3.5 min	20 digits
Three-loop	5.7 CPU-h	≤ 6	- - 200	full(1373) simplified(972) uniform(806)	- - 1.1 h	20 digits

Results with timing

- Rate limiting step is AMFlow (computation of full function)
- Could be sped up (additonal points much faster than first)

Example 2: Double box

Loop



12 independent letters

$$\tilde{A} = \left\{u, v, 1+u, 1+v, u+v, 1+u+v, \frac{\beta_u - 1}{\beta_u + 1}, \frac{\beta_v - 1}{\beta_v + 1}, \frac{\beta_{uv} - 1}{\beta_{uv} + 1}, \frac{\beta_{uv} - \beta_u}{\beta_{uv} + \beta_u}, \frac{\beta_{uv} - \beta_v}{\beta_{uv} + \beta_v}, \frac{\beta_{uv} - \beta_u\beta_v}{\beta_{uv} + \beta_u\beta_v}\right\}$$

with $\beta_u = \sqrt{1+u}$, $\beta_v = \sqrt{1+v}$ and $\beta_{uv} = \sqrt{1+u+v}$.

- 12^4 = 20,736 weight-4 symbols +(?) lower weight terms
- square root letters are hard to integrate analytically

method 1:
rationalize the square roots

$$u = \frac{(1-w^2)(1-z^2)}{(w-z)^2} \quad \text{and} \quad v = \frac{4wz}{(w-z)^2},$$

$$\tilde{A}_2 = \{w, z, 1 \pm w, 1 \pm z, w \pm z, 1 \pm wz, 1 \pm w \mp z + wz\}\,.$$

- Now symbols can be integrated analytically
  - Takes FiberSymbol hours to integrate
  - Result is hundreds or thousands of terms
- GiNaC can get numbers out, but very slow

method 2:
numerically integate along a contour

- integrate first and last symbol analytically
- need to be careful with branch cuts
  - euclidean region requires some thought
- using integrable contributions helps a lot

Example 2: Double box

Landau boostrap

Analytic regression

relatively easy

Easy ↓ Hard

Constraints	# Coeffs
All Symbols	20736
Integrability	6993
Galois symmetry	861
Physical branch cuts	161
Genealogical constraints	28
α-positive thresholds	6

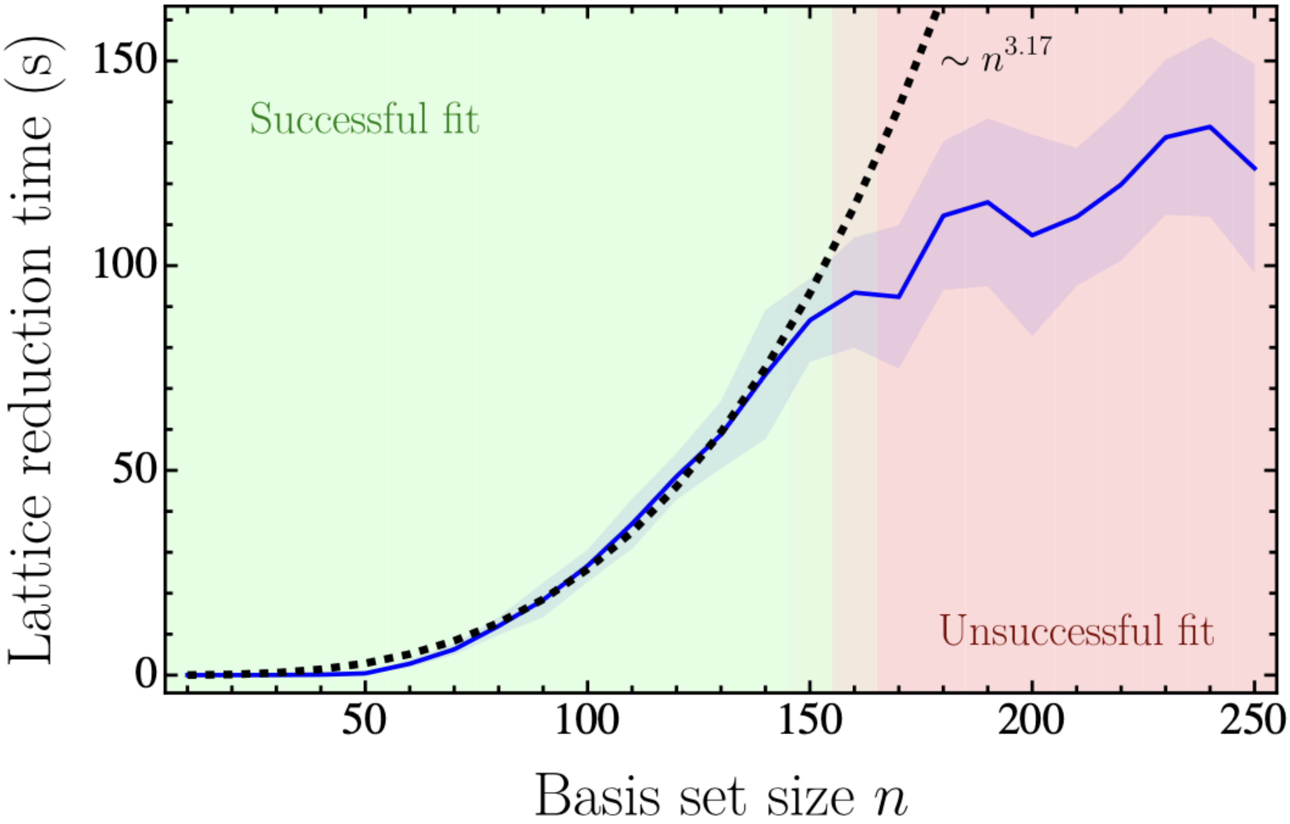
Hard ↑ Easy

✓ 200pts 17 digits, $t \backsim 30$s

✓ 200pts, 8 digits, $t < 1$s

✓ 200pts, 6 digits, $t < 1$s

Double box: limitations



Should work with more functions

- With our compute, hard to succeed with more than n $\lesssim$ 200

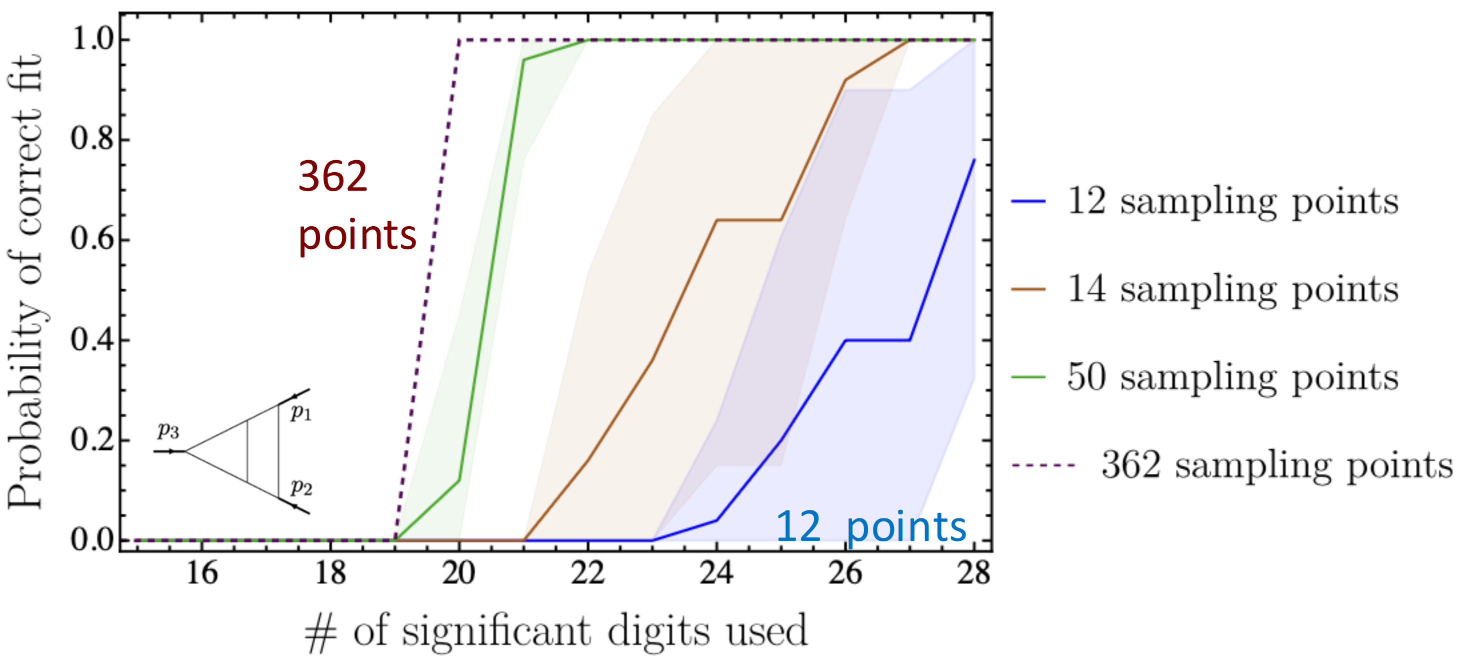
Choosing points

- Before we said you can get away with fewer digits if you use more points

basis functions

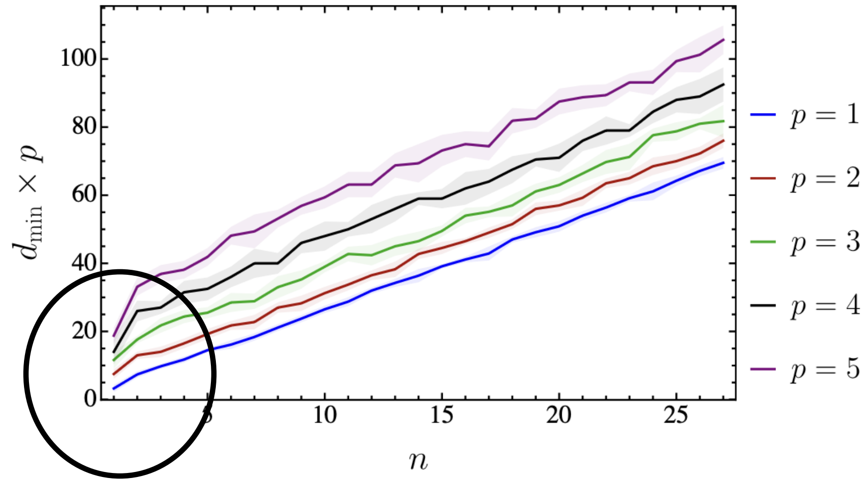
size of integers

digits of precision req'd → $$d \lesssim \frac{nR}{p}$$ ← number of points



- Can never succeed below some digit lower bound
- Why did scaling fail?

Choosing points



offset near d=0

$$d_{\min} \approx R_{\text{eff}} \frac{n}{p} + d_0$$

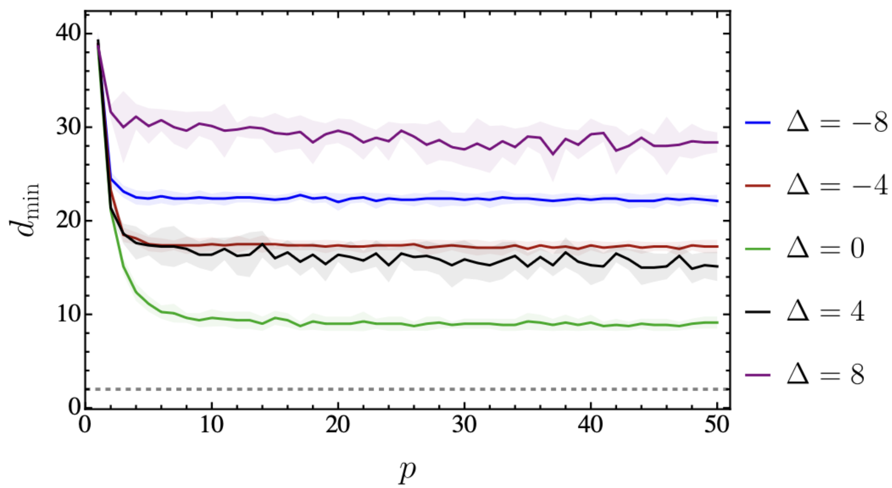
- will never work with 1 or 2 digits

$$f(p_j) = \{103.2, \quad 2.5, \quad 2.3\}$$

unrecovarable information loss if we truncate to 2 digits

choose points in a range

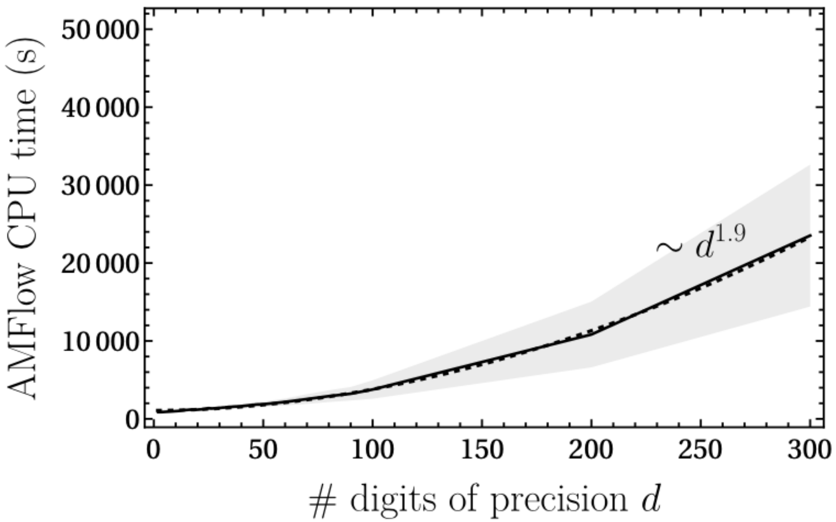
$$\tfrac{1}{2} - 10^{\Delta} \leq x \leq \tfrac{1}{2} + 10^{\Delta}$$



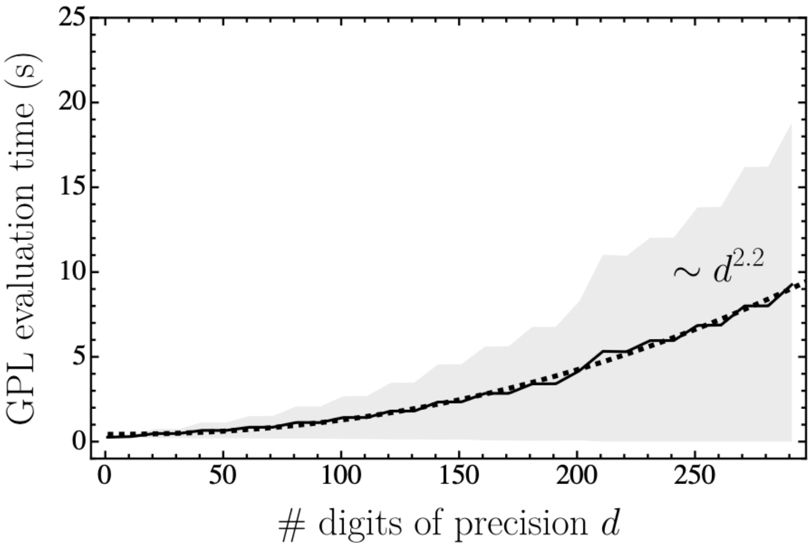
- want to choose points
  - not too close (lose information)
  - not too far (need more digits)
  - Need 10-20 digits at least

Double box: timing

AMFlow + Kira (for IBP)



GPL evaluation (GINSH)



$$t(p,n)/\mathrm{ns} \approx \underbrace{10^9 \cdot p \cdot d_{\min}^2(n)}_{\textsc{AMFlow}} + \underbrace{10^4 \cdot n \cdot p \cdot d_{\min}^2(n)}_{\textsc{Ginsh}} + \underbrace{p \cdot n^4}_{\text{fitting}},$$

- Trading digits for points makes scaling go from quadradic to linear!

Application to EECs

Toward the Analytic Bootstrap of Energy Correlators
Jianyu Gong[a,b], Andrzej Pokraka[c], Kai Yan[a,b], Xiaoyuan Zhang[d]
2509.22782

Consider some types of energy-energy correlators (unequal energy weights)

$$\frac{\mathrm{d}\sigma}{\mathrm{d}x_{12}\cdots \mathrm{d}x_{(N-1)N}} \equiv \sum_m \sum_{1\le i_1,\cdots i_N\le m} \int \mathrm{d}\sigma_m \times \prod_{1\le k\le N} \frac{E_{i_k}}{Q} \prod_{1\le j<l\le N} \delta\left(x_{jl} - \frac{1-\cos\theta_{i_j i_l}}{2}\right).$$

Uses of lattice reducion

- Find linear dependence among basis functions

$$G(z) = D(z) + a_{0,11}\, g_1(z) + a_{0,12}\, g_2(z) + a_{1,12}\, g_3(z) + a_{2,12}\, g_4(z) + c_{1,12}\, g_5(z) + d_{2,12}\, g_6(z)$$

$$\begin{aligned} g_3(z) = \frac{1}{(z-1)^2 z^2 (\bar z-1)^2 \bar z^2}\Big[& z^4\bar z^2 - z^4\bar z + 8z^3\bar z^3 - 14z^3\bar z^2 + 8z^3\bar z - z^3 + z^2\bar z^4 - 14z^2\bar z^3 \\ &+ 24z^2\bar z^2 - 14z^2\bar z + z^2 + \big(2z^4\bar z^3 - 3z^4\bar z^2 + z^4\bar z + 2z^3\bar z^4 - 8z^3\bar z^3 + 9z^3\bar z^2 - 4z^3\bar z + z^3 \\ &- 3z^2\bar z^4 + 9z^2\bar z^3 - 12z^2\bar z^2 + 9z^2\bar z - 3z^2 + z\bar z^4 - 4z\bar z^3 + 9z\bar z^2 - 8z\bar z + 2z + \bar z^3 - 3\bar z^2 \\ &+ 2\bar z\big)\log((z-1)(\bar z-1)) + \big(-2z^4\bar z^3 + 3z^4\bar z^2 - z^4\bar z - 2z^3\bar z^4 + 8z^3\bar z^3 - 9z^3\bar z^2 + 2z^3\bar z \\ &+ 3z^2\bar z^4 - 9z^2\bar z^3 + 6z^2\bar z^2 - z\bar z^4 + 2z\bar z^3\big)\log(z\bar z) - z\bar z^4 + 8z\bar z^3 - 14z\bar z^2 + 8z\bar z - \bar z^3 + \bar z^2\Big] \end{aligned}$$

- Treat $\{g_i\}$ as transcendental basis and $a_{i,j}, b_{i,j}, c_{i,j}, d_{i,j}$ as coefficients to determine
- (1). Run lattice reduction among $\{g_i(z)\}$ to get a linear-independent basis $\{\tilde g_i(z)\}$
- (2). Run lattice reduction among both $G(z)$ and $\{\tilde g_i(z)\}$

In this case, 2 numerical points and 13 digits fix all six parameters

Pros and cons: Landau bootstrap

Landau boostrap

- Can eliminate large swaths of symbols with physical constraints
- Don’t need to do integrals
- Leads to new deep understanding in what amplitdues are

- Requires subtle understanding of singularities
  - Analytic structure of amplitudes
  - Branch points, euclidean regions
  - Algebraic geometry
- Rational prefactors not fixed by singularities alone
- Often still requires some integration at the end

Pros and cons: analytic regression

Analytic regression with lattice reduction

- Easy to automate
- Can work for any functions
  - not just polylogs with symbols
  - elliptic polylarithms? no problem!
  - cross sections, EECs, etc.
- Can find linear dependencies
- Can trade digits of accuracy for points
  - Scaling $t \sim n^2$ or worse to $t \sim n$

- Becomes computationally challenging above $n \sim 200$
  - AMFlow scales like $(\text{\# digits})^2$
  - lattice reduction scales like $(\text{\# constants})^4$
- Minimum number of digits needed ($\sim$ 5 or 6)

$$f(\mathbf{x}_j) = \sum_{i=1}^{n} c_i \, \mathcal{B}_i(\mathbf{x}_j) \qquad c_i \in \mathbb{Q}$$

→ problem solved

Conclusions

Happy Birthday Symbol