Poopćene partonske distribucije

Filip Bilandžija
Prirodoslovno matematički fakultet,
Fizički odsjek, Sveučilište u Zagrebu,
Bijenička cesta 32, 10000, Zagreb

Mentor: prof. dr. sc. Krešimir Kumerički

(Dated: Siječanj 2023)

Krenuvši od problema strukture materije, objašnjeno je današnje viđenje tog problema, posebno za protone, te su uvedene poopćene partonske distribucije (GPD) koje nam daju uvid u kompleksnu strukturu protona. Ukratko su objašnjeni eksperimenti te njihove observable koje nam daju uvid u GPD-ove. Objašnjeno je uvođenje jednadžbe evolucije te uvođenje konformnih momenata i njihovih inverza kako bi se ta jednadžba evolucije pojednostavila. Potom je pojašnjena numerička implementacije inverzne transformacije te je, naposljetku, objašnjena optimizacija dane numeričke transformacije.

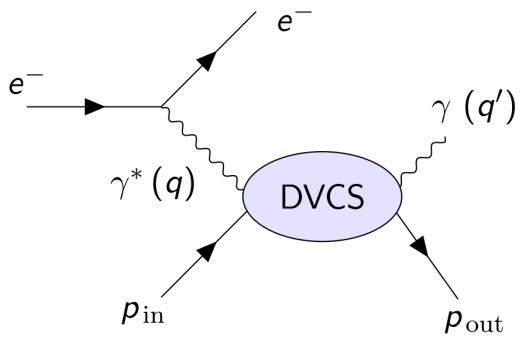
I. UVOD

Još od starogrčkih filozofa poput Demokrita ili Aristotela, pitanje od čega se sastojimo predmet je rasprave. Krajem 19. st. pa do danas postoji cijeli niz eksperimentalnih istraživanja ne bi li razriješio to pitanje. No, ono što povezuje sva ta istraživanja je jedna činjenica - što naša proba (foton, elektron ili nešto drugo) ima veću energiju, to ona ima bolje razlučivanje.

Današnje shvaćanje pitanja strukture materije opisano je standardnim modelom čestica. On se sastoji od 12 fermiona spina $1/2$, koji čine tvar, 4 baždarna bozona spina 1 koji su prijenosnici sile, te 1 skalaranog bozona (spina 0) - Higgsovog bozona - bitnog zbog problema masa čestica. Fermioni se dijele na 6 leptona, koji se mogu propagirati slobodno, i 6 kvarkova, koji obično bivaju zatočeni u hadronima - vezanim stanjima 2 ili 3 kvarka.[1] Zatočenje kvarkova posljedica je jake sila koju opisuje kvantna kromodinamika (engl. quantum chromodynamics, QCD). To je povezano s činjenicom da je QCD u režimu niskih energija neperturbativna teorija, što sprječava proučavanje hadrona izravnim putem te se tom problema mora pristupiti indirektno.

U drugoj polovici 20. st., tadašnji znanstvenici proučavali su interakciju elektrona s protonom. Naime, kvantna elektrodinamika, koja opisuje ovu reakciju, bila je dovoljno dobro poznata te je time testirana njena ispravnost. U tom eksperimentu, na nižim energijama elektroni su vidjeli proton kao točkastu česticu. No povećanjem energije elektrona, vidjelo se da proton zapravo nije elementarna čestica, nego se sastoji od 3 čestice spina $1/2$ - kvarkova. Daljnjim povećanjem energije elektrona i jakosti interakcije

[1] U ovoj raspravi zanemarujemo egzotična stanja poput tetrakvarkova ili pentakvarkova.



Slika 1: Prikaz duboko virtualnog komptonskog raspršenja. Elektron interagira s hadronom kroz virtualni foton, a u izlaznom kanalu, uz elektron i hadron, imamo i realni foton. Posuđeno iz [1].

u eksperimentu, proton se čini da je napravljen ne samo od tri valentna kvarka, nego da sadrži i more kvarkova i gluona. Dakle, struktura protona je puno bogatija i kompleksnija nego što se to ranije mislilo.

Iz tih eksperimenata, bilo je moguće izvući nukleonske form faktore i partonske distribucijske funkcije (engl. Parton Distribution Functions, PDF). Nukleonski form faktori predstavljaju transverzalnu raspodjelu naboja u nukleonu, dok su partonske distribucijske funkcije one koje daju raspodjelu longitudinalnog impulsa partona u nukleonu. No, ni nukleonski form faktori i PDF-ovi zajedno nam ne daju punu kvalitativnu sliku raspodjele partona u proton, tj. samu 3D sliku protona, te su zbog toga uvedene poopćene partonske distribucije (engl. Generalized Parton Distributions, GPD).

Poopćene partonske distribucije mogu se proučavati kroz nekoliko duboko ekskluzivnih procesa poput dubokog virtualnog komptonskog raspršenja (engl. Deeply Virtual Compton Scattering, DVCS) ili duboko virtualne mezonske produkcije (engl. Deeply Virtual Meson Production, DVMP). Kroz takve procese dobivamo pristup informacijama poput raspodjele transverzalnog impulsa, angularnog momenta i tlaka u protonu, koje nismo prethodno mogli mjeriti. Duboko virtualno komptonsko raspršenje, prikazano na slici 1, je najjednostavniji takav proces, a u kojem imamo virtualni foton u ulaznom kanalu te realni foton (koji naš eksperimentalni postav može mjeriti) u izlaznom. Duboko virtualna mezonska produkcija je sličan proces, ali u njemu, umjesto fotona, imamo mezon u izlaznom kanalu. Mjereći DVCS proces imamo direktnu vezu s komptonskim form faktorima, koji su konvolucije GPD-ova i poznatih perturbativnih amplituda te time imamo način na koji možemo testirati modele GPD-ova. [2] [1, 2]

GPD očigledno sadrži više informacija od PDF-a pa zbog toga i ovisi o više varijabli: x, η, t i Q^2. x je definiran kao $x = \frac{k^+ + k'^+}{2P^+}$, gdje su k i k' longitudinalni impulsi emitiranog i apsorbiranog kvarka, što odgovara frakciji longitudinalnog impulsa. η, varijabla asimetrije (engl. skewness), definirana je kao $\eta = -\frac{\Delta^+}{P^+}$, što otprilike odgovara izmjeni impulsa, dok je t Mandelstanova varijabla. Ovisnost o Q^2, izmjeni energije, je relativno slaba pa se često i ne piše. U svim izrazima je korištena konvencija $P = (p + p')/2$ i $\Delta = p - p'$, a varijable su izražene preko koordinata svjetlosnog stošca: $a^{\pm} = \frac{1}{\sqrt{2}}(a^0 \pm a^3)$

[2] Za DVMP postoje analogne funkcije koje se zovu prijelazni form faktori (engl. transition form factors).

i $a = (a^+, \boldsymbol{a}, a^-)$. [3–5]

GPD-ovi, primjerice u nepolariziranom kvarkovskom sektoru, definirani su kao:

$$F^q = \frac{1}{2}\int \frac{\mathrm{d}z^-}{2\pi} e^{i\pi P^+ z^-} \left\langle p' \right| \bar{q}\left(-z\right)\gamma^0 q\left(z\right)\left|p\right\rangle \Big|_{z^+=0,\boldsymbol{z}=0}, \tag{1}$$

što možemo interpretirati kao vjerojatnost da proton prijeđe iz stanja impulsa p u stanje impulsa p' emisijom i reapsorpcijom kvarka. [2]

Kako je ovisnost o Q^2 dobro poznata, možemo perturbativno korigirati GPD-ove. Razvojem po Q^2 možemo ispitivati i dobiti više informacija o ovisnosti GPD-a o drugim varijablama. Jednadžba evolucije u vodećem redu je:

$$Q\frac{\mathrm{d}}{\mathrm{d}Q}F^q(x,\eta,\Delta^2,Q^2) = -\frac{\alpha_s(Q)}{2\pi}\gamma^{(0)} \otimes F^q(x,\eta,\Delta^2,Q^2). \tag{2}$$

gdje je operator konvolucije definiran kao:

$$\gamma^{(0)} \otimes F^q(x,\eta,\Delta^2,Q^2) \equiv \int_{-\infty}^{\infty} \frac{\mathrm{d}y}{|\eta|}\gamma^{(0)}\left(\frac{x}{\eta},\frac{y}{\eta}\right)F^q(y,\eta,\Delta^2,Q^2), \tag{3}$$

Evolucijski kernel $\gamma^{(0)}$ pritom je dan izrazom:

$$\gamma^{(0)}(x,y) = \left[\Theta(x,y)\frac{1+x}{1+y}\left(1+\frac{2}{y-x}\right) + \Theta(-x,-y)\frac{1-x}{1-y}\left(1+\frac{2}{x-y}\right)\right]_+, \tag{4}$$

gdje +-znak u indeksu zagrade označava izraz $\left[K(x,y)\right]_+ = K(x,y) - \delta(x-y)\int \mathrm{d}z K(z,y)$, a $\Theta(x,y)$ je skraćena notacija za izraz:

$$\Theta(x,y) = \mathrm{sign}(1+y)\,\theta\left(\frac{1+x}{1+y}\right)\theta\left(\frac{y-x}{1+y}\right). \tag{5}$$

Iz jednadžbi (2)-(5), evidentno je da se radi o potpuno netrivijalnoj konvoluciji GPD-a s evolucijskim kernelom $\gamma^{(0)}$. Iako se načelno ovakav račun može numerički izvesti, on iziskuje rješavanje dosta tehničkih poteškoća. Ono čime se može doskočiti tom problemu je uvođenje konformnih momenata. Naime, evolucijski kernel u jednadžbi (2) je dijagonalan u bazi Gegenbauerovih polinoma, te stoga u priču uvodimo konformne momente. [6–8]

II. KONFORMNI MOMENTI

Konformni momenti GPD-a definirani su relacijom:

$$F_n(\eta,t) = \int_{-1}^{1} \mathrm{d}x c_n(x,\eta)F(x,\eta,t), \tag{6}$$

gdje je $F(x,\eta,t)$ GPD, dok je c_n dan izrazom:

$$c_n(x,\eta) = \eta^n \frac{\Gamma(3/2)\Gamma(1+n)}{2^n\Gamma(3/2+n)} C_n^{3/2}\left(\frac{x}{\eta}\right), \tag{7}$$

kod kojeg $C_n^{3/2}$ označava Gegenbauerov polinom. Uvođenjem konformnih momenata, jednadžba evolucije se pojednostavljuje na:

$$Q\frac{\mathrm{d}}{\mathrm{d}Q}F_j^q(\eta,t,Q^2) = -\frac{\alpha_s(\mu)}{2\pi}\gamma_j^{(0)}F_j^q(\eta,t,Q^2). \tag{8}$$

U ovoj jednadžbu više nemamo netrivijalnu desnu stranu na kojoj smo imali konvoluciju, nego sada imamo jednostavno množenje. Dakle, u j-prostoru puno je jednostavnije provesti evoluciju GPD-a nego u x-prostoru. No, to dolazi s cijenom jer u j-prostoru GPD-ove je teško interpretirati. Naime, važna fizikalna svojstva GPD-ova vidljiva su tek u x-prostoru: GPD je u x-prostoru povezan s raspodjelom kvarkova po momentu i položaju (imamo 3D sliku protona), poznavanje GPD-ova u x-prostoru rješava problem spina protona, u x-prostoru GPD-ovi daju pristup matričnim elementima tenzora energije i impulsa poput tlaka u protonu. Iz tih razloga, važno nam je invertirati operaciju (6), te time omogućiti transformaciju iz j-prostora nazad u x-prostor.

Može se pokazati [6] da je problem inverza riješen uvođenjem analitičkog produljenja Gegenbauerovih polinoma p_j:

$$p_j(x,\eta) = \theta(\eta-|x|)\eta^{-j-1}\mathscr{P}_j\left(\frac{x}{\eta}\right) + \theta(x-\eta)\eta^{-j-1}\mathscr{Q}_j\left(\frac{x}{\eta}\right) \tag{9}$$

gdje su $\mathscr{P}_j$ i $\mathscr{Q}_j$ definirani kao:

$$\mathscr{P}_j(x) = \frac{2^{j+1}\Gamma(5/2+j)}{\Gamma(1/2)\Gamma(1+j)}(1+x)_2F_1\left(-j-1,j+2,2;\frac{x+1}{2}\right) \tag{10}$$

i

$$\mathscr{Q}_j(x) = -\frac{\sin(\pi j)}{\pi}x^{-j-1}{}_2F_1\left(\frac{j+1}{2},\frac{j+2}{2},\frac{5}{2}+j;\frac{1}{x^2}\right) \tag{11}$$

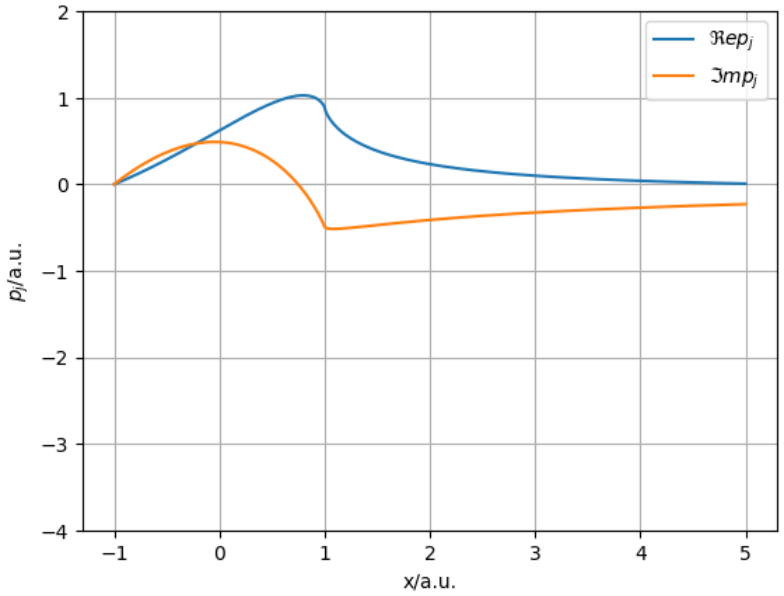
Inverz je onda dan tzv. Mellin-Barnes (MB) integralnom transformacijom:

$$F(x,\eta,t) = \frac{1}{2i}\int_{c-i\infty}^{c+i\infty}\mathrm{d}j\frac{1}{\sin(\pi j)}p_j(x,\eta)F(\eta,t). \tag{12}$$

III. NUMERIČKA IMPLEMENTACIJA I OPTIMIZACIJA

A. Implementacija hipergeometrijske funkcije

Kao što se i vidi iz izraza (10) i (11), definicije za $\mathscr{P}_j$ i $\mathscr{Q}_j$ u sebi sadrže hipergeometrijsku funkciju ${}_2F_1$. Iako postoje njene numeričke implementacije u Python-u, one obuhvaćaju samo realne parametre, dok naš račun iziskuje i rad s kompleksnim parametrima - što se najbolje vidi iz izraza (12). Stoga, prije bilo kojeg daljnjeg računa, bilo je potrebno napraviti vlastitu implementaciju hipergeometrijske funkcije koja će podržavati rad i s kompleksnim parametrima. Nasreću, našli smo potrebnu funkciju napisanu u C-u [9] te je prepisali i prilagodili u Python-u.



Slika 2: Graf funkcije p_j za njen realni (plava linija) i imaginarni (narančasta linija) dio u ovisnosti o x za $j = -1/4 + i/2$ i $\eta = 1$.

Algoritam počiva na računu hipergeometrijskog reda:

$$ {}_2F_1(a,b,c;z) = 1 + \frac{ab}{c}\frac{z}{1!} + \frac{a(a+1)b(b+1)}{c(c+1)}\frac{z^2}{2!} + \ldots \tag{13} $$

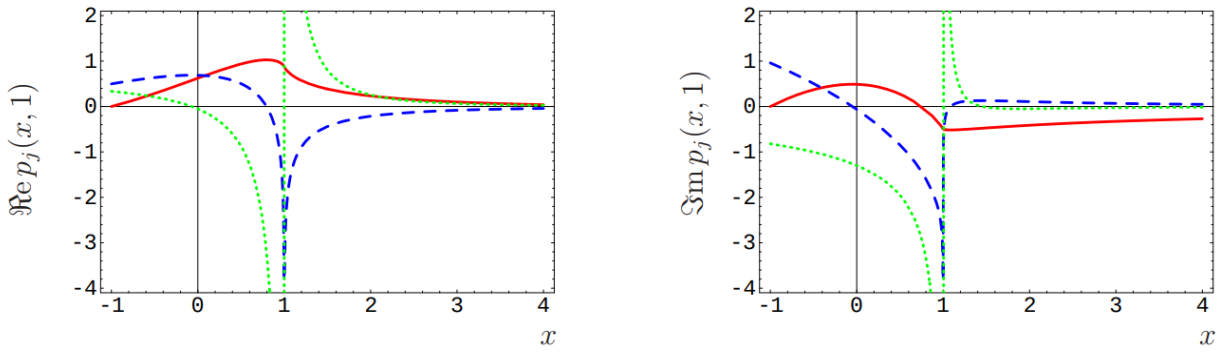
Ukoliko je apsolutna vrijednost željene točke z_1 manja ili jednaka 0.5, red dovoljno brzo konvergira (red općenito konvergira za $|z_1| < 1$) da vrijednost hipergeometrijske točke možemo izračunati samo računajući red. U suprotnom, prvo izabiremo početnu točku z_0 (koja ovisi o željenoj točki z_1), zatim određujemo vrijednost hipergeometrijske funkcije i vrijednost njene derivacije u toj točki, te naposljetku integriramo put od točke z_0 do točke z_1 pomoću diferencijalnih jednadžbi:

$$ \begin{aligned} \frac{\mathrm{d}F}{\mathrm{d}s} &= (z_1 - z_0)F', \\ \frac{\mathrm{d}F'}{\mathrm{d}s} &= (z_1 - z_0)\left(\frac{abF - [c-(a+b+1)z]F'}{z(1-z)}\right), \end{aligned} \tag{14} $$

gdje je F hipergeometrijska funkcija, F' njena derivacija, a z funkcija dana kao $z(s) = z_0 + s(z_1 - z_0)$, gdje s ide od 0 do 1. Integracijom gornjih diferencijalnih jednadžbi dobivamo vrijednosti hipergeometrijske funkcije na cijeloj kompleksnoj ravnini.

B. Implementacija $p_j(x,\eta)$

Nakon što smo uspješno implementirali kod za računanje hipergeometrijske funkcije i testirali ga na više točaka, mogli smo implementirati funkciju oko koje nam se vrti cijela transformacija, funkciju



Slika 3: Realni (lijevo) i imaginarni (desno) dio funkcije p_j u ovisnosti o x za $j = -1/4 + i/2$ i $\eta = 1$. Posuđeno iz [6].

p_j definiranu izrazom (9). Njena implementacija je bila relativno jednostavna te smo nakon toga mogli prekontrolirati radi li ona očekivano tako da smo nacrtali graf realnog i imaginarnog dijela za konkretne vrijednosti $j = -1/4 + i/2$ i $\eta = 1$ (slika 2) te usporedili s poznatim rezultatom prikazanom na slici 3. Iz usporedba ove dvije slike, vidi se da naš algoritam za funkciju p_j radi kao što smo i očekivali.

C. Numerički Mellin-Barnes integral

Za samu transformaciju iz j-prostora u x-prostor, potreban nam je integral (12). Taj integral ide po liniji paralelnoj s imaginarnom osi kompleksne ravnine. Za numeričko izvrednjavanje to je potrebno pretvoriti u normalni realni integral. Korištenjem Schwartzovog principa refleksije ($f(\bar{z}) = \overline{f(z)}$), integral (12) moguće je pretvoriti u integral po samo polovici konture iznad realne osi:

$$\frac{1}{2i}\int_{c-i\infty}^{c+i\infty} dj\, f(j) = \mathrm{Im}\left(e^{i\phi}\int_0^{\infty} dy\, f(j = c + ye^{i\phi})\right) \tag{15}$$

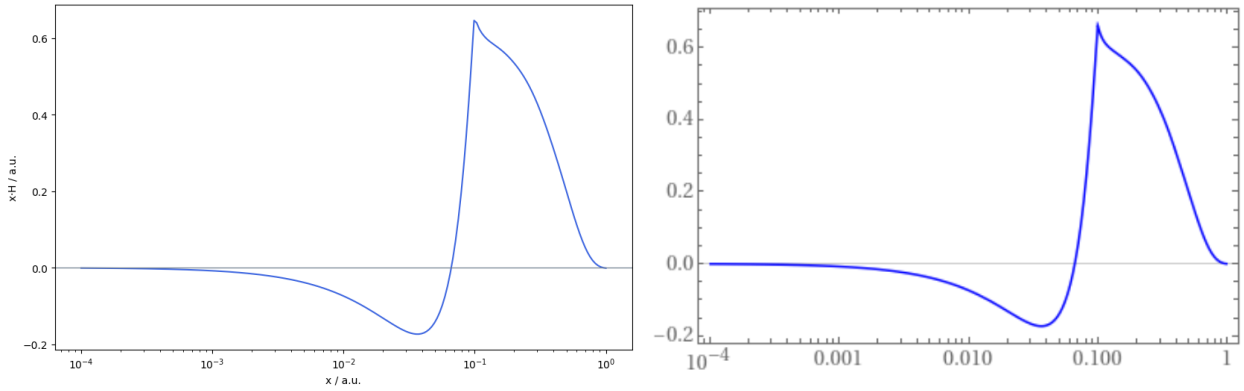
Nekad je radi brže konvergencije integrala pogodno konturu integracije, parametriziranu kao $j = c + ye^{i\phi}$ nagnuti od imaginarne osi kao na slici, no za početak mi smo fiksirali $\phi = \pi/2$ i $c = 0.35$ te tako implementirali ovaj integral. Za prve testove koristili smo jednostavnu GPD funkciju u j-prostoru oblika

$$H_j = 2\frac{B(1/2+j,4)}{B(1/2,4)} = 2\frac{(1/2)_4}{(1/2+j)_4}, \tag{16}$$

te rezultate integracija usporedili s već poznatim rezultatima, koji su se slagali.

D. Pretvorba GPD-ija iz j-prostora u x-prostor

Na kraju, ostalo je samo provesti punu transformaciju funkcije (16) iz j-prostora u x-prostor za fiksni $\eta = 0.1$. Rezultat takvog postupka vidljiv je na slici 4a. Na horizontalnoj osi nalazi se x,



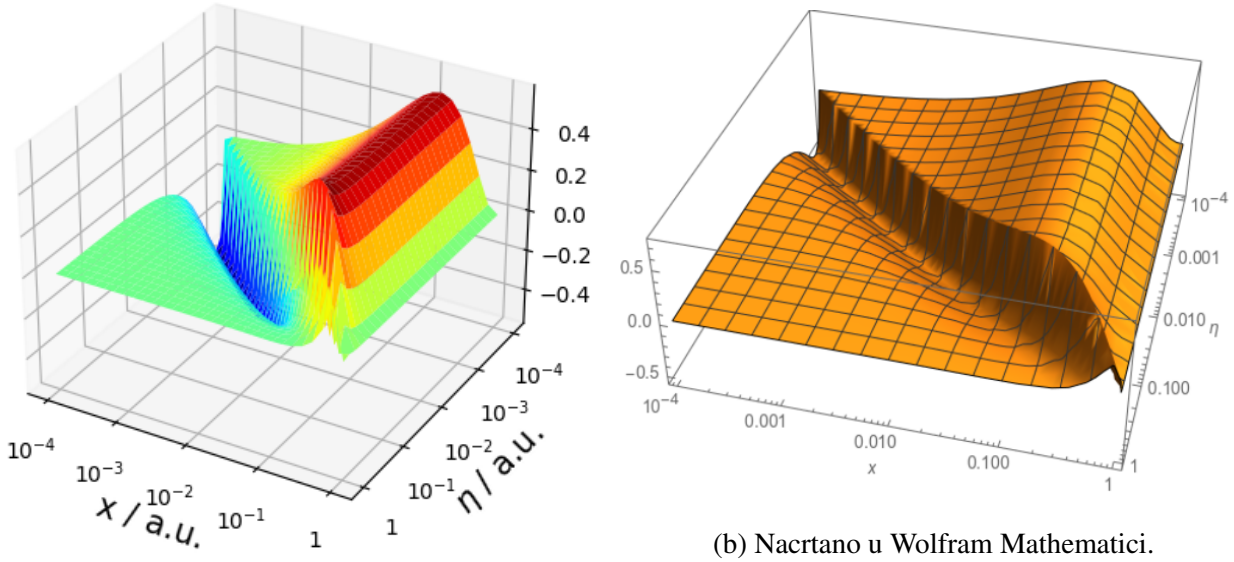
(a) Nacrtano u Pythonu.

(b) Nacrtano u Wolfram Mathematici.

Slika 4: Graf GPD test funkcije H_j (16) u x-prostoru za $\eta = 0.1$ u ovisnosti o x.

dok se na vertikalnoj nalazi umnožak $x \cdot H$, što je standardan prikaz zbog činjenice da GPD-ovi imaju tendenciju divergirati na niskim vrijednostima. Takav rezultat se slaže s dobivenim prijašnjim rezultatima prikazanim na slici 4b.

Nadalje, fiksni η zamijenili smo ovisnošću o η i time dobili 3D graf test funkcije (16) o x i η, koji je prikazan na slici 5a. Taj se graf u potpunosti slaže s identičnim postupkom koji smo dobili u Wolfram Mathematici koji je prikazan na slici 4b.



(a) Nacrtano u Pythonu.

(b) Nacrtano u Wolfram Mathematici.

Slika 5: 3D graf funkcije GPD test funkcije H_j (16) u x-prostoru u ovisnosti o x i η.

E. Optimizacije

Prvotni kod za sliku 4a izvršavao se oko 2 i pol minute, dok se kod za sliku 5a izvršavao čak 30 minuta. Dakako, to je bilo predugo te se kod trebao optimizirati kako bi se mogao koristiti u stvarne istraživačke svrhe. Prvotni je kod bio potpuno sekvencijalan (račun se izvršavao za svaku točku i međukorak pojedinačno) te su korištene funkcije `quad` za numeričku integraciju i `odeint` za rješavanje diferencijalnih jednadžbi iz `scipy` paketa za Python. Te su točke označene kao one koje imaju najveći potencijal za napredak, pa smo istražili opcije koje su nam bile dostupne kako bi ih unaprijedili.

1. Korištenje numpy matrica

Kako bismo doskočili prvom problemu, iskoristili smo svojstvo `numpy` paketa da ima korutine napisane u C-u za račun s matricama, koje su znatno brže nego ekvivalentan kod u Python-u. Time smo uspjeli vektorizirati kod jer se više kod nije izvršavao za svaku točku zasebno, nego je izvršavao se na vektoru ili matrici što ga je znatno ubrzalo. Naravno, nije paralelizacija koda svugdje pomogla. Na primjer, pri računanju izraza (9), račun s matricama bi prvo izračunao funkcije $\mathscr{P}_j$ i $\mathscr{Q}_j$ za sve točke u matrici, a tek onda primijenio θ funkciju, što je rezultiralo usporavanjem koda jer se račun vodio za apsolutno sve vrijednosti umjesto samo za neke. Kod takvih primjera, ostavili smo sekvencijalan račun te tako maksimizirali performanse našeg koda.

2. Integracije kvadraturom

Sljedeća točka je bila optimizacija numeričke integracije za izraz (12). Funkcija koju smo prvotno koristili bila je `quad` iz `scipy` paketa i ona je namijenjena za širok spektar problema zbog čega je ona popularan i moćan alat. No upravo zbog svoje široke primjene, ona ne posjeduje performanse koje bi posjedovao kod specijaliziran za računanje samo jednog integrala.

U našem smo kodu stoga integraciju pomoću `quad` funkcije s integracijom Gauss–Legendrovom kvadraturom. Ona svodi računanje integrala na sumu po točkama s različitim težinama. Kako naš integrand brzo oscilira blizu nule, raspodijelili smo područja integracije na 12 intervala danih sa sljedećim točkama: [0., 0.01, 0.08, 0.15, 0.3, 0.5, 1.0, 1.5, 2.0, 4.0, 6.0, 8.0, 10.0]. U svakom intervalu zatim provodimo Gauss–Legendrovu kvadraturu te na kraju sve sumiramo i dobivamo vrijednost našeg integrala. Time je naš kod bio optimiziran za računanje našeg željenog integrala.

3. Rješavanje diferencijalne jednadžbe Runge-Kutta metodom

Zadnji korak kod optimizacije bio je zamjena funkcije `odeint` iz `scipy` paketa za računanje diferencijalnih jednadžbi sa znatno rudimentarnijim kodom. Slično kao i s funkcijom `quad`, funkcija `odeint`

je namijenjena da daje precizan rezultat u najraličitijim situacijama, što je i prednost, ali i mana. Glavni je problem to što `odeint` obično ima testiranja svoje preciznosti svaki korak i nakon toga prilagođuje rezoluciju svoje mreže točaka danoj situaciji, što je usporava. Stoga, smo tu funkciju zamijenjili vlastitom funkcijom za računanje diferencijalnih jednadžbi temeljenoj na Runge-Kutta metodi 4. stupnja.

Runge-Kutta metoda je iterativna metoda koja se koristi za rješavanja problema početne vrijednosti, ali je također i prilagodljiva za dani problem mijenjanjem duljine koraka, što je čini povoljnom za optimizaciju. Eksperimentirajući s raznim duljinama koraka, značajno smo skratili vrijeme izvršavanja koda, a da pritom gotovo da nismo izgubili preciznost.

Ukupan efekt svih naših optimizacija je bilo značajno ubrzavanje izvršavanja našeg koda. Za sliku 4a za koju nam je prvotno trebalo oko dvije i pol minute, nakon optimizacije trebalo je oko 7 sekundi. Za sliku 5a, efekt je bio još drastičniji. Umjesto prvotnih 30-ak minuta, za izvršavanje koda nakon svih optimizacija bilo je potrebno tek 20-ak sekundi.

IV. ZAKLJUČAK

U ovome smo seminaru izložili problem strukture hadrona i generaliziranih partonskih distribucija kao njegovog rješenja, te iznijeli fizikalna svojstva GPD-ova te problem rada s njima. Zatim smo se posvetili problemu transformacije GPD-ova iz prostora konformnih momenata (j-prostora) u prostor frakcije longitudinalnih impulsa (x-prostora) te numeričkoj implementaciji rješenja tog problema. Nadalje, u numeričkoj implementaciji identificirali smo potencijalne točke koje "koče" brzinu izvršavanja računa te naveli i istražili moguće optimizacije tih točaka. Na kraju, ukupan efekt tih optimizacija drastično je smanjio vrijeme izvršavanja računa te time je ova numerička implementacija postala moguće sredstvo koje će se koristiti u daljnjim istraživanjima u ovome području.

[1] C. Mezrag, An introductory lecture on generalised parton distributions, Few-Body Systems **63**, 10.1007/s00601-022-01765-x (2022).

[2] K. Kumerički, S. Liuti, and H. Moutarde, GPD phenomenology and DVCS fitting, The European Physical Journal A **52**, 10.1140/epja/i2016-16157-3 (2016).

[3] S. Horvat, Funkcije strukture protona (2019).

[4] M. Cvitković, *Proučavanje duboko virtualnog komptonskog raspršenja pomoću strojnog učenja*, Master's thesis, Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet (2020).

[5] I. Ćorić, *Istraživanje kvarkovsko-gluonske strukture protona pomoću strojnog učenja*, Master's thesis, Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet (2019).

[6] D. Müller and A. Schäfer, Complex conformal spin partial wave expansion of generalized parton distributions and distribution amplitudes, Nuclear Physics B **739**, 1 (2006).

[7] A. Belitsky and A. Radyushkin, Unraveling hadron structure with generalized parton distributions, Physics Reports **418**, 1 (2005).

[8] V. Bertone, H. Dutrieux, C. Mezrag, J. M. Morgado, and H. Moutarde, Revisiting evolution equations for generalised parton distributions, The European Physical Journal C **82**, 10.1140/epjc/s10052-022-10793-0 (2022).

[9] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed. (Cambridge University Press, Cambridge, USA, 1992).