

LogBook Tesi

Kamil Laurent

Monday 29th July, 2024

Settimana del 25 marzo 2024

0.1 argomenti trattati

- Test di una forma funzionale complessa (MAP22g52) su una forma semplice (PV19 o PV17): assicurarsi che la parte di Hard Process (tabelle) sia la stessa usata per generare e fittare i dati, deve cambiare solo la parametrizzazione.
- Se fittiamo i dati generati con la PV19, stiamo al N3LL, ci aspettiamo una parametrizzazione consistente per la PDF (Drell-Yan) e dei parametri casuali per la FF (SIDIS).
- se fittiamo dati generati con PV17, ci aspettiamo che questi pseudo-dati siano molto diversi dai dati reali, ma questo ci permette comunque di vedere se la procedura trova il minimo nello spazio dei parametri. In questo caso dobbiamo usare la parametrizzazione g52 con le tabelle al NLL.
- Closure Test L0: è utile vedere in che modo si comporta la replica 0 (non fluttuata), non ci interessano le altre repliche. Ci interessa comunque vedere come si comporta il minimizzatore se parto da un punto diverso nello spazio dei parametri

0.2 da fare

- capire se ComputeMeanReplica ha senso
- fittare 20 volte la ‘replica 0’ lasciando *Paramfluct = true* nella card. Fare questo usando come minimizer sia ceres che minuit-*j*, sembra che anche con *Paramfluct = true* il minimizer parta sempre dallo stesso punto nello spazio delle fasi. Provare a cambiare il seed.
- implementare un pezzo di codice che confronti i parametri trovati con i fit e quelli usati per generare i dati, qualcosa del tipo:

$$\sum_{i=1}^{21} \frac{|p_i^{fit} - p_i^{true}|}{p_i^{true}}$$

dove p_i è il parametro i -esimo della forma funzionale della TMD

- fare un CT L0 su dati generati da PV19. Usare 20 repliche come sopra e vedere il χ^2 per la bontà del test. Qui non possiamo confrontare i parametri uno per uno, per cui vediamo se le forme delle TMD generate in Report coincidono.
- generare gli pseudo-dati L1 usando lo stesso file .py usato per i dati L0 che tenga in considerazione le incertezze sui dati sperimentali. I punti vengono generati con la formula:

$$d_i^{L1} = (1 + \sum_{q=1}^{N_{norm}} r_q^{norm} \sigma_{q,i}^{norm})(t_i^0 + \sum_{p=1}^{N_{sys}} r_p^{sys} \sigma_{p,i}^{sys} + r_i^{unc} \sigma_i^{unc})$$

dove t_i^0 è il valore centrale teorico (senza incertezza) $r \in [-1, 1]$ sono dei valori generati randomicamente da una distribuzione gaussiana (associati alle varie incertezze, sono dipendenti dal datapoint solo per le incertezze non correlate), σ sono le incertezze, che possono essere correlate (di normalizzazione, norm, o sistematiche, sys) e non correlate, unc. Sui datasets che usiamo per i fit queste 3 forme di incertezza sono così mappate: norm = mult, unc = unc, sys = add.

0.3 risultati

- MAP22 su MAP22 ceres: su 20 repliche, 16 hanno un $\chi^2 \propto O(e^{-14})$, 2 hanno $\chi^2 \propto O(e^{-6})$ e 2 hanno $\chi^2 \propto O(10^{-3})$. Sospetto che il minimizzatore si fermi anche se non ha trovato il minimo, nel caso in cui $\chi^2 \propto O(10^{-3})$ perché è soddisfatta la seguente condizione che si può vedere nel output del fit:

```
CONVERGENCE (Function tolerance reached.
|cost_change|/cost: 2.554821e-06 <= 1.000000e-05)
```

- MAP22 su MAP22 minuit: qui il range di valori del χ^2 è più stretto ma i valori sono più distribuiti. In generale il χ^2 è più alto rispetto al caso in cui si utilizza ceres
- MAP22 su PV19 Ceres: il χ^2 varia ma è sempre di $O(10^{-1})$ in tutte le repliche, qui bisogna vedere i grafici delle TMD per farsi un'idea della bontà dei fit e della parametrizzazione.
- MAP22 su PV19 minuit: nessuna delle 20 repliche è arrivata a convergenza.
- creato il dataset L1 a partire da MAP22, fatto un primo fit (replica 0) a partire da questi dati con la parametrizzazione MAP22g52. In questo caso il χ^2 è peggiore del caso L0 (ci aspettiamo che sia così)

- comparando la distribuzione della $f_{NP}(x, b_T, \zeta)$ con la parametrizzazione PV19 e con la parametrizzazione MAP22, ci aspettiamo che abbiano la stessa forma? Dalla parametrizzazione PV19x otteniamo 9 parametri dai quali dipende la nostra distribuzione:

$$f_{NP}(x, \zeta, b_T) = \left(\frac{1 - \lambda}{1 + g_1(x) \frac{b_T^2}{4}} + \lambda \exp \left(-\frac{g_{1B}(x) b_T^2}{4} \right) \right) \exp \left[-g_2 \log \left(\frac{\zeta}{Q_0^2} \right) \frac{b_T^2}{4} - g_{2B} \log \left(\frac{\zeta}{Q_0^2} \right) \frac{b_T^4}{4} \right]$$

in cui:

$$g_1(x) = \frac{N_1}{x\sigma} \exp \left[-\frac{\ln^2 \left(\frac{x}{\alpha} \right)}{2\sigma^2} \right]$$

$$g_{1B}(x) = \frac{N_{1B}}{x\sigma_B} \exp \left[-\frac{\ln^2 \left(\frac{x}{\alpha_B} \right)}{2\sigma_B^2} \right]$$

La parte non perturbativa della PDF entra nella PDF "totale" in questo modo:

$$\hat{f}_1(x, b_T; \mu, \zeta) = f_{NP}(x, b_T, \zeta) \hat{f}_1(x, b_*(b_T); \mu, \zeta)$$

quest'ultima entra nella sezione d'urto (osservabile fisica) in questo modo:

$$\frac{d\sigma}{dQ dy dq_T} = \frac{8\pi\alpha^2 q_T P}{9Q^3} H(Q, \mu) \sum_q C_q(Q) \int_0^\infty db_T b_T J_0(b_T q_T) x_1 \hat{f}_1^q(x_1, b_T; \mu, \zeta) x_2 \hat{f}_1^{\bar{q}}(x_2, b_T; \mu, \zeta)$$

- la parametrizzazione di MAP22 invece ha una forma funzionale diversa, data da 11 parametri (qui inserisco solo la parametrizzazione della f_{NP} perché fittando dati ottenuti da Drell-Yan non stiamo fittando le fragmentation functions):

$$f_{NP}(x, \zeta, b_T) = \exp(S_{NP}(\zeta, b_T))$$

$$\frac{g_1(x) \exp(-g_1(x) \frac{b_T^2}{4}) + \lambda^2 g_{1B}^2(x) (1 - g_{1B}(x) \frac{b_T^2}{4}) \exp(-g_{1B}(x) \frac{b_T^2}{4}) + \lambda^2 g_{1C}(x) \exp(-g_{1C}(x) \frac{b_T^2}{4})}{g_1(x) + \lambda^2 g_{1B}^2(x) + \lambda^2 g_{1C}(x)}$$

dove:

$$S_{NP} = \exp \left[-g_2^2 \frac{b_T^2}{4} \log \left(\frac{\zeta}{Q_0^2} \right) \right]$$

$$g_{1,1B,1C}(x) = N_{1,1B,1C} \frac{x^{\sigma_{1,2,3}} (1-x)^{\alpha_{1,2,3}^2}}{\hat{x}^{\sigma_{1,2,3}} (1-\hat{x})^{\alpha_{1,2,3}^2}}$$

$$f_{\text{NP}}(x, \zeta, b_T) = \frac{N_1}{x\sigma_1} \left(\frac{x^{2\sigma_1}(1-x)2\alpha_1}{0.12\sigma_1(1-0.1)^{2\alpha_1}} \right)$$

Sia in MAP22g52 che in PV19 abbiamo che $Q_0 = 1\text{GeV}$, in MAP22 abbiamo $\hat{x} = 0.1$

Questa entra nella PDF "totale" in questo modo:

0.4 domande

- cosa fa esattamente l'opzione *Paramfluct* nella card di configurazione?
- piccole variazioni nella formula per la generazione degli pseudodati L1 (ad esempio ci sono più incertezze non correlate)
- al L1 vogliamo testare cose diverse. Con il test L0 ci siamo accorti che *ceres* funziona meglio. Continuiamo con *ceres* abbandonando *minuit*? Questa volta se faccio un fit su 50 repliche, queste vanno fluttuate o cambiamo solo il punto di partenza nello spazio dei paramtri? Per ora sto facendo 50 repliche tutte fluttuando la replica 0.
- pensavo che in qesto caso sarebbe utile testare i sisultati con e senza la t_0 prescription, è corretto?
- serve ancora fare un fit sugli pseudodati generati con PV19x o abbiamo già testato che la nostra parametrizzazione funziona su dati più semplici? -¿ serve ancora.

1 Settimana del 8 aprile 2024

1.1 argomenti trattati

- **Parametrizzazioni PV19 e MAP22:** quando facciamo un fit usando la parametrizzazione MAP22 su dati generati da PV19, dovremmo ottenere la stessa forma funzionale della parte non perturbativa della TMD. Questo perché nulla cambia nella relazione tra la f_{NP} e la sezione d'urto nei due articoli PV19 e MAP22. L'unica cosa che cambia è la parametrizzazione della f_{NP} .
- **generazione degli pseudo-dati L1:** c'era una imprecisione nel modo in cui venivano pesati gli errori dei datapoint per la generazione degli pseudo-dati. In particolare nei fattori r_q , r_p , r_i , i quali devono essere generati randomicamente da una distribuzione normale $N(0,1)$. Questi non devono per forza essere 'pinzati' nell'intervallo $[-1,1]$. Generare di nuovo gli pseudo-dati con la distribuzione corretta.

- **fit degli pseudo-dati L1:** Per il fit L1, anche in questo caso le repliche non devono fluttuare i dati, ma devono partire da punti diversi nello spazio dei parametri. Ci aspettiamo che la distribuzione del χ^2 sia intorno a 1. Qui vorremmo testare le seguenti cose:

- parametrizzazione MAP22 su pseudodati generati da PV19
- MAP22 su dati generati da MAP22 (procedura standard)
- MAP22 su MAP22 con la t_0 prescription
- altro?

1.2 da fare

- Fare più repliche del fit con dati non fluttuati (50-100 invece di 20)
- provare con queste 100 repliche a rifare il fit e fare un plot di tutti i risultati che includa:
 - la parametrizzazione con cui sono stati generati i dati (distribuzione **vera**)
 - le 100 distribuzioni ottenute con il fit.
- capire perché il χ^2 al L1 viene intorno a 2.45, a questo proposito farò 100 repliche con i dati che ho generato con un apposito codice e 100 repliche in cui uso la replica 9 (che in teoria dovrebbe essere fluttuata in qualche modo su cui non ho il controllo) cambiando in ogni replica il punto di partenza nello spazio dei parametri.

1.3 risultati

- I nuovi dati generati a una distribuzione normale $N(0, 1)$ danno un $\chi^2 \cong 2.5$, che è molto alto rispetto a quello che ci aspettiamo. Potrebbe esserci un errore in come il codice per la generazione degli pseudo-dati è stato implementato oppure nella distribuzione utilizzata
- ho fatto un test usando 100 volte la replica 9, qui mi è venuta una *error function* = $E \cong 1$, che è quello che ci aspettiamo. il χ^2 è molto piccolo e diverso dalla E perché probabilmente la E viene calcolata sui dati fluttuati, mentre il χ^2 è calcolato sui dati non fluttuati.

2 Settimana del 15 aprile 2024

2.1 argomenti trattati

- Discusso degli argomenti della settimana scorsa

- g52 su PV19: troviamo 2 classi di funzioni, una vicina alla distribuzione vera, una abbastanza lontana. Capire se quella più vicina da un buon risultato.
- test L1: il test sembra funzionare se uso dati fluttuati da NangaParbat, non funziona se uso un codice a parte per aggiungere delle fluttuazioni sui dati usati in L0, capire perché

2.2 da fare

- g52 su PV19: implementare nel codice che genera il fit una parte che calcoli la distanza di una distribuzione dalla distribuzione vera
- closure test L1: forse quello che sbaglio è definire dei nuovi pesi per gli errori moltiplicativi e sistematici ad ogni sotto-dataset dell'esperimento e non a tutto l'esperimento. Ad esempio prendiamo l'esperimento E605: i dataset di questo esperimento sono divisi in 5 file, che corrispondono a dati presi per diversi valori di Q. Al momento il codice definisce dei pesi r_{norm} e r_{sys} ad ogni nuovo file, provo a definirli una volta sola per tutto l'esperimento E605. NO
- qui: <https://arxiv.org/pdf/0808.1231.pdf> (formule 13 e 14) viene spiegato come generare le repliche Monte Carlo per un certo dataset. Questo è il modo in cui vengono generate le repliche in Nanga Parbat. In questo paper viene specificato che anche la matrice di covarianza viene cambiata (le incertezze vengono riscalate), questo è analogo a fare il fit con la t_0 prescription e le incertezze non riscalate?

2.3 Risultati

- Fluttuare allo stesso modo tutti i dati per un certo esperimento non ha prodotto risultati buoni, il χ^2 rimane molto alto
- fare un fit L1 con la t_0 prescription attiva
- g52 su PV19 al L0: ho implementato una condizione sulla distanza media in quadratura di una certa distribuzione ottenuta dal fit dalla distribuzione vera. Chiamo Mean Squared Distance la distanza:

$$MSD = \frac{1}{N_{points}} \sum_{i=1}^{N_{points}} (f_{NP,i}^{pred} - f_{NP,i}^{real})^2$$

Ottengo:

- $MSD < 0.0001$ per 0 repliche (0%)
- $MSD < 0.0005$ per 47 repliche (47%)
- $MSD < 0.01$ per 52 repliche (52%)

3 Settimana del 22 aprile 2024

3.1 Argomenti trattati

- MAP22g52 su PV19: test al L0 quasi concluso, troviamo due classi di funzioni, una delle quali approssima bene la distribuzione vera, una no. Circa nel 50% dei casi troviamo una distribuzione che approssima quella vera. Prossimi passi:
 - verificare cosa succede al L1
 - esplorare il comportamento di una parametrizzazione più libera, come una rete neurale. In questo caso mi aspetto che la distribuzione vera sia riprodotta correttamente in più repliche rispetto al caso della parametrizzazione MAP22g52.
 - si può pensare di utilizzare la parametrizzazione MAP24
- La formula per la generazione degli pseudo-dati L1 scritta in precedenza è sbagliata. Dobbiamo tenere in considerazione due cose:
 - ci sono due tipo di incertezze nei datasets (assolute e relative). Per prima cosa dobbiamo convertire tutte le incertezze in assolute
 - le incertezze moltiplicative vanno trattate come tali (vedi formula aggiornata sotto)

La formula corretta da usare è la seguente:

$$d_i^{L1} = \prod_{q=1}^{N_{norm}} (1 + r_q^{norm} \sigma_{q,i}^{norm}) (t_i^0 + \sum_{p=1}^{N_{sys}} r_p^{sys} \sigma_{p,i}^{sys} + r_i^{unc} \sigma_i^{unc})$$

- Provare a fare un fit con la parametrizzazione PV19 su dati MAP22 come esercizio. Mi aspetto che il closure test non chiuda o che chiuda peggio di quanto non faccia con il setup opposto.
- incertezze sperimentali: nei file di dati ci sono 3 tipi di incertezze, *unc*, *mult* e *add*, le ultime due sono relative e sono da convertire in incertezze assolute.

3.2 risultati

3.3 domande

4 Settimana del 29 aprile 2024

4.1 Argomenti trattati

- **parametrizzazione PV19 su MAP22:** la parametrizzazione sembra funzionare molto bene, anche se abbiamo una discrepanza elevata in condizioni particolari. Ad esempio quando scegliamo di fare un plot della

TMD a $Q = 1\text{GeV}$ e $x = 0.1$, la parte non perturbativa ha un comportamento non fisico per grandi b_T (dovrebbe andare a 0 ma non lo fa). Per contro il fit riproduce molto bene i nostri dati sperimentali, per cui la parametrizzazione funziona molto bene nelle regioni cinematiche in cui abbiamo preso i nostri dati sperimentali.

- **fit in diverse condizioni:** Visto che il fit funziona molto bene ad energie dell'ordine di grandezza di 10GeV , provare a plottare le TMD della distribuzione vera e di quelle ottenute sia nel caso MAP22oPV19 che nel caso PV19oMAP22. Vedere nella regione cinematica considerata quale fit chiude meglio (contando che rispetto a MAP22oMAP22 nessuno dei due chiude al L0)
- **tabelle e testo:** fare delle tabelle con i risultati ottenuti fino ad ora e accompagnarle da alcuni paragrafi per spiegare il tutto. Cosa inserire:
 - χ^2 di: $L0MAP22oMAP22$, $L0PV19oMAP22$ e $L0MAP22oPV19$.
 - MSD (= Mean Squared Distance) dei plot delle TMD a $x = 0.1$, $Q = 10\text{GeV}$ di $L0PV19oMAP22$ e $L0MAP22oPV19$
- **Dati per l'esecuzione dei fit:** La differenza principale tra le parametrizzazioni MAP22g52 e PV19 sta nel fatto che MAP22 è un fit globale fatto su esperimenti di DY e SIDIS. Per cui la forma funzionale della TMD può essere usata anche per riprodurre dati sperimentali di SIDIS, cosa che non può essere fatta con la TMD di PV19. Dovremmo provare a riprodurre anche dati L0 di tipo SIDIS partendo dalla TMD con la parametrizzazione PV19 e una FF nota che mettiamo noi dall'esterno, e vedere in che modo si comporta MAP22g52 su questo dataset.
- **L1:** ancora dei problemi sulla generazione dei dataset L1, capire cosa non va.

4.2 risultati

- **dati L1:** La formula per l'implementazione di dati L1 viene presa dal modo in cui Nanga Parbat genera le repliche e usata su un dataset L0 (perfetto-non fluttuato). L'implementazione è la seguente:

$$t_{L1}^i = t_0^1 \prod_{j=1}^{N_{norm}} \sqrt{(1 + r_j^{mult} \sigma_{i,j}^{mult})} (1 + \sum_{j=1}^{N_{norm}} r_j^{add} \sigma_{i,j}^{add} + r_j^{unc} \sigma_i^{unc})$$

dove $\sigma_i^{unc} = \sqrt{\sum_{j=1}^{N_{unc}} \sigma_{i,j}^2}$. Gli errori in questa formula sono relativi, per cui convertiamo l'incertezza non correlata.

Lo faccio in due modi diversi: fluttuando le predizioni ottenute al livello 0, oppure partendo dai dati sperimentali, generando le predizioni direttamente da li. L'unica differenza sarà nel modo in cui le incertezze non

correlate vengono convertite in incertezze relative (nel primo caso σ_i^{unc}/t_0 , nel secondo caso σ_i^{unc}/t_{exp})

- **Nuovo Dataset:**

4.3 domande

5 Settimana del 27 maggio 2024

5.1 argomenti discussi

- confronto tra pseudodati generati con Nanga Parbat e generati con un codice a parte:

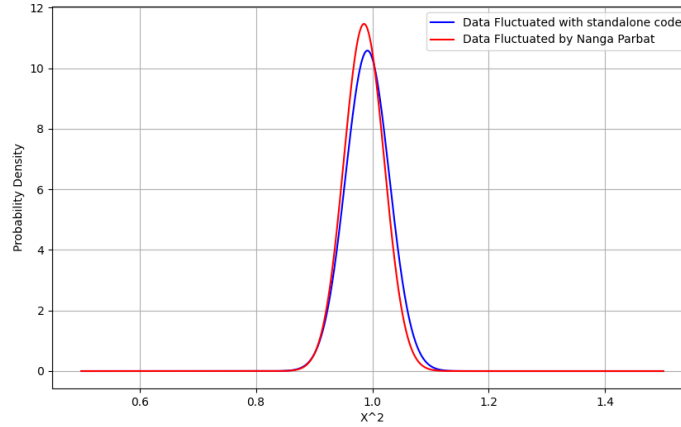


Figure 1: distribuzione χ^2

Credo che questa differenza (molto piccola) sia dovuta ai dati di Drell-Yan, alcuni dei quali hanno errori e valori centrali molto piccoli. In particolare credo che la causa sia il modo in cui python e C++ trattano i numeri molto piccoli. Infatti dobbiamo togliere dei datasets (Phoenix, D0 Run II e Run II μ , Atlas a 8, 13 TeV , LHCb 7 e 13 TeV) quando usiamo python per effettuare la Cholesky decomposition, altrimenti si lamenta che la matrice di covarianza per questi datasets non è definita positiva.

- L1 tests: ecco una tabella dei test al livello 1:

Table 1: L1 results

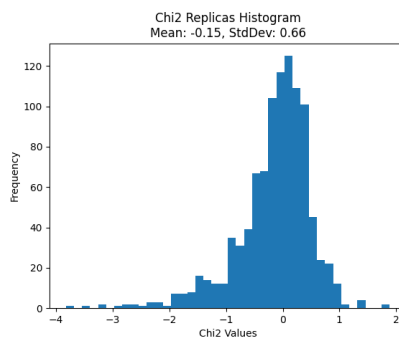
name	fit parametrization:	pseudo-data generated with:	mean χ^2
g52oMAP22	MAP22g52	MAP22g52	da fare
PV19oPV19	PV19x	PV19x	0.9947 ± 0.0002
g52oPV19	MAP22g52	PV19x	1.3203 ± 0.246
PV19oMAP22	PV19x	MAP22g52	0.8424 ± 0.0013

6 Settimana del 3 giugno 2024

6.1 argomenti discussi

- Test della procedura di generazione dei dati al L1: Ho provato a generare 100 repliche dei dati Drell-Yan L1, ogni volta fluttuati in modo diverso. Cambiando solamente il modo in cui vengono fluttuati i dati e nient'altro otteniamo risultati molto diversi. Gli istogrammi sotto riportano il χ^2 della replica 0 su 100 set di dati fluttuati in modo diverso (casuale). I test fatti fino ad ora sono:

- L1 con il metodo di fluttuazione "tradizionale" (python):

Figure 2: $\chi^2 = 0.8333 \pm 0.0727$

- L1, Cholesky decomposition, usando Numpy (i dataset che davano problemi sono stati fluttuati usando il metodo 'tradizionale')

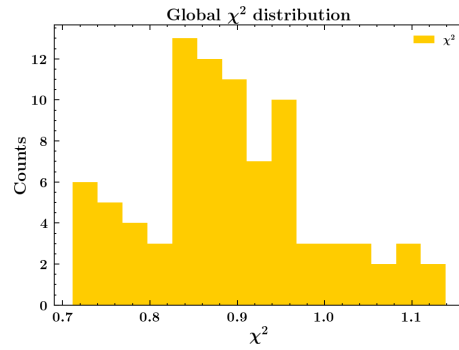


Figure 3: $\chi^2 = 0.8935 \pm 0.0992$

- L1, Cholesky decomposition, usando Scipy (i dataset che davano problemi sono stati fluttuati usando il metodo 'tradizionale')

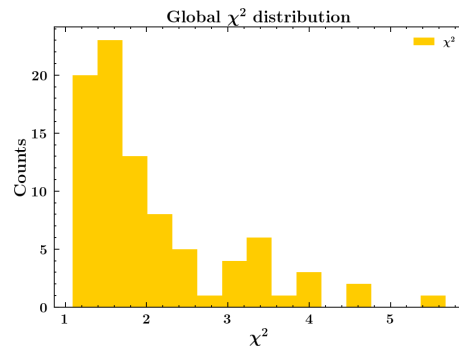


Figure 4: $\chi^2 = 2.0617 \pm 0.9322$

- L1, Cholesky decomposition, usando Eigen (C++) (nessun dataset ha dato problemi)

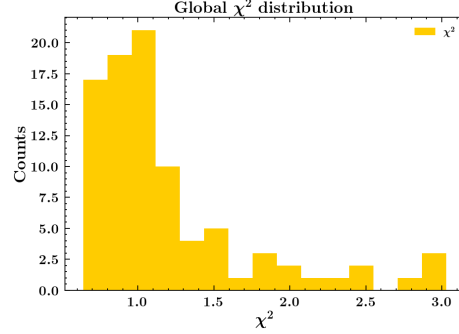


Figure 5: $\chi^2 = 1.1871 \pm 0.5471$

- L0 sulla parametrizzazione mista: ho generato delle predizioni da un fit sui dati usando le PDF di PV19 e le FF di MAP22. Dopodiché ho provato a ottenere nuovamente la distribuzione MIX24 sia con MAP22 che con PV19. Qui ci sono i risultati dei test su MIX24

Table 2: L0 su MIX24

name	fit parametrization:	pseudo-data generated with:	mean χ^2
$L0_{g52oMIX24}$	MAP22g52	MIX24	0.1546 ± 0.0049
$L0_{PV19oMIX24}$	PV19x	MIX24	10^{-7}
$L0_{MIX24oMIX24}$	MIX24	MIX24	0.0054 ± 0.0045

- Test L1 su due repliche di pseudo-dati diversi: ottengo due distribuzioni piccate su valori centrali diversi, a seconda del tipo di distribuzione che prendo

6.2 Da fare

- **check della fluttuazione dei dati L1:** per assicurarsi che i dati L1 siano fluttuati correttamente, facciamo il seguente test:

- prendo i dati L0 e li fluttuo 100 volte (con il metodo che preferisco). Ottengo 100 repliche di dati fluttuati L1. Ad esempio consideriamo un singolo dataset (E605) al livello 0, l'i-esimo punto sarà t_i^0 . Ogni punto avrà adesso 100 repliche associate $t_{i,j}^1$ con $j = 1, 2, \dots, 100$
- per ogni dato sperimentale di queste 100 repliche calcolo il valore medio 'orizzontalmente' su queste 100 repliche: $\langle t_i^1 \rangle = \sum_{j=1}^{100} \frac{t_{i,j}^1}{100}$. Se il codice è giusto devo avere:

$$t_i^0 = \langle t_i^1 \rangle$$

- posso calcolare anche le incertezze correlate e non correlate (costruendo la matrice di covarianza dei dati?), ma non ho capito bene come fare

- **L0–MIX24oMAP22:** verificare che cosa succede se faccio un fit L0 con la parametrizzazione MIX24 (che ha le PDF di PV19 e le FF di MAP22g52) sui dati di MAP22.

6.3 Risultati

- **Test sulla generazione degli pseudodati:** Per prima cosa testiamo i vari metodi con una condizione sulle discrepanze dei valori medi:

$$\delta_i = \frac{|t_i^0 - \langle t_i^1 \rangle|}{t_i^0}$$

Ho verificato che anche i punti con una discrepanza piuttosto alta, se

Table 3: numero di punti non convergenti su 100 repliche generate con differenti metodi

method	number of $\delta > 0.01$	$\delta > 0.02$	$\delta > 0.05$
traditional	122	71	10
Cholesky, numpy	139	81	14
Cholesky, scipy	122	74	16
Cholesky, eigen	106	69	9

aumento il numero di repliche drasticamente, torno alla media "vera"
Per quanto riguarda la deviazione standard, ottengo una σ_i dalla distribuzione dei dati fluttuati e la confronto con

$$\sigma_i = \sqrt{\sigma_{unc}^2 + (t_i^0 \sigma_{add})^2 + (t_i^0 \sigma_{mult})^2}$$

Facendo lo stesso test fatto per la media ottengo molti punti (decine o in alcuni casi centinaia) con una σ piuttosto diversa da quella di partenza ($\delta_\sigma > 0.1$)

- **test χ^2 sulle fluttuazioni:** posso verificare che le fluttuazioni siano costruite in modo perfetto (Gaussiano con le corrette σ) calcolando il χ^2 tra la distribuzione dei dati fluttuati (L1) e la distribuzioni dei dati veri.

$$\langle \chi^2 \rangle = \frac{1}{N_{dat}} \sum_{i=1}^{N_{dat}} \frac{(t_i^0 - t_i^1)^2}{\sigma_i^2}$$

Da questo test si vede chiaramente come alcuni dataset danno dei problemi. In particolare i dataset problematici sono: D0, PHENIX, LHCb. Comunque vengano fluttuati, questi danno una distribuzione del χ^2 strana (vedi , dando origine anche a valori negativi. Questi valori negativi e le distribuzioni strane sono dovute molto probabilmente alle matrici di covarianza che secondo python non sono definite positive.

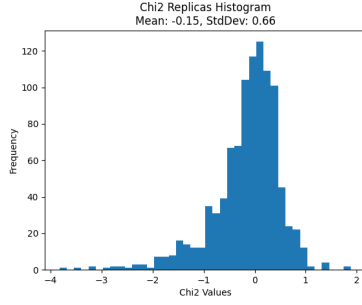


Figure 6: χ^2 distribution of D0 replicas

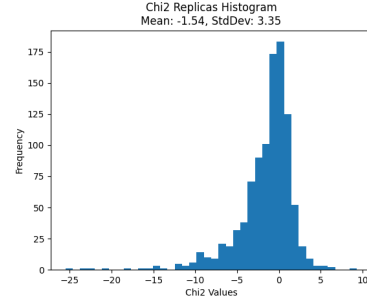


Figure 7: χ^2 distribution of LHCb replicas

Figure 8: some nonsense χ^2 distribution

Ecco un confronto tra la distribuzione del χ^2 medio su 1000 repliche con e senza i dataset problematici:

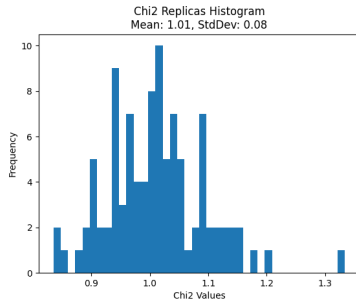


Figure 9: χ^2 distribution excluding D0, PHENIX, LHCb

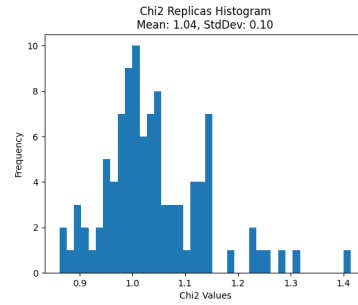


Figure 10: χ^2 of the complete dataset with prescriptions

Figure 11: χ^2 distribution tests

Propongo questa soluzione: genero insieme le fluttuazioni di tutti i dati che non danno problemi, fatto questo genero le fluttuazioni dei datasets problematici in maniera controllata, implementando delle condizioni sul χ^2 medio

7 Settimana del 17 giugno 2024

7.1 Discussioni della settimana precedente:

- **Generazione dati L1:** la questione è delicata perché non siamo ancora sicuri che tutti i punti siano fluttuati correttamente (senza errori

numerici).

- Come gestiamo quei punti che essendo fluttuati assumono dei valori negativi? N.B. Nanga Parbat non ha un modo per gestire questi punti. Un buon modo per eliminare questo problema può essere questo: ogni volta che si presenta un numero negativo, rifacciamo tutta la fluttuazione del dataset, fino a che tutti i datapoint non diventano positivi

7.2 risultati

- **Test L1 su 6 repliche:** prendiamo 6 repliche di dati fluttuati al L1 (usando Eigen, C++). Per ognuna di queste repliche faccio 50 fit senza fluttuare ulteriormente i dati, partendo da 50 punti diversi dello spazio dei parametri. A priori conosciamo il χ^2 dei dati fluttuati rispetto ai dati "reali" ($\langle \chi^2 \rangle_{L0-L1}$), mentre il fit ci dice il χ^2 della distribuzione ottenuta rispetto ai dati fluttuati ($\langle \chi^2 \rangle_{L1-fit}$). La tabella sotto riassume il valore di questi due stimatori statistici per le 6 repliche

Table 4: L1 fluctuations with Eigen

replica	$\langle \chi^2 \rangle_{L0-L1}$	$\langle \chi^2 \rangle_{L1-fit}$
0	0.921024	0.5743
1	1.03024	0.8257
2	0.965785	0.7304
3	0.980609	0.7449
4	0.744909	0.5027
5	1.20093	1.0048

- **stesso test fatto per g52oMAP22** In questo caso specifico abbiamo un risultato più incoraggiante

7.3 da fare:

- **Repliche dei datasets al L1:** abbiamo adesso 6 repliche di dati fluttuati in modo diverso (riproducibile con un seed). Concludiamo il fit al L1 su ciascuna di queste 6 repliche.
- **test L2 g52oMAP22:** scegliere una delle 6 repliche fluttuate al L1 e fare un test L2

- **Incertezze su g52oMAP22** quando abbiamo tutti questi fit, facciamo un plot delle incertezze delle TMD ottenute:
 - incertzze di interpolazione
 - incertezze funzionali
 - incertezze sperimentali

8 Settimana del 25 giugno 2024

8.1 Argomenti discussi

- **test L0:** abbiamo concluso i test al livello 0 e gli unici test che riusciamo a portare a chiusura sono g52oMAP22 e PV19oPV19. Questo significa che le nostre parametrizzazioni funzionano bene nel caso in cui la legge utilizzata sia molto simile alla legge vera, il che è poco probabile.
- **Caratterizzazione delle incertezze:** ci sono diversi modi per generare dei plot che tengano in considerazione le incertezze, il modo più semplice è quello di plottare punto per punto le bande d'errore usando media e deviazione standard del vettore che riempio con i valori delle f_{NP} , ma in questo modo ottengo una banda d'errore piccola e non riesco a distinguere con chiarezza le incertezze del L1 e del L2.
Un secondo modo è quello di plottare i rapporti:

$$\frac{f_{NP,true}}{f_{NP,pred}}$$

avendo così una media e una deviazione standard del vettore contenente questi rapporti.

- **componenti delle incertezze:** possiamo fare il plot di media e deviazione standard nei 3 casi (L0, L1, L2) e confrontare i livelli di incertezze di interpolazione/estrapolazione, funzionali e sperimentali facendo i rapporti di queste incertezze. In questo modo possiamo accorgerci dei contributi di ogni tipo di incertezza all'incertezza totale.
- **migliore visualizzazione delle incertezze:** Provare a ripetere lo stesso plot a scale di energie più basse ($Q = 1GeV$) e basso x ($x < 0.1$). Ci aspettiamo che l'incertezza cresca e si possa vedere più chiaramente.
- **Metà tesi:** la probabile data della metà tesi è il 22 luglio, commissario esterno: Miguel Onorato, contro-relatore: Simon Badger

8.2 Da fare

- **test L1:** completare il test L1 per g520MAP22 con 250 repliche

- **TMD PDF e FF** scrivere il codice che plotta anche la parte di FF ottenuta da g52oMAP22
- **Vetori che contengono i risultati:** a livello di codice è conveniente avere due vettori contenenti i valori delle TMDs a fissato valore di b_T , in modo da poter plottare con facilità:
 - media e deviazione standard
 - media e deviazione standard dei rapporti
 - rapporti tra incertezze $\frac{\sigma_{L0}}{\sigma_{L2}}$ e $\frac{\sigma_{L1}}{\sigma_{L2}}$

9 Settimana del 25 giugno 2024

9.1 Risultati Ottenuti

- plot delle TMD PDFs e FFs di MAP22g52 per diversi valori di Q e x e z . I valori di x e Q utilizzati in MAP22 sono i seguenti:
 - PDFs: $Q = 10GeV$ e $2GeV$; $x = 0.001, 0.01$ e 0.1
 - FFs: $Q = 10GeV$ e $2GeV$; $z = 0.3$ e 0.6

Plottando i rapporti a queste scale, usando sempre $b_T \in (0, 10)$ otteniamo i seguenti risultati per la scala $Q = 10GeV$

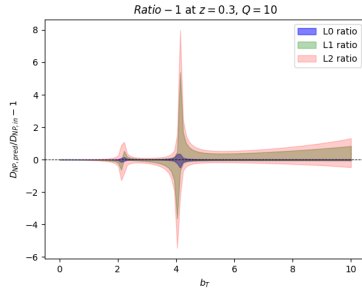


Figure 12: $z = 0.3$

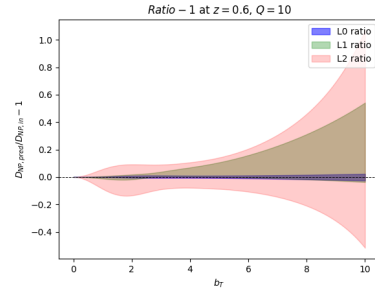


Figure 13: $z = 0.6$

Figure 14: $\frac{D_{NP,pred}}{D_{NP,in}} - 1$ at $Q = 10GeV$

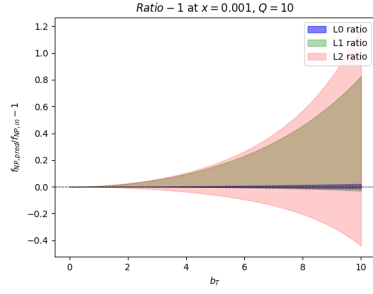


Figure 15: $x = 0.001$

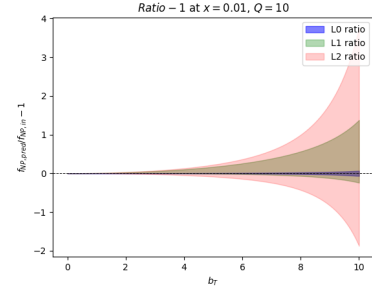


Figure 16: $x = 0.01$

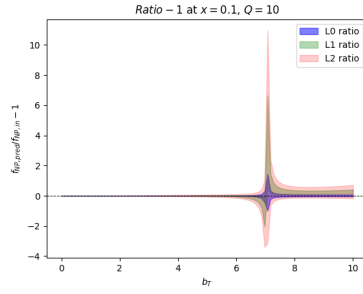


Figure 17: $x = 0.1$

Figure 18: $\frac{f_{NP,pred}}{f_{NP,in}} - 1$ at $Q = 10 GeV$

I plot seguenti rappresentano gli stessi risultati a $Q = 2 GeV$

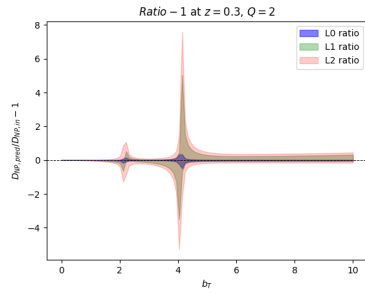


Figure 19: $z = 0.3$

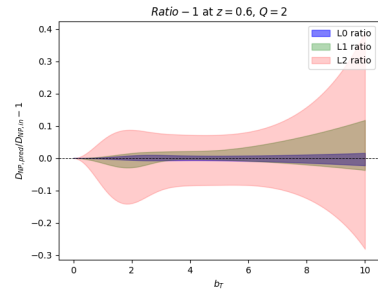


Figure 20: $z = 0.6$

Figure 21: $\frac{D_{NP,pred}}{D_{NP,in}} - 1$ at $Q = 2 GeV$

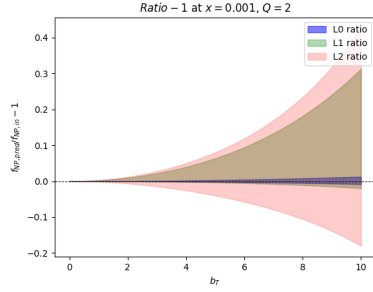


Figure 22: $x = 0.001$

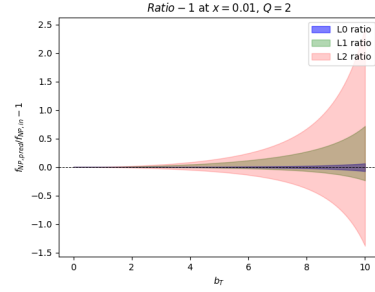


Figure 23: $x = 0.01$

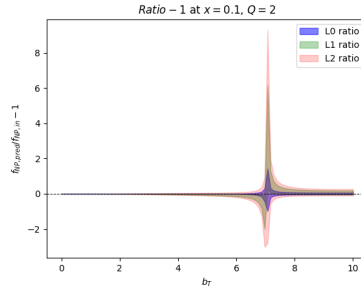


Figure 24: $x = 0.1$

Figure 25: $\frac{f_{NP,pred}}{f_{NP,in}} - 1$ at $Q = 2\text{GeV}$

- **Rapporto tra le incertezze:** Possiamo anche studiare il rapporto tra le incertezze data dai fit sui dati al L0, L1 e L2 in ogni punto di b_T considerato per avere un andamento che rappresenti la percentuale di ogni incertezza rispetto all'incertezza totale.

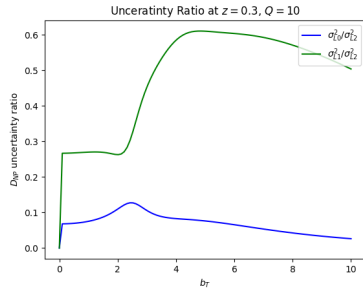


Figure 26: $z = 0.3$

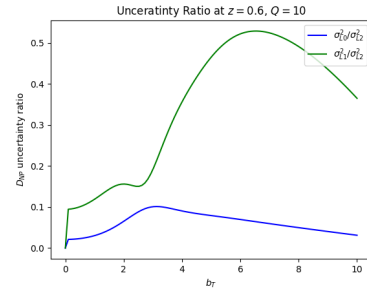


Figure 27: $z = 0.6$

Figure 28: $D_{NP,pred}$ uncertainty ratios at $Q = 10\text{GeV}$

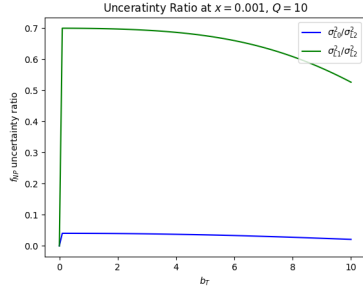


Figure 29: $x = 0.001$

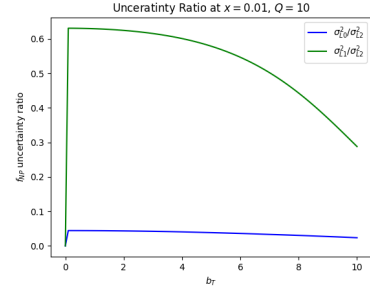


Figure 30: $x = 0.01$

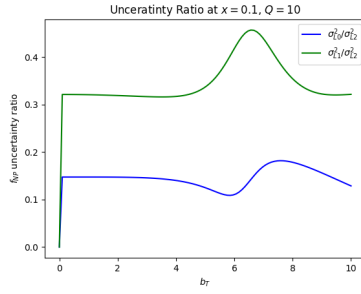


Figure 31: $x = 0.1$

Figure 32: $f_{NP,pred}$ uncertainty ratios at $Q = 10GeV$

Seguono gli stessi plot a $Q = 2GeV$

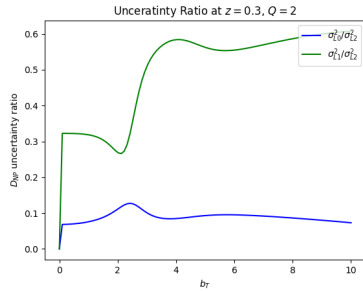


Figure 33: $z = 0.3$

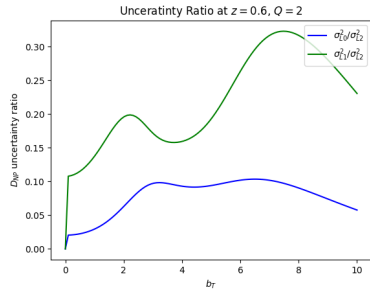


Figure 34: $z = 0.6$

Figure 35: $D_{NP,pred}$ uncertainty ratios at $Q = 10GeV$

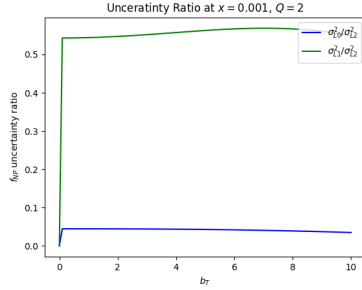


Figure 36: $x = 0.001$

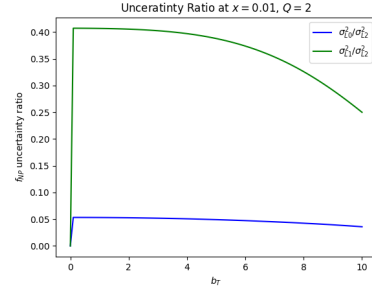


Figure 37: $x = 0.01$

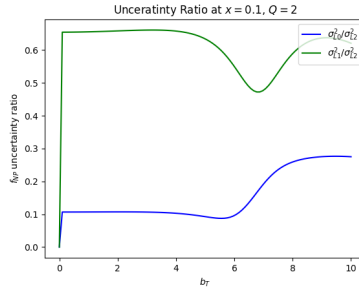


Figure 38: $x = 0.1$

Figure 39: $f_{NP,pred}$ uncertainty ratios at $Q = 10 GeV$

10 Domande

- **diagrammi tagliati:** Ho ancora dei problemi sull'interpretazione di certi diagrammi tagliati che compaiono nella descrizione delle sezioni d'urto
- **errori con/senza taglio degli outliers:** le barre di errore che si vedono nei plot sono ricavate senza escludere nessuna replica (escludiamo solo le repliche non convergenti). Potremmo fare un test che risponde alla domanda "cosa cambia nelle barre d'errore se escludo gli outliers?"
- **scala logaritmica:** può avere senso usare la scala logaritmica in y nei casi in cui l'incertezza diventa molto alta? Oppure (come nell'articolo NNPDF) conviene troncare il plot a certi valori di y ?
- **incertezza totale:** l'incertezza totale su una certa distribuzione corrisponde all'incertezza sperimentale (L2)?