Numerical Hankel Transforms for TMDs

J.D. Terry

In collaboration with Z. Kang, N. Sato, A. Prokudin

Department of Physics and Astronomy
University of California, Los Angeles

QCD Evolution, 2018

1. Motivation
   - Why do we need Hankel Transforms in the CSS Formalism
   - Numerical Difficulties
2. What do we do?
   - Use Ogata's Method
   - Optimizing Ogata Parameters for TMDs
3. Comparison with other Numerical Methods
4. Preliminary Fits Using Ogata Algorithm

The SIDIS differential cross section is written in CSS formalisms as the following, Bacchetta et al (2006).

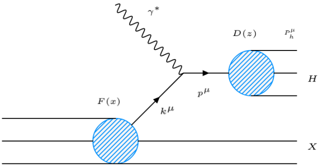
$$\frac{d\sigma}{dx\,dy\,d_z\,d^2P_{h,\perp}} = W + Y \equiv W + FO - ASY \qquad y = \frac{Q^2}{xS}$$

Defining $q_\perp = P_{h,\perp}/z$, the W terms describes the cross section in the region where $q_\perp^2/Q^2$. The FO term describes the cross section in the region where $m^2/q_\perp^2$. There is overlap in this region which is eliminated by the ASY term. Precision matching of W and Y requires precision definition of W.

$$\frac{W}{\sigma_0} \approx F_{UU}(x, z, Q, P_{h,\perp}) = \sum_q e_q^2 \int d^2k\,d^2p\,\delta^2(z\vec{k}_\perp + \vec{p}_\perp - \vec{P}_{h,\perp}) D_1(z, p_\perp^2) F_1(x, k_\perp^2)$$

$$\sigma_0 = \frac{2\pi\alpha_{em}^2}{Q^2}\,\frac{1+(1-y)^2}{y}$$

The single photon picture of SIDIS is given by



CSS formalism is carried out by Fourier transforming this to b space

$$\delta^2(z\vec{k}_\perp + \vec{p}_\perp - \vec{P}_{h,\perp}) = \frac{1}{z^2}\int \frac{d^2b}{(2\pi)^2} e^{-i(\vec{k}_\perp + \vec{p}_\perp/z - \vec{P}_{h,\perp}/z)\cdot\vec{b}}$$

$$F_{UU}^{SIDIS}(x, z, q_\perp) = \frac{1}{z^2}\int \frac{bdb}{2\pi} J_0(\frac{P_{h\perp}b}{z}) \sum_q e_q^2 D_1(z, b) F_1(x, b)$$

$$D_1(z, b) = C_{i\leftarrow q} \otimes D(z, \mu_{b_*}) e^{-\frac{1}{2}S_{pert} - S_{NP}^D}$$

$$F_1(x, b) = C_{q\leftarrow i} \otimes f_1^i(x, \mu_{b_*}) e^{-\frac{1}{2}S_{pert} - S_{NP}^F}$$

$$S_{pert} = \ln(\frac{Q^2}{\mu_{b_*}^2})\tilde{K}(b_*(b), \mu_{b_*}) + \int_{\mu_{b_*}}^{\mu_Q} \frac{d\mu'}{\mu}[2\gamma(\alpha_s(\mu')) - \ln\frac{Q^2}{\mu'^2}\gamma_K(\alpha_S(\mu'))]$$

$$S_{NP}^F = g_q b^2 + g_2/2\ln(\frac{b}{b_*})\ln(\frac{Q}{Q_0}) \qquad S_{NP}^D = g_h\frac{b^2}{z^2} + g_2/2\ln(\frac{b}{b_*})\ln(\frac{Q}{Q_0})$$

Formulas given by Kang, Prokudin, Sun, Yuan (2015).

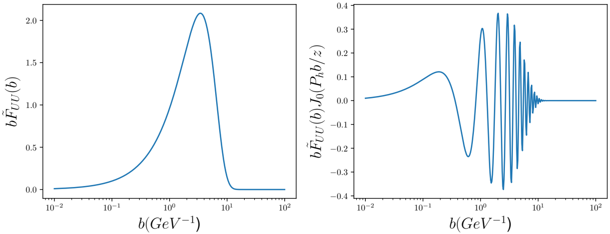
$$\tilde{F}_{UU}^{SIDIS}(b) = \sum_q e_q^2 (C_{i\leftarrow q} \otimes D(z, \mu_{b_*}))(C_{q\leftarrow i} \otimes f_1^i(x, \mu_{b_*}))e^{-S_{pert} - S_{NP}^D - S_{NP}^F}$$

$$F_{UU}^{SIDIS}(q_\perp) = \frac{1}{z^2} \int \frac{bdb}{2\pi} J_0(q_\perp b) \tilde{F}_{UU}(b)$$

$$b_* = \frac{b}{\sqrt{1 + \frac{b^2}{b_{max}^2}}} \qquad \mu_{b_*} = \frac{2e^{-\gamma_E}}{b_*}$$

What are the Numerical issues?

$$F_{UU}^{SIDIS} \approx \frac{1}{z^2} \int \frac{bdb}{2\pi} J_0(\frac{P_{h\perp} b}{z}) \tilde{F}_{UU}(b)$$



Current numerical methods are plagued by large contributions of positive and negative values which generate noise.

- Adaptive Quadrature Monte Carlo Integration (Vegas Monte Carlo)

JLab data sets will roughly 4,000-10,000 points. There will also be large SIDIS data sets with HERMES and COMPASS. These methods will negatively impact fits/matching for W/Y terms

Outline

Ogata established his quadrature formalism around the following approximation, Ogata (2005).

$$\int_0^\infty dx J_\nu(x) f(x) \approx \pi \sum_{k=1}^\infty w_{\nu k} \psi'(h\xi_k) J_\nu(x_k) f(x_k)$$

$$w_{\nu k} = \frac{2}{\pi^2 \xi_k J_{\nu+1}(\pi \xi_k)}, \qquad J_\nu(\pi \xi_k) = 0 \tag{1}$$

$$x_{\nu k} = \frac{\pi}{h} \psi(h\xi_k), \qquad \psi(z) = z \tanh\left(\frac{\pi}{2} \sinh\{z\}\right) \tag{2}$$

$$x_k = \pi \xi_k \tanh\left(\frac{\pi}{2} \sinh\{h\xi_{\nu k}\}\right)$$

Spacing between nodes goes like $1/h$. Ogata showed that the left hand side can be written as a contour integral with approaching the poles at the zeros of the bessel functions. This formula can be thought of as an expansion in residues of a contour integral. The double exponential die off of the terms eliminates the issue with the summing of large positive and negative values. This eliminates the noise in the inversion. The quadrature weights are defined in equation 1 along with the nodes in equation 2.

$$x_k = \pi\xi_k \tanh\left(\frac{\pi}{2}\sinh\{h\xi_k\}\right) \qquad J_\nu(\pi\xi_k) = 0$$

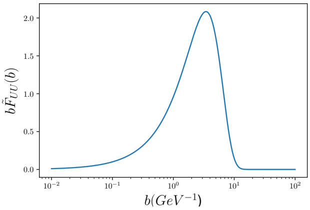
For a sufficiently large $h\xi_k$ the nodes approach the zeros of the bessel function. This analysis offers us two insights into the Ogata's quadrature method.

- For sufficiently large $h\xi_k$ we may introduce a cutoff to the sum.

$$\pi\sum_{k=1}^{\infty} w_{\nu k}\psi'(h\xi_k)J_\nu(x_k)f(x_k) \approx \pi\sum_{k=1}^{N} w_{\nu k}\psi'(h\xi_k)J_\nu(x_k)f(x_k)$$

- The rate at which the nodes approach the zeros of the bessel function is inversely related to the magnitude of the h parameter.
- The position of the nodes and therefore computational efficiency is controlled by the parameters h and N.

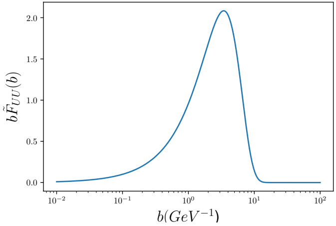
Since computation time scales with the number of function calls N, we will need to minimize N while maintaining convergence.



$$\frac{1}{z^2}\int \frac{bdb}{2\pi} J_0(bq_\perp)\tilde{F}_{UU}(b) \approx \frac{1}{2z^2q_\perp^2}\sum_{k=1}^{N} w_k \psi'(h\xi_k) J_\nu(x_k) x_k \tilde{F}_i(x_k)$$

Here $b > 10$ terms go to zero like a Gaussian due to non-perturbative effects. If we fix our parameters so that the nodes have converged to the zeros of the Bessel functions in this region as well, then the terms will go to zero even faster! We notice that sampling near the origin is weighted by $x_{\nu k}$ this means that sampling close to the origin will also give small contributions. Need to sample where $\tilde{F}_{UU}(b)$ is largest.

$$b^1 = \frac{\pi\xi_1}{q_\perp}\tanh\left(\frac{\pi}{2}\sinh\{h\xi_1\}\right) \qquad b^N = \frac{\pi\xi_N}{q_\perp}\tanh\left(\frac{\pi}{2}\sinh\{h\xi_N\}\right)$$



The largest contributions to the sum will come from the terms in the sum that are sampled near the peak of the this function. Parisi et al (1979), Collins et al (1983) showed that TMDs peak in b space near $1/Q$. The end points of the nodes in b space are

$$b^1 = \frac{\alpha_1}{Q} = \frac{\pi\xi_1}{q_\perp}\tanh\left(\frac{\pi}{2}\sinh\{h\xi_1\}\right) \qquad b^N = \frac{\alpha_N}{Q} = \frac{\pi\xi_N}{q_\perp}\tanh\left(\frac{\pi}{2}\sinh\{h\xi_N\}\right)$$

We enforce $\alpha_1 < 1$ and $1 < \alpha_N$. We can then invert this set of equations to solve for h and N.

We currently have three working algorithms.

- Ogata inversion with fixed α_1, α_N for all input functions. Inversion algorithm takes a function, ν and a value for $q_\perp$ as an input.
- An inversion method with an adaptive sampler for determining α_1 and α_N that is specific to the function. Inversion algorithm takes a function, ν, a value for $q_\perp$ as an input, and the maximum number of sampling points as an input.
- A full adaptive integration method. Inversion algorithm takes a function, ν, a value for $q_\perp$ as an input, a relative error ϵ_R as input.

First two methods give approximately the same numerical cost at 20 sampling points. The third method is much more costly but is the most accurate.

Let's take the parameterization (Gaussian)

$$F_{UU}(x,z,q_\perp) = \frac{1}{z^2}\sum_q e_q^2 D_1(z) F_1(x) \int \frac{bdb}{2\pi} J_0(q_\perp b) e^{-b^2/4\sigma^2}$$

The parameterization was used by Anselmino, Boglione, Gonzalez, Melis,and Prokudin (2013) to fit HERMES multiplicities for π/K production. One point was $z = 0.15$ $P_{h,\perp} = 1$ and $\sigma = 0.4$.

This has an analytic inversion given by

$$F_{UU}(x,z,P_{h,\perp},Q^2 = 1.80) = \sigma^2 \sum_q F_1(x) D_1(z) \frac{e^{-P_{h,\perp}^2 \sigma^2}}{\pi z^2}$$

We will use this to look at the error.

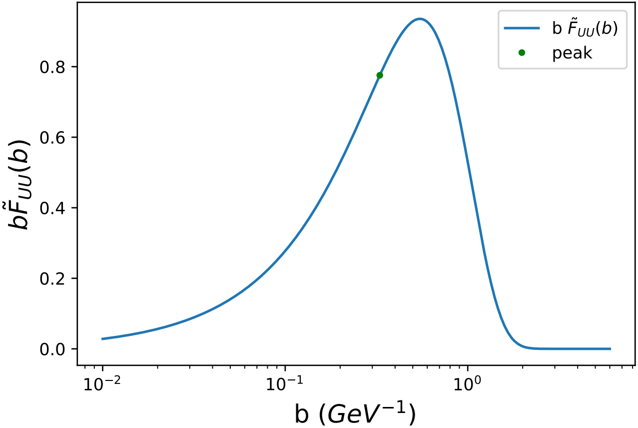


Figure: b-space function to be inverted

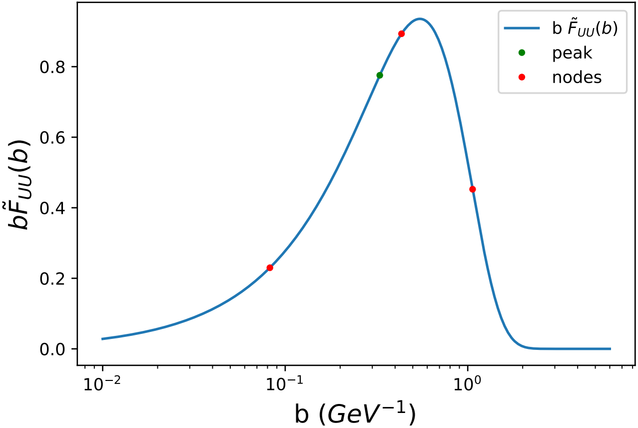


Figure: Initial sampling. Error is approximately 0.03

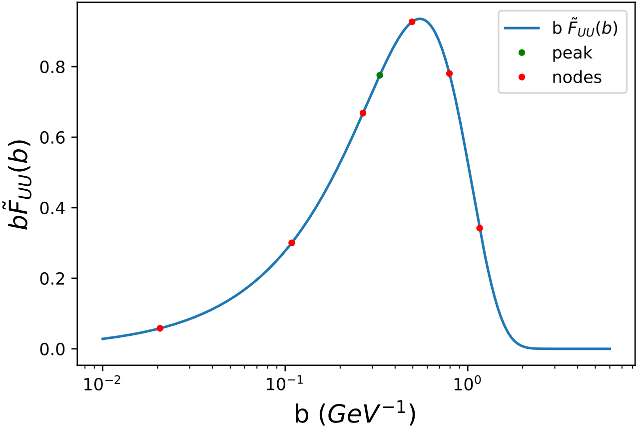


Figure: One iteration. Function decreased α_1. Error is approximately 10^{-5}

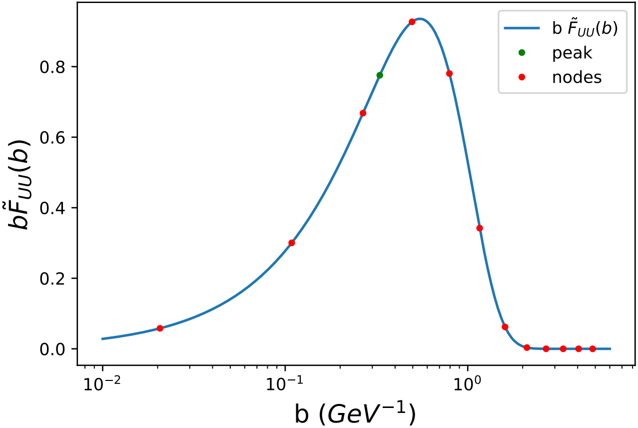


Figure: Two iterations. Function increased α_N. Error is approximately 10^{-6}

Outline

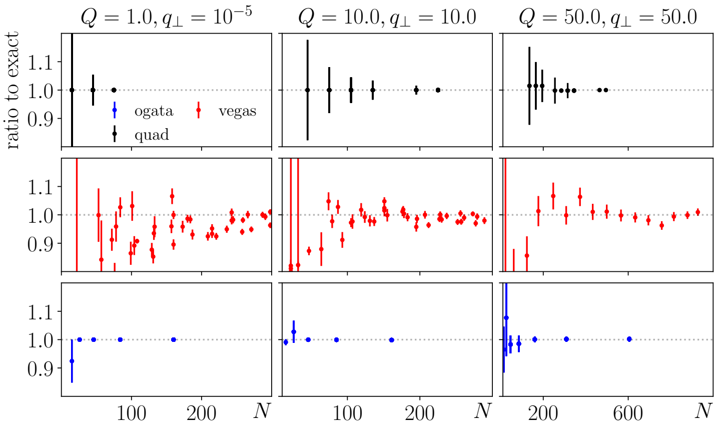


Figure: Error was estimated by doubling α_N and halfing α_1. The difference between this inversion and the previous one was used to estimate error.

$$F_{UU}^{SIDIS}(x, z, q_\perp, Q) \equiv \frac{x}{z^2} H(Q) \int \frac{bdb}{2\pi} J_0(\frac{P_{h\perp} b}{z}) \sum_q e_q^2 (C_D^q \otimes D^q(z))(C_F^q \otimes F^q(x)) e^{-S_{pert} - S_{NP}}$$

We compute the operator coefficient expansions and the hard coefficients order α_S and precalculate them.

We use

$$S_{pert} = \ln(\frac{Q^2}{\mu_{b_*}^2}) \tilde{K}(b_*(b), \mu_{b_*}) + \int_{\mu_{b_*}}^{\mu_Q} \frac{d\mu'}{\mu} [2\gamma(\alpha_s(\mu')) - \ln \frac{Q^2}{\mu'^2} \gamma_K(\alpha_S(\mu'))]$$

Where we use the perturbative Sudakovs a NLL.

$$S_{NP} = g_{q,i} b^2 + g_{h,i} \frac{b^2}{z^2} + g_2 \ln(\frac{b}{b_*}) \ln(\frac{Q}{Q_0})$$

Here the i represents whether or not the quark is valence or sea. Which we fit to have different widths.

We fit the parameters

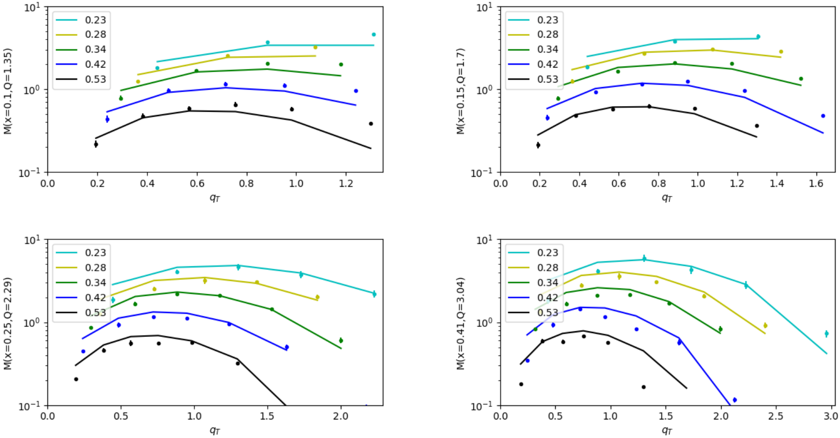
$$g_{q,val} \qquad g_{q,sea} \qquad g_{h,val} \qquad g_{h,sea} \quad g_2 \qquad b_{max} \qquad C_2 \qquad Q_0$$

Here $\mu_Q = C_2 Q$. We choose the cuts by Anselmino, Boglione, Gonzalez, Melis,and Prokudin (2013)

$$Q^2 > 1.69 \qquad z < 0.6 \qquad P_{h,\perp}/z < \frac{Q}{2}$$

and fit to Hermes multiplicities.

This is not a fit!



This is the fit!

```
JAM FITTER
count = 1213
elapsed time(mins)=70.786222
shifts  = 1
npts    = 64
chi2    = 159.082504
rchi2   = 0.000000
nchi2   = 0.000000
chi2tot = 159.082504
dchi2(iter)  = 0.000000
dchi2(local) = 0.000000

sidis
   1000     proton       pi+     hermes   M_Hermes    64      159.08        0.00        0.00
pdf         widths0 sea           2.76286e+00
pdf         widths0 valence       8.34776e-01
gk          C2                    2.92179e+00
gk          Q0                    1.00000e+01
gk          bmax                  1.02791e-01
gk          g2                    2.69802e-01
ff          widths0 k+ fav        2.35211e-01
ff          widths0 k+ unfav      1.37471e-01
ff          widths0 pi+ fav       2.24456e-01
ff          widths0 pi+ unfav     3.14590e-01
 Opening /u/home/j/jdterry/Projects/Hankel/fitpack/external/CJLIB/tbl_CJ15lo/CJ15lo_00.tbl
 Opening /u/home/j/jdterry/Projects/Hankel/fitpack/external/CJLIB/tbl_CJ15nlo/CJ15nlo_00.tbl
```

This gives a $\chi^2/dof \approx 2.5$. Fit ran in 70 mins. I did the same fit with adaptive quadrature and it took 48 hours. This method is roughly a factor of 50 times faster.

- Our Ogata algorithm avoids noise and performs efficient sampling.
- Remark: This has been tested for other $\nu \neq 0$ and showed similar efficiency.

- Future Work
  - Fit to Drell-Yan.
  - Application to $W + Y$ for SIDIS.

Thank you for your time!