

České vysoké učení technické v Praze  
Fakulta jaderná a fyzikálně inženýrská

Katedra dozimetrie a aplikace ionizujícího záření  
Obor: Jaderné inženýrství  
Zaměření: Dozimetrie a aplikace ionizujícího záření



**Simulace femtoskopických korelací  
částic  
Study of femtoscopic particle  
correlations**

DIPLOMOVÁ PRÁCE

Vypracoval: Ivan Horňák  
Vedoucí práce: RNDr. Jan Smolík, Ph.D.  
Rok: 2013

### **Poděkování**

Chtěl bych poděkovat RNDr. Janu Smolíkovi Ph.D. za odbornou pomoc a rady, které pomohly ke vzniku této práce.

Ivan Hornák

## **Prohlášení**

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze literaturu a publikace uvedené v příloženém seznamu.

Nemám závažný důvod proti použití tohoto školního díla ve smyslu §60 Zákona č.121/1200Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne .....

.....  
Ivan Horňák

*Název práce:* **Simulace femtoskopických korelací částic**

*Autor:* Ivan Hornák

*Obor:* Jaderné inženýrství

*Druh práce:* Diplomová práce

*Vedoucí práce:* RNDr. Jan Smolík, Ph.D.

Katedra dozimetrie a aplikace ionizujícího záření, Fakulta jaderná a fyzikálně inženýrská, České vysoké učení technické v Praze

*Abstrakt:*

Korelační funkce je významným nástrojem částicové fyziky a je používána při mnoha příležitostech. Femtoskopie patří mezi její nejdůležitější aplikace. Je také používána při experimentu DIRAC v CERNu, který ověřuje teorii QCD pro oblast nízkých energií určením doby života atomů  $A_{2\pi}$  a  $A_{\pi K}$

Tato diplomová práce se zabývá korelační funkcí, její teorií. Dále byly simulované korelační funkce za vlivu silné interakce, kvantové statistiky a za vlivu kompletního FSI a kvantové statistiky. Nakonec byla zpracována data.

*Klíčová slova:* korelační funkce, femtoskopie, korelace pionů, FSI, kvantová statistika

*Název práce:* **Study of femtoscopic particle correlations**

*Author:* Ivan Hornák

*Abstract:* Correlation function is a powerful tool of particle physics, which is used on in many occasions. Femtoscopy belongs among the most important applications. It is also used in DIRAC experiment in CERN, which verifies QCD theory at low energy region by lifetime determination of  $A_{2\pi}$  and  $A_{\pi K}$  atoms.

This thesis deals with correlation function, with its theory. It contains simulation of correlation function under influence of strong interaction, quantum statistics and complete FSI with quantum statistic. It also contains data elaboration.

*Key words:* correlation function, femtoscopy, pion correlation, FSI, quantum statistics

# Obsah

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Úvod</b>  | <b>8</b>  |
| <b>2</b> | <b>Popis systému dvou částic a chaotičnost zdroje</b>                  | <b>9</b>  |
| 2.1      | Popis jedné částice . . . . .  | 10        |
| 2.1.1    | Šíření ze zdroje . . . . .   | 10        |
| 2.1.2    | Produkční amplituda a fázový faktor . . . . .                          | 11        |
| 2.1.3    | Rozdělení pravděpodobnosti . . . . .                                   | 11        |
| 2.2      | Popis dvoučásticového systému . . . . .                                | 12        |
| 2.3      | Rozdělení hybnosti pro dvě částice . . . . .                           | 14        |
| 2.4      | Chaotický zdroj . . . . .  | 16        |
| 2.5      | Koherentní zdroj . . . . .   | 18        |
| <b>3</b> | <b>Korelace, kvantová statistika a FSI</b>                             | <b>22</b> |
| 3.1      | Formalismus . . . . .  | 22        |
| 3.2      | Neinteragující neidentické částice: Časoprostorová koherence . . . . . | 25        |
| 3.3      | Neinteragující identické částice: QS korelace . . . . .                | 25        |
| 3.3.1    | Předpoklad hladkosti . . . . .   | 26        |
| 3.4      | Interagující částice: FSI korelace . . . . .                           | 27        |
| 3.4.1    | Předpoklad současnosti . . . . .                                       | 28        |
| 3.4.2    | Simulace korelační funkce . . . . .                                    | 29        |
| <b>4</b> | <b>Monte Carlo</b>   | <b>31</b> |
| 4.1      | Dostatečné navzorkování . . . . .                                      | 31        |
| <b>5</b> | <b>Program</b>   | <b>33</b> |
| 5.1      | Simulace podmínek experimentu . . . . .                                | 33        |

|          |  |           |
|----------|--|-----------|
| 5.2      | Tvorba korelační funkce . . . . .  | 35        |
| 5.3      | Histogramy . . . . .   | 36        |
| 5.4      | Root a GSL . . . . .   | 37        |
| 5.5      | Generátory náhodných čísel . . . . .                                     | 38        |
| 5.6      | Analýza kódu . . . . .   | 38        |
| 5.6.1    | Hlavičkové soubory, definice konstant, pomocné funkce . . . . .          | 38        |
| 5.6.2    | Struktura na uložení vlastností jedné částice . . . . .                  | 39        |
| 5.6.3    | Třída <i>Pripad</i> . . . . .  | 40        |
| 5.6.4    | Konstruktor třídy <i>Pripad</i> . . . . .                                | 41        |
| 5.6.5    | Třída <i>Generator</i> . . . . .   | 43        |
| 5.6.6    | Pomocné struktury . . . . .  | 43        |
| 5.6.7    | Funkce <i>Main</i> . . . . .   | 44        |
| <b>6</b> | <b>Výsledné funkce</b>   | <b>53</b> |
| 6.1      | Korelační funkce jen za vlivu kvantové statistiky . . . . .              | 53        |
| 6.2      | Korelační funkce jen za vlivu silné interakce. . . . .                   | 54        |
| 6.3      | Korelační funkce jen za vlivu Coulombické interakce. . . . .             | 54        |
| 6.4      | Korelační funkce za vlivu kompletního FSI. . . . .                       | 56        |
| 6.5      | Korelační funkce za vlivu kompletního FSI a kvantové statistiky. . . . . | 58        |
| 6.6      | Vliv generátoru rozdělení . . . . .                                      | 59        |
| <b>7</b> | <b>Zpracování dat</b>  | <b>61</b> |
| 7.1      | Korelační funkce identických částic . . . . .                            | 62        |
| 7.2      | Korelační funkce $\pi^+ \pi^-$ . . . . .                                 | 63        |
| <b>8</b> | <b>Závěr</b>   | <b>68</b> |
|          | <b>Appendices</b>  | <b>70</b> |
|          | <b>A Program</b>   | <b>71</b> |
|          | <b>B Script</b>  | <b>87</b> |
| B.1      | script.c . . . . .   | 87        |
| B.2      | script.h . . . . .   | 92        |



# Kapitola 1

## Úvod

Femtoskopické korelace mezi protony a elektrony emitovanými z určitého zdroje byly známy více než sedmdesát let. Korelace kvantové statistiky produkovaných identických částic byly pozorovány přibližně před padesáti lety, při zkoumání párů identických pionů s malými vzájemnými úhly (GGLP efekt). Základy moderní korelační femtoskopie byly zavedeny Kopylovem a Podgoretskym v raných sedmdesátých letech minulého století. Tehdy také byla objasněna analogie mezi femtoskopickými korelacemi hybnosti a HBT efektem. Vedle časoprostorových charakteristik částicové produkce, femtoskopické korelace také poskytují velice cenné informace o nízkoenergetické silné interakci mezi určitými částicemi, která nemůže být získána jinými prostředky.

Následující text je organizován tímto způsobem: V druhé kapitole se budeme zabírat popisem systému dvou částic. Zmíníme vliv chaotičnosti zdroje.

Ve třetí kapitole budeme probírat vliv kvantové statistiky, Coulombické interakce a silné interakce. Rozebereme případ interagujících a neinteragujících částic. Dále budeme diskutovat předpoklady hladkosti a současnosti.

Ve čtvrté kapitole stručně zmíníme metodu Monte Carlo.

Pátá kapitola se bude věnovat programu na simulace korelačních funkcí. Její účel je rozebrání úkolů programu a podmínek experimentu. Dále bude následovat analýza kódu. Bude zmíněna numerická knihovna `GSL` a pracovní prostředí `Root`.

V šesté kapitole budou uvedeny výsledné funkce a demonstrovány efekty FSI na korelační funkci. Popíšeme vliv Coulombické interakce, silné interakce a kvantové statistiky.

Sedmá kapitola se věnuje práci s daty. Rozebereme z nich získané korelační funkce a zmíníme jejich analogii se simulovanými.



## Kapitola 2

# Popis systému dvou částic a chaotičnost zdroje

Na začátku tohoto textu je třeba si položit otázku, proč se objevuje korelace v detekci dvou identických částic. Tento jev je nemožné popsat bez předchozího popisu podmínek, za kterých částice vzniká v určitých prostorových (prostorčasných) bodech. Z bodu vzniku musí částice dorazit do bodu detekce. Z toho logicky vyplývá, že potřebujeme ustanovit metodu popisu šíření částic. Pracujeme s dvoučásticovými korelacemi, což vyžaduje detekci dvou částic v přibližně shodném čase, musíme tedy do popisu tohoto jevu zahrnout symetrii. Ta je nezbytná z toho důvodu, že částice jsou identické a mohou se prohodit. Tato změna nemůže způsobit změnu popisující funkce. Z toho důvodu v případě identických částic musí být ta funkce symetrická, nebo antisymetrická, podle spinu částic. V této kapitole, stejně jako v následující kapitole budeme vycházet z [1].

Než přistoupíme ke konkrétnímu popisu šíření a detekce částic a dále k popisu celkové korelační funkce, je třeba ujasnit, jakou soustavu budeme používat pro popsání těchto jevů. Nebudeme používat laboratorní soustavu, jak by se zdálo na první pohled nejlepším řešením, ale budeme používat těžišťovou soustavu. Těžiště zdroje je v tomto stavu v klidu. Tato soustava je přirozenou soustavou k popisu srážek nukleonů v urychlovači. Na první pohled se může zdát, že tato soustava má jednu velkou nevýhodu, protože detektory jsou v ní v pohybu, když uvažujeme vysokoenetické srážky s pevnými terči. Tato nevýhoda vymizí, protože jak dále uvidíme, pravděpodobnosti detekce částic na poloze detektorů zanedbáváme.

Zabývejme se nyní značením proměnných. Piony se značí  $\pi^+$ ,  $\pi^-$ ,  $\pi^0$ . Čtyřhybnost značíme  $\vec{k} = (k_0, \mathbf{k})$ . Vektor  $\mathbf{k}$  je hybnost a skalár  $k_0$  je  $E/c$ . Časoprostorové souřadnice zdroje označujeme  $\vec{x} = (\vec{x}, t)$ . Časoprostorové souřadnice detektoru označujeme stejně, jenom čárkovaně:  $\vec{x}' = (\vec{x}', t')$ .

## 2.1 Popis jedné částice

### 2.1.1 Šíření ze zdroje

Zabývejme se nejprve detekcí částice o čtyřhybnosti  $\vec{k}$  v detekčním bodě  $\vec{x}'$ . Naše měření nemá dostatečné časové, ani prostorové rozlišení k tomu, abychom mohli rozhodnout, ze kterého bodu zdroje částice vylétla. Víme jenom, že vylétla z nějakého bodu zdroje. Klasickou (přímou) trajektorii přibližně vystihuje tato rovnice:

$$\vec{x}' - \vec{x} \approx (\vec{k}/k^0)(t' - t), \quad (2.1)$$

kde  $\vec{k}/k^0 = d\vec{x}/dt$  je rychlost částice.

Zaměříme se nejprve na pravděpodobnost detekce částice v bodě  $\vec{x}'$ , jestliže vznikla v bodě  $\vec{x}$ . Získání amplitudy pravděpodobnosti je kvantově-mechanický problém, a proto k němu používáme metodu Feynmanova dráhového integrálu. Tato amplituda je dána sumací fázových faktorů  $e^{iS}$  přes všechny dráhy:

$$\text{Amplituda} : [(\vec{x}, t) - (\vec{x}', t')] = \Psi(\vec{k} : \vec{x} \rightarrow (\vec{x}', t')) \equiv \Psi(\vec{k} : \vec{x} \rightarrow \vec{x}') = \sum_{\text{trajektorie}} e^{iS}, \quad (2.2)$$

kde  $S$  značí akci částice, která závisí jenom na časoprostorové trajektorii. Všechny tyto trajektorie mají začátek v bodě  $\vec{x}$  a konec v bodě detekce  $\vec{x}'$ . Částice se může pohybovat po nekonečném množství trajektorií. Hlavní příspěvek ale pochází od klasické trajektorie. Je tomu tak proto, že ať už si představíme jakoukoliv trajektorii, tak si můžeme ihned představit jinou trajektorii, která je s tou první osově souměrná podle přímký, kterou protneme body  $\vec{x}$  a  $\vec{x}'$ . Příspěvky těchto dvou trajektorií se navzájem vyruší. Je tedy přirozené aproximovat kvantově mechanickou pravděpodobnostní amplitudu výrazem  $e^{iS}$ . Tedy dostáváme tento výraz:

$$\Psi(k : \vec{x} \rightarrow \vec{x}') = e^{iS}, \quad (2.3)$$

kde  $S$  je akce podle klasické trajektorie. Poznamenejme, že podél trasy částice nepředpokládáme jevy, jako je třeba absorpce, nebo přerozptyl.

Na tomto místě je třeba zmínit, jak se počítá akce podél trajektorie. Je známým faktem, že akce se spočte jako:

$$S = \int L dt, \quad (2.4)$$

kde  $L$  značí Lagrangián částice. Ten se dá zapsat jako:

$$L = T - V, \quad (2.5)$$

kde  $T$  označuje kinetickou energii a  $V$  potenciální. Není úkolem této práce zabývat se celým výpočtem akce. Proto uvedu výsledek:

$$S(\vec{k} : \vec{x} \rightarrow \vec{x}') = \vec{k} \cdot (\vec{x} - \vec{x}'). \quad (2.6)$$

Je třeba dodat, že pro relativistický případ dostáváme to samé jako pro klasický. Odvození tohoto výsledku je možné najít v [1].

## 2.1.2 Produkční amplituda a fázový faktor

Kromě amplitudy šíření je třeba použít amplitudu produkce částic v daném bodě. Tuto amplitudu produkce můžeme popsat veličinou  $A(\vec{k}, \vec{x})$  a fázovým faktorem  $\phi(\vec{x})$ . Bez újmy na obecnosti můžeme psát, že veličina  $A(\vec{k}, \vec{x})$  je vždy reálná a nezáporná. Samotnou veličinou  $A(\vec{k}, \vec{x})$  se tu nebudeme detailně zabývat. Nyní jen řekneme, že závisí na produkčním procesu. Veličinu  $\phi(\vec{x})$  nazýváme produkční fází. Chování produkční fáze v různých bodech zdroje charakterizuje koherenci, nebo chaotičnost zdroje. Jinak řečeno  $\phi(\vec{x})$  charakterizuje stupeň koherence nebo chaotičnosti. Například chaotický zdroj je charakterizován náhodnou produkční fází v jednotlivých bodech zdroje, zatímco koherentní zdroj má produkční fázi nenáhodnou. Jestliže je nenáhodná, tak je funkcí bodu daného zdroje a jestliže máme dva body  $x$  a  $y$  zdroje, funkce  $\phi(\vec{x})$  a  $\phi(\vec{y})$  spolu souvisí.

Celková amplituda pravděpodobnosti, že částice vznikne v bodě  $\vec{x}$  s hybností  $\vec{k}$  a bude detekována v bodě  $\vec{x}'$  je pak dána součinem amplitudy produkce a šíření. Proto celková amplituda nalezení částice v bodě  $\vec{x}'$  je suma přes všechny body zdroje, nebo jak uvidíme později v případě spojitého zdroje, integrál amplitudy přes něj. V tom případě amplitudu pravděpodobnosti detekce částice o hybnosti  $k$  v bodě  $\vec{x}'$  zapíšeme jako:

$$\Psi(\vec{k} : \vec{x} \rightarrow \vec{x}') = \sum_{\vec{x}} A(\vec{k}, \vec{x}) e^{i\phi(\vec{x})} \Psi(\vec{k} : \vec{x} \rightarrow \vec{x}') = \sum_{\vec{x}} A(\vec{k}, \vec{x}) e^{i\phi(\vec{x})} e^{i\vec{k} \cdot (\vec{x} - \vec{x}')} . \quad (2.7)$$

Poslední rovnici můžeme vyjádřit integrálem přes zdroj, pakliže zavedeme pojem hustoty zdrojových bodů  $\rho(\vec{x})$ . Potom tedy nahrazujeme sumu v předchozí rovnici integrálem:

$$\sum_{\vec{x}} \dots \rightarrow \int dx \rho(\vec{x}) . \quad (2.8)$$

Z amplitudy pravděpodobnosti teď vyjádříme pravděpodobnost, tedy jednočásticové rozdělení pravděpodobnosti, že částice o hybnosti  $\vec{k}$  produkovaná ve zdroji dorazí do detekčního bodu  $\vec{x}'$ . To se dá vyjádřit jako čtverec absolutní hodnoty produkční amplitudy:

$$P(\vec{k}) = |\Psi(\vec{k} : \vec{x}(\text{zdroj}) \rightarrow \vec{x}')|^2 = \left| \sum_{\vec{x}} A(\vec{k}, \vec{x}) e^{i\phi(\vec{x})} e^{i\vec{k} \cdot \vec{x}} \right|^2 . \quad (2.9)$$

## 2.1.3 Rozdělení pravděpodobnosti

Poslední rovnici budeme používat pro vyšetření rozdělení pravděpodobnosti hybnosti z chaotického zdroje. Pro nás je chaotický zdroj nejdůležitější v tom, že produkční fáze  $\phi(\vec{x})$  je náhodná funkce souřadnic  $\vec{x}$  bodu zdroje.

Rovnici (2.9) upravíme na:

$$P(\vec{k}) = \sum_{\vec{x}} A^2(\vec{k}, \vec{x}) + \sum_{\vec{x}, \vec{y} | \vec{x} \neq \vec{y}} A(\vec{k}, \vec{x}) A(\vec{k}, \vec{y}) e^{i\phi(\vec{x})} e^{-i\phi(\vec{y})} e^{i\vec{k} \cdot (\vec{x} - \vec{y})} . \quad (2.10)$$

První člen pravé strany rovnice znamená součet přes všechny body zdroje. V tom případě je jasné, že produkční fáze  $e^{i\phi(\vec{x})}$  a člen  $e^{i\vec{k}\vec{x}}$  se nám vynulují kvůli komplexní sdruženosti. Odvození konečného tvaru je poměrně dlouhé, proto zde uvedeme výsledek:

$$P(\vec{k}) = \sum_{\vec{x}} A^2(\vec{k}, \vec{x}). \quad (2.11)$$

Tuto rovnici můžeme opět zapsat používaje integrál:

$$P(\vec{k}) = \int \rho(\vec{x}) A^2(\vec{k}, \vec{x}) dx. \quad (2.12)$$

Provedeme malou úpravu:

$$P(\vec{k}) = \int f(\vec{k}, \vec{x}) dx. \quad (2.13)$$

Veličina  $f(\vec{k}, \vec{x})$  je nazývána distribuční funkce a platí pro ní:

$$f(\vec{k}, \vec{x}) = \rho(\vec{x}) A^2(\vec{k}, \vec{x}). \quad (2.14)$$

Její vztah s amplitudou pravděpodobnostní produkce je popsán rovnicí:

$$A(\vec{k}, \vec{x}) = \sqrt{\frac{f(\vec{k}, \vec{x})}{\rho(\vec{x})}}. \quad (2.15)$$

Než přistoupíme k rozdělení hybnosti pro dvě částice, musíme se věnovat kvantové statistice. V následujících kapitolách ji budeme používat pro symetrizaci funkce. Jak později uvidíme při rozboru konkrétních případů korelační funkce, kvantová statistika a symetrizace nejsou jen forma popisu, ale mají vliv na konkrétní fyzikální jevy.

## 2.2 Popis dvoučásticového systému

Jestliže chceme spočítat něco, co je popsáno kvantovou mechanikou, musíme si uvědomit jestli částice se kterými pracujeme jsou identické, nebo neidentické. Toto je problém se kterým se nesetkáme, jestliže pracujeme s klasickou mechanikou, protože tehdy můžeme vždy rozlišit jednu částici od druhé. V kvantové mechanice je situace poněkud složitější, protože existuje možnost, že se budeme zabývat částicemi, které mají všechny vnitřní parametry stejné. Vnitřní parametry jsou například hmotnost, magnetický moment, náboj částice atd. Takové částice jsou nerozlišitelné.

V klasické mechanice tento pojem není podstatný, neboť každá částice se pohybuje po dané křivce určené pohybovými rovnicemi a pokud si částice na začátku

experimentu označíme např. jako "první", "druhá" atd., je možné v každém čase rozhodnout, o kterou částici se jedná a všechny částice lze tedy považovat za rozlišitelné.

Stav kvantové částice není určen její rychlostí a pozicí, ale její pozicí ve fázovém prostoru. Popisujeme je pomocí vlnové funkce:

$$\psi = \psi(t, \vec{x}), \quad (2.16)$$

kde  $t$  je čas a  $\vec{x}$  poloha částice.

Konkrétní tvar vlnové funkce závisí jak na podmínkách daného systému, tak i na druhu částic. Jak každý jistě ví z Bohrova postulátu, funkce  $\psi = \psi(t, \vec{x})$  neurčuje přesně polohu částice, ale kvadrát její absolutní hodnoty nám dává pravděpodobnost nalezení částice v okolí určitého bodu.

Funkce popisující stav dvou částic je zapisovaná jako

$$\psi = \psi(t, \vec{x}_1, \vec{x}_2), \quad (2.17)$$

$t$  je čas a vektory  $\vec{x}_1$  a  $\vec{x}_2$  jsou souřadnice první a druhé částice.

Jak zapsat funkci dvou částic  $\psi = \psi(t, \vec{x}_1, \vec{x}_2)$ , jestliže máme funkce jednotlivých částic? Na první pohled by se zdálo intuitivní, že jestliže funkce  $\psi$  je vlnová funkce částic v bodech  $\vec{x}_1$  a  $\vec{x}_2$ , tak pro pravděpodobnost současného nalezení částic v dvou bodech platí:

$$\psi = \psi(t, \vec{x}_1, \vec{x}_2) = \psi_{a_1}(t, \vec{x}_1) \cdot \psi_{a_2}(t, \vec{x}_2) \quad (2.18)$$

Poslední rovnice je platná pro neinteragující neidentické částice. Neplatí však pro všechny možné případy, protože by to znamenalo, že pravděpodobnost jedné částice byla nezávislá na druhé a to není obecně pravda.

Před tím, než zodpovíme tuto otázku, věnujme pozornost následující skutečnosti. Při záměně částic se stavová funkce  $\psi = \psi(\vec{x}_1, \vec{x}_2)$  změní na  $\tilde{\psi}(\vec{x}_1, \vec{x}_2) = \psi(\vec{x}_2, \vec{x}_1)$ . Pro nerozlišitelné částice se ale výsledky měření na dvoučásticovém systému touto záměnou nemohou změnit. Jelikož také musí platit:

$$\psi(\vec{x}_1, \vec{x}_2) = C_\psi \psi(\vec{x}_2, \vec{x}_1) = C_\psi^2 \psi(\vec{x}_1, \vec{x}_2), \quad (2.19)$$

kde  $C_\psi$  je reálné číslo. Z toho vyplývá:

$$C_\psi = \pm 1. \quad (2.20)$$

Funkce tedy může být jen symetrická, nebo antisymetrická. Pro částice, jejichž funkce je symetrická, platí

$$\psi(\vec{x}_1, \vec{x}_2) = \psi(\vec{x}_2, \vec{x}_1). \quad (2.21)$$

Tyto částice nazýváme bosony. Ty mají celý spin a jsou to například fotony nebo  $\pi$  mezony. Částice s polovičním spinem se nazývají fermiony a jejich funkce je antisymetrická. tj:

$$\psi(\vec{x}_1, \vec{x}_2) = -\psi(\vec{x}_2, \vec{x}_1). \quad (2.22)$$

Jsou to například elektrony nebo nukleony.

Jak tedy vytvářet funkci  $\psi = \psi(\vec{x}_1, \vec{x}_2)$ ? Prohození dvou souřadnic nemění stav systému, proto pro symetrickou funkci platí:

$$\psi_{a_1, a_2}(\vec{x}_1, \vec{x}_2) = \frac{1}{\sqrt{2}}[\psi_{a_1}(\vec{x}_1)\psi_{a_2}(\vec{x}_2) + \psi_{a_1}(\vec{x}_2)\psi_{a_2}(\vec{x}_1)]. \quad (2.23)$$

A pro antisymetrickou:

$$\psi_{a_1, a_2}(\vec{x}_1, \vec{x}_2) = \frac{1}{\sqrt{2}}[\psi_{a_1}(\vec{x}_1)\psi_{a_2}(\vec{x}_2) - \psi_{a_1}(\vec{x}_2)\psi_{a_2}(\vec{x}_1)]. \quad (2.24)$$

Podobnými metodami se dá popsat i funkce určující stav více částic. Tím už se zde nebudeme zabývat, protože v této práci nás budou zajímat jen korelace dvou částic. Než opustíme tuto kapitolu, zdůraznil bych, že je třeba věnovat zvýšenou pozornost posledním dvěma rovnicím. V této kapitole jsme značili vlnovou funkci znakem  $\Psi$ , stejně jako v předchozích podkapitolkách amplitudy pravděpodobnosti. Tato kapitola sloužila jenom jako úvod do nejnútnejšího základu kvantové mechaniky, s kterým budeme dále pracovat. Proto nebyla zvolena nová proměnná a vlnová funkce se značila jako ve většině publikací. Dále znak  $\Psi$  bude mít význam jako v textu před touto podkapitolkou.

## 2.3 Rozdělení hybnosti pro dvě částice

Nyní obrátíme svojí pozornost ke korelacím hybnosti při detekci dvou identických částic z obecného zdroje. Uvažujeme nyní dvě částice s hybnostmi  $\vec{k}_1$  a  $\vec{k}_2$ , které byly detekované v bodech  $\vec{x}'_1$  a  $\vec{x}'_2$ . Tyto dvě částice mohly pocházet z jakýchkoliv bodů ve zdroji. Označme tyto dva body  $\vec{x}_1$  a  $\vec{x}_2$ . Nyní nás zajímá pravděpodobnost respektive amplituda pravděpodobnosti, že částice dorazí z bodů vzniku do bodů detekce.

Začneme s případem, kdy částice s hybností  $\vec{k}_1$  putuje z bodu  $\vec{x}_1$  do bodu  $\vec{x}'_1$  a částice  $\vec{k}_2$  putuje z bodu  $\vec{x}_2$  do bodu  $\vec{x}'_2$ . Amplituda pravděpodobnosti tohoto jevu bude součinem amplitudy jevů  $\psi(\vec{k}_1 : \vec{x}_1 \rightarrow \vec{x}'_1)$  a  $\psi(\vec{k}_2 : \vec{x}_2 \rightarrow \vec{x}'_2)$ . Opět používáme Feynmanovu integrální metodu, to znamená, že hlavní příspěvek k amplitudě pravděpodobnosti pochází z klasických trajektorií. Nejprve dostáváme výraz:

$$\psi(\vec{k}_1 : \vec{x}_1 \rightarrow \vec{x}'_1)\psi(\vec{k}_2 : \vec{x}_2 \rightarrow \vec{x}'_2). \quad (2.25)$$

Z předcházejícího víme, že:

$$S_{cl}(\vec{k} : \vec{x}_i \rightarrow \vec{x}'_i) = \vec{k}_i \cdot (\vec{x}_i - \vec{x}'_i). \quad (2.26)$$

Nyní tento vztah použijeme a výraz (2.25) s jejich pomocí přepíšeme do tvaru:

$$\psi(\vec{k}_1 : \vec{x}_1 \rightarrow \vec{x}'_1) \psi(\vec{k}_2 : \vec{x}_2 \rightarrow \vec{x}'_2) = e^{\vec{k}_1 \cdot (\vec{x}_1 - \vec{x}'_1)} e^{\vec{k}_2 \cdot (\vec{x}_2 - \vec{x}'_2)}. \quad (2.27)$$

Exponenciály na pravé straně vychází z (2.3). Toto je amplituda pravděpodobnosti, že částice dorazí z bodů vzniku do bodů detekce. My potřebujeme amplitudu toho, že částice vzniknou v počátečních bodech a dorazí do bodů detekce. Proto do předchozí rovnice potřebujeme vložit ještě produkční amplitudu, která nám říká, s jakou pravděpodobností se vytvoří částice o hybnosti  $\vec{k}_1$  v bodě  $\vec{x}_1$ . Z předcházejících kapitol víme, že tato amplituda pravděpodobnosti produkce se dá vyjádřit jako  $A(\vec{k}_i, \vec{x}_i) e^{i\phi(\vec{x}_i)}$ . Pravděpodobnost produkce dvou částic ve dvou bodech se vyjádří součinem. Tedy:

$$A(\vec{k}_1, \vec{x}_1) e^{i\phi(\vec{x}_1)} A(\vec{k}_2, \vec{x}_2) e^{i\phi(\vec{x}_2)}. \quad (2.28)$$

Pravděpodobnost produkce v bodě  $\vec{x}_1$  a šíření do bodu  $\vec{x}_2$  získáme, když tímto výrazem vynásobíme levou stranu rovnice (2.27).

$$A(\vec{k}_1, \vec{x}_1) e^{i\phi(\vec{x}_1)} A(\vec{k}_2, \vec{x}_2) e^{i\phi(\vec{x}_2)} \psi(\vec{k}_1 : \vec{x}_1 \rightarrow \vec{x}'_1) \psi(\vec{k}_2 : \vec{x}_2 \rightarrow \vec{x}'_2). \quad (2.29)$$

Vyjádříme jí ve tvaru:

$$A(\vec{k}_1, \vec{x}_1) e^{i\phi(\vec{x}_1)} A(\vec{k}_2, \vec{x}_2) e^{i\phi(\vec{x}_2)} e^{\vec{k}_1 \cdot (\vec{x}_1 - \vec{x}'_1)} e^{\vec{k}_2 \cdot (\vec{x}_2 - \vec{x}'_2)}. \quad (2.30)$$

Problém tohoto vyjádření je ten, že zahrnuje pouze možnost, že částice produkovaná v bodě  $\vec{x}_1$  bude zaznamenána v bodě detekce  $\vec{x}'_1$  a to samé platí pro body  $\vec{x}_2$  a  $\vec{x}'_2$ . Neuvažovali jsme tu možnost, že by se částice mohly prohodit. To znamená, že částice produkovaná v bodě  $\vec{x}_1$  přiletí do  $\vec{x}'_2$  a obdobně pro druhou variantu. Proto musíme symetrizovat.

$$\begin{aligned} & \frac{1}{\sqrt{2}} \{ A(\vec{k}_1, \vec{x}_1) e^{i\phi(\vec{x}_1)} A(\vec{k}_2, \vec{x}_2) e^{i\phi(\vec{x}_2)} e^{\vec{k}_1 \cdot (\vec{x}_1 - \vec{x}'_1)} e^{\vec{k}_2 \cdot (\vec{x}_2 - \vec{x}'_2)} \\ & + A(\vec{k}_1, \vec{x}_2) e^{i\phi(\vec{x}_2)} A(\vec{k}_2, \vec{x}_1) e^{i\phi(\vec{x}_1)} e^{\vec{k}_1 \cdot (\vec{x}_2 - \vec{x}'_1)} e^{\vec{k}_2 \cdot (\vec{x}_1 - \vec{x}'_2)} \} \\ & \equiv \frac{1}{\sqrt{2}} e^{i\phi(\vec{x}_1)} e^{i\phi(\vec{x}_2)} \Phi(\vec{k}_1 \vec{k}_2 : \vec{x}_1 \vec{x}_2 \rightarrow \vec{x}'_1 \vec{x}'_2). \end{aligned} \quad (2.31)$$

Ve jmenovateli vystupuje  $\sqrt{2}$ . Tato odmocnina se objevila právě kvůli symetrizaci. Funkce  $\Phi(\vec{k}_1 \vec{k}_2 : \vec{x}_1 \vec{x}_2 \rightarrow \vec{x}'_1 \vec{x}'_2)$  je část pravděpodobnostní amplitudy, která nezávisí na  $\phi$ . Můžeme ji snadno definovat pomocí předchozí rovnice:

$$\begin{aligned} \Phi(\vec{k}_1 \vec{k}_2 : \vec{x}_1 \vec{x}_2 \rightarrow \vec{x}'_1 \vec{x}'_2) \equiv & \frac{1}{\sqrt{2}} \{ A(\vec{k}_1, \vec{x}_1) A(\vec{k}_2, \vec{x}_2) e^{\vec{k}_1 \cdot (\vec{x}_1 - \vec{x}'_1)} e^{\vec{k}_2 \cdot (\vec{x}_2 - \vec{x}'_2)} \\ & + A(\vec{k}_1, \vec{x}_2) A(\vec{k}_2, \vec{x}_1) e^{\vec{k}_1 \cdot (\vec{x}_2 - \vec{x}'_1)} e^{\vec{k}_2 \cdot (\vec{x}_1 - \vec{x}'_2)} \}. \end{aligned} \quad (2.32)$$

Částice samozřejmě nemusí vyjít ze dvou přesně určených bodů zdroje, ale mohou být produkovány v jakékoliv dvojici bodů. Celkovou amplitudu bychom mohli vyjádřit jako sumu přes všechny možné kombinace dvou produkčních bodů. Proto můžeme zapsat amplitudu pravděpodobnosti toho, že dvě identické částice s hybnostmi  $\vec{k}_1$  a  $\vec{k}_2$  vyrazí z nějakých bodů zdroje a jsou detekovány v bodech  $\vec{x}'_1$  a  $\vec{x}'_2$ , jako:

$$\Psi(\vec{k}_1 \vec{k}_2 : zdroj \rightarrow \vec{x}'_1 \vec{x}'_2) = \sum_{\vec{x}_2, \vec{x}_1} e^{i\phi(\vec{x}_1)} e^{i\phi(\vec{x}_2)} \Phi(\vec{k}_1 \vec{k}_2 : \vec{x}_1 \vec{x}_2 \rightarrow \vec{x}'_1 \vec{x}'_2). \quad (2.33)$$

Rozdělení pravděpodobnosti produkce částic s hybnostmi  $\vec{k}_1$  a  $\vec{k}_2$  a jejich následná detekce v bodech  $\vec{x}'_1$  a  $\vec{x}'_2$  je tedy definovaná jako:

$$P(\vec{k}_1, \vec{k}_2) = \frac{1}{2} |\Psi(\vec{k}_1 \vec{k}_2 : zdroj \rightarrow \vec{x}'_1 \vec{x}'_2)|^2. \quad (2.34)$$

Dále se dostaneme k chaotickému a koherentnímu zdroji. Budeme je rozlišovat podle vlastností fází  $\phi$ . Po nastínění popisu systému částic můžeme zavést pojem korelace.

## 2.4 Chaotický zdroj

Pro chaotické zdroje budeme využívat vlastností fází  $\phi$ . Už víme, že chaotický zdroj je charakterizován náhodností fáze  $\phi$ , tedy fáze  $\phi$  je náhodnou funkcí souřadnice zdroje. Vraťme se k rovnicím (2.33) a (2.34) z minulé kapitoly. Rozdělíme části závislejší a nezávislejší na  $\phi$  ve výrazu pro pravděpodobnost. Dostaneme:

$$\begin{aligned} P(\vec{k}_1, \vec{k}_2) = & \frac{1}{2} \sum_{\vec{x}_1, \vec{x}_2} \{ \Phi^*(\vec{k}_1, \vec{k}_2 : \vec{y}_1, \vec{y}_2 \rightarrow \vec{x}'_1, \vec{x}'_2) * \Phi(\vec{k}_1, \vec{k}_2 : \vec{x}_1, \vec{x}_2 \rightarrow \vec{x}'_1, \vec{x}'_2)_{\vec{y}_1 = \vec{x}_1, \vec{y}_2 = \vec{x}_2} \\ & + \Phi^*(\vec{k}_1, \vec{k}_2 : \vec{y}_1, \vec{y}_2 \rightarrow \vec{x}'_1, \vec{x}'_2) * \Phi(\vec{k}_1, \vec{k}_2 : \vec{x}_1, \vec{x}_2 \rightarrow \vec{x}'_1, \vec{x}'_2)_{\vec{y}_2 = \vec{x}_1, \vec{y}_1 = \vec{x}_2} \} \\ & + \frac{1}{2} \sum_{\vec{x}_1, \vec{x}_2, \vec{y}_1, \vec{y}_2} \{ e^{i\phi(\vec{x}_1) + i\phi(\vec{x}_2) - i\phi(\vec{y}_1) - i\phi(\vec{y}_2)} * \\ & \Phi^*(\vec{k}_1, \vec{k}_2 : \vec{y}_1, \vec{y}_2 \rightarrow \vec{x}'_1, \vec{x}'_2) * \Phi(\vec{k}_1, \vec{k}_2 : \vec{x}_1, \vec{x}_2 \rightarrow \vec{x}'_1, \vec{x}'_2)_{\vec{x}_1, \vec{x}_2 \neq \vec{y}_1, \vec{y}_2} \}. \end{aligned} \quad (2.35)$$

Dvě části v první sumě na pravé straně jsou stejné kvůli symetrii. Druhá suma na pravé straně rovnice je nula, protože příspěvky od velkého množství čísel podobné velikosti a odlišných fází se odečtou. Kvůli těmto důvodům se nám rovnice zjednoduší na:

$$P(\vec{k}_1, \vec{k}_2) = \sum_{\vec{x}_1, \vec{x}_1} |\Phi(\vec{k}_1, \vec{k}_2 : \vec{x}_1, \vec{x}_2 \rightarrow \vec{x}'_1, \vec{x}'_2)|^2. \quad (2.36)$$



Přejdeme k případu, kde je možno sumu nahradit integrálem. Vyskytují se nám v rovnici dvě proměnné, tak musíme integrovat přes dvě souřadnice  $\vec{x}_1$  a  $\vec{x}_2$ . V tom případě dostáváme:

$$P(\vec{k}_1, \vec{k}_2) = \int \rho(\vec{x}_1)\rho(\vec{x}_2)|\Phi(\vec{k}_1, \vec{k}_2 : \vec{x}_1, \vec{x}_2 \rightarrow \vec{x}'_1, \vec{x}'_2)|^2 dx_1 dx_2. \quad (2.37)$$

Tuto rovnici upravíme na:

$$\begin{aligned} P(\vec{k}_1, \vec{k}_2) = & \int \rho(\vec{x}_1)A^2(\vec{k}_1, \vec{x}_1)dx_1 \int \rho(\vec{x}_2)A^2(\vec{k}_2, \vec{x}_2)dx_2 + \\ & \int \rho(\vec{x}_1)A(\vec{k}_1, \vec{x}_1)A(\vec{k}_2, \vec{x}_2)e^{i(\vec{k}_1-\vec{k}_2)\vec{x}_1} dx_1 \times \\ & \int \rho(\vec{x}_2)A(\vec{k}_2, \vec{x}_2)A(\vec{k}_1, \vec{x}_1)e^{i(\vec{k}_2-\vec{k}_1)\vec{x}_2} dx_2. \end{aligned} \quad (2.38)$$

Produkční amplitudy pravděpodobnosti jsme již popisovali v předchozím textu, kde jsme popisovali situaci jedné částice. Amplituda byla zapsaná jako:

$$P(\vec{k}_1) = \int \rho(\vec{x}_1)A(\vec{k}_1, \vec{x}_1)dx_1. \quad (2.39)$$

Za použití tohoto vztahu prepíšeme předchozí rovnici na tvar:

$$P(\vec{k}_1, \vec{k}_2) = P(\vec{k}_1)P(\vec{k}_2) + \left| \int e^{i(\vec{k}_1-\vec{k}_2)\vec{x}} \rho(\vec{x})A(\vec{k}_1, \vec{x})A(\vec{k}_2, \vec{x})dx \right|^2. \quad (2.40)$$

Teď je na čase, abychom zavedli velice důležitý pojem efektivní hustoty. Tu budeme definovat jako:

$$\rho_{eff}(\vec{x}; \vec{k}_1, \vec{k}_2) = \frac{\rho(\vec{x})A(\vec{k}_1, \vec{x})A(\vec{k}_2, \vec{x})}{\sqrt{P(\vec{k}_1)P(\vec{k}_2)}}. \quad (2.41)$$

Pomocí efektivní hustoty prepíšeme předchozí rovnici na tvar:

$$P(\vec{k}_1, \vec{k}_2) = P(\vec{k}_1)P(\vec{k}_2)(1 + \left| \int e^{i(\vec{k}_1-\vec{k}_2)\vec{x}} \rho_{eff}(\vec{x}; \vec{k}_1, \vec{k}_2)dx \right|^2). \quad (2.42)$$

Jestliže použijeme distribuční funkci  $f(\vec{k}, \vec{x})$  pro chaotické zdroje, efektivní hustota může být vyjádřena:

$$\rho_{eff}(\vec{x}; \vec{k}_1, \vec{k}_2) = \frac{\sqrt{f(\vec{k}_1, \vec{x})f(\vec{k}_2, \vec{x})}}{\sqrt{\int f(\vec{k}_1, \vec{x}_1)dx_1 \int f(\vec{k}_2, \vec{x}_2)dx_2}}. \quad (2.43)$$

Využijeme Fourierovy transformace definované jako:

$$\tilde{\rho}_{eff}(\vec{x}; \vec{k}_1, \vec{k}_2) = \int e^{i\vec{q}\vec{x}} \rho_{eff}(\vec{x}; \vec{k}_1, \vec{k}_2), \quad (2.44)$$

$\vec{q} = \vec{k}_1 - \vec{k}_2$ . Nyní můžeme přepsat pravděpodobnost do tvaru:

$$P(\vec{k}_1, \vec{k}_2) = P(\vec{k}_1)P(\vec{k}_2)(1 + |\tilde{\rho}_{eff}(\vec{q}; \vec{k}_1, \vec{k}_2)|^2). \quad (2.45)$$

Celé předcházející odvození jsme dělali, abychom dospěli ke korelační funkci  $C_2(k_1, k_2)$ . Je definovaná jako poměr pravděpodobnosti detekce dvou částic s hybnostmi  $\vec{k}_1$  a  $\vec{k}_2$  současně, tj.  $P(\vec{k}_1, \vec{k}_2)$  a násobku dvou pravděpodobností, když pozorujeme jednotlivé funkce samostatně.

$$C(\vec{k}_1, \vec{k}_2) = \frac{P(\vec{k}_1, \vec{k}_2)}{P(\vec{k}_1)P(\vec{k}_2)} \quad (2.46)$$

Vyjádřeno pomocí rovnice (2.45) dostaneme jiný tvar:

$$C(\vec{k}_1, \vec{k}_2) = (1 + |\tilde{\rho}_{eff}(\vec{q}; \vec{k}_1, \vec{k}_2)|^2). \quad (2.47)$$

Pro chaotický zdroj vidíme, že korelační funkce přímo souvisí s Fourierovou transformací efektivní hustoty. Dvoučásticová korelační funkce může být použita jako způsob, jímž lze získat informace o časoprostorovém rozložení zdroje v čase emise.

Zavedeme korelační funkci  $R(\vec{k}_1, \vec{k}_2)$ , která souvisí s  $C(\vec{k}_1, \vec{k}_2)$  tímto způsobem:

$$R(\vec{k}_1, \vec{k}_2) = C(\vec{k}_1, \vec{k}_2) - 1. \quad (2.48)$$

Takže je nyní zcela jasně viditelná souvislost mezi  $R(\vec{k}_1, \vec{k}_2)$  a  $\tilde{\rho}_{eff}(\vec{q}; \vec{k}_1, \vec{k}_2)$ . Dostaneme tento vztah:

$$R(\vec{k}_1, \vec{k}_2) = |\tilde{\rho}_{eff}(\vec{q}; \vec{k}_1, \vec{k}_2)|^2. \quad (2.49)$$

## 2.5 Koherentní zdroj

V předcházející kapitole bylo řečeno, že identické částice z chaotického zdroje jsou korelované. Jaký typ korelace můžeme očekávat u koherentních zdrojů?

To nejzákladnější pro náš popis koherentních zdrojů je fakt, že produkční fáze není náhodná funkce, to znamená, že není náhodnou funkcí  $\phi(\vec{x})$  souřadnice  $\vec{x}$ . Produkční fáze je pro tento případ jiná. V anglické literatuře se vystihuje spojením "well-behaved function". Vynecháme doslovný překlad tohoto pojmu a o dané funkci řekneme to nejdůležitější, a to je fakt, že hodnoty  $\phi(\vec{x})$  a  $\phi(\vec{y})$  ve dvou různých

bodech zdroje  $\vec{x}$  a  $\vec{y}$  spolu souvisí. Můžeme hned uvést velice jednoduchý příklad koherentního zdroje, respektive fáze koherentního zdroje a to takový zdroj, jehož fáze je všude konstantní, tj:  $\phi(\vec{x}) = c$ . Je vhodné uvést ještě jeden příklad. Představme si, že fáze  $\phi(\vec{x})$  je nezávislá na prostorové souřadnici  $x$  a závisí jen na čase.

Nejdříve je třeba se vrátit k jednočásticové amplitudě pravděpodobnosti. To znamená, amplituda pravděpodobnosti produkce v bodě  $\vec{x}$  s hybností  $\vec{k}$  a její detekce v bodě  $\vec{x}'$ . Tuto amplitudu jsme už viděli v rovnici (2.7). Vyjádříme amplitudu pravděpodobnosti produkce rovnou ve všech bodech zdroje:

$$\Psi(\vec{k} : zdroj \rightarrow \vec{x}') = \sum_{\vec{x}} A(\vec{k}, \vec{x}) e^{i\phi(\vec{x})} e^{i\vec{k}(\vec{x}-\vec{x}')}. \quad (2.50)$$

Protože produkční fáze není náhodná, ale je "dobře se chovající", můžeme provést sumaci v poslední rovnici. Vhodnější je přepsat sumaci do formy integrálu. Používáme přepis:

$$\left(\sum_{\vec{x}}\right) \rightarrow \left(\int \rho(\vec{x})\right). \quad (2.51)$$

Pomocí tohoto přepisu vyjádříme rovnici (2.50) a pro amplitudu pravděpodobnosti produkce získáme jiný zápis:

$$\Psi(\vec{k} : zdroj \rightarrow \vec{x}') = \int \rho(\vec{x}) A(\vec{k}, \vec{x}) e^{i\phi(\vec{x})} e^{i\vec{k}(\vec{x}-\vec{x}')} dx. \quad (2.52)$$

Máme tedy amplitudu produkce částice o hybnosti  $\vec{k}$  ve zdroji a její následné putování do bodu detekce. Nyní chceme pravděpodobnost daného jevu. Tu získáme jednoduše jako čtverec absolutní hodnoty amplitudy pravděpodobnosti:

$$\begin{aligned} P(\vec{k}) &= \left| \int \rho(\vec{x}) A(\vec{k}, \vec{x}) e^{i\phi(\vec{x})} e^{i\vec{k}(\vec{x}-\vec{x}')} dx \right|^2 \\ &= \left| \int \rho(\vec{x}) A(\vec{k}, \vec{x}) e^{i\phi(\vec{x})} e^{i\vec{k}\vec{x}} dx \right|^2. \end{aligned} \quad (2.53)$$

Z toho vidíme, že pro koherentní zdroj jednočásticové rozdělení podle hybnosti je čtverec absolutní hodnoty Fourierovy transformace výrazu  $\rho(\vec{x}) A(\vec{k}, \vec{x}) e^{i\phi(\vec{x})}$ .

Budeme nyní studovat korelaci hybností při produkci dvou identických částic z koherentního zdroje. Jak už jsme viděli v obecném odvození, musíme získat amplitudu pravděpodobnosti vzniku dvou částic ze všech kombinací dvou bodů ve zdroji. V případě spojitého rozložení, když používáme integrál, integrujeme přes dvě proměnné. Jestliže rychle zrekapitulujeme důležité body odvození, víme že část amplitudy nezávislá na fázi  $\phi(\vec{x})$  je:

$$\begin{aligned} \Phi(\vec{k}_1 \vec{k}_2 : \vec{x}_1 \vec{x}_2 \rightarrow \vec{x}'_1 \vec{x}'_2) &= \frac{1}{\sqrt{2}} \{ A(\vec{k}_1, \vec{x}_1) A(\vec{k}_2, \vec{x}_2) e^{i\vec{k}_1(\vec{x}_1 - \vec{x}'_1)} e^{i\vec{k}_2(\vec{x}_2 - \vec{x}'_2)} + \\ & A(\vec{k}_1, \vec{x}_2) A(\vec{k}_2, \vec{x}_1) e^{i\vec{k}_1(\vec{x}_2 - \vec{x}'_1)} e^{i\vec{k}_2(\vec{x}_1 - \vec{x}'_2)} \}. \end{aligned} \quad (2.54)$$

Amplitudu získáme vynásobením daného výrazu členy obsahujícími produkční fáze:

$$\Psi(\vec{k}_1 \vec{k}_2 : zdroj \rightarrow \vec{x}'_1 \vec{x}'_2) = \sum_{\vec{x}_1 \vec{x}_2} e^{i\phi(\vec{x}_1)} e^{i\phi(\vec{x}_2)} \Phi(\vec{k}_1 \vec{k}_2 : \vec{x}_1 \vec{x}_2 \rightarrow \vec{x}'_1 \vec{x}'_2). \quad (2.55)$$

A konečně získáme pravděpodobnost:

$$P(\vec{k}_1, \vec{k}_2) = \frac{1}{2} |\Psi(\vec{k}_1 \vec{k}_2 : zdroj \rightarrow \vec{x}'_1 \vec{x}'_2)|^2. \quad (2.56)$$

Vyjádříme funkci  $\Psi$  pomocí funkce  $\phi$ :

$$\begin{aligned} \Psi(\vec{k}_1 \vec{k}_2 : zdroj \rightarrow \vec{x}'_1 \vec{x}'_2) &= \frac{1}{\sqrt{2}} \sum_{\vec{x}_1 \vec{x}_2} e^{i\phi(\vec{x}_1)} e^{i\phi(\vec{x}_2)} * \\ & \{ A(\vec{k}_1, \vec{x}_1) A(\vec{k}_2, \vec{x}_2) e^{i\vec{k}_1(\vec{x}_1 - \vec{x}'_1)} e^{i\vec{k}_2(\vec{x}_2 - \vec{x}'_2)} + A(\vec{k}_1, \vec{x}_2) A(\vec{k}_2, \vec{x}_1) e^{i\vec{k}_1(\vec{x}_2 - \vec{x}'_1)} e^{i\vec{k}_2(\vec{x}_1 - \vec{x}'_2)} \}. \end{aligned} \quad (2.57)$$

Toto byl obecný postup. Nyní budeme situaci konkretizovat pro koherentní zdroj. Tady už máme odlišnou situaci než u chaotického zdroje, kde jsme kvůli chaotickému chování museli rozdělit sumu na části obsahující a neobsahující produkční fázi  $\phi(\vec{x})$ .

Budeme provádět sumu přes  $\vec{x}_1$  a  $\vec{x}_2$ . Výhoda je, že to lze provádět nezávisle:

$$\begin{aligned} \Psi(\vec{k}_1 \vec{k}_2 : zdroj \rightarrow \vec{x}'_1 \vec{x}'_2) &= \frac{1}{\sqrt{2}} \sum_{\vec{x}_1} e^{i\phi(\vec{x}_1)} A(\vec{k}_1, \vec{x}_1) e^{i\vec{k}_1(\vec{x}_1 - \vec{x}'_1)} \\ & \sum_{\vec{x}_2} e^{i\phi(\vec{x}_2)} A(\vec{k}_2, \vec{x}_2) e^{i\vec{k}_2(\vec{x}_2 - \vec{x}'_2)} + \\ & \frac{1}{\sqrt{2}} \sum_{\vec{x}_2} e^{i\phi(\vec{x}_2)} A(\vec{k}_1, \vec{x}_2) e^{i\vec{k}_1(\vec{x}_2 - \vec{x}'_1)} \sum_{\vec{x}_1} e^{i\phi(\vec{x}_1)} A(\vec{k}_2, \vec{x}_1) e^{i\vec{k}_2(\vec{x}_1 - \vec{x}'_2)}. \end{aligned} \quad (2.58)$$

To zjednodušíme do tvaru:

$$\begin{aligned} \Psi(\vec{k}_1 \vec{k}_2 : zdroj \rightarrow \vec{x}'_1 \vec{x}'_2) &= \\ & \frac{1}{\sqrt{2}} \sum_{\vec{x}_1} e^{i\phi(\vec{x}_1)} A(\vec{k}_1, \vec{x}_1) e^{i\vec{k}_1(\vec{x}_1 - \vec{x}'_1)} \sum_{\vec{x}_2} e^{i\phi(\vec{x}_2)} A(\vec{k}_2, \vec{x}_2) e^{i\vec{k}_2(\vec{x}_2 - \vec{x}'_2)} \\ & = \sqrt{2} \int \rho(\vec{x}) A(\vec{k}_1, \vec{x}_1) e^{i\phi(\vec{x}_1)} e^{i\vec{k}_1(\vec{x}_1 - \vec{x}'_1)} dx_1 \times \\ & \int \rho(\vec{x}) A(\vec{k}_2, \vec{x}_2) e^{i\phi(\vec{x}_2)} e^{i\vec{k}_2(\vec{x}_2 - \vec{x}'_2)} dx_2. \end{aligned} \quad (2.59)$$

Pro pravděpodobnost produkce dvou částic s danými hybnostmi a jejich detekce v bodech  $\vec{x}'_1$  pro  $\vec{k}_1$  a  $\vec{x}'_2$  pro  $\vec{k}_2$  dostaneme:

$$\begin{aligned}
P(\vec{k}_1, \vec{k}_2) &= \left| \int \rho(\vec{x}) A(\vec{k}_1, \vec{x}_1) e^{i\phi(\vec{x}_1)} e^{\vec{k}_1(\vec{x}_1 - \vec{x}'_1)} dx_1 \right. \\
&\quad \left. \int \rho(\vec{x}) A(\vec{k}_2, \vec{x}_2) e^{i\phi(\vec{x}_2)} e^{\vec{k}_2(\vec{x}_2 - \vec{x}'_2)} dx_2 \right|^2 = \\
&= \left| \int \rho(\vec{x}) A(\vec{k}_1, \vec{x}_1) e^{i\phi(\vec{x}_1)} e^{\vec{k}_1 \vec{x}_1} dx_1 \int \rho(\vec{x}) A(\vec{k}_2, \vec{x}_2) e^{i\phi(\vec{x}_2)} e^{\vec{k}_2 \vec{x}_2} dx_2 \right|^2.
\end{aligned} \tag{2.60}$$

Když teď srovnáme pravděpodobnost  $P(\vec{k}_1, \vec{k}_2)$  pro dvě částice s pravděpodobnostmi pro jednotlivé částice, viz (2.53), uvidíme jednoduchý vztah:

$$P(\vec{k}_1, \vec{k}_2) = P(\vec{k}_1)P(\vec{k}_2). \tag{2.61}$$

Z toho jde už jednoduše vyjádřit vztah pro korelační funkci pro koherentní zdroj:

$$C_2(\vec{k}_1, \vec{k}_2) = \frac{P(\vec{k}_1, \vec{k}_2)}{P(\vec{k}_1)P(\vec{k}_2)} = 1. \tag{2.62}$$

Pro funkci  $R(\vec{k}_1, \vec{k}_2)$ , platí:

$$R(\vec{k}_1, \vec{k}_2) = C(\vec{k}_1, \vec{k}_2) - 1 = 0. \tag{2.63}$$

Z toho je jasně vidět, že neexistuje žádná souvislost mezi detekcí jedné částice určité hybnosti a pravděpodobností detekce druhé částice s odlišnou hybností. Dá se tedy říci, že produkce a detekce dvou částic jsou navzájem nezávislé. Tedy jestliže zdroj je koherentní, není přítomna korelace hybností v dvoučásticové distribuční funkci.

Z tohoto poměrně detailního odvození korelační funkce je možné vysledovat příčinu hybnostní korelace pro chaotické zdroje. Můžeme pozorovat, že v případě koherentního zdroje je funkce amplitudy pravděpodobnosti symetrická a faktorizovatelná. Z faktorizovatelnosti amplitudy vyplývá, že dvoučásticová pravděpodobnostní funkce je součin dvou jednočásticových rozdělení hybnosti. Z rovnic (2.62) a (2.63) vidíme, že nemůže být žádná korelace v případě koherentního zdroje. V případě chaotického zdroje se fázová funkce  $\phi(\vec{x})$  "nechová hezky", to znamená, že je náhodnou funkcí souřadnice  $\vec{x}$ , a se sumací jsou větší potíže. Musíme nejprve separovat členy, kde se fázová funkce  $\phi(\vec{x})$  navzájem vyruší a ty členy, kde kvůli odlišnosti hodnot fáze zůstanou. Ty co budou záviset na náhodné fázi se navzájem vyruší, tak jak bylo vysvětleno v kapitole (2.4). Kvůli této skutečnosti existuje korelace u chaotických zdrojů. Pravděpodobnostní funkce  $P(\vec{k}_1, \vec{k}_2)$  není rovna součinu jednočásticových funkcí. Proto funkce  $R(\vec{k}_1, \vec{k}_2)$  není nulová, dvoučásticová korelace existuje a korelační funkce obsahuje Fourierovu transformaci funkce prostorového rozložení. Důvod, proč existuje korelace pro chaotické zdroje, je tedy výměnná symetrie.

# Kapitola 3

## Korelace, kvantová statistika a FSI

### 3.1 Formalismus

Abychom mohli popsat produkci dvou částic o malé vzájemné rychlosti, musíme se nejdřív zabývat otázkou, čím je daný jev ovlivňován. V předchozí kapitole jsme se setkali s vlivem kvantové statistiky. V této kapitole se budeme mimo jiné zabývat i možnou vzájemnou interakcí v konečném stavu (FSI). V této kapitole budeme vycházet z [5].

Výhodou, se kterou můžeme pracovat co se týče druhého jevu, je možnost oddělení FSI efektu od produkční amplitudy poskytnutím dostatečně dlouhého času pro interakci dvou částic v konečném stavu ve srovnání s časem produkčního procesu. Tato podmínka vyžaduje velikost jejich vzájemné hybnosti  $\vec{q} = \vec{k}_1 - \vec{k}_2 = 2\vec{k}$  menší než několik stovek MeV/c. Tento požadavek se dá vysvětlit dost jednoduše. Při malých vzájemných rychlostech je jejich čas vzájemného působení podstatně delší, než čas typický pro produkční procesy a tudíž můžou být tyto dva jevy navzájem separovány.

Než se budeme věnovat dalšímu postupu, je vhodné zmínit pár veličin, které budeme používat. První a patrně nejdůležitější veličina bude účinný průřez, definovat ho jistě není třeba. Budeme ho značit  $\sigma$ . Připomeneme, že čtyřhybnost značíme  $\vec{k}_i = (\vec{\mathbf{k}}_i, k_{0i})$ , kde  $\vec{\mathbf{k}}_i$  je hybnost (index  $i$  se vztahuje k určité částici). Budeme také pracovat s Lorentzovými faktory:  $\gamma_i = \vec{k}_{0i}/m_i$ .

Uvažujeme nyní účinný průřez pro dvě neidentické částice s čtyřhybnostmi  $\vec{k}_1 = (\vec{\mathbf{k}}_1, k_{01})$  a  $\vec{k}_2 = (\vec{\mathbf{k}}_2, k_{02})$ . Může být vyjádřen přes invariantní produkční amplitudu  $T(\vec{k}_1, \vec{k}_2, \alpha)$  jako:

$$\sum_{\alpha} (2\pi)^6 \gamma_1 \gamma_2 \frac{d^6 \sigma}{d^3 \mathbf{k}_1 d^3 \mathbf{k}_2} = |T(\vec{k}_1, \vec{k}_2; \alpha)|^2, \quad (3.1)$$

kde suma je daná přes  $\alpha = [S, M, \alpha']$ , přičemž  $S$  značí totální spin páru,  $M$  jeho projekce a  $\alpha'$  je kvantové číslo ostatních produkovaných částic a zahrnuje integraci přes jejich čtyřhybnosti, přičemž se bere v úvahu zákon zachování energie a hybnosti.

Nyní nás zajímají páry částic (1, 2) produkované s malou vzájemnou relativní rychlostí v procesu s obvyklou hustotou fázového prostoru produkovaných částic. Tyto páry by měly mít velkou rychlost vzhledem ke všem ostatním částicím. Vzhledem k rychlému úbytku FSI s rostoucí relativní rychlostí, můžeme zanedbat FSI ve všech kombinacích kromě (1, 2). To jinak řečeno znamená, že se ovlivňují pouze částice (1, 2) navzájem. Amplitudu procesu potom můžeme zapsat jako:

$$T(\vec{k}_1, \vec{k}_2; \alpha) = T_0(\vec{k}_1, \vec{k}_2; \alpha) + \Delta T(\vec{k}_1, \vec{k}_2; \alpha), \quad (3.2)$$

kde  $T_0(\vec{k}_1, \vec{k}_2; \alpha)$  je produkční amplituda v případě, že neexistuje žádné FSI a  $\Delta T(\vec{k}_1, \vec{k}_2; \alpha)$  představuje příspěvek FSI mezi částicemi 1 a 2.

Rovnici můžeme přepsat jako:

$$\begin{aligned} T(\vec{k}_1, \vec{k}_2; \alpha) &= \int d^4x_1 d^4x_2 \Psi_{\vec{k}_1, \vec{k}_2}^{S(-)}(\vec{k}_1, \vec{k}_2) \Upsilon(\vec{x}_1, \vec{x}_2; \alpha) \\ &= \int d^4x \Psi_{\vec{q}}^{S(-)}(\vec{x}) \tau_P(\vec{x}; \alpha), \end{aligned} \quad (3.3)$$

kde  $\vec{q} = (1 - P^2)q/P^2$ ,  $\vec{q} = \vec{k}_1 - \vec{k}_2$ ,  $\vec{x} = \vec{x}_1 - \vec{x}_2$ ,  $\vec{P} \equiv 2\vec{p} = \vec{k}_1 + \vec{k}_2$  a

$$\Psi_{\vec{k}_1, \vec{k}_2}^{S(-)}(\vec{x}_1, \vec{x}_2) = [\Psi_{\vec{k}_1, \vec{k}_2}^{S(+)}(\vec{x}_1, \vec{x}_2)]^* \quad (3.4)$$

se shoduje s Bethe-Salpeterovou amplitudou ve spojitém spektru. Funkce  $\Upsilon(\vec{x}_1, \vec{x}_2; \alpha)$  reprezentuje produkční amplitudu částic 1 a 2 v časoprostorových bodech  $\vec{x}_1$  a  $\vec{x}_2$ . Druhá rovnost (3.3) vyvstává po integraci v těžišťovém systému přes souřadnici  $\vec{X} = [(\vec{k}_1 \vec{P})\vec{x}_1 + (\vec{k}_2 \vec{P})\vec{x}_2]/P^2$ . Tedy za předpokladu kvazi volného šíření, závislost dvoučásticové amplitudy na hybnosti je dána konvolucí redukované produkční amplitudy:

$$\tau_{\vec{P}}(\vec{x}; \alpha) = \int dX^4 e^{-i\vec{P}\vec{X}} \Upsilon(\vec{x}_1, \vec{x}_2; \alpha) \quad (3.5)$$

a redukované Bethe-Salpeterovy amplitudy  $\Psi_{\vec{q}}^{S(-)}(\vec{x})$ . Tento poslední výraz závisí jen na čtyřvektoru polohy a zobecněné čtyřhybnosti  $\vec{q}$ . Role FSI se demonstruje ve faktu, že při přechodu k prostorové reprezentaci nahrazuje fázové exponenciální funkce Bethe-Salpeterovým amplitudami  $\Psi_{\vec{k}_1, \vec{k}_2}^{S(-)}(\vec{x}_1, \vec{x}_2)$ .

Bethe-Salpeterovy amplitudy je nutné pro případ identických částic symetrizovat (antisymetrizovat):

$$\Psi_{\vec{k}_1, \vec{k}_2}^{S(-)}(\vec{x}_1, \vec{x}_2) \rightarrow \frac{1}{\sqrt{2}} [\Psi_{\vec{k}_1, \vec{k}_2}^{S(-)}(\vec{x}_1, \vec{x}_2) + \Psi_{\vec{k}_2, \vec{k}_1}^{S(-)}(\vec{x}_1, \vec{x}_2)]. \quad (3.6)$$

V tomto případě platí, že  $m_1 = m_2$ ,  $\vec{q} = q$  a  $\vec{X} = (\vec{x}_1 + \vec{x}_2)/2$ . Dříve řečené skutečnosti o FSI platí i pro identické částice. Jenom poznamenejme, že znaménko plus je pro bosony. Pro fermiony by bylo opačné znaménko.

Účinný průřez nabývá tvaru:

$$(2\pi)^6 \gamma_1 \gamma_2 \frac{d^6 \sigma}{d^3 \vec{k}_1 d^3 \vec{k}_2} = \sum_S \int d^4 x_1 d^4 x_2 d^4 x'_1 d^4 x'_2 \rho_{PS}(\vec{x}_1, \vec{x}_2; \vec{x}'_1, \vec{x}'_2) \Psi_{\vec{k}_1, \vec{k}_2}^{S(-)}(\vec{x}_1, \vec{x}_2) \Psi_{\vec{k}_1, \vec{k}_2}^{S(-)*}(\vec{x}'_1, \vec{x}'_2), \quad (3.7)$$

kde funkce

$$\rho_{PS}(\vec{x}_1, \vec{x}_2, \vec{x}'_1, \vec{x}'_2) = \sum_{M, \alpha'} \Upsilon(\vec{x}_1, \vec{x}_2; S, M, \alpha') \Upsilon^*(\vec{x}'_1, \vec{x}'_2; S, M, \alpha') \quad (3.8)$$

reprezentuje elementy nenormalizované dvoučásticové časoprostorové matice hustoty závisící na čtyřhybnosti  $\vec{P}$ .

Jestliže eliminujeme FSI a QS efekty, můžeme definovat referenční účinný průřez jako:

$$(2\pi)^6 \gamma_1 \gamma_2 \frac{d^6 \sigma_0}{d^3 \vec{k}_1 d^3 \vec{k}_2} = \sum_S \int d^4 x_1 d^4 x_2 d^4 x'_1 d^4 x'_2 \rho_{PS}(\vec{x}_1, \vec{x}_2; \vec{x}'_1, \vec{x}'_2) e^{i\vec{k}_1(\vec{x}_1 - \vec{x}'_1)} e^{i\vec{k}_2(\vec{x}_2 - \vec{x}'_2)}. \quad (3.9)$$

Dále přepíšeme

$$\frac{d^6 \sigma}{d^3 \vec{k}_1 d^3 \vec{k}_2} = \frac{d^6 \sigma_0}{d^3 \vec{k}_1 d^3 \vec{k}_2} \sum_S B_S(\vec{k}_1, \vec{k}_2) \langle \Psi_{\vec{k}_1, \vec{k}_2}^{S(-)}(\vec{x}_1, \vec{x}_2) \Psi_{\vec{k}_1, \vec{k}_2}^{S(-)*}(\vec{x}'_1, \vec{x}'_2) \rangle'_{\vec{k}_1 \vec{k}_2 S}. \quad (3.10)$$

Kam jsme vložili částečně zprůměrovaný výraz v  $\langle \dots \rangle$  bilineárních produktů Bethe-Salpeterovy amplitudy.

$B_S$  je statistický faktor pravděpodobnosti spinu páru  $S$  za absence korelačního efektu:

$$B_S(\vec{k}_1, \vec{k}_2) = \frac{\int d^4 x_1 d^4 x_2 d^4 x'_1 d^4 x'_2 \rho_{PS}(\vec{x}_1, \vec{x}_2; \vec{x}'_1, \vec{x}'_2) e^{i\vec{k}_1(\vec{x}_1 - \vec{x}'_1)} e^{i\vec{k}_2(\vec{x}_2 - \vec{x}'_2)}}{\sum_S \int d^4 x_1 d^4 x_2 d^4 x'_1 d^4 x'_2 \rho_{PS}(\vec{x}_1, \vec{x}_2; \vec{x}'_1, \vec{x}'_2) e^{i\vec{k}_1(\vec{x}_1 - \vec{x}'_1)} e^{i\vec{k}_2(\vec{x}_2 - \vec{x}'_2)}}. \quad (3.11)$$

Je vhodné zmínit, že pro nepolarizované částice se spiny  $j_1$  a  $j_2$  dostaneme:

$$B_S = (2S + 1)[(2j_1 + 1)(2j_2 + 1)]^{-1}, \quad (3.12)$$

$$\sum_S B_S = 1. \quad (3.13)$$

Obecně spinové faktory jsou senzitivní k polarizaci částice. Například jestliže dvě částice s polocelým spinem jsou emitovány nezávisle na polarizaci  $P_1$  a  $P_2$ , pak  $B_0 = (1 - \vec{P}_1 \vec{P}_2)/4$  a  $B_1 = (3 + \vec{P}_1 \vec{P}_2)/4$ .



## 3.2 Neinteragující neidentické částice: Časoprostorová koherence

Aby byl lépe vidět význam časoprostorové matice hustoty  $\rho_{PS}(\vec{x}_1, \vec{x}_2; \vec{x}'_1, \vec{x}'_2)$ , zanedbejme FSI efekt a zasubstitujeme Bethe-Salpeterovu amplitudu rovinnou vlnou. Zaměníme integrační proměnné novými:

$$\bar{x}_i = \frac{1}{2}(\vec{x}_i + \vec{x}'_i), \quad (3.14)$$

$$\vec{\epsilon}_i = \vec{x}_i - \vec{x}'_i \quad (3.15)$$

a přepíšeme produkční účinný průřez dvou neidentických částic jako:

$$(2\pi)^6 \gamma_1 \gamma_2 \frac{d^6 \sigma_0}{d^3 \vec{k}_1 d^3 \vec{k}_2} = \sum_S \int d^4 \bar{x}_1 d^4 \bar{x}_2 G_S(\bar{x}_1, \vec{k}_1; \bar{x}_2, \vec{k}_2), \quad (3.16)$$

kde  $\bar{x} = \bar{x}_1 - \bar{x}_2$  a kde

$$G_S(\bar{x}_1, \vec{k}_1; \bar{x}_2, \vec{k}_2) = \int d^4 \epsilon e^{-i(\vec{k}_1 \epsilon_1 - \vec{k}_2 \epsilon_2)} \rho_{PS}(\bar{x}_1 + \frac{\vec{\epsilon}_1}{2}, \bar{x}_2 + \frac{\vec{\epsilon}_2}{2}, \bar{x}_1 - \frac{\vec{\epsilon}_1}{2}, \bar{x}_2 - \frac{\vec{\epsilon}_2}{2}) d^4 \epsilon_1 d^4 \epsilon_2. \quad (3.17)$$

Funkce  $G_S$  se obvykle nazývá emisní funkce a je parciální Fourierovou transformací matice časoprostorové hustoty. V kapitole (2) jsme používali produkční amplitudu  $A(\vec{k}, \vec{x})$ .  $G_S$  je obdobná. Rozdíl je ten, že  $G_S$  je dvoučásticová a  $A(\vec{k}, \vec{x})$  jednočásticová. Tady je vidět velké zjednodušení předchozího zápisu. V něm jsme amplitudu  $A(\vec{k}, \vec{x})$  brali jako danou, a nezabývali jsme se jejími vlastnostmi. Nyní vidíme, kolik kroků bylo zapotřebí, abychom ji odvodili. V předchozích výpočtech jsme neuvažovali šířku zdroje, který nám vyjadřuje proměnná  $\vec{\epsilon}_i$ . V současném zápisu ji uvádíme a integrujeme přes ní a dále ji pak nemusíme uvažovat.

Je vidět, že čím užší je šířka diagonály matice časoprostorové hustoty (šířka  $\vec{\epsilon}_i$  distribuce), tím širší je distribuce čtyřhybnosti. To je ve shodě s Heisenbergovým principem neurčitosti. Jinak řečeno, když souřadnice je přesně lokalizovaná, rozdělení hybnosti je nekonečně široké.

## 3.3 Neinteragující identické částice: QS korelace

První se zaměříme na produkci neinteragujících identických částic. Korelační efekt je tedy ovlivněn jen kvantovou statistikou, respektive její symetrizací. To znamená, že Bethe-Salpeterova amplituda musí být substituována její symetrizovanou variantou rovinných vln, v případě fermionů asymetrickou kombinací. Jako výsledek interference těchto vln se objeví dodatečný člen:

$$(2\pi)^2 \gamma_1 \gamma_2 \frac{d^6 \sigma}{d^3 \vec{k}_1 d^3 \vec{k}_2} = \sum_S \int d^4 \bar{x}_1 d^4 \bar{x}_2 [G_S(\bar{x}_1, \vec{k}_1; \bar{x}_2, \vec{k}_2) + G_S(\bar{x}_1, \vec{k}; \bar{x}_2, \vec{k}) (-1)^S \cos(\vec{q}\vec{x})] \quad (3.18)$$

Hybnost částice mimo hmotnostní slupku (off-mass-shell four momentum)  $\vec{k} = (\vec{k}_1 + \vec{k}_2)/2$  je argumentem emisní funkce  $G_S$  v interferenčním termínu.

Je vhodné definovat korelační funkci  $C(\vec{k}_1, \vec{k}_2)$  jako poměr účinného průřezu  $d^6 \sigma$  a účinného průřezu, který by byl pozorován bez působení QS a FSI.

$$C(\vec{k}_1, \vec{k}_2) = \frac{\frac{d^6 \sigma(\vec{k}_1, \vec{k}_2)}{d^3 \vec{k}_1 d^3 \vec{k}_2}}{\frac{d^6 \sigma_0(\vec{k}_1, \vec{k}_2)}{d^3 \vec{k}_1 d^3 \vec{k}_2}} \quad (3.19)$$

Tuto funkci jsme viděli v (2) a objevovala se v ní produkční amplituda  $A(\vec{k}, \vec{x})$  a distribuční funkce  $f(\vec{k}_1, \vec{x})$ . Dříve jsme ukázali analogii mezi funkcemi  $A(\vec{k}, \vec{x})$  a  $G_S$ .  $C(\vec{k}_1, \vec{k}_2)$  je nyní zadaná pomocí účinných průřezů. Od počátku třetí kapitoly nikde nebyl zmíněn vliv cesty částice na korelační funkci. To je na první pohled překvapující. Když si však vzpomeneme na konečný tvar korelační funkce z druhé kapitoly pro jakýkoliv zdroj, tak jsme v něm také pracovali jen s veličinami týkajícími se zdroje.

V kolizích jader o vysoké energii můžeme zanedbat kinematické omezení a konstruovat referenční rozdělení ve jmenovateli (3.19) používající částice z různých případů se stejnou topologií. V případě zanedbatelné FSI, neexistuje korelace pro neidentické částice:  $C(\vec{k}_1, \vec{k}_2) = 1$ . Pro identické částice korelace vyvstává díky interferenčnímu efektu:

$$\begin{aligned} C(\vec{k}_1, \vec{k}_2) &= 1 + \frac{\sum_S \int d^4 \bar{x}_1 d^4 \bar{x}_2 G_S(\bar{x}_1, \vec{k}; \bar{x}_2, \vec{k}) (-1)^S \cos(\vec{q}\vec{x})}{\sum_S \int d^4 \bar{x}_1 d^4 \bar{x}_2 G_S(\bar{x}_1, \vec{k}_1; \bar{x}_2, \vec{k}_2)} \\ &\equiv 1 + \sum_S B_S (-1)^S \langle \cos(\vec{q}\vec{x}) \rangle''_{\vec{k}_1 \vec{k}_1 S}, \end{aligned} \quad (3.20)$$

kde

$$\langle \cos(\vec{q}\vec{x}) \rangle''_{\vec{k}_1 \vec{k}_1 S} = \langle e^{i\vec{k}_1(\vec{x}_1 - \vec{x}'_2) + i\vec{k}_2(\vec{x}_2 - \vec{x}'_1)} \rangle'_{\vec{k}_1 \vec{k}_1 S} = \langle \cos(\vec{q}\vec{x}) \rangle'_{\vec{q} P S}. \quad (3.21)$$

Faktory  $B_S$  reprezentují pravděpodobnost existence stavu o spinu  $S$ .

### 3.3.1 Předpoklad hladkosti

Uvažujme případ jednoduchého modelu jednoho typu emitoreů přispívajících k výslednému interferenčnímu efektu, které navíc nejsou v relativním pohybu. Tento model

jsme viděli na konci předchozí kapitoly. Potom šířka korelační funkce v oblasti malých  $|q|$  je výhradně určena časoprostorovou vzdáleností mezi jednočásticovými emitory a nezávisí na parametrech  $r_A$  a  $\tau_A$ . Uvažujme rozšíření produkční oblasti ( $r_0^2 \rightarrow r_0^2 + \frac{1}{2}r_A^2, \tau_0^2 \rightarrow \tau_0^2 + \frac{1}{2}\tau_A^2$ ), kde proměnné  $r_A$  a  $\tau_0$  charakterizují časoprostorové rozsahy jednotlivých emitorů. Takové rozšíření je v korelační funkci kompenzováno. To je vidět, jestliže spočítáme korelační funkci přímo z rovnice (3.1), po substituci produkční amplitudy  $T(\vec{k}_1, \vec{k}_1, \alpha)$  symetrizovanou Kopylov-Podgorestkého jednočásticovou amplitudou v hybnostní reprezentaci. Jaký vliv má časoprostorový rozsah jednotlivých emitorů na interferenční efekt? Předpokládáme, že emitory se rozpadají podle jedné univerzální amplitudy a liší se pouze čtyřvektory souřadnic jejich středů. Jestliže je tento předpoklad splněn, časoprostorový rozsah jednotlivých emitorů nemá vliv na interferenční efekt, pakliže překrytí emitorů je dostatečně malé.

Obecně i v případě že emitory jsou nezávislé, částice jsou emitovány emitory různých typů. Korelační funkce závisí na jejich časoprostorovém rozložení, tedy na parametrech  $r_A$  a  $\tau_A$ . Efekt časoprostorového rozložení je zanedbatelný, jestliže jsou splněny tyto podmínky:

$$\frac{r_A^2}{2} \ll r_0^2, \quad (3.22)$$

$$\frac{\tau_A^2}{2} \ll \tau_0^2. \quad (3.23)$$

Tyto podmínky zaručují hladkou závislost emisní funkce  $G_S(\vec{x}_1, \vec{k}_2; \vec{x}_2, \vec{k}_2)$  na čtyřhybnostech. Hladkost funkce pro nás obvykle znamená, že funkce má spojitě derivace všech řádů. Požadavek hladkosti v tomto případě dostává navíc podmínku, že funkce se v závislosti na proměnných mění zvolna. Můžeme tedy zanedbat závislost funkce  $G_s$  na  $\vec{q}$  v oblasti interferenčního efektu, která je charakterizovaná inverzní časoprostorovou vzdáleností produkčních bodů částice. Funkce  $C(\vec{k}_1, \vec{k}_2)$  nabývá tvaru:

$$C(\vec{k}_1, \vec{k}_2) \doteq 1 + \sum (-1)^S B_S \langle \cos(qx) \rangle_{qPS}. \quad (3.24)$$

Platí:

$$\langle \cos(\vec{q}\vec{x}) \rangle_{qPS} = \frac{\int d^4x_1 d^4x_2 G_S(\vec{x}_1, \vec{k}_1; \vec{x}_2, \vec{k}_2) \cos(\vec{q}\vec{x})}{\int d^4x_1 d^4x_2 G_S(\vec{x}_1, \vec{k}_1; \vec{x}_2, \vec{k}_2)}. \quad (3.25)$$

Proměnná  $q$  se může podstatně měnit, aniž by se to adekvátně projevilo na emisní funkci. Jestli je rozdělení hybnosti nekonečně široké, rozdělení polohy je prakticky  $\delta$  funkce. Aby částice byla absolutně přesně lokalizovaná je nemožné, ale rozdělení může být velice úzké. Předpoklad hladkosti nám tedy říká, že matice hustoty je diagonální, tj.  $\epsilon \rightarrow 0$ .

### 3.4 Interagující částice: FSI korelace

Je zřejmé, že předpoklad hladkosti umožňuje napsat účinný průřez přes emisní funkci  $G_S(\vec{x}_1, \vec{k}_1; \vec{x}_2, \vec{k}_2)$  i v případě interagujících částic. Jestliže zapíšeme fázový

faktor jako  $e^{i\vec{P}(\vec{X}-\vec{X}')} = e^{i(\vec{k}_1-\vec{q}/2)\vec{\epsilon}_1+i(\vec{x}_2+\vec{q}/2)\vec{\epsilon}_2}$  a používajíce předpoklad hladkosti k zanedbání  $\tilde{q}$  ve srovnání s  $k_{1,2}$  a k substituování v amplitudě  $\psi_{\vec{q}}^{S(+)}$  souřadnic  $x = \bar{x} + (\vec{\epsilon}_1 - \vec{\epsilon}_2)/2$  a  $\vec{x}' = \bar{x} - (\vec{\epsilon}_1 - \vec{\epsilon}_2)$  jejich průměrnou hodnotou  $\bar{x}$ , můžeme přepsat rovnici (3.7) na:

$$(2\pi)^6 \gamma_1 \gamma_2 \frac{d^6 \sigma}{d^3 \vec{k}_1 d^3 \vec{k}_2} = (2\pi)^6 \gamma_1 \gamma_2 \frac{d^6 \sigma_0}{d^3 \vec{k}_1 d^3 \vec{k}_2} \sum_S B_S \langle |\psi_{\vec{q}}^{S(+)}(\bar{x})|^2 \rangle_{\bar{q}PS}, \quad (3.26)$$

kde  $\sigma_0$  je produkční účinný průřez neinteragujících částic. Zprůměrování  $\langle \rangle_{\bar{q}PS}$  jsme viděli dříve. Korelační funkce, která je definovaná jako zlomek  $\frac{d^6 \sigma}{d^6 \sigma_0}$  nabývá tvaru:

$$C(\vec{k}_1, \vec{k}_2) \doteq \sum_S B_S \langle |\psi_{\vec{q}}^{S(+)}(\bar{x})|^2 \rangle_{\bar{q}PS}. \quad (3.27)$$

Připomeňme, že pro identické částice je Bethe-Salpeterova amplituda symetrizovaná.

### 3.4.1 Předpoklad současnosti

Pro neinteragující částice, nesymetrizovaná Bethe Salpeterova amplituda  $\psi_{\vec{q}}^\pm = e^{-ik^*r^*}$  je nezávislá na relativním emisním čase  $t^*$  v těžišтовém systému. Naproti tomu amplituda dvou interagujících částic přímo závisí na čase  $t^*$ . Interakční efekt mizí až pro  $|t^*| \rightarrow \infty$ . Dá se ale ukázat, že efekt nestejného času se dá zanedbat za podmínky:

$$|t^*| \ll m(t^*)r^{*2}, \quad (3.28)$$

kde  $m(t^* > 0) = m_2$  a  $m(t^* < 0) = m_1$ . Za této podmínky můžeme použít předpoklad současnosti  $t^* = 0$  a substituovat Bethe-Salpeterovu amplitudu běžnou nerelativistickou dvoučásticovou vlnovou funkcí. Podmínka  $|t^*| \ll m(t^*)r^{*2}$  je obvykle splněna pro těžké částice jako kaony nebo nukleony. Ale i pro piony tato aproximace jen mírně přeceňuje (o pár procent) příspěvek silné FSI k účinnému průřezu produkce.

Efekt různého emisního času roste s klesající rychlostí párů a s rostoucí dobou života emitorek. Pro dostatečně velké rychlosti a radius  $r_0 > 1$  fm, efekt je spíše malý, nepřesahující 5% příspěvku FSI v oblasti malého  $k^*$ . Jeho příspěvek korelační funkci je několik promile.

Jestliže vezmeme do úvahy předpoklad hladkosti a aproximaci stejného času (s přesností na několik promile), můžeme přepsat rovnici (3.26) na:

$$\gamma_1 \gamma_1 \frac{d^6 \sigma_0}{d^3 \vec{k}_1 d^3 \vec{k}_2} \doteq \gamma_1 \gamma_1 \frac{d^6 \sigma_0}{d^3 \vec{k}_1 d^3 \vec{k}_2} \sum_S B_S \langle |\psi_{-k^*}^s(r^*)| \rangle_{\bar{q}PS}. \quad (3.29)$$

Tím jsme dosáhli toho, že Bethe-Salpeterova amplituda byla nahrazena stacionárním řešením rozptylového problému Schroedingerovy rovnice s Coulombickým potenciálem, tj:  $\psi_{-k^*}^s(r^*)$ .

Rozšíření teorie, kterou jsme viděli v této a předchozí kapitole, lze nalézt v (6).

### 3.4.2 Simulace korelační funkce

Tato podkapitola bude sloužit jako uzavření teorie a přímá příprava k programu. Zde rozebereme, jak budeme korelační funkce simulovat. Jen bych připomněl, že stále se nacházíme v oblasti teorie a jestli se zde objeví něco málo o praktické aplikaci, je to jen z důvodu ilustrace, neboť její místo je až v kapitole o programu.

Korelační funkce se simulují pomocí dělení dvou spekter té samé proměnné, vážené a nevážené. Když řekneme, že simulujeme vliv jistého vlivu, znamená to, že najdeme vztah vyjadřující onen jev a ten poté použijeme k získání váhy zapisované hodnoty, což je v našem případě velikost vektoru vzájemné hybnosti v těžišťové soustavě.

#### Kvantová statistika

První zmíníme korelační funkce jen za vlivu kvantové statistiky. Důvod jejího prvenství je její jednoduchost. Vliv kvantové statistiky je vystižen rovnicí (3.24), kterou rozvádí rovnice (3.25). Tyto rovnice vypadají poměrně složité, ale je to jednoduché vystředování.

Vystředování se provede velmi snadno a suma spinových faktorů je konečná a nezávislá na středování. Proto se dá spočítat od středování odděleně. Potom tedy dostaneme jen součin sumy a onoho středování.

Situace se komplikuje, jestliže uvažujeme korelační funkci za vlivu pouze FSI. Abychom postupovali metodicky, budeme první uvažovat korelační funkci jen za vlivu jednoho FSI a to konkrétně silné interakce.

#### Silná interakce

Než přistoupíme k rozboru, rád bych připomněl, že v programu uvažujeme předpoklad hladkosti a současnosti. Proto jsme přešli od čtyřvektorů hybnosti a polohy k "třívektorům". Vektor vzájemné hybnosti v těžišťové soustavě budeme značit jako  $\vec{\mathbf{q}}$  a jeho velikost  $q$ . Vektor  $\vec{\mathbf{k}} = \vec{\mathbf{q}}/2$  a jeho velikost značíme  $k^*$ . Vzájemnou polohu produkčních bodů v těžišťové soustavě značíme  $\vec{\mathbf{r}}$  a jeho velikost  $r^*$ .

V případě, kdy dvoučásticová FSI je dominována jen silami malého dosahu (což silná interakce nepochybně je), v oblasti malých vzájemných hybností v dvoučásticové interakci dominují "s vlny" (s-waves). Bez hlubšího rozboru oné skutečnosti řekneme, že efekt FSI je určen asymptotickým chováním rozptýlené vlny vně oblasti silné interakce  $r^* > d$ , kde  $r^*$  je vzdálenost produkčních bodů v těžišťové soustavě a  $d$  je oblast silné interakce. Tuto skutečnost vystihneme rovnicí:

$$\Delta\psi_{-\vec{\mathbf{k}}}(\vec{\mathbf{r}}) = f(k^*)e^{ik^*r^*}/r^*. \quad (3.30)$$

Amplituda s-vlny, což je funkce  $f(k^*)$  závisí jen na velikosti vektoru  $\vec{\mathbf{k}}$ . Za splnění jistých podmínek ji lze vyjádřit jako:

$$f = (K^- - ik^*)^{-1}. \quad (3.31)$$

Jak jsme zmínili, uvažujeme jen oblast malých  $q$ , tedy oblast malých  $\vec{k}$ . Proměnnou  $K$  je možno vyjádřit různými způsoby. Jestliže ji chceme rozšířit do oblasti vyšších energií, vyjádříme ji následujícím zápisem:

$$K = \frac{2}{\sqrt{s}} \frac{s_{th} - s_0}{s - s_0} \sum_{j=0}^3 A_j x^{2j}, \quad (3.32)$$

kde

$$x = \frac{q}{\sqrt{s_{th}}} \quad (3.33)$$

a  $s = (w_1^* + w_2^*)^2$ ,  $w_{1,2}^* = (m_{1,2}^2 + k^{*2})^{1/2}$  a  $s_{th} = (m_1 + m_2)^2$ . Koefficienty  $A_i$  a  $s_0$  jsou uvedeny v programu.

Nyní přistoupíme k funkci, která pro nás bude mít zvláštní důležitost, neboť jí budeme v programu vážit zápisy do histogramů.

$$\psi_{-\vec{k}}(\vec{r}) = e^{i\delta_0} \sqrt{A_c(\eta)} [e^{-i\vec{k}\vec{r}} + f(k^*) e^{ik^*r^*}/r^*] \quad (3.34)$$

Kde  $\delta_0 = \arg\Gamma(1 + i\eta)$  a  $\eta = (k^*a)^{-1}$ . Proměnná  $a$  je Bohrovův poloměr, který nabývá hodnot  $-387.5$  pro případ interakce pionů stejného náboje a  $-387.5$  při interakci pionů rozdílného náboje. Poslední je Coulombický penetrační faktor  $A_c = 2\pi\eta(e^{2\pi\eta} - 1)^{-1}$ .

## Kompletní FSI

Zápis pro korelační funkci za vlivu Coulombické interakce i silné interakce je složitější:

$$\psi_{-\vec{k}}(\vec{r}) = e^{i\delta_0} \sqrt{A_c(\eta)} [e^{-i\vec{k}\vec{r}} F(-\eta, 1, i\varepsilon) + f_c(k^*) \frac{\tilde{G}(\rho, \eta)}{r^*}], \quad (3.35)$$

kde  $\varepsilon = \vec{k}\vec{r} + k^*r^*$  a  $\rho = +k^*r^*$ .

Funkce  $F(-\eta, 1, i\varepsilon)$  je konfluentní hypergeometrická funkce a  $\tilde{G} = \sqrt{A_c}(G_0 + iF_0)$  je kombinací regulární ( $F_0$ ) a singulární ( $G_0$ ) Coulombické vlnové funkce.

Funkce  $f_c(k^*)$ , která zde vyjadřuje silnou interakci, není shodná s funkcí  $f(k^*)$  z předchozí podkapitolky, jak vidíme z následující rovnice:

$$f_c(k^*) = (K^{-1} - \frac{2\chi(\eta)}{a})^{-1}, \quad (3.36)$$

kde

$$\chi(\eta) = h(\eta) + \frac{iA_c(\eta)}{2\eta} \quad (3.37)$$

a

$$h(\eta) = \frac{\psi(i\eta) + \psi(-i\eta) + \ln(\eta^2)}{2}. \quad (3.38)$$

Poslední výraz, který neznáme je  $\psi(i\eta)$ , což je digamma funkce v komplexní rovině.

## Kompletní FSI a kvantová statistika

V tomto případě musíme funkci  $\psi_{-\vec{k}}(\vec{r})$  symetrizovat.

# Kapitola 4

## Monte Carlo

Metoda Monte Carlo představuje velmi mocný nástroj při simulacích. Nepoužívá se jen v oblasti vědy, ale našla si rozsáhlé uplatnění i v soukromém sektoru. Jako příklad můžeme zmínit oblast financí. Pole jejího použití se nadále rozšiřují. Roku 1930 s ní experimentoval Enrico Fermi při studiu difuze neutronů. Její moderní verze byla vynalezena Stanislawem Ulamem v pozdních čtyřicátých letech minulého století při práci na jaderných zbraních v Los Alamos National Laboratory.

Existují procesy, jevy, které nelze popsat analyticky. Jestliže jsou to jevy, které se skládají z mnoha dílčích událostí, nabízí se metoda Monte Carlo. Nemusíme umět popsat jev deterministickými rovnicemi, ale můžeme ho popsat na základě jednotlivých událostí, které přispívají k sumárnímu efektu. Myšlenkou je generovat soubor individuálních událostí popisující chování systému. Klíčové je, abychom měli dostatečné navzorkování všech možných jevů přispívajících k sumárnímu efektu.

V programu používáme metodu Monte Carlo. Podle náhodných rozdělení generujeme složky čtyřvektorů hybnosti a polohy jednotlivých částic.

### 4.1 Dostatečné navzorkování

Musíme mít dostatečné množství všech možných případů, které se mohou vyskytnout. Korelační funkce pracuje s interakcemi mezi dvěma částicemi. Interakce bude ovlivňovat funkce, které slouží k získání vah. Pro dostatečné navzorkování je nezbytný dostatečně velký počet pokusů. Pracoval jsem se sto miliony pokusů (případů dvou částic). Sto milionů je dostatečný statistický soubor a přitom nechává výpočetní čas snesitelný.

Zajistit počet pokusů nestačí pro zajištění dostatečného navzorkování všech možných případů. Uvedu příklad: Používám Gaussovo rozdělení  $q$ . V tom případě je evidentní, že nejvíc budou zastoupené biny kolem střední hodnoty, tedy parametru  $\mu$ . V nejnižších, nebo naopak nejvyšších binech se může stát (záleží na parametru  $\sigma$ ), že dostaneme malý počet "zásahů". V konkrétním případě této práce šlo o oblast malých  $q$ , která je pro nás nejzajímavější. Proto musíme vždy kontrolovat, jestli jsou dostatečně navzorkované všechny možnosti. To lze například dělat vykreslením

spektra  $q$ , velikosti vzájemné hybnosti dvou částic v těžiškové soustavě.



# Kapitola 5

## Program

Kompletní výpis programu nalezneme v příloze (A). Než přistoupíme k tvorbě programu, měli bychom říci, co konkrétně od programu potřebujeme a s čím jsme se setkávali při jeho tvorbě.

Tato kapitola bude rozvržena následujícím způsobem: Prvně si řekneme o simulaci podmínek experimentu, dále rozebereme způsob tvorby korelační funkce.

Dále zmíníme numerickou knihovnu `GSL` a pracovní prostředí `Root`.

Poté přistoupíme k analýze kódu.

V programu se používá formalismus částicové fyziky, kde  $c = h = 1$ . Jednotky jsou  $fm$  a  $MeV$ . Mezi nimi platí převodní vztah  $MeV = fm/197.3$ .

### 5.1 Simulace podmínek experimentu

V programu budeme současně simulovat dva piony. Dále budeme zkoumat jejich vzájemné působení. To získáme pomocí funkcí z podkapitolky (3.4.2).

V podmínkách experimentu máme zdroj, ze kterého vyletují částice. V sekci (3.1) jsme viděli, že nás zajímají částice produkované s malou vzájemnou hybností. Dále, jak bylo řečeno, v našem programu uvažujeme předpoklady současnosti a hladkosti. Budeme uvažovat vzájemnou interakci jen mezi produkovanými částicemi. Máme tedy případ dvou částic. Je evidentní, že případy se mohou lišit. Budeme chtít dostatečně pokrýt všechny možné varianty (pro dostatečné navzorkování), proto těchto případů budu generovat velmi mnoho.

Dvě částice budou mít jistou hmotnost, hybnost a bod vzniku, což bude ovlivňovat korelační funkci. Rozsah jejich hybností závisí na podmínkách experimentu. Nejvíce informací pro nás nesou jevy, které měly malou vzájemnou hybnost. V naší práci se nezabýváme způsobem vzniku částic. Bereme, že máme dvě částice jistého druhu opouštějící zdroj s jistou energií a směrem.

U čtyřvektoru hybnosti budeme generovat složky hybnosti. Ty budou mít rovnoměrné rozdělení. Interval je volitelný, ale velikost vektoru vzájemné hybnosti ( $q$ ) je pro nás

zajímavá v intervalu 0-200. Proto nemá smysl volit interval složek hybnosti nesympetně široký. Generujeme tedy složky hybnosti a z nich a z klidové hmotnosti spočítáme energii.

Obě částice mají jistý bod vzniku, časoprostorové souřadnice. Čas bereme jako nulový, složky vektoru polohy generujeme pomocí náhodných čísel.

Zdroj částic je stacionární. Při experimentu jsou částice generovány mnoha emitory, které dohromady tvoří zdroj. Vezměme zdroj po složkách. Potom můžeme říci, že hustota emitů kolem středu se dá popsat Gaussovým rozdělením. To platí pro všechny tři složky. Je tedy evidentní, že pravděpodobnost výskytu částice je největší u středu zdroje.

Proto i jednotlivé složky vektoru polohy vzniku budu generovat pomocí čísel z Gaussova rozdělení.

Budeme muset zavést ještě čtyřvektor tzv. "rozmazané hybnosti". V reálných podmínkách experimentu nejsou energie částic měřeny s naprostou přesností. Podmínky experimentu, použité detektory, to vše ovlivňuje měření. Proto, i kdybychom měli částice několika diskrétních energií, naměřené spektrum nebude diskrétní, ale spojitě- rozmazané. Tato funkce nám tedy simuluje vliv detektorů a více nás přibližuje reálným podmínkám měření. Jednotlivé složky hybnosti budu rozmazávat pomocí Gaussova rozdělení.

S částicemi pracujeme v jejich těžišťové soustavě, takže musíme provést jejich transformace. Budeme provádět relativistické transformace. Vektory hybností mají maximální velikost kolem  $100 \text{ MeV}$ , což je dost na to, aby se relativistické jevy uplatnily. Proto u každé částice budeme mít i čtyřvektory transformované polohy a hybnosti. Transformovat budu i rozmazanou hybnost.

Zopakuji to nejdůležitější z této podkapitolky. Budu mít případ dvou částic. Identita částic nám určí jejich klidovou hmotnost. U obou musím generovat jejich vlastnosti. To je čtyřvektor polohy a hybnosti. Ty si transformujeme do těžišťové soustavy.

Zaměříme se nyní na způsob generování složek vektorů. Nejprve se věnujme generování hybnosti.

První způsob je generování složek hybnosti jednotlivých částic v laboratorní soustavě. Z těchto složek se potom dopočítá velikost vektoru hybnosti. Jak jsem již zmínil, v korelační funkci vystupují čtyřvektory hybnosti a polohy v těžišťové soustavě. Proto musíme všechny složky čtyřvektoru transformovat Lorentzovými transformacemi. Z hybností jednotlivých částic spočítám vzájemnou hybnost.

Druhý případ je generování velikosti vektoru vzájemné hybnosti v těžišťové soustavě. Z ní získám jednotlivé složky vektoru vzájemné hybnosti.

Je evidentní, že druhý způsob generování je jen pomůcka, ale neodpovídá realitě experimentu. Generování přímo složek vektoru vzájemné hybnosti je možný přístup, protože do korelační funkce vstupuje vektor vzájemné hybnosti. Nezáskáme ho tedy zprostředkovaně, ale přímo. Výhoda této pomůcky je ta, že dostaneme rozdělení hybnosti, které nám vyhovuje. Uvedu případ: Používáme rovnoměrné rozložení složek hybností v laboratorní soustavě. Rozdělení velikosti vektoru  $q$  v těžišťové sou-

tavě bude mít přibližně Gaussovský tvar. Tím v binech u nejmenšího  $q$  (které nás nejvíce zajímají) dostaneme malou statistiku a tím se nám zvětší chyba. Ve středu je naopak statistika neúžitečně velká. Proto si budeme volit rovnoměrné rozdělení a tím zvýšíme statistiku v krajních binech. Rozdělení hybnosti "vyhladíme". Toto je velice účinný prostředek, neboť malou statistiku v krajních binech bychom jinak mohli zvýšit jen počtem pokusů, což by nám silně zvyšovalo výpočetní čas. Proto generování rovnoměrného rozložení  $q$  je velmi účinná pomůcka.

Co se týče generování čtyřvektorů polohy, používám dvě možnosti. První je generování složek třívektoru polohy v klidové soustavě. Potom musím čtyřvektor transformovat do těžišťové soustavy.

Druhý způsob je generování složek vektoru vzájemné polohy produkčních bodů v těžišťové soustavě. Generování přímo vektoru  $r^*$  je výhodné používat v případě složitých zdrojů při studiu jisté kinematické oblasti.

Stejně jako u hybností platí, že pouze první způsob odpovídá podmínkám experimentu. Druhý způsob je pouze aproximace.

Generování přímo složek vzájemných vektorů se může jevit jako pomůcka, která není obvykle používaná. Program, ze kterého jsem vycházel, neobsahoval generování složek vektorů jedné částice v laboratorní soustavě. Generoval přímo vzájemné vektory. Proto se dá říci, že generování v laboratorní soustavě je originální.

## 5.2 Tvorba korelační funkce

Korelační funkce se tvoří pomocí dělení dvou spekter. Jedná se o spektra proměnné  $q$ , což je velikost vektoru vzájemné hybnosti dvou částic v těžišťové soustavě. Jedno spektrum je vážené a druhé nevážené. Vážené spektrum představuje spektrum za působení kvantové statistiky, silné interakce, Coulombické interakce, nebo jejich libovolné kombinace. Interakci budeme simulovat pomocí vah z kapitoly (3.4.2). Vážit budeme pomocí výrazů, které jsme zmiňovali v podkapitole [3.4.2]. Je evidentní, že v programu budeme pracovat s histogramy. V programu budou plněny a zpracovány budou v `Rootu`.

V programu jsem simuloval případ dvou částic, které byly produkovány ve stejný čas ve zdroji. U těch jsem simuloval všechny jejich vlastnosti. Používáme metodu Monte Carlo. Těchto případů jsem si vytvořil dostatečné množství a u každého jsem si spočítal  $q$  a váhy. Proměnnou  $q$  jsem zapisoval do histogramů a tím jsem dostal vážené a nevážené spektrum.

Nejjednodušší je výpočet váhy pro kvantovou statistiku. Postupujeme podle rovnice (3.24).

Pro případ silné interakce je situace poněkud složitější. Tehdy budeme používat výrazy v sekci (3.4.2) v podkapitole "Silná interakce". Výpočet bude trochu delší a komplikovanější. V rovnici (3.32) se objevují koeficienty  $A_j$  a koeficient  $s_0$ . Jejich hodnoty jsou:

1.  $A_2[0]=0.22$
2.  $A_2[1]=0.268$
3.  $A_2[2]=-0.0139$
4.  $A_2[3]=-0.00139$
5.  $s_02=36770000$

Tyto koeficienty by platily jen pro případ pionů stejného znaménka.

V případě, kdy máme kompletní FSI, se výpočet silné interakce liší. V podkapitole 3.4.2 vidíme, že funkce  $f_c(k^*)$  se neshoduje s funkcí  $f$ . Počítá se podle (3.36). Opět v nich vystupuje parametr  $K$ .

V případě interakce  $\pi^+ \pi^-$  nemůžeme počítat jen s uvedenými koeficienty. V případě interakce identických pionů jde o čistě izospinový stav, kdy se izospin rovná dvěma. Pro interakci mezi  $\pi^+ \pi^-$  jde o kombinaci dvou izospinových stavů, dvou a nuly. Ty dva stavy musím vážit. Pro případ  $pi^+ pi^-$  se funkce  $f_c(k^*)$  spočítá jako:

$$f_c(k^*) = 0.66666 * f_{c1}(k^*) + 0.33333 * f_{c2}(k^*) \quad (5.1)$$

Funkce  $f_{ci}(k^*)$  se spočítají podle vzorce (3.31). Pro  $f_{c2}(k^*)$  platí parametry  $A_{2j}$ , pro  $f_{c1}(k^*)$  používáme parametry  $A_{1j}$ .

1.  $A_1[0]=-0.0444$
2.  $A_1[1]=-0.0857$
3.  $A_1[2]=-0.00221$
4.  $A_1[3]=-0.000129$
5.  $s_01=21620000$

Dále je případ korelační funkce za vlivu kompletního FSI a kvantové statistiky. Ta se obdrží symetrizací amplitudy  $\psi_{-\vec{k}}(\vec{r})$  z rovnice (3.35).

Pro případ korelační funkce jen za vlivu Coulombické interakce můžeme vzít jako váhu výraz z rovnice (3.35), kde  $f_c(k^*) = 0$ .

Spektra zapisujeme do histogramů, proto je zmíníme v následující kapitole.

## 5.3 Histogramy

Histogramy pro zaznamenání spekter  $q$  budou mít rozsah na ose  $x$  0- 50 MeV a osa  $x$  bude rozdělena na padesát binů. V případě korelační funkce tvořené za vlivu kompletní FSI a kvantové statistiky budeme mít rozsah osy  $x$  0- 200 MeV. Důvod je ten, že jevy, které nás zajímají, se dějí při větším  $q$ .

V histogramech nebudeme zaznamenávat jen nevážené spektrum  $q$  a spektrum vážené vahou pro získání korelační funkce. Budeme spektrum vážit i členy, které vystupují v korelačních funkcích. Důvod je jednoduchý a to ten, že musíme vědět, jestli se nám členy chovají tak, jak čekáme. Jinak řečeno, ty histogramy budou sloužit pro průběžnou kontrolu.

Zavedeme si tedy několik histogramů, do kterých zaznamenáme spektrum  $q$  a do každého z nich ho budeme zapisovat s jinou vahou.

Budeme používat histogramy pro záznam spektra dalších veličin, což bude například  $r$ , velikost vektoru vzájemné polohy bodů vzniku částic. To nám slouží opět pro účely kontroly.

Abych shrnul tuto kapitolku a načtrl postup tvorby programu, zopakuji čtyři základní kroky. První je generování částic pomocí Monte Carla. V druhém kroku (ve většině funkce `main`), budeme počítat váhy. Za třetí budeme plnit histogramy, do kterých budeme zapisovat vážené a nevážené spektrum  $q$ . Předchozí tři kroky se týkají programu. Tvorba samotné korelační funkce, tzv. dělení spekter, se děje v `Rootu` a představuje poslední, čtvrtý krok.

## 5.4 Root a GSL

Jak již bylo řečeno v úvodu, program je psán v jazyce C++. Výstupem programu jsou histogramy. V programu se také používají funkce z `GSL`. Používají se také knihovny z pracovního prostředí `Root`. Proto je namístě, abychom řekli pár slov o `GSL` a `Rootu`.

`GSL` je numerická knihovna pro jazyky C a C++. Poskytuje velké množství matematických aplikací, jako jsou třeba generátory náhodných čísel. V tomto programu jsou použité matematické funkce z `GSL`, například Coulombické vlnové funkce. Na webové stránce [4] nalezneme více o `GSL`.

`Root` je pracovní prostředí vyvíjené laboratoří v CERN pro potřeby experimentální částicové fyziky. Není účelem této práce podrobněji rozebírat tento software, jen zmiňme to nejdůležitější. `Root` byl vyvíjen pro práci s daty, jejich ukládání, přístup k nim a jejich zpracování a k zobrazování výsledku. Práce s daty je jeho velmi silnou stránkou a je hojně využívána. V průběhu času se `Root` velmi rozrostl a zabudoval do sebe další aplikace. Informace o `Rootu` je možné najít na webové stránce [3]. Důvod, proč jsem používal právě `Root` při této práci je, že mi umožňoval efektivní vykreslení a práci s histogramy. Další důvod je, že `Root` výborně spolupracuje s C++ a oba programy mají stejnou syntaxi. Další důvod použití `Rootu` je ten, že moje práce souvisí s částicovou fyzikou, v jejímž světě je `Root` používáný. V programu uvidíme, že jsou používány třídy z `Rootu`. Na tomto místě zmíníme jednu z věcí, které nám `Root` poskytuje a to jsou generátory náhodných čísel.

## 5.5 Generátory náhodných čísel

Metoda Monte Carlo je založená na náhodných jevech. V našem případě jsou to náhodná čísla. Tři nejobvyklejší způsoby generování náhodných čísel jsou: fyzikální generátor, například získávání pomocí bílého šumu, tabulky náhodných čísel, nebo pseudonáhodná čísla, tj. posloupnost čísel generovaná počítačem, která splňuje pravidla náhodného rozdělení. Používat budeme třetí způsob.

V programu budeme generovat náhodná čísla dvěma způsoby. První z nich je generátor, který jsem si sám napsal. Je to funkce `double rnd(m1,m2)`. Funkce `rnd(m1,m2)` generuje čísla rovnoměrného rozložení mezi body `m1` a `m2`. Tento generátor je dosti primitivní. Je používán na generování hybností. Jednotlivé složky hybnosti mají rovnoměrné rozložení a pro tyto účely je funkce `rnd` dostačující. Jeho výhodou je velká rychlost. Funkci uvedu na následujících řádcích:

```
double rnd( double m1,double m2)
{
    double intrv=m2-m1;
    double res=((double)rand()/(double)RAND_MAX)* intrv + m1;

    return res;
}
```

Dále bychom náhodná čísla mohli generovat generátory z `GSL`, nebo z `Rootu`. Osobně jsem volil generátory `Rootu`. To je například generování čísel z Gaussova rozdělení, které se objevuje například v konstruktoru. Výhody těchto generátorů z `TRandom3.h` jsou, že to jsou "Mersenne Twister" generátory, které jsou rychlé a mají periodu kolem  $2^{19937}$ . `Root` nabízí i jiné třídy, ale tato je pro statistické výpočty nejlepší.

## 5.6 Analýza kódu

### 5.6.1 Hlavičkové soubory, definice konstant, pomocné funkce

V této části budeme analyzovat program. První krok je, podívat se na vložené hlavičkové soubory. Ty jsou rozděleny na tři skupiny. První skupina jsou běžně používané hlavičkové soubory a není nutné se jimi zde zabývat. Druhá skupina jsou hlavičkové soubory, které umožňují používat aplikace `Rootu`. Všimněme si, že všechny začínají písmenem "T".

`TRandom3.h` je nezbytný pro generování náhodných čísel z aparátu `Rootu`, jak jsme zmínili v kapitole (5.5).

Hlavičkový soubor `TFile.h` mi umožňuje vytvářet `Rootovský` soubor, do kterého můžu ukládat výsledky ve formě `TTree`. `TTree.h` zprostředkovává `Tree`, stromovou strukturu ukládání dat. `TMath.h` obsahuje základní matematické funkce.

TComplex.h zprostředkovává komplexní čísla, která jsou nezbytná pro popis amplitud. TH1.h a TH2.h obsahují jednodimenzionální a dvoudimenzionální histogramy.

Poté následují hlavičkové soubory z GSL. Header file gsl/gsl\_sf\_coulomb.h umožňuje používání Coulombických vlnových funkcí.

Dále definujeme několik konstant. Konstanta  $N$  udává počet simulovaných případů. Zvyšující se počet samozřejmě prodlužuje dobu běhu programu, ale je nezbytný pro dostatečný statistický soubor. Pod sto tisíc případů je statistický soubor naprosto nedostatečný. Dále si definuji  $\pi$  a Bohrov poloměr, který se liší pro různé případy. Bohrov poloměr se rovná  $-387,5$  pro kombinaci  $\pi^+ \pi^-$  a  $387.5$  pro případ repulse.

Funkce double  $rnd(m1, m2)$  je generátor náhodných čísel, který jsme popsali v (5.5).

Funkce double  $rozmazat(a)$  slouží k rozmazání hodnoty  $a$ .

## 5.6.2 Struktura na uložení vlastností jedné částice

K popisu částice nám poslouží struktura "částice".

```
struct castice
{
    double hmotnost;
    double poloha[4];
    double polohaT[4];
    double hybnost[4];
    double hybnostT[4];
    double hybnostR[4];
    double hybnostRT[4];
};
```

Částice má jistou hmotnost (i kdyby byla nulová). Potom má časoprostorový bod vzniku. K popisu čtyřvektoru vzniku používáme pole  $poloha[4]$ . První složka je čas, tři ostatní udávají vektor polohy. Polohu generujeme v laboratorní soustavě, ze které budeme přecházet do těžišťové relativistickými transformacemi. Proto zavedeme pole  $polohaT[4]$ , do kterého budeme zaznamenávat transformovaný čtyřvektor. Druhá složka pole náleží ose  $x$ , další pak ose  $y$  a poslední ose  $z$ .

Stejná situace je u hybnosti, kterou simulujeme polem  $hybnost[4]$ , kde  $hybnost[0]$  je energie a ostatní složky pole udávají složky hybnosti. Jednotlivé komponenty pole jsou přiřazeny osám stejně jako v případě polohy. Pole  $hybnostT[4]$  je transformovaná hybnost. Potom použijeme pole  $hybnostR[4]$  a  $hybnostRT[4]$  pro rozmazanou hybnost.

### 5.6.3 Třída *Pripad*

V programu simulujeme dříve zmiňovaný případ dvou částic. Tento případ budeme simulovat třídou "Pripad".

```
class Pripad
{
    public:
        Pripad();
        ~Pripad();

        castice prvni;
        castice druha;
        double V[3];
        double gama[3];
        double hybnostT[3];
        double hybnostRT[3];
        double poloha[3];
        double polohaT[3];
        double q;
        double k;
        double qR;
        double r;
        double rNT;
        double skalarKR;
};
```

Třída *Pripad* je uvedena na konci kapitoly. Třída obsahuje dvě struktury *castice*, pojmenované jako "prvni" a "druha". Potom obsahuje dvě pole:  $V[3]$  a  $gama[3]$ . První pole obsahuje vektor rychlosti, kterou se těžišťová soustava pohybuje vůči laboratorní. Pole *gama* se vztahuje k členu *gama*, vystupujícímu v Lorentzovských transformacích.

Pole  $hybnostT[3]$  a  $hybnostRT[3]$  představují vektor vzájemné hybnosti a rozmazané hybnosti v těžišťové soustavě. Vektor rozmazané hybnosti se stanoví jako rozdíl vektorů rozmazaných hybností jednotlivých částic. Jak vidíme, ony vektory jsou hybnosti, ne čtyřhybnosti. Připomínám, že považujeme předpoklad současnosti za platný. V korelační funkci nevystupují ani vzájemné hybnosti, ale často se objevuje například skalární součin vektorů transformované hybnosti a polohy. Je to skalární součin "třívektorů". Proto není nutné zavádět čtyřvektory. V korelační funkci nevystupují jak ony samy, tak se nepoužívají k získání členů, které jsou argumenty korelační funkce.

Další dva vektory (pole) představují vzájemnou polohu v laboratorní a těžišťové soustavě. To znamená vzájemnou polohu bodů produkce. To jsou pole  $poloha[3]$  a  $polohaT[3]$ .



Další data již nejsou představovaná poli. První je  $q$ , velikost vektoru vzájemné hybnosti. Počítá z pole  $hybnostT[3]$ . Následuje  $k$ , spočítané jako  $k = q/2$ . Potom je  $qR$ , které představuje velikost vektoru vzájemné rozmazané hybnosti. Potřebujeme také velikost vzájemné polohy bodů produkce v těžišťové soustavě, kterou nám poskytuje  $r$ . Dále následuje  $rNT$ . To znamená netransformované  $r$ , což je velikost vektoru vzájemné polohy v laboratorní soustavě.

Poslední mezi daty je  $skalarKR$ . Ten představuje skalární součin vektorů vzájemné polohy a hybnosti v těžišťové soustavě dělený dvěma.

Nabízí se otázka, proč jsme jako data třídy *Pripad* zvolili právě tyto. Volím vše, co budu dále potřebovat pro simulaci korelační funkce. Druhá motivace je jejich role při tvorbě histogramů, ať již zásadních pro spektrum  $q$ , nebo pomocných.

Všechna data této třídy jsou public.

Nepoužíváme ukazatele, takže destruktory třídy je primitivní. Konstruktor je velmi komplexní, proto mu věnujeme více prostoru.

### 5.6.4 Konstruktor třídy *Pripad*

V konstruktoru budeme provádět mimo jiné i Lorentzovské transformace a budeme generovat hybnost a polohu. Je logické, že v konstruktoru musíme přiřadit hodnoty všem parametrům částice.

První zavedeme hmotnost částic. Pion+ a pion- mají stejnou hmotnost. Hmotnost zadáváme v  $MeV$ . Její hodnota je  $m = 139.57018MeV$ . Potom budeme generovat hybnosti polohy částice.

V kapitole (5.5) jsme řekli, že máme více způsobů, jak generovat souřadnice. První způsob je generování složek vektorů v laboratorní soustavě a následně transformovat. Druhý způsob je generování složek vzájemné hybnosti a polohy v těžišťové soustavě.

#### Generování složek vektorů v laboratorní soustavě

Zde se budu zabývat generováním po složkách v laboratorní soustavě. Uvedu fragment kódu:

```
for (unsigned int i = 1; i <4; i++)
{
    prvni.hybnost[i] = rnd( 0.0, 200.0);
    druha.hybnost[i] = rnd( 0.0, 200.0);
    prvni.hybnostR[i] = rozmazat(prvni.hybnost[i] );
    druha.hybnostR[i] = rozmazat( druha.hybnost[i] );
    prvni.poloha[i]=gRandom->Gaus(0,2.15);
    druha.poloha[i]=gRandom->Gaus(0,2.15);
    T1=T1+prvni.hybnost[i]*prvni.hybnost[i];
    T2=T2+druha.hybnost[i]*druha.hybnost[i];
}
```

Ve *for* cyklu generujeme hybnost částice v jednotlivých osách. Pohybuje se v intervalu 0 - 200 MeV a její rozdělení je v jednotlivých osách rovnoměrné. Polohu generujeme také po jednotlivých složkách a to Gaussovým rozdělením. Jeho parametry jsou  $\mu$  a  $\sigma$ , se kterými budeme manipulovat pro potřeby programu. *For* cyklus také počítá kvadrát energie částice v laboratorní soustavě (značíme  $T$ ). Je to jednoduché sčítání kvadrátů hybnosti přes jednotlivé složky, ke kterým se přičte kvadrát klidové hmotnosti. Je to rovnice, která se používá v přítomnosti relativistických jevů. Jak jsem již poznamenal  $c = 1$ .

Po skončení *for* cyklu se spočte energie z kvadrátu. Tím dostaneme kompletní čtyřvektor hybnosti. Čtyřvektor polohy dostaneme generováním času. Ten můžeme generovat rozdělením, které uznáme za vhodné v podmínkách simulace. V tomto programu je brán čas produkce jako nulový, tedy částice jsou generovány současně. Používáme předpoklad současnosti uvedený v (3.4.1).

Dále v konstruktoru provedeme relativistické transformace. V transformaci vidíme, proč je gama zavedená jako vektor. Dále jsou v konstruktoru Lorentzovské transformace.

Po ukončení relativistických transformací následuje *for* cyklus, který spočítá data třídy *Pripad*, konkrétně *hybnostT*[3], *hybnostRT*[3], *poloha*[3] a *polohaT*[3]. V rychlosti bych rád upozornil, že složky polí na pravé straně rovnic jsou posunuté o jednu, např.  $hybnostT[i] = prvni.hybnostT[1 + i] - druha.hybnostT[1 + i]$ . Je to proto, že na pravé straně jsou čtyřvektory.

Následujících několik *for* cyklů spočítá další data naší třídy. Jelikož je jejich funkce zřejmá, nemá smysl ji rozebírat. Jen poznamenam, že vše by se dalo počítat v jednom cyklu. V oddělených to dělám proto, že nechci ztěžovat orientaci v již tak složitém konstruktoru.

## Generování souřadnic v těžišťové soustavě

Druhý způsob je generování velikosti vektoru hybnosti v těžišťové soustavě. Následuje získání jednotlivých složek onoho vektoru. Ten vidíme na následujících řádkách. Připomínám, že velikost vektoru vzájemné hybnosti v těžišťové soustavě se značí jako  $q$ .

```
q=rnd( 0.0, 50.0);
double qkvadrat=q*q;
double tmp11=rnd( 0.0, 1.0);
double tmp22=rnd( 0.0, 1.0);
if(tmp22<tmp11)
{
    double pom=tmp11;
    tmp11=tmp22;
    tmp22=pom;
}
hybnostT[1]=sqrt(tmp11*qkvadrat);
```

```
hybnostT[2]=sqrt((tmp22-tmp11)*qkvadrat);
hybnostT[3]=sqrt((1.0-tmp22)*qkvadrat);
```

Zde používám rovnoměrné rozdělení  $q$ . Dále uvedu generování složek vektoru vzájemné polohy, ze kterých si spočítám velikost onoho vektoru, který značím proměnou  $r$ .

```
polohaT[2]=gRandom->Gaus(0,sqrt(2*1.5));
polohaT[0]=gRandom->Gaus(0,sqrt(2*1.5));
polohaT[1]=gRandom->Gaus(0,sqrt(2*1.5));
r=sqrt(polohaT[1]*polohaT[1]+polohaT[2]*polohaT[2]+polohaT[3]*polohaT[3]);
```

Všimněme si, že *skalarKR* je počítán až po uvedení těchto generátorů. Bude se tedy počítat nezávisle na generátorech. V něm se vyskytuje dříve zmiňované dělení číslem 197.3.

### 5.6.5 Třída *Generator*

Poslední před funkcí *main* je třída *Generator*. Její význam je ten, že nám umožňuje jednoduše generovat náhodná čísla z Gaussova rozdělení. Hodí se to například tehdy, když v *mainu* potřebuji něco "gaussovsky" rozmýtl.

```
1 class Generator
2 {
3     public:
4     Generator();
5     ~Generator();
6     double r;
7 };
```

### 5.6.6 Pomocné struktury

V průběhu programu bude třeba spočítat vektory, které se nevztahují k jedné částici, ale k oběma. Je to například vektor vzájemné hybnosti. Je evidentní, že jsme nuceni pracovat se vzájemnou polohou (vzájemnou polohou bodů vzniku), vzájemnou hybností, nebo transformovanou hybností. To vše se dá popsat vektorem (polem) o třech složkách.

Zavedeme si strukturu "vzajemna".

```
struct vzajemna
{
    double x;
    double y;
    double z;
};
```

Vidíme, že v této struktuře jsou tři složky. Do těchto složek budeme ukládat hodnoty, jaké budou vektory nabývat pro jednotlivé osy. Podívejme se proto do *while* cyklu v *main*. Pro každý krok *while* cyklu se vytvoří "Piony", což je jeden objekt třídy *Pripad*. Zavedli jsme si "vzájemné" struktury *polohaT*, *poloha*, *hybnostT* a *hybnostRT*. Do těchto struktur uložíme vzájemnou polohu, hybnost a rozmazanou hybnost v těžišťové soustavě. To nám právě udělá funkce *priradVz*. Jak vidíme, tato funkce je přetížená. Jednou má jako argumenty *Pripad* a tři struktury, jednou jen jednu strukturu. Je to proto, že zřídka kdy pracujeme se vzájemnou polohou v laboratorní soustavě. Jestliže jí nepotřebujeme, nemá smysl provádět výpočty.

K práci se strukturami si zavedeme několik pomocných funkcí. Ty nemá smysl zde rozebírat, jejich smysl je zřejmý z programu.

Jak uvidíme později v programu, s těmito strukturami se nepracuje. Tedy vyvstává otázka, proč pracně zavádíme něco, co dále nepoužíváme. Ty struktury nejsou to jediné, co jsem musel psát, neboť bylo nutné napsat i funkce sloužící k práci s nimi. Důvodem je složitost programu. Jak již jsme viděli, můžeme používat různé způsoby na tvorbu vzájemné polohy a hybnosti. Tvorba tohoto programu a práce s ním zahrnuje i neustálé zkoušení a ověřování jednotlivých částí. Jak později uvidíme, korelační funkce se skládá z několika dalších funkcí. Struktury jsem používal například pro ukládání vektorů před pokusem nového generátoru. Proto, jakmile struktury byly naplněné, mohl jsem skalární součiny počítat pomocí třídy *Pripad*, nebo k tomu využít ony nezávislé struktury. Skalární součin není přirozeně to jediné, co se jejich pomocí dalo vyjádřit.

Dalo by se to samozřejmě vyjádřit pomocí polí, s kterými by se pracovalo lépe pomocí *for* cyklu. Důvod je ten, že jsem pro to, co má sloužit jako neměnné, chtěl zvolit odlišnou proměnnou.

Ještě jednou zopakuji, že tyto struktury standartně nepoužívám. Svoji roli hrají při kontrole výsledků.

Následující tři funkce se používají při výpočtu konfluentní hypergeometrické funkce.

## 5.6.7 Funkce *Main*

### Úvod

Funkce *main* je velmi dlouhá a obsahově plná. Proto při jejím průchodu budu dělit text na velké množství podkapitolek, které se budou zabývat jednotlivými částmi programu.

### Použití histogramy

Příkaz `TH1::SetDefaultSumw2()` nastavuje vykreslení histogramů s chybami.

Potom následuje dlouhý sloupec histogramu. V histogramech h1-h5 se zapisují složky vektoru vzájemné polohy, velikost vektoru vzájemné polohy v těžišťové soustavě a

laboratorní soustavě. Histogramy h5 a h6 budou použité pro spektrum  $q$ . Jeden histogram má rozsah 0-50 a druhý 0-200. Druhý bude sloužit pro tvorbu korelační funkce pro případ  $\pi^+ \pi^-$ . Histogram h7 slouží pro rozmazanou hybnost.

Do histogramů h10 až h13 zapisujeme spektrum  $|\eta|^{-1}$ . Proměnou  $\eta$  jsme viděli v podkapitole (3.4.2). V prvním histogramu je nevážené spektrum a dále ho vážím Coulombickým penetračním faktorem a reálnou a imaginární složkou funkce  $\chi$ , kterou jsme viděli v rovnici (3.37).

Histogram  $h20$  slouží pro vykreslení spektra  $q$  za vlivu kvantové statistiky. Histogramy  $h21$ - $h22$  slouží pro vykreslení silné interakce. První případ je čistě silná interakce, druhý je silná interakce v případě kompletní FSI.

Histogramy  $h30$  a  $h31$  slouží pro vykreslení regulární a singulární Coulombické vlnové funkce.

Histogramy  $h40$  slouží pro vykreslení konfluentní hypergeometrické funkce.

Histogramy  $h50$  a  $h51$  uchovávají spektrum  $q$  vážené vahou pro získání korelační funkce. První případ je pro  $\pi^+ \pi^-$ , který neobsahuje kvantovou statistiku. Druhý je pro symetrizovaný případ. Histogram  $h52$  slouží k uchování spektra ovlivněného jen Coulombickou interakcí.

## *While* cyklus

První si zdůvodníme funkci cyklu. Jak jsme si již řekli, korelační funkci simulujeme dělením dvou histogramů. V nich jsou spektra  $q$ . Proto v každém kroku cyklu vytvořím pro každý z histogramů (teď myslím histogramy používané pro tvorbu námi požadovaných funkcí, ne pomocné histogramy) jednu hodnotu  $q$  a spočítám pro ni váhu podle požadovaných funkcí,  $q$  zapíšu do příslušného histogramu.

V každém kroku cyklu si utvořím objekt dříve vysvětlené třídy *Pripad* a nazvu ho podle simulovaných částic "Piony". Jak jsme dříve viděli, tato třída simuluje situaci dvou částic. Takže první řádek cyklu náleží tvorbě objektu dané třídy.

Jak víme, tato třída má data, která slouží k výpočtu funkcí. Již dříve jsme řekli, že její data volíme účelově, aby se nám potom s třídou lépe pracovalo. Proto si v prvních řádcích cyklu vypíšeme ona data.

Nabízí se otázka, proč zavádím nové proměnné typu *double*, proč rovnou nepoužívám například místo  $r$  *piony.r*. Důvod je ten, že je možné i během chodu cyklu  $r$  upravovat a potom není vhodné přepisovat data objektu. Další důvod spočívá v praktičnosti, neboť je nepochybně pohodlnější do rovnic psát  $r$  místo *piony.r* a hlavně se potom rovnice mnohem lépe čtou. Jelikož pohodlnost programátora nemusí být braná jako podstatný důvod, je třeba říci, že tímto značením jsem zachovával jednotu s literaturou, z níž jsem vycházel, což je v tomto případě [5].

Zdůvodníme si, proč používáme právě *while* cyklus. Jde nám o to, že histogramy chceme naplnit přesným množstvím hodnot  $q$ . Proč tedy nepoužíváme *for* cyklus? Je to z toho důvodu, že hodnoty některých případů třídy *Piony* nemůžeme použít. V korelační funkci na místě vystupuje ve jmenovateli  $r$ . Kdyby  $r$  pokleslo

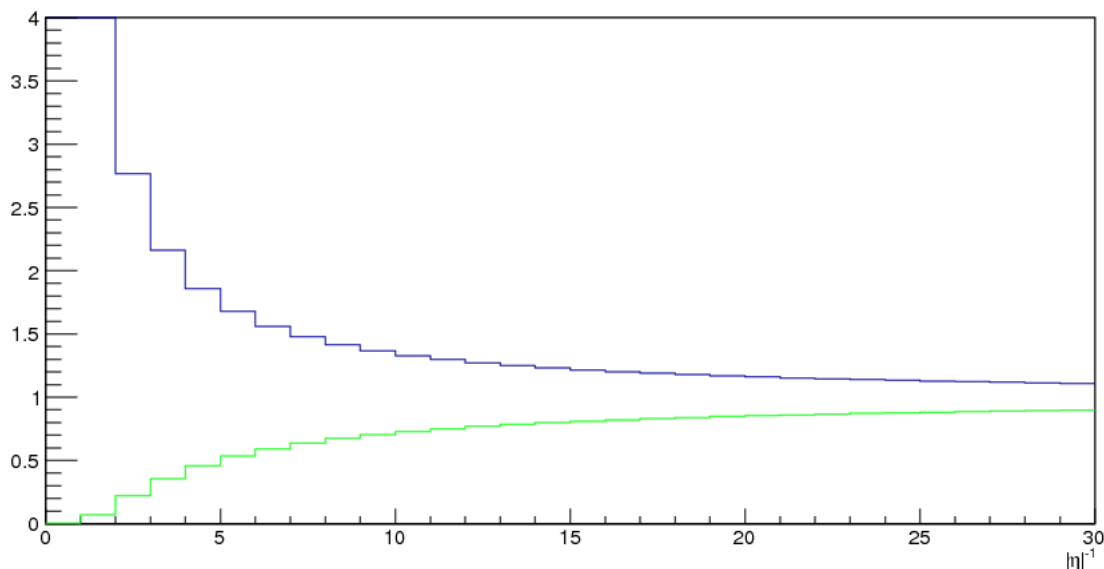
pod označenou hodnotu, váha vystoupá prudce nahoru, limitně do nekonečna. Proto hlavně v případě malých produkčních oblastí by korelační funkce byla při zanedbání této podmínky nepoužitelná. Jelikož trvám na přesném množství zápisů do histogramů, používám *while* cyklus.

Když jsme si zdůvodnili základy cyklu, přistoupíme k výpočtu jednotlivých funkcí.

### Proměnná $\eta$ a Coulombický penetrační faktor

Dále v programu vyjádříme proměnou  $\eta$ , Coulombický penetrační faktor, a proměnnou, kterou nazývám "*etaReverse*". Ta nám slouží k tomu, abychom v závislosti na ní vykreslili Coulombický vlnový faktor. Vyjadřuji ho tak proto, že v literatuře, ze které jsem vycházel byl takto vykreslen a to pro mě bylo důležitým ověřením správnosti mé práce.

Na následujícím grafu je vykreslen Coulombický penetrační faktor v závislosti na  $|\eta|^{-1}$ . Byl vykreslen pomocí Metody Monte Carlo. Byl vykreslen pro případ atrakce i repulse.



Obrázek 5.1: Coulombický penetrační faktor. Zelená: repulse, Modrá: atrakce.

Dejme tomu, že se nám povedlo vykreslit funkci. Jsme si jistí, že ji máme správně? Můžeme si být bezpečně jisti, že ji používáme správně i v programu, kde pracujeme pomocí metody Monte Carlo? Když vykreslujeme funkci například pomocí histogramů, máme osu rozdělenou na miniaturní biny, z nichž každý použijeme pouze jednu. V Monte Carlu tomu tak není, neboť hodnoty osy  $x$  budou sledovat jisté rozdělení, takže počet zásahů jednotlivých binů se bude lišit podle rozdělení a ani pro jedno rozdělení a dva případy nebude stejný. Musíme se přesvědčit, že vykreslovaná funkce není ovlivněná zvoleným rozdělením. Proto každá funkce musí mít ověřené vykreslení i pomocí metody Monte Carlo.

Abych předchozí vyjádřil stručněji: vykreslení funkcí pomocí Monte Carla je velice neefektivní a vnáší chybu. Přesto je nutné ho provádět pro ověření správnosti použitých výrazů.

## Kvantová statistika

Simulace korelační funkce jen za vlivu kvantové statistiky je nejjednodušší případ. Tím vysvětluje smysl histogramu  $h_{20}$ , kam zapisujeme  $q$  vážené vahou  $w$ .

## Silná interakce

Při výpočtu váhy pro silnou interakci vycházíme z rovnice (3.30), kterou dále rozvíjíme (3.31) a (3.32).

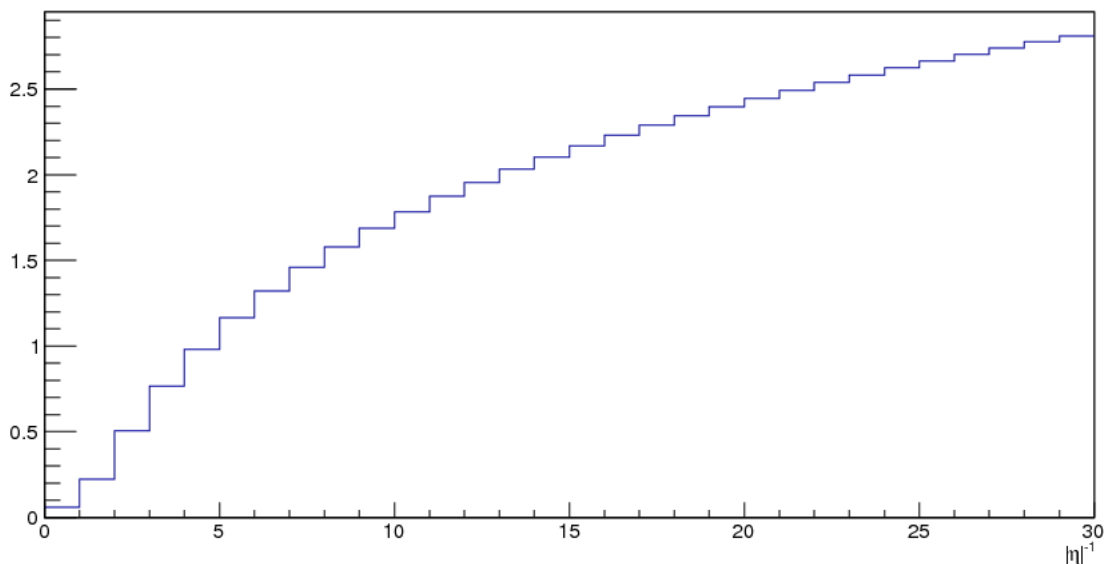
Povšimněme si, že v programu jsou uvedena dvě pole, značená  $A1[3]$  a  $A2[3]$ . Tato pole v sobě ukrývají hodnoty koeficientů  $A_i$ , které se nacházejí v rovnici (3.32). Stejně tak trochu pod nimi lze nalézt proměnné  $s_{01}$  a  $s_{02}$ , které skýtají hodnoty pro koeficienty  $s_0$  téže rovnice. Zatím nás zajímá pouze pole  $A2[3]$  a koeficient  $s_{02}$ , který použijeme pro vyjádření váhy silné interakce. Důvod těchto parametrů byl zmíněn v kapitole (5.3).

Následovně přistoupíme ke korelační funkci za vlivu kompletního FSI.

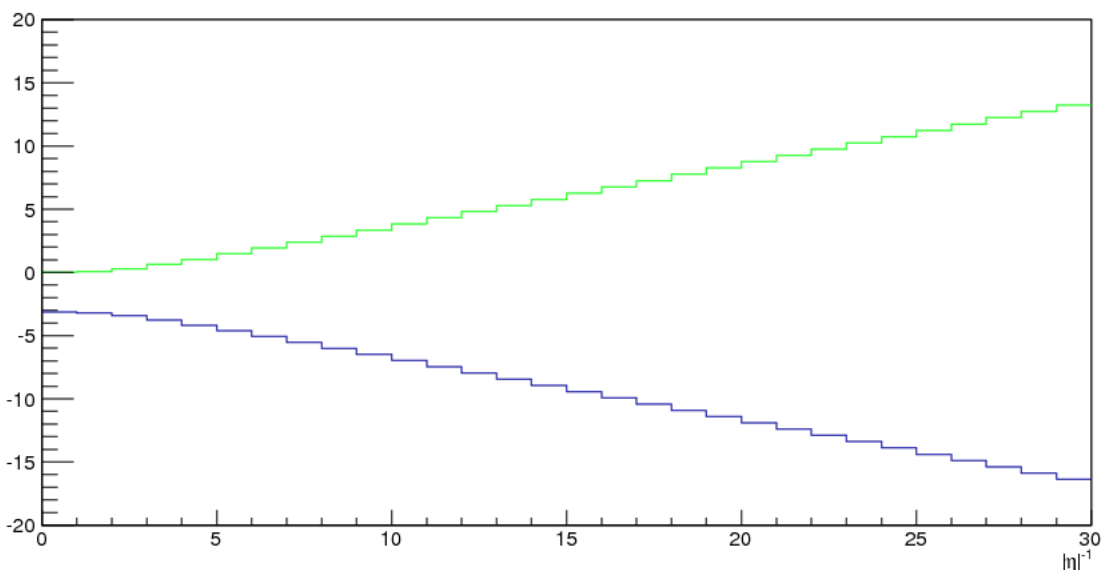
## Silná interakce v korelační funkci pro kompletní FSI

První úskalí, kterému čelíme při popisu kompletní FSI je, že funkci pro silnou interakci, kterou jsme použili v předchozí části, nemůžeme použít v této. V rovnici (3.32) počítáme proměnnou  $K$ , kterou jsme si v programu vyjádřili pomocí pole  $A2[3]$ . Vycházíme zde z podkapitolky (5.2).

Funkce silné interakce vystupující v (3.35) je dále rozvinutá v (3.36). V ní vystupuje funkce  $\chi(\eta)$ , která je vystižena rovnicí (3.37). Na následujících grafech jsou vykresleny její reálná a imaginární část. Reálná část nezávisí na tom, jestli máme případ atrakce, nebo repulse.



Obrázek 5.2: Reálná část funkce  $\chi(\eta)$ .



Obrázek 5.3: Imaginární část funkce  $\chi(\eta)$ . Zelená: repulse. Modrá: atrakce.

Jedny z funkcí, které tvoří onu korelační funkci, jsou Coulombické vlnové funkce. Před tím, než budeme kompletní FSI rozebírat, zaměříme se na strukturu *gsl\_sf\_result* z GSL.

### Struktura na ukládání výsledků

Struktura `gsl_sf_result` je částí GSL a pro její používání je nutné do programu vložit header file `gsl_sf_result.h`. Struktura se používá na ukládání výsledků. Uvedu jí a okomentuji.



```

8     typedef struct
9     {
10        double val;
11        double err;
12    } gsl_sf_result;
13

```

14 Proměnná *val* ukládá výsledek funkce a proměnná *err* ukládá chybu výsledku. Může  
15 se stát, že výsledek, který se ukládá do typu *double* je příliš velký a *double* "přeteče".  
16 Proto existuje strukturagsl\_sf\_result.h , která tuto nevýhodu řeší.

```

17     typedef struct
18     {
19        double val;
20        double err;
21        int     e10;
22    } gsl_sf_result_e10;
23

```

V tomto případě vracíme výsledek pomocí škálovacího exponentu, který je uložen v *e10*. Výsledek je potom vrácen ve tvaru, který pro ilustraci uvedeme v následující rovnici:

$$vysledek = val * 10^{e10}. \quad (5.2)$$

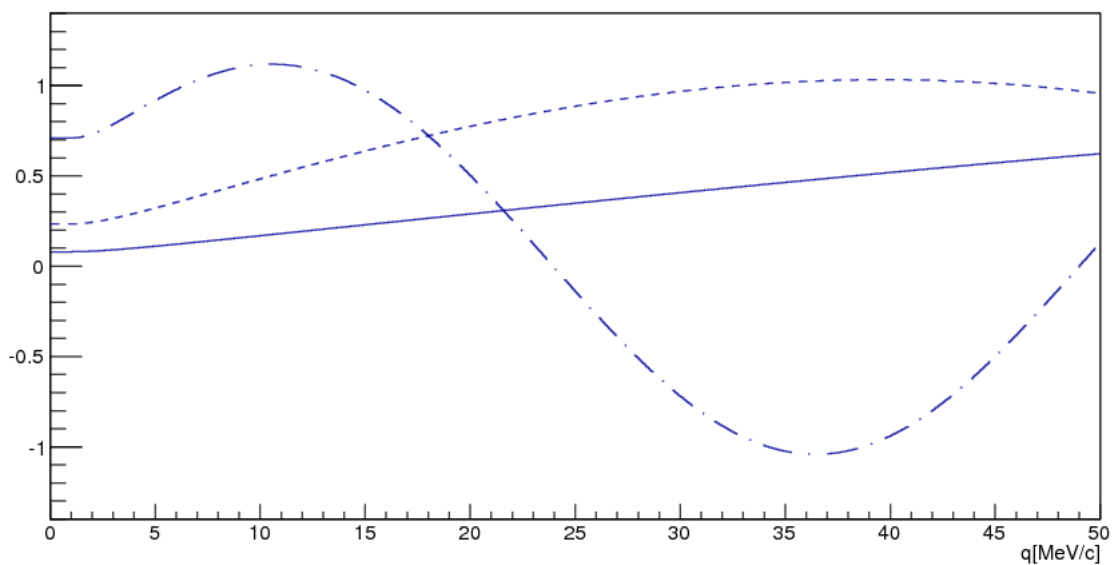
Jaká je motivace k používání těchto struktur? Jak víme, v C++ je možné jako výsledek funkce vrátit jenom jednu hodnotu. To nepředstavuje problém, neboť výsledky můžeme vrátit pomocí ukazatelů. Co by se stalo, kdybychom nechtěli vracet jenom výsledek, ale kdybychom chtěli vracet i jeho chybu? S tímto případem se v příští podkapitolce setkáme při výpočtu Coulombických vlnových funkcí. Potom oceníme výše uvedené struktury, neboť pomocí nich můžeme jednoduše získat výsledky.

## Coulombické vlnové funkce

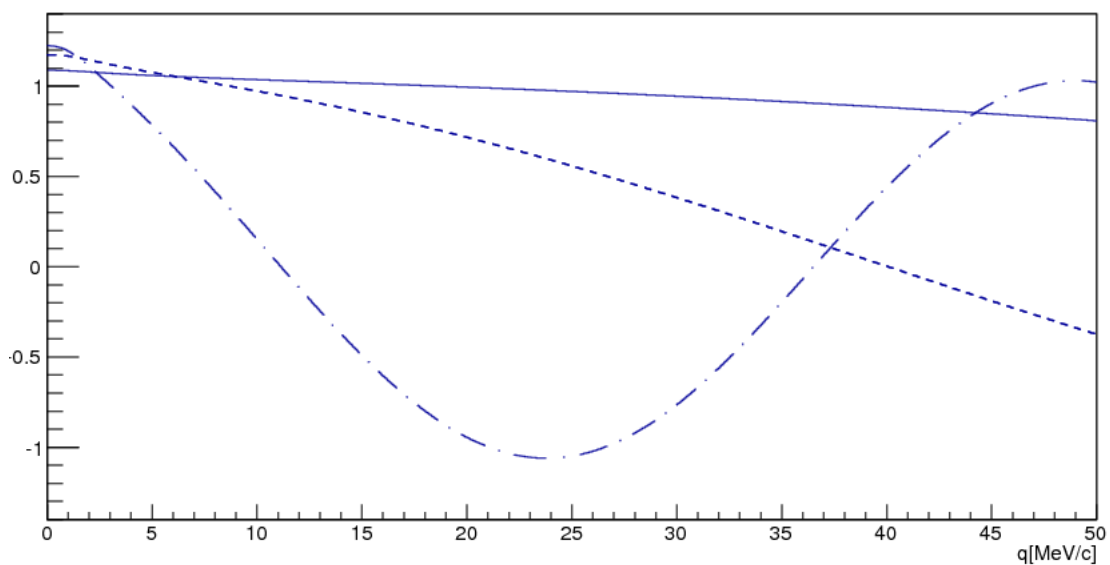
Předchozí rovnice byla vysvětlena v kapitolce (5.3). Následovně se budu věnovat Coulombickým vlnovým funkcím. V rovnici (3.35) vystupuje funkce  $\tilde{G} = \sqrt{A_c}(G_0 + iF_0)$ . To je kombinace regulární  $F_0$  a singulární  $G_0$  Coulombické funkce.

Na dalších řádcích jsou počítány Coulombické vlnové funkce. První jsou definovány čtyři struktury typu `gsl_sf_result` a ukazatele na ně. Funkce

`gsl_sf_coulomb_wave_FG_e` z GSL vrací proměnnou typu integer, která pro nás není důležitá. Podstatné je, že tato funkce pomocí výše uvedených struktur vrací výsledky Coulombických funkcí, ze kterých můžu spočítat  $\tilde{G}$ . Dodatečné informace o těchto funkcích lze nalézt na [3]. Singulární a regulární Coulombické vlnové funkce jsou uvedeny na následujících grafech. Simuloval jsem ji pro tři zdroje, aby byl vidět vliv velikosti produkční oblasti.



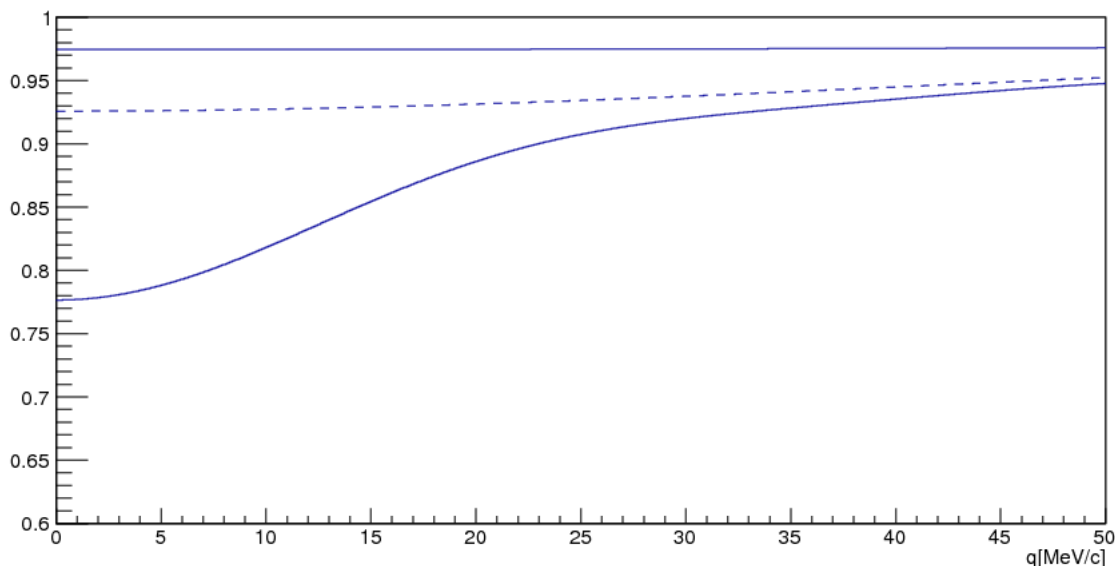
Obrázek 5.4: Regulární Coulombická vlnová funkce. Spojitá  $r^* = 5\text{fm}$ , čárkovaná  $r^* = 15\text{fm}$ , čerchovaná  $r^* = 50\text{fm}$ .



Obrázek 5.5: Singulární Coulombická vlnová funkce. Spojitá  $r^* = 5\text{fm}$ , čárkovaná  $r^* = 15\text{fm}$ , čerchovaná  $r^* = 50\text{fm}$ .

## Konfluentní hypergeometrická funkce

Konfluentní hypergeometrickou funkci jsme již zmiňovali před funkcí *main*. Musel jsem jí napsat, protože neexistuje funkce z *GSL*, nebo z *Rootu* jako byla například funkce na výpočet Coulombických vlnových funkcí. Algoritmus jsem vzal z programu *fsi4pi3n2c\_histo.f*, který jsem získal po osobní komunikaci s Richardem Lednickým. Opět jsem ji vyjádřil pro tři produkční oblasti různých velikostí.



Obrázek 5.6: Konfluentní hypergeometrická funkce. vlnová funkce. Spojitá  $r^* = 5 fm$ , čárkovaná  $r^* = 15 fm$ , čerchovaná  $r^* = 50 fm$ .

## Kompletní FSI

Na následujících řádcích se počítá váha, která slouží k výpočtu korelační funkce za vlivu kompletního FSI, při čemž vycházím ze vzorce (3.35). Ještě bych upozornil na proměnnou  $fCoulombP$ , jejíž účel je kontrola správnosti ostatních funkcí. Zapojím ji do korelační funkce místo proměnné  $fCoulomb2$  a pakliže korelační funkce vychází chybně, vím, že na vině jsou ostatní funkce.

Při simulaci této funkce se opět braly zdroje o třech velikostech.

## Kompletní FSI a kvantová statistika

Poslední, co potřebujeme udělat, je zahrnutí kvantové statistiky. To uděláme jednoduše symetrizací předchozího případu. Proměnná typu *double* je váha pro korelační funkci za vlivu FSI i kvantové statistiky.

## Coulombická interakce

Coulombickou interakci vyjádříme jednoduše. V kapitole (3.4.2) jsme zmínili, že v tom případě v rovnici (3.35) bereme  $f_c(k^*) = 0$ .

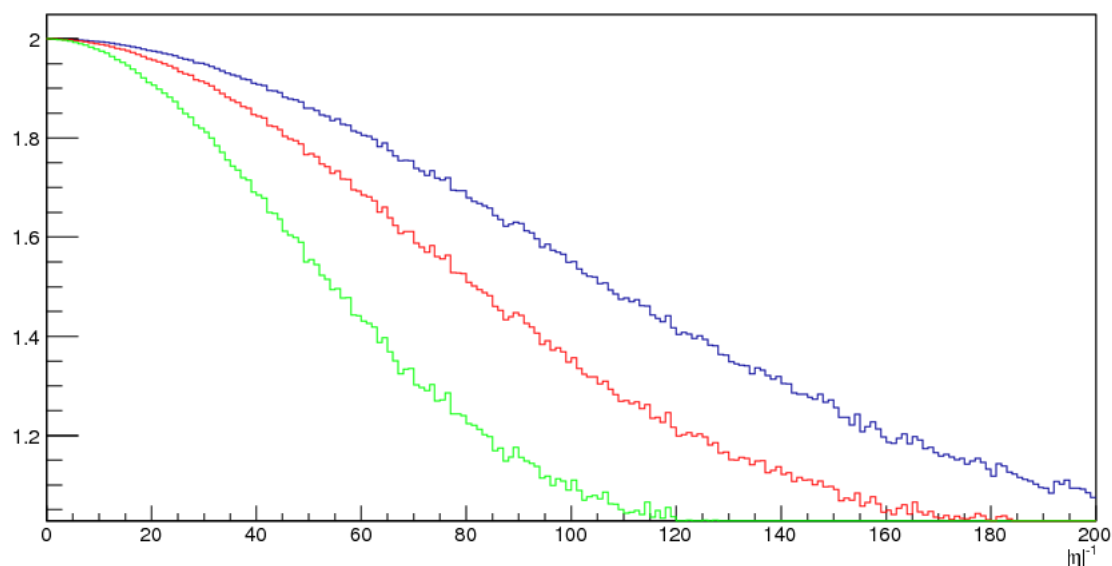
# Kapitola 6

## Výsledné funkce

V této sekci si uvedeme výsledné simulované korelační funkce.

### 6.1 Korelační funkce jen za vlivu kvantové statistiky

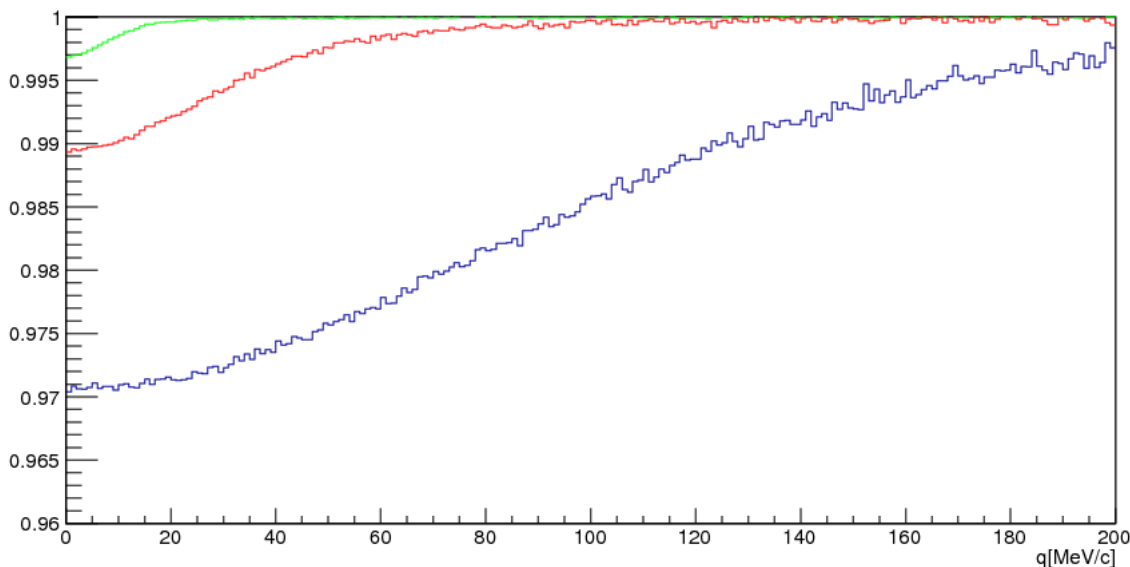
Na obrázku (6.1) vidíme, že vliv kvantové statistiky je největší při malém  $q$ . Se zvětšujícím se  $q$  její vliv klesá. Kdybychom pozorovali vliv velikosti zdroje, viděli bychom, že průběh je obdobný. Funkce nám začíná stejně vysoko a zmenšuje se s rostoucím  $q$ . Průběh je rychlejší se zvětšující se velikostí zdroje.



Obrázek 6.1: Korelační funkce za vlivu kvantové statistiky. Modrá:  $r^* = 1,5 fm$ , červená  $r^* = 2 fm$ , zelená  $r^* = 3 fm$ .

## 6.2 Korelační funkce jen za vlivu silné interakce.

Vliv silné interakce opět klesá s rostoucím  $q$ . Připomínám, že jistý vliv na korelační funkci se projeví tím, jak moc je korelační funkce vzdálená od jedné. Vidíme, že silná interakce závisí na velikosti zdroje. Čím větší je zdroj, tím menší je vliv za malých  $q$  a tím rychleji se funkce dostane k jedné.



Obrázek 6.2: Korelační funkce za vlivu silné interakce. Modrá:  $r^* = 5\text{fm}$ , červená  $r^* = 15\text{fm}$ , zelená  $r^* = 50\text{fm}$ .

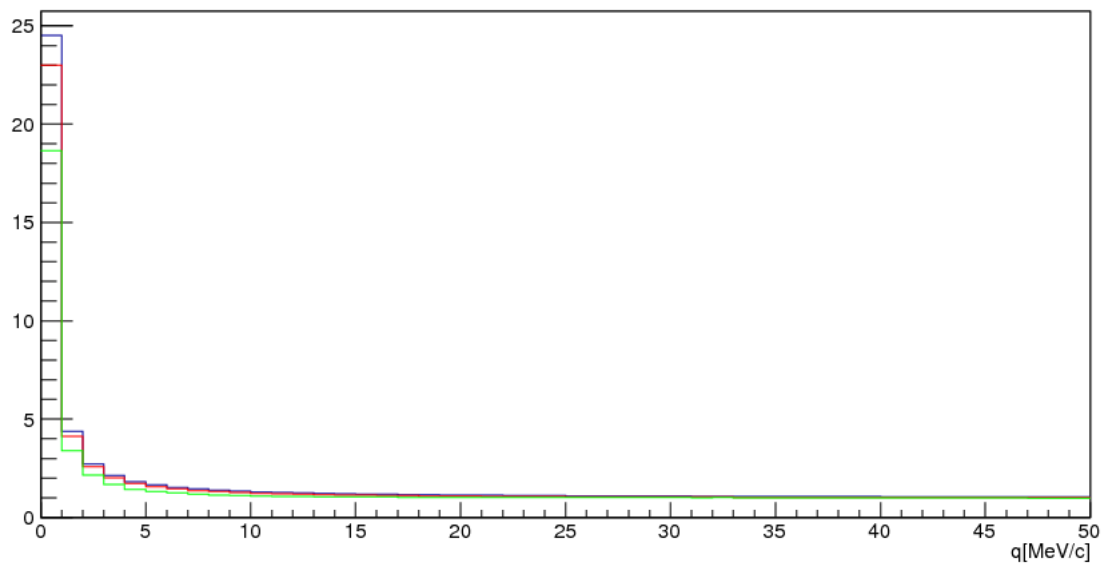
## 6.3 Korelační funkce jen za vlivu Coulombické interakce.

Pro Coulombickou interakci máme dvě možnosti: repulze a atrakce. První uvedeme případ atrakce.

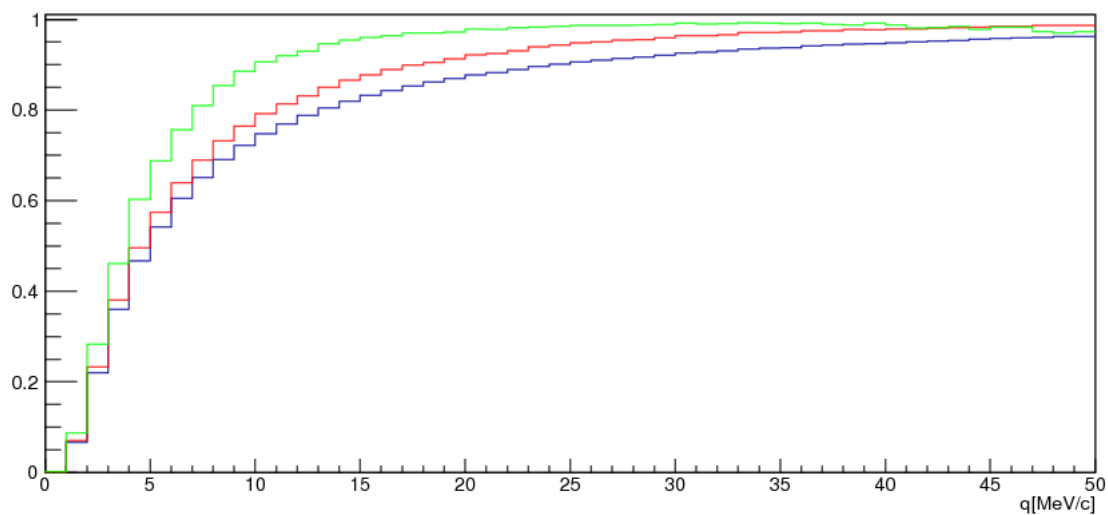
Na obrázku (6.3) vidíme, že čím je menší zdroj, tím výše funkce začíná a tím pomaleji jde k jedné. Průběh je jinak obdobný. Pro případ atrakce bude větší pravděpodobnost, že částice poletí blízko sebe, přitáhnou se. Proto bude větší pravděpodobnost nalezení dvou částic s malým vzájemným  $q$  a korelační funkce je tedy vysoko nad jednou.

Vliv zdroje je následující: čím je zdroj menší, tím více se interakce uplatní (Coulombická interakce klesá se čtvercem). Proto je u menších zdrojů korelační funkce výše a pomaleji klesá.

Dále uvedu korelační funkce jen za vlivu Coulombické interakce pro případ repulze. Na obrázku (6.4) vidíme, že korelační funkce je skoro nulová v oblasti malých  $q$ . Důvod je ten, že Coulombická interakce nedovolí, aby částice letěli blízko sebe, tj.



Obrázek 6.3: Korelační funkce za vlivu Coulombické interakce  $\pi^+ \pi^-$ . Modrá:  $r^* = 5 fm$ , červená  $r^* = 15 fm$ , zelená  $r^* = 50 fm$ .



Obrázek 6.4: Korelační funkce za vlivu Coulombické interakce  $\pi^+ \pi^+$ . Modrá:  $r^* = 5 fm$ , červená  $r^* = 15 fm$ , zelená  $r^* = 50 fm$ .

s malým vzájemným  $q$ . Vliv velikosti zdroje se projevuje stejně jako v minulém případě: čím je zdroj menší, tím větší je vliv interakce.

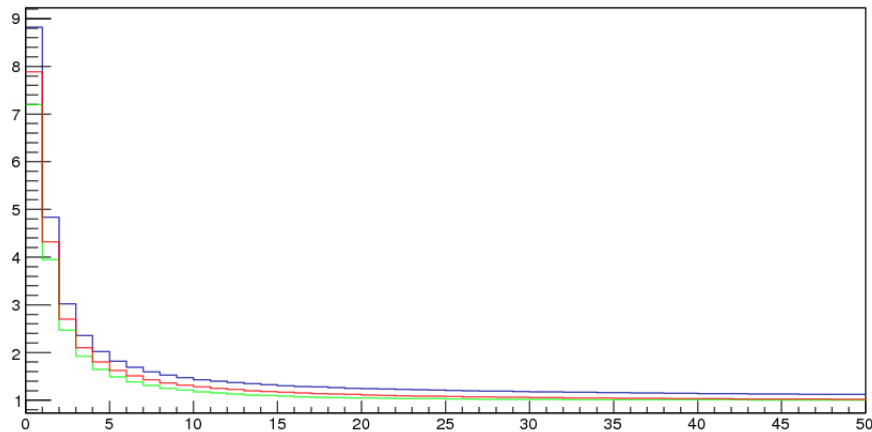
Dále bych rád poznamenal, že Coulombická interakce klesá mnohem rychleji s rostoucím  $q$  než kvantová statistika. Můžeme srovnat obrázky (6.3) a (6.4) s (6.1).

## 6.4 Korelační funkce za vlivu kompletního FSI.

Korelační funkce za vlivu kompletního FSI v sobě obsahuje vliv Coulombické a silné interakce. Coulombická interakce je přitažlivá pro případ interakce mezi piony  $\pi^+$  a  $\pi^-$ . Proto její vliv na korelační funkci se projeví tak, že korelační funkce bude velká za malých  $q$  a bude klesat s rostoucím  $q$ . Vysvětlení je jednoduché. Korelační funkce má vztah k pravděpodobnosti. Při kladné interakci je větší pravděpodobnost, že se částice budou pohybovat s malým vzájemným  $q$ . Jinak řečeno, přitáhnou se.

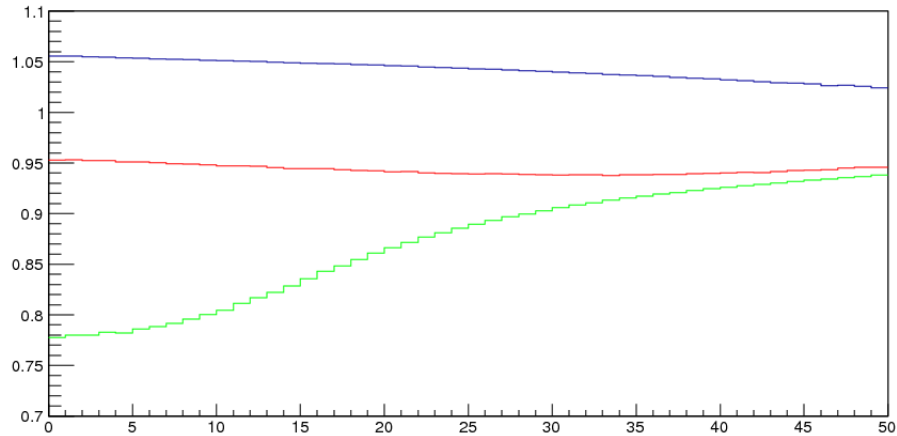
Můžeme studovat vliv velikosti zdroje. Vidíme, že průběh je obdobný. Korelační funkce začíná nad jednou. Čím větší je zdroj, tím níže začíná. U malého zdroje bude logicky Coulombická interakce větší, protože Coulombická interakce klesá se čtvercem vzdálenosti. Je třeba ještě poznamenat, že silná interakce se příliš neprosadí vedle Coulombické. Křivku ovlivňuje spíše tím, že ovlivňuje její sklon, ale velmi jemně.

Jak uvidíme na obrázku (6.5), korelační funkce jsou podobné. Dominantní vliv na ně má Coulombický penetrační faktor. Proto uvedu i korelační funkce dělené Coulombickým penetračním faktorem.



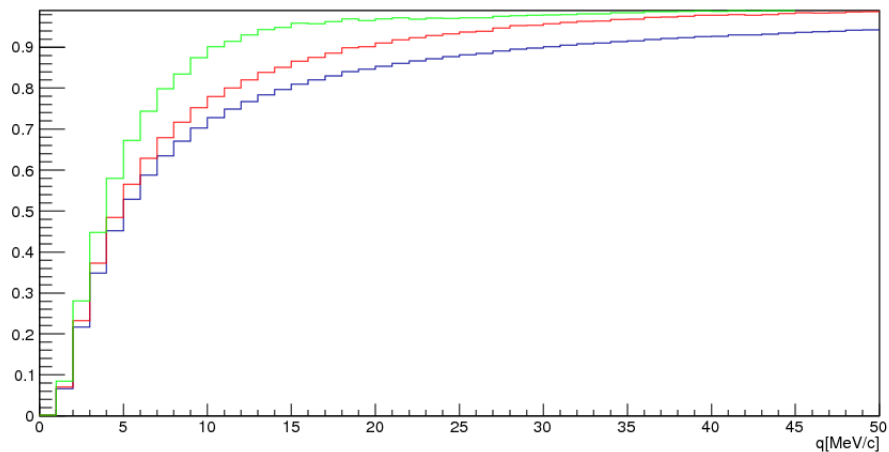
Obrázek 6.5: Korelační funkce za vlivu kompletní FSI. Modrá:  $r^* = 5 fm$ , červená  $r^* = 15 fm$ , zelená  $r^* = 50 fm$ .





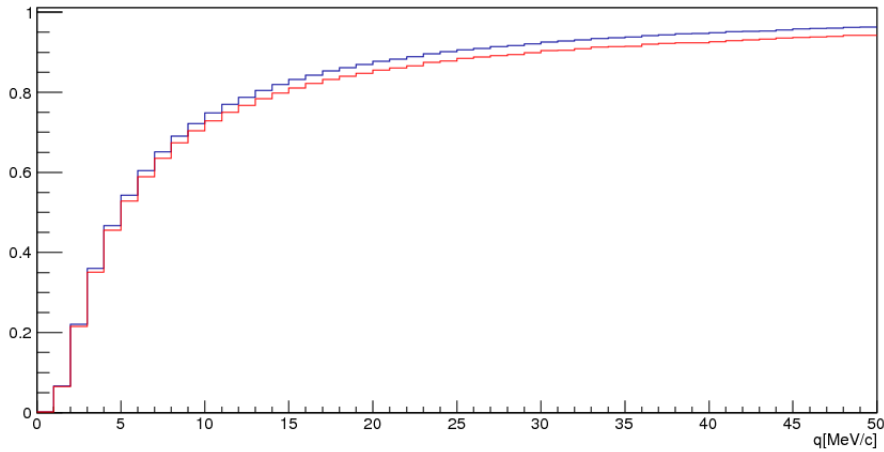
Obrázek 6.6: Korelační funkce za vlivu kompletní FSI dělená  $Ac(\eta)$ . Modrá:  $r^* = 5 fm$ , červená  $r^* = 15 fm$ , zelená  $r^* = 50 fm$ .

Pro případ repulse dostaneme graf (6.7). Opět vidíme, že vliv Coulombické interakce klesá s rostoucím rozměrem zdroje. Korelační funkce začíná u nuly kvůli Coulombické interakci a s rostoucím  $q$  jde k jedné. Čím větší je zdroj, tím rychlejší je průběh.



Obrázek 6.7: Korelační funkce za vlivu kompletní FSI. Modrá:  $r^* = 5 fm$ , červená  $r^* = 15 fm$ , zelená  $r^* = 50 fm$ .

Zajímavé je srovnání korelačních funkcí kompletního FSI a korelačních funkcí jen za působení Coulombické interakce. Uvedu srovnání pro případ repulse pro nejmenší zdroj, tedy  $r^* = 5 fm$ . Na obrázku (6.8) vidíme, že vliv silné interakce je velmi malý ve srovnání s Coulombickou interakcí.



Obrázek 6.8: Korelační funkce. Modrá: za vlivu kompletní FSI, červená: Coulombická interakce.

## 6.5 Korelační funkce za vlivu kompletního FSI a kvantové statistiky.

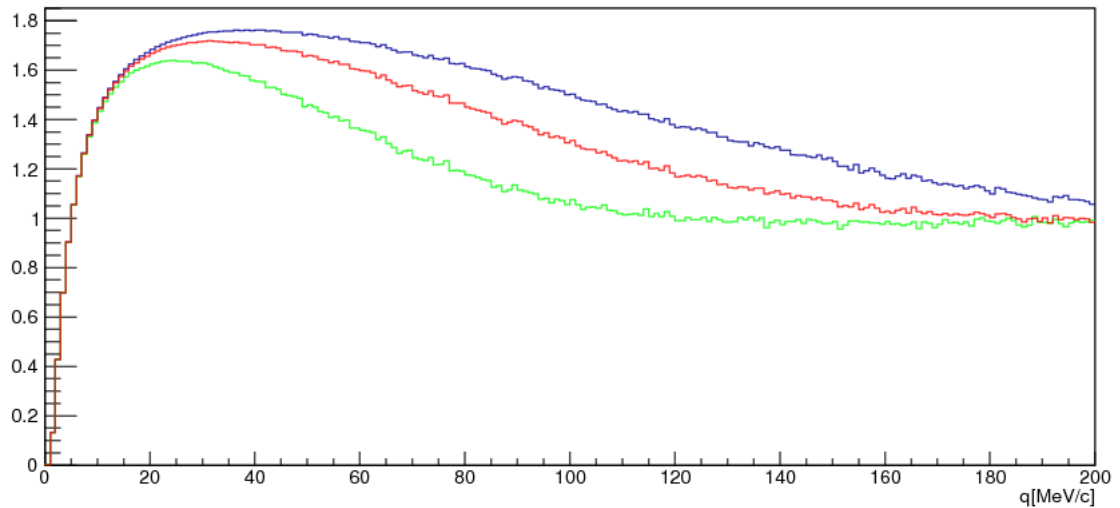
Pro případ interakce mezi identickými interagujícími částicemi musíme uvažovat silnou interakci, Coulombickou interakci jako v předchozím případě. Musíme ale uvážit i kvantovou statistiku.

Jak jsme zmínili v předchozím případě, silná interakce má minoritní vliv. Hlavní vliv bude mít kvantová statistika a Coulombická interakce. Na případě kvantové statistiky jsme viděli, že korelační funkce začíná nad jednou a klesá. U Coulombické interakce v minulém případě jsme viděli něco obdobného, ale to se nyní nebude opakovat. Pro případ shodného náboje je Coulombická interakce odpudivá.

Na grafu (6.9) vidíme případ tří korelačních funkcí pro různé zdroje.

Coulombická interakce snižuje pravděpodobnost, že by se částice pohybovaly s malým  $q$ . Vliv Coulombické interakce je silnější, než vliv kvantové statistiky. Klesá však rychleji a proto korelační funkce stoupne nad 1. Potom však klesá i vliv kvantové statistiky a korelační funkce se blíží k jedné.

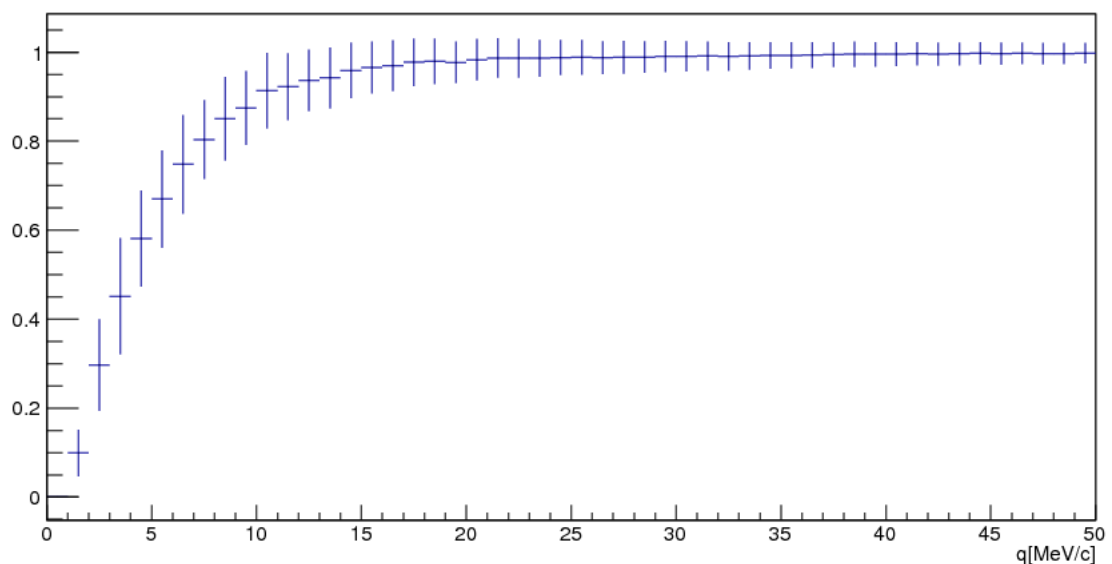
Kdybychom pozorovali vliv velikosti zdroje, vidíme opět, že nemění průběh, spíše ho urychluje. Klesá i velikost píku a poloha jeho maxima.



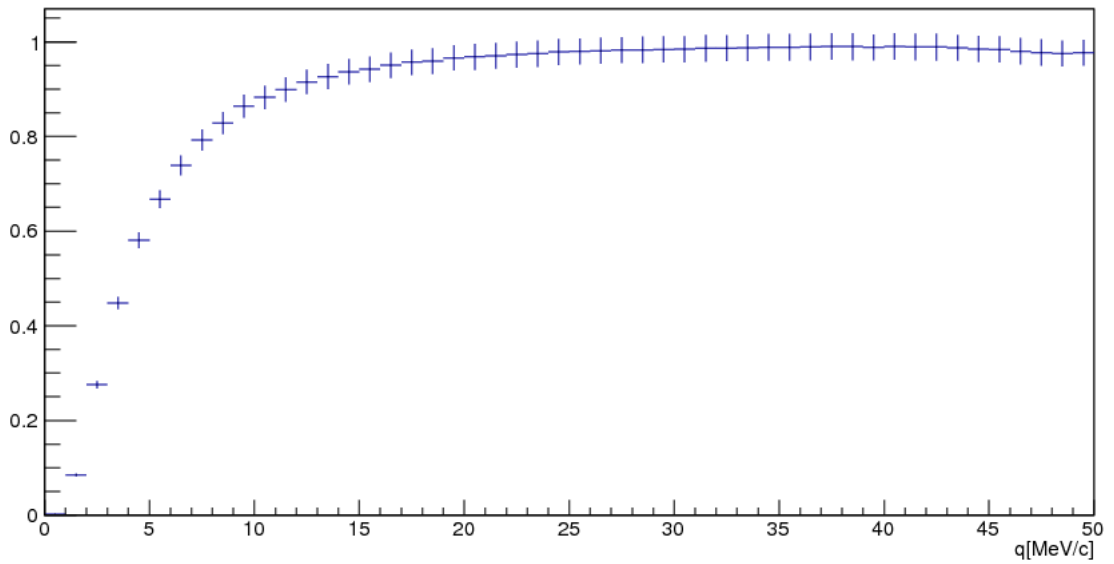
Obrázek 6.9: Korelační funkce za vlivu kompletní FSI s kvantové statistiky. Modrá:  $r^* = 1,5 fm$ , červená  $r^* = 2 fm$ , zelená  $r^* = 3 fm$ .

## 6.6 Vliv generátoru rozdělení

Už jsme mluvili o různých způsobech generování náhodných čísel. Zmínili jsme, že generování velikosti vektoru vzájemné hybnosti nám ušetří jisté problémy s nedostatečnou statistikou v oblasti malých  $q$ . Na následujících grafech uvedu korelační funkce za vlivu Coulombické interakce a silné interakce. V grafech jsou zobrazeny chyby.



Obrázek 6.10: Korelační funkce za vlivu kompletní FSI. Generování složek hybnosti jednotlivých částic v laboratorní soustavě.



Obrázek 6.11: Korelační funkce za vlivu kompletní FSI. Generování velikosti vektoru hybnosti v těžišťové soustavě.

Vidíme, že chyba v krajních binech vzroste, jestliže použijeme generátor složek v laboratorní soustavě. Chyby v krajních binech jsou hodně velké v prvním případě, ale v druhém případě se neliší od středních binů. Zde vidíme, nakolik je generování velikosti vektoru vzájemné hybnosti užitečné.

# Kapitola 7

## Zpracování dat

Data byla získána při srážkách jader uranu. Jednalo se o centrální srážky. Centralita byla do 12%. Energie při srážkách byla 193  $GeV$  na nukleon.

Data jsem obdržel v padesáti souborech. Soubory obsahovaly stromové struktury typu TTree, tedy stromové struktury `Rootu`. TTree obsahoval mnoho "případů" (event). Každý z případů se týkal jednotlivé srážky dvou jader. V každém případě vznikaly v proměnném množství pionů, jinak řečeno vznikaly dráhy pionů (tracks). Každý případ obsahuje číslo určující počet pionů vzniklých při srážce.

Každý TTree nesl mnoho informací. Jinak řečeno, na každý případ srážky se vztahovalo mnoho informací. Pro mě byly podstatné tyto: hybnost jednotlivých pionů a jejich znaménko. Potom jsem samozřejmě musel znát počet pionů vzniklých při jedné srážce.

Pro obdržení spekter jsem napsal skript, který je uveden v příloze, kde se nachází i jeho header file.

Korelační funkci jsem získal z těchto dat dělením dvou spekter velikosti vzájemných hybností dvou částic. Získání korelační funkce sestávalo z několika kroků.

Prvně jsem musel vybrat všechny částice, které jsou schopné vzájemně interagovat. Je evidentní, že to budou částice vzniklé při jedné srážce (nebudou reagovat s částicemi z jiných případů z důvodů vzdálenosti, nebo odlišného času produkce). Zaznamenávám tedy spektrum vzájemných hybností dvou částic z jednoho případu. Musím vzít všechny možné kombinace.

V druhém kroku zaznamenávám spektrum  $q$  částic, jež neměly možnost se ovlivnit. Ty získáme takzvaným mixováním. Vezmeme částice z jednoho případu a spočteme  $q$  s každou částicí z následujícího případu.

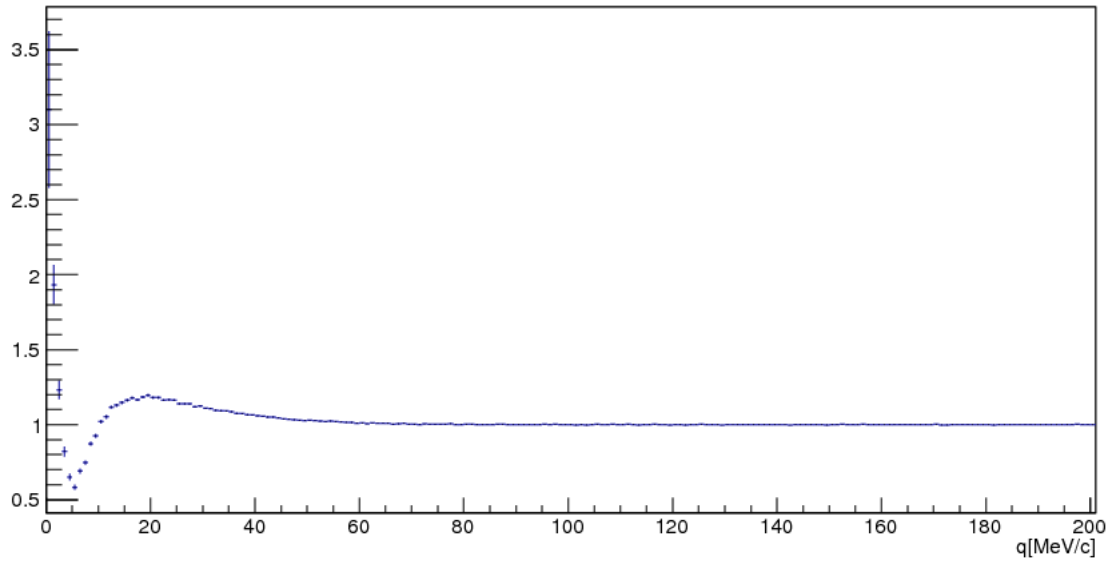
Je evidentní, že kombinací částic, které se navzájem ovlivňují, je mnohem méně než těch, co se neovlivňují. Proto zaznamenávám  $q$  ze všech případů ovlivňujících se částic. Nemusím zaznamenat všechny kombinace neovlivňujících se částic. Proto zaznamenávám vždy jen  $q$  z dvou po sobě jdoucích případů.

Třetí krok je dělení nemixovaného spektra mixovaným. Spektra jsou zaznamenána v histogramech. Histogramy budou mít odlišný počet záznamů. Proto je nutné

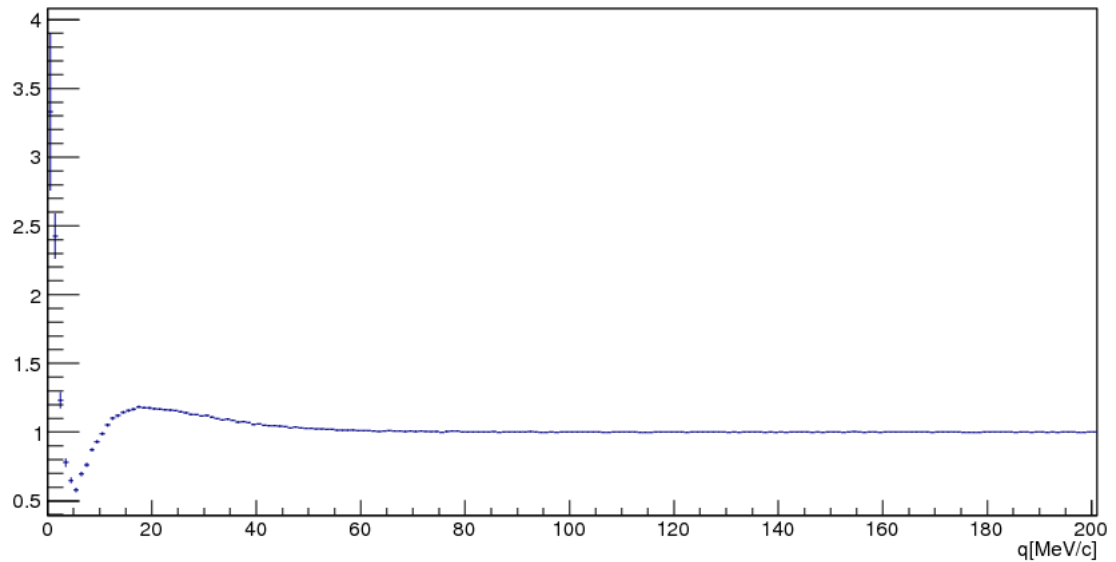
výsledný histogram škálovat.

## 7.1 Korelační funkce identických částic

První uvedu korelační funkci  $\pi^- \pi^-$ . Jako druhou si uvedu korelační funkci pro  $\pi^+$  a  $\pi^+$ .



Obrázek 7.1: Korelační funkce získaná z dat. Příklad  $\pi^- \pi^-$ .

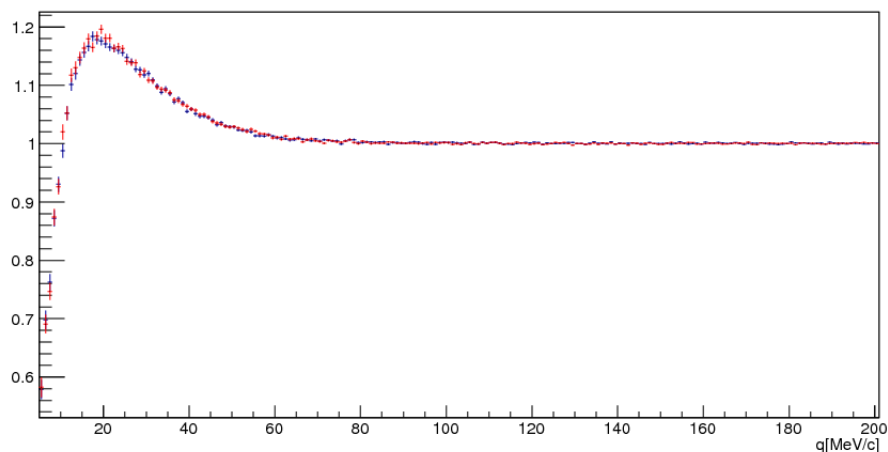


Obrázek 7.2: Korelační funkce získaná z dat. Příklad  $\pi^+ \pi^+$ .

Podívejme se prvně, kde se uvedené funkce liší. V prvních pěti binech, tj.  $q < 5\text{MeV}$  bychom čekali, že korelační funkce klesne na nulu. Připomněl bych například obrázek (??).

Důvod, proč se funkce v těchto binech zvedá, je tzv. "track splitting". Jde o chybu vyhodnocovacích algoritmů. Jedna dráha je mylně interpretována jako dvě s malým vzájemným  $q$ . Track splitting se objevuje v oblasti  $q < 5\text{MeV}$ . Záznam v těchto binech můžeme ignorovat.

Jestliže zanedbáme tyto biny, funkce jsou totožné, jak vidíme z následujícího grafu. To je v souladu s logikou experimentu.

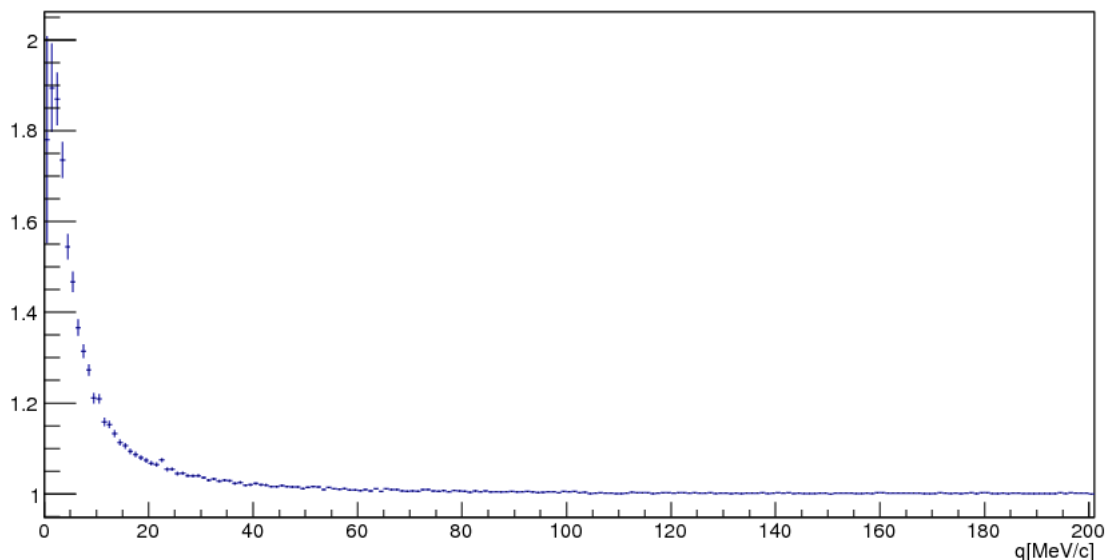


Obrázek 7.3: Korelační funkce. Modrá:  $\pi^- \pi^-$ , červená:  $\pi^+ \pi^+$ .

Rozeberu, co vidím v korelačních funkcích. V blízkosti nuly korelační funkce klesá k nule díky dominanci repulsivních sil z Coulombické interakce. Její vliv klesá rychleji s rostoucím  $q$  než vliv kvantové statistiky. Kvantová statistika se uplatňuje v oblastech vyšších  $q$ . Vliv kvantové statistiky převládne v oblastech vyšších  $q$  a korelační funkce stoupne nad jedna a dosáhne vrcholu píku. Poté klesá i vliv kvantové statistiky a funkce se blíží k jedné.

## 7.2 Korelační funkce $\pi^+ \pi^-$

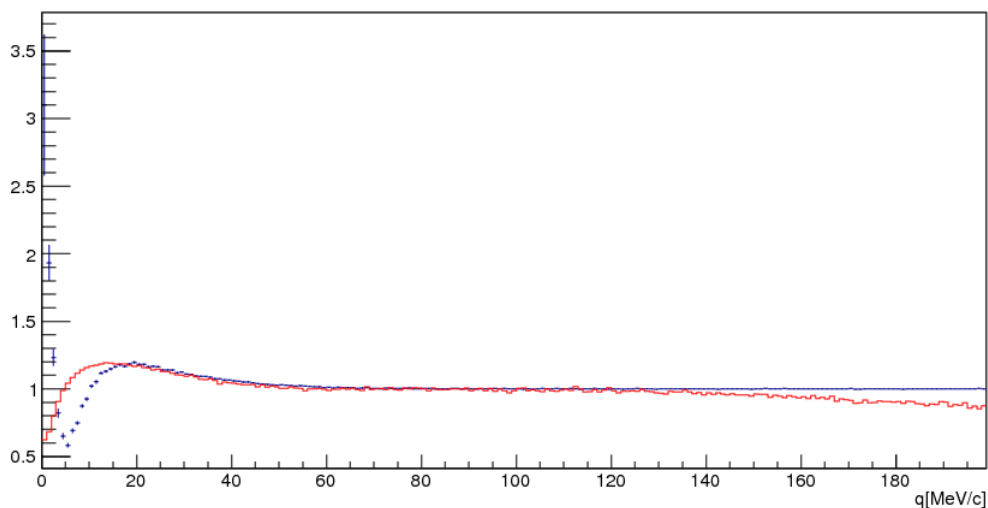
Dále uvedu graf korelační funkce pro případ  $\pi^+ \pi^-$ .



Obrázek 7.4: Korelační funkce získaná z dat. Příklad  $\pi^+ \pi^-$ .

Coulombická interakce je v tomto případě přitažlivá. Proto korelační funkce neklesá v oblasti malých  $q$ . Kvantová statistika i Coulombická interakce zvedá korelační funkci nad jedna. Na rozdíl od předchozího případu, kdy kvantová statistika a Coulombická interakce působily proti sobě, zde působí stejným směrem. Z toho důvodu začíná korelační funkce na vysokých hodnotách a monotónně klesá.

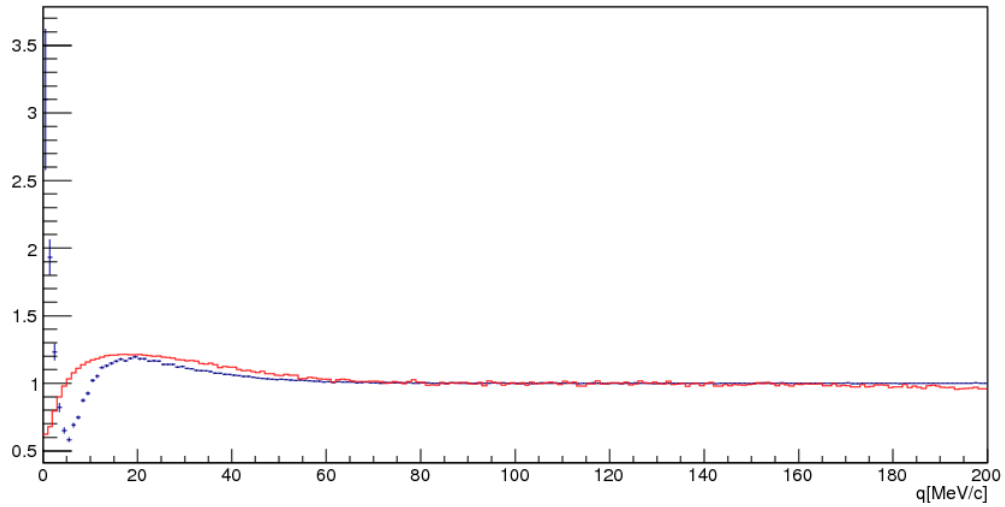
Pomocí nasimulované korelační funkce se pokusím určit rozměr zdroje. Nasimulovanou korelační funkci vykreslím do grafu korelační funkce získané z dat. První volím zdroj s Gaussovým rozložením o velikosti  $14fm$ .



Obrázek 7.5: Modrá: Korelační funkce  $\pi^+ \pi^+$  získaná z dat, červená: simulovaná korelační funkce  $r^* = 14fm$

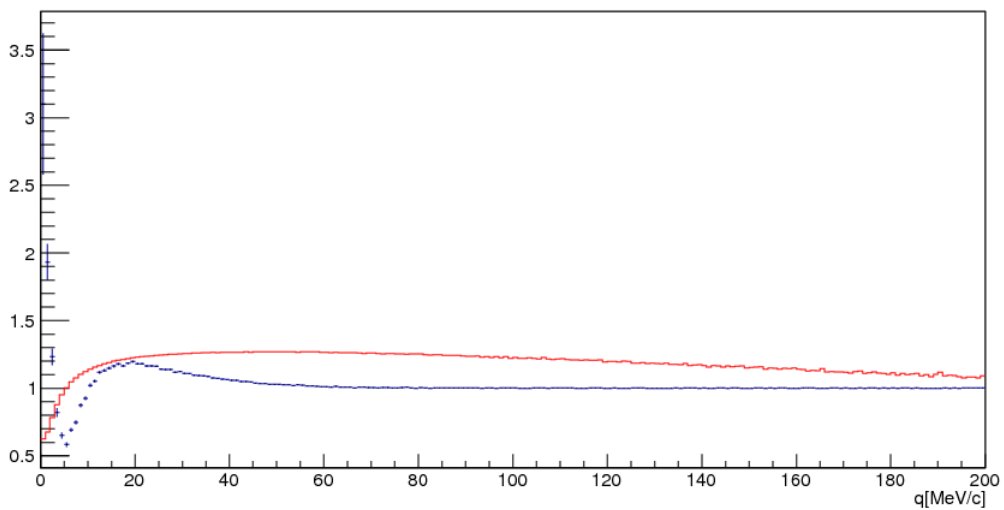


Vidím, že pík simulované korelační funkce je u nižšího  $q$ . Z toho se dá usoudit, že zdroj je menší. Vykreslím se simulovanou korelační funkcí pro  $r^* = 10 fm$



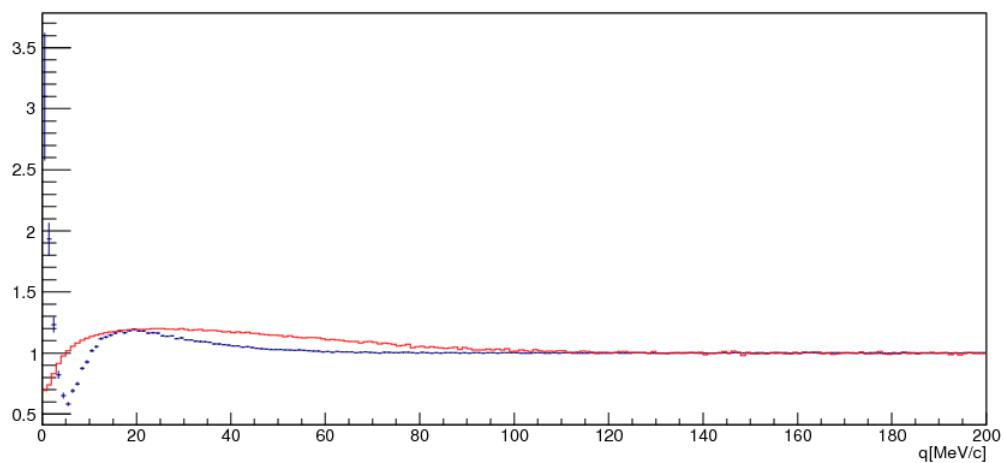
Obrázek 7.6: Modrá: Korelační funkce  $\pi^+ \pi^+$  získaná z dat, červená: simulovaná korelační funkce  $r^* = 10 fm$

Pík simulované funkce se přiblížil k druhému píku, ale stále je na nižších  $q$ . Na následujícím grafu vidím korelační funkci pro  $r^* = 2 fm$ .

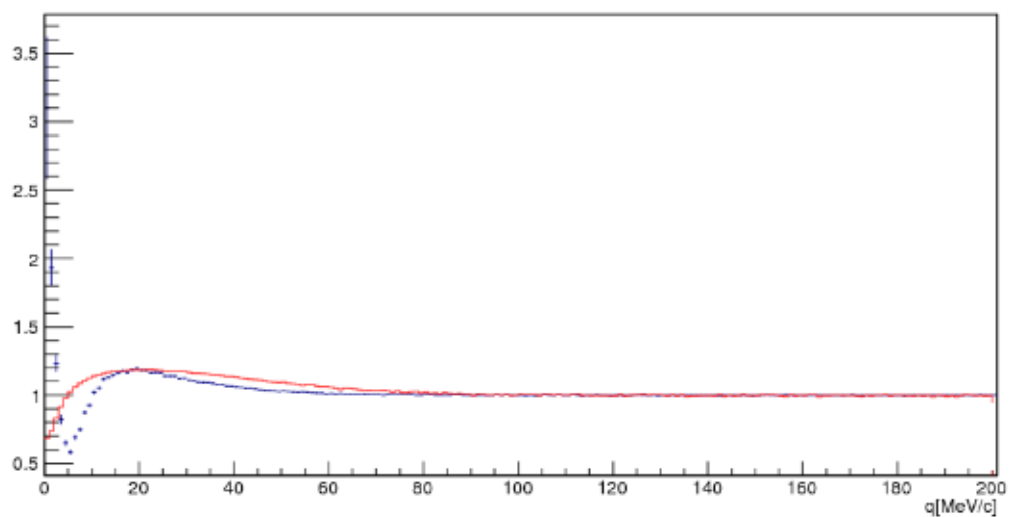


Obrázek 7.7: Modrá: Korelační funkce  $\pi^+ \pi^+$  získaná z dat, červená: simulovaná korelační funkce  $r^* = 2 fm$

Zde vidíme, že pík simulované korelační funkce je rozlehlejší a vrchol je na větších  $q$ . Je evidentní, že hledaný rozměr je mezi 10 a 2  $fm$ . Dále uvedu korelační funkci pro rozměry 6 a 8  $fm$ .

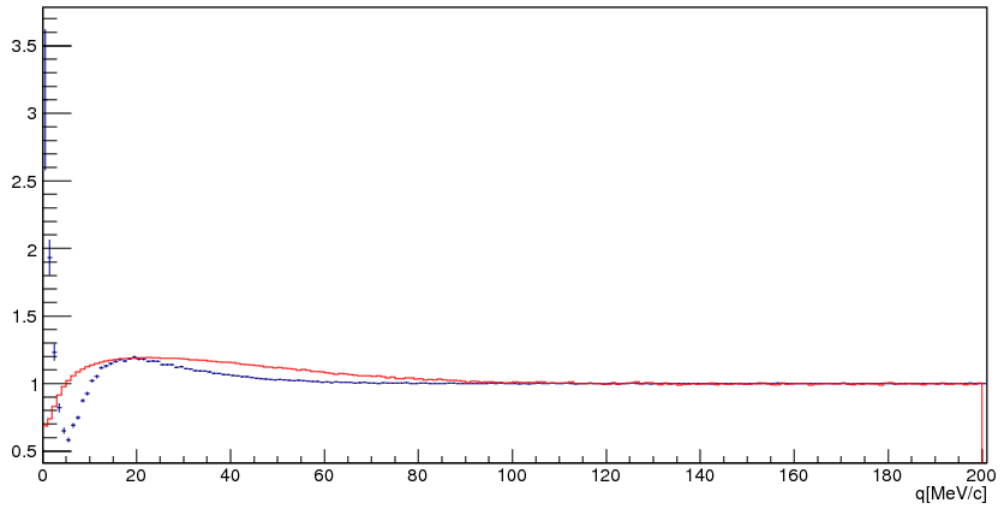


Obrázek 7.8: Modrá: Korelační funkce  $\pi^+ \pi^+$  získaná z dat, červená: simulovaná korelační funkce  $r^* = 6 fm$



Obrázek 7.9: Modrá: Korelační funkce  $\pi^+ \pi^+$  získaná z dat, červená: simulovaná korelační funkce  $r^* = 8 fm$

Vidíme, že simulovaná korelační funkce pro  $r^* = 8fm$  má pík u trochu menších  $q$  a korelační funkce pro  $r^* = 6fm$  má pík u větších. Dále uvedu simulovanou korelační funkci pro  $r^* = 7fm$ .



Obrázek 7.10: Modrá: Korelační funkce  $\pi^+ \pi^+$  získaná z dat, červená: simulovaná korelační funkce  $r^* = 7fm$

Zde vidíme, že pík této simulované korelační funkce je nejbližší píku korelační funkce z dat. Proto řekneme, že u produkční oblasti částic skutečné korelační funkce  $r^* \simeq 7fm$ .

# Kapitola 8

## Závěr

Práce se věnovala problematice korelačních funkcí. Byla nastíněna teorie vedoucí ke vzniku korelačních funkcí a rozebrána pro několik konkrétních případů. Dále byly zmíněny předpoklady hladkosti a současnosti. Od teorie jsme přešli k simulacím korelačních funkcí pomocí programu.

Zmínili jsme požadavky, které na program klademe. Rozebrali jsme podmínky experimentu, ze kterých musíme vycházet. Dále jsme se zabírali způsobem generování jednotlivých vlastností částic. Následovně jsme viděli začlenění jednotlivých částic do dvoučásticového případu, pomocí kterého jsme zkoumali interakci částic a sestrojili korelační funkce. Ty byly vykreslené pro případ pouze silné interakce, kvantové statistiky, Coulombické interakce a dále pro příklad kompletního FSI a kvantové statistiky. Vlivy jednotlivých jevů byly rozebrány.

Dále jsme se stručně věnovali zpracování dat. Z dat jsme získali korelační funkce pro případy  $\pi^+ \pi^+$ ,  $\pi^- \pi^-$  a  $\pi^+ \pi^-$ .

Pomocí programu jsme vykreslili korelační funkce. Korelační funkce vycházely přesně jako funkce v [5]. Pro všechny případy se funkce shodovaly s článkem. Na rozdíl programu `fsi4pi3n2c_histo.f` jsem korelační funkce simuloval v C++, ne ve Fortranu. C++ je dnes mnohem populárnější a proto přínos práce spočívá mimo jiné v napsání programu v tomto jazyce. Podle mých informací je velmi málo programů, které by se zabývaly korelačními funkcemi a toto je první v C++. Jelikož funkce vycházely velmi dobře, je možné použití tohoto programu v praxi.

Program byl psaný pro nabitě piony. To ale neznamená, že by nebyl použitelný na jiné částice. Ke kaonům by se dalo přejít triviální záměnou, která spočívá v nahrazení Bohrovského poloměru. Potom by interakce kaonů program zvládal stejně jako piony.

Program by se do budoucna mohl silně rozvíjet. Logický krok je rozšíření jeho působnosti na další částice. Například přechod na protony by nebyl složitý. Rozšíření na další části je logický, ale ne jediný možný krok. Dále je možné zpracovat na generátorech náhodných čísel, které používám například pro generování složek hybností částic. Tady jsou možnosti nepřeborné, neboť každý experiment je specifický ve svých podmínkách a potřebujeme generátory, které se nejlépe přiblíží realitě.

Jak je vidět, možností kam se ubírat je mnoho. Pro pokračování lze najít účinnou motivaci, neboť korelační femtoskopie je velmi mocná a rozvíjející se metoda, která nalézá uplatnění na mnoha experimentech.

# Appendices

# Dodatek A

## Program

```
#include<iostream>
#include<string.h>
#include <fstream>
#include<math.h>
#include <stdlib.h>
#include <TVersionCheck.h>
#include "TRandom.h"
#include "TFile.h"
#include "TH1.h"
#include "TH2.h"
#include "TH3.h"
#include "TTree.h"
#include "TNtuple.h"
#include "TMath.h"
#include<stdio.h>
#include "TComplex.h"
#include <cmath>
#include "gsl/gsl_sf_legendre.h"
#include "gsl/gsl_sf_laguerre.h"
#include "gsl/gsl_sf_hyperg.h"
#include "gsl/gsl_sf_ellint.h"
#include "gsl/gsl_sf_expint.h"
#include "gsl/gsl_sf_zeta.h"
#include "gsl/gsl_sf_airy.h"
#include "gsl/gsl_sf_coupling.h"
#include "Math/SpecFuncMathMore.h"
#include "gsl/gsl_sf_coulomb.h"
#include "gsl/gsl_sf.h"
#include <gsl/gsl_sf_bessel.h>
#include <gsl/gsl_sf_result.h>
```

```

using namespace std;

#define N 1000000
#define PI 3.14159265
#define BohrRadius 387.5
#define CEuler 0.5772
#define m 139.57018

//E(pion)=139,57018 MeV

double rnd( double m1,double m2)
{
    double intrv=m2-m1;
    double res=((double)rand()/((double)RAND_MAX)* intrv + m1;

    return res;
}

double rozmazat( double a)
{
    double b=sqrt(a);
    double c=gRandom->Gaus(0,b);
    double E=a+c;
    if (E<0)
    {
        E=-E;
    }
    return E;
}

struct castice
{
    double hmotnost;
    double poloha[4];
    double polohaT[4];
    double hybnost[4];
    double hybnostT[4];
    double hybnostR[4];
}

```



```
    double hybnostRT[4];  
};
```

```
class Pripad  
{  
    public:  
        Pripad();  
        ~Pripad();  
  
        castice prvni;  
        castice druha;  
        double V[3];  
        double gama[3];  
        double hybnostT[3];  
        double hybnostRT[3];  
        double polohaT[3];  
        double poloha[3];  
        double skalarKR;  
        double q;  
        double k;  
        double qR;  
        double r;  
        double rNT;  
  
};
```

```
Pripad::Pripad()  
{
```

```
    prvni.hmotnost=139.57018;  
    druha.hmotnost=139.57018;
```

```
    double T1=139.57018*139.57018;  
    double T2=139.57018*139.57018;
```

```
    for (unsigned int i = 1; i <4; i++)  
    {
```

```
        prvni.hybnost[i] = rnd( 0.0, 50.0);  
        druha.hybnost[i] = rnd( 0.0, 50.0);  
        prvni.hybnostR[i] = rozmazat(prvni.hybnost[i] );  
        druha.hybnostR[i] = rozmazat( druha.hybnost[i] );  
        prvni.poloha[i]=gRandom->Gaus(0,2.15);
```

```

        druha.poloha[i]=gRandom->Gaus(0,2.15);
        T1=T1+prvni.hybnost[i]*prvni.hybnost[i];
        T2=T2+druha.hybnost[i]*druha.hybnost[i];
    }
    druha.polohaT[0]=gRandom->Gaus(0,0.0);
    prvni.polohaT[0]=gRandom->Gaus(0,0.0);
    prvni.hybnost[0]=sqrt(T1);
    druha.hybnost[0]=sqrt(T2);
    V[0]=(prvni.hybnost[1]+druha.hybnost[1])/(prvni.hybnost[0]+druha.hybnost[0]);
    gama[0]=1/sqrt((1-V[0]*V[0]));

    //***** transformace

    double tmp1=prvni.hybnost[0];
    double tmp2=druha.hybnost[0];

    for(int i=1; i<4; i++)
    {
        prvni.hybnostT[i]=-gama[i-1]*V[i-1]*tmp1+gama[i-1]*prvni.hybnost[i];
        prvni.hybnostT[0]=gama[i-1]*tmp1-gama[i-1]*V[i-1]*prvni.hybnost[i];

        druha.hybnostT[i]=-gama[i-1]*V[i-1]*tmp2+gama[i-1]*druha.hybnost[i];
        druha.hybnostT[0]=gama[i-1]*tmp2-gama[i-1]*V[i-1]*druha.hybnost[i];

        prvni.hybnostRT[i]=-gama[i-1]*V[i-1]*tmp1+gama[i-1]*prvni.hybnostR[i];
        prvni.hybnostRT[0]=gama[i-1]*tmp1-gama[i-1]*V[i-1]*prvni.hybnostR[i];

        druha.hybnostRT[i]=-gama[i-1]*V[i-1]*tmp2+gama[i-1]*druha.hybnostR[i];
        druha.hybnostRT[0]=gama[i-1]*tmp2-gama[i-1]*V[i-1]*druha.hybnostR[i];

        prvni.polohaT[i]=-gama[i-1]*V[i-1]*tmp1+gama[i-1]*prvni.poloha[i];
        prvni.polohaT[0]=gama[i-1]*tmp1-gama[i-1]*V[i-1]*prvni.poloha[i];

        druha.polohaT[i]=-gama[i-1]*V[i-1]*tmp2+gama[i-1]*druha.poloha[i];
        druha.polohaT[0]=gama[i-1]*tmp2-gama[i-1]*V[i-1]*druha.poloha[i];

        tmp1=prvni.hybnostT[0];
        tmp2=druha.hybnostT[0];

        if(i!=3)
        {
            V[i]=(prvni.hybnost[i+1]+druha.hybnost[i+1])/(tmp1+tmp2);

```

```

        gama[i]=1/sqrt((1-V[i]*V[i]));
    }
}

for(int i=0; i<3; i++)
{
    hybnostT[i]=prvni.hybnostT[1+i]-druha.hybnostT[1+i];
    hybnostRT[i]=prvni.hybnostRT[i+1]-druha.hybnostRT[i+1];
    polohaT[i]=prvni.polohaT[i+1]-druha.polohaT[i+1];
    poloha[i]=prvni.poloha[i+1]-druha.poloha[i+1];
}

q=0.0;
for(int i=0; i<3; i++)
{
    q=q+hybnostT[i]*hybnostT[i];
}
q=sqrt(q);
k=q/2.0;

r=0.0;
for(int i=0; i<3; i++)
{
    r=r+polohaT[i]*polohaT[i];
}
r=sqrt(r);
rNT=0.0;
for(int i=0; i<3; i++)
{
    rNT=rNT+poloha[i]*poloha[i];
}
rNT=sqrt(rNT);

polohaT[0]=gRandom->Gaus(0,sqrt(2)*2.5);
polohaT[1]=gRandom->Gaus(0,sqrt(2)*2.5);
polohaT[2]=gRandom->Gaus(0,sqrt(2)*2.5);
r=sqrt(polohaT[0]*polohaT[0]+polohaT[1]
*polohaT[1]+polohaT[2]*polohaT[2]);

r=6.0;
double alfa=rnd(-90.0,90.0);
alfa=alfa*3.14/180.0;
double beta=rnd(0.0,360.0);
beta=beta*3.14/180.0;
polohaT[2]=r*sin(alfa);

```

```

polohaT[0]=r*cos(alfa)*cos(beta);
polohaT[1]=r*cos(alfa)*sin(beta);

q=rnd( 0.0, 1000.0);
double qkvadrat=q*q;
double tmp11=rnd( 0.0, 1.0);
double tmp22=rnd( 0.0, 1.0);
if(tmp22<tmp11)
{
    double pom=tmp11;
    tmp11=tmp22;
    tmp22=pom;
}
hybnostT[0]=sqrt(tmp11*qkvadrat);
hybnostT[1]=sqrt((tmp22-tmp11)*qkvadrat);
hybnostT[2]=sqrt((1.0-tmp22)*qkvadrat);

hybnostT[0]=0.0;
hybnostT[1]=0.0;
hybnostT[2]=q;
hybnostRT[0]=rozmazat(hybnostT[0]);
hybnostRT[1]=rozmazat(hybnostT[1]);
hybnostRT[2]=rozmazat(hybnostT[2]);
qR=0.0;
for(int i=0; i<3; i++)
{
    qR=qR+hybnostRT[i]*hybnostRT[i];
}
qR=sqrt(qR);

skalarKR=0.0;
for(int i=0; i<3; i++)
{
    skalarKR=skalarKR+hybnostT[i]*polohaT[i];
}
skalarKR=skalarKR/2.0/197.3;

}

Pripad::~Pripad()
{
}

```

```

struct vzajemna
{
    double x;
    double y;
    double z;
};

void priradVz(vzajemna &a, vzajemna &c, vzajemna &d, Pripad &b)
{
    a.x=b.prvni.polohaT[1]-b.druha.polohaT[1];
    a.y=b.prvni.polohaT[2]-b.druha.polohaT[2];
    a.z=b.prvni.polohaT[3]-b.druha.polohaT[3];

    c.x=b.prvni.hybnostT[1]-b.druha.hybnostT[1];
    c.y=b.prvni.hybnostT[2]-b.druha.hybnostT[2];
    c.z=b.prvni.hybnostT[3]-b.druha.hybnostT[3];

    d.x=b.prvni.hybnostRT[1]-b.druha.hybnostRT[1];
    d.y=b.prvni.hybnostRT[2]-b.druha.hybnostRT[2];
    d.z=b.prvni.hybnostRT[3]-b.druha.hybnostRT[3];
}

void priradVz(vzajemna &a, Pripad &b)
{
    a.x=b.prvni.poloha[1]-b.druha.poloha[1];
    a.y=b.prvni.poloha[2]-b.druha.poloha[2];
    a.z=b.prvni.poloha[3]-b.druha.poloha[3];
}

double skalar(vzajemna &prvni, vzajemna &druha)
{
    double pom=prvni.x*druha.x+prvni.y*druha.y+prvni.z*druha.z;
    return pom;
}

double skalar(double prvni[3], double druha[3])
{
    double pom=0.0;
    for(int i=0; i<3; i++)
    {
        pom=pom+prvni[i]*druha[i];
    }
    return pom;
}

```

```

}

double kvadrat(vzajemna &prvni)
{
    double vrat=prvni.x*prvni.x+prvni.y*prvni.y+prvni.z*prvni.z;
    return vrat;
}

double absolHod(vzajemna &prvni)
{
    double vrat=prvni.x*prvni.x+prvni.y*prvni.y+prvni.z*prvni.z;
    return sqrt(vrat);
}

double funkce(double x)
{
    double k=sin(x);
    return k;
}

double hybnostCelku(Pripad &pripad)
{
    double tmp=0.0;
    for(int i=1; i<4; i++)
    {
        tmp=tmp+(pripad.prvni.hybnost[i]+pripad.druha.hybnost[i])
            *(pripad.prvni.hybnost[i]+pripad.druha.hybnost[i]);
    }
    tmp=sqrt(tmp);
    return tmp;
}

```

```

TComplex EIDC(double eta)
{
    double zr=1.0;
    double zi=eta;
    gsl_sf_result lnrstr, argstr;
    gsl_sf_result *lnr=&lnrstr;
    gsl_sf_result *arg=&argstr;
    int gammafunkce=
    gsl_sf_lngamma_complex_e(zr, zi, lnr, arg);
    double logAbsVal=lnrstr.val;
}

```

```

        double argument=argstr.val;
        double AbsVal=exp (logAbsVal);
        double Rel=AbsVal*cos(argument);
        double Img=AbsVal*sin(argument);
        TComplex Z8 (Rel,Img);
        TComplex A (AbsVal,0.0);
        Z8=Z8/A;
        return Z8; //funguje
    }

double ACP(double eta)
{
    double x=1/eta;
    double y=6.2831853/x;
    double ACP;
    if(y>100.0)
    {
        ACP=0.0;
    }
    else
    {
        ACP=y/(exp(y)-1);
    }

    return ACP;
}

TComplex FAS(double RKS,TComplex Eidc,double eta)
{
    double D1=log (RKS)*eta;
    double D2=eta*eta/RKS; //zeptat se
    double Rel=cos(D1);
    double Img=sin(D1);
    TComplex Z1 (Rel, Img);
    Z1=Z1/Eidc;
    TComplex FAS (1.0,-D2);
    FAS=FAS*Z1;
    TComplex tmp (cos(RKS),sin(RKS));
    FAS=FAS-tmp;
    FAS=FAS*eta/RKS;
    FAS=FAS/Z1;
    double ACHR=ACP(eta);
    FAS=FAS/ACHR;

    return FAS;
}

```

```
}
```

```
class Generator  
{  
    public:  
    Generator();  
    ~Generator();  
    double r;  
};
```

```
Generator::Generator()  
{  
    r=gRandom->Gaus(0.0,3.14);  
}
```

```
Generator::~~Generator()  
{  
}
```

```
int main()  
{
```

```
//    TH1::SetDefaultSumw2();  
TFile file("File5.root","recreate");  
TH1D h1("rx","rx",100,-10,10);  
TH1D h2("ry","ry",100,-10,10);  
TH1D h3("rz","rz",100,-10,10);  
TH1D h4("r","r",100,0,40);  
TH1D h5("rNT","rNT",100,-10,10);  
  
TH1D h6("q","q",50,0,50);  
TH1D h7("q2","q2",200,0,200);  
TH1D h8("qR","qR",200,0,200);  
  
TH1D h10("etaR","etaR",30,0,30);  
TH1D h11("Ac","Ac",30,0,30);  
TH1D h12("hn","hn",30,0,30);  
TH1D h13("ImXn","ImXn",30,0,30);
```



```

TH1D h20("kvantova","kvantova",200,0,200);
TH1D h21("Silna","Silna",200,0,200);
TH1D h22("silna","silna",200,0,200);

TH1D h30("F_regular","F_regular",200,0,200);
TH1D h31("G_singular","G_singular",200,0,200);

TH1D h40("ConHyp","ConHyp",200,0,200);

TH1D h50("Psi","Psi",50,0,50);
TH1D h51("PsiSym","PsiSym",200,0,200);
TH1D h52("Coul","Coul",50,0,50);

```

```

int pocitadlo=0;
do
{
    Pripad piony;
    h1.Fill(piony.polohaT[0]);
    h2.Fill(piony.polohaT[1]);
    h3.Fill(piony.polohaT[2]);
    double r=piony.r;
    if(r<1.0)
    {
        continue;
    }
    else
    {
        pocitadlo++;
    }
    h4.Fill(r);
    double rNT=piony.rNT;
    h5.Fill(rNT);

    double q=piony.q;
    h6.Fill(q);
    h7.Fill(q);
    double k=q/2.0;
    if(k<0.01)
    {
        continue;
    }

    double qR=piony.qR;

```

```

h8.Fill(qR);

double skalar=piony.skalarKR;

double eta=1/(BohrRadius/197.3*k);
double etaR=BohrRadius/197.3*k;
h10.Fill(fabs(etaR));
double Ac=2*PI*eta/(exp(2*PI*eta)-1);
h11.Fill(fabs(etaR),Ac);
TComplex Ac2 (Ac,0.0);
double moje1=gsl_sf_psi_1piy (eta);
double hn=(2*moje1-log(eta*eta))/2.0;
h12.Fill(etaR,hn);
double XnI= Ac/2.0/eta;
h13.Fill(etaR,XnI);
TComplex Xn (hn, XnI);
TComplex bbb=2.0*Xn/BohrRadius*197.3;

double A1[4];
    A1[0]=0.22;
    A1[1]=0.268;
    A1[2]=-0.0139;
    A1[3]=-0.00139;
double s01=36770000;
double A2[4];
    A2[0]=-0.0444;
    A2[1]=-0.0857;
    A2[2]=-0.00221;
    A2[3]=-0.000129;
double s02=-21620000;

double sth=4*m*m;
double x=q/sqrt(sth);
double w1=sqrt(m*m+(q/2.0)*(q/2.0));
double s=4*w1*w1;

double suma2=0.0;
for (int i=0; i<4; i++)
{
    suma2=suma2+A2[i]*pow(x,2*i);
}
double K2=2/sqrt(s)*(sth-s02)/(s-s02)*suma2;
double K2Reverse=1.0/K2;

double suma1=0;
for (int i=0; i<4; i++)

```

```

{
    suma1=suma1+A1[i]*pow(x,2*i);
}
double K1=2/sqrt(s)*(sth-s01)/(s-s01)*suma1;
double K1Reverse=1.0/K1;

TComplex aaa1 (K1Reverse,0.0);
TComplex aaa2 (K2Reverse,0.0);
TComplex fCoulomb11 =aaa1-bbb;
TComplex fCoulomb22 =aaa2-bbb;
TComplex jedna (1.0,0.0);
fCoulomb11=jedna/fCoulomb11;
fCoulomb22=jedna/fCoulomb22;
TComplex fCoulomb2=0.66666*fCoulomb11
+0.33333*fCoulomb22;
TComplex fCoulomb3=fCoulomb22;
double fCoul=TComplex::Abs(fCoulomb2);
double w=1+cos(skalar*2.0);
h20.Fill(q,w);
h21.Fill(q,fCoul);

TComplex jmenovatel (K2Reverse,-k);
TComplex f=jedna/jmenovatel;

TComplex pomExp1 (0.0,-skalar);
TComplex e1=TComplex::Exp(pomExp1);

TComplex pomExp2 (0.0,k*r/197.0);
TComplex e2=TComplex::Exp(pomExp2);

TComplex psiS=(e1+f*e2/r*197.0);
TComplex psiS2=TComplex::Conjugate(psiS);
double vahaSilna=psiS*psiS2;
h22.Fill(q,vahaSilna);

gsl_sf_result  Ggsl,Fgsl,Gpgsl,Fpgsl;
gsl_sf_result *Gstruct;
gsl_sf_result *Fstruct;
gsl_sf_result *Gp=&Gpgsl;
gsl_sf_result *Fp=&Fpgsl;
Gstruct=&Ggsl;
Fstruct=&Fgsl;

double G_L, F_L;

```

```

double *expG_L=&G_L;
double *expF_L=&F_L;
int vysledek;
double rho=k*r/197.0;

vysledek=gsl_sf_coulomb_wave_FG_e
(eta, rho, 0.0, 0 ,Fstruct ,Fp
,Gstruct ,Gp , expF_L, expG_L);
double F_regular=Fgsl.val;
double G_singular=Ggsl.val;
F_regular=F_regular*sqrt(Ac);
h30.Fill(q,F_regular);
G_singular=G_singular*sqrt(Ac);
h31.Fill(q,G_singular);
TComplex Gfun(G_singular,F_regular);

TComplex S12 (1.0,0.0);
TComplex S21 (1.0,0.0);
TComplex F12 (1.0,0.0);
TComplex F21 (1.0,0.0);
double H=skalar;
double C=cos(H);
double S=sin(H);
double RHO=r*k/197.0;
double RHOP=RHO+H;
double RHOM=RHO-H;
TComplex ALF (0.0,-eta );
TComplex ALF1 (-1.0,-eta );
double ZR12;
double ZI12;
double ZR21;
double ZI21;
double Eidc=EIDC(eta);
if(RHOP<20.0)
{
    TComplex Z12 (0.0,RHOP);
    int j=0;
    do
    {
        j=j+1;
        double j2=(double)j;
        TComplex jj (j2,0.0);
        TComplex A=(ALF1+jj)/jj/jj;
        S12=S12*A*Z12;
        F12=F12+S12;
        ZR12=S12.Re();
    }
}

```

```

        ZR12=fabs(ZR12);
        ZI12=S12.Im();
        ZI12=fabs(ZI12);
    }while (ZR12+ZI12>0.00001);
}
else
{
    F12=FAS(RHOP,Eidc,eta);
}
if(RHOM<15.0)
{
    TComplex Z21 (0.0,RHOM);
    int j=0;
    do
    {
        j=j+1;
        double j2=(double)j;
        TComplex jj (j2,0.0);
        TComplex A=(ALF1+jj)/jj/jj;
        S21=S21*A*Z21;
        F21=F21+S21;
        ZR21=S21.Re();
        ZR21=fabs(ZR21);
        ZI21=S21.Im();
        ZI21=fabs(ZI21);
    }while (ZR21+ZI21>0.00001);

}
else
{
    F21=FAS(RHOP,Eidc,eta);
}

double tttt=TComplex::Abs(F12);
h40.Fill(q,tttt);

TComplex fCoulombP (0.232/197.0,0.0);
TComplex rCom (r/197.3,0.0);
TComplex psiC=e1*F12*sqrt(Ac);
double vahaSymC=TComplex::Abs(psiC)
*TComplex::Abs(psiC);
h52.Fill(q,vahaSymC);
TComplex f2= Gfun*fCoulomb3 /rCom;

TComplex psiCS=sqrt(Ac)*(e1*F12+f2);;
double vahaKompletni=TComplex::Abs(psiCS)

```

```

        *TComplex::Abs(psiCS);
        h50.Fill(q,vahaKompletni);

        pomExp1(0.0,skalar);
        e1=TComplex::Exp(pomExp1);
        TComplex pomocnaPsi2=sqrt(Ac)*(e1*f21+f2);
        pomocnaPsi2=TComplex::Conjugate(pomocnaPsi2);
        TComplex pomocnaLast=psiCS*pomocnaPsi2;
        double vahaSym=TComplex::Abs(psiCS)*
        TComplex::Abs(pomocnaPsi2)+pomocnaLast.Re();
        h51.Fill(q,vahaSym);

    }while(pocitadlo<N);

    h1.Write();
    h2.Write();
    h3.Write();
    h4.Write();
    h5.Write();
    h6.Write();
    h7.Write();
    h8.Write();
    h10.Write();
    h11.Write();
    h12.Write();
    h13.Write();
    h20.Write();
    h21.Write();
    h22.Write();
    h30.Write();
    h31.Write();
    h40.Write();
    h50.Write();
    h51.Write();
    h52.Write();
    file.Close();
    return 0;
}

```

# Dodatek B

## Script

### B.1 script.c

```
#define script_cxx
#include "script.h"
#include <TH2.h>
#include <TStyle.h>
#include <TCanvas.h>

void script::Loop()

{
    TH1::SetDefaultSumw2();
    TFile file("test.root","recreate");
    TH1F *piplus=new TH1F("piplus","piplus",1000,0.0,1.0);
    TH1F *piplusJ=new TH1F("piplusJ","piplusJ",1000,0.0,1.0);
    TH1F *piminus=new TH1F("piminus","piminus",1000,0.0,1.0);
    TH1F *piminusJ=new TH1F("piminusJ","piminusJ",1000,0.0,1.0);
    TH1F *pi=new TH1F("pi","pi",1000,0.0,1.0);
    TH1F *piJ=new TH1F("piJ","piJ",1000,0.0,1.0);
    TH1F *qPlus=new TH1F("qPlus","qPlus",1000,0.0,1.0);
    TH1F *qMinus=new TH1F("qMinus","qMinus",1000,0.0,1.0);
    TH1I *znamenkoTH=new TH1I("znamenkoT","znamenkoT",10,-2.0,2.0);
    TH1F *multiplicita=new
    TH1F("multiplicita","multiplicita",710,0.0,710);
    TH1F *multiplicitaPlus=new
    TH1F("multiplicitaPlus","multiplicitaPlus",710,0.0,710);
    TH1F *multiplicitaMinus=new
    TH1F("multiplicitaMinus","multiplicita",710,0.0,710);
    Float_t qJedne;
    int pocitadloPlus=0;
    int pocitadloMinus=0;
    int pocitadloPP=0;
```

```

int pocitadloMM=0;
int pocitadlo=0;
int pocitadloPPJ=0;
int pocitadloMMJ=0;
int pocitadloJ=0;
int znamenko;
int znamenkoT;
Char_t znamenkoPom;
Char_t znamenkoPom2;
int meze;

Float_t qx;
Float_t qy;
Float_t qz;
Float_t q;

Float_t px[kMaxfPrimaryTracks];
Float_t py[kMaxfPrimaryTracks];
Float_t pz[kMaxfPrimaryTracks];
Char_t charge[kMaxfPrimaryTracks];

if (fChain == 0) return;

fChain->SetBranchStatus("*",0);
fChain->SetBranchStatus("fPrimaryTracks",1);
fChain->SetBranchStatus("fPrimaryTracks.fPx",1);
fChain->SetBranchStatus("fPrimaryTracks.fPy",1);
fChain->SetBranchStatus("fPrimaryTracks.fPz",1);
fChain->SetBranchStatus("fPrimaryTracks.fCharge",1);
Long64_t nbytes = 0;
Long64_t nb = 0;

Long64_t ientry = LoadTree(1);
if (ientry < 0) return;
nb = fChain->GetEntry(1);
nbytes += nb;
for(int i=0; i<fPrimaryTracks_; i++)
{
    px[i]=fPrimaryTracks_fPx[i];
    py[i]=fPrimaryTracks_fPy[i];
    pz[i]=fPrimaryTracks_fPz[i];
    charge[i]=fPrimaryTracks_fCharge[i];
}
meze=fPrimaryTracks_;

```



```

for(int i=0; i<fPrimaryTracks_; i++)
{
    qJedne=sqrt(px[i]*px[i]+py[i]*py[i]+pz[i]*pz[i]);
    int chargeP=(int)charge[i];
    if(chargeP==1)
    {
qPlus->Fill(qJedne);
pocitadloPlus++;
    }
    if(chargeP==-1)
{
    qMinus->Fill(qJedne);
pocitadloMinus++;
}
}
}

```

```

Long64_t nentries = fChain->GetEntries();
for (Long64_t jentry=2; jentry<nentries;jentry++)
{
    multiplicita->Fill(fPrimaryTracks_);
    int multiplicita1=0;
    int multiplicita2=0;
    Long64_t ientry = LoadTree(jentry);
    if (ientry < 0) break;
    nb = fChain->GetEntry(jentry);
    nbytes += nb;

    for(int i=0; i<meze; i++)
{
for(int j=0; j<fPrimaryTracks_; j++)
{
    qx=px[i]-fPrimaryTracks_fPx[j];
    qy=py[i]-fPrimaryTracks_fPy[j];
    qz=pz[i]-fPrimaryTracks_fPz[j];
    qJedne=sqrt(fPrimaryTracks_fPx[j]*
        fPrimaryTracks_fPx[j]+fPrimaryTracks_fPy[j]
*fPrimaryTracks_fPy[j]+
        fPrimaryTracks_fPz[j]*fPrimaryTracks_fPz[j]);
    q=sqrt(qx*qx+qy*qy+qz*qz);
    znamenkoPom=fPrimaryTracks_fCharge[j];
    znamenkoT=(int)znamenkoPom;

    if(i==0)
    {

```

```

znamenkoTH->Fill(znamenkoT);
if(znamenkoT==1)
    {
        qPlus->Fill(qJedne);
pocitadloPlus++;
        multiplicita1++;
    }
if(znamenkoT==-1)
{
    qMinus->Fill(qJedne);
    pocitadloMinus++;
        multiplicita2++;
}

}
    }
    znamenkoPom2=charge[i];
    znamenko=(int)znamenkoPom2;
    if(q>1.0)
    {
        continue;
    }
    if((znamenko+znamenkoT)==-2)
    {
pminusJ->Fill(q);
pocitadloMMJ++;
    }
    if((znamenko+znamenkoT)==2)
    {
piplusJ->Fill(q);
pocitadloPPJ++;
    }
    if((znamenko+znamenkoT)==0)
    {
piJ->Fill(q);
pocitadloJ++;
    }
}

for(int ii=i+1; ii< meze; ii++)
{
    qx=px[i]-px[ii];
    qy=py[i]-py[ii];
    qz=pz[i]-pz[ii];
        q=sqrt(qx*qx+qy*qy+qz*qz);

    znamenkoT=(int)charge[ii];
    znamenko=(int)charge[i];

```

```

        if(q>1.0)
        {
continue;
        }
        if(znamenko==znamenkoT*(-1))
        {
pi->Fill(q);
pocitadlo++;
        }
        if((znamenko+znamenkoT)==2)
        {
piplus->Fill(q);
pocitadloPP++;
        }
        if((znamenko+znamenkoT)==-2)
        {
piminus->Fill(q);
pocitadloMM++;
        }
    }
}

for(int i=0; i<fPrimaryTracks_; i++)
{
    px[i]=fPrimaryTracks_fPx[i];
    py[i]=fPrimaryTracks_fPy[i];
    pz[i]=fPrimaryTracks_fPz[i];
    charge[i]=fPrimaryTracks_fCharge[i];
}
meze=fPrimaryTracks_;

multiplicitaMinus->Fill(multiplicita2);
multiplicitaPlus->Fill(multiplicita1);
}

piplus->Write();
piminus->Write();
pi->Write();
piplusJ->Write();
piminusJ->Write();
qPlus->Write();
qMinus->Write();
piJ->Write();
multiplicita->Write();
multiplicitaPlus->Write();
multiplicitaMinus->Write();
znamenkoTH->Write();

```

```

file.Close();
}

```

## B.2 script.h

```

#ifndef script_h
#define script_h

#include <TRoot.h>
#include <TChain.h>
#include <TFile.h>
#include <iostream>

    const Int_t kMaxfPrimaryTracks = 900;
    const Int_t kMaxfGlobalTracks = 1;
    const Int_t kMaxfTowers = 1;
    const Int_t kMaxfVOs = 1;
    const Int_t kMaxfTrigObjs = 1;

class script {
public :
    TTree          *fChain;
    Int_t          fCurrent;
    Int_t          fEventHeader_fEventId;
    Int_t          fEventHeader_fRunId;
    Int_t          fEventHeader_fRefMult;
    Float_t        fEventHeader_fReactionPlaneAngle;
    Int_t          fEventHeader_fNOfPrimaryTracks;
    Int_t          fPrimaryTracks_;
    UInt_t
    fPrimaryTracks_fUniqueID[kMaxfPrimaryTracks];
    Char_t
    fPrimaryTracks_fCharge[kMaxfPrimaryTracks];
    Short_t
    fPrimaryTracks_fFlag[kMaxfPrimaryTracks];
    Float_t
    fPrimaryTracks_fPx[kMaxfPrimaryTracks];
    Float_t
    fPrimaryTracks_fPy[kMaxfPrimaryTracks];
    Float_t
    fPrimaryTracks_fPz[kMaxfPrimaryTracks];

    TBranch
    *b_PicoJetTree_fEventHeader_fEventId;

```

```

TBranch          *b_PicoJetTree_fEventHeader_fRunId;
TBranch          *b_PicoJetTree_fEventHeader_fRefMult;
TBranch
*b_PicoJetTree_fEventHeader_fReactionPlaneAngle;
TBranch
*b_PicoJetTree_fEventHeader_fNOfPrimaryTracks;
TBranch          *b_PicoJetTree_fPrimaryTracks_;
TBranch          *b_fPrimaryTracks_fCharge;
TBranch          *b_fPrimaryTracks_fPx;
TBranch          *b_fPrimaryTracks_fPy;
TBranch          *b_fPrimaryTracks_fPz;

script(TTree *tree=0);
virtual ~script();
virtual Int_t    Cut(Long64_t entry);
virtual Int_t    GetEntry(Long64_t entry);
virtual Long64_t LoadTree(Long64_t entry);
virtual void     Init(TTree *tree);
virtual void     Loop();
virtual Bool_t   Notify();
virtual void     Show(Long64_t entry = -1);
};

#endif

#ifdef script_cxx
script::script(TTree *tree)
{
    TChain * chain = new TChain("JetTree","");
    chain->Add("PicoDst_1.root/JetTree");
    chain->Add("PicoDst_2.root/JetTree");
    chain->Add("PicoDst_3.root/JetTree");
    chain->Add("PicoDst_4.root/JetTree");
    chain->Add("PicoDst_5.root/JetTree");
    chain->Add("PicoDst_6.root/JetTree");
    chain->Add("PicoDst_7.root/JetTree");
    chain->Add("PicoDst_8.root/JetTree");
    chain->Add("PicoDst_9.root/JetTree");
    tree = chain;
    Init(tree);
}

script::~~script()
{
    if (!fChain) return;
    delete fChain->GetCurrentFile();
}

```

```

Int_t script::GetEntry(Long64_t entry)
{
    if (!fChain) return 0;
    return fChain->GetEntry(entry);
}
Long64_t script::LoadTree(Long64_t entry)
{
    if (!fChain) return -5;
    Long64_t centry = fChain->LoadTree(entry);
    if (centry < 0) return centry;
    if (!fChain->InheritsFrom(TChain::Class())) return centry;
    TChain *chain = (TChain*)fChain;
    if (chain->GetTreeNumber() != fCurrent) {
        fCurrent = chain->GetTreeNumber();
        Notify();
    }
    return centry;
}

void script::Init(TTree *tree)
{
    if (!tree) return;
    fChain = tree;
    fCurrent = -1;
    fChain->SetMakeClass(1);

    fChain->SetBranchAddress("fEventHeader.fEventId",
        &fEventHeader_fEventId, &b_PicoJetTree_fEventHeader_fEventId);
    fChain->SetBranchAddress("fEventHeader.fRunId",
        &fEventHeader_fRunId, &b_PicoJetTree_fEventHeader_fRunId);
    fChain->SetBranchAddress("fEventHeader.fRefMult",
        &fEventHeader_fRefMult, &b_PicoJetTree_fEventHeader_fRefMult);
    fChain->SetBranchAddress("fEventHeader.fReactionPlaneAngle",
        &fEventHeader_fReactionPlaneAngle,
        &b_PicoJetTree_fEventHeader_fReactionPlaneAngle);
    fChain->SetBranchAddress("fEventHeader.fNOfPrimaryTracks"
        , &fEventHeader_fNOfPrimaryTracks,
        &b_PicoJetTree_fEventHeader_fNOfPrimaryTracks);
    fChain->SetBranchAddress("fPrimaryTracks",
        &fPrimaryTracks_, &b_PicoJetTree_fPrimaryTracks_);
    fChain->SetBranchAddress("fPrimaryTracks.fCharge",
        fPrimaryTracks_fCharge, &b_fPrimaryTracks_fCharge);
    fChain->SetBranchAddress("fPrimaryTracks.fPx",
        fPrimaryTracks_fPx, &b_fPrimaryTracks_fPx);
}

```

```

    fChain->SetBranchAddr("fPrimaryTracks.fPy",
    fPrimaryTracks_fPy, &b_fPrimaryTracks_fPy);
    fChain->SetBranchAddr("fPrimaryTracks.fPz",
    fPrimaryTracks_fPz, &b_fPrimaryTracks_fPz);
    Notify();
}

Bool_t script::Notify()
{
    return kTRUE;
}

void script::Show(Long64_t entry)
{
    if (!fChain) return;
    fChain->Show(entry);
}

Int_t script::Cut(Long64_t entry)
{
    return 1;
}
#endif

```

# Seznam použitých zdrojů

- [1] WONG, C.Y. : Introduction to high-energy heavy-ion collisions. London: World Scientific Publishing Co. Pte. Ltd, 1994. ISBN 9810202636
- [2] Collaboration DIRAC: [online]. CERN. [vid. 5.5.2013]. Dostupné z: <http://dirac.web.cern.ch/DIRAC/>
- [3] GSL-GNU Operating System: [online] [vid. 5.5.2013]. Dostupné z: <http://www.gnu.org/software/gsl/>
- [4] ROOT— A Data Analysis Framework - CERN [online]. CERN. [vid. 5.5.2013]. Dostupné z: <http://root.cern.ch/drupal/>.
- [5] LEDNICKY Richard, Finite-size effect on two-particle production. Physics of Particles and Nuclei, 2009, Vol40, No 3, pp. 597-672, ISSN 1547-4771
- [6] HORŇÁK, Ivan. Korelace částic v experimentu DIRAC: Praha, 2013. Bakalářská práce, ČVUT, Fakulta jaderná a fyzikálně inženýrská, Katedra dozimetrie a aplikace ionizujícího záření