Scattering @mplitudes @Liverpool

Jonathan Ronca

Loop Integrals Numerical Evaluation with LINE

In collaboration with: ***Renato Maria Prisco, Francesco Tramontano***

Based on: arXiv:2501.01943

Dipartimento di Fisica e Astronomia Galileo Galilei

INFN PADOVA
Istituto Nazionale di Fisica Nucleare
Sezione di Padova

26 Mar 2025

Motivation

Feynman Integrals

$$I(\mathbf{s},\epsilon) = \int_{k_1,\ldots,k_l} \prod_{\alpha=0}^{N} \frac{1}{D_\alpha^{\nu_\alpha}}, \quad D_\alpha = q_\alpha^2 - m_\alpha^2 + i\varepsilon$$

MonteCarlo Integration Methods

- Sector decomposition
- Tropical Integration
- Loop-Tree duality
- …

pySecDec Lotty FIESTA FeynTrop

- Evaluating integrals in a single point
- No need of additional inputs
- Relatively heavy computational needs

Solution of Differential Equations

- Series expansions methods
- Auxiliary-mass flow
- …

AMFlow DiffExp DESS SeaSyde

- Solutions via series expansions
- DEs and boundary conditions required
- Computationally efficient

AMFlow implements series expansion methods with automated boundary constants

Motivation

Auxiliary mass flow
(AMFlow)

[Liu,Ma:2201.11669]

- Introducing a mass parameter η into propagators
- Numerical IBPs + DE system depending on η only
- Automatic Boundary condition at $\eta \sim \infty$
- Propagating boundaries to $\eta \to 0$

Series expansion methods
(DiffExp, SeaSyde, LINE)

[Hidding:2006.05510]
[Armadillo,Bonciani,Devoto,Rana,Vicini:2205.03345]
[Prisco,JR,Tramontano:to appear]

- Analytical IBPs + Differential Equation system
- Boundary condition as input
- Propagating boundary to input PS-points

Sector Decomposition
(SecDec, pySecDec)

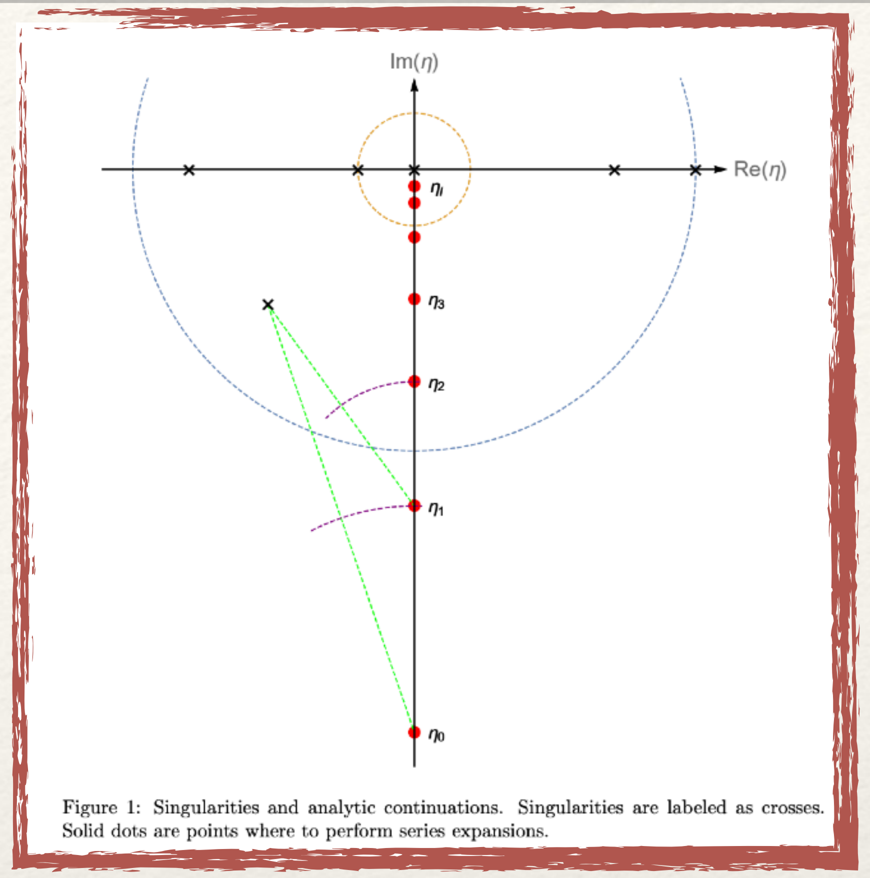
[Heinrich,Jones,Kerner,Magerya,Olsson,Schlenk:2305.19768]

- Feynman parametrization
- Splitting integration domain
- End-point subtraction of singularities and expansion
- contour deformation + expansion-by-region
- MonteCarlo integration of finite integrals

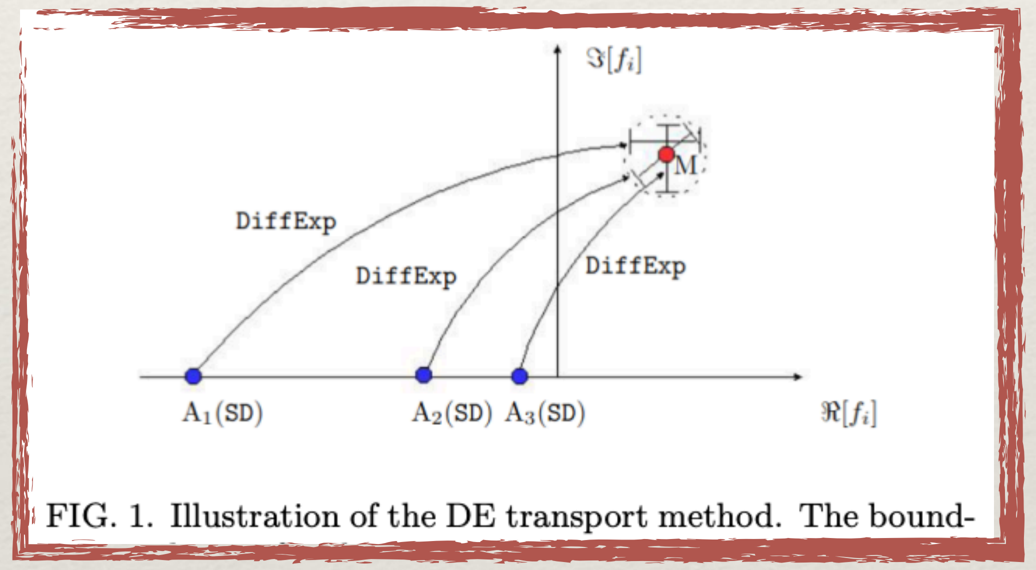
Tropical integration
(FeynTrop)

[Borinski,Munch,Tellander:2302.08955]

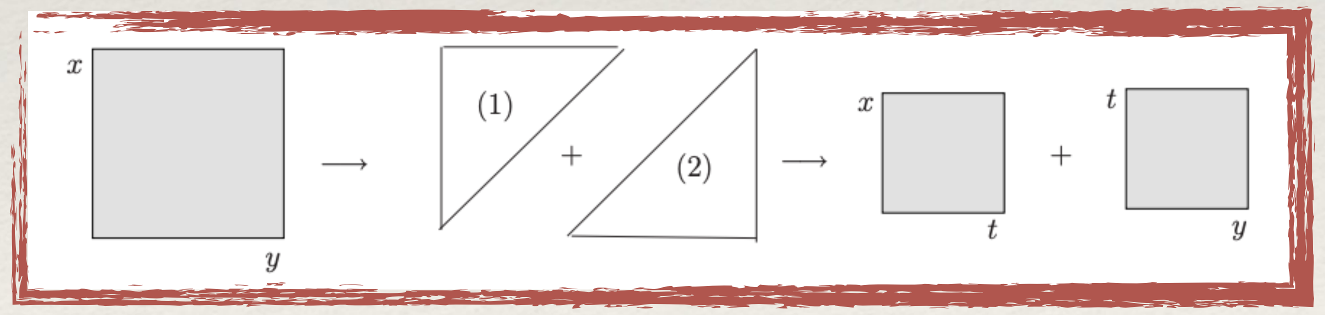
- Feynman parameters + contour deformation $x_i \to x_i e^{-i\lambda \frac{d}{dx_i}\left(\frac{\mathcal{U}(\bar{x})}{\mathcal{F}(\bar{x})}\right)}$
- Tropical approximation of Symanzik Polynomial
- MonteCarlo integration improved with tropical sampling
- Improving sampling by geometrical insights



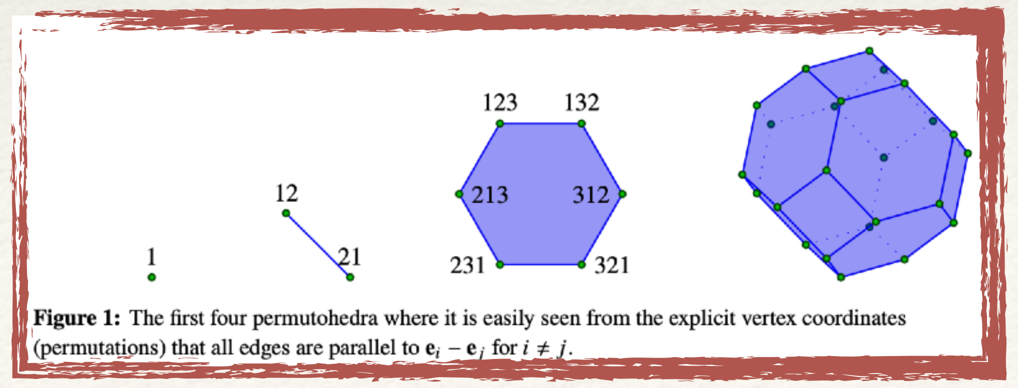
[Liu,Ma:2201.11669]



[Dubovyk,Freitas,Gluza,Grzanka,Hidding,Usovitsch:2201.0257]



[Heinrich:0803.4177]



[Borinsky,Munch,Tellander:2310.19890]

Motivation

Codes implementing the DE method via series expansions

AMFlow: DEs w.r.t. an auxiliary mass

DiffExp: DEs via series expansions

SeaSyde: DEs + complex masses

Mathematica packages

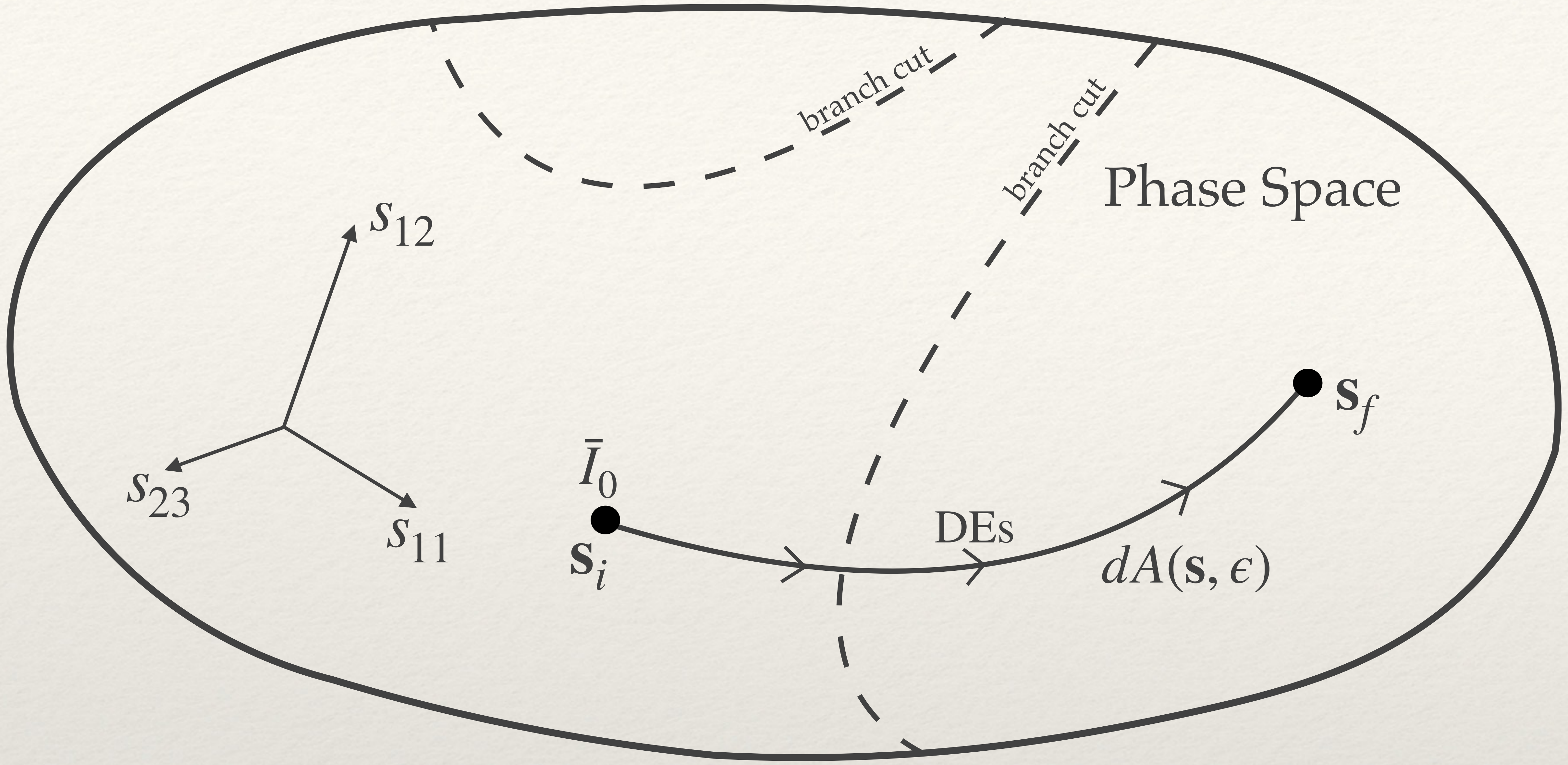
- Multi-purpose
- High-level
- Licenses issues

We propose a novel tool: LINE (**L**oop **I**ntegrals **N**umerical **E**valuation)

C Implementation

- Low-level language
- Open source
- Suitable for clusters

Solution by series expansion



Differential Equations for Feynman Integrals

$$\frac{d}{ds_{ij}}\bar{I}(\mathbf{s}, \epsilon) = A_{ij}(\mathbf{s}, \epsilon)\bar{I}(\mathbf{s}, \epsilon), \quad \bar{I}(\mathbf{s}_i, \epsilon) = \bar{I}_0$$

Parametrizing path w.r.t. the line parameter η

$$\mathbf{s} = \mathbf{s}(\eta)$$

Differential Equations w.r.t the line parameter η

$$\frac{d}{d\eta}\bar{I}(\eta, \epsilon) = A(\eta, \epsilon)\bar{I}(\eta, \epsilon)$$

$$A(\eta, \epsilon) = \begin{pmatrix} \times & 0 & 0 & 0 & 0 & 0 & \cdots & & & \\ \times & \times & \times & 0 & 0 & 0 & \cdots & & & \\ \times & \times & \times & 0 & 0 & 0 & \cdots & & & \\ \times & \times & \times & \times & 0 & 0 & \cdots & & & \\ \vdots & & & & & & \ddots & & & \\ \times & & \cdots & & & & & \times & \times & \times \\ \times & & \cdots & & & & & \times & \times & \times \\ \times & & \cdots & & & & & \times & \times & \times \end{pmatrix}$$

Block-Triangular structure

Rational functions of polynomials of η and ϵ

Solution by series expansion

Selecting a straight path connecting $\mathbf{s}_i$ and $\mathbf{s}_f$

$$\mathbf{s} = (\mathbf{s}_f - \mathbf{s}_i)\eta + \mathbf{s}_i, \ \eta \in [0,1]$$

ϵ-expansion of the solution from **interpolation**

$$\bar{I}(\eta, \epsilon) = \epsilon^{-2l} \sum_{j=0}^{\text{ord}} \bar{I}_j(\eta)\, \epsilon^j + O(\epsilon^{\text{ord}+1})$$

Interpolating

$$\begin{cases} \epsilon_1^{2l} \bar{I}(\eta, \epsilon_1) = \bar{I}_0(\eta) + \bar{I}_1(\eta)\epsilon_1 + \bar{I}_2(\eta)\epsilon_1^2 + \cdots \\ \vdots \\ \epsilon_{\text{ord}}^{2l} \bar{I}(\eta, \epsilon_{\text{ord}}) = \bar{I}_0(\eta) + \bar{I}_1(\eta)\epsilon_{\text{ord}} + \bar{I}_2(\eta)\epsilon_{\text{ord}}^2 + \cdots \end{cases}$$

Trading a **two-variables DE** with **several one-variable** problems

Linking gmp, mpfr, mpc **libraries**

Solution by series expansion

$$\frac{d}{d\eta}\bar{I}(\eta) = A(\eta)\bar{I}(\eta)$$

Propagating the boundary along the path $\eta \in [0,1]$

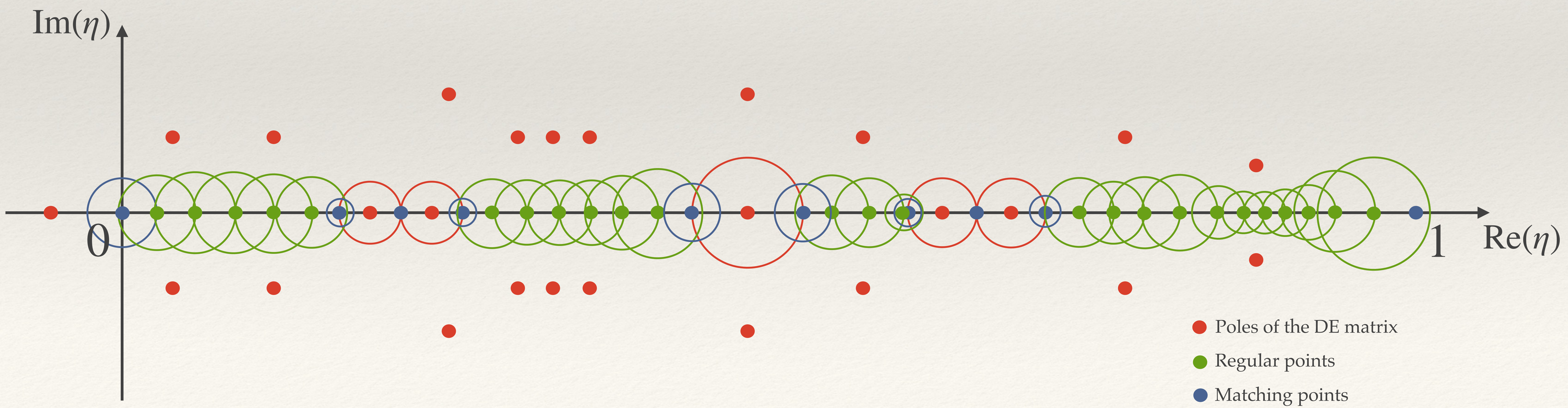
- Find poles of the DEs
- Define a proper path, i.e. sets of points between 0 and 1
- Use DE to fix the coefficients of the solutions ansatz

Ansatz around regular points

$$\bar{I}(\eta) = \sum_{k=0}^{\infty} c_k \, \eta^k$$

Ansatz around regular-singular points

$$\bar{I}(\eta) = \sum_{\lambda \in S} \eta^{\lambda} \sum_{l=0}^{L_\lambda} \log^l \eta \sum_{k=0}^{\infty} c_{\lambda,l,k} \eta^k$$



Normalized Fuchsian form

Solutions admit at most **regular-singular** points

If $\left(\begin{array}{c} \textbf{Poincaré rank} > 0 \\ \frac{d}{d\eta}\bar{I}(\eta) = A(\eta)\bar{I}(\eta) \end{array} \right)$, Then $\left(\begin{cases} \bar{I}(\eta) = T(\eta)\bar{I}'(\eta) \\ A'(\eta) = T^{-1}(\eta)A(\eta)T(\eta) - T^{-1}(\eta)\frac{d}{d\eta}T(\eta) \end{cases}, \begin{array}{c} \textbf{Poincaré rank} = 0 \\ \frac{d}{d\eta}\bar{I}'(\eta) = A'(\eta)\bar{I}'(\eta) \end{array} \right)$

$T(\eta)$ can be found **algorithmically**
Automated method from [Lee '15, arxiv:1411.0911]

Normalized Fuchsian form

$$A'(\eta) = \frac{1}{\eta}\sum_{k=0}^{\infty} A'_k \eta^k$$

At most simple poles

Solutions ansatz around **regular-singular** points

$$\bar{I}(\eta) = \sum_{\lambda \in S} \eta^{\lambda} \sum_{l=0}^{L_\lambda} \log^l \eta \sum_{k=0}^{\infty} c_{\lambda,l,k}\eta^k$$

Eigenvalues of the leading term A'_0

Dimension of the Jordan chain related to λ

Mathematical expression in C

Typical matrix element

```
2*(-4+d)^2*(10-7*d+d^2)*s^9*t^5*(3*s+4*t)*(3*(-3+d)*s+(-16+5*d)*t)-m2*s^7*t^4*(8-6*d+d^2)* (2*(-4150+3205*d-805*d^2+66*d^3)*s^4+(-42520+32724*d-
8219*d^2+675*d^3)*s^3*t+2*(-35500+27140*d-6743*d^2+532*d^3+3*d^4)*s^2*t^2+4*(-9914+7471*d-1782*d^2+117*d^3+4*d^4)*s*t^3+8*(-50-149*d+148*d^2-
43*d^3+4*d^4)*t^4)+262144*(-56+58*d-19*d^2+2*d^3)*m2^10*(s+t)*(4*(20-9*d+d^2)*s^5+(535-272*d+33*d^2)*s^4*t+(1265-753*d+135*d^2-7*d^3)*s^3*t^2+(1797-1281*d+309*d^2-
25*d^3)*s^2*t^3-2*(-638+525*d-144*d^2+13*d^3)*s*t^4+4*(71-70*d+21*d^2-2*d^3)*t^5)+4*(-2+d)*m2^2*s^6*t^3*((7640-6653*d+2015*d^2-243*d^3+9*d^4)*s^5+4*(33430-
33161*d+12130*d^2-1942*d^3+115*d^4)*s^4*t+(444340-442838*d+161309*d^2-24828*d^3+1165*d^4+36*d^5)*s^3*t^2+2*(249308-242356*d+83109*d^2-
10606*d^3+2*d^4+67*d^5)*s^2*t^3+2*(69640-49054*d+2605*d^2+5514*d^3-1535*d^4+122*d^5)*s*t^4+4*(-1048+7970*d-7887*d^2+3094*d^3-545*d^4+36*d^5)*t^5)+16*(-
2+d)*m2^3*s^5*t^2*(6*(3100-3465*d+1444*d^2-265*d^3+18*d^4)*s^6+(68620-79124*d+33975*d^2-6404*d^3+445*d^4)*s^5*t+(-29500+1205*d+15979*d^2-7473*d^3+1253*d^4-
72*d^5)*s^4*t^2-3*(69988-44332*d-1513*d^2+6687*d^3-1654*d^4+124*d^5)*s^3*t^3+(77798-240621*d+204195*d^2-74961*d^3+12691*d^4-814*d^5)*s^2*t^4-8*(-35409+56641*d-
35712*d^2+11089*d^3-1695*d^4+102*d^5)*s*t^5-4*(-8742+20263*d-15499*d^2+5397*d^3-887*d^4+56*d^5)*t^6)-65536*m2^9*(2*(-7760+11692*d-6838*d^2+1952*d^3-
273*d^4+15*d^5)*s^7+(-133880+204796*d-121754*d^2+35335*d^3-5021*d^4+280*d^5)*s^6*t-2*(230470-363039*d+225204*d^2-69833*d^3+11179*d^4-832*d^5+19*d^6)*s^5*t^2+(-
881296+1460192*d-974010*d^2+335937*d^3-63289*d^4+6178*d^5-244*d^6)*s^4*t^3+(-1117108+1966668*d-1416163*d^2+536687*d^3-113295*d^4+12661*d^5-586*d^6)*s^3*t^4-
2*(425080-786430*d+597375*d^2-239205*d^3+53345*d^4-6287*d^5+306*d^6)*s^2*t^5-4*(75164-145432*d+115125*d^2-47790*d^3+10987*d^4-1328*d^5+66*d^6)*s*t^6-8*(3976-
8038*d+6585*d^2-2802*d^3+655*d^4-80*d^5+4*d^6)*t^7)-64*m2^4*s^4*t*((-27440+43838*d-27405*d^2+8407*d^3-1267*d^4+75*d^5)*s^7+2*(-137060+215942*d-
132876*d^2+40069*d^3-5932*d^4+345*d^5)*s^6*t+(-830000+1325770*d-834559*d^2+262061*d^3-42123*d^4+3063*d^5-60*d^6)*s^5*t^2+(-1369568+2316650*d-
1582833*d^2+560776*d^3-108795*d^4+10962*d^5-448*d^6)*s^4*t^3-2*(1107084-1971420*d+1439151*d^2-554278*d^3+119240*d^4-13619*d^5+646*d^6)*s^3*t^4+(-
2689240+4874294*d-3633815*d^2+1433455*d^3-316573*d^4+37173*d^5-1814*d^6)*s^2*t^5-2*(613448-1148204*d+884122*d^2-359659*d^3+81686*d^4-9831*d^5+490*d^6)*s*t^6-
4*(25760-49326*d+38467*d^2-15660*d^3+3517*d^4-414*d^5+20*d^6)*t^7)+16384*m2^8*(4*(-3340+5218*d-3180*d^2+949*d^3-139*d^4+8*d^5)*s^8+(-168680+260686*d-
156935*d^2+46219*d^3-6677*d^4+379*d^5)*s^7*t+(-776760+1215342*d-744523*d^2+225335*d^3-34281*d^4+2243*d^5-28*d^6)*s^6*t^2+(-1750824+2832518*d-
1824311*d^2+597425*d^3-104127*d^4+9011*d^5-292*d^6)*s^5*t^3+(-2527592+4334208*d-3021960*d^2+1102421*d^3-222816*d^4+23735*d^5-1044*d^6)*s^4*t^4-2*(1257572-
2284780*d+1704803*d^2-671531*d^3+147648*d^4-17202*d^5+830*d^6)*s^3*t^5+(-1357816+2585562*d-2022013*d^2+832937*d^3-190839*d^4+23071*d^5-1150*d^6)*s^2*t^6-
2*(142736-285392*d+232778*d^2-99235*d^3+23358*d^4-2883*d^5+146*d^6)*s*t^7-4*(3976-8038*d+6585*d^2-2802*d^3+655*d^4-80*d^5+4*d^6)*t^8)+256*m2^5*s^3*(2*(-
3400+5400*d-3354*d^2+1022*d^3-153*d^4+9*d^5)*s^8+(-119920+186864*d-113586*d^2+33817*d^3-4943*d^4+284*d^5)*s^7*t+(-644040+1006058*d-614765*d^2+185243*d^3-
27921*d^4+1779*d^5-18*d^6)*s^6*t^2+(-1660112+2669004*d-1703084*d^2+549991*d^3-93801*d^4+7826*d^5-236*d^6)*s^5*t^3-2*(1458844-2455646*d+1670234*d^2-
590065*d^3+114572*d^4-11626*d^5+483*d^6)*s^4*t^4+(-3834288+6658564*d-4715934*d^2+1753646*d^3-362739*d^4+39719*d^5-1804*d^6)*s^3*t^5-2*(1286796-
2271580*d+1638513*d^2-621235*d^3+131070*d^4-14633*d^5+677*d^6)*s^2*t^6-4*(108724-179486*d+117200*d^2-38339*d^3+6476*d^4-507*d^5+12*d^6)*s*t^7+8*(-680-
6274*d+11075*d^2-7130*d^3+2215*d^4-336*d^5+20*d^6)*t^8)+4096*m2^7*s*(4*(-2180+3146*d-1742*d^2+465*d^3-60*d^4+3*d^5)*s^8+2*(-14200+20880*d-11748*d^2+3163*d^3-
407*d^4+20*d^5)*s^7*t+(203300-309420*d+181227*d^2-50405*d^3+6257*d^4-155*d^5-20*d^6)*s^6*t^2+(1010576-1573724*d+954516*d^2-282685*d^3+40814*d^4-2257*d^5-
8*d^6)*s^5*t^3+2*(986890-1627267*d+1075037*d^2-364353*d^3+66629*d^4-6194*d^5+226*d^6)*s^4*t^4+(2712280-4827004*d+3520554*d^2-1353979*d^3+290645*d^4-
33096*d^5+1564*d^6)*s^3*t^5+(2270132-4284872*d+3330633*d^2-1368343*d^3+313753*d^4-38077*d^5+1910*d^6)*s^2*t^6+16*(48454-97161*d+79799*d^2-34386*d^3+8206*d^4-
1029*d^5+53*d^6)*s*t^7+4*(18044-38504*d+33167*d^2-14795*d^3+3617*d^4-461*d^5+24*d^6)*t^8)-1024*m2^6*s^2*(8*(-2230+3441*d-2069*d^2+609*d^3-88*d^4+5*d^5)*s^8+(-
166360+254642*d-151519*d^2+44024*d^3-6265*d^4+350*d^5)*s^7*t+(-596680+937046*d-578117*d^2+177600*d^3-27977*d^4+2018*d^5-42*d^6)*s^6*t^2+(-1084336+1798006*d-
1197961*d^2+411803*d^3-77217*d^4+7505*d^5-296*d^6)*s^5*t^3-2*(661356-1160694*d+833066*d^2-314989*d^3+66520*d^4-7468*d^5+349*d^6)*s^4*t^4+(-775816+1362616*d-
978084*d^2+369317*d^3-77690*d^4+8657*d^5-400*d^6)*s^3*t^5+2*(310184-624798*d+520701*d^2-229750*d^3+56511*d^4-7331*d^5+391*d^6)*s^2*t^6+2*(351912-
727320*d+616792*d^2-274703*d^3+67760*d^4-8775*d^5+466*d^6)*s*t^7+4*(23880-58276*d+55626*d^2-26933*d^3+7046*d^4-951*d^5+52*d^6)*t^8)
```

~O(10) MB

Mathematical expression in C

Parsing mathematical expressions

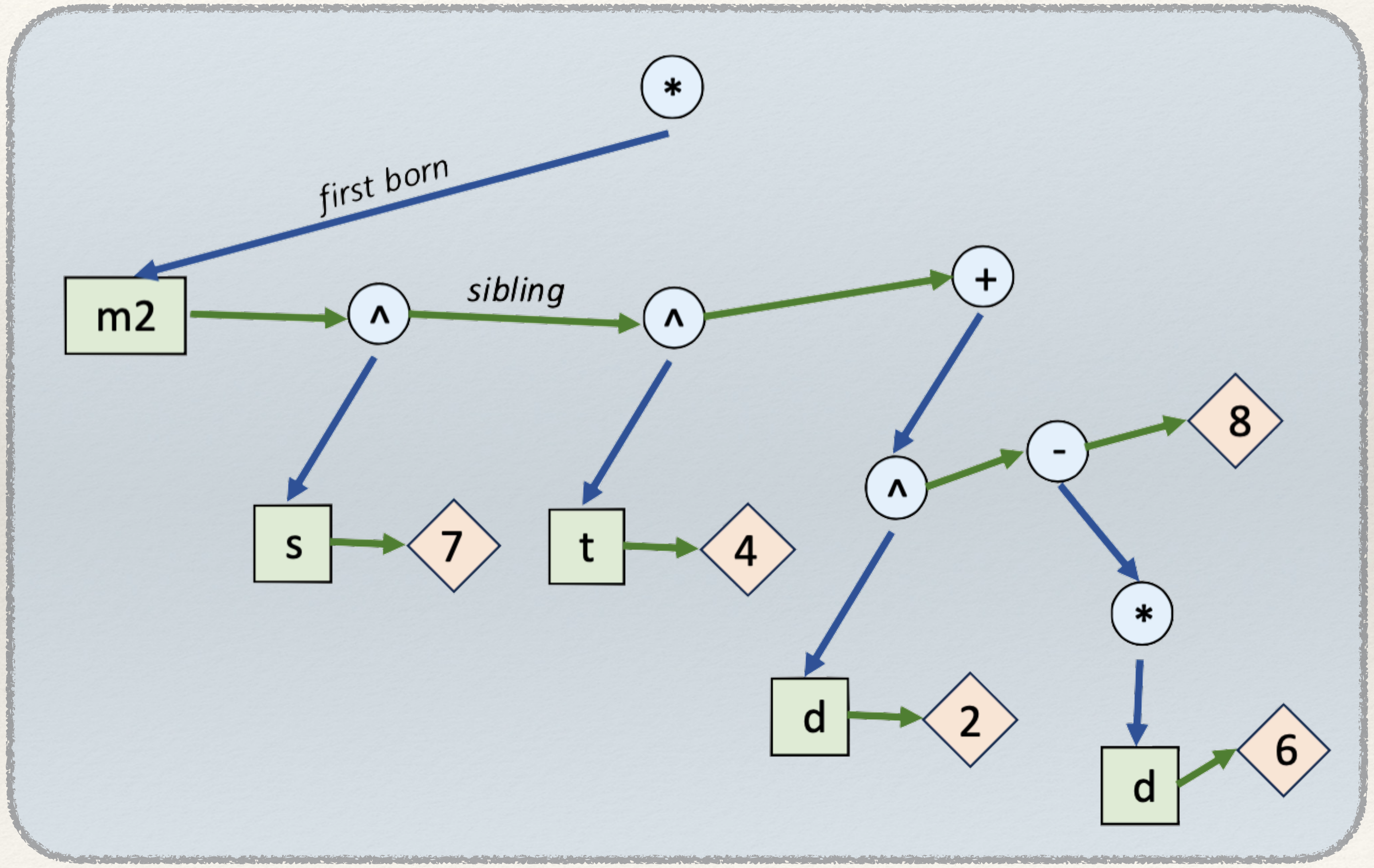
Basic operations:

- sum
- products
- division
- expansions
- …

LINE implements a dedicated parser

- No needs for general purpose tools
- Expressions trees via **linked lists**
- Operations via **lists management**

`m2 * s^7 * t^4 * (8 – 6 * d + d^2)`



Mathematical expression in C

What expressions we need to manipulate?

Rational functions of **large polynomials:** $\dfrac{p(\eta)}{q(\eta)}$

$$A(\eta, \epsilon) = \begin{pmatrix} \times & 0 & 0 & 0 & 0 & 0 & \cdots & \\ \times & \times & \times & 0 & 0 & 0 & \cdots & \\ \times & \times & \times & 0 & 0 & 0 & \cdots & \\ \times & \times & \times & \times & 0 & 0 & \cdots & \\ \vdots & & & & & & & \\ \vdots & & & & & & \ddots & \\ \times & & \cdots & & & & \times & \times & \times \\ \times & & \cdots & & & & \times & \times & \times \\ \times & & \cdots & & & & \times & \times & \times \end{pmatrix}$$

$\dfrac{p(\eta)}{q(\eta)}$

Storing coefficients

Numerator $p(\eta) = a_0 + a_1\eta + a_2\eta^2 + \dots$

Denominator $q(\eta) = \eta^{m_0}(\eta - \eta_1)^{m_1}(\eta - \eta_2)^{m_2}\cdots$

Storing roots and their multiplicity

Operations

- Shifts
- Products, sums, LCM
- Limits on singular points

Mathematical expression in C

$$\text{roots} = \{\eta_0, \eta_1, \ldots, \eta_N\}$$

$$\text{roots} = \{\mathtt{0.0000e0}, \mathtt{4.2000e1}, \cdots, \mathtt{1.0000e1}\}$$

Labelling each unique roots

$$\begin{pmatrix} \frac{p_{11}(\eta)}{\eta(\eta-\eta_1)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \frac{p_{n1}(\eta)}{\eta^3(\eta-\eta_4)(\eta-\eta_6)^2\cdots} & \cdots & \frac{p_{nn}(\eta)}{\eta(\eta-\eta_2)^4(\eta-\eta_{24})^2\cdots} \end{pmatrix}$$

$$\begin{pmatrix} \{\mathtt{r0}:1, \mathtt{r1}:1\} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \{\mathtt{r0}:3, \mathtt{r4}:1, \mathtt{r6}:2, \cdots\} & \cdots & \{\mathtt{r0}:1, \mathtt{r2}:4, \mathtt{r24}:2, \cdots\} \end{pmatrix}$$

$$p(\eta)\,\eta^2\,(\eta-\eta_6)^2 = \{\mathtt{a}'_0, \mathtt{a}'_1, \ldots\}$$

Manipulating lists of coefficients

$$\frac{p(\eta)}{\eta(\eta-\eta_2)^4(\eta-\eta_{24})^2} + \frac{q(\eta)}{\eta^3(\eta-\eta_2)(\eta-\eta_6)^2} = \frac{p(\eta)\,\eta^2\,(\eta-\eta_6)^2 + q(\eta)\,(\eta-\eta_2)^3\,(\eta-\eta_{24})^2}{\eta^3(\eta-\eta_2)^4(\eta-\eta_6)^2(\eta-\eta_{24})^2}$$

$$\mathtt{LCM}(\{\mathtt{r0}:3, \mathtt{r2}:4, \mathtt{r24}:2\}, \{\mathtt{r0}:1, \mathtt{r2}:1, \mathtt{r6}:2\}) = \{\mathtt{r0}:3, \mathtt{r2}:4, \mathtt{r6}:2, \mathtt{r24}:2\}$$

Merging sets of roots : multiplicity

Analytic continuation

$$\bar{I}(\eta) = \sum_{\lambda \in S} \eta^{\lambda} \sum_{l=0}^{L_\lambda} \log^l \eta \sum_{k=0}^{\infty} c_{\lambda,l,k} \eta^k$$

Logarithms introduce **branch cuts**

Parametrizing invariants

$$\begin{cases} s_1 = (s_{1f} - s_{1i})\,\eta + s_{1f} \\ \vdots \\ m_1^2 = (m_{1f}^2 - m_{1i}^2)\,\eta + m_{1f}^2 \\ \vdots \end{cases}$$

Needs for selecting **physical** branch cuts

Cutkowski invariants

$$z = c_1 s_1 + c_2 s_2 + \ldots + (m_1 + m_2 + \cdots)^2 = z(\eta)$$

Branch cut

$z > 0$

Branch points

$z = 0$

Feynman prescription

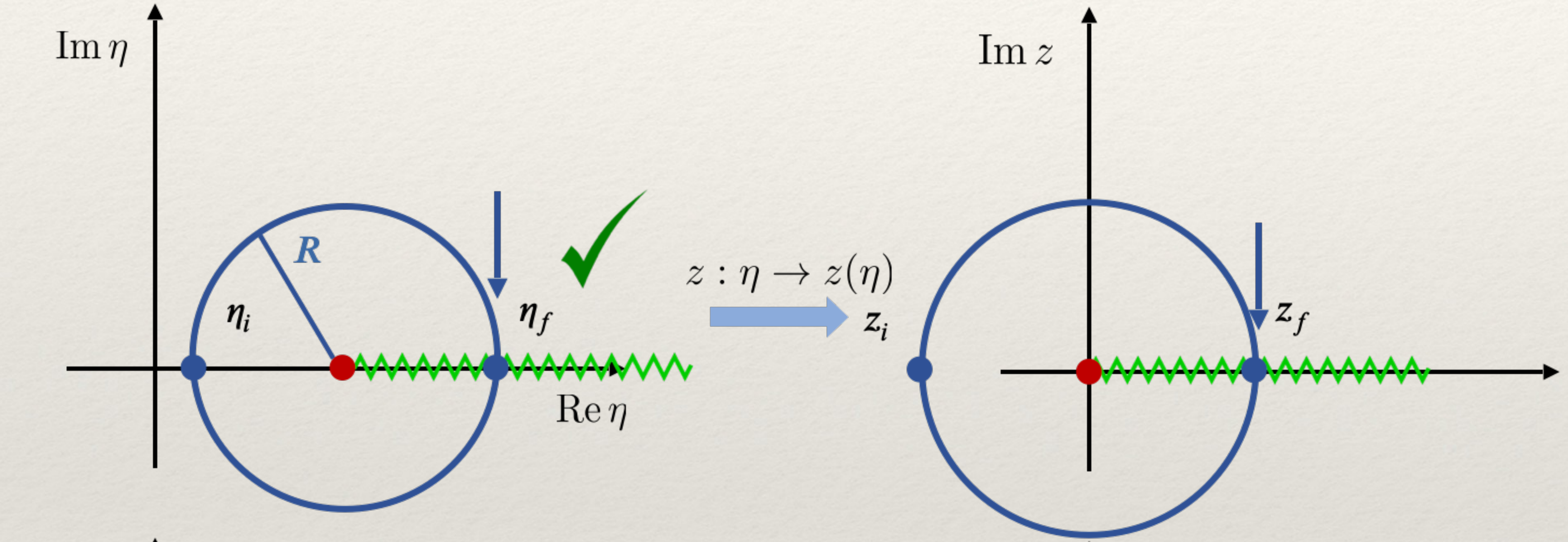
In the $z(\eta)$ plain, branch cuts must be approached from the **upper-half** plane

Analytic continuation: fixed masses

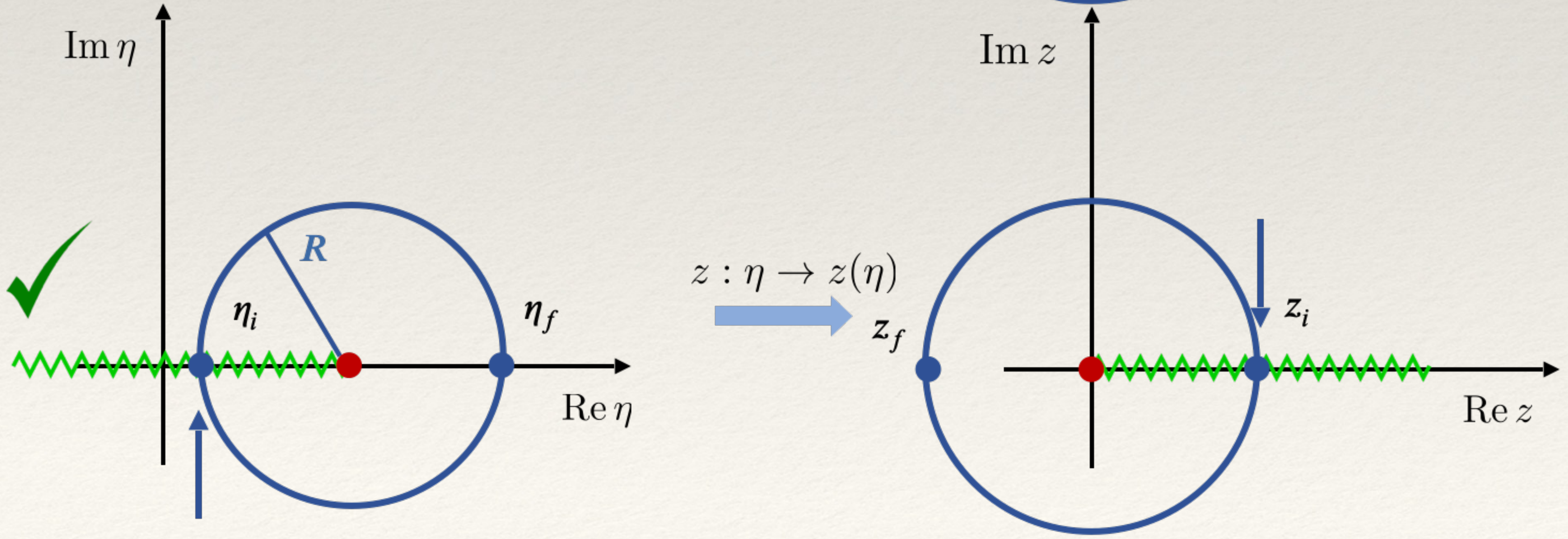
The $z : \eta \to z(\eta)$ map **preserves orientation**

Varying **invariants only**: $z(\eta) = c_1 s_1(\eta) + c_2 s_2(\eta) + \ldots + (m_1 + m_2 + \cdots)^2$

$$\mathbf{s}(\eta) = \begin{cases} s_1 = (s_{1f} - s_{1i})\,\eta + s_{1f} \\ \vdots \\ m_1^2 = \text{const} \\ \vdots \end{cases}$$



$$\log(\eta_i) = \log|\eta_i| + i\pi + i2\pi n$$
$$\log(\eta_f) = \log|\eta_f| + i2\pi n$$



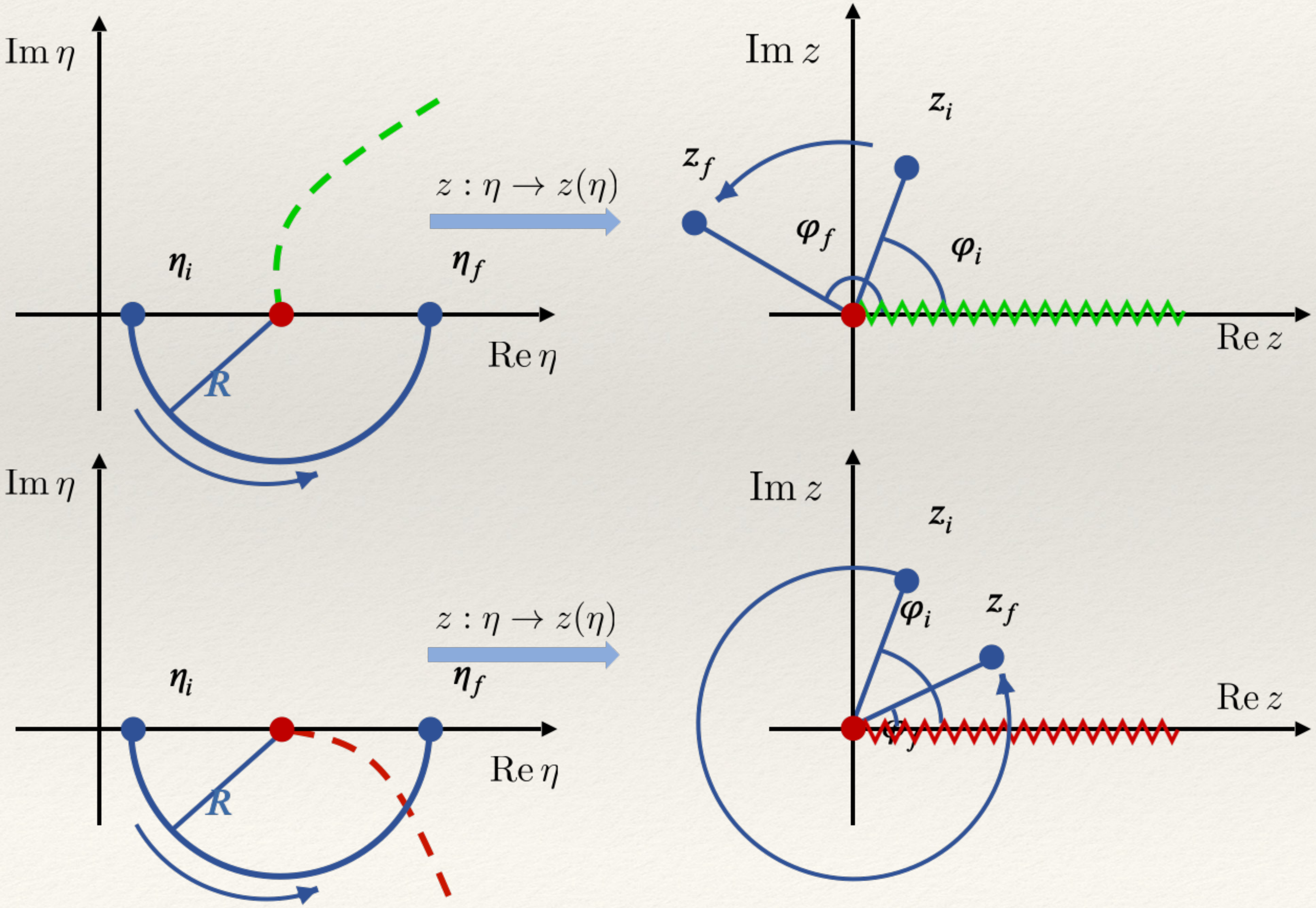
$$\log(\eta_i) = \log|\eta_i| - i\pi + i2\pi n$$
$$\log(\eta_f) = \log|\eta_f| + i2\pi n$$

Analytic continuation: varying masses

The $z : \eta \rightarrow z(\eta)$ map **preserves orientation**

Varying **masses too**: $z(\eta) = c_1 s_1(\eta) + c_2 s_2(\eta) + \ldots + (m_1(\eta) + m_2(\eta) + \cdots)^2$

$$\mathbf{s}(\eta) = \begin{cases} s_1 = (s_{1f} - s_{1i})\,\eta + s_{1f} \\ \vdots \\ m_1^2 = [(m_{1f} - m_{1i})\,\eta + m_{1f}]^2 \\ \vdots \end{cases}$$

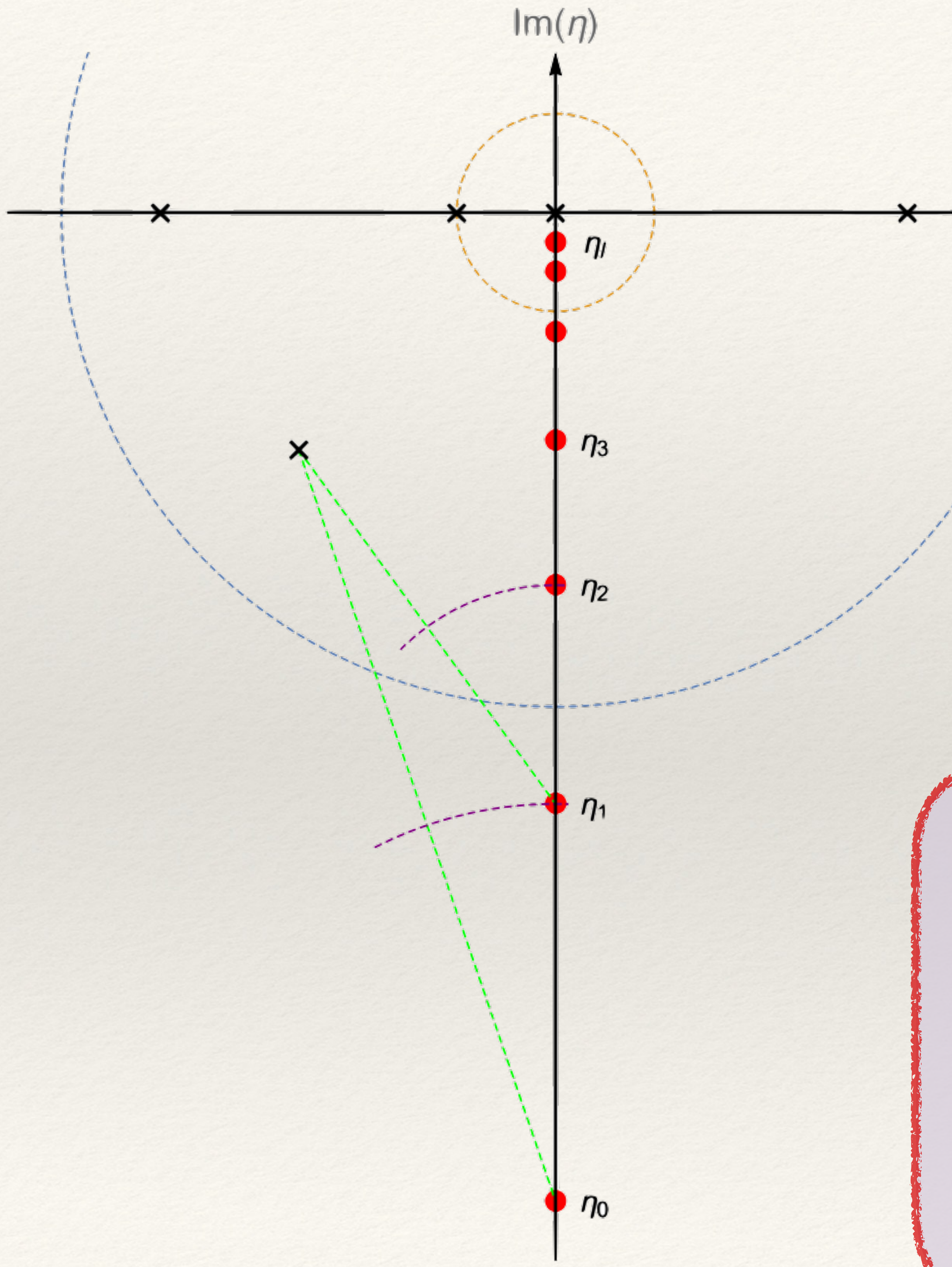


$$\log(\eta_i) = \log|\eta_i| - i\pi + i2\pi n$$
$$\log(\eta_f) = \log|\eta_f| + i2\pi n$$

$$\log(\eta_i) = \log|\eta_i| + i\pi + i2\pi n$$
$$\log(\eta_f) = \log|\eta_f| + i2\pi n$$

$n \in \mathbb{Z}$

Boundaries: auxiliary-mass flow



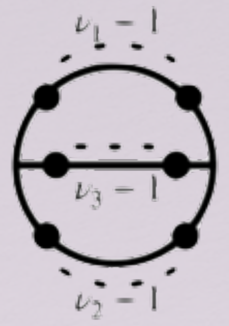
Automated method for **boundary conditions**

Auxiliary mass flow method

- Fixing numerical kinematics
- Insert auxiliary mass parameter η
- Known boundaries for large η
- Propagating η to 0

Relevant Integrals for boundaries

$$= (-1)^{\nu} \frac{\Gamma(\nu - 2 + \epsilon)}{\Gamma(\nu)}$$



$$=(-1)^{\nu}\Bigg[\frac{\Gamma(\nu_3 - 2 + \epsilon)\Gamma(\nu_1 + \nu_2 - 2 + \epsilon)}{\Gamma(\nu_3)\Gamma(\nu_1 + \nu_2)}\ {}_4F_3\left(\begin{matrix}2-\epsilon, \nu_1, \nu_2, \nu_1+\nu_2-2+\epsilon \\ \frac{\nu_1+\nu_2}{2}, \frac{\nu_1+\nu_2}{2}+\frac{1}{2}, 3-\nu_3-\epsilon\end{matrix}; \frac{1}{4}\right)$$

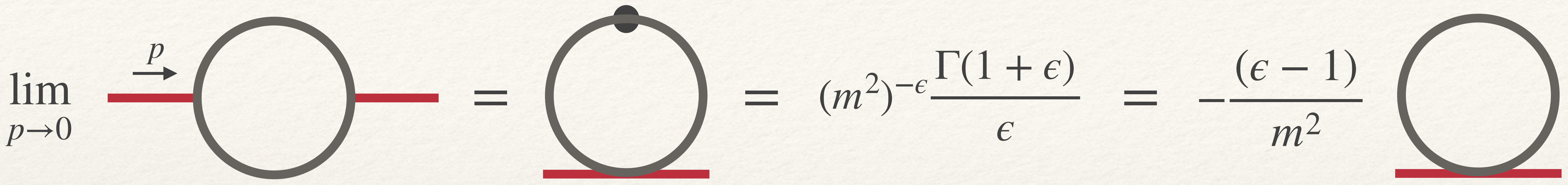
$$+ \frac{\Gamma(2-\nu_3-\epsilon)\Gamma(\nu_1+\nu_3-2+\epsilon)\Gamma(\nu_2+\nu_3-2+\epsilon)\Gamma(\nu+2\epsilon-4)}{\Gamma(\nu_1)\Gamma(\nu_2)\Gamma(2-\epsilon)\Gamma(\nu+\nu_3-4+2\epsilon)}$$

$$\times\ {}_4F_3\left(\begin{matrix}\nu_3, \nu_1+\nu_3-2+\epsilon, \nu_2+\nu_3-2+\epsilon, \nu-4+2\epsilon \\ \nu_3-1+\epsilon, \frac{\nu+\nu_3-4}{2}+\epsilon, \frac{\nu+\nu_3-3}{2}+\epsilon\end{matrix}; \frac{1}{4}\right)\Bigg],$$

Boundaries: Expansion-by-regions

Limit of vanishing external momentum for the 1-loop bubble

$$-(m^2)\frac{(m^2)^{-\epsilon}}{(\epsilon-1)}\frac{\Gamma(1+\epsilon)}{\epsilon}$$



Known contribution from a kinematical limit

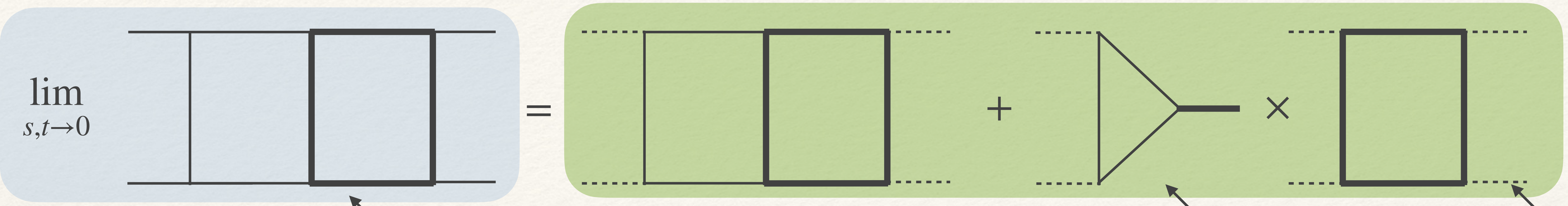
Regular contribution

Exploiting the DEs and imposing regularity

$\eta \to 0$ limit must be regular

$\eta \to 0$ and DE impose constraints on the solution

Boundaries: Expansion-by-regions



$c_1\eta^{\lambda_1-n_1}\bar{H}_1(\eta) + c_2\eta^{\lambda_2-n_2}\bar{H}_2(\eta) + \cdots + \bar{P}(\eta)$

Regular $\quad \eta^{-1-\epsilon} \quad$ Regular

Ansatz of the solution

Behaviour predicted by Expansion-by-Regions

Idea:

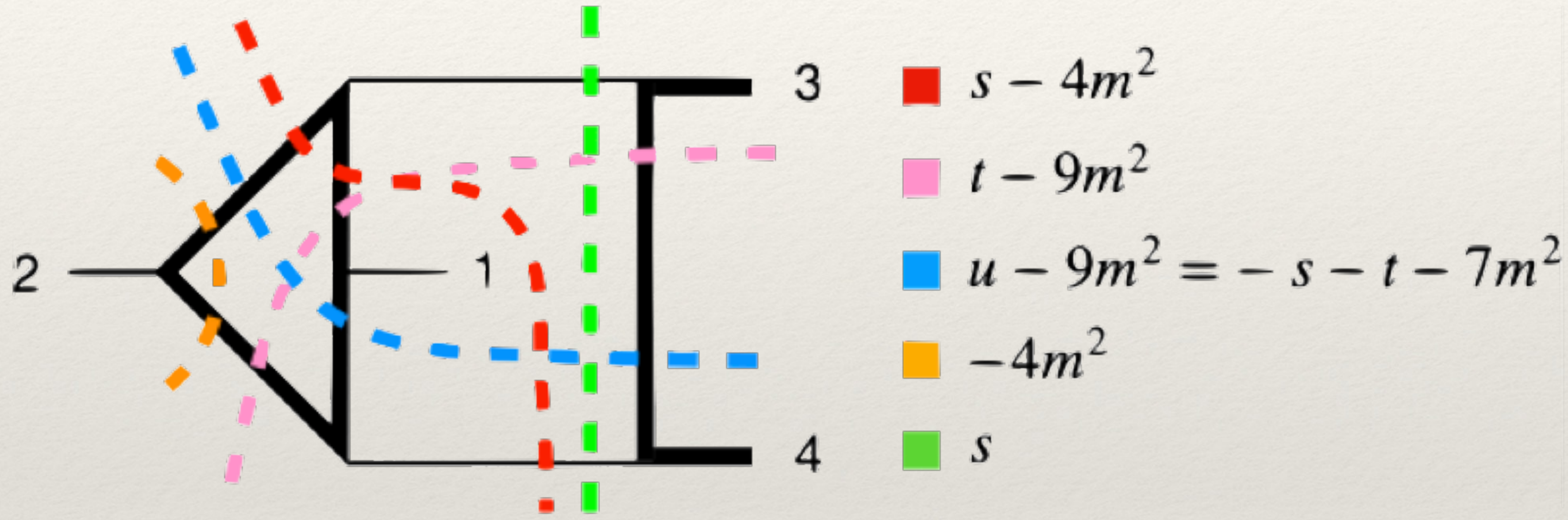
- Impose behaviour coming from Expansion-by-regions
- Impose cancellation of unwanted power behaviours
- Getting linear relations between coefficients c_i

Pros:

- DEs can be exploited to **generate boundary constants**
- Only a **limited set of integrals** have to be known
- Possible **iterative strategy** to evaluate missing integrals

Implementation under investigation

Examples



Examples: 1L triangle

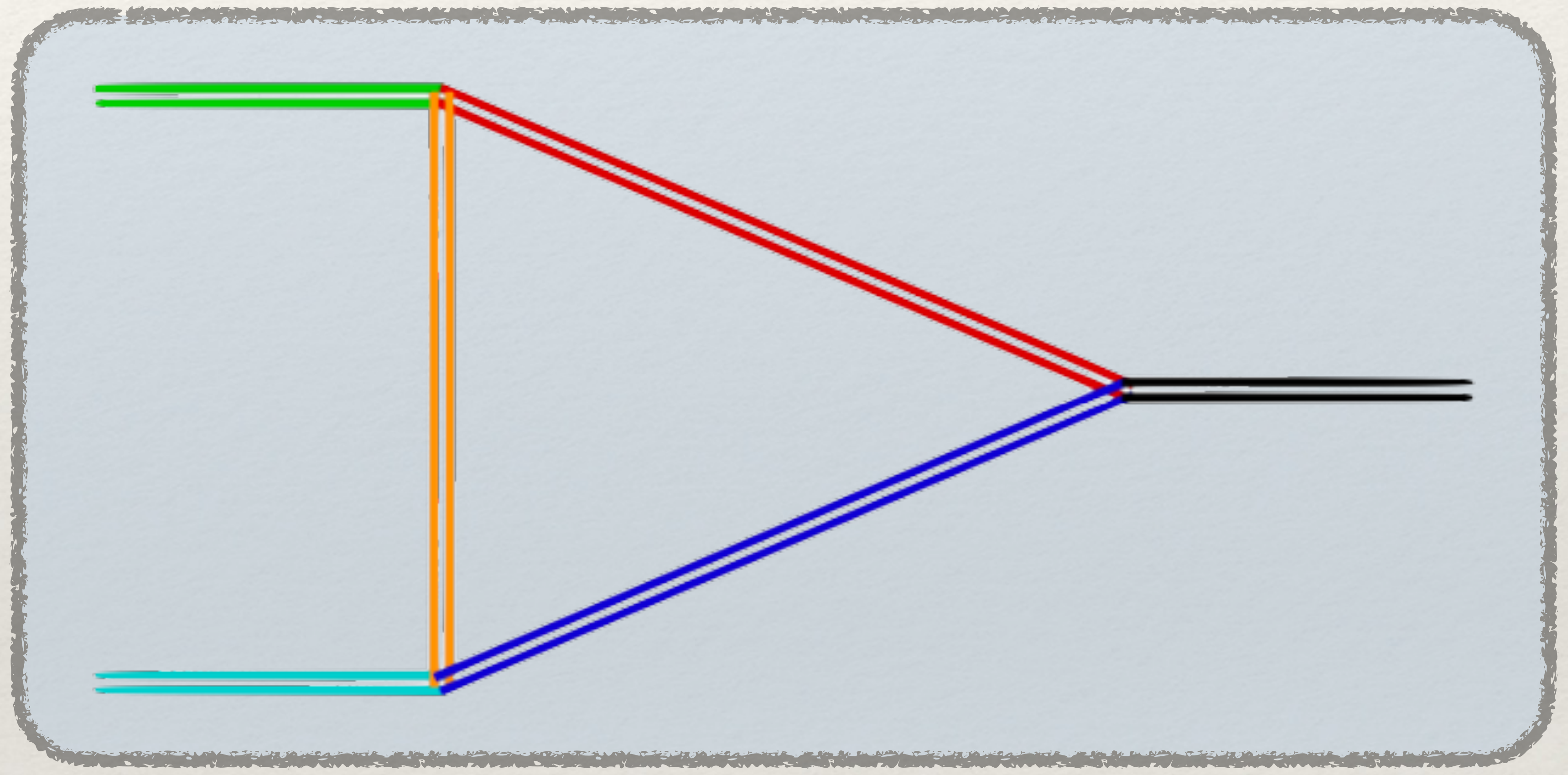
P_1: $(p_1^2, p_2^2, s, m_1^2, m_2^2, m_3^2) = (2, -1/3, 50, 5, 7, 10)$

P_2: $(p_1^2, p_2^2, s, m_1^2, m_2^2, m_3^2) = (2, -1/3, 1, 10, 10, 10)$

P_3: $(p_1^2, p_2^2, s, m_1^2, m_2^2, m_3^2) = (2, -1/3, 1, 1-i, 8/3-2i, 17-i/4)$

P_4: $(p_1^2, p_2^2, s, m_1^2, m_2^2, m_3^2) = (2, -1/3, -1, 0, 0, 0)$

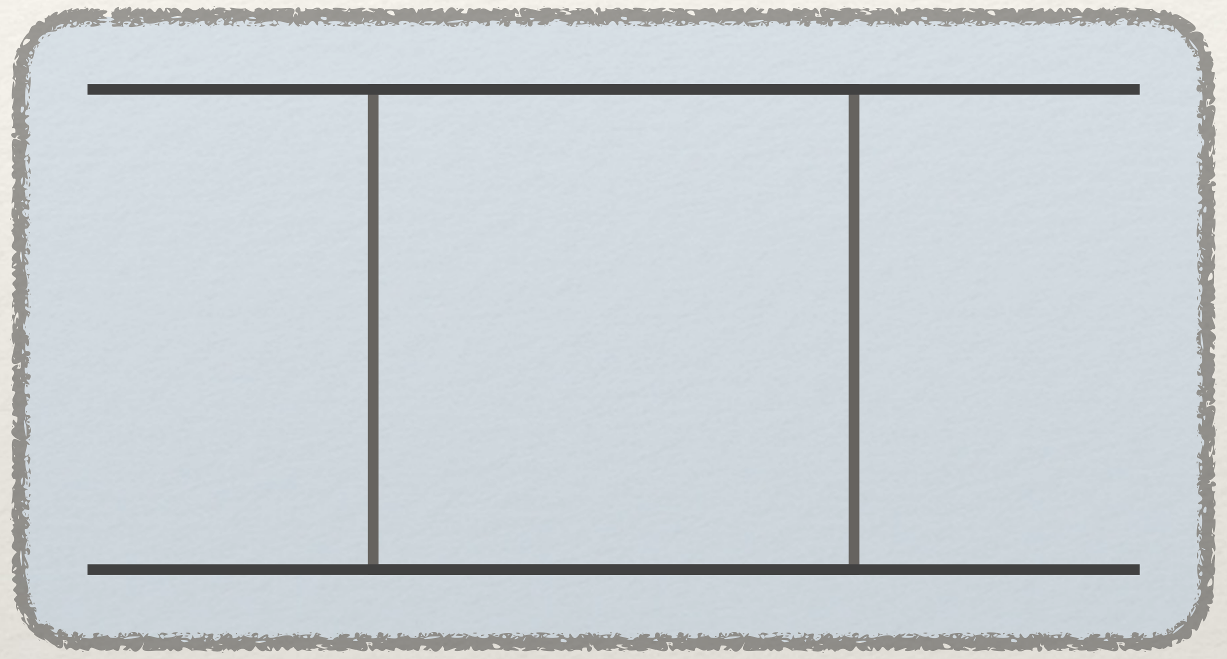
target	P_1	P_2	P_3	P_4
from	AMF^0, EBR	AMF^0, P_1	P_1	AMF^0, P_3
ϵ^{-2}	0	0	0	-1.000000000000000e0
ϵ^{-1}	0	0	0	+5.772156649015329e-1
ϵ^{0}	-7.599624851460716e-2 -1.024202715501841e-1*i	-5.114624184386078e-2	-9.105983456552547e-2 -3.405963008295366e-2*i	+6.558780715202539e-1
ϵ^{1}	+2.851448508579519e-1 +1.498241156232269e-1*i	+1.461267744725764e-1	+2.054866656214297e-1 +2.780936409230585e-2*i	+2.362111171285093e0
ϵ^{2}	-4.359339557414683e-1 -7.119426049903811e-2*i	-2.508159227043435e-1	-3.033284294289876e-1 -2.327298560596528e-2*i	+1.692738940537638e0
ϵ^{3}	+4.673966245020759e-1 +5.243128182287680e-3*i	+3.394894906445344e-1	+3.792260921703711e-1 +1.589606675868420e-2*i	+2.728361494345973e0
ϵ^{4}	-4.703087868710451e-1 +4.807793030293406e-3*i	-4.033919909274164e-1	-4.294046913943785e-1 -9.903139892953955e-3*i	+1.673348221588670e0



Examples: 1L massless box

$P_1: \ (s,t) = (1, -3)$
$P_2: \ (s,t) = (-11,5)$

target	P_1	P_2
from	AMF^0, EBR	AMF^0, P_1
ϵ^{-2}	-1.333333333333333e0	-7.272727272727273e-2
ϵ^{-1}	+1.502029078980784e0 -2.094395102393195e0*i	+1.877005278194741e-1 -1.142397328578107e-1*i
ϵ^{0}	+3.741614747275086e0 +3.509845858409871e0*i	+2.698156090946971e-3 +3.398758787451875e-1*i
ϵ^{1}	-2.706665331892672e0 +5.235878433110419e0*i	-2.846794253590710e-1 -1.143352529230017e-1*i
ϵ^{2}	-5.048478376080319e0 -1.796965802540394e0*i	+9.893611975701797e-2 -1.978243414027738e-1*i
ϵ^{3}	+6.051530711191679e-1 -7.108042701350626e0*i	+1.402991837463381e-1 -3.176949541572250e-2*i
ϵ^{4}	+6.960674788336404e0 -6.425634195584692e0*i	+1.001382259037354e-1 +7.729488085430293e-3*i

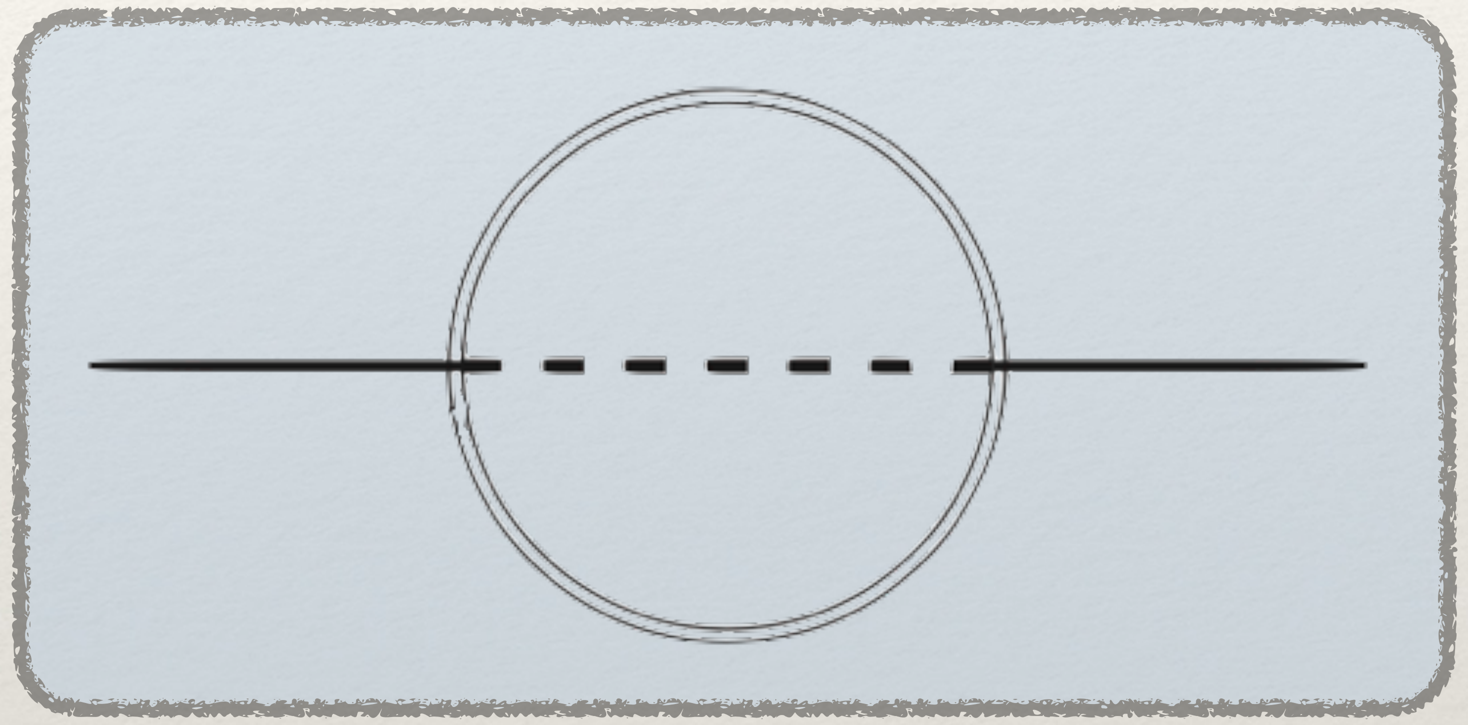


$\text{EBR}: \ u \to 0, \ t \to -s$

Examples: 2L sunrise

P_1: $(s, m_1^2, m_2^2, m_3^2) = (-1,2,3,5)$
P_2: $(s, m_1^2, m_2^2, m_3^2) = (60,2,3,5)$
P_3: $(s, m_1^2, m_2^2, m_3^2) = (-1,1,5,5)$
P_4: $(s, m_1^2, m_2^2, m_3^2) = (-1,0,0,0)$

target	P_1	P_2	P_3	P_4
from	AMF^0, EBR	AMF^0, P_1	P_1	AMF^0, P_3
ϵ^{-4}	0	0	0	0
ϵ^{-3}	0	0	0	0
ϵ^{-2}	+5.000000000000000e0	+5.000000000000000e0	+5.500000000000000e0	0
ϵ^{-1}	-3.251477438310050e0	-1.850147743831005e1	-5.693751438257865e0	+2.500000000000000e-1
ϵ^{0}	+1.188378767646979e1	+6.552872234370230e1 -1.758371010882413e1*i	+1.867337540448070e1	+1.336392167549234e0
ϵ^{1}	+1.952137703514755e1	-1.091156083475895e2 +2.967183417356042e1*i	+4.626131138234519e0	+5.066904534261821e0
ϵ^{2}	-2.160341605441262e1	+3.251877471184126e2 -1.125285386030628e1*i	+7.465797730320954e0	+1.434873581897869e1

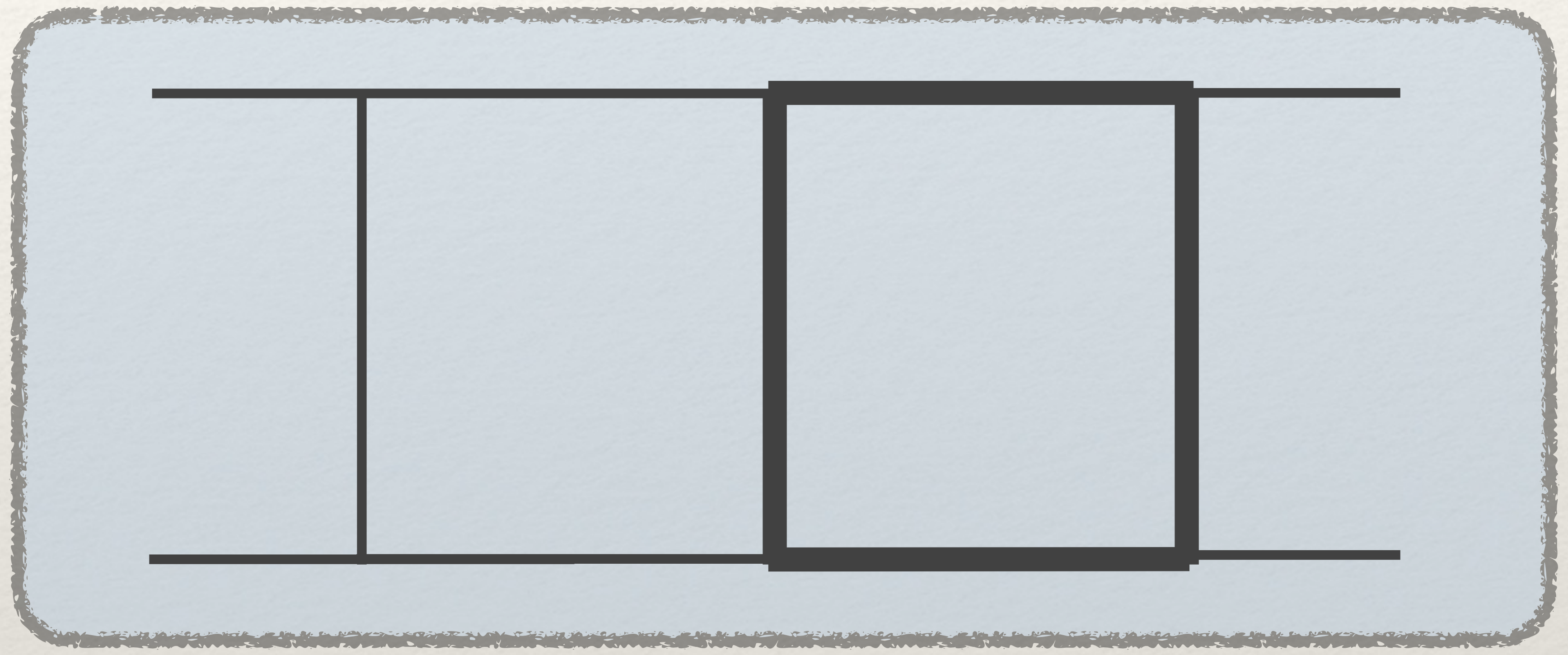


EBR : $s \to 0$

Examples: 2L planar box

P_1: $(s, t, m^2) = (-1, 2, 1)$
P_2: $(s, t, m^2) = (70, 50, 10)$
P_3: $(s, t, m^2) = (70, 50, 0)$

target	P_1	P_2	P_3
from	AMF^0, EBR	P_1	AMF^0, P_2
ϵ^{-4}	0	0	+1.632653061224490e-5
ϵ^{-3}	0	0	-1.507074533571472e-4 +1.025826172600749e-4*i
ϵ^{-2}	-1.684311982263061e-3	+7.121750612221514e-5 +1.223851404355579e-4*i	+2.720746512604996e-4 -9.469228566160803e-4*i
ϵ^{-1}	+4.026956116103587e-3	-7.645333935948279e-4 -3.758110807119310e-4*i	+1.572347464421193e-3 +3.059428585636381e-3*i
ϵ^{0}	-3.997722931454625e-3	+1.621191987913520e-3 -1.376157443003446e-4*i	-8.340803170789194e-3 -2.581654837967916e-3*i
ϵ^{1}	+6.237012138664067e-3	-2.779941041112323e-3 -3.108819053117712e-5*i	+1.483674698459523e-2 -8.593463886823766e-3*i
ϵ^{2}	-4.987777863769356e-3	+5.841649978319638e-3 -1.900890782973601e-3*i	-4.995133665555594e-3 +2.645276326148751e-2*i



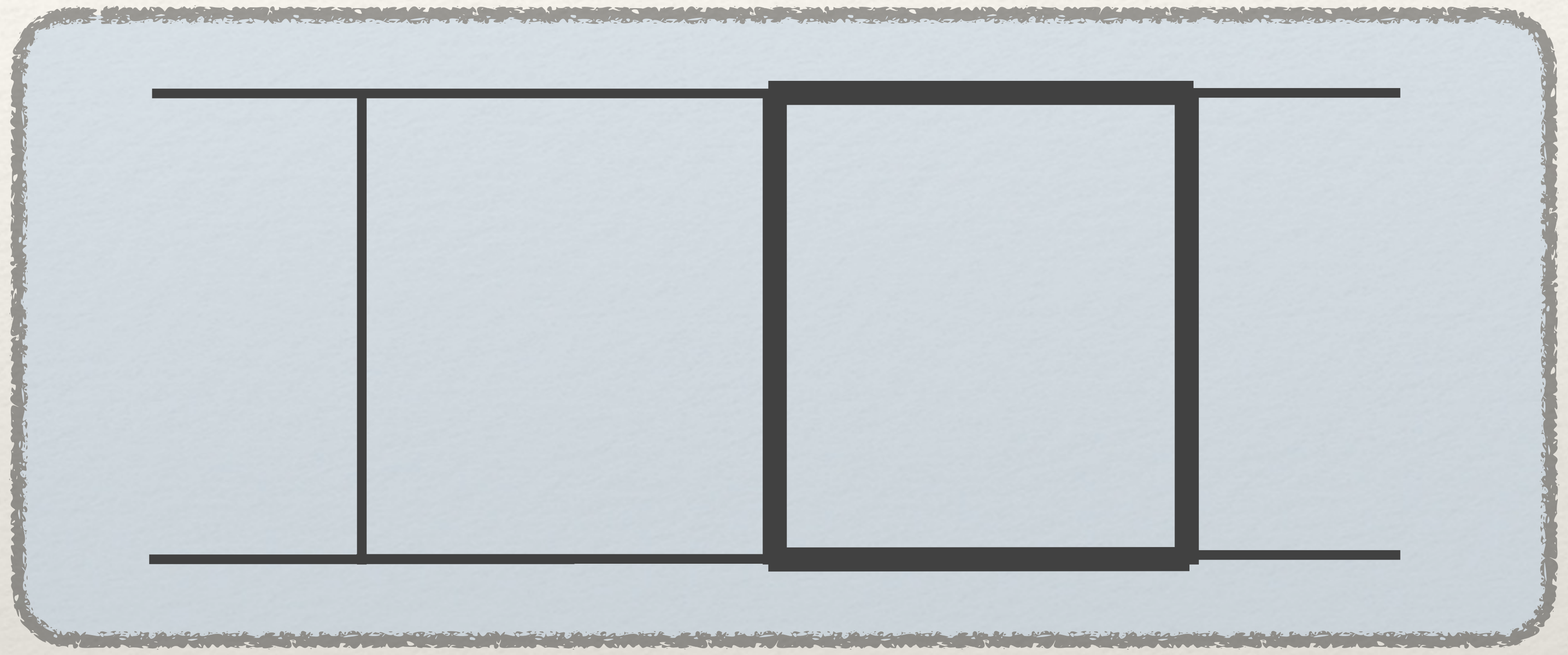
EBR : $s \to 0,\ t \to 0$

Examples: 2L planar box

P_1: $(s,t,m^2) = (-1,2,1)$
P_2: $(s,t,m^2) = (70,50,10)$
P_3: $(s,t,m^2) = (70,50,0)$

6 orders in ϵ 8 digits accuracy	**6 orders in ϵ 16 digits accuracy**	**6 orders in ϵ 32 digits accuracy**

• n. MI(DE): 32
• n. MI(ηDE): 68
• $\text{AMF}^0 - P_1$: 12 reg + 2 sing
  • kira: 133s
  • LINE(prop): 158s
  • AMFlow(prop): 1121s
• $\text{EBR}(s,t \to 0) \to P_1$:
  • LINE: 4s
• $P_1 \to P_2$: 18 reg + 4 sing
  • LINE: 23s
	• n. MI(DE): 32
	• n. MI(ηDE): 68
	• $\text{AMF}^0 - P_1$: 12 reg + 2 sing
	  • kira: 133s
	  • LINE(prop): 286s
	  • AMFlow(prop): 1740s
	• $\text{EBR}(s,t \to 0) \to P_1$:
	  • LINE: 6s
	• $P_1 \to P_2$: 18 reg + 4 sing
	  • LINE: 41s
		• n. MI(DE): 32
		• n. MI(ηDE): 68
		• $\text{AMF}^0 - P_1$: 12 reg + 2 sing
		  • kira: 133s
		  • LINE(prop): 762s
		  • AMFlow(prop): 2827s
		• $\text{EBR}(s,t \to 0) \to P_1$:
		  • LINE: 22s
		• $P_1 \to P_2$: 18 reg + 4 sing
		  • LINE: 101s

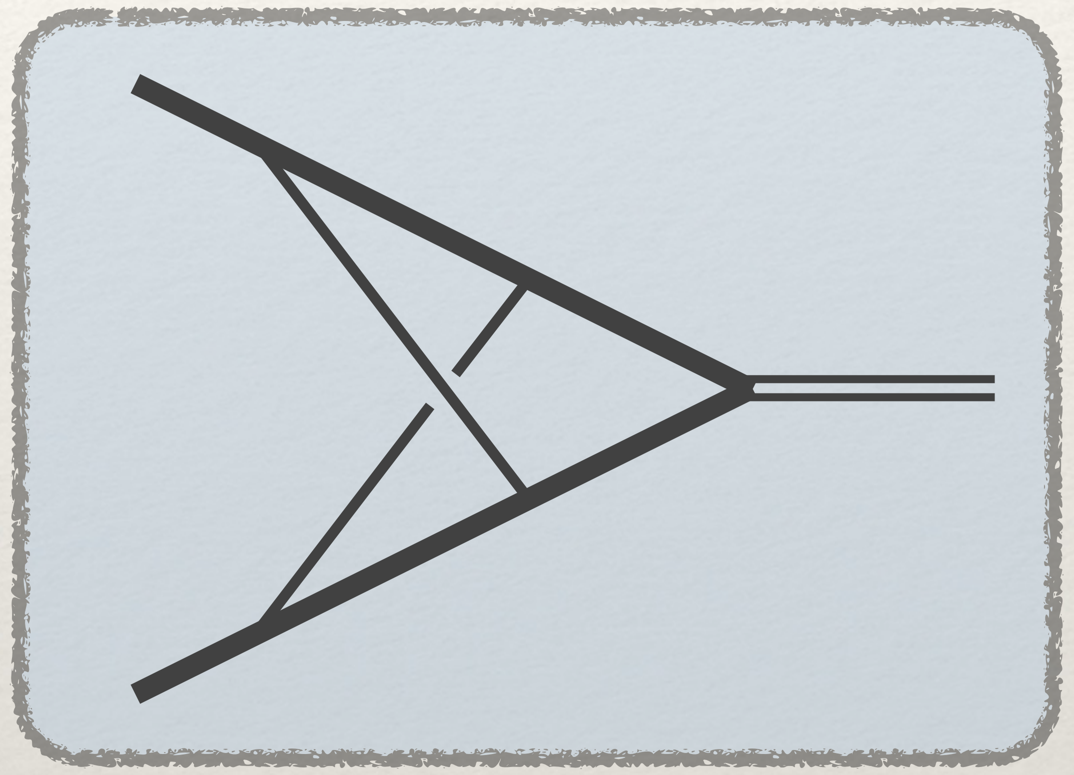


$\text{EBR}: \; s \to 0, \; t \to 0$

Examples: 2L non-planar triangle

P_1: $(s, m^2) = (10,1)$
P_2: $(s, m^2) = (1,3)$
P_3: $(s, m^2) = (1,0)$

target	P_1	P_2	P_3
from	AMF^0	P_1	AMF^0, P_2
ϵ^{-4}	0	0	+1.000000000000000e0
ϵ^{-3}	0	0	-1.154431329803066e0 +6.283185307179586e0*i
ϵ^{-2}	0	0	-2.894245735565264e1 -7.253505969566414e0*i
ϵ^{-1}	+2.532501153536048e-1 +1.376560680870821e-1*i	-3.058450755305179e-2	+6.680132569623135e-1 -9.916741832990889e1*i
ϵ^{0}	-1.137868788629137e0 +1.315450793632957e0*i	+6.882432933483959e-2	+2.306015883275194e2 -9.125506150626736e1*i
ϵ^{1}	-5.535444498587951e0 -1.578608277056101e0*i	+5.232509250247894e-2	+4.317677285401460e2 +3.615355918032282e2*i
ϵ^{2}	-1.199497745643981e1 -8.780073080609521e0*i	+8.195254040212031e-1	+1.850496772277360e1 +1.260787755350661e3*i

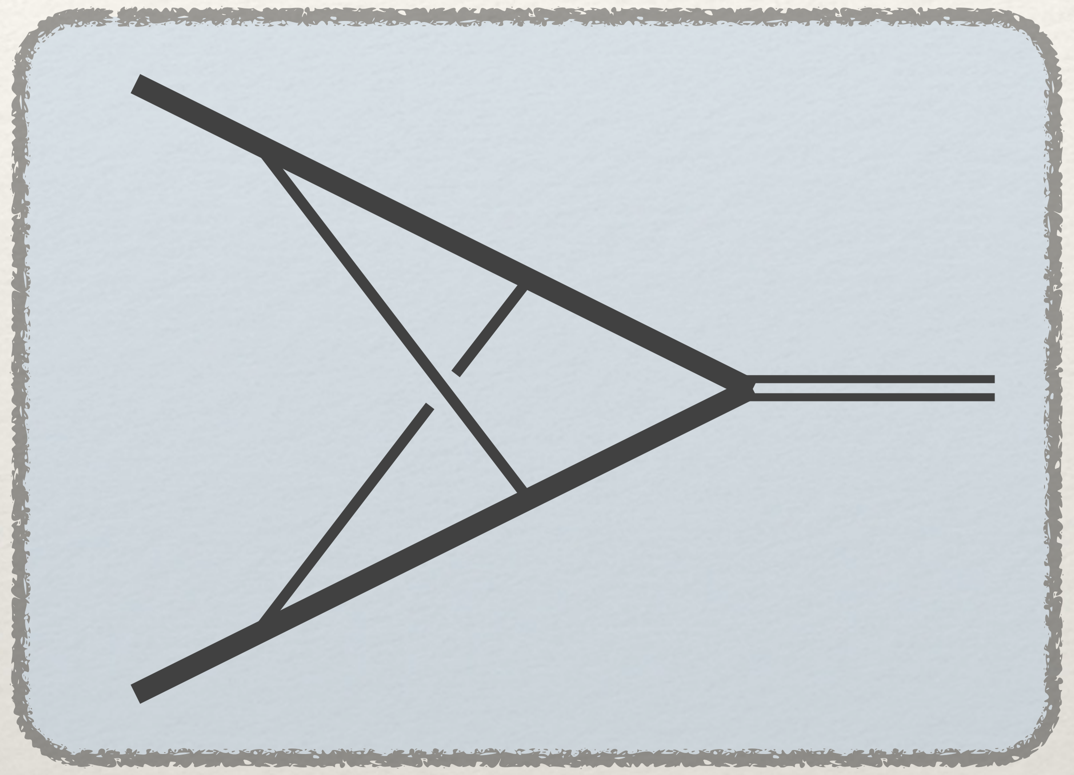


Examples: 2L non-planar triangle

P_1: $(s, m^2) = (10,1)$
P_2: $(s, m^2) = (1,3)$
P_3: $(s, m^2) = (1,0)$

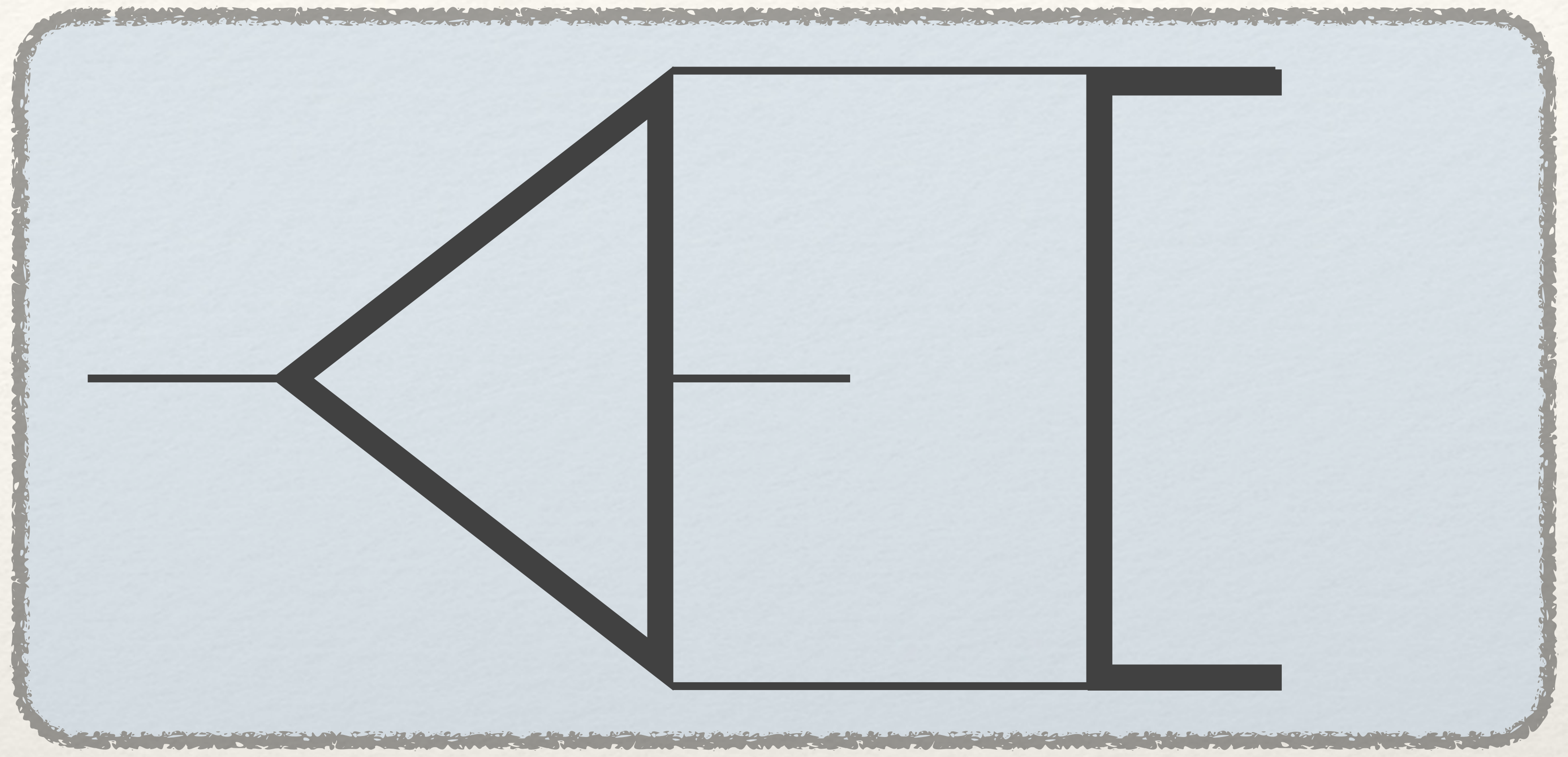
6 orders in ϵ 8 digits accuracy	**6 orders in ϵ 16 digits accuracy**	**6 orders in ϵ 32 digits accuracy**

• n. MI(DE): 16
 • n. MI(ηDE): 52
 • $\mathrm{AMF}^0 - P_1$: 16 reg + 2 sing
   • kira: 28s
   • LINE(prop): 102s
   • AMFlow(prop): 1087s
 • $P_1 \to P_2$: 5 reg + 1 sing
   • LINE: 2s
	• n. MI(DE): 16
	 • n. MI(ηDE): 52
	 • $\mathrm{AMF}^0 - P_1$: 16 reg + 2 sing
	   • kira: 28s
	   • LINE(prop): 210s
	   • AMFlow(prop): 1200s
	 • $P_1 \to P_2$: 5 reg + 1 sing
	   • LINE: 4s
		• n. MI(DE): 16
		 • n. MI(ηDE): 52
		 • $\mathrm{AMF}^0 - P_1$: 16 reg + 2 sing
		   • kira: 28s
		   • LINE(prop): 531s
		   • AMFlow(prop): 1597s
		 • $P_1 \to P_2$: 5 reg + 1 sing
		   • LINE: 8.5s



Examples: 2L non-planar box, 5 masses

P_1: $(s, t, m^2) = (3,2,1)$
P_2: $(s, t, m^2) = (5,2,1)$
P_3: $(s, t, m^2) = (2,8,1)$
P_4: $(s, t, m^2) = (2,10,1)$
P_5: $(s, t, m^2) = (-3, -5,1)$
P_6: $(s, t, m^2) = (-1, -3,1)$

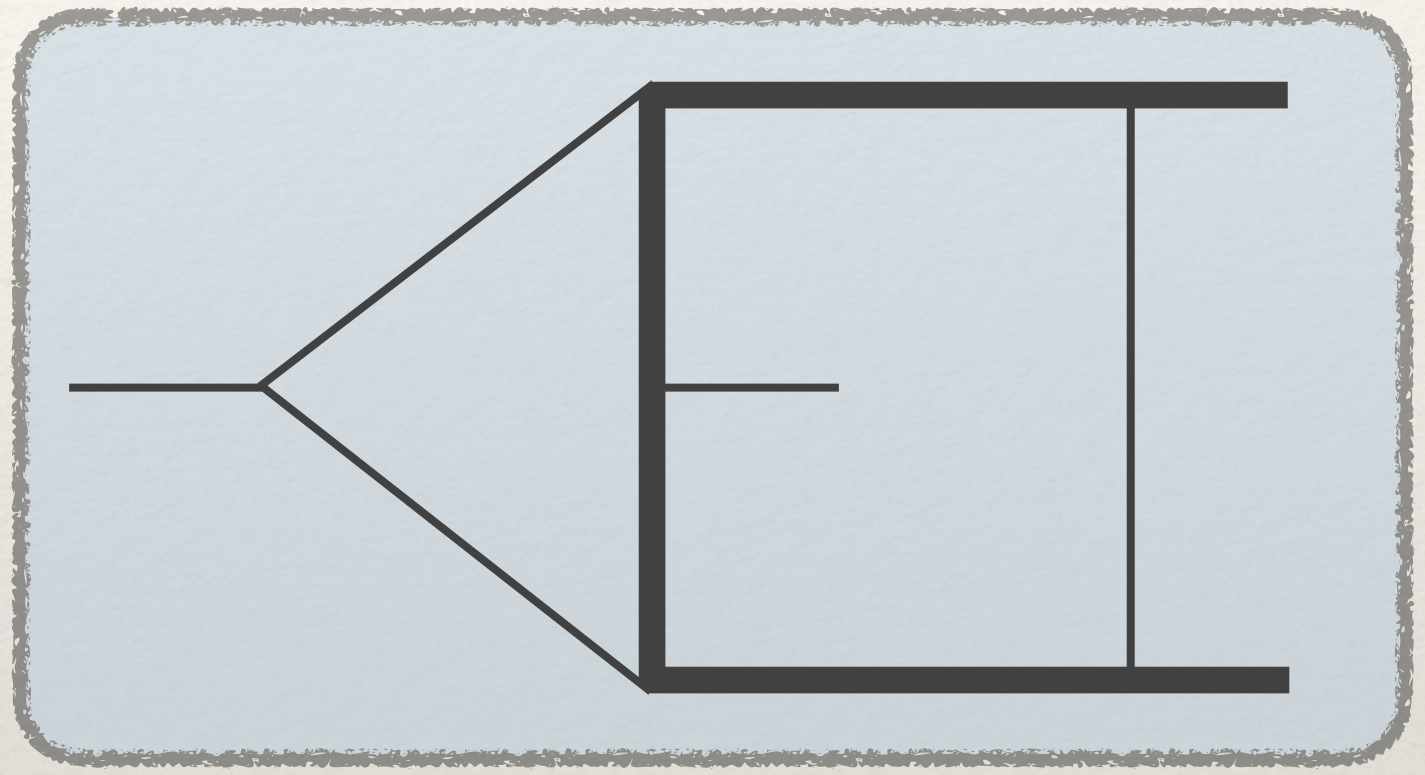


target	P_1	P_2	P_3	P_4	P_5	P_6
from	AMF^0	AMF^0, P_1	AMF^0	AMF^0, P_3	AMF^0	AMF^0, P_5
ϵ^0	+2.576938753803745e-1 -2.465521721983634e-1*i	+2.518740723653660e-1 -1.169079848124980e-1*i	+2.751593454707949e-1 -3.815281539209958e-1*i	+2.506591535092400e-1 -4.235680397875819e-1*i	-2.405260844173886e-1 -5.984661196233730e-3*i	-4.831181490833649e-1
ϵ^1	+9.839059948409147e-1 -1.447010196851563e-1*i	+8.377932210850515e-1 +2.609108724913395e-1*i	+1.257054227433279e0 +4.342974425182124e-1*i	+1.187415013371159e0 -5.997939132016630e-1*i	-5.588054474320729e-1 -3.250774673987693e-2*i	-1.396083737425863e0
ϵ^2	+1.881565035678200e0 -4.606206236766448e-3*i	+1.544162064068738e0 +1.125263466661532e0*i	+2.478546160626464e0 -2.47049854421279e-1*i	+2.372121269639779e0 -5.961585949177441e-1*i	-1.124284189077083e0 -8.467242364778369e-2*i	-3.146872480560270e0

Examples: 2L non-planar box, 4 masses

P_1: $(s, t, m^2) = (1,2,100)$
P_2: $(s, t, m^2) = (500,150,100)$

target	Q_1	Q_2
from	AMF^0	AMF^0, Q_1
ϵ^{-4}	0	0
ϵ^{-3}	-2.634309928357791e-7	+7.825617108436437e-8 -2.554478084014810e-7*i
ϵ^{-2}	+2.177434402618331e-6 -1.655185743641498e-6*i	+5.136099594647812e-9 +3.245051324395477e-6*i
ϵ^{-1}	+2.177434402618331e-6 +1.533076938553119e-5*i	+5.136099594647812e-9 -3.407024087192466e-5*i
ϵ^{0}	-2.810879169233962e-5 -3.761642841819541e-5*i	+2.470711494037188e-4 -6.343358651146831e-5*i
ϵ^{1}	+6.424181660342731e-5 +3.595559671704640e-5*i	+3.561272520516187e-5 +6.872261543040661e-4*i
ϵ^{2}	-1.721862393547420e-4 -1.231788432398794e-5*i	-7.247299398344942e-4 +6.092012063072394e-5*i

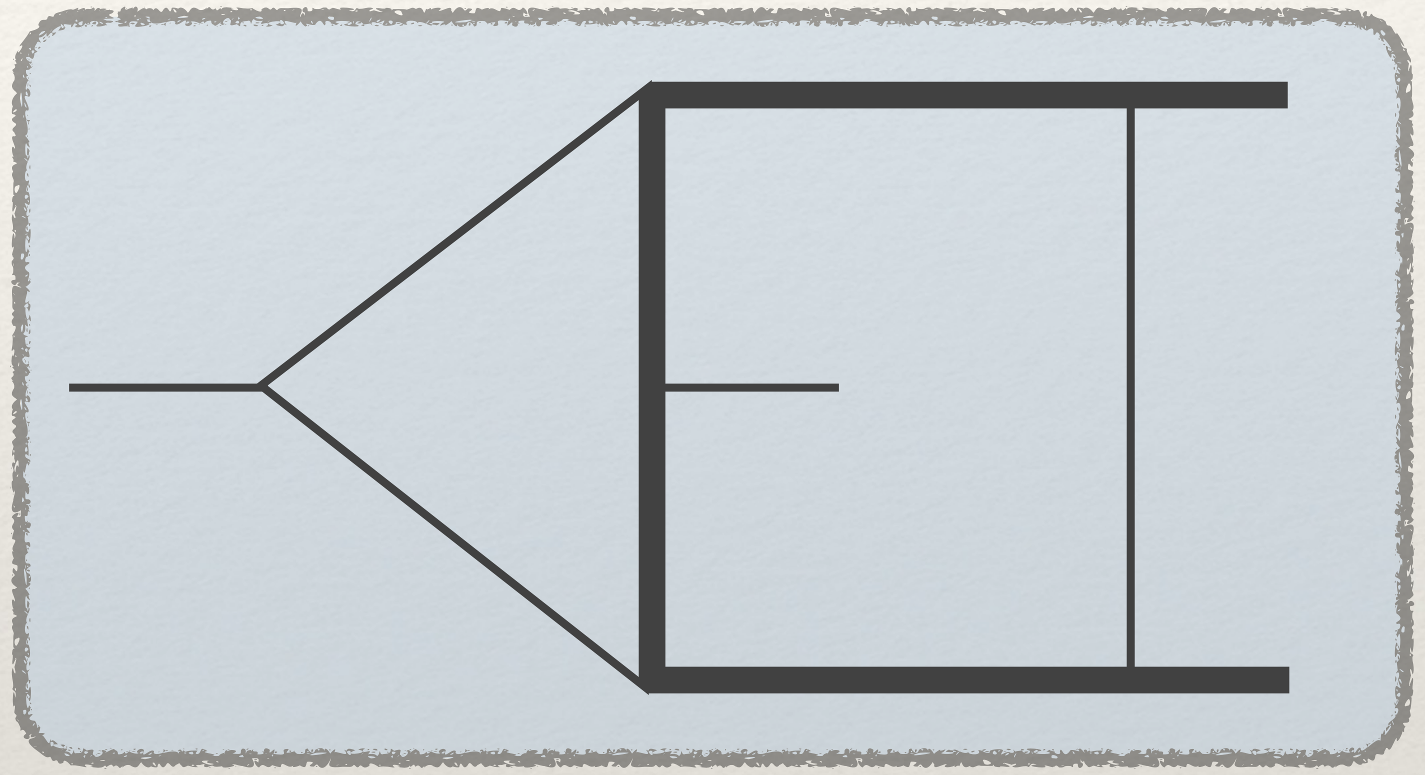


Examples: 2L non-planar box, 4 masses

P_1: $(s, t, m^2) = (1,2,100)$
P_2: $(s, t, m^2) = (500,150,100)$

6 orders in ϵ **8 digits accuracy**	**6 orders in ϵ** **16 digits accuracy**	**6 orders in ϵ** **32 digits accuracy**

• n. MI(DE): 55
• n. MI(ηDE): 144
• $\mathrm{AMF}^0 - Q_1$: 31 reg + 2 sing
  • kira: 15180s
  • LINE(prop): 3066s
• $Q_1 \to Q_2$: 26 reg + 6 sing
  • LINE: 108s
	• n. MI(DE): 55
	• n. MI(ηDE): 144
	• $\mathrm{AMF}^0 - Q_1$: 31 reg + 2 sing
	  • kira: 15180s
	  • LINE(prop): 6600s
	• $Q_1 \to Q_2$: 26 reg + 6 sing
	  • LINE: 214s
		• n. MI(DE): 55
		• n. MI(ηDE): 144
		• $\mathrm{AMF}^0 - Q_1$: 31 reg + 2 sing
		  • kira: 15180s
		  • LINE(prop): 14350s
		• $Q_1 \to Q_2$: 26 reg + 6 sing
		  • LINE: 498s



Examples

Typical computational complexity

- Addition/Subtraction: $O(n)$
- Multiplication: $O(n^{\log_2 3})$ ← Karatsuba
- Division: $O(n^2)$ ← Schoolbook long, SRT…
- Roots/Exponentiation: $O(n^2)$ ← Newton

Notice:

- $\log_2 3 = 1.585$
- Some division/roots/exponentiation algorithm share the complexity with the chosen multiplication

	binary digits		
results	32(8 dec)	64(16 dec)	107(32 dec)
internal	313	506	893
	timing(sec)		
np-triangle	102	210	531
planar box	158	286	762
np-box	3066	6600	14350

np-triangle, planar box: $[t] \propto [n]^{1.6}$

np-box: $[t] \propto [n]^{1.4}$

Conclusions

We present LINE, a **novel C implementation** of the solution of **DEs via series expansions**

- Fully **open source**, available at https://github.com/line-git/line.git
- LINE implements the **auxiliary-mass flow method**, allowing to find BC within the tool up to 2L
- Self-contained evaluation of the **numerical accuracy**

What's next?

- Exploring **expansion-by-region method** for generalizing the extraction of BC from the DE
- Testing and extending LINE at **higher loops**
- **High-level structure**, allowing new features:
  - **Unitarity cuts**
  - **Recursive BC** at $\eta \to \infty$ a la AMFlow
  - Managing **linear propagators**
  - Investigating phase space for a **smart choice of the paths**

Thank you for your attention!