A new method for the reconstruction of rational functions

Xiao Liu

University of Oxford

Based on e-print: 2306.12262

QCD@LHC 2023
5 September 2023, Durham

Outline

High precision particle physics

➢ Era of Precision

- Theoretical predictions & experimental measurements
- Test particle physics Standard Model & probe signals of New Physics
- Higher order corrections required

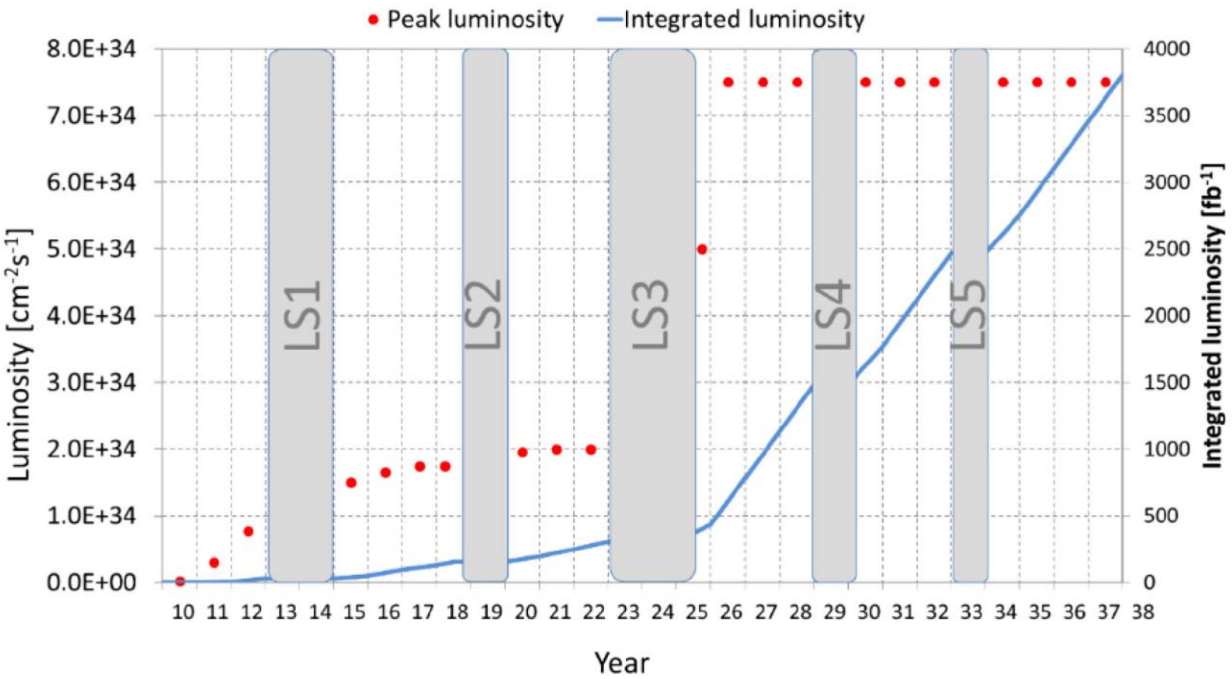


*Figure from [Apollinari, Alonso, Bruning, et al, 2015]

High precision particle physics

➢ Multiloop scattering amplitudes

- Construct the amplitude

$$\mathcal{A} = \sum c_i I_i$$

  - I_i: scalar Feynman integrals in dimensional regularization

$$I(\vec{\nu}) = \int \prod_{i=1}^{L} \frac{\mathrm{d}^D \ell_i}{\mathrm{i}\pi^{D/2}} \frac{\mathcal{D}_{K+1}^{-\nu_{K+1}} \cdots \mathcal{D}_N^{-\nu_N}}{(\mathcal{D}_1 + \mathrm{i}0)^{\nu_1} \cdots (\mathcal{D}_K + \mathrm{i}0)^{\nu_K}}$$

- Compute the scalar integrals: reduction + computation
  - reduction: express all the scalar integrals in terms of a smaller set of independent integrals (master integrals)

$$I_i = \sum_j b_{ij} M_j$$

  - computation: compute the master integrals as expansions in the dimensional regulator $\epsilon = (4 - D)/2$

$$M_j = \sum_{l=-2L} d_{jk} \epsilon^k$$

Feynman integrals

➢ Integration-by-parts (IBP) reduction [Chetyrkin and Tkachov, Nucl. Phys. B, 1981] [Laporta, Int. J. Mod. Phys. A, 2000]

$$0 = \int \prod_{i=1}^{L} \frac{\mathrm{d}^D \ell_i}{\mathrm{i}\pi^{D/2}} \frac{\partial}{\partial \ell_j^\mu} \left(\frac{v_k^\mu}{\mathcal{D}_1^{\nu_1} \cdots \mathcal{D}_m^{\nu_m}} \right)$$

- Number of master integrals is finite. [Smirnov and Petukhov, Lett. Math. Phys., 2011]
- Computer programs
  - AIR [Anastasiou and Lazopoulos, JHEP, 2004] FIRE [Smirnov, JHEP, 2008] [Smirnov, Smirnov, Comput. Phys. Commun., 2013] [Smirnov, Comput. Phys. Commun., 2015] [Smirnov and Chuharev, Comput. Phys. Commun., 2020] Reduze [Studerys, Comput. Phys. Commun., 2010] [Manteuffel and Studerus, e-Print: 1201.4330] Kira [Maierhofer, Usovitsch and Uwer, Comput. Phys. Commun., 2018] [Klappert, Lange, Maierhofer and Usovitsch, Comput. Phys. Commun., 2021] LiteRed [Lee, 2012] [Lee, 2014] NeatIBP [Wu, Boehm, Ma, et al, 2305.08783]

Finite field arithmetic [Manteuffel and Schabinger, Phys. Lett. B, 2015] [Peraro, JHEP, 2016]

- Motivation: to avoid intermediate expression swelling
  - numerical sampling + reconstruction over finite fields
- Univariate polynomials: Newton's interpolation formula

$$f(x) = a_0 + (x - x_0)\Big(a_1 + (x - x_1)\Big(a_2 + (x - x_2)(\cdots + (x - x_{R-1})a_R)\Big)\Big)$$

- Univariate rational functions: Thiele's interpolation formula

$$f(x) = a_0 + (x - x_0)\left(a_1 + (x - x_1)\left(a_2 + (x - x_2)\left(\cdots + \frac{x - x_{R-1}}{a_R}\right)^{-1}\right)^{-1}\right)^{-1}$$

- Multivariate polynomials: recursively applying Newton's formula
- Multivariate rational functions
  - univariate rational functions + multivariate polynomials
- Programs: FiniteFlow [Peraro, JHEP, 2019] FireFly [Klappert and Lange, Comput. Phys. Commun., 2020]

Summary

Refined IBP systems:

syzygy equations [Gluza, Kajda and Kosower, Phys. Rev. D, 2011] [Larsen and Zhang, Phys. Rev. D, 2016]

block-triangular systems [Guan, **XL**, Ma, Chin.Phys.C, 2020]

More powerful linear solver:

Kira [Maierhofer, Usovitsch and Uwer, Comput. Phys. Commun., 2018]

RATRACER [Magerya, e-Print: 2211.03572]

Better interpolation methods [Klappert and Lange, Comput.Phys.Commun. 2020] [Belitsky, Smirnov, Yakovlev, 2023.02511]

More compact ansatz[Badger, Hansen, Chicherin, et al, JHEP 2021][Laurentis, Page, JHEP 2022][Abreu, Laurentis, Ita, et al, 2305.17056]

The method in this talk

$$\text{time} = \frac{\text{time for a single sample} \times \text{number of samples}}{\text{number of CPUs}}$$

Outline

I. Introduction

II. The method

III. Examples

IV. Summary and outlook

Motivation

➢ A simple observation

- Traditional strategy: reconstructing functions individually & neglecting common structures
- Example

$$f_i(x) = \left(\frac{1+x}{1-x}\right)^{i-1}, \quad i \in [1, 100]$$

  - approximately 200 samples using Thiele's interpolation formula
  - linear relations

$$(1-x)f_{i+1}(x) - (1+x)f_i(x) = 0, \quad i \in [1, 99]$$

  - ansatz + linear fit → 4 samples

$$(a_i + b_i x)f_{i+1}(x) + (c_i + d_i x)f_i(x) = 0$$

- Linear relations → common structures utilized → number of samples reduced

The method

General description

- Goal: all $n-1$ independent relations among k-variate functions $f_1(\vec{x}), \ldots, f_n(\vec{x})$
- Ansatz

$$Q_1(\vec{x})f_1(\vec{x}) + \cdots + Q_n(\vec{x})f_n(\vec{x}) = 0$$

  - $Q_i(\vec{x})$: polynomial of $\vec{x}$ $\Rightarrow$ specifying monomials $x_1^{\alpha_1} \cdots x_k^{\alpha_k}$

- Definition 1:

$$P(\{x_{i_1}, \ldots, x_{i_l}\}, m) := \{x_{i_1}^{\alpha_1} \cdots x_{i_l}^{\alpha_l} \,|\, \sum \alpha_i \leq m\}$$

  - $P(\{x_1\},0) = \{1\}$; $P(\{x_1\},2) = \{1, x_1, x_1^2\}$; $P(\{x_2, x_3\},1) = \{1, x_2, x_3\}$

- Definition 2:

$$P_1 \times P_2 := \{p_1 p_2 \,|\, p_1 \in P_1, p_2 \in P_2\}$$

  - $P(\{x_1\},2) \times P(\{x_2,x_3\},1) = \{1, x_1, x_1^2, x_2, x_1x_2, x_1^2x_2, x_3, x_1x_3, x_1^2x_3\}$

The method

- Divide variables $\vec{x}$ into r subsets $S_1, \dots, S_r$
  - 3-variate example: $S_1 = \{x_1\}, S_2 = \{x_2, x_3\}$
- For a specific $Q_i(\vec{x})$, given integers $\vec{z} = \{z_1, \dots, z_r\}$

$$M(\vec{z}) := P(S_1, z_1) \times \cdots \times P(S_r, z_r)$$

- In practice: use the same $\vec{z}$ for all the Q's
  - $\vec{z} = \{0,0\} \rightarrow M(0,0) = \{1\} \rightarrow a_1 f_1 + \cdots + a_n f_n = 0$
  - $\vec{z} = \{1,0\} \rightarrow M(1,0) = \{1, x_1\} \rightarrow (b_1 + c_1 x_1) f_1 + \cdots + (b_n + c_n x_1) f_n = 0$
  - $\vec{z} = \{0,1\} \rightarrow M(0,1) = \{1, x_2, x_3\} \rightarrow (d_1 + e_1 x_2 + g_1 x_3) f_1 + \cdots = 0$
- Generalized algorithm from [Guan, **XL**, Ma, Chin.Phys.C, 2020]
  - 1. start with $\sum z_i = 0$;
  - 2. for each solution of $\vec{z}$, make the ansatz and fit the unknowns
  - 3. test the number of independent relations: if sufficient, terminate; otherwise increase $\sum z_i$ by 1 and go back to step 2.

Summary

- Summary
  - build a generator of the numerical samples for the target functions
    - e.g., IBP system + linear solver
  - find the system of all the independent linear relations over a finite field
    - various ansatz of $Q_1(\vec{x})f_1(\vec{x}) + \cdots + Q_n(\vec{x})f_n(\vec{x}) = 0$
    - linear fit = samples ($N_{\text{sample}} \sim N_{\text{unknown}}$) + dense solve ($N_{\text{sample}} \times N_{\text{unknown}}$)
  - solve the linear system to obtain explicit solutions
    - traditional rational functions reconstruction strategy
    - additional finite fields + rational numbers reconstruction (Chinese Remainder Theorem + Wang's algorithm)

Outline

I. Introduction

II. The method

III. **Examples**

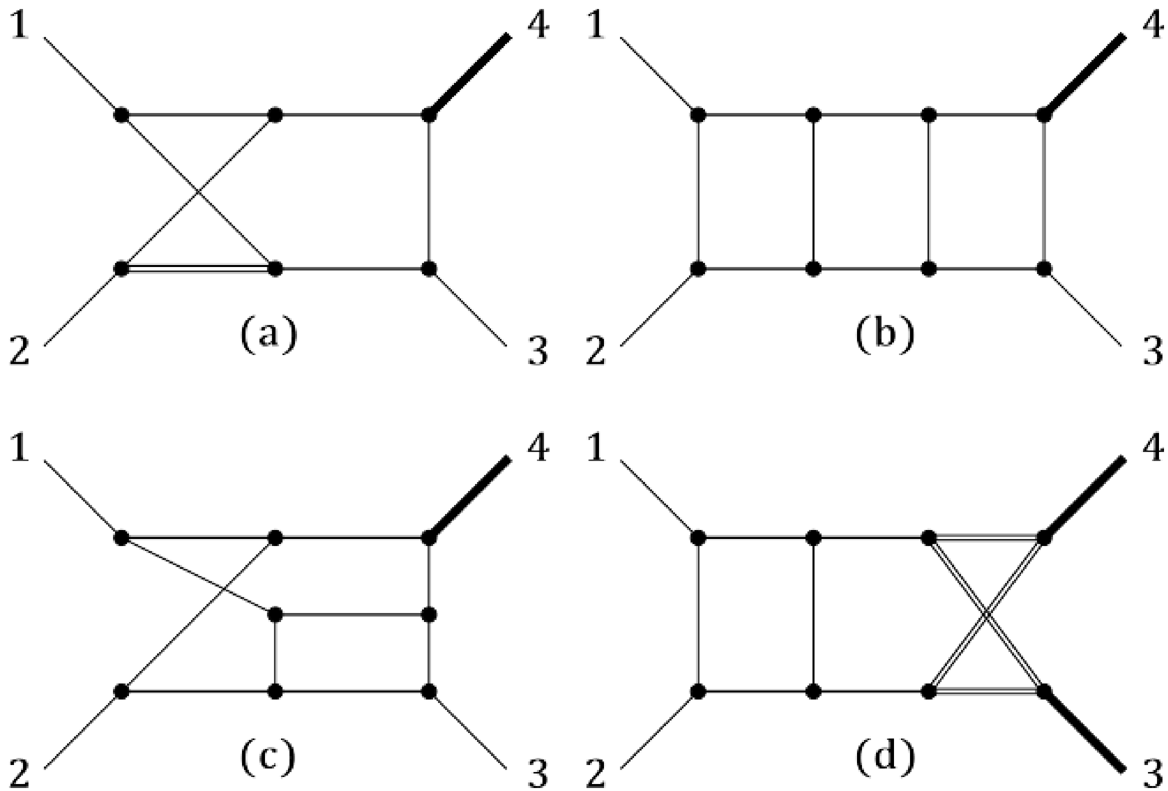
IV. Summary and outlook

Examples

- Reduction coefficients of Feynman integrals or amplitudes

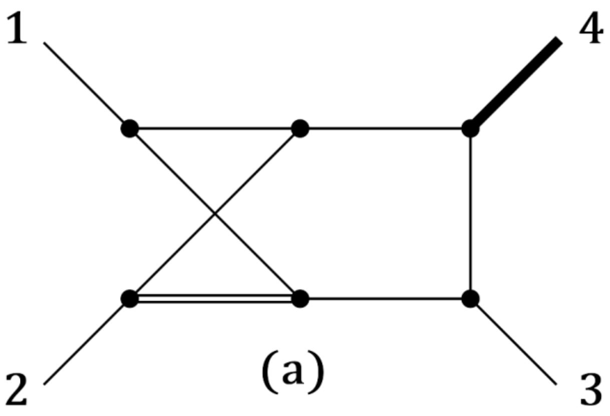
$$\mathcal{A} = f_1\mathcal{M}_1 + \cdots + f_n\mathcal{M}_n$$

  - a common set of denominators reflecting the singularities
  - auxiliary function $f_{n+1} = 1$



Examples

- Topology (a): two-loop amplitude of the mixed QCD-electroweak correction to $pp \to Z + j$ [Bargiela, Caola, Chawdhry, **XL**, to appear]
- Setup
  - $m_Z^2 = 1, m_W^2 = 7/9$
  - remaining: $\{\epsilon, s_{12}, s_{13}\}$
  - 56 master integrals $\Rightarrow$ 56 rational functions
  - LiteRed + FiniteFlow



(a)

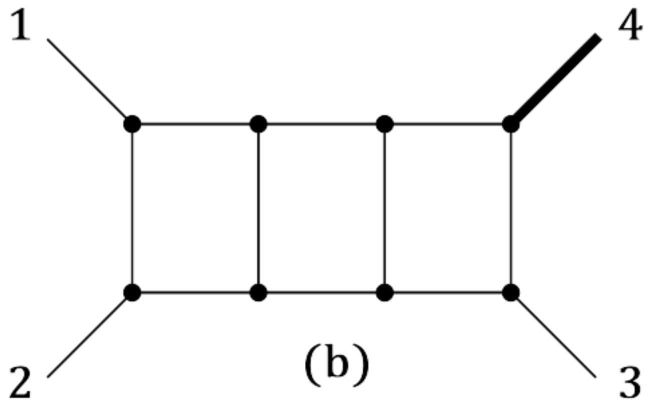
- Details
  - $S_1 = \{\epsilon\}, S_2 = \{s_{12}, s_{13}\}$
  - $z_1 + z_2 = 6$
  - 1+2 finite fields with 64-bit prime numbers
  - samples: $18326 \times 3 \to 2199 + 1561 \times 2 \Rightarrow$ a factor of 10.3
  - computational cost: 4.6h $\to$ 0.44h + 0.03h $\Rightarrow$ a factor of 9.8

Examples

$\vec{z}$	N_{unknown}	N_{sample}	N_{relation}	$N2_{\text{sample}}$
{0, 0}	57	58	0	-
{1,0}	114	115	0	-
{0,1}	171	172	1	4
{2,0}	171	172	0	-
{1,1}	340	341	0	-
{0,2}	339	340	1	9
{3,0}	228	229	0	-
…				
{2,2}	1014	1015	1	31
…				
{2,3}	1680	1681	20	1394
…				
{3,3}	2198	2199	33	1561
summary		2199	56	1561

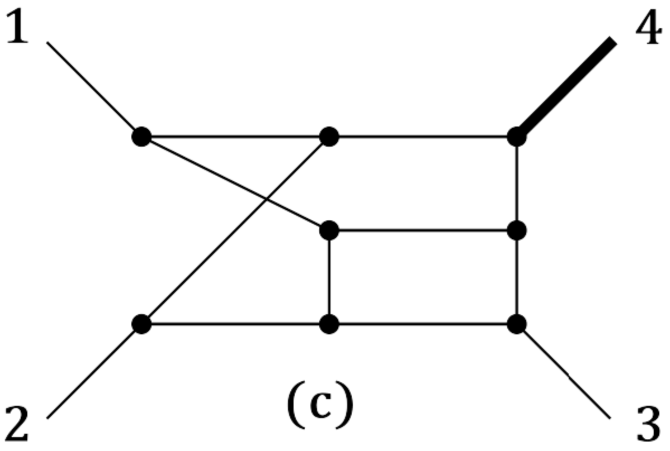
Examples

- Topology (b): an integral with rank-6 numerator
- Setup
  - $p_4^2 = 1$
  - remaining: $\{\epsilon, s_{12}, s_{13}\}$
  - 83 master integrals $\Rightarrow$ 83 rational functions
  - NeatIBP + FiniteFlow
- Details
  - $S_1 = \{\epsilon\}, S_2 = \{s_{12}, s_{13}\}$
  - $z_1 + z_2 = 8$
  - 1+2 finite fields
  - samples: $48574 \times 3 \rightarrow 6010 + 4599 \times 2$ $\Rightarrow$ a factor of 9.6
  - computational cost: 78.5h $\rightarrow$ 8.03h + 0.12h $\Rightarrow$ a factor of 9.6



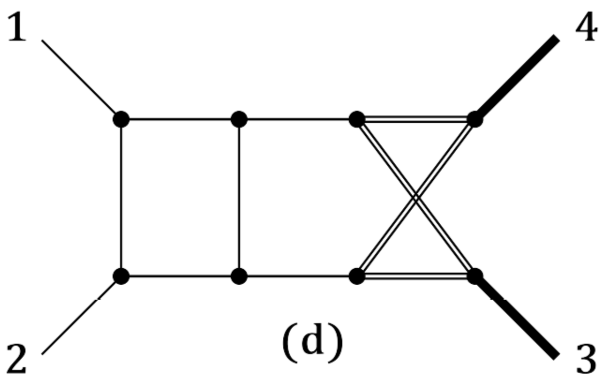
Examples

- Topology (c): differential equations of master integrals w.r.t. Mandelstam variables [Kotikov, Phys. Lett. B, 1991] [Henn, Phys. Rev. Lett., 2013] : $\frac{\partial}{\partial s_{12}} \vec{M}, \frac{\partial}{\partial s_{13}} \vec{M}$
- Setup
  - $p_4^2 = 1$
  - remaining: $\{\epsilon, s_{12}, s_{13}\}$
  - 280 master integrals
  - LiteRed + FiniteFlow
- Details
  - $S_1 = \{\epsilon\}, S_2 = \{s_{12}, s_{13}\}$
  - $z_1 + z_2 \leq 8$
  - 1+5 finite fields
  - samples: $391937 \times 6 \rightarrow 9612 + 6810 \times 5 \Rightarrow$ a factor of 54
  - computational cost: 450728h* → 8369h + 180h ⇒ a factor of 53



Examples

- Topology (d): differential equations with respect to internal squared masses
- Extensively involved in the auxiliary mass flow method [**XL**, Ma, Wang, Phys.Lett.B., 2018] [XL, Ma, Comput.Phys.Commun., 2023]
- Setup
  - $p_3^2 = p_4^2 = 1, s_{12} = 10, s_{13} = -22/9$
  - remaining: $\{\epsilon, m^2\}$
  - 336 master integrals
  - LiteRed + FiniteFlow



(d)

- Details
  - $S_1 = \{\epsilon\}, S_2 = \{m^2\}$
  - $z_1 + z_2 \leq 5$
  - 1+32 finite fields
  - samples: $14362 \times 33 \rightarrow 1414 + 1248 \times 32 \Rightarrow$ a factor of 11.5
  - computational cost: 3230h → 281h + 59h ⇒ a factor of 9.5

Examples

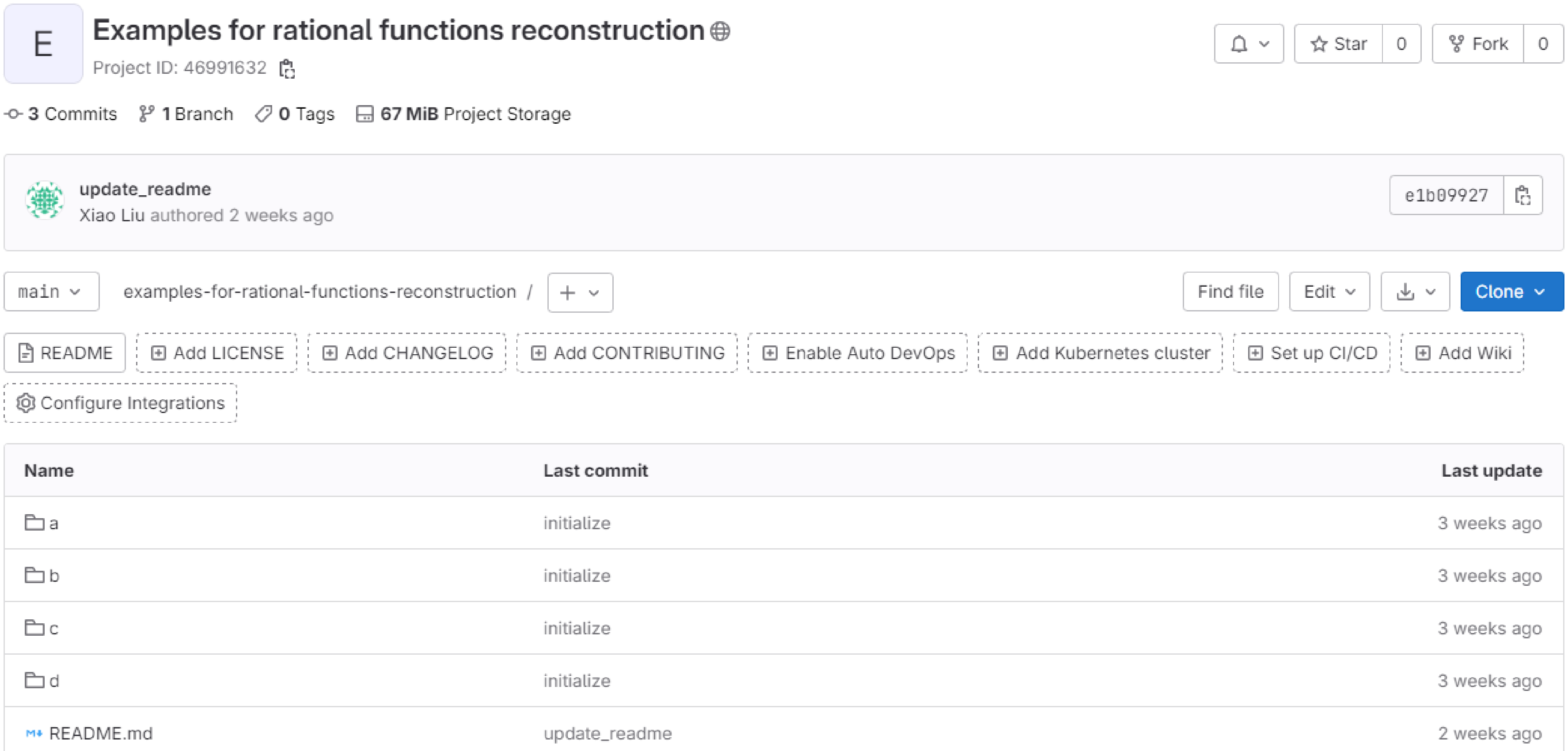
- More details

Topology	d_{Num}	d_{Den}	d_{Rel}	N_{IBP}	N_{Rel}	t_{IBP}/s	t_{Rel}/s
(a)	40	39	6	34336	56	0.3	0.00075
(b)	56	55	8	200074	83	1.9	0.0024
(c)	102	103	8	3461628	280	690	0.013
(d)	144	143	5	625070	336	24.5	0.019

- degree of polynomials reduced
  - number of samples reduced
- size of system reduced
  - time for explicit solutions negligible

Examples

- https://gitlab.com/xiaoliu222222/examples-for-rational-functions-reconstruction
  - explicit reduction coefficients & the linear system they satisfy



Outline

Summary and Outlook

- A new method for the reconstruction of rational functions is proposed, which works by exploiting all the independent linear relations among the target functions.
- Better scaling behavior
  - improvement factor: univariate ≤ 2-variate ≤ 3-variate
- The current form of the method is not so good to solve problems with more than 3 variables → linear fit becomes dominant and sometimes prohibitive
  - refined ansatz for the relations: sparse or semi-sparse?
  - refined choice of auxiliary functions, rather than a naïve $f_{n+1}(x) = 1$
- $t_{\text{sol}} \ll t_{\text{sam}}$ in most cases $\Rightarrow$ improvements in the generators
  - for cases where $t_{\text{sol}} \geq t_{\text{sam}}$, refined approach to grouping the functions

Thank you!

The method

- Redundant relations: relations from $\vec{z}$ will be redundantly obtained from $\vec{z}'$ if $z_i \leq z_i'$ for any i
  - $z_1 = 1, z_2 = 0$: $(x_1 - 1)f_1 + (3x_1 + 2)f_2 + \cdots = 0$
  - $z_1 = 1, z_2 = 1$: $(x_1x_2 - x_2)f_1 + (3x_1x_2 + 2x_2)f_2 + \cdots = 0$
- Solution: eliminate "solvable" monomials
  - $z_1 = 1, z_2 = 0$: x_1 of f_1 is "solved", $x_1f_1 = f_1 + (-3x_1 - 2)f_2 + \cdots$
  - $z_1 = 1, z_2 = 1$: x_1 and x_1x_2 of f_1 should be "solvable"
  - in general, eliminate $\Lambda_i(\vec{z}) \times M(\vec{z}' - \vec{z})$ for all possible $\vec{z}$

- Comment on the variables division
  - any division works: $S_1 = \{x_1, \ldots, x_k\}$; $S_1 = \{x_1\}, \ldots, S_k = \{x_k\}$; ...
  - but the number of samples can vary significantly
  - do some tests to gain insights