Covariant extension of DGLAP GPDs to the ERBL region: the inverse Radon transform

Jose Manuel Morgado Chávez[1]

Progress in algorithms and numerical tools for QCD
IJCLab. Orsay, France.
7-8th June 2022.

Email: josemanuel.morgado@dci.uhu.es

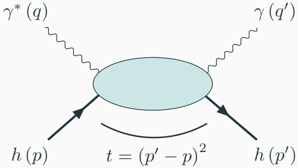
FÍSICA
MATEMÁTICAS
COMPUTACIÓN

Universidad de Huelva

Introduction: The physical problem

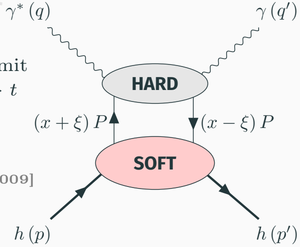
Hadron structure

How do quarks and gluons combine to make hadrons up?



Generalized Bjorken limit
$Q^2 \to \infty$ with $Q^2 >> t$
and ξ fixed.

Factorization
[Phys.Rev.D59(1999)074009]



$$\mathcal{M}\left(\xi, t; Q^2\right) = \sum_{p=q,g} \int_{-1}^{1} \frac{dx}{\xi} \mathcal{K}^p \left(\frac{x}{\xi}, \frac{Q^2}{\mu_F^2}, \alpha_s\left(\mu_F^2\right) \right) F^p\left(x, \xi, t; \mu_F^2\right)$$

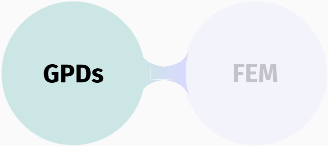
Hard kernel, $\mathcal{K}^p$: perturbative information

Generalized Parton distributions F^p: non perturbative QCD

Generalized parton distributions:

Overview

Generalized parton distributions



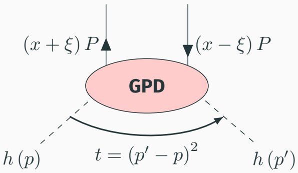
(GPD) – Generalized parton distributions:
Non-local quark and gluon operators, evaluated between hadron states in non-forward kinematics and projected onto the light front.

[Fortsch.Phys.:42(1994)101]
[Phys.Lett.B:380(1996)417]
[Phys.Rev.D:55(1197)7114]

Example: Twist-two chiral-even quark GPD of a spinless hadron.

$$H^q\left(x,\xi,t\right)=\frac{1}{2}\int\frac{d\lambda}{2\pi}e^{i\lambda x}\left\langle h\left(p'\right)\right|\psi^q\left(-\lambda n/2\right)\not{n}\psi^q\left(\lambda n/2\right)\left|h\left(p\right)\right\rangle$$

Generalized parton distributions



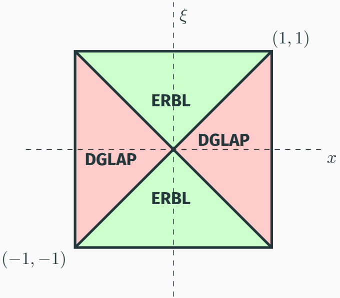
x: Momentum fraction of P.

ξ: Fraction of momentum longitudinally tranfered.

t: Momentum transfer.

Kinematics:
[Phys.Rept:388(2003)41]

- **DGLAP** ($|x| > |\xi|$): Emits/takes a quark ($x > 0$) or antiquark ($x < 0$).
- **ERBL:** ($|x| < |\xi|$): Emits pair quark-antiquark.



Generalized parton distributions

- **Support:** [Phys.Lett.B:428(1998)359]

$$(x, \xi) \in [-1,1] \otimes [-1,1]$$

- **Positivity:** [Phys.Rev.D:65(2002)114015, Eur.Phys.J.C:8(1999)103]

$$|H^q(x,\xi,t=0)| \leq \sqrt{q\left(\frac{x+\xi}{1+\xi}\right) q\left(\frac{x-\xi}{1-\xi}\right)} \quad , \quad |x| \geq \xi$$

Hilbert space norm

- **Polynomiality:** Order-m Mellin moments are degree-$(m+1)$ polynomials in ξ. [J.Phys.G: 24(1998)1181, Phys.Lett.B:449(1999)81]

$$\int_{-1}^{1} dx x^m H^q(x,\xi,t) = \sum_{\substack{k=0 \\ k \text{ even}}}^{m+1} c_k^{(m)}(t)\, \xi^k$$

Lorentz invariance

GPD modeling: general strategy

1. **Overlap representation**
   [Nucl.Phys.B:596(2001)33]

   Based on LFWFs, $\Psi^q\left(x, k_\perp^2\right)$

   } Polynomiality ? Positivity ✓

2. **Double Distribution representation**
   [Fortsch.Phys.:42(1994)101, JLAB-THY-00-33]

   Relying on Radon transform, $\mathcal{R}$

   } Polynomiality ✓ Positivity ?

Different modeling strategies and **different problems**

GPD modeling: general strategy

1. **Overlap representation**
   [Nucl.Phys.B:596(2001)33]

   Based on LFWFs, $\Psi^q\left(x, k_\perp^2\right)$

   } Polynomiality ? Positivity ✓

---

2. **Double Distribution representation**
   [Fortsch.Phys.:42(1994)101, JLAB-THY-00-33]

   Relying on Radon transform, $\mathcal{R}$

   } Polynomiality ✓ Positivity ?

Different modeling strategies and **different problems**

Solution!: Covariant extension
N.Chouika et al.-Eur.Phys.J.C:77(2017)12,906]

Given a DGLAP-GPD, the corresponding ERBL-GPD can be found, such that polynomiality is satisfied.

Covariant extension

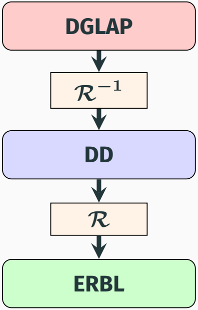
$$H(x,\xi) = \int_{\Omega} d\beta d\alpha \delta(x-\beta-\alpha\xi) h(\beta,\alpha) + \frac{1}{|\xi|} D^{+}(x/\xi) + \text{sign}(\xi) D(x/\xi)$$

[Eur.Phys.J.C:77(2017)12,906]

Covariant extension

$$H(x,\xi) = \int_{\Omega} d\beta d\alpha \delta(x-\beta-\alpha\xi)\, h(\beta,\alpha) + \frac{1}{|\xi|} D^{+}(x/\xi) + \mathrm{sign}(\xi)\, D(x/\xi)$$

[Eur.Phys.J.C:77(2017)12,906]



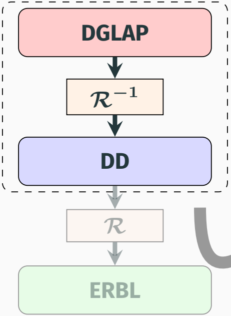
1. Build positive DGLAP GPD
2. Covariant extension: ERBL GPD
   - 2.1. Invert Radon transform
   - 2.2. Determine double distribution
   - 2.3. Compute ERBL GPD

GPD properties	
Support	✓
Positivity	✓
Polynomiality	✓

Covariant extension

$$H\left(x,\xi\right) = \int_{\Omega} d\beta d\alpha \delta\left(x-\beta-\alpha\xi\right) h\left(\beta,\alpha\right) + \frac{1}{|\xi|} D^{+}\left(x/\xi\right) + \text{sign}\left(\xi\right) D\left(x/\xi\right)$$

[Eur.Phys.J.C:77(2017)12,906]



1. Build positive DGLAP GPD
2. Covariant extension: ERBL GPD
   - 2.1. Invert Radon transform
   - 2.2. Determine double distribution
   - 2.3. Compute ERBL GPD

GPD properties	
Support	✓
Positivity	✓
Polynomiality	✓

$$H\left(x,\xi\right)\big|_{|x|\geq\xi} = \mathcal{R}\left[h\left(\beta,\alpha\right)\right] \Rightarrow h\left(\beta,\alpha\right) = \mathcal{R}^{-1}\left[H\left(x,\xi\right)\right]$$

Can we find the inverse Radon transform?

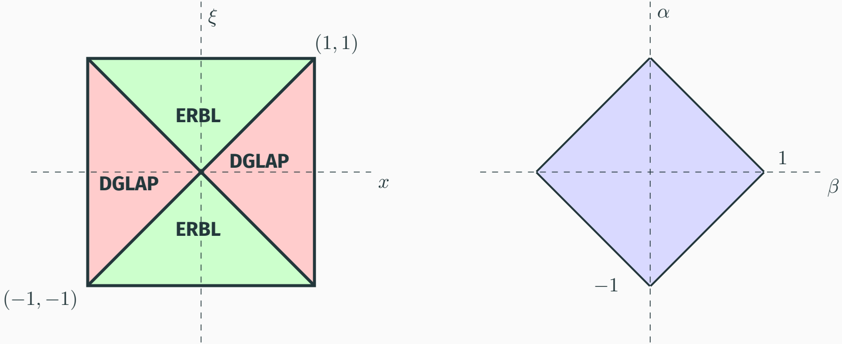
The inverse Radon transform

Inverse Radon transform: graphical realization

$$H(x,\xi) = \mathcal{R}[h] \equiv \int_\Omega d\beta d\alpha \delta(x - \beta - \alpha\xi)\, h(\beta,\alpha)$$

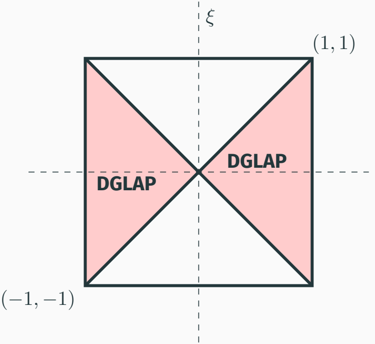
Inverse Radon transform: graphical realization

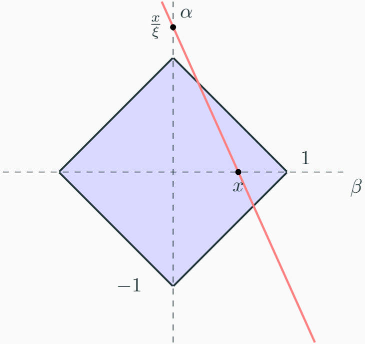
$$H\left(x,\xi\right)=\mathcal{R}\left[h\right]\equiv\int_{\Omega}d\beta d\alpha\delta\left(x-\beta-\alpha\xi\right)h\left(\beta,\alpha\right)$$



Inverse Radon transform: graphical realization

$$H\left(x,\xi\right)=\mathcal{R}\left[h\right]\equiv\int_{\Omega}d\beta d\alpha\delta\left(x-\beta-\alpha\xi\right)h\left(\beta,\alpha\right)$$



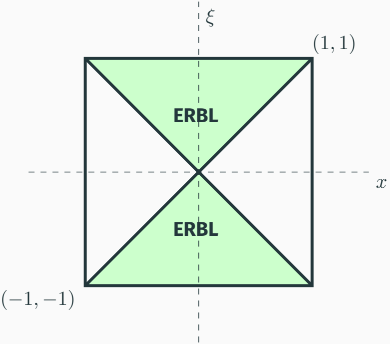


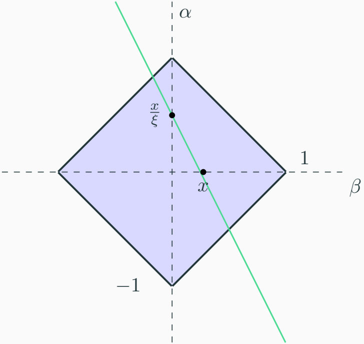
The Radon transform can be realized as a line integral over:

$$\alpha=\frac{x}{\xi}-\frac{\beta}{\xi}$$

Inverse Radon transform: graphical realization

$$H(x,\xi) = \mathcal{R}[h] \equiv \int_\Omega d\beta d\alpha \delta(x - \beta - \alpha\xi)\, h(\beta,\alpha)$$



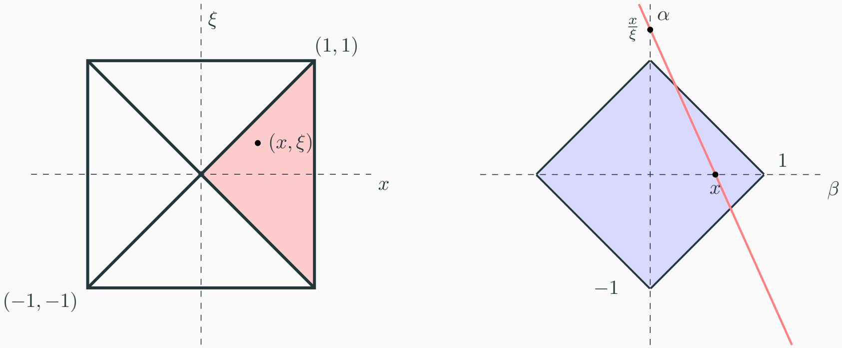


The Radon transform can be realized as a line integral over:

$$\alpha = \frac{x}{\xi} - \frac{\beta}{\xi}$$

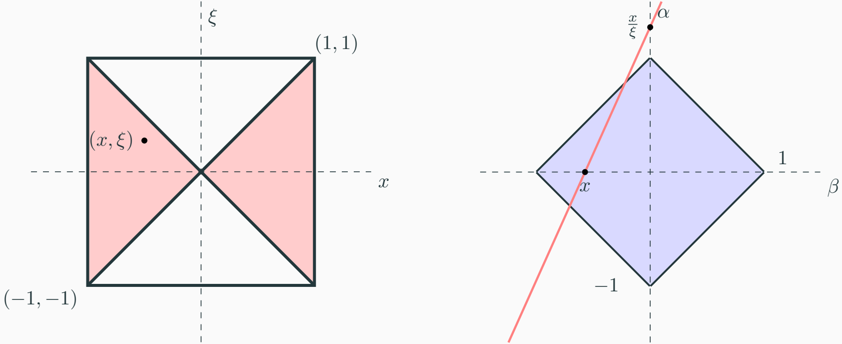
Inverse Radon transform: problem simplification

$$H\left(x,\xi\right)\big|_{|x|\geq\xi} = \mathcal{R}\left[h\right] = \int_{\Omega} d\beta d\alpha \delta\left(x-\beta-\alpha\xi\right) h\left(\beta,\alpha\right)$$



Inverse Radon transform: problem simplification

$$H\left(x,\xi\right)\big|_{|x|\geq\xi} = \mathcal{R}\left[h\right] = \int_{\Omega} d\beta d\alpha \delta\left(x-\beta-\alpha\xi\right) h\left(\beta,\alpha\right)$$



Inverse Radon transform: problem simplification

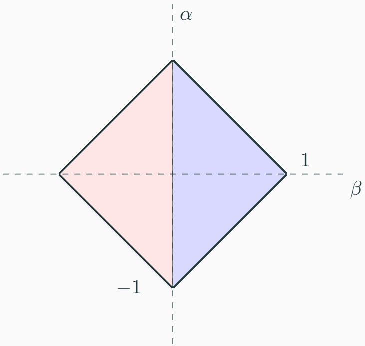
$$H\left(x,\xi\right)\big|_{|x|\geq\xi} = \mathcal{R}\left[h\right] = \int_\Omega d\beta d\alpha \delta\left(x-\beta-\alpha\xi\right) h\left(\beta,\alpha\right)$$

Problem simplification

Uncorrelated $\beta \geq 0$ and $\beta < 0$ regions. [Phys.Rept:388(2003)41]

$$h\left(\beta,\alpha\right) = \boldsymbol{\theta\left(\beta\right) h^{>}\left(\beta,\alpha\right)} + \boldsymbol{\theta\left(-\beta\right) h^{<}\left(\beta,\alpha\right)}$$

$$H\left(x,\xi\right)\big|_{|x|\geq|\xi|} = \boldsymbol{H^{>}\left(x,\xi\right)\big|_{x\geq\xi}} + \boldsymbol{H^{<}\left(x,\xi\right)\big|_{x\leq-\xi}}$$



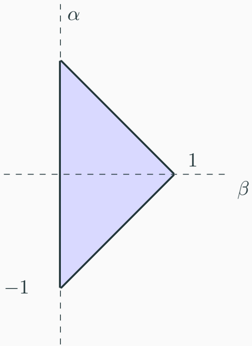
Inverse Radon transform: problem simplification

$$H(x,\xi)\big|_{|x|\geq\xi} = \mathcal{R}[h] = \int_{\Omega} d\beta d\alpha \delta(x-\beta-\alpha\xi)\, h(\beta,\alpha)$$

Problem simplification

Uncorrelated $\beta \geq 0$ and $\beta < 0$ regions. [Phys.Rept:388(2003)41]

Focus on quark GPDs
($\beta \geq 0$)



Inverse Radon transform: problem simplification

$$H(x,\xi)\big|_{x\geq\xi} = \mathcal{R}[h] \equiv \int_{\Omega_>} d\beta d\alpha \delta(x-\beta-\alpha\xi)\, h(\beta,\alpha)$$

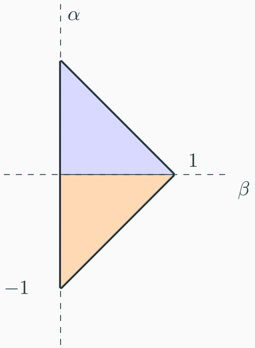
Problem simplification

Uncorrelated $\beta \geq 0$ and $\beta < 0$ regions. [Phys.Rept:388(2003)41]

Focus on quark GPDs
$(\beta \geq 0)$

Symmetry of DDs.
[Eur.Phys.J.C:5(1998)119]

$$h(\beta,\alpha) = h(\beta,-\alpha)$$



Inverse Radon transform: problem simplification

$$H(x,\xi)\big|_{x\geq\xi} = \mathcal{R}[h] \equiv \int_{\Omega^{>}} d\beta d\alpha \delta(x-\beta-\alpha\xi)\, h(\beta,\alpha)$$

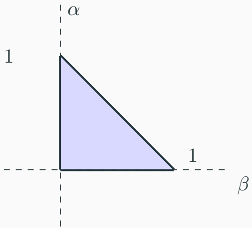
Problem simplification

Uncorrelated $\beta \geq 0$ and $\beta < 0$ regions. [Phys.Rept:388(2003)41]

Focus on quark GPDs ($\beta \geq 0$)

Symmetry of DDs. [Eur.Phys.J.C:5(1998)119]

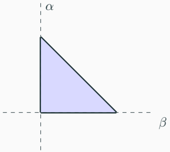
Focus on upper triangle ($\alpha \geq 0$)



$\Omega^{+} = \Omega \cap \{\alpha + \beta \leq 1\}$

Inverse Radon transform

$$H(x,\xi)|_{x\geq\xi} = \mathcal{R}[h] \equiv \int_{\Omega^+} d\beta d\alpha \delta(x-\beta-\alpha\xi)\, h(\beta,\alpha)$$

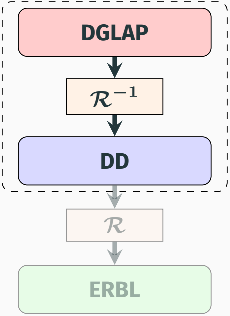


How can we find the inverse Radon transform?

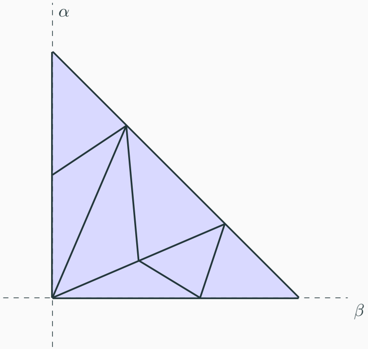
1. Discretize DD domain
2. Interpolate DD

} FEM

3. Sample DD domain
   3.1. Build system's matrix
4. Find sytem's solution



Step 1: Problem discretization

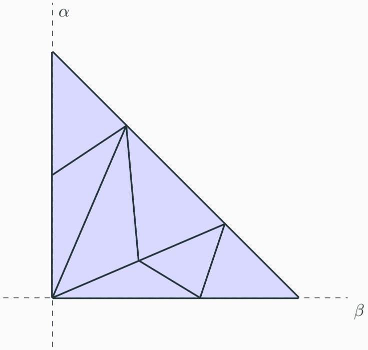


- Build *Delaunay* triangulation (`Triangle C` library*)

$$H(x,\xi) = \mathcal{R}\left[h(\beta,\alpha)\right]$$

* [J. R. Shewchuk. Applied Computational Geometry Towards Geometric Engineering, pp. 203–222, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.]

Step 1: Problem discretization



- Build *Delaunay* triangulation (`Triangle C` library*)

$$H\left(x,\xi\right)=\mathcal{R}\left[h\left(\beta,\alpha\right)\right]$$

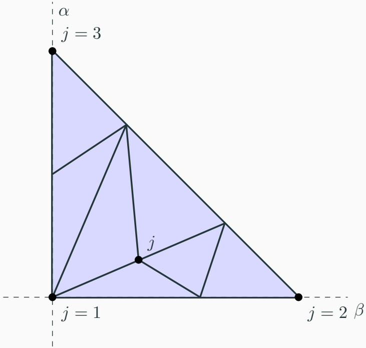
Integral problem becomes a system of equations

$$H^{\mathrm{DGLAP}}\left(x_i,\xi_i\right)=\mathcal{R}_{ij}\left[h\left(\beta_j,\alpha_j\right)\right]$$

*[J. R. Shewchuk. Applied Computational Geometry Towards Geometric Engineering, pp. 203–222, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.]

Inverse Radon transform: Step 2 (interpolation)

Step 2: Interpolate DD within discretized domain.



- Approximate DD within discrete domain

$$h(\beta, \alpha) = \sum_{j=1}^{n} h_j v_j(\beta, \alpha)$$

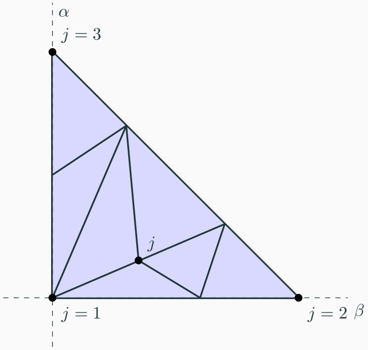
Nodes: j

Basis functions: $v_j(\beta, \alpha)$.

DD value at node: h_j.

Inverse Radon transform: Step 2 (interpolation)

Step 2: Interpolate DD within discretized domain.



- Approximate DD within discrete domain

$$h(\beta, \alpha) = \sum_{j=1}^{n} h_j v_j (\beta, \alpha)$$

Nodes: j

Basis functions: $v_j(\beta, \alpha)$.

DD value at node: h_j.

- Discretize integral problem

$$H^{\mathrm{DGLAP}}(x, \xi) = \sum_{j=1}^{n} h_j \mathcal{R}\left[v_j(\beta, \alpha)\right]$$

Inverse Radon transform: Step 2 (interpolation)

Lagrange P1 polynomials: defined with respect to a given node, j, are degree one polynomials in two dimensions, $v_j(\beta, \alpha)$, satisfying:

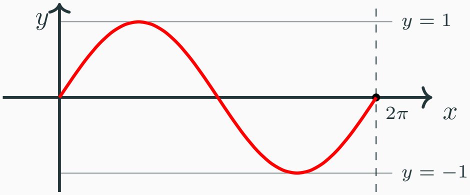
- $v_j(\beta_j, \alpha_j) = 1$
- $v_j(\beta_{i\neq j}, \alpha_{i\neq j}) = 0$
- Domain restricted to elements adjacent to node j.

Inverse Radon transform: Step 2 (interpolation)

Lagrange P1 polynomials: defined with respect to a given node, j, are degree one polynomials in two dimensions, $v_j(\beta, \alpha)$, satisfying:

- $v_j(\beta_j, \alpha_j) = 1$
- $v_j(\beta_{i\neq j}, \alpha_{i\neq j}) = 0$
- Domain restricted to elements adjacent to node j.

1D example: $f(x) = \sin(x) \quad x \in [0, 2\pi]$

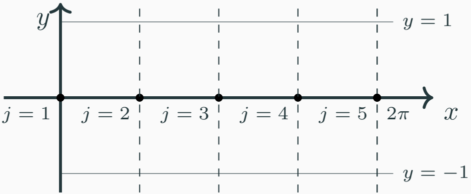


Inverse Radon transform: Step 2 (interpolation)

Lagrange P1 polynomials: defined with respect to a given node, j, are degree one polynomials in two dimensions, $v_j\left(\beta, \alpha\right)$, satisfying:

- $v_j\left(\beta_j, \alpha_j\right) = 1$
- $v_j\left(\beta_{i\neq j}, \alpha_{i\neq j}\right) = 0$
- Domain restricted to elements adjacent to node j.

1D example: $f\left(x\right) = \sin\left(x\right) \quad x \in [0, 2\pi]$



1. Discretize domain

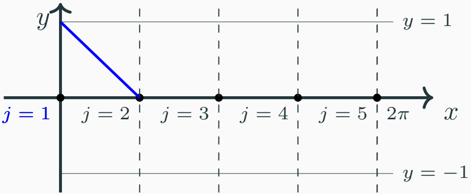
4 elements (5 nodes)

Inverse Radon transform: Step 2 (interpolation)

Lagrange P1 polynomials: defined with respect to a given node, j, are degree one polynomials in two dimensions, $v_j(\beta, \alpha)$, satisfying:

- $v_j(\beta_j, \alpha_j) = 1$
- $v_j(\beta_{i \neq j}, \alpha_{i \neq j}) = 0$
- Domain restricted to elements adjacent to node j.

1D example: $f(x) = \sin(x) \quad x \in [0, 2\pi]$



1. Discretize domain

4 elements (5 nodes)

2. Build basis: v_j

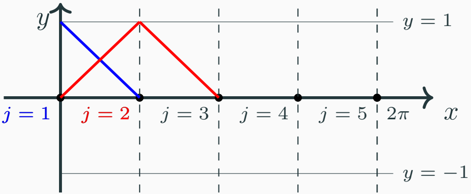
$v_1(x)$

Inverse Radon transform: Step 2 (interpolation)

Lagrange P1 polynomials: defined with respect to a given node, j, are degree one polynomials in two dimensions, $v_j(\beta, \alpha)$, satisfying:

- $v_j(\beta_j, \alpha_j) = 1$
- $v_j(\beta_{i \neq j}, \alpha_{i \neq j}) = 0$
- Domain restricted to elements adjacent to node j.

1D example: $f(x) = \sin(x) \quad x \in [0, 2\pi]$



1. Discretize domain

4 elements (5 nodes)

2. Build basis: v_j

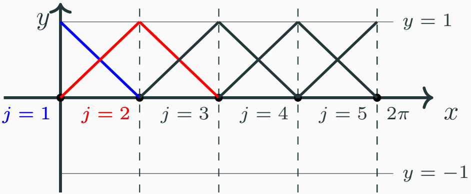
$v_1(x)$ $v_2(x)$

Inverse Radon transform: Step 2 (interpolation)

Lagrange P1 polynomials: defined with respect to a given node, j, are degree one polynomials in two dimensions, $v_j(\beta, \alpha)$, satisfying:

- $v_j(\beta_j, \alpha_j) = 1$
- $v_j(\beta_{i\neq j}, \alpha_{i\neq j}) = 0$
- Domain restricted to elements adjacent to node j.

1D example: $f(x) = \sin(x) \quad x \in [0, 2\pi]$



1. Discretize domain

   4 elements (5 nodes)

2. Build basis: v_j

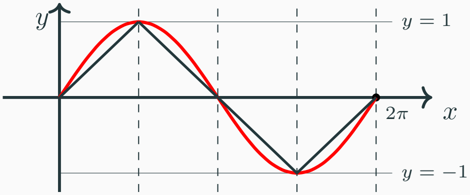
   $v_1(x)$ $v_2(x)$ $\cdots$

Inverse Radon transform: Step 2 (interpolation)

Lagrange P1 polynomials: defined with respect to a given node, j, are degree one polynomials in two dimensions, $v_j(\beta, \alpha)$, satisfying:

- $v_j(\beta_j, \alpha_j) = 1$
- $v_j(\beta_{i \neq j}, \alpha_{i \neq j}) = 0$
- Domain restricted to elements adjacent to node j.

1D example: $f(x) = \sin(x) \quad x \in [0, 2\pi]$



1. Discretize domain

4 elements (5 nodes)

2. Build basis: v_j

$v_1(x)\ v_2(x) \cdots$

3. Interpolate $f(x)$

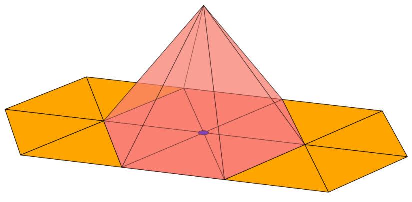
$$f(x) = \sum_{j=1}^{5} f_j v_j(x)$$

Inverse Radon transform: Step 2 (interpolation)

Lagrange P1 polynomials: defined with respect to a given node, j, are degree one polynomials in two dimensions, $v_j(\beta, \alpha)$, satisfying:

- $v_j(\beta_j, \alpha_j) = 1$
- $v_j(\beta_{i \neq j}, \alpha_{i \neq j}) = 0$
- Domain restricted to elements adjacent to node j.

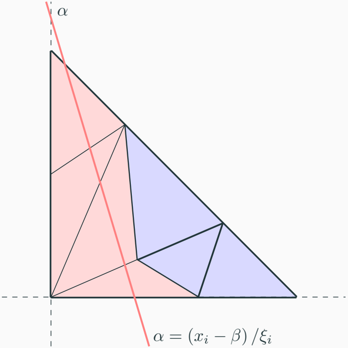
2D case: $h(\beta, \alpha) \quad (\beta, \alpha) \in \Omega^+$



Step 3: Domain sampling

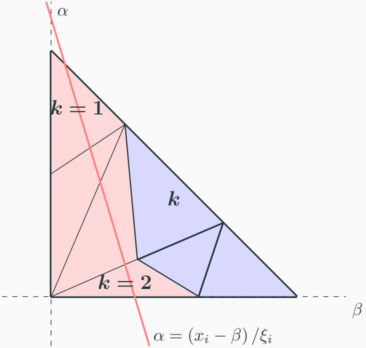
$$H^{\mathrm{DGLAP}}\left(x_i,\xi_i\right)=\sum_{j=1}^{n}h_j\mathcal{R}_i\left[v_j\left(\beta,\alpha\right)\right]=\sum_{j=1}^{n}h_j\left[\int_{\Omega^+}d\beta d\alpha\delta\left(x_i-\beta-\alpha\xi_i\right)v_j\left(\beta,\alpha\right)\right]$$

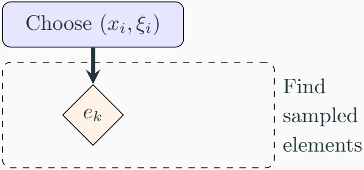
Choose (x_i,ξ_i)



Step 3: Domain sampling

$$H^{\mathrm{DGLAP}}\left(x_i,\xi_i\right)=\sum_{j=1}^{n}h_j\mathcal{R}_i\left[v_j\left(\beta,\alpha\right)\right]=\sum_{j=1}^{n}h_j\left[\int_{\Omega^+}d\beta d\alpha\delta\left(x_i-\beta-\alpha\xi_i\right)v_j\left(\beta,\alpha\right)\right]$$

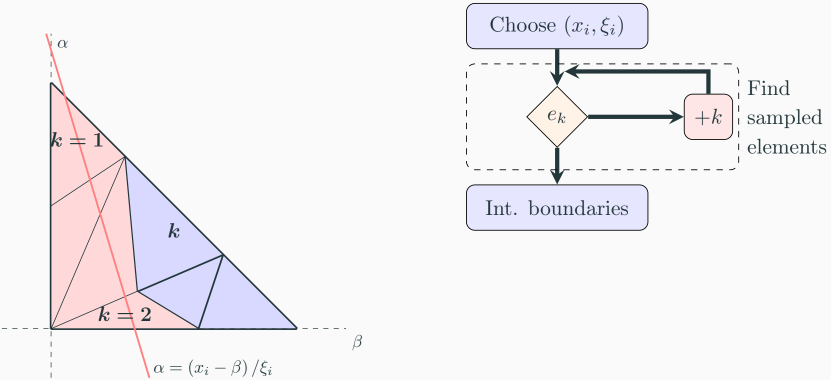




Inverse Radon transform: Step 3 (sampling)

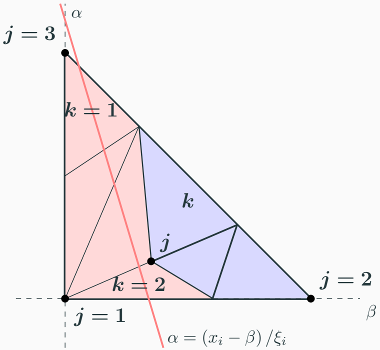
Step 3: Domain sampling

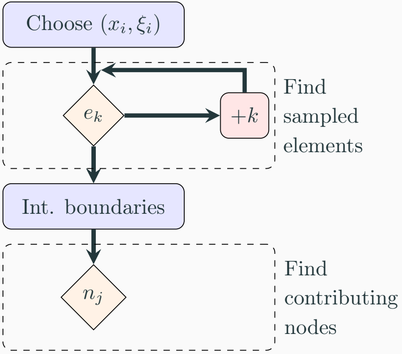
$$H^{\mathrm{DGLAP}}\left(x_i,\xi_i\right)=\sum_{j=1}^{n}h_j\mathcal{R}_i\left[v_j\left(\beta,\alpha\right)\right]=\sum_{j=1}^{n}h_j\left[\int_{\Omega+}d\beta d\alpha\delta\left(x_i-\beta-\alpha\xi_i\right)v_j\left(\beta,\alpha\right)\right]$$



Step 3: Domain sampling

$$H^{\mathrm{DGLAP}}\left(x_i,\xi_i\right)=\sum_{j=1}^{n}h_j\mathcal{R}_i\left[v_j\left(\beta,\alpha\right)\right]=\sum_{j=1}^{n}h_j\left[\int_{\Omega+}d\beta d\alpha\delta\left(x_i-\beta-\alpha\xi_i\right)v_j\left(\beta,\alpha\right)\right]$$

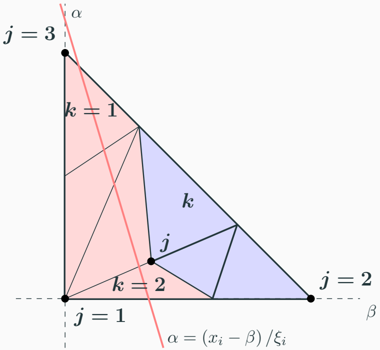


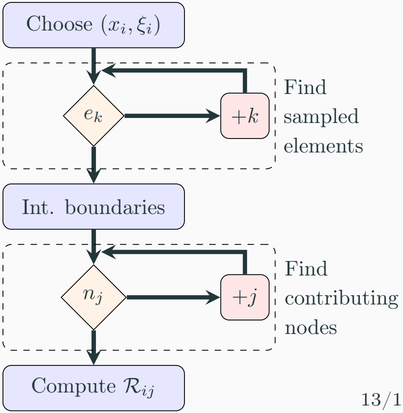


Inverse Radon transform: Step 3 (sampling)

Step 3: Domain sampling

$$H^{\mathrm{DGLAP}}(x_i,\xi_i)=\sum_{j=1}^{n}h_j\mathcal{R}_i\left[v_j(\beta,\alpha)\right]=\sum_{j=1}^{n}h_j\left[\int_{\Omega+}d\beta d\alpha\delta\left(x_i-\beta-\alpha\xi_i\right)v_j(\beta,\alpha)\right]$$





Inverse Radon transform: Step 3 (sampling)

$$H^{\mathrm{DGLAP}}\left(x_i,\xi_i\right)=\sum_{j=1}^{n}h_j\mathcal{R}_i\left[v_j\left(\beta,\alpha\right)\right]=\sum_{j=1}^{n}h_j\left[\int_{\Omega^+}d\beta d\alpha\delta\left(x_i-\beta-\alpha\xi_i\right)v_j\left(\beta,\alpha\right)\right]$$

$$\begin{pmatrix} H^{\mathrm{DGLAP}}\left(x_1,\xi_1\right) \\ H^{\mathrm{DGLAP}}\left(x_2,\xi_2\right) \\ \vdots \\ H^{\mathrm{DGLAP}}\left(x_m,\xi_m\right) \end{pmatrix} = \begin{pmatrix} \mathcal{R}_1\left[v_1\left(\beta,\alpha\right)\right] & \cdots & \mathcal{R}_1\left[v_n\left(\beta,\alpha\right)\right] \\ \mathcal{R}_2\left[v_1\left(\beta,\alpha\right)\right] & \cdots & \mathcal{R}_2\left[v_n\left(\beta,\alpha\right)\right] \\ \vdots & \ddots & \vdots \\ \mathcal{R}_m\left[v_1\left(\beta,\alpha\right)\right] & \cdots & \mathcal{R}_m\left[v_n\left(\beta,\alpha\right)\right] \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_n \end{pmatrix}$$

Inverse Radon transform: Step 3 (sampling)

$$H^{\mathrm{DGLAP}}(x_i,\xi_i)=\sum_{j=1}^{n}h_j\mathcal{R}_i\left[v_j(\beta,\alpha)\right]=\sum_{j=1}^{n}h_j\left[\int_{\Omega^+}d\beta d\alpha\delta(x_i-\beta-\alpha\xi_i)\,v_j(\beta,\alpha)\right]$$

$$\begin{pmatrix} H^{\mathrm{DGLAP}}(x_1,\xi_1)\\ H^{\mathrm{DGLAP}}(x_2,\xi_2)\\ \vdots \\ H^{\mathrm{DGLAP}}(x_m,\xi_m)\end{pmatrix}=\begin{pmatrix}\mathcal{R}_1\left[v_1(\beta,\alpha)\right] & \cdots & \mathcal{R}_1\left[v_n(\beta,\alpha)\right]\\ \mathcal{R}_2\left[v_1(\beta,\alpha)\right] & \cdots & \mathcal{R}_2\left[v_n(\beta,\alpha)\right]\\ \vdots & \ddots & \vdots\\ \mathcal{R}_m\left[v_1(\beta,\alpha)\right] & \cdots & \mathcal{R}_m\left[v_n(\beta,\alpha)\right]\end{pmatrix}\begin{pmatrix}h_1\\ h_2\\ \vdots\\ h_n\end{pmatrix}$$

Integral problem is turned into a system of algebraic equations

$$H^{DGLAP}=\mathcal{R}h$$

Inverse Radon transform: Step 4 (inversion)

Step 4: Solve the inverse problem

$$H^{\mathrm{DGLAP}}(x_i,\xi_i) \equiv \boxed{\boldsymbol{H_i^{\mathrm{DGLAP}} = \mathcal{R}_{ij} h_j}} \equiv \sum_{j=1}^{n} h_j \left[\int_{\Omega^+} d\beta d\alpha \delta\left(x_i - \beta - \alpha\xi_i\right) v_j\left(\beta,\alpha\right)\right]$$

Inverse Radon transform: Step 4 (inversion)

Step 4: Solve the inverse problem

$$H^{\mathrm{DGLAP}}(x_i,\xi_i) \equiv \boxed{\boldsymbol{H_i^{\mathrm{DGLAP}} = \mathcal{R}_{ij} h_j}} \equiv \sum_{j=1}^{n} h_j \left[\int_{\Omega^+} d\beta d\alpha \delta\left(x_i - \beta - \alpha\xi_i\right) v_j\left(\beta,\alpha\right)\right]$$

- Compute inverse Radon transform matrix (Least-Squares)

$$\chi^2 = \frac{1}{\sigma^2}\sum_i \left(H_i^{\mathrm{DGLAP}} - \sum_j \mathcal{R}_{ij} h_j\right)^2 \xrightarrow[\frac{\partial}{\partial h_k}]{} \sum_i H_i \mathcal{R}_{ik} = \sum_{i,j} \mathcal{R}_{ij} h_j \mathcal{R}_{ik}$$

Inverse Radon transform: Step 4 (inversion)

Step 4: Solve the inverse problem

$$H^{\mathrm{DGLAP}}(x_i,\xi_i) \equiv \boxed{\boldsymbol{H_i^{\mathrm{DGLAP}} = \mathcal{R}_{ij} h_j}} \equiv \sum_{j=1}^{n} h_j \left[\int_{\Omega^+} d\beta d\alpha \delta\left(x_i - \beta - \alpha\xi_i\right) v_j\left(\beta,\alpha\right)\right]$$

- Compute inverse Radon transform matrix (Least-Squares)

$$\chi^2 = \frac{1}{\sigma^2}\sum_i \left(H_i^{\mathrm{DGLAP}} - \sum_j \mathcal{R}_{ij} h_j\right)^2 \xrightarrow[\frac{\partial}{\partial h_k}]{} \sum_i H_i \mathcal{R}_{ik} = \sum_{i,j} \mathcal{R}_{ij} h_j \mathcal{R}_{ik}$$

$$\mathcal{R}^T H^{DGLAP} = \mathcal{R}^T \mathcal{R} h \Rightarrow h = \left(\boldsymbol{\mathcal{R}^T \mathcal{R}}\right)^{-1} \mathcal{R}^T H^{\mathrm{DGLAP}}$$

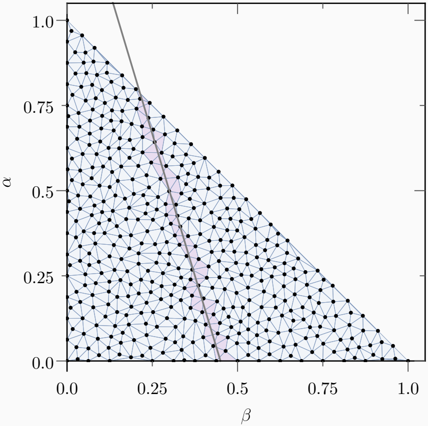
The matrix $\mathcal{R}^T\mathcal{R}$ can be inverted

[Phys.Rev.D:105(2022)9,094012]

Hands on!

How can we find the inverse Radon transform?

1. Discretization **(area < 0.01)**
   (`Triangle C` library*)
   427 vertices - 780 elements
2. P1 interpolation
3. Sample DD domain
   3120 $(4 \cdot 780)$ lines
   Good conditioning
4. Find system's solution
   (`Eigen3` library†)

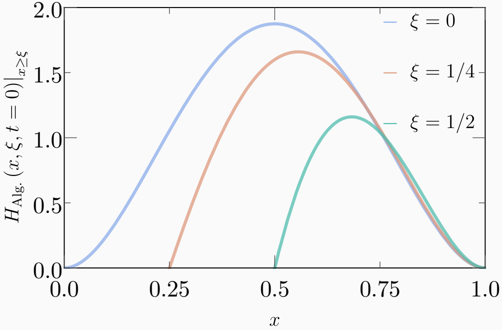


*[J. R. Shewchuk. Applied Computational Geometry Towards Geometric Engineering, pp. 203–222, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.]
†[Gaël Guennebaud and Benoît Jacob and others, Eigen v3, 2010.]

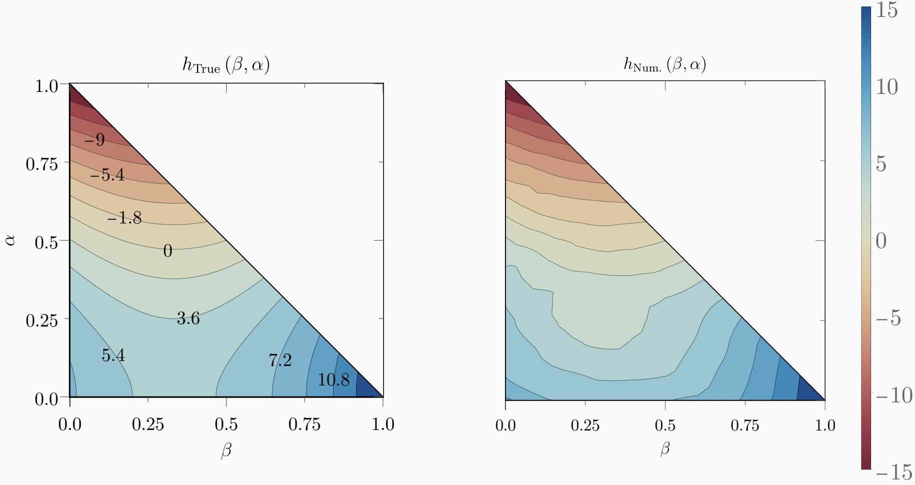
Nakanishi-based model for pions

[Phys.Lett.B:780(2018)287]

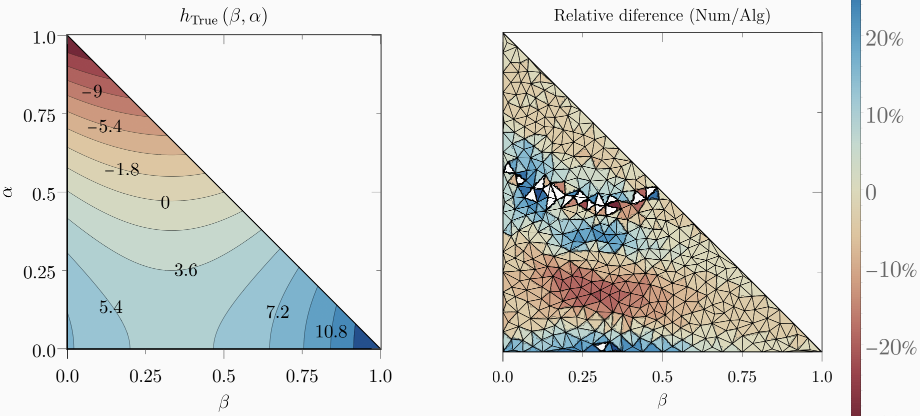
$$H\left(x,\xi\right)\big|_{x\geq\xi} = 30\,\frac{\left(1-x\right)^{2}\left(x^{2}-\xi^{2}\right)}{\left(1-\xi^{2}\right)^{2}}$$



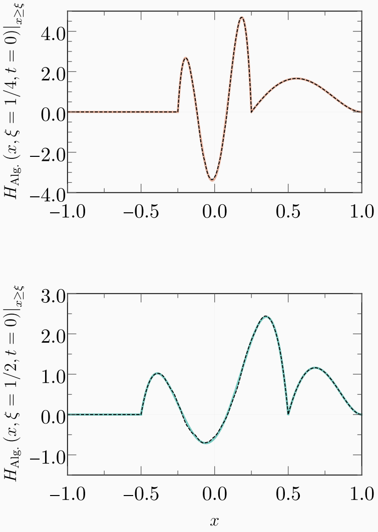
Inverse Radon transform: application to pion GPDs



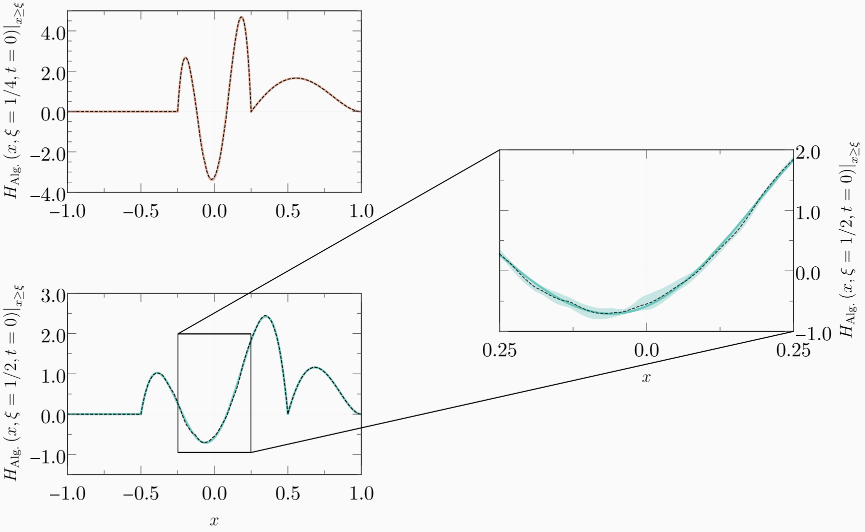
Inverse Radon transform: application to pion GPDs



Inverse Radon transform: application to pion GPDs



Inverse Radon transform: application to pion GPDs



Summary and perspectives

Summary and perspectives

Summary

Covariant extension:

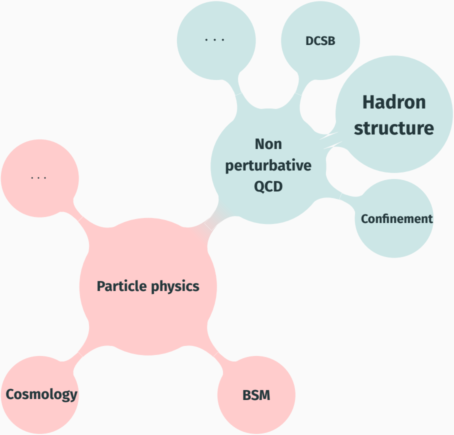
- Systematic procedure to design models for hadron GPDs accounting for all theoretical requirements.
- Crossover of techniques from hadron physics and numerical analysis.

Perspectives

- Explore the effect of adaptive meshes.
- Generalize of the interpolation basis to degree > 1 polynomials.
- Account for correlations in the assessment of uncertainties.
- Suggestions?

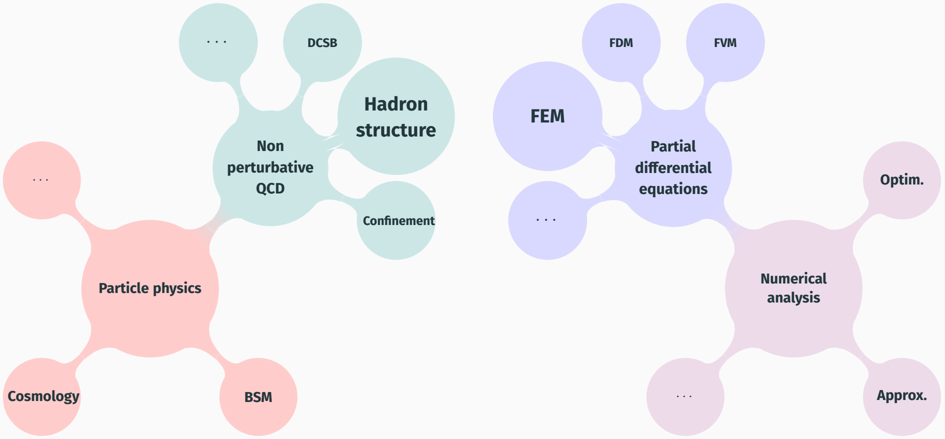
Thank you!

Invitation



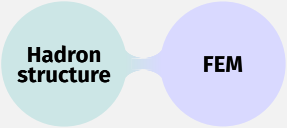
. . .
DCSB
Hadron structure
Non perturbative QCD
. . .
Confinement
Particle physics
Cosmology
BSM

Invitation



. . .
DCSB
Hadron structure
Non perturbative QCD
Confinement
. . .
Particle physics
Cosmology
BSM
FDM
FVM
FEM
Partial differential equations
. . .
Optim.
Numerical analysis
. . .
Approx.

Invitation



Crossover between particle physics and numerical analysis

Finite element methods (FEM) applied to hadron structure (GPDs)

[N.Chouika et al.-Eur.Phys.J.C:77(2017)12,906]

GPD modeling: overlap representation

Overlap representation - GPDs written as overlap of LFWFs.
[Nucl.Phys.B:596(2001)33]

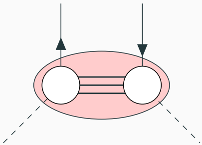
GPD modeling: overlap representation

Overlap representation - GPDs written as overlap of LFWFs.
[Nucl.Phys.B:596(2001)33]

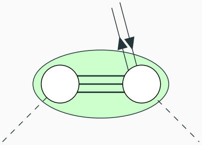
Quantizing a quantum field theory on the lightfront allows to expand a hadron state in a Fock-space basis, *e.g.*:
[Phys.Rept.301(1998)299]

$$|h(p)\rangle \sim \sum_{\beta} \Psi^q_{\beta,N=2} |q\bar{q}\rangle + \Psi^q_{\beta,N=4} |q\bar{q}q\bar{q}\rangle + \dots$$

whose "coefficients" are lightfront wave functions: $\Psi^q\left(x, k_\perp^2\right)$.



Same N LFWFs



N and $N+2$ LFWFs

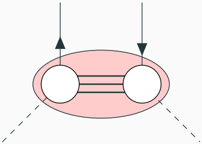
GPD modeling: overlap representation

Overlap representation - GPDs written as overlap of LFWFs.
[Nucl.Phys.B:596(2001)33]

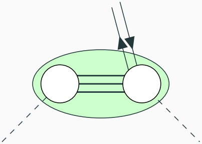
Quantizing a quantum field theory on the lightfront allows to expand a hadron state in a Fock-space basis, *e.g.*:
[Phys.Rept.301(1998)299]

$$|h(p)\rangle \sim \sum_{\beta} \Psi^q_{\beta,N=2} |q\bar{q}\rangle + \Psi^q_{\beta,N=4} |q\bar{q}q\bar{q}\rangle + \ldots$$

whose "coefficients" are lightfront wave functions: $\Psi^q\left(x, k_\perp^2\right)$.



Same N LFWFs



N and $N+2$ LFWFs

Overlap representation: positivity inbuilt but polynomiality is lost

GPD modeling: double distribution representation

DD representation - GPDs written as Radon transform of DDs.
[Fortsch.Phys.:42(1994)101, JLAB-THY-00-33]

$$H(x,\xi) = \int_{\Omega} d\beta d\alpha \delta(x-\beta-\alpha\xi)\left[f(\beta,\alpha)+\xi g(\beta,\alpha)\right]$$

Polynomiality is explictly fulfilled

$$\int_{-1}^{1} dx x^n H^q(x,\xi) = \sum_{j=0}^{n} \binom{n}{j} \xi^j \int_{\Omega} d\beta d\alpha \beta^{n-j}\alpha^j \left[f(\beta,\alpha)+\xi g(\beta,\alpha)\right]$$

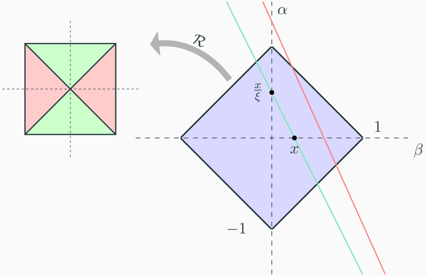
GPD modeling: double distribution representation

DD representation - GPDs written as Radon transform of DDs.
[Fortsch.Phys.:42(1994)101, JLAB-THY-00-33]

$$H(x,\xi) = \int_\Omega d\beta d\alpha \delta(x-\beta-\alpha\xi)\left[f(\beta,\alpha)+\xi g(\beta,\alpha)\right]$$

Polynomiality is explictly fulfilled

$$\int_{-1}^{1} dx x^n H^q(x,\xi) = \sum_{j=0}^{n} \binom{n}{j} \xi^j \int_\Omega d\beta d\alpha \beta^{n-j}\alpha^j \left[f(\beta,\alpha)+\xi g(\beta,\alpha)\right]$$



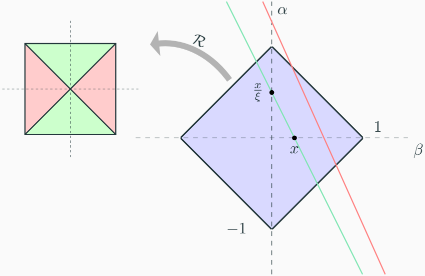
GPD modeling: double distribution representation

DD representation - GPDs written as Radon transform of DDs.
[Fortsch.Phys.:42(1994)101, JLAB-THY-00-33]

$$H\left(x,\xi\right)=\int_{\Omega}d\beta d\alpha\delta\left(x-\beta-\alpha\xi\right)\left[f\left(\beta,\alpha\right)+\xi g\left(\beta,\alpha\right)\right]$$

Polynomiality is explictly fulfilled

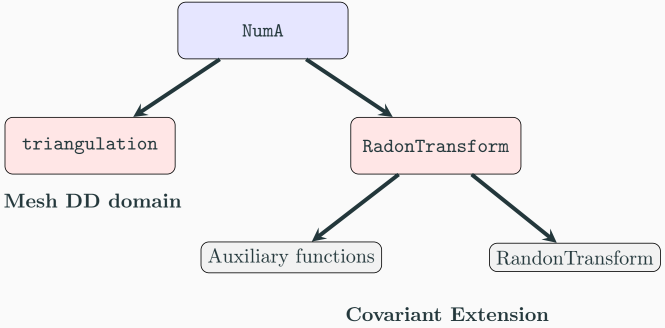
$$\int_{-1}^{1}dx x^{n}H^{q}\left(x,\xi\right)=\sum_{j=0}^{n}\binom{n}{j}\xi^{j}\int_{\Omega}d\beta d\alpha\beta^{n-j}\alpha^{j}\left[f\left(\beta,\alpha\right)+\xi g\left(\beta,\alpha\right)\right]$$



Polynomiality fulfilled, positivity not granted.

The Radon transform module

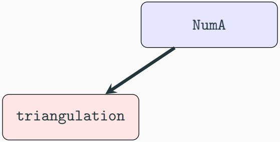
`RadonTransform` is a module implemented in `NumA` allowing to perform the covariant extension of GPDs from the DGLAP to the ERBL region.



The triangulation module (Step 1)

Step 1: Problem discretization

Triangulation takes care of the first step, i.e. builds a mesh over the double distribution domain.



It is made up from two main blocks

- **Triangle** software (compiled as an static library)
  Builds Delaunay triangulations over a given domain.
- Class **Mesh**

  Objects

  - std::vector<points> vertices
  - std::vector<vector<int» elements   Labels for vertices (sort **vertices**).
  - std::vector«vector<int» vneighbors
  - std::vector«vector<double» nodes

  Methods:

  - Mesh::SetMaximumArea(float area)
  - Mesh::GenerateMesh(): Feeds **triangle** to build mesh.
  - Mesh::Report(int ele, int ver, int neig,std::string)

Covariant extension: DD representation revisited

Given a function $D(\alpha)$ with compact support $\alpha \in [-1,1]$ such that

$$\int_{-1}^{1} d\alpha \alpha^m D(\alpha) = c_{m+1}^m$$

then,

$$\int_{-1}^{1} dx x^m \left[H(x,\xi) - \text{sign}(\xi) D(x/\xi)\right]$$

is a polynomial of order m in ξ.

Under these conditions, Hertle's theorem guarantees that:
[Mat.Zeit.:184(1983)165, Phys.Lett.B::510(2001)125, Eur.Phys.J.C:77(2017)12,906]

$$\begin{aligned} H(x,\xi) &= \text{sign}(\xi) D(x/\xi) + \int_\Omega d\beta d\alpha \delta(x-\beta-\alpha\xi) f(\beta,\alpha) \\ &\equiv \text{sign}(\xi) D(x/\xi) + \mathcal{R}\left[f(\beta,\alpha)\right] \end{aligned}$$

A GPD can always be written as the **Radon transform** of double distributions, thus guaranteeing fulfillment of **polynomiality**.

Covariant extension: existence and uniqueness

Write:

$$\frac{1}{|\xi|} D\left(x/\xi\right) = \mathcal{R}\left[D\left(\alpha\right)\delta\left(\beta\right)\right] \equiv \mathcal{R}\left[g\left(\beta,\alpha\right)\right]$$

$$H\left(x,\xi\right) = \int_{\Omega} d\beta d\alpha \left[f\left(\beta,\alpha\right) + \xi g\left(\beta,\alpha\right)\right]\delta\left(x-\beta-\alpha\xi\right)$$

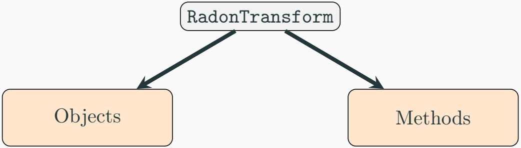
Covariant extension - Boman and Todd-Quinto theorem
[Eur.Phys.J.C:77(2017)12,906, Duke Math.J.:55-4(1987)943]

If $H\left(x,\xi\right) = 0\,\forall\left(x,\xi\right) \in [-1,1]\otimes[-1,1]\,/\,|x| \geq |\xi| \Rightarrow f\left(\beta,\alpha\right) = 0\,\forall\left(\beta \neq 0,\alpha\right) \in \Omega$

DGLAP region characterizes the entire GPD up to ambiguities along the $\beta = 0$ line.

- Ambiguity along $\beta = 0$: $\delta\left(\beta\right) D\left(\alpha\right)$
- If $f\left(\beta,\alpha\right)$ is a distribution, further ambiguity: $\delta\left(\beta\right) D^{+}\left(\alpha\right)$

The Radon transform module (Step 2)



1. NumA::Mesh mesh;
2. std::vector<double> x,y,xi;
3. Eigen::MatrixXd RTMatrix;

Methods:

- RadonTransform::init(): **Main functionality!**
- RadonTransform::build_matrix(x,y,xi)
- RadonTransform::matrix_assembly(x,y,xi)

- RadonTransform::computeDD(const Eigen::VectorXd & GPD)
- RadonTransform::computGPD(const Eigen::VectorXd & DD, const double x, const double xi)
- RadonTransform::computeDterm(const Eigen::VectorXd & DD, const double x, const double xi)

How does it work? (I)

Step 2: Domain sampling (and matrix assembly)

```
RadonTransform::init()
{
  // Step 1:  Discretization
    mesh.SetMaximumArea(0.001);
    mesh.GenerateMesh();

  // Step 2:  Sampling
  // Random distribution of samples
    ...
    for( int i = 0; i < 12*mesh.elements.size(); i++ )
    {
      x[i] = unif(re);
      ...
    }

  // Fill-in Radon transform matrix
    RTMatrix = build_matrix(x,y,xi);
}
```

How does it work? (II)

```
RadonTransform::matrix_assembly(x,y,xi)
{
    std::vector<int> indenti( mesh.elements.size() );
    ...
    // Iteration over sampling lines
    for( int i = 0; i < 12*mesh.elements.size(); i++ )
    {
      // Identify elements ‘‘touched’’ by the chosen line
        indenti=sampling(mesh,x[i],y[i],xi[i]);
        ...
      // Iteration over sampled elements
        for( int j = 0; j < mesh.elements.size(); j++ )
        {
          if(identi[j] )
          {
            ...
            // Compute contribution to Radon transform
              for( int k = 0; k < 3; k++ )
              {
                RTMatrix(i,mesh.elements[j][k]) = integral;
              }
          }
        }
    }
}
```

The Radon transform module (Step 3)

Step 3: Solve inverse problem

```
RadonTransform::computeDD( const Eigen::VectorCd & GPD)
{
```

Once the Radon transform matrix is built and stored in `RTMatrix`, the functionalities of `Eigen` library allow to find "its inverse" and thus determine the double distribution.

```
}
```