Evaluation of the Boson Loop Diagram for Scalar Meson Electroproduction off the Scalar Target in 3+1 Dimensions using Light Front Dynamics

Andrew Lundeen

NC State University

October 4, 2019

Motivation

- Predict a kinematic region where beam spin asymmetry is non-zero
- Requires non-zero imaginary part of a form factor

$$J_S^\mu = F_1(q^2\Delta^\mu - q\cdot\Delta\, q^\mu) + F_2[(\bar{P}\cdot q + q^2)\Delta^\mu - q\cdot\Delta\,(\bar{P}^\mu + q^\mu)]$$

$$\frac{d\sigma^S_{\lambda=+1} - d\sigma^S_{\lambda=-1}}{d\sigma^S_{\lambda=+1} + d\sigma^S_{\lambda=-1}} = \frac{d\sigma^S_{\rm BSA}}{d\sigma^S_T(1+\epsilon\cos(2\phi)) + d\sigma^S_L\epsilon_L + d\sigma^S_{LT}\cos\phi\sqrt{\frac{1}{2}\epsilon_L(1+\epsilon)}}$$

$$\begin{bmatrix} d\sigma^S_T \\ d\sigma^S_L \\ d\sigma^S_{LT} \\ d\sigma^S_{\rm BSA} \end{bmatrix} = \begin{bmatrix} T_1 & T_2 & T_3 & 0 \\ L_1 & L_2 & L_3 & 0 \\ I_1 & I_2 & I_3 & 0 \\ 0 & 0 & 0 & S_A \end{bmatrix} \begin{bmatrix} |F_1|^2 \\ |F_2|^2 \\ F_{12}^+ \\ F_{12}^- \end{bmatrix} \qquad F_{12}^\pm = F_1F_2^* \pm F_2F_1^*$$

Plan of attack

- Use LFD. Compute the +,-, perpendicular components of J_S^μ for a given kinematics (boson loop diagrams)
- Obtain confidence in numerical result by checking that the divergence of the current vanishes
- Compare the real and imaginary components of the current to construct system of equations to solve for the real and imaginary components of F1 and F2.
- Repeat at varying Q^2, x, t values

Boson loop… Huh?

- Fermion loop is difficult. This is a “stepping stone” to tackling the more realistic model.
- For the boson loop, we will use a derivative coupling at the virtual photon vertex, the identity operator at the scalar meson vertex, and a constant vertex function for the hadronic vertices
- This is still not going to be easy to evaluate!

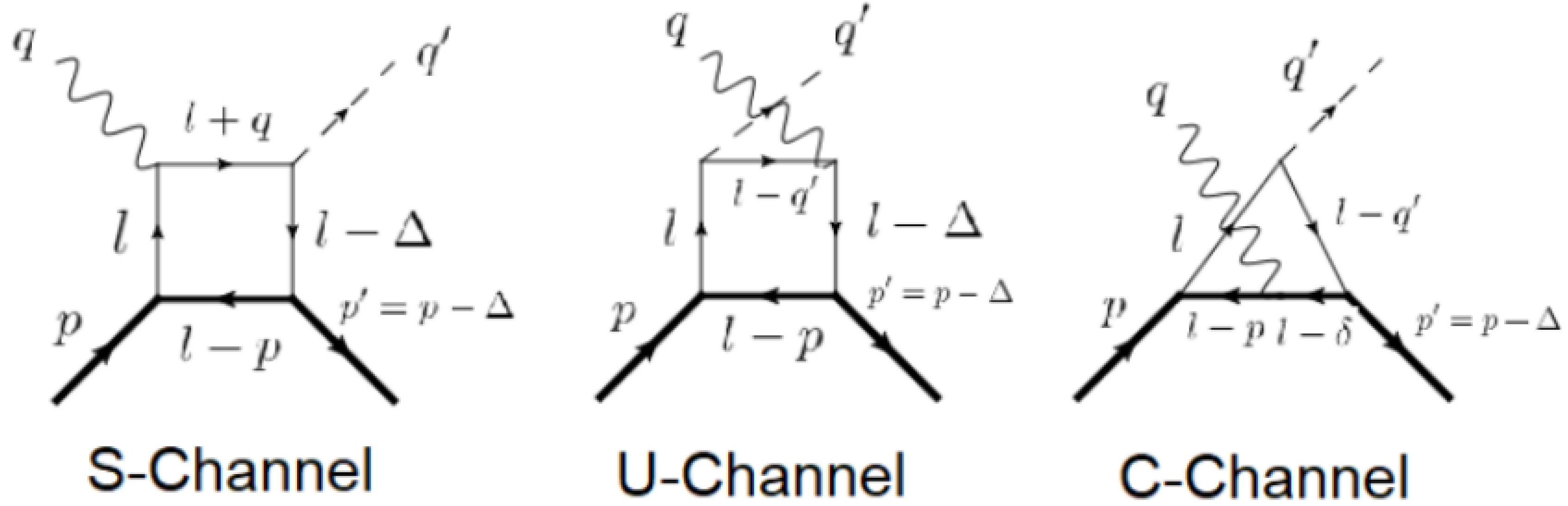


FIG. 1: The photon comes in with momentum q, the target initially is p, the final state meson has momentum q', and the final target has momentum $p' = p - \Delta$. I also introduce $\delta = p + q$. In this note I will use M_t, M_s, m, m_s to denote the masses of the target, scalar meson, boson propagator, and spectator boson propagator, respectively.

$$i\mathcal{M}_s^\mu = \mathcal{N} g^2 \int \frac{d^2 l^\perp}{(2\pi)^2} \int \frac{dl^+}{2\pi} \int \frac{dl^-}{2\pi} \frac{2l^\mu + q^\mu}{d_l d_{l+q} d_{l-\Delta} D_{l-p}}$$

$$d_l = l^2 - m^2 + i\epsilon$$
$$d_{l\pm p} = (l \pm p)^2 - m^2 + i\epsilon,$$

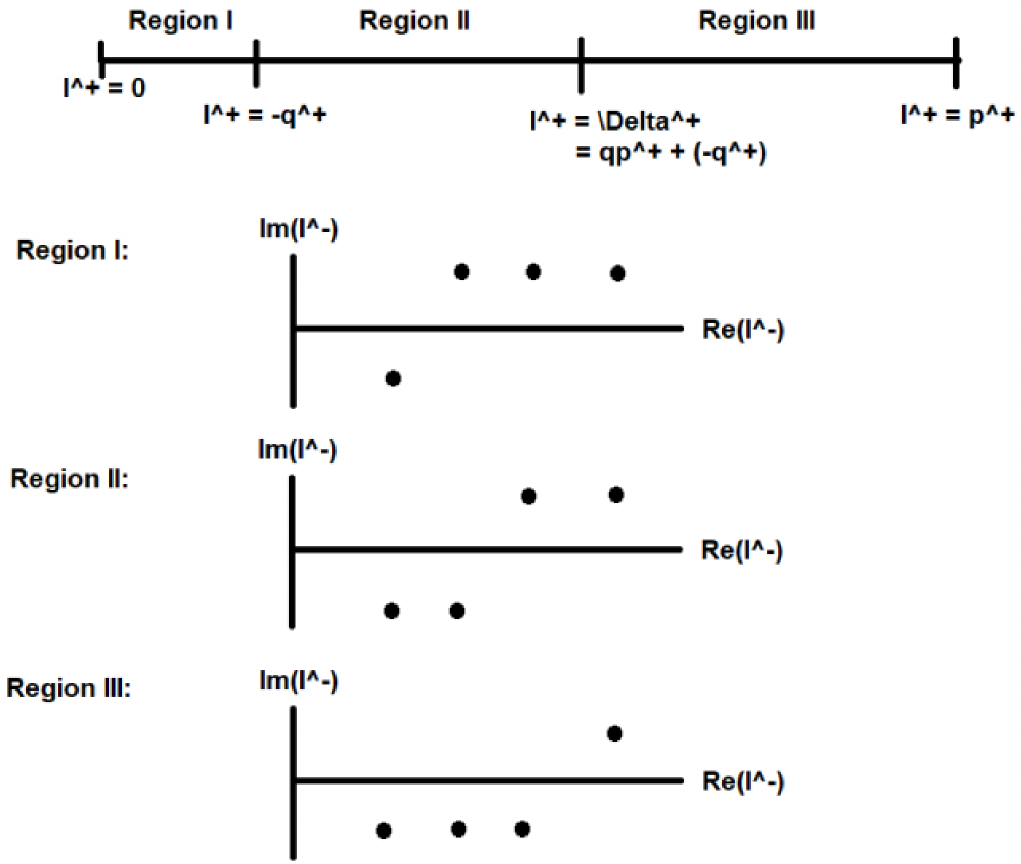
$$D_{l\pm p} = (l \pm p)^2 - m_s^2 + i\epsilon$$

$$\begin{aligned}
d_l &= 2l^+ \left\{ l^- - \left[\frac{l^{\perp 2} + m^2}{2l^+} - i\frac{\epsilon}{2l^+} \right] \right\} \\
&= 2l^+ \left(l^- - l_l^- \right) \\
d_{l\pm p} &= 2\left(l^+ \pm p^+\right) \left\{ l^- - \left[\frac{\left(l^\perp \pm p^\perp\right)^2 + m^2 \mp 2p^- \left(l^+ \pm p^+\right)}{2\left(l^+ \pm p^+\right)} - i\frac{\epsilon}{2\left(l^+ \pm p^+\right)} \right] \right\} \\
&= 2\left(l^+ \pm p^+\right) \left(l^- - l_{l\pm p}^- \right) \\
D_{l\pm p} &= 2\left(l^+ \pm p^+\right) \left\{ l^- - \left[\frac{\left(l^\perp \pm p^\perp\right)^2 + m_s^2 \mp 2p^- \left(l^+ \pm p^+\right)}{2\left(l^+ \pm p^+\right)} - i\frac{\epsilon}{2\left(l^+ \pm p^+\right)} \right] \right\} \\
&= 2\left(l^+ \pm p^+\right) \left(l^- - L_{l\pm p}^- \right)
\end{aligned}$$

Integration over l^-: Depends on the position of the poles!

	Domain 1	Domain 2	Domain 3	Domain 4	Domain 5
	0 -q^+	qp^+	\Delta^+	\delta^+	p^+
s	Region I	Region II		Region III	
u	Region I		Region II	Region III	
c	Region I		Region II		Region III

S-channel example:



$$
\begin{aligned}
i\mathcal{M}_a^\mu = \mathcal{N}_g \int_0^\infty |l^\perp| \,\mathrm{d}\,|l^\perp| \int_0^{2\pi} \mathrm{d}\psi \Bigg\{ - \int_{p_{a,1}}^{p_{a,2}} \frac{dl^+}{D_a^+} \left[\frac{N_a^\mu}{(l^- - l^-_{l-p_{a,2}})(l^- - l^a_{l-p_{a,3}})(l^- - L^-_{l-p_{a,4}})} \right]_{l^-=l_l^-} \\
+ \Bigg(\int_{p_{a,2}}^{p_{a,3}} \frac{dl^+}{D_a^+} \left[\frac{N_a^\mu}{(l^- - l_l^-)(l^- - l^-_{l-p_{a,2}})(l^- - L^-_{l-p_{a,4}})} \right]_{l^-=l^a_{l-p_{a,3}}} + \int_{p_{a,2}}^{p_{a,3}} \frac{dl^+}{D_a^+} \left[\frac{N_a^\mu}{(l^- - l_l^-)(l^- - l^-_{l-p_{a,2}})(l^- - l^a_{l-p_{a,3}})} \right]_{l^-=L^-_{l-p_{a,4}}} \Bigg) \\
+ \int_{p_{a,3}}^{p_{a,4}} \frac{dl^+}{D_a^+} \left[\frac{N_a^\mu}{(l^- - l_l^-)(l^- - l^-_{l-p_{a,2}})(l^- - l^a_{l-p_{a,3}})} \right]_{l^-=L^-_{l-p_{a,4}}} \Bigg\},
\end{aligned}
$$

where $a \in \{s, u, c\}$, $\mu \in \{+, \perp, -\}$, $\mathcal{N}_g = 2\pi i \frac{\mathcal{N} g^2}{2^4 (2\pi)^4}$, and

$$
\begin{aligned}
l &= \{l^+, |l^\perp| \cos\psi, |l^\perp| \sin\psi, l^-\} \\
p_{a,1} &= 0;\ p_{s,2} = -q^+;\ p_{u=c,2} = q'^+;\ p_{s=u,3} = \Delta^+;\ p_{c,3} = \delta^+;\ p_{a,4} = p^+; \\
D_s^+ &= l^+(l^+ + q^+)(l^+ - \Delta^+)(l^+ - p^+) \\
D_u^+ &= l^+(l^+ - q'^+)(l^+ - \Delta^+)(l^+ - p^+) \\
D_c^+ &= l^+(l^+ - q'^+)(l^+ - \delta^+)(l^+ - p^+) \\
N_s^\mu &= 2l^\mu + q^\mu \\
N_u^\mu &= 2(l^\mu - q'^\mu) + q^\mu \\
N_c^\mu &= 2l^\mu - \delta^\mu \\
l^{s=u}_{l-p_{a,3}} &= l^-_{l-\Delta};\, l^c_{l-p_{a,3}} = L^-_{l-\delta}.
\end{aligned}
$$

For example,

$$i\mathcal{M}_c^- = \mathcal{N}_g \int_0^\infty |l^\perp| \, \mathrm{d} \left|l^\perp\right| \int_0^{2\pi} \mathrm{d}\psi \Bigg\{ - \int_0^{q'^+} \frac{dl^+}{D_c^+} \left[\frac{2l^- - \delta^-}{(l^- - l_{l-q'}^-)(l^- - L_{l-\delta}^-)(l^- - L_{l-p}^-)} \right]_{l^- = l_l^-}$$

$$+ \Bigg(\int_{q'^+}^{\delta^+} \frac{dl^+}{D_c^+} \left[\frac{2l^- - \delta^-}{(l^- - l_l^-)(l^- - l_{l-q'}^-)(l^- - L_{l-p}^-)} \right]_{l^- = L_{l-\delta}^-} + \int_{q'^+}^{\delta^+} \frac{dl^+}{D_c^+} \left[\frac{2l^- - \delta^-}{(l^- - l_l^-)(l^- - l_{l-q'}^-)(l^- - L_{l-\delta}^-)} \right]_{l^- = L_{l-p}^-} \Bigg)$$

$$+ \int_{\delta^+}^{p^+} \frac{dl^+}{D_c^+} \left[\frac{2l^- - \delta^-}{(l^- - l_l^-)(l^- - l_{l-q'}^-)(l^- - L_{l-\delta}^-)} \right]_{l^- = L_{l-p}^-} \Bigg\}$$

Opportunity to check numerical work

- Do the integrands within each channel match at the boundaries of the regions?
- At $l^+ = -q^+$, qp^+, Δ^+, and δ^+, does the sum from the 3 channels match from the “left” and the “right”?

l^+ Integration

- Before, the poles only depended on one integration variable
- Now there can be poles that are functions of both l^perp and \psi
- Need to find terms causing poles, and find solutions

$$l^+ = l^+_\pm \equiv \frac{-a_1 \pm \sqrt{|a_1|^2 - 4a_2a_0}}{2a_2}$$

$$\begin{aligned} c_r &= |l^\perp|^2 - r^2 \pm_r 2|l^\perp| \left(r^1 \cos\psi + r^2 \sin\psi\right) + m^2 \\ a_2 &= 2(\pm_q q^- \mp_p p^-) \\ a_1 &= \pm 2p^+q^- \mp 2p^-q^+ + c_p - c_q \\ a_0 &= \pm_q q^+ c_p \mp_p p^+ c_q \end{aligned} \qquad (12)$$

Note: the $\pm$ signs in the definitions of a_2, a_1, and a_0 are a little tricky. If there is a subscript, match it to the appropriate $l^-_{l\pm p}$. If there is no subscript, it depends on the sign of both. Same sign means take the top of the $\pm$ or $\mp$, different sign means take the bottom.

How to find which terms cause poles?

- For a while, I would create animations of 3D plots, but since then have simply tried to brute-force Nintegrate and if I got error messages, assumed a pole problem occurred
- After getting the error messages, looked at each denominator term and visually searched for contours that crossed 0

l^+ Integration

- Poles depend on which term is causing the pole, the values of the kinematics, and are functions of the integration variables
- Need to split the integration region into four possibilities (if just one term causes poles):
  - Region I: Where double poles occur. I tried to make this region empty by choosing appropriate kinematics.
  - Regions II and III: Where one simple pole exists, but the other is outside the bounds of integration for a given l^perp and \psi range
  - Region IV: Where two simple poles exist

C. Treating the case of one simple pole or no poles

Suppose one pole, say l_-^+ exists within the integration range. Then the integral can be evaluated as

$$\lim_{\epsilon\to 0^+}\int_0^{q'^+} dl^+ \frac{f\left(l^+\right)}{a_2\left(l^+ - l_+^+\right)\left(l^+ - l_-^+\right) + i\epsilon} = \mathcal{P}\left(\int_0^{\left(q'^+ + l_-^+\right)/2} dl^+ \frac{f_+\left(l^+\right)}{l^+ - l_-^+}\right) - i\pi f_+\left(l_-^+\right) + \int_{\left(q'^+ + l_-^+\right)/2}^{q'^+} dl^+ \frac{f\left(l^+\right)}{a_2\left(l^+ - l_+^+\right)\left(l^+ - l_-^+\right)} \tag{25}$$

$$f(l^+, l^\perp, \psi) = \left[\frac{N_u^\mu}{\left(l^+ - \Delta^+\right)\left(l^+ - p^+\right)\left(l_l^- - l_{l-\Delta}^-\right)\left(l_l^- - L_{l-p}^-\right)}\right]_{\substack{l^- = l_l^- \\ \epsilon = 0}}$$

$$f_+(l^+) = \frac{f\left(l^+\right)}{a_2\left(l^+ - l_+^+\right)}; \quad f_-(l^+) = \frac{f\left(l^+\right)}{a_2\left(l^+ - l_-^+\right)}$$

For better numerical accuracy, I actually need to place the boundary of integration at the pole and do two principal value integrations

B. Treating the case of two simple poles

If two simple poles exist, then the integral just needs to be split into two parts. Using the real line version of the Sokhotski Plemelj theorem, I can write

$$\lim_{\epsilon \to 0^+} \int_0^{q'^+} dl^+ \frac{f(l^+)}{a_2 (l^+ - l^+_+)(l^+ - l^+_-) + i\epsilon} = \mathcal{P}\left(\int_0^{(l^+_+ + l^+_-)/2} dl^+ \frac{f_+(l^+)}{l^+ - l^+_-} \right) - i\pi f_+(l^+_-) + \mathcal{P}\left(\int_{(l^+_+ + l^+_-)/2}^{q'^+} dl^+ \frac{f_-(l^+)}{l^+ - l^+_+} \right) - i\pi f_-(l^+_+) \tag{23}$$

where I have "absorbed" the non-singular denominators in each integrand into the functions f_+ and f_- as

$$f_+(l^+) = \frac{f(l^+)}{a_2 (l^+ - l^+_+)}; \quad f_-(l^+) = \frac{f(l^+)}{a_2 (l^+ - l^+_-)} \tag{24}$$

$$f(l^+, l^\perp, \psi) = \left[\frac{N^\mu_u}{(l^+ - \Delta^+)(l^+ - p^+)(l^-_l - l^-_{l-\Delta})(l^-_l - L^-_{l-p})} \right]_{\substack{l^- = l^-_l \\ \epsilon = 0}}$$

For better numerical accuracy, I actually need to split the integrals in Equation 23 into 3 integrals, placing the boundaries of the integrals at the singular values

Example: Handling a pole in S-channel Region III

```
Clear[θ]
ϵ = 0;
Q = 3;
Mt = 4; (* Target mass *)
Ms = 0.98; (* Meson mass *)
xBj = 0.45; (* Bjorken x *)
Eqp = 1.68; (* Meson energy *)
ϕ = 0; (* Lepton-hadron plane angle *)
m = 1; (* Light boson propagator mass *)
ms = 3; (* "Bottom line" heavy boson propagator mass *)
ν = Q^2 / (2 Mt * xBj); (* Virtual photon energy *)

θ = ArcCos[Solve[((mlfdot[plf - Δlf, plf - Δlf]) /. {Cos[θ] → cosθ}) == Mt^2, cosθ][[1, 1, 2]]];
(* Angle between virtual photon 3-momentum (z-axis) and produced meson. *)

kinvalues =
 Grid[{{"Q²", "θ", "t", "|t|/Q²", "qplf⁺", "-qlf⁺", "Δlf⁺", "δlf⁺", "plf⁺"},
   {Q^2, θ, t, Abs[t] / Q^2, qplf[[1]], -qlf[[1]], Δlf[[1]], δlf[[1]], N[plf[[1]]]}}, Frame → All]

mlfdot[qplf, qplf] - Ms^2 (* Should be 0 *)
(mlfdot[plf - Δlf, plf - Δlf]) - Mt^2 (* Should be 0 *)
((plf - Δlf) + qplf) - (plf + qlf) (* Should be a 0 vector *)
```

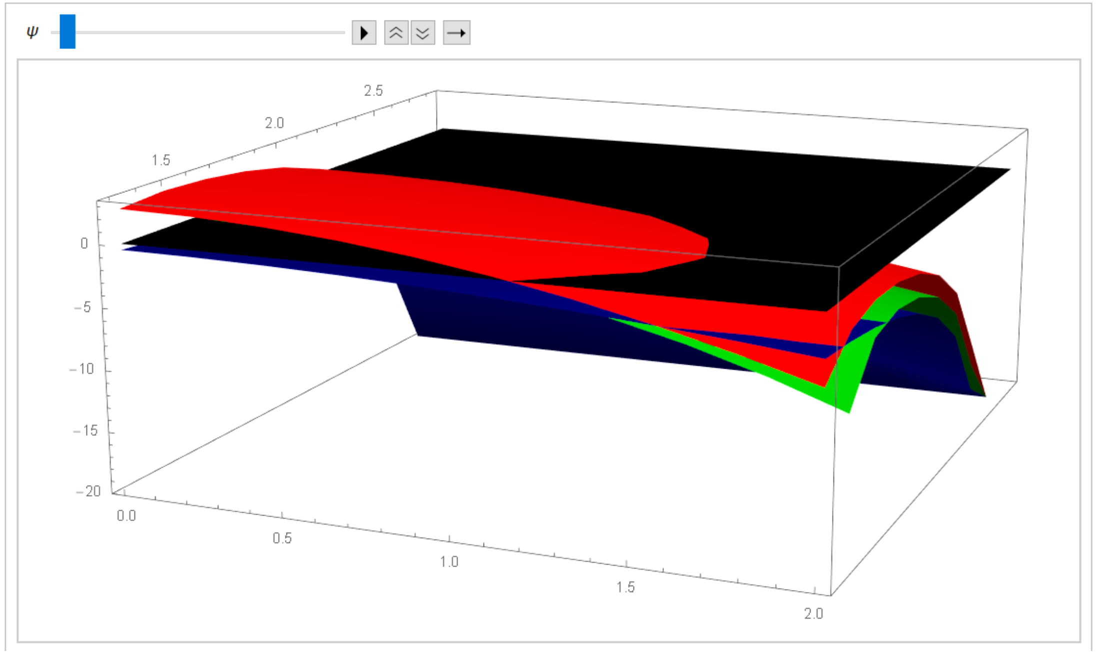
Q^2	θ	t	$\lvert t\rvert/Q^2$	$qplf^+$	$-qlf^+$	Δlf^+	δlf^+	plf^+
9	2.75716	-6.56	0.728889	0.293482	0.993573	1.28706	1.83485	2.82843

NIntegrate: The global error of the strategy GlobalAdaptive has increased more than 2000 times. The global error is expected to decrease monotonically after a number of integrand evaluations. Suspect one of the following: the working precision is insufficient for the specified precision goal; the integrand is highly oscillatory or it is not a (piecewise) smooth function; or the true value of the integral is 0. Increasing the value of the GlobalAdaptive option MaxErrorIncreases might lead to a convergent numerical integration. NIntegrate obtained −1.28964 and 3.4278074457925407` for the integral and error estimates.

^ Obviously, there is a problem!

S-Channel Region 3 pole term: (Lm[-plf]-lm[qlf])

```
In[319]:= (* S-channel Region III *)
Lmplfll[lperp_, ψ_, lplus_] = (Lm[-plf] - ll);
Lmplflmqlf[lperp_, ψ_, lplus_] = (Lm[-plf] - lm[qlf]);
Lmplflm∆lf[lperp_, ψ_, lplus_] = (Lm[-plf] - lm[-∆lf]);
Animate[Plot3D[{Lmplfll[lperp, ψ, lplus], Lmplflmqlf[lperp, ψ, lplus], Lmplflm∆lf[lperp, ψ, lplus], 0}, {lperp, 0, 2},
  {lplus, ∆lf[[1]], N[plf[[1]]]}, PlotStyle → {Blue, Red, Green, Black}] , {ψ, 0, 2 Pi, 0.2}]
```



For all values of \psi, only the red plane has a contour that crosses 0, the black surface

Difficulties with defining geometric regions to numerically integrate over

```
DPRsIII1 = ImplicitRegion[lplusplus[lperp, ψ, a2sIII1q, a1sIII1q[lperp, ψ], a0sIII1q[lperp, ψ]][[2, 2, 1]] ≥ 0 && lplusplus[lperp, ψ, a2sIII1q, a1sIII1q[lperp, ψ], a0sIII1q[lperp, ψ]][[2, 2, 1]] ≤ 0 &&
    lplusplus[lperp, ψ, a2sIII1q, a1sIII1q[lperp, ψ], a0sIII1q[lperp, ψ]] ≤ N[plf[[1]]], {{lplus, Δlf[[1]], plf[[1]]}, {lperp, 0, Infinity}, {ψ, 0, 2 Pi}}];
(* Double pole region for S-channel Region III integrand 1. I really want this to be an empty region! *)
DSPRsIII1 = ImplicitRegion[lplusplus[lperp, ψ, a2sIII1q, a1sIII1q[lperp, ψ], a0sIII1q[lperp, ψ]][[2, 2, 1]] > 0 && lplusplus[lperp, ψ, a2sIII1q, a1sIII1q[lperp, ψ], a0sIII1q[lperp, ψ]] ≤ N[plf[[1]]] &&
    Δlf[[1]] ≤ lplusminus[lperp, ψ, a2sIII1q, a1sIII1q[lperp, ψ], a0sIII1q[lperp, ψ]], {{lplus, Δlf[[1]], plf[[1]]}, {lperp, 0, Infinity}, {ψ, 0, 2 Pi}}];
(* Variable notation: double simple pole region for the S-channel region III integrand 1. *)
sIII1 = ImplicitRegion[lperp ≥ 0 || lperp < 0, {{lplus, Δlf[[1]], plf[[1]]}, {lperp, 0, Infinity}, {ψ, 0, 2 Pi}}]; (* Putting in a region argument that is always true, I don't know how else to define an ImplicitRegion *)
(* I know that lpluslplus and lplusminus are real and in range when lperp=1.4 and ψ=1 from the above. I want to check for the non-emptiness of DSPRsIII1 by seeing if this point is in fact in that region!*)

{2, 1.4, 1} ∈ DSPRsIII1 (* Should return True *)
{2, 0.05, 1} ∈ DSPRsIII1 (* Should return False *)

RegionEqual[DPRsIII1, EmptyRegion[3]] (* If this is True, there are no double poles! *)
DiscretizeRegion[DSPRsIII1] (* This needs to give me some output without an error otherwise there is no hope I can numerically integrate over this region *)
DiscretizeRegion[sIII1] (* Same here as I'm going to effectively use this to define the region that I haven't integrated over by the end *)
]= True

]= False

]= True
```

I need to put in enough arguments to correctly specify my regions, but few enough and simple enough arguments that Mathematica is able to automatically discretize my region! If it fails to discretize, there is no hope that it can numerically Integrate the region. I also want to make sure no double poles occur: notice how my lplusplus arguments are functions of functions of lperp, \psi

$$: \mathcal{P}\left(\int_{0}^{(l_+^{\pm}+l_-^{\pm})/2} dl^+ \frac{f_+(l^+)}{l^+ - l_-^{\pm}}\right)$$

```
part1 = NIntegrate[lperp * ƒplussIIIplus[lplus, lperp, ψ] / (lplus - lplusminus[lperp, ψ, a2sIII1q, a1sIII1q[lperp, ψ], a0sIII1q[lperp, ψ]]),
  {lplus, lperp, ψ} ∈ ImplicitRegion[lplusplus[lperp, ψ, a2sIII1q, a1sIII1q[lperp, ψ], a0sIII1q[lperp, ψ]][[2, 2, 1]] > 0 && lplusplus[lperp, ψ, a2sIII1q, a1sIII1q[lperp, ψ], a0sIII1q[lperp, ψ]] ≤ N[plf[[1]]] &&
    Δlf[[1]] ≤ lplusminus[lperp, ψ, a2sIII1q, a1sIII1q[lperp, ψ], a0sIII1q[lperp, ψ]], {{lplus, Δlf[[1]], (lplusminus[lperp, ψ, a2sIII1q, a1sIII1q[lperp, ψ], a0sIII1q[lperp, ψ]])}, {lperp, 0, Infinity}, {ψ, 0, 2 Pi}}],
  Method → {"SymbolicDomainDecomposition", Method → {"GlobalAdaptive", "MaxErrorIncreases" → 2000}}];
```

NIntegrate: Numerical integration converging too slowly; suspect one of the following: singularity, value of the integration is 0, highly oscillatory integrand, or WorkingPrecision too small.

NIntegrate: The global error of the strategy GlobalAdaptive has increased more than 2000 times. The global error is expected to decrease monotonically after a number of integrand evaluations. Suspect one of the following: the working precision is insufficient for the specified precision goal; the integrand is highly oscillatory or it is not a (piecewise) smooth function; or the true value of the integral is 0. Increasing the value of the GlobalAdaptive option MaxErrorIncreases might lead to a convergent numerical integration. NIntegrate obtained 9.282848755020739` and 0.178460470998281` for the integral and error estimates.

In case that's too small to read: I tried to numerically integrate over a region that had the pole on the boundary of the region of integration, and I was somewhat successful. I got some value around 10 with an error of about 0.2. I tried increasing the value of MaxErrorIncreases, but this heavily increases integration time with minimal error reduction.

I suspect that this is because the integrand is a "nasty" function…

(iii) The integration is badly conditioned [KrUeb98]. For example, the reason might be that the integrand is defined by complicated expressions or in terms of approximate solutions of mathematical problems (such as differential equations or nonlinear algebraic equations).

The strategy "GlobalAdaptive" keeps track of the number of times the total error estimate has not decreased after the bisection of the region with the largest error estimate. When that number becomes bigger than the value of the "GlobalAdaptive" option "MaxErrorIncreases", the integration stops with a message (NIntegrate::eincr).

The default value of "MaxErrorIncreases" is 400 for one-dimensional integrals and 2000 for multidimensional integrals.

I could increase the WorkingPrecision but…

- I put more digits into the kinematics variables, which effectively raised WorkingPrecision from 4->16. The error dropped from 0.17 to 0.12. I could keep going, but…

Expect solution times to increase exponentially as a function of working precision:

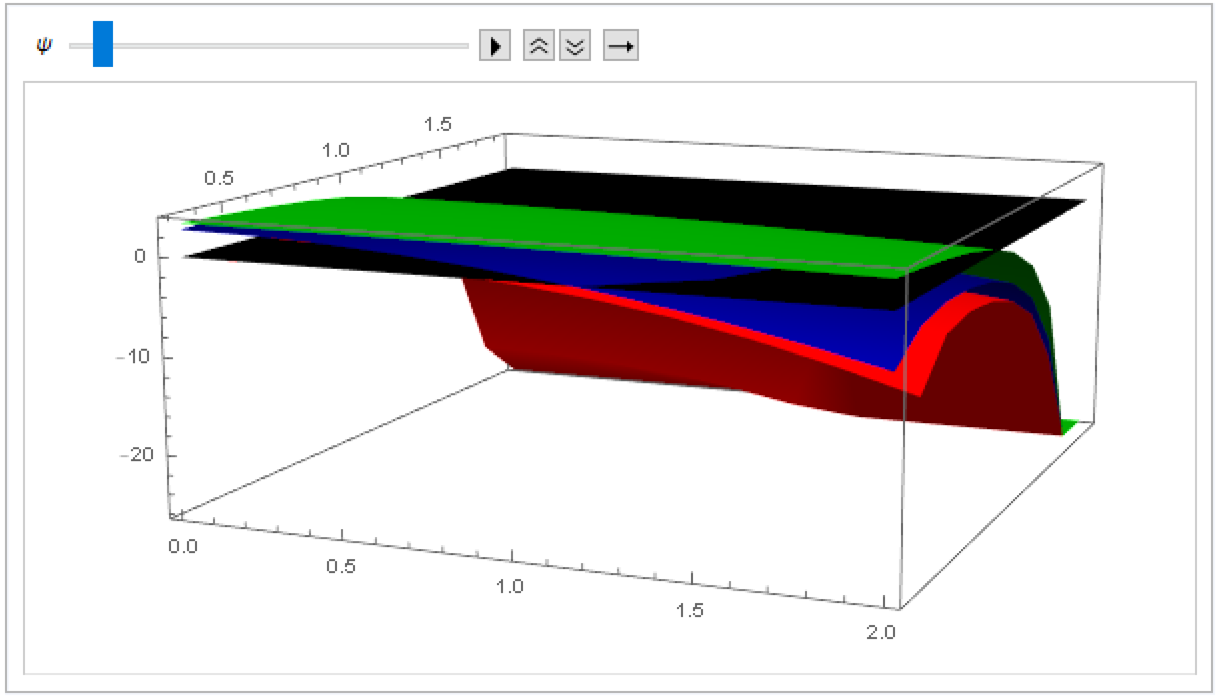
```
In[1]:= deqn = Block[{γ = 15/100, ϵ = 3/10, ω = 1},
           {x''[t] + γ x'[t] - x[t] + x[t]^3 == ϵ Cos[ω t], x[0] == 1, x'[0] == 0}];

In[2]:= times = Table[{wp, First[Timing[First[x /. NDSolve[deqn, x, {t, 0, 100},
                       WorkingPrecision → wp, Method → "StiffnessSwitching"]]]]},
           {wp, Join[{MachinePrecision}, Range[20, 160, 10]]}];
        TableForm[times, TableHeadings → {{}, {"Precision", "Timing"}}]
```

Out[2]//TableForm=

Precision	Timing
MachinePrecision	0.0156001
20	0.124801
30	0.234002
40	0.312002
50	0.499203
60	0.670804
70	0.936006
80	1.27921
90	1.65361
100	2.19961
110	2.85482
120	3.52562
130	4.50843
140	5.38203
150	6.50524
160	7.98725

Setting aside for now numerical errors, there is another problem

C-Channel Region II Integrand 1: Two denominators contribute poles!



How to handle this?

- I’d need to split my region into a plethora of regions.
  - Region I: Quadruple-pole. Want this region to be 0.
  - Region II, III, IV, V: triple-poles coming from a double pole from either term and a simple pole in the other term, and the two coinciding. 4 possibilities, one for each term that doesn’t contribute to the triple pole
  - Regions VI-XI: Double pole regions. There are 4-choose-2 = 6 possibilities.
  - Region XII: Four simple poles
  - Regions XIII-XVI: Three simple poles, one region for each term that doesn’t contribute
  - Regions XVII-XXII: Two simple poles. Again 4-choose-2 = 6 possible combos.
  - Regions: XXIII-XXVI: One simple pole.
  - Region XXVII: No poles

Can I avoid this?

- This is significantly harder than the 1+1 case, where the poles were just functions of the kinematics and were just *numbers* from which you could pick values such that you could ensure 4 simple poles, for instance.
- I’ve sought kinematics, mass values that give me no plots with two or more terms from a denominator contributing, but no luck so far.
- I’d like to avoid this; would need to check the 6 possible double pole regions to ensure they’re empty (might not be easy), then integrate over the remaining 1+4+6+4+1 = 16 regions with the four simple pole region having 9 integrals. It would take a while to code this, then also check to make sure I didn’t make a mistake somewhere.

The difficulty of handling the double pole

A. Treating the case of the double pole

Note: I will seek kinematics that completely avoid double poles. However, I thought I would try to make some headway in regards to potentially treating the issue in the future.

In the case that $l_+^+ = l_-^+$, there is a double pole of the form

$$\lim_{\epsilon \to 0^+} \int_0^{q'^+} dl^+ \frac{f(l^+)}{(l^+ - l_+^+)^2 + i\epsilon} \tag{15}$$

Let $y = l^+ - l_+^+$ so $l^+ = y + l_+^+$ and $\mathrm{d}y = \mathrm{d}l^+$. Then this becomes

$$\begin{aligned}
\lim_{\epsilon\to 0^+} \int_{-l_+^+}^{q'^+ - l_+^+} dy \frac{f(y+l_+^+)}{y^2 + i\epsilon} &= \lim_{\epsilon\to 0^+} \int_{-l_+^+}^{q'^+ - l_+^+} dy \left(-i\pi \frac{\epsilon}{\pi\left(y^4+\epsilon^2\right)} f(y+l_+^+) + \frac{y^4}{y^4+\epsilon^2}\frac{f(y+l_+^+)}{y^2} \right) \\
&= \lim_{\epsilon'\to 0^+} \int_{-l_+^+}^{q'^+ - l_+^+} dy \left(-i\pi \frac{\epsilon'^2}{\pi\left(y^4+2y^2\epsilon'^2+\epsilon'^4\right)} f(y+l_+^+) + \frac{y^4}{y^4+\epsilon'^4}\frac{f(y+l_+^+)}{y^2} \right) \\
&= \lim_{\epsilon'\to 0^+} \int_{-l_+^+}^{q'^+ - l_+^+} dy \left(-\frac{\partial}{\partial y}\left[-i\pi \frac{\epsilon'}{\pi\left(y^2+\epsilon'^2\right)} \right] \frac{1}{2}\frac{\epsilon'}{y} \cdot f(y+l_+^+) + \frac{y^4}{y^4+\epsilon'^4}\frac{f(y+l_+^+)}{y^2} \right) \\
&= \lim_{\epsilon'\to 0^+} \int_{-l_+^+}^{q'^+ - l_+^+} dy \left(i\pi \frac{\partial}{\partial y}\left[\frac{\epsilon'}{\pi\left(y^2+\epsilon'^2\right)} \right] \left(\frac{1}{2}\epsilon' y^{\epsilon'-1}\right) \cdot f(y+l_+^+) + \frac{y^4}{y^4+\epsilon'^4}\frac{f(y+l_+^+)}{y^2} \right)
\end{aligned} \tag{16}$$

In the first step, I multiplied numerator and denominator by $y^2 - i\epsilon$, then I split it into two parts and multiplied the first term by π/π, and the second term by y^2/y^2. In the second step, I defined $\epsilon' = \sqrt{\epsilon}$ and since y is always a finite value the limit is the same as the original expression after adding $2y^2\epsilon'^2$ to the denominator. In the third step you can check that the derivative of the terms in square brackets, multiplied by $\epsilon'/2y$, is equal to the previous step's term. In the final step I effectively multiplied by $y^{\epsilon'}$ which again gives a term with the same limit as the original expression in the first line as $\epsilon' \to 0$ due to y being finite. Now we can use the identities

$$\delta(x) = \lim_{\epsilon\to 0} \frac{\epsilon}{\pi\left(x^2+\epsilon^2\right)} = \lim_{\epsilon\to 0}\left(\frac{1}{2}\epsilon\,|x|^{\epsilon-1}\right) \tag{17}$$

to simplify things. I also note that the term $\frac{y^4}{y^4+\epsilon^4}$ goes to 0 when $|y| \ll \epsilon$ and goes to 1 when $|y| \gg \epsilon$ and is symmetric about 0, so the second term will be a principal value integration. Putting it all together, I have

$$
\begin{aligned}
\lim_{\epsilon\to 0}\int_0^{q'^+} dl^+ \frac{f(l^+)}{\left(l^+ - l^+_+\right)^2 + i\epsilon} &= -i\pi \int_{-l^+_+}^{0} dy \frac{\partial\delta(y)}{\partial y}\delta(y) f(y + l^+_+) + i\pi \int_0^{q'^+ - l^+_+} dy \frac{\partial\delta(y)}{\partial y}\delta(y) f(y + l^+_+) \\
&+ \mathcal{P}\left(\int_{-l^+_+}^{q'^+ - l^+_+} dy \frac{f(y + l^+_+)}{y^2}\right)
\end{aligned} \tag{18}
$$

In the first integral I used the fact that I know y is negative for the whole range so I could send $y \to -|y|$ and substitute for $\delta(y)$, and for the second integral we similarly know y is positive so $y \to |y|$. As far as I'm aware, the validity of the expression holds so long as $f(y + l^+_+)$ is a continuous function on the real line. When I integrate by parts, for the first integral for instance, I get

$$
\begin{aligned}
\int_{-l^+_+}^{0} dy \frac{\partial\delta(y)}{\partial y}\delta(y) f(y + l^+_+) &= \delta(y)\delta(y) f(y + l^+_+)\Big|_{-l^+_+}^{0} - \int_{-l^+_+}^{0} dy\delta(y) \frac{\mathrm{d}}{\mathrm{d}y}\left[\delta(y) f(y + l^+_+)\right] \\
\Rightarrow \int_{-l^+_+}^{0} dy \frac{\partial\delta(y)}{\partial y}\delta(y) f(y + l^+_+) &= \frac{1}{2}\left[\delta(0)\delta(0) f(l^+_+) - \int_{-l^+_+}^{0} \delta(y)\delta(y) \frac{\partial f(y + l^+_+)}{\partial y}\mathrm{d}y\right]
\end{aligned} \tag{19}
$$

The first term is obviously *very* badly divergent. However, let's convert the delta functions back to the limiting procedure representation, and consider the following prescription:

$$\lim_{\epsilon\to 0}\int_0^{q'^+} dl^+ \frac{f(l^+)}{\left(l^+ - l^+_+\right)^2 + i\epsilon} = -i\pi \lim_{\xi\to 0}\left[\int_{-l^+_+}^{-\xi} dy \frac{\partial\delta(y)}{\partial y}\delta(y) f(y+l^+_+) - \int_{\xi}^{q'^+ - l^+_+} dy \frac{\partial\delta(y)}{\partial y}\delta(y) f(y+l^+_+)\right] + \mathcal{P}\left(\int_{-l^+_+}^{q'^+ - l^+_+} dy \frac{f(y+l^+_+)}{y^2}\right) \tag{20}$$

Once again integrating by parts the first two terms become

$$\begin{aligned} & - i\frac{\pi}{2}\lim_{\xi\to 0}\left[\delta(-\xi)\delta(-\xi) f(l^+_+) - \delta(\xi)\delta(\xi) f(l^+_+) - \left(\delta(-\xi)\frac{\partial f(l^+_+)}{\partial y} - \delta(\xi)\frac{\partial f(l^+_+)}{\partial y}\right)\right] \\ & = -i\frac{\pi}{2}\lim_{\substack{\xi\to 0\\ \epsilon\to 0}}\left\{\left(\frac{\epsilon}{\pi\left((-\xi)^2+\epsilon^2\right)}\right)^2 f(l^+_+) - \left(\frac{\epsilon}{\pi\left((\xi)^2+\epsilon^2\right)}\right)^2 f(l^+_+) - \left[\left(\frac{\epsilon}{\pi\left((-\xi)^2+\epsilon^2\right)}\right) - \left(\frac{\epsilon}{\pi\left((\xi)^2+\epsilon^2\right)}\right)\right]\frac{\partial f(l^+_+)}{\partial y}\right\} \\ & = 0 \end{aligned} \tag{21}$$

Everything cancels to zero! Another way of seeing the cancelation is by using

$$\delta(g(x)) = \sum_i \frac{\delta(x - x_i)}{|g'(x_i)|} \tag{22}$$

with $g(x) = \{-x, x\}$ we immediately have $\delta(x) = \delta(-x)$ and the cancellation can be seen directly from the first line.

Even if I did everything correctly…

- Principal value integrals at least in Mathematica can't be handled numerically if the pole diverges worse than 1/Sqrt[x^2+y^2+…]. I'm already having trouble with the principal value integrations with just one factor of "y" in the denominator.

Looking back at the big picture

- 36 integrals to evaluate: some of them have terms in the denominators that cause poles. For my current kinematics, this happens for 18 of them, meaning I need to check 6 terms, each with 3 denominators that may be causing the poles
- Let’s say I succeed in evaluating all these things. Then what?

Solve for the Form Factors!

$$J_S^\mu = F_1(q^2\Delta^\mu - q\cdot\Delta\, q^\mu) + F_2[(\bar{P}\cdot q + q^2)\Delta^\mu - q\cdot\Delta\,(\bar{P}^\mu + q^\mu)]$$

Left hand side: take, say, \mu = +, and separate into Re and Im parts. Both of these are numerically "measured" quantities
Right hand side: Expand F1 and F2 into their Re and Im parts; these are the 4 unknowns. The tensor parts are just numbers for a chosen kinematics.

Repeat for \mu = 1, 2, -. We now have 4 equations for 4 unknowns: Re[F1], Im[F1], Re[F2], Im[F2]

Repeat for other values of Q^2, x, t. Hopefully we find a kinematic regime where imaginary parts of form factors show up, but actually we'd like to extend this method to the fermion loop based off what we've learned from doing the boson loop!

Concluding remarks

- 1+1 dimension case has been evaluated by Dr. Ji and Yongwoo; however there is only 1 form factor there and no beam asymmetry
- Short term goal: finish this evaluation for a given Q^2
- Near term goal: Repeat calculations for a range of Q^2 (and x, t, ideally within a kinematics achievable by J-Lab)
- Long term goals: Extend method to fermion loop, work on the nucleon target, etc.