Theory and Phenomenology of Generalized Parton Distributions

Nabil Chouika

Irfu/SPhN, CEA Saclay - Université Paris-Saclay

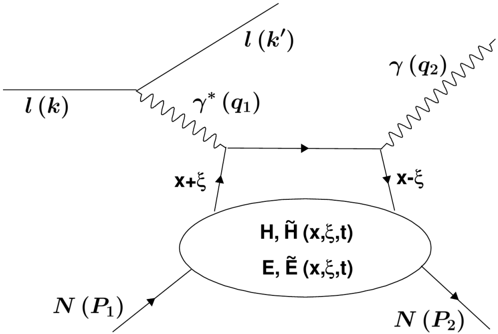
HUGS 2016, 30 May - 18 June, JLab

Outline

Outline

1. Introduction to Generalized Parton Distributions
   - Experimental access to GPDs
   - Nucleon imaging
   - Theoretical constraints on GPDs
2. Ab initio Model for GPDs
   - From Dyson-Schwinger equations to GPDs
   - Radon Transform
3. PARTONS platform
4. Conclusion
   - Summary
   - Bibliography

Experimental access (example: DVCS)



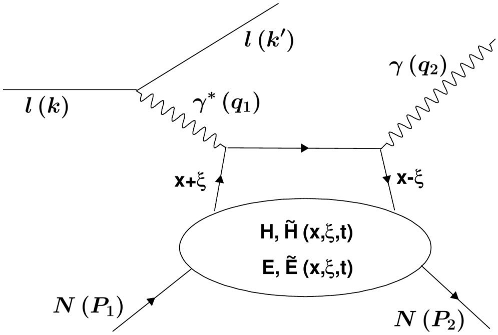
Deeply Virtual Compton Scattering channel of photon electroproduction.

$$\Delta = P_2 - P_1 \ , \ t = \Delta^2 < 0$$

$$Q^2 = -q_1^2 > 0$$

$$P = \frac{1}{2}(P_1 + P_2) \ , \ \xi = -\frac{\Delta^+}{2\,P^+}$$

Experimental access (example: DVCS)



Deeply Virtual Compton Scattering channel of photon electroproduction.

$$\Delta = P_2 - P_1 \;,\; t = \Delta^2 < 0$$

$$Q^2 = -q_1^2 > 0$$

$$P = \frac{1}{2}\left(P_1 + P_2\right) \;,\; \xi = -\frac{\Delta^+}{2\,P^+}$$

Compton Form Factors: **(Belitsky et al., 2002)**

$$\mathcal{F}\left(\xi, t, Q^2\right) = \int_{-1}^{1} \mathrm{d}x\, C\left(x, \xi, \alpha_S\left(\mu_F\right), \frac{Q}{\mu_F}\right) F\left(x, \xi, t, \mu_F\right), \qquad (1)$$

where $F \in \left\{H,\, E,\, \tilde{H},\, \tilde{E},\, \dots\right\}$ is a Generalized Parton Distribution.

Nucleon imaging

- Correlation of the longitudinal momentum and the transverse position of the partons inside the hadron.

Nucleon imaging

- Correlation of the longitudinal momentum and the transverse position of the partons inside the hadron.
- Probability density (Fourier transform of GPD): **(Burkardt, 2000)**

$$q\left(x, \vec{b_\perp}\right) = \int \frac{\mathrm{d}^2 \vec{\Delta_\perp}}{(2\pi)^2} e^{-i\,\vec{b_\perp}\cdot\vec{\Delta_\perp}} H^q\left(x, 0, -\vec{\Delta_\perp}^2\right) . \tag{2}$$

Nucleon imaging

- Correlation of the longitudinal momentum and the transverse position of the partons inside the hadron.
- Probability density (Fourier transform of GPD): **(Burkardt, 2000)**

$$q\left(x,\vec{b_\perp}\right) = \int \frac{\mathrm{d}^2\vec{\Delta_\perp}}{(2\pi)^2}\, e^{-i\,\vec{b_\perp}\cdot\vec{\Delta_\perp}}\, H^q\left(x,0,-\vec{\Delta_\perp}^{\,2}\right)\,. \qquad (2)$$

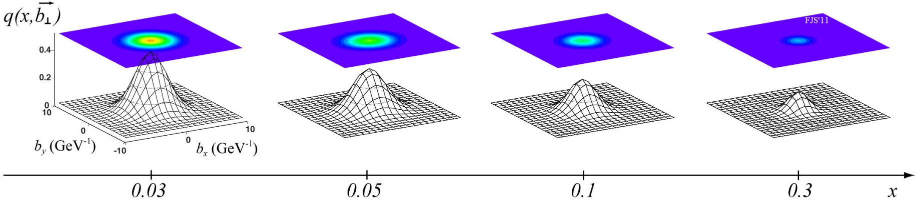


Figure: Hadron tomography.

Theoretical constraints on GPDs

Main properties: **(Diehl, 2003)**

- Support: $x, \xi \in [-1, 1]$.

Theoretical constraints on GPDs

Main properties: **(Diehl, 2003)**

- Support: $x, \xi \in [-1, 1]$.
- Polynomiality:

$$\int_{-1}^{1} \mathrm{d}x\, x^m\, H\,(x, \xi, t) = \text{Polynomial in } \xi\,. \tag{3}$$

Theoretical constraints on GPDs

Main properties: (Diehl, 2003)

- Support: $x, \xi \in [-1, 1]$.
- Polynomiality:

$$\int_{-1}^{1} \mathrm{d}x \, x^m \, H(x, \xi, t) = \text{Polynomial in } \xi \, . \tag{3}$$

  - From Lorentz invariance.

Theoretical constraints on GPDs

Main properties: **(Diehl, 2003)**

- Support: $x, \xi \in [-1, 1]$.
- Polynomiality:

$$\int_{-1}^{1} \mathrm{d}x \, x^m \, H(x, \xi, t) = \text{Polynomial in } \xi \, . \tag{3}$$

  - From Lorentz invariance.

- Positivity:

$$H^q(x, \xi, t) \leq \sqrt{q\left(\frac{x-\xi}{1-\xi}\right) q\left(\frac{x+\xi}{1+\xi}\right)} \, . \tag{4}$$

Theoretical constraints on GPDs

Main properties: (Diehl, 2003)

- Support: $x, \xi \in [-1, 1]$.
- Polynomiality:

$$\int_{-1}^{1} \mathrm{d}x \, x^m \, H(x, \xi, t) = \text{Polynomial in } \xi \, . \tag{3}$$

  - From Lorentz invariance.
- Positivity:

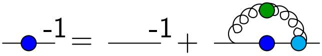
$$H^q(x, \xi, t) \leq \sqrt{q\left(\frac{x-\xi}{1-\xi}\right) q\left(\frac{x+\xi}{1+\xi}\right)} \, . \tag{4}$$

  - Cauchy-Schwarz theorem in Hilbert space.

Outline

From Dyson-Schwinger equations to GPDs

Dyson-Schwinger equations

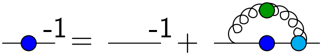


(Maris and Roberts, 1997)

Quark Propagator $S(p)$

From Dyson-Schwinger equations to GPDs

Dyson-Schwinger equations

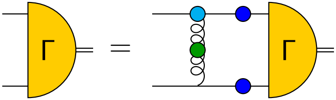


(Maris and Roberts, 1997)

Quark Propagator $S(p)$

↓ ↓

Bethe-Salpeter equation (2-body bound-state)

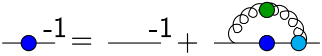


(Eichmann, 2009)

Bethe-Salpeter amplitude

From Dyson-Schwinger equations to GPDs

Dyson-Schwinger equations

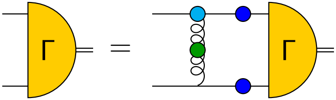


(Maris and Roberts, 1997)

Quark Propagator $S(p)$

↓ ↓

Bethe-Salpeter equation (2-body bound-state)

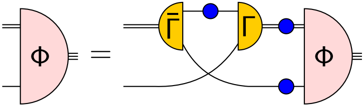


(Eichmann, 2009)

Bethe-Salpeter amplitude

↓ ↓

(Faddeev equation (3-body bound-state)



Faddeev amplitude)

From Dyson-Schwinger equations to GPDs

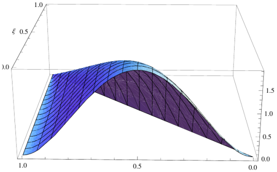
↓ ↓

Hadronic Fock space

$$\Psi\left(k^+, \vec{k_\perp}, P\right) \propto \int \mathrm{d}k^- \; \mathrm{Tr}\left[\gamma^+\gamma_5\,\chi\left(k, P\right)\right]$$

Lightcone Wavefunction Ψ

From Dyson-Schwinger equations to GPDs

↓		↓
Hadronic Fock space	$\Psi\left(k^+, \vec{k_\perp}, P\right) \propto \int \mathrm{d}k^- \,\mathrm{Tr}\left[\gamma^+\gamma_5\,\chi\left(k, P\right)\right]$	Lightcone Wavefunction Ψ
↓		↓
Overlap of LCWF		Generalized Parton Distribution $H\left(x, \xi, t\right)$

(Mezrag, 2015)

Double Distribution (DD)

- Overlap representation: **positivity** fulfilled!

Double Distribution (DD)

- Overlap representation: **positivity** fulfilled!
- DGLAP region only: $|x| > |\xi|$. Need ERBL to complete **polynomiality**.

Double Distribution (DD)

- Overlap representation: **positivity** fulfilled!
- DGLAP region only: $|x| > |\xi|$. Need ERBL to complete **polynomiality**.
- Model fulfills Lorentz invariance:

Double Distribution (DD)

- Overlap representation: **positivity** fulfilled!
- DGLAP region only: $|x| > |\xi|$. Need ERBL to complete **polynomiality**.
- Model fulfills Lorentz invariance:
  - Double distribution $f(\beta, \alpha)$ (1CDD): **(Belitsky et al., 2001)**

$$H(x,\xi) = x \int_{|\alpha|+|\beta|\leq 1} \mathrm{d}\beta \, \mathrm{d}\alpha \, f(\beta,\alpha) \, \delta(x-\beta-\alpha\xi) \, .$$

Double Distribution (DD)

- Overlap representation: **positivity** fulfilled!
- DGLAP region only: $|x| > |\xi|$. Need ERBL to complete **polynomiality**.
- Model fulfills Lorentz invariance:
  - Double distribution $f(\beta, \alpha)$ (1CDD): **(Belitsky et al., 2001)**

$$H(x,\xi) = x \int_{|\alpha|+|\beta|\leq 1} \mathrm{d}\beta \, \mathrm{d}\alpha \, f(\beta,\alpha) \, \delta(x - \beta - \alpha\xi) \; .$$

  - Exists and is unique (up to an ambiguity on $\beta = 0$ axis).

Double Distribution (DD)

- Overlap representation: **positivity** fulfilled!
- DGLAP region only: $|x| > |\xi|$. Need ERBL to complete **polynomiality**.
- Model fulfills Lorentz invariance:
  - Double distribution $f(\beta, \alpha)$ (1CDD): **(Belitsky et al., 2001)**

$$H(x,\xi) = x \int_{|\alpha|+|\beta|\leq 1} \mathrm{d}\beta \, \mathrm{d}\alpha \, f(\beta,\alpha) \, \delta(x - \beta - \alpha\xi) \, .$$

  - Exists and is unique (up to an ambiguity on $\beta = 0$ axis).
  - Reconstruct GPD everywhere.

(Moutarde, 2015)

Inverse Radon Transform

Problem

Find $f(\beta, \alpha)$ on rhombus $\{|\alpha| + |\beta| \leq 1\}$ such that

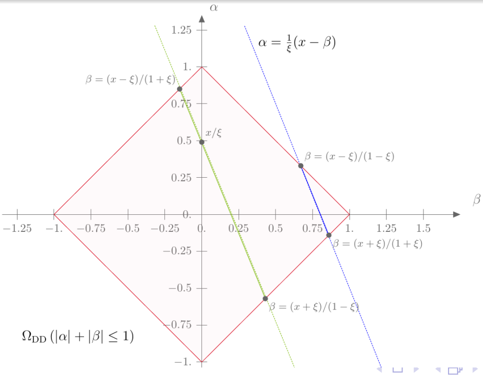
$$H(x, \xi)|_{\mathrm{DGLAP}} = x \int \mathrm{d}\beta\, \mathrm{d}\alpha\, f(\beta, \alpha)\, \delta(x - \beta - \alpha\xi)\ .$$

Inverse Radon Transform

Problem

Find $f(\beta, \alpha)$ on rhombus $\{|\alpha| + |\beta| \leq 1\}$ such that

$$H(x,\xi)|_{\mathrm{DGLAP}} = x \int \mathrm{d}\beta\, \mathrm{d}\alpha\, f(\beta,\alpha)\, \delta(x - \beta - \alpha\xi)\ .$$



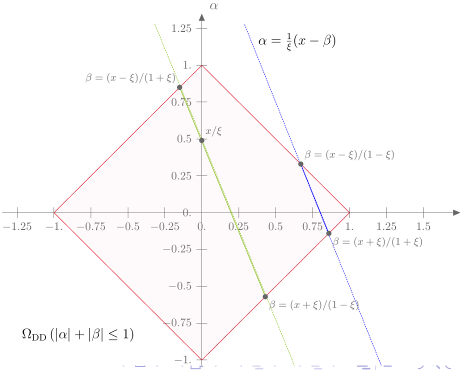
Inverse Radon Transform

Problem

Find $f(\beta, \alpha)$ on rhombus $\{|\alpha| + |\beta| \leq 1\}$ such that

$$H(x, \xi)|_{\mathrm{DGLAP}} = x \int \mathrm{d}\beta \, \mathrm{d}\alpha \, f(\beta, \alpha) \, \delta(x - \beta - \alpha\xi) \, .$$

- Inverse Radon Transform: (mildly) ill-posed problem!



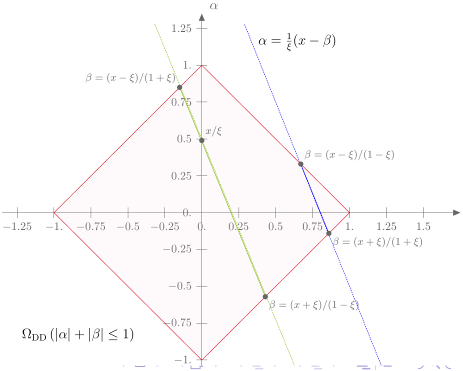
Inverse Radon Transform

Problem

Find $f(\beta,\alpha)$ on rhombus $\{|\alpha|+|\beta|\leq 1\}$ such that

$$H(x,\xi)|_{\mathrm{DGLAP}} = x\int \mathrm{d}\beta\,\mathrm{d}\alpha\, f(\beta,\alpha)\,\delta(x-\beta-\alpha\xi)\ .$$

- Inverse Radon Transform: (mildly) ill-posed problem!
- Limited angle inverse transform ($|\xi|<1$): severely ill-posed!



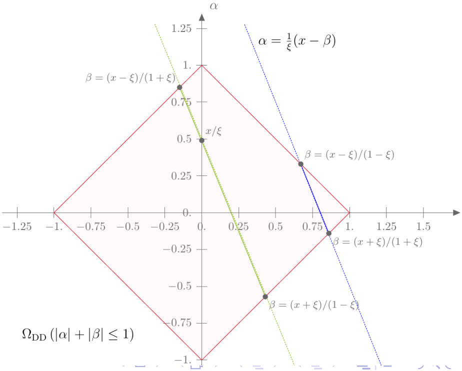
Inverse Radon Transform

Problem

Find $f(\beta, \alpha)$ on rhombus $\{|\alpha| + |\beta| \leq 1\}$ such that

$$H(x,\xi)|_{\mathrm{DGLAP}} = x \int \mathrm{d}\beta\, \mathrm{d}\alpha\, f(\beta,\alpha)\, \delta(x - \beta - \alpha\xi) \; .$$

- Inverse Radon Transform: (mildly) ill-posed problem!
- Limited angle inverse transform ($|\xi| < 1$): severely ill-posed!
- Access only to a limited region (DGLAP: $|x| > |\xi|$): things are probably worse!



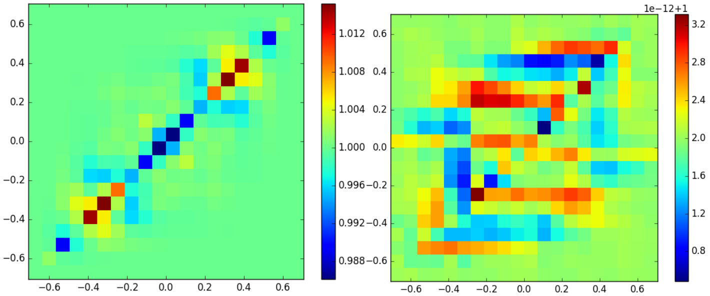
Inverse Radon Transform

Problem

Find $f(\beta, \alpha)$ on rhombus $\{|\alpha| + |\beta| \leq 1\}$ such that

$$H(x, \xi)|_{\mathrm{DGLAP}} = x \int \mathrm{d}\beta \, \mathrm{d}\alpha \, f(\beta, \alpha) \, \delta(x - \beta - \alpha\xi) \; .$$

Test with a constant DD:



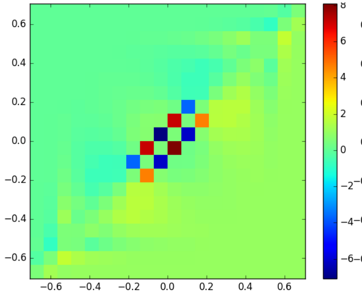
Inverse Radon Transform

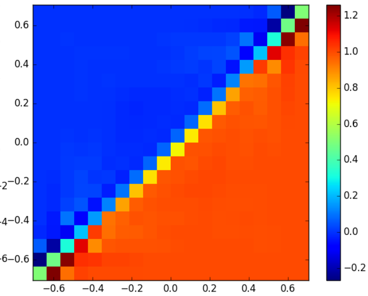
Problem

Find $f(\beta, \alpha)$ on rhombus $\{|\alpha| + |\beta| \leq 1\}$ such that

$$H(x,\xi)|_{\mathrm{DGLAP}} = x \int \mathrm{d}\beta\, \mathrm{d}\alpha\, f(\beta,\alpha)\, \delta(x-\beta-\alpha\xi)\ .$$

Test with a constant DD on half rhombus:





Outline

Computing GPDs

gpdExample()

```
// Lots of includes
#include <src/Partons.h>
...

// Retrieve GPD service
GPDService* pGPDService = ServiceObjectRegistry::getGPDService();
// Load GPD module with the BaseModuleFactory
GPDModule* pGK11Model = ModuleObjectFactory::newGPDModule(
GK11Model::classId);
// Create a GPDKinematic(x, xi, t, MuF, MuR)
GPDKinematic gpdKinematic(0.1, xBToXi(0.001), -0.3, 8., 8.);
// Compute data and store results
GPDResult gpdResult = pGPDService->
computeGPDModelRestrictedByGPDType(gpdKinematic, pGK11Model,
GPDType::ALL);
// Print results
std::cout << gpdResult.toString() << std::endl;

delete pGK11Model;
pGK11Model = 0;
```

(Berthou et al., 2016)

Computing GPDs

gpdExample()

```cpp
// Lots of includes
#include <src/Partons.h>
...

// Retrieve GPD service
GPDService* pGPDService = ServiceObjectRegistry::getGPDService();
// Load GPD module with the BaseModuleFactory
GPDModule* pGK11Model = ModuleObjectFactory::newGPDModule(
GK11Model::classId);
// Create a GPDKinematic(x, xi, t, MuF, MuR)
GPDKinematic gpdKinematic(0.1, xBToXi(0.001), -0.3, 8., 8.);
// Compute data and store results
GPDResult gpdResult = pGPDService->
computeGPDModelRestrictedByGPDType(gpdKinematic, pGK11Model,
GPDType::ALL);
// Print results
std::cout << gpdResult.toString() << std::endl;

delete pGK11Model;
pGK11Model = 0;
```

computeOneGPD.xml

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<scenario id="01" date="" description="Example␣:␣computation␣of␣one␣GPD
␣model␣(GK11)␣without␣evolution">
        <!-- Select type of computation -->
        <task service="GPDService" method="computeGPDModel" >
                <!-- Specify kinematics -->
                <GPDKinematic>
                        <param name="x" value="0.1" />
                        <param name="xi" value="0.00050025" />
                        <param name="t" value="-0.3" />
                        <param name="MuF2" value="8" />
                        <param name="MuR2" value="8" />
                </GPDKinematic>
                <!-- Choose GPD model and set parameters -->
                <GPDModule>
                        <param name="id" value="GK11Model" />
                </GPDModule>
        </task>
</scenario>
```

(Berthou et al., 2016)

Computing GPDs

gpdExample()

```
// Lots of includes
#include <src/Partons.h>
...

// Retrieve GPD service
GPDService* pGPDService = ServiceObjectRegistry::getGPDService();
// Load GPD module with the BaseModuleFactory
GPDModule* pGK11Model = ModuleObjectFactory::newGPDModule(
GK11Model::classId);
// Create a GPDKinematic(x, xi, t, MuF, MuR)
GPDKinematic gpdKinematic(0.1, xBToXi(0.001), −0.3, 8., 8.);
// Compute data and store results
GPDResult gpdResult = pGPDService−>
computeGPDModelRestrictedByGPDType(gpdKinematic, pGK11Model,
GPDType::ALL);
// Print results
std::cout << gpdResult.toString() << std::endl;

delete pGK11Model;
pGK11Model = 0;
```

computeOneGPD.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<scenario id="01" date="" description="Example␣:␣computation␣of␣one␣GPD
␣model␣(GK11)␣without␣evolution">
    <!−− Select type of computation −−>
    <task service="GPDService" method="computeGPDModel" >
        <!−− Specify kinematics −−>
        <GPDKinematic>
            <param name="x" value="0.1" />
            <param name="xi" value="0.00050025" />
            <param name="t" value="-0.3" />
            <param name="MuF2" value="8" />
            <param name="MuR2" value="8" />
        </GPDKinematic>
        <!−− Choose GPD model and set parameters −−>
        <GPDModule>
            <param name="id" value="GK11Model" />
        </GPDModule>
    </task>
</scenario>
```

- Keep it simple.

(Berthou et al., 2016)

Computing GPDs

gpdExample()

```cpp
// Lots of includes
#include <src/Partons.h>
...

// Retrieve GPD service
GPDService* pGPDService = ServiceObjectRegistry::getGPDService();
// Load GPD module with the BaseModuleFactory
GPDModule* pGK11Model = ModuleObjectFactory::newGPDModule(
GK11Model::classId);
// Create a GPDKinematic(x, xi, t, MuF, MuR)
GPDKinematic gpdKinematic(0.1, xBToXi(0.001), -0.3, 8., 8.);
// Compute data and store results
GPDResult gpdResult = pGPDService->
computeGPDModelRestrictedByGPDType(gpdKinematic, pGK11Model,
GPDType::ALL);
// Print results
std::cout << gpdResult.toString() << std::endl;

delete pGK11Model;
pGK11Model = 0;
```

computeOneGPD.xml

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<scenario id="01" date="" description="Example : computation of one GPD
 model (GK11) without evolution">
        <!-- Select type of computation -->
        <task service="GPDService" method="computeGPDModel" >
                <!-- Specify kinematics -->
                <GPDKinematic>
                        <param name="x" value="0.1" />
                        <param name="xi" value="0.00050025" />
                        <param name="t" value="-0.3" />
                        <param name="MuF2" value="8" />
                        <param name="MuR2" value="8" />
                </GPDKinematic>
                <!-- Choose GPD model and set parameters -->
                <GPDModule>
                        <param name="id" value="GK11Model" />
                </GPDModule>
        </task>
</scenario>
```

- Keep it simple.
- Do no reinvent the wheel.

(Berthou et al., 2016)

Computing GPDs

gpdExample()

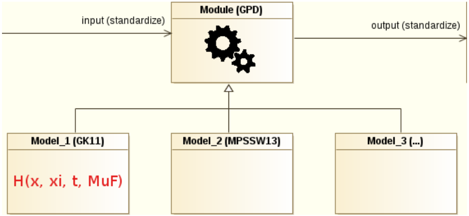
```cpp
// Lots of includes
#include <src/Partons.h>
...

// Retrieve GPD service
GPDService* pGPDService = ServiceObjectRegistry::getGPDService();
// Load GPD module with the BaseModuleFactory
GPDModule* pGK11Model = ModuleObjectFactory::newGPDModule(
GK11Model::classId);
// Create a GPDKinematic(x, xi, t, MuF, MuR)
GPDKinematic gpdKinematic(0.1, xBToXi(0.001), -0.3, 8., 8.);
// Compute data and store results
GPDResult gpdResult = pGPDService->
computeGPDModelRestrictedByGPDType(gpdKinematic, pGK11Model,
GPDType::ALL);
// Print results
std::cout << gpdResult.toString() << std::endl;

delete pGK11Model;
pGK11Model = 0;
```

computeOneGPD.xml

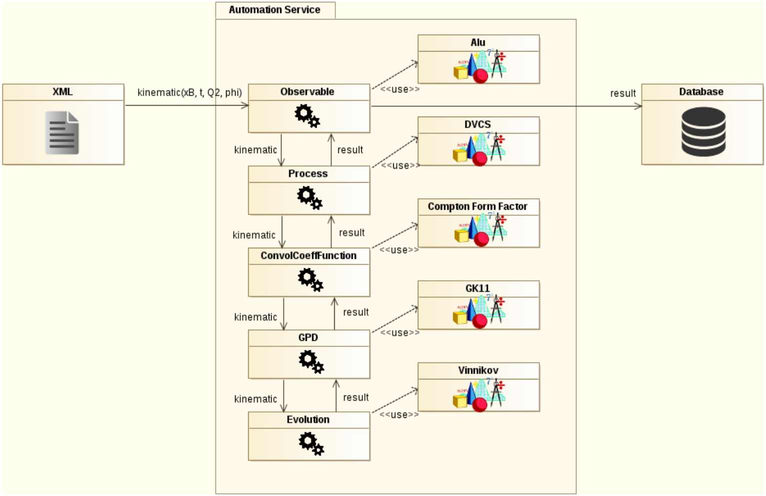
```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<scenario id="01" date="" description="Example␣:␣computation␣of␣one␣GPD
␣model␣(GK11)␣without␣evolution">
        <!-- Select type of computation -->
        <task service="GPDService" method="computeGPDModel" >
                <!-- Specify kinematics -->
                <GPDKinematic>
                        <param name="x" value="0.1" />
                        <param name="xi" value="0.00050025" />
                        <param name="t" value="-0.3" />
                        <param name="MuF2" value="8" />
                        <param name="MuR2" value="8" />
                </GPDKinematic>
                <!-- Choose GPD model and set parameters -->
                <GPDModule>
                        <param name="id" value="GK11Model" />
                </GPDModule>
        </task>
</scenario>
```

- Keep it simple.
- Do no reinvent the wheel.
    - Modularity.

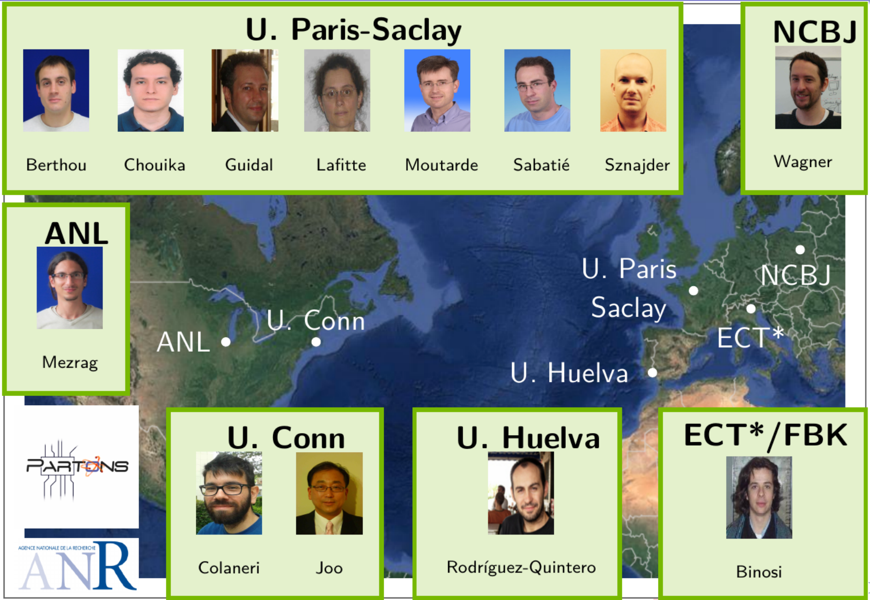
(Berthou et al., 2016)



From GPDs to observables



PARTONS team: multidisciplinary and international!



Summary

- Dyson-Schwinger model for GPDs in progress:

Summary

- Dyson-Schwinger model for GPDs in progress:
  - Inverse Radon Transform to reconstruct the GPD in the entire physical domain.

Summary

- Dyson-Schwinger model for GPDs in progress:
  - Inverse Radon Transform to reconstruct the GPD in the entire physical domain.
  - Move from pion to **proton**.

Summary

- Dyson-Schwinger model for GPDs in progress:
  - Inverse Radon Transform to reconstruct the GPD in the entire physical domain.
  - Move from pion to **proton**.
- PARTONS project near release:

Summary

- Dyson-Schwinger model for GPDs in progress:
  - Inverse Radon Transform to reconstruct the GPD in the entire physical domain.
  - Move from pion to **proton**.
- PARTONS project near release:
  - Validation and testing!

Summary

- Dyson-Schwinger model for GPDs in progress:
  - Inverse Radon Transform to reconstruct the GPD in the entire physical domain.
  - Move from pion to **proton**.
- PARTONS project near release:
  - Validation and testing!
  - Next step: incorporate ab initio GPD model when ready!

Summary

- Dyson-Schwinger model for GPDs in progress:
  - Inverse Radon Transform to reconstruct the GPD in the entire physical domain.
  - Move from pion to **proton**.
- PARTONS project near release:
  - Validation and testing!
  - Next step: incorporate ab initio GPD model when ready!
- Thank you!

Summary

- Dyson-Schwinger model for GPDs in progress:
  - Inverse Radon Transform to reconstruct the GPD in the entire physical domain.
  - Move from pion to **proton**.
- PARTONS project near release:
  - Validation and testing!
  - Next step: incorporate ab initio GPD model when ready!
- Thank you!
  - Any questions?

Bibliography I

A. V. Belitsky, D. Mueller, and A. Kirchner, "Theory of deeply virtual Compton scattering on the nucleon", *Nucl. Phys.* **B629** (2002) 323–392, arXiv:hep-ph/0112108 [hep-ph].

M. Burkardt, "Impact parameter dependent parton distributions and off forward parton distributions for zeta —> 0", *Phys. Rev.* **D62** (2000) 071503, arXiv:hep-ph/0005108 [hep-ph], [Erratum: Phys. Rev.D66,119903(2002)].

M. Diehl, "Generalized parton distributions", *Phys. Rept.* **388** (2003) 41–277, arXiv:hep-ph/0307382 [hep-ph].

P. Maris and C. D. Roberts, "π- and K-meson Bethe-Salpeter amplitudes", *Phys. Rev. C* **56** Dec (1997) 3369–3383.

G. Eichmann, "Hadron Properties from QCD Bound-State Equations", *PhD Thesis*, 2009 arXiv:0909.0703 [hep-ph].

Bibliography II

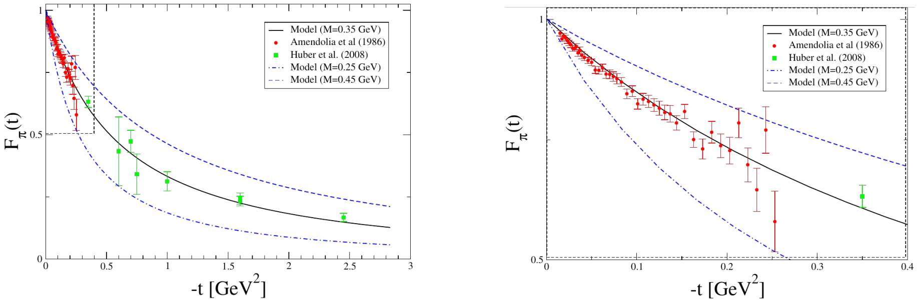
C. Mezrag, "Generalised Parton Distributions : from phenomenological approaches to Dyson-Schwinger equations", PhD thesis, IRFU, SPhN, Saclay, 2015.

A. V. Belitsky, D. Mueller, A. Kirchner, and A. Schafer, "Twist three analysis of photon electroproduction off pion", *Phys. Rev.* **D64** (2001) 116002, `arXiv:hep-ph/0011314` [hep-ph].

H. Moutarde, "Nucleon Reverse Engineering: Structuring hadrons with colored degrees of freedom", 2015.

B. Berthou, D. Binosi, N. Chouika, M. Guidal, C. Mezrag, H. Moutarde, F. Sabatié, P. Snajder, and J. Wagner, "PARTONS: **PAR**tonic **T**omography **Of** **N**ucleon **S**oftware: A computing platform for the phenomenology of Generalized Parton Distributions", *Eur. Phys. J.* **C** (2016) `arXiv:1512.06174` [hep-ph].

Form factors for the pion



(Mezrag, 2015)

Quark Dyson-Schwinger equation (Gap Equation)

- Quark Propagator:

$$S(p)^{-1} = A\left(p^2\right)\left(i\gamma\cdot p + M\left(p^2\right)\right)\ , \tag{5}$$

with $M\left(p^2\right)$ the dynamical mass of the quark.

- Quark Gap equation (represented in Page 8):

$$S(p)^{-1} = Z_2\left(i\gamma\cdot p + m_b\right) \tag{6}$$

$$+ \quad Z_1 \int^{\Lambda} \frac{\mathrm{d}^4 q}{(2\pi)^4}\, g^2\, D_{\mu\nu}(p-q)\, \frac{\lambda^a}{2}\gamma_\mu\, S(q)\, \frac{\lambda^a}{2}\Gamma_\nu(q,p)\,. \tag{7}$$

- Truncation for the Vertex. Example (Rainbow-Ladder truncation):

$$\Gamma^{RL}_{\mu}(q,p) = \gamma_\mu\,. \tag{8}$$

- Must choose a (phenomenological) model for the Gluon Propagator in the infra-red (ultra-violet domain reproduces perturbative QCD). Example of effective interaction (Maris-Tandy):

$$\mathcal{G}^{MT}_{IR}\left(k^2\right) = \frac{4\pi^2}{\omega^6} D\, k^2 e^{-\frac{k^2}{\omega^2}}\ . \tag{9}$$

Dynamical mass for the quark

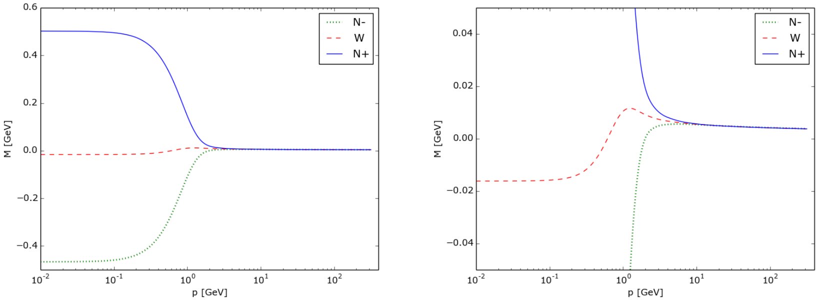


Figure: **Dynamical mass** M function of $\sqrt{p^2}$ (in GeV). Numerical resolution of the Quark Dyson-Schwinger equation on the real axis. In blue (solid line), the positive Nambu solution displays dynamical mass generation: large constituent-quark mass at small momenta, small current-quark mass at large momenta. In red (dashed), solution with no mass generated; in green (dots), negative Nambu solution.