A proof-of-concept implementation of auxiliary mass flow method

Xiao Liu
University of Oxford

Based on 1711.09572, 2009.07987, 2107.01864 and 2112.*****
Available at https://gitlab.com/xiaoliu222222/amflow-mma

圈积分和相空间积分系列讲座
2021年11月25日

Auxiliary mass flow

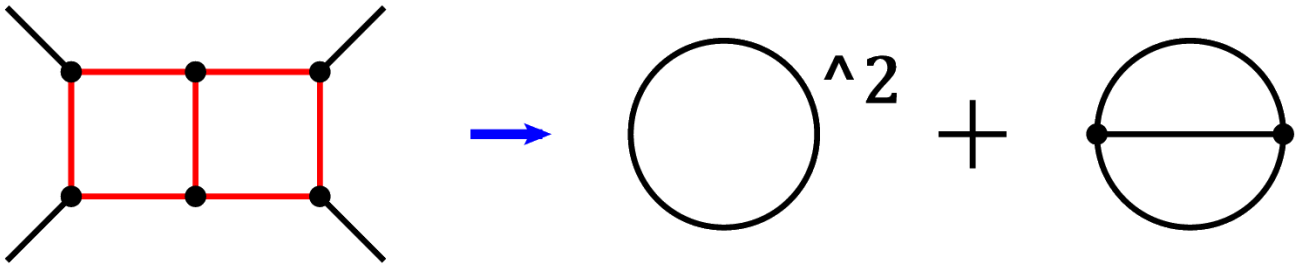
➢ Feynman integrals with auxiliary mass

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

- physical integrals

$$I(\vec{\nu}) = \lim_{\eta \to 0^+} I_{\text{mod}}(\vec{\nu};\eta)$$

- systematic and simple boundary at $\eta = \infty$



Auxiliary mass flow

- differential equations through power series expansion – a well studied theory

$$\frac{\partial}{\partial \eta}\vec{\mathcal{I}}_{\text{mod}}(\eta) = A(\eta)\vec{\mathcal{I}}_{\text{mod}}(\eta)$$

- basic facts:
  - truncated order N ~ time consumption t

$$\frac{1}{1-\eta}\sum_{i=0}^{\infty} a_i \eta^i = \sum_{i=0}^{\infty} b_i \eta^i \quad \Rightarrow \quad b_i = a_i + b_{i-1}$$

  - number of correct digits p ~ truncated order N

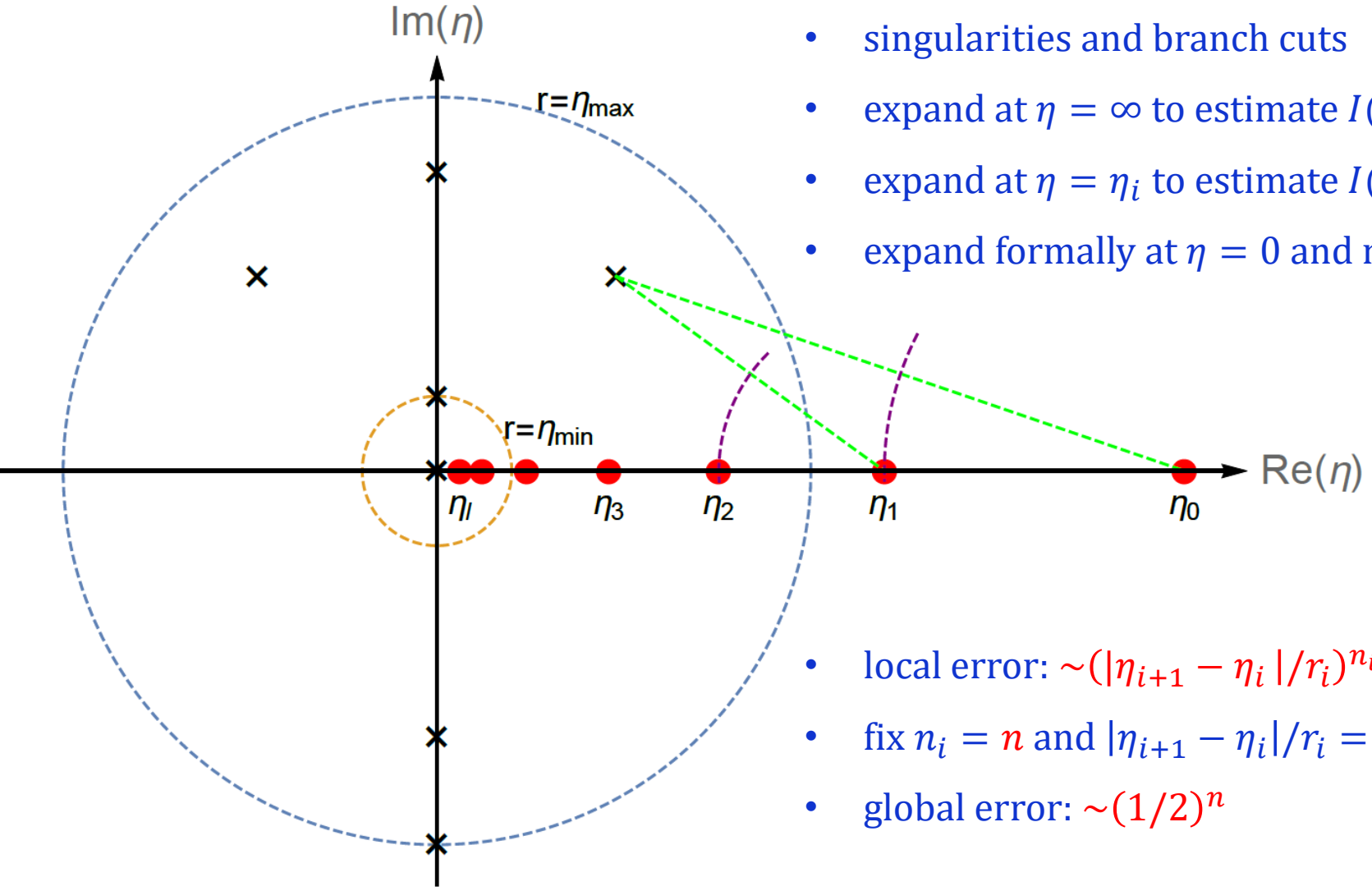
$$p \sim -\log(e) \sim -\log\left(\frac{|\eta - \eta_0|}{r}\right)^N \sim N$$

  - asymptotic solution near a regular singular point $\eta = 0$

$$\vec{\mathcal{I}}_{\text{mod}}(\eta) = \sum_{\mu \in S} \eta^\mu \sum_{k=0}^{k_\mu} \log^k(\eta) \sum_{n=0}^{\infty} \vec{I}_{\mu,k,n}\eta^n$$

Auxiliary mass flow

- differential equations through power series expansion – applications



- singularities and branch cuts
- expand at $\eta = \infty$ to estimate $I(\eta_0)$
- expand at $\eta = \eta_i$ to estimate $I(\eta_{i+1})$
- expand formally at $\eta = 0$ and match at η_l

- local error: $\sim(|\eta_{i+1} - \eta_i|/r_i)^{n_i}$
- fix $n_i = n$ and $|\eta_{i+1} - \eta_i|/r_i = k \sim 1/2$
- global error: $\sim(1/2)^n$

Auxiliary mass flow

- an important trick
  - evaluate with numerical $\epsilon = \epsilon_0, \epsilon_1, \epsilon_2, \dots, \epsilon_n$, where $|\epsilon_i| \sim r \ll R \sim 1$, with relative error p
  - fit the Taylor (Laurent) expansion $f(\epsilon) = a_0 + a_1\epsilon + \cdots + a_n\epsilon_n$

$$e_i \sim \left(\frac{r}{R}\right)^{n+1-i}$$

  - Choose reasonable $\{n, r, p\}$ such that $e_k \sim e$?
  - $n = 2k - 1,\ \ r \sim R\sqrt[k]{e},\ p \leq e_0 \sim e^2$
  - $e_4 \sim 10^{-20} \rightarrow n = 7,\ r \sim 10^{-5}R,\ p \leq 10^{-40}$
  - total time consumption: $T \sim -\log(e)$

Auxiliary mass flow

➢ AMFlow

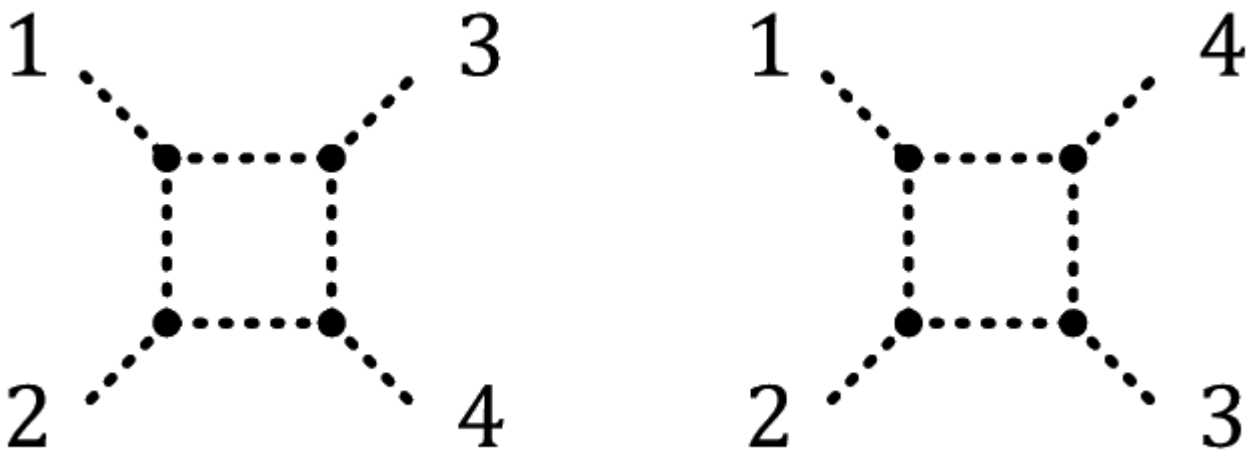
- main package: AMFlow.m
  - provides the main functions to realize auxiliary mass flow: SetupSystem, MISolve, SolveIntegrals
- differential equation solver: diffeq_solver/DESolver.m
  - provides functions to solve differential equations numerically
- interface to FiniteFlow+LiteRed: ibp_interface/FiniteFlow/interface.m(sup.m)
- interface to Kira-2.2 or later: ibp_interface/Kira/interface.m
- examples
  - examples/box: introduction to automatic evaluations like SecDec
  - examples/sunrise: introduction to manual evaluations
  - examples/tt: introduction to validation
  - examples/phase: introduction to evaluations of phase-space integrations

Auxiliary mass flow

- installation
  - FiniteFlow+LiteRed route
    - install FiniteFlow: https://github.com/peraro/finiteflow/
    - install LiteIBP.m: https://github.com/peraro/finiteflow-mathtools/
    - install LiteRed.m: https://www.inp.nsk.su/~lee/programs/LiteRed/
    - reset variables defined in ibp_interface/FiniteFlow/install.m
    - $FFPath, $FFLibraryPath, $LiteIBPPath, $LiteRedPath
  - Kira route
    - install Kira-2.2 or later: https://kira.hepforge.org/
    - reset variables defined in ibp_interface/Kira/install.m
    - $KiraExecutable
    - include the command: export FERMATPATH="/path/to/fermat/executable", in your ".bashrc"

Auxiliary mass flow

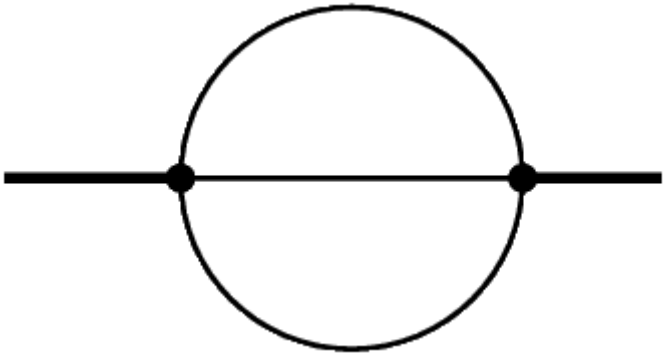
- examples/box



- SolveIntegrals[integrals_, goal_, order_]
  - integrals: a list of integrals, e.g., {j[box1,2,0,1,0], j[box1,-2,1,1,2], j[box1,1,2,2,1]}
  - goal: number of correct digits, e.g., 40
  - order: number of correct orders from ϵ^{-2L}, e.g., 2

Auxiliary mass flow

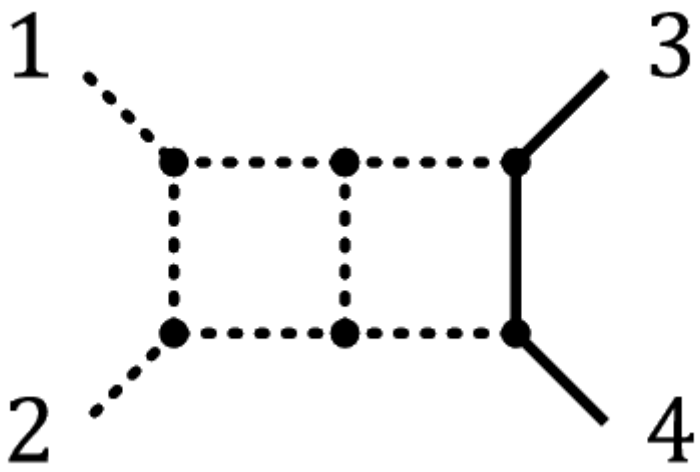
- examples/sunrise



- IBPSystem[preferred_] + AnalyticReduction[target_]
  - preferred: a list of preferred master integrals
  - target: a list of target integrals
- SetAMFOptions["AMFMode" → "Mass", …]
- SetupSystem[prop_, top_, cut_, preferred_]
  - A system ID is returned
- MISolve[epslist_, sysid_]

Auxiliary mass flow

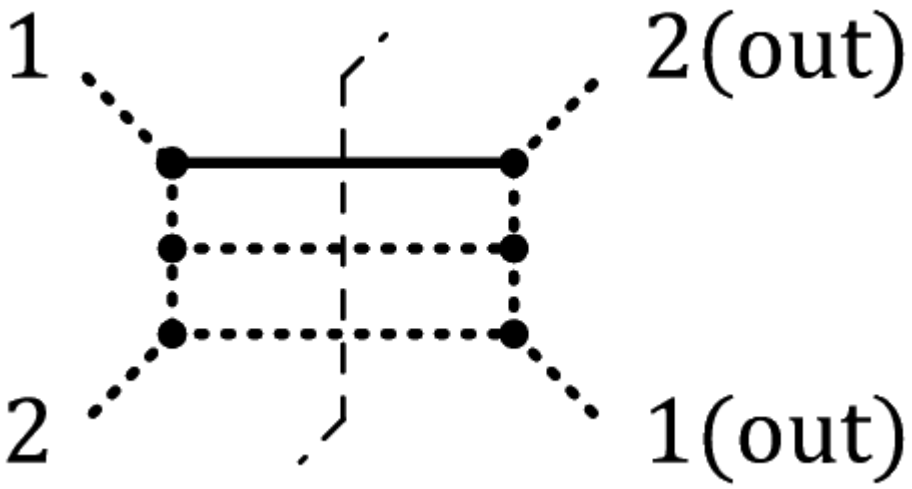
- examples/tt



- usage of diffeq_solver/DESolver.m
  - RegularRun[system_, run_]
    - run: a list of regular points to perform where we compute Taylor expansions

Auxiliary mass flow

- examples/phase



- phase-space integration using reverse unitarity
  - Cut = {1, 0, 1, 0, 1, 0, 0}
- asymptotic expansion
  - SolveAsyExp[system_]