# Efficient computation of Fourier-Bessel transforms for TMDs and other functions using the Levin's method

Oskar Grocholski

IRFU, CEA, Université Paris-Saclay


9-10 June 2026

# Motivation

TMD factorization: unpolarized structure functions expressed as

$$\tilde{W}(q) = \int \frac{d^2\mathbf{z}}{(2\pi)^2}\, e^{-i\mathbf{z}\cdot\mathbf{q}}\, W(z) = \int_0^\infty \frac{dz}{2\pi} J_0(qz)\, zW(z)\,. \tag{1}$$

$z = |\mathbf{z}|$.

$W(z)$ – product of TMDs and FFs.
Polarized structure functions $\implies$ also integrals involving $J_1, J_2$.

# Motivation

TMD factorization: unpolarized structure functions expressed as

$$\tilde{W}(q) = \int \frac{d^2\mathbf{z}}{(2\pi)^2}\, e^{-i\mathbf{z}\cdot\mathbf{q}}\, W(z) = \int_0^\infty \frac{dz}{2\pi} J_0(qz)\, zW(z)\,. \tag{1}$$

$z = |\mathbf{z}|$.

$W(z)$ – product of TMDs and FFs.
Polarized structure functions $\implies$ also integrals involving $J_1, J_2$.

Quadrature methods: grid-points depend on $q$ $\implies$ expensive if we consider many values of $q$ and a computationally-costly $W(z)$.

**Goal:** find a method that allows the use a fixed grid in a wide range of $q$.

# Levin's method: an overview

D. Levin, *Fast integration of rapidly oscillatory functions*,
J. of Computational and Applied Mathematics 67 (1996) 95-101

Rewrite the integral as

$$\int_{z_0}^{z_1} \vec{\omega} \cdot \vec{g}\, dz = \int_{z_0}^{z_1} \Big( \sum_i \omega_i g_i \Big)\, dz\,. \tag{2}$$

$\vec{\omega}$ – vector of oscillatory functions such that

$$\frac{d}{dz}\vec{\omega} = A^T \vec{\omega}, \tag{3}$$

$A^T$ – matrix of non-oscillatory functions.

# Levin's method: an overview

One can choose

$$\vec{\omega}(z) = \begin{bmatrix} J_\nu(qz) \\ \\ J_{\nu+1}(qz) \end{bmatrix}, \quad \vec{g}(z) = \begin{bmatrix} f(z) \\ \\ 0 \end{bmatrix}. \tag{4}$$

In that case:

$$A^T = \begin{bmatrix} \nu/z & q \\ \\ -q & -(\nu+1)/z \end{bmatrix}. \tag{5}$$

Side remark: integral with $J_{\nu+1} \rightarrow$ just use $\vec{g} = \big[0,\ f(z)\big]^T$.

# Levin's method: an overview

Find a function $\vec{h}(z)$ such that

$$\Big(\frac{d}{dz} + A\Big)\vec{h} = \vec{g}, \tag{6}$$

then

$$\frac{d}{dz}(\vec{h}\cdot\vec{\omega}) = (\frac{d}{dz}\vec{h})\cdot\vec{\omega} + \vec{h}\cdot\big(A^T\vec{\omega}\big) = \vec{g}\cdot\vec{\omega}. \tag{7}$$

# Levin's method: an overview

Find a function $\vec{h}(z)$ such that

$$\Big(\frac{d}{dz} + A\Big)\vec{h} = \vec{g}, \tag{6}$$

then

$$\frac{d}{dz}(\vec{h}\cdot\vec{\omega}) = (\frac{d}{dz}\vec{h})\cdot\vec{\omega} + \vec{h}\cdot\big(A^T\vec{\omega}\big) = \vec{g}\cdot\vec{\omega}. \tag{7}$$

The integral can be easily computed:

$$\int_{z_0}^{z_1} \vec{\omega}\cdot\vec{g}\,dz \;=\; \vec{h}(z)\cdot\vec{\omega}(z)\Big|_{z_0}^{z_1}. \tag{8}$$

# Levin's method: application to the Hankel transform

Obtained system of differential equations:

$$\left(\frac{d}{dz} + \begin{bmatrix} \nu/z & -q \\ q & -(\nu+1)/z \end{bmatrix}\right) \begin{bmatrix} h_1(z) \\ h_2(z) \end{bmatrix} = \begin{bmatrix} f(z) \\ 0 \end{bmatrix}. \tag{9}$$

✗ Singularity at $z = 0$ !

# Levin's method: singularity at $z = 0$

Make a different splitting of the oscillatory and non-oscillatory part of the integral:

$$\int_0^\infty J_\nu(qz) f(z)\, dz = \int_0^\infty \Big(z^{-\nu} J_\nu(qz)\Big) z^\nu f(z)\, dz. \tag{10}$$

# Levin's method: singularity at $z = 0$

Make a different splitting of the oscillatory and non-oscillatory part of the integral:

$$\int_0^\infty J_\nu(qz) f(z)\, dz = \int_0^\infty \Big(z^{-\nu} J_\nu(qz)\Big) z^\nu f(z)\, dz. \tag{10}$$

Use the rescaled vector of oscillatory functions:

$$\vec{\omega}_2 = \begin{bmatrix} z^{-\nu} J_\nu\big(qz\big) \\ \\ z^{-\nu} J_{\nu+1}\big(qz\big) \end{bmatrix}, \qquad \lim_{z\to 0} \vec{\omega}_2(z) \text{ is finite.} \tag{11}$$

# Levin's method: singularity at $z = 0$

Make a different splitting of the oscillatory and non-oscillatory part of the integral:

$$\int_0^\infty J_\nu(qz) f(z)\, dz = \int_0^\infty \Big(z^{-\nu} J_\nu(qz)\Big) z^\nu f(z)\, dz. \tag{10}$$

Use the rescaled vector of oscillatory functions:

$$\vec{\omega}_2 = \begin{bmatrix} z^{-\nu} J_\nu\big(qz\big) \\ \\ z^{-\nu} J_{\nu+1}\big(qz\big) \end{bmatrix}, \qquad \lim_{z\to 0} \vec{\omega}_2(z) \text{ is finite.} \tag{11}$$

The resulting matrix $A_2$ is

$$A_2 = \begin{bmatrix} 0 & -q \\ \\ q & -(2\nu+1)/z \end{bmatrix}. \tag{12}$$

# Levin's method: singularity at $z = 0$

Introduce $z\tilde{h}_3 = \tilde{h}_2$, to remove the $1/z$ factor.

$\nu \geq 1 \implies \tilde{h}_3$ is well-behaved at $z = 0$.

# Levin's method: singularity at $z = 0$

Introduce $z\tilde{h}_3 = \tilde{h}_2$, to remove the $1/z$ factor.

$\nu \geq 1 \implies \tilde{h}_3$ is well-behaved at $z = 0$.

Differential equations for the rescaled functions:

$$\begin{cases} z^\nu f(z) &= \frac{d}{dz}\tilde{h}_1(z) - qz\tilde{h}_3, \\ 0 &= z\frac{d}{dz}\tilde{h}_3 - 2\nu\tilde{h}_3 + q\tilde{h}_1. \end{cases} \tag{13}$$

**Remark:** for larger $z$, it is better to use $\frac{z}{z+1}\tilde{h}_3 = \tilde{h}_2$, and scale the oscillatory part $\vec{\omega}_2$ by $\big(z/(1+z)\big)^{-\nu}$ rather than $z^{-\nu}$.
$\implies$ longer formulas, but the method is the same.

# The case $0 \leq \nu < 1$

$0 \leq \nu \leq 1 \rightarrow \frac{d}{dz}\tilde{h}_3(z=0)$ can become infinite!

Integrate by parts:

$$\int_{z_0}^{z_1} J_\nu(qz) f(z)\, dz = \frac{1}{q} J_{\nu+1}(qz) f(z)\Big|_{z_0}^{z_1} - \frac{1}{q}\int_{z_0}^{z_1} z^\nu \Big(\frac{d}{dz}\big(zf(z)\big) - (\nu+2)f(z)\Big) z^{-(\nu+1)} J_{\nu+1}(qz). \qquad (14)$$

# Infinite interval

In general, one can make a variable transformation:

$$\int_0^\infty f(z) J_\nu(qz)\, dz = \int_{-1}^{1} \Big(\frac{dz}{du}\Big) f\big(z(u)\big) J_\nu\big(z(u)\big)\, du. \tag{15}$$

The resulting equation for $\tilde{h}(u)$ reads:

$$\begin{bmatrix} f\big(z(u)\big) \\ \\ 0 \end{bmatrix} = \left( \Big(\frac{du}{dz}\Big) \frac{d}{du} + A \right) \begin{bmatrix} \tilde{h}_1(u) \\ \\ \tilde{h}_2(u) \end{bmatrix} \tag{16}$$

# Chebyshev pseudospectral method

$p_1, p_3$ - Chebyshev polynomials of order $N-1$ approximating $\tilde{h}_{1,3}$.

$$u_j = \cos\left(\frac{j\pi}{N}\right), \quad j \in \{0, ..., N-1\} \qquad \text{interpolation points,} \tag{17}$$

$$p_{1,3}(u_j) = \tilde{h}_{1,3}(u_j), \qquad f(z_j) = f\big(z(u_j)\big), \tag{18}$$

$$\frac{d}{du}p(u_j) = \sum_{k=0}^{N-1} D_{jk}p(u_k) \approx \frac{d}{du}\tilde{h}(u_j). \tag{19}$$

$D$ - Chebyshev differentiation matrix.

# Chebyshev pseudospectral method

Find the approximate solution by discretizing the system:

Let $z_j = z(u_j)$,

$$\begin{cases} z_j^{\nu} f(z_j) &= \left(\frac{du}{dz}\right) \sum_{jk} D_{jk} p_1(u_k) - q z_j p_3(u_j), \\ 0 &= z_j \left(\frac{du}{dz}\right) \sum_{jk} D_{jk} p_3(u_k) - 2\nu p_3(u_j) + q p_1(u_j). \end{cases} \tag{20}$$

# Chebyshev pseudospectral method

Find the approximate solution by discretizing the system:

Let $z_j = z(u_j)$,

$$\begin{cases} z_j^\nu f(z_j) &= \left(\frac{du}{dz}\right) \sum_{jk} D_{jk} p_1(u_k) - q z_j p_3(u_j), \\ 0 &= z_j \left(\frac{du}{dz}\right) \sum_{jk} D_{jk} p_3(u_k) - 2\nu p_3(u_j) + q p_1(u_j). \end{cases} \tag{20}$$

Let:

$$\vec{F} = \left[z_0^\nu f(z_0), ..., z_{N-1}^\nu f(z_{N-1}), 0, ...., 0\right]^T, \tag{21}$$

$$\vec{P} = \left[p_1(u_0), ..., p_1(u_{N-1}), p_3(u_0), ..., p_3(u_{N-1})\right]^T, \tag{22}$$

$$B\vec{P} = \vec{F}. \tag{23}$$

# Chebyshev pseudospectral method

$\longrightarrow 2N$ equations.

$\longrightarrow$ Need to find an appropriate variable transformation (so that one can interpolate the integrand and the solution well on the resulting grid).

M. Diehl, R. Nagar, F. J. Tackmann, *ChiliPDF: Chebyshev Interpolation for Parton Distributions*, Eur. Phys. J. C 82, 257 (2022) – efficient use of Chebyshev grids, many kinds of variable transformation implemented.

# Technical remarks

For $\sim 50$ nodes the best accuracy is obtained when splitting the integral into 2 parts:

$$\int_0^\infty J_\nu(qz) f(z)\, dz = \int_0^{z_1} J_\nu(qz) f(z)\, dz + \int_{z_1}^\infty J_\nu(qz) f(z)\, dz. \tag{24}$$

# Technical remarks

For $\sim 50$ nodes the best accuracy is obtained when splitting the integral into 2 parts:

$$\int_0^\infty J_\nu(qz) f(z)\, dz = \int_0^{z_1} J_\nu(qz) f(z)\, dz + \int_{z_1}^\infty J_\nu(qz) f(z)\, dz. \tag{24}$$

- $qz_1$ smaller than the first zero of $J_\nu \implies$ integrate on the first subinterval using the Clenshaw-Curtis quadrature.

# Technical remarks

For $\sim 50$ nodes the best accuracy is obtained when splitting the integral into 2 parts:

$$\int_0^\infty J_\nu(qz) f(z)\, dz = \int_0^{z_1} J_\nu(qz) f(z)\, dz + \int_{z_1}^\infty J_\nu(qz) f(z)\, dz. \tag{24}$$

- $qz_1$ smaller than the first zero of $J_\nu \implies$ integrate on the first subinterval using the Clenshaw-Curtis quadrature.
- larger $qz_1 \implies$ construct the matrix $B$ needed to solve (20).

# Technical remarks

For $\sim 50$ nodes the best accuracy is obtained when splitting the integral into 2 parts:

$$\int_0^\infty J_\nu(qz) f(z)\, dz = \int_0^{z_1} J_\nu(qz) f(z)\, dz + \int_{z_1}^\infty J_\nu(qz) f(z)\, dz. \tag{24}$$

- $qz_1$ smaller than the first zero of $J_\nu \implies$ integrate on the first subinterval using the Clenshaw-Curtis quadrature.
- larger $qz_1 \implies$ construct the matrix $B$ needed to solve (20).
  - First, compute the LU decomposition of $B$.

# Technical remarks

For $\sim 50$ nodes the best accuracy is obtained when splitting the integral into 2 parts:

$$\int_0^\infty J_\nu(qz) f(z)\, dz = \int_0^{z_1} J_\nu(qz) f(z)\, dz + \int_{z_1}^\infty J_\nu(qz) f(z)\, dz. \tag{24}$$

- $qz_1$ smaller than the first zero of $J_\nu \implies$ integrate on the first subinterval using the Clenshaw-Curtis quadrature.
- larger $qz_1 \implies$ construct the matrix $B$ needed to solve (20).
  - First, compute the LU decomposition of $B$.
  - If $B$ is ill-conditioned (zero or small elements on diagonal), use the singular value decomposition (SVD) of $B$.

# Variable transformation: Gaussian behavior

For Gaussian behavior of the integrand it is better to use the exponential variable transformation $u \sim \exp\left(-\text{const.} \times z\right)$ rather than the Gaussian:

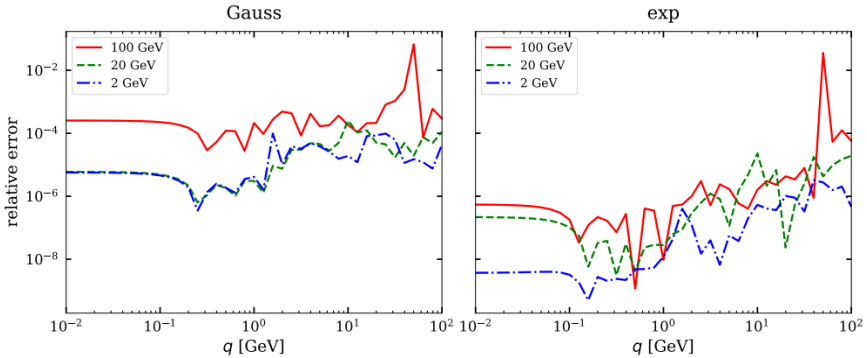



Figure: Integration precision for $W(z)$ with the Gaussian behavior of TMD with Gaussian (left) and exponential (right) variable transformation.

# Variable transformation: exponential behavior

Analogously, for exponential vanishing of the integrand, a variable transformation scaling like $\exp\left(-\text{const.} \times \sqrt{z}\right)$ gives better precision:

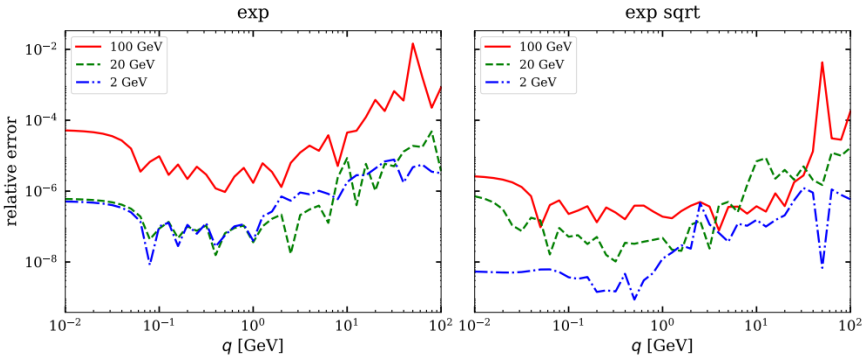



Figure: Integration precision for $W(z)$ with the exponential behavior of TMD with exponential (left) and square-root-exponential (right) variable transformation.

# 1 vs 2 vs 3 subgrids

A single grid performs poorly (worse precision + slower computation):

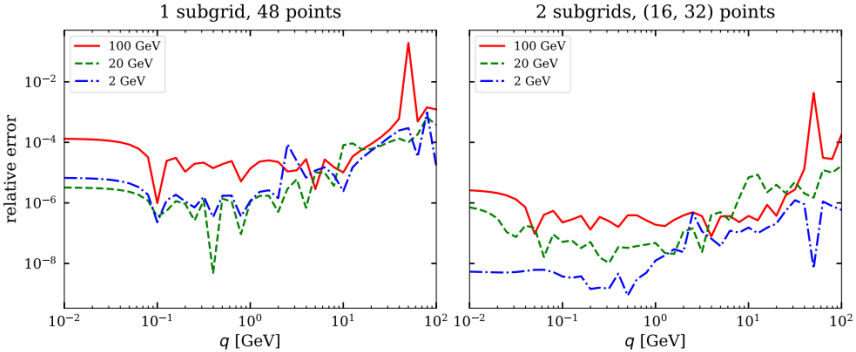



Figure: Integration precision for $W(z)$ with the exponential behavior of TMD for 1 (left) and 2 (right) subgrids.

# 1 vs 2 vs 3 subgrids

3 subgrids are faster than 2 (with the same total number of points) but less precise:

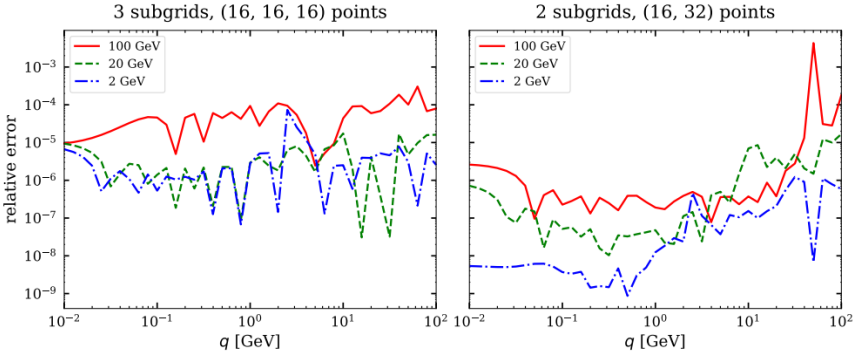



Figure: Integration precision for $W(z)$ with the exponential behavior of TMD for 3 (left) and 2 (right) subgrids.

# Variable transformation and grid settings: summary

- Gaussian vanishing: $W(z) \sim \exp\big(-\lambda^2 z^2\big)$
  $\rightarrow$ exponential transformation $u(z) = -\exp\big(-(r\lambda)z/4\big)$.

# Variable transformation and grid settings: summary

- Gaussian vanishing: $W(z) \sim \exp\big(-\lambda^2 z^2\big)$
  $\rightarrow$ exponential transformation $u(z) = -\exp\big(-(r\lambda)z/4\big)$.
- Exponential behavior: $W(z) \sim \exp\big(-\kappa z\big)$
  $\rightarrow u(z) = -\exp\Big(1 - \sqrt{1 + (r\kappa)z/2}\Big)$.

# Variable transformation and grid settings: summary

- Gaussian vanishing: $W(z) \sim \exp\big(-\lambda^2 z^2\big)$
  $\rightarrow$ exponential transformation $u(z) = -\exp\big(-(r\lambda)z/4\big)$.
- Exponential behavior: $W(z) \sim \exp\big(-\kappa z\big)$
  $\rightarrow u(z) = -\exp\Big(1 - \sqrt{1 + (r\kappa)z/2}\Big)$.
- In both cases, a good precision can be expected for $r \in [2, 6]$.

# Variable transformation and grid settings: summary

- Gaussian vanishing: $W(z) \sim \exp\big(-\lambda^2 z^2\big)$
  $\rightarrow$ exponential transformation $u(z) = -\exp\big(-(r\lambda)z/4\big)$.
- Exponential behavior: $W(z) \sim \exp\big(-\kappa z\big)$
  $\rightarrow u(z) = -\exp\Big(1 - \sqrt{1 + (r\kappa)z/2}\Big)$.
- In both cases, a good precision can be expected for $r \in [2, 6]$.
- Division into 2 subgrids: $[0,\ z_1,\ \infty]$ performs significantly better than 1 or 3 subgrids.

# Variable transformation and grid settings: summary

- Gaussian vanishing: $W(z) \sim \exp\big(-\lambda^2 z^2\big)$
  $\rightarrow$ exponential transformation $u(z) = -\exp\big(-(r\lambda)z/4\big)$.
- Exponential behavior: $W(z) \sim \exp\big(-\kappa z\big)$
  $\rightarrow u(z) = -\exp\Big(1 - \sqrt{1 + (r\kappa)z/2}\Big)$.
- In both cases, a good precision can be expected for $r \in [2, 6]$.
- Division into 2 subgrids: $[0,\ z_1,\ \infty]$ performs significantly better than 1 or 3 subgrids.
- For the tested functions W(z) with $Q \in \{2,\ 20,\ 100\ \text{GeV}\}$, in a wide range $q \in [10^{-2}, 10^{2}\ \text{GeV}]$, good precision is obtained for $z_1 \in [0.05,\ 0.15\ \text{GeV}]$.

# Variable transformation and grid settings: summary

- Gaussian vanishing: $W(z) \sim \exp\big(-\lambda^2 z^2\big)$
  $\rightarrow$ exponential transformation $u(z) = -\exp\big(-(r\lambda)z/4\big)$.
- Exponential behavior: $W(z) \sim \exp\big(-\kappa z\big)$
  $\rightarrow u(z) = -\exp\Big(1 - \sqrt{1 + (r\kappa)z/2}\Big)$.
- In both cases, a good precision can be expected for $r \in [2, 6]$.
- Division into 2 subgrids: $[0,\ z_1,\ \infty]$ performs significantly better than 1 or 3 subgrids.
- For the tested functions W(z) with $Q \in \{2,\ 20,\ 100\ \text{GeV}\}$, in a wide range $q \in [10^{-2}, 10^{2}\ \text{GeV}]$, good precision is obtained for $z_1 \in [0.05,\ 0.15\ \text{GeV}]$.

For a broader overview, see Section 4 and 5 in EPJC 84, 858 (2024).

# Benchmarking the precision

Compare with precision obtained using optimised Ogata quadrature:
Z.B. Kang, A. Prokudin, N. Sato, J. Terry, *Efficient Fourier Transforms for Transverse Momentum Dependent Distributions*, Computer Physics Communications 258 (2021) 107611.

# Benchmarking the precision

Compare with precision obtained using optimised Ogata quadrature:
Z.B. Kang, A. Prokudin, N. Sato, J. Terry, *Efficient Fourier Transforms for Transverse Momentum Dependent Distributions*, Computer Physics Communications 258 (2021) 107611.

$W(z)$ including LO evolution effects at low $z$ and different Ansätze for large $z$ behavior:

$$W(z) = \exp\Big( -2S(\mu_z, Q)\Big) \times \big[f_{\mathsf{np}}(z)\big]^2 \tag{25}$$

- $\exp\big( -2S(\mu_z, Q)\big)$ – LO evolution with a $z$-dependent renormalization scale $\mu_z \propto 1/z^*$.
- $f_{\mathsf{np}}(z)$ – Ansatz for large-$z$ behavior of TMDs,
- $Q$ – hard scale.

# Benchmarking the precision

Gaussian behavior from Bacchetta et al., 10.1007/JHEP06(2017)081

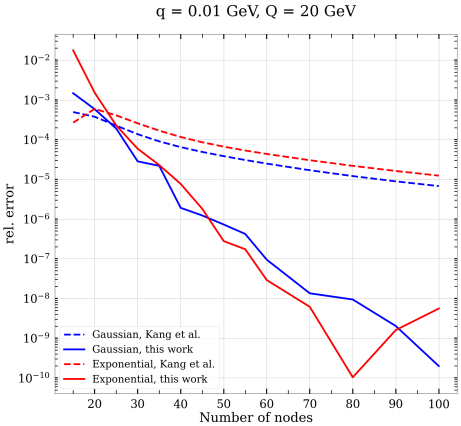
- $\left[f_{\text{np}}(z)\right]^2 \sim \exp\left(-\lambda^2 z^2\right)$, $\lambda = 0.374$ GeV,
- Variable transformation: $u(z) = -\exp\left(-mz/4\right)$, where $m/\lambda = 5$.

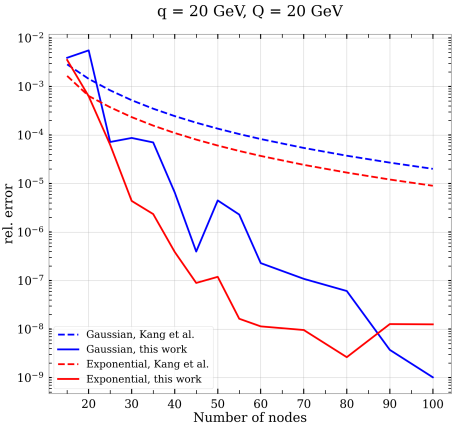
Exponential form from Scimemi and Vladimirov, 10.1140/epjc/s10052-018-5557-y

- $\left[f_{\text{np}}(z)\right]^2 \sim \exp\left(-\kappa z\right)$, $\kappa = 0.642$ GeV.
- Variable transformation: $u(z) = -\exp\left(1 - \sqrt{1 + mz/2}\right)$, where $m/\kappa = 3$

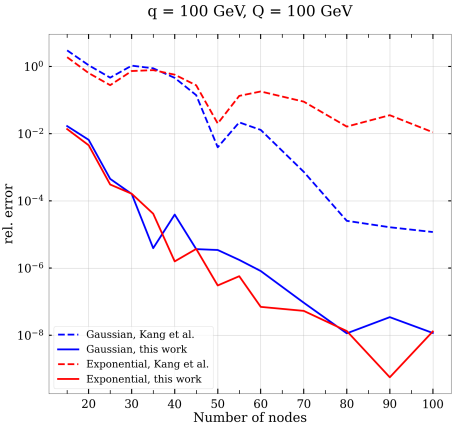
In both cases we use the grid splitting: $[0,\ 0.05\,\text{GeV},\ \infty]$.

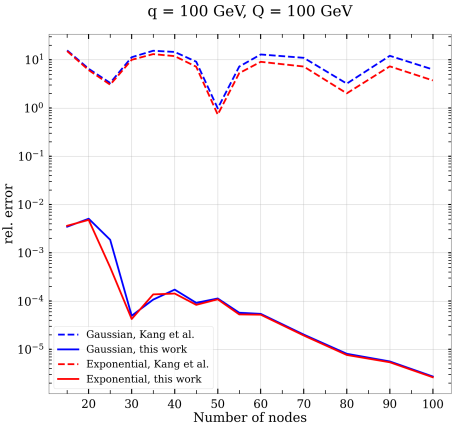
# Benchmarking the precision



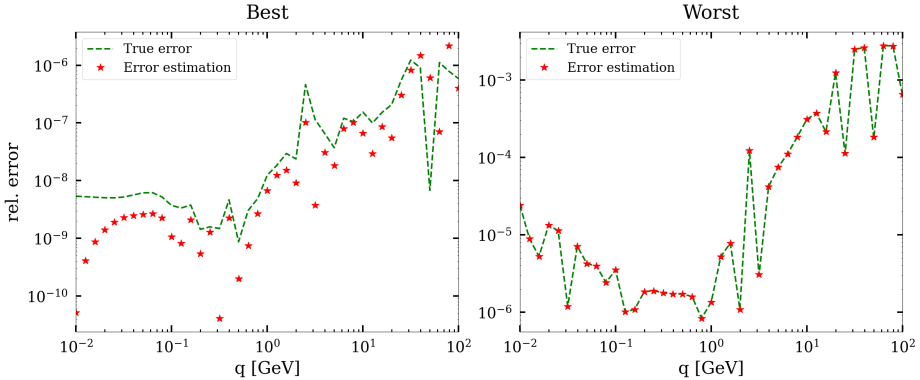
# Benchmarking the precision



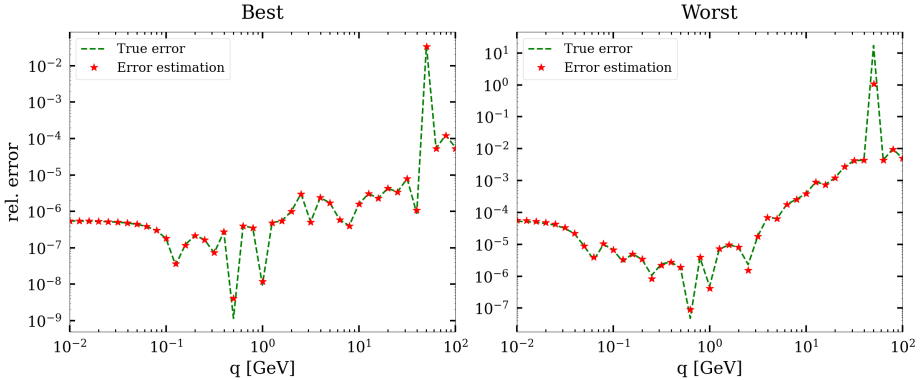
# Error estimation

Estimate the error by comparing with results for grid with twice as many points.



# Error estimation

Estimate the error by comparing with results for grid with twice as many points.



# Integrals defined in the distribution sense

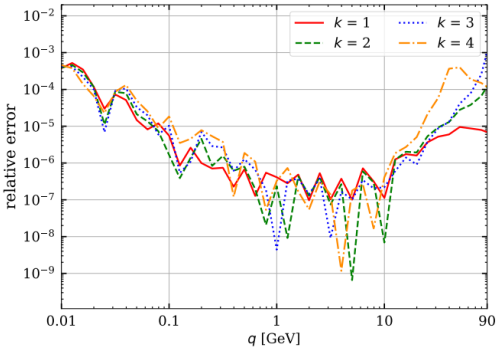
In some cases (see [arXiv:2306.16458] one considers $W(z)$ that does not vanish at $z \to \infty$, e.g.

$$\int_0^\infty dz\, J_\nu(qz)\, z\, \ln^k\left(Qz\right). \tag{26}$$

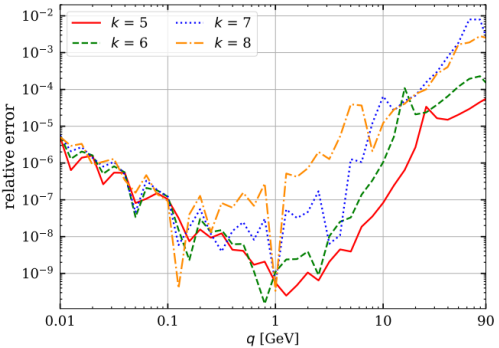
$\to$ Can be defined in the sense of a distribution!

$\to$ A slight modification of the method allows one to compute the finite part of the resulting distribution at $q \neq 0$ (see Section 6 in EPJC 84, 858 (2024)).

# Integrals defined in the distribution sense

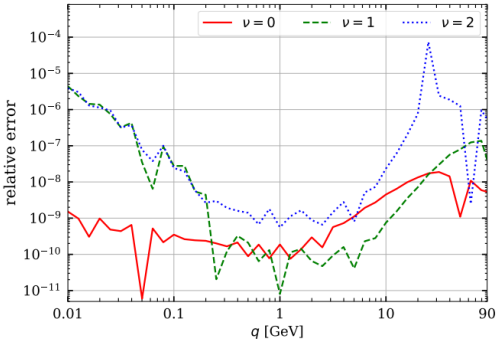



(a) $\int dz\, J_0(qz)\, z\, \ln^k w(z), \quad k = 1, 2, 3, 4$

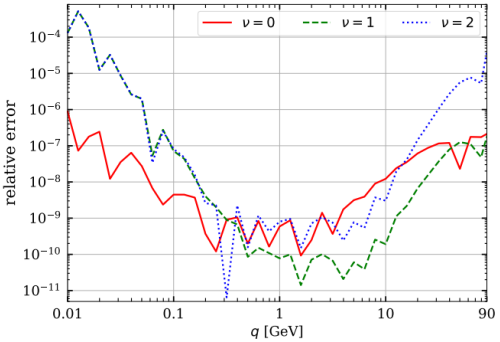



(b) $\int dz\, J_0(qz)\, z\, \ln^k w(z), \quad k = 5, 6, 7, 8$

# Integrals defined in the distribution sense




(c) $\int dz\, J_\nu(qz)\, z\, \ln w(z), \quad \nu = 0, 1, 2$




(d) $\int dz\, J_\nu(qz)\, z\, \ln^2 w(z), \quad \nu = 0, 1, 2$

# Summary

- Levin's method allows to compute the Bessel-Fourier transform using a $q$-independent grid in a broad range of $q$.
- Better precision than the quadrature methods, especially at larger $q$.
- Need to provide the grid parameters manually (subgrid splitting, variable transformation), but the good settings are known and provided in the paper: M. Diehl, OG, *Efficient computation of Fourier–Bessel transforms for transverse-momentum dependent parton distributions and other functions*, EPJC 84, 858 (2024).
- BestLime: a `C++` library implementing the discussed method is available at https://zenodo.org/records/13844084.

# Backup: LU decomposition of the matrix B

$$PB = LU \tag{27}$$

$$L = \begin{pmatrix} 1 & 0 & ... & 0 \\ l_{1,2} & 1 & ... & 0 \\ & & \cdots & \\ l_{1,n-1} & l_{2,n-1} & \ddots & 0 \\ l_{1,n} & ... & l_{n-1,n} & 1 \end{pmatrix} \quad U = \begin{pmatrix} u_{1,1} & u_{2,1} & ... & u_{n,1} \\ 0 & u_{2,2} & ... & u_{n,2} \\ & \cdots & & \\ 0 & ... & \ddots & u_{n,n-1} \\ 0 & ... & 0 & u_{n,n} \end{pmatrix} \tag{28}$$

$P$ - permutation matrix.

Can solve $B\vec{h} = \vec{g}$ using the backward substitution method.

# Backup: SV decomposition of the matrix B

$$B = U\big[\mathrm{diag}(w_j)\big]V^T, \tag{29}$$

$U, V$ - orthogonal matrices, $w_j$ - singular values.

$$B^{-1} = V\big[\mathrm{diag}(1/w_j)\big]U^T. \tag{30}$$

$w_j = 0$ (or $|w_j| < \varepsilon$ - arbitrarily chosen small value) $\rightarrow$ replace $1/w_j$ by 0. Solution $h'$ obtained this way minimizes the error

$$\sum_j \left| \left(B\vec{h}' - \vec{g}\right)_j \right|. \tag{31}$$