

Relazione 3

La terza relazione si focalizza su autovalori, valori singolari e problema dei minimi quadrati per approssimazione di dati.

Il primo esercizio richiama un primitivo algoritmo utilizzato dal motore di ricerca Google per indicizzare le pagine web in modo da mostrare i risultati più rilevanti. L'algoritmo utilizza gli **autovalori** ed i relativi **autovettori** della matrice che descrive i *link* che collegano le varie pagine per avere una stima della rilevanza di una certa pagina. L'idea nasce dal fatto che gli autovalori sono informazioni che descrivono bene il contenuto di una matrice.

L'informazione migliore che si possa ricavare è quella data dall'autovalore **più grande** in modulo e dall'autovettore associato. Il calcolo di queste due entità è stato effettuato implementando l' *algoritmo delle potenze*, ed i risultati sono i seguenti:

Figura 2:

- autovalore massimo: 1.9497
- peso pagina 1: 0.2887
- peso pagina 2: 0.1481
- peso pagina 3: 0.3390
- peso pagina 4: 0.2240

Figura 3:

- autovalore massimo: 1.6593
- peso pagina 1: 0.2914
- peso pagina 2: 0.2664
- peso pagina 3: 0.2248
- peso pagina 4: 0.0816
- peso pagina 5: 0.1355

I risultati sul peso delle pagine sono stati ottenuti normalizzando tutti i valori dell'autovettore associato al valore massimo. I risultati sono confermati dalla funzione MATLAB `eig`.

Il secondo esercizio richiedeva di utilizzare la fattorizzazione **SVD** per approssimare un'immagine. Ciò è possibile grazie alla natura della fattorizzazione e dei valori singolare: la loro costruzione infatti permette di ottenere una buona approssimazione di una matrice utilizzando molto meno spazio. Un'immagine non è altro che una matrice (tre, nel caso delle immagini a colori) in cui sono memorizzati i valori del codice relativo al colore da visualizzare nel pixel che corrisponde a quell'elemento. Come nel caso degli autovalori, le approssimazioni migliori si ottengono utilizzando i **valori singolari** più grandi. Le immagini che seguono evidenziano questo fenomeno.

Come è evidente, l'approssimazione data dal valore singolare *maggior* offre già un'anteprima di quella che potrebbe essere l'immagine originale, mentre quella

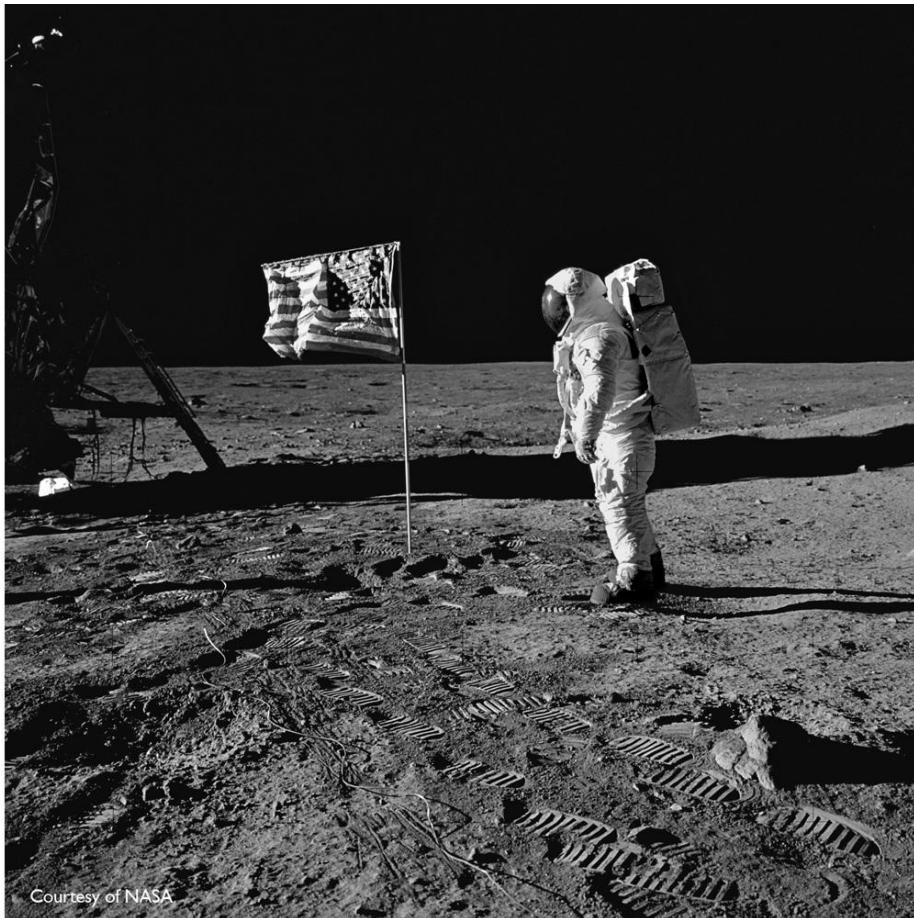


Figure 1: Immagine originale

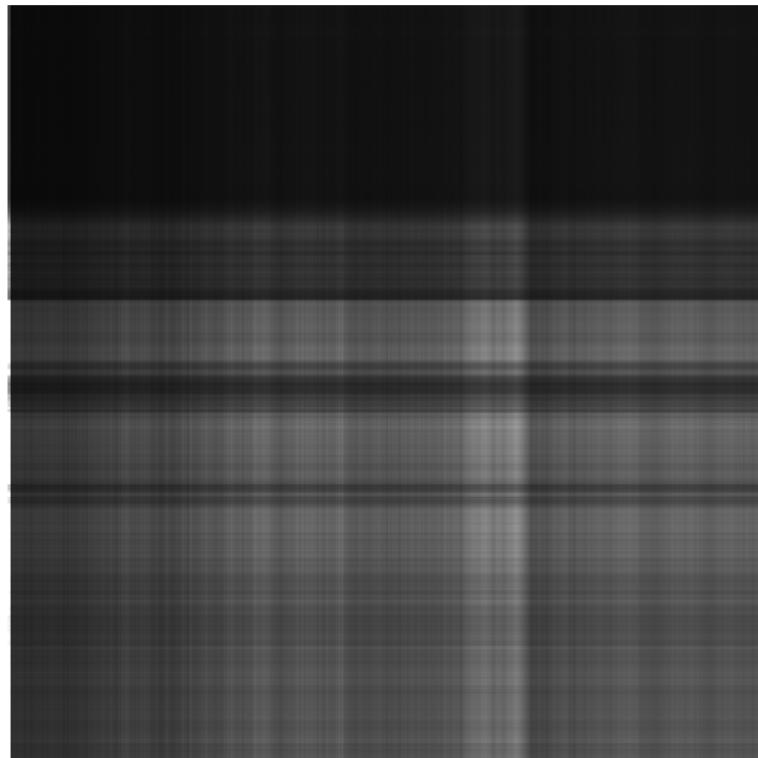


Figure 2: Immagine approssimata dalla diade relativa al valore singolare maggiore

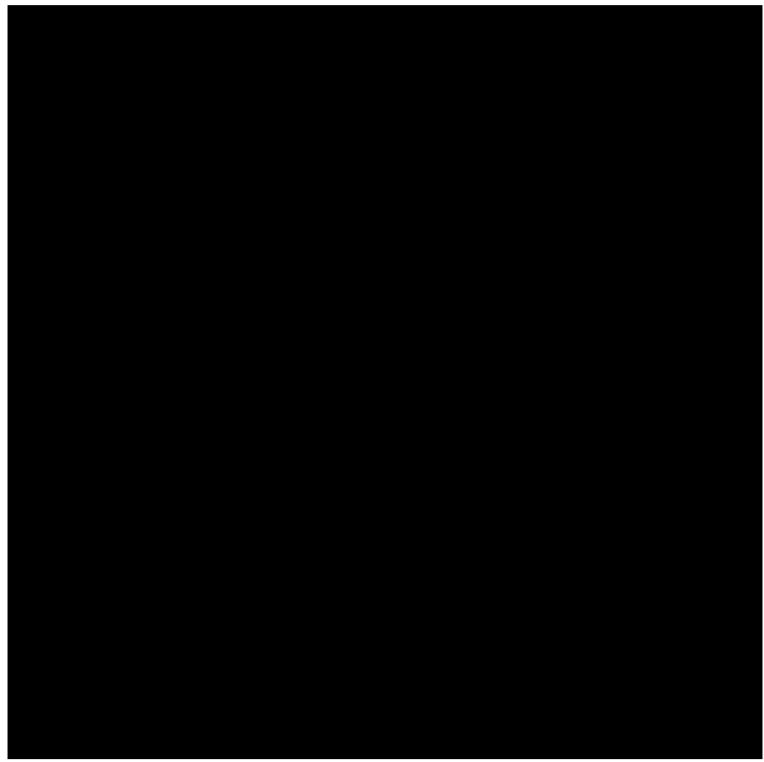


Figure 3: Immagine approssimata dalla diade relativa al valore singolare minore



Figure 4: Immagine approssimata usando le prime 20 diadi

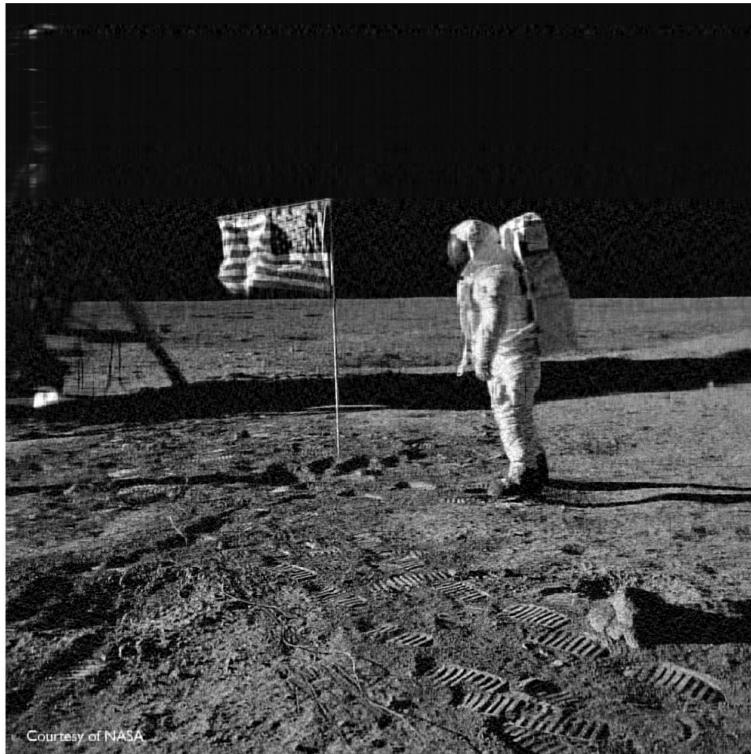


Figure 5: Immagine approssimata usando le prime 100 diadi

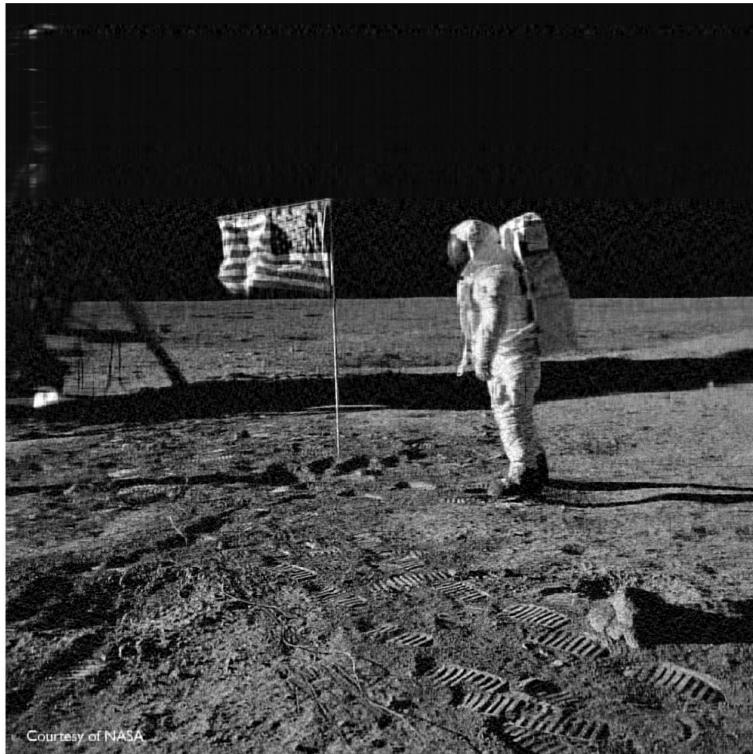


Figure 6: Immagine approssimata usando un quarto delle diadi

ottenuta dal valore singolare *minore* è completamente nera. Usando solamente le prime 20 diadi si ha già una compressione dell'immagine che permette comunque di distinguere nettamente il contenuto, mentre usando le diadi associate ai valori singolari più piccoli forniscono un'approssimazione poco precisa, anche utilizzandone 100.

A dimostrazione di questo fatto, il seguente grafico dell'errore relativo mostra come l'errore descresca molto velocemente prendendo le diadi associate ai valori singolari maggiori.

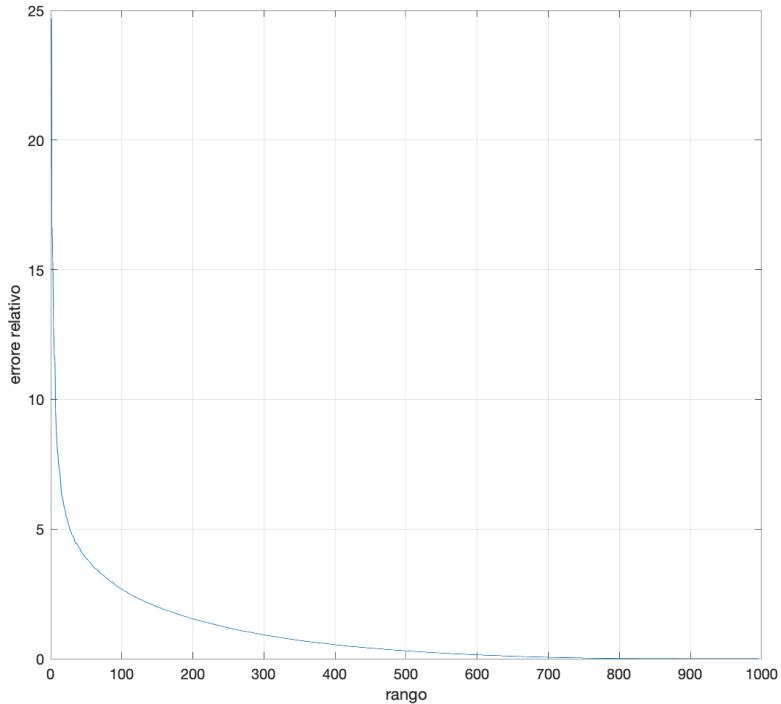


Figure 7: Errore relativo

Il grafico del fattore di compressione e dei valori singolari non necessitano di ulteriori commenti.

L'errore dell'approssimazione quindi decresce molto velocemente usando questo metodo, ed è possibile rendersene conto osservando che i valori singolari decrescono piuttosto lentamente, permettendo proprio di avere approssimazioni piuttosto efficaci. La tabella sottostante indica il numero minimo di diadi che sono necessarie per avere un errore minore di una certa percentuale.

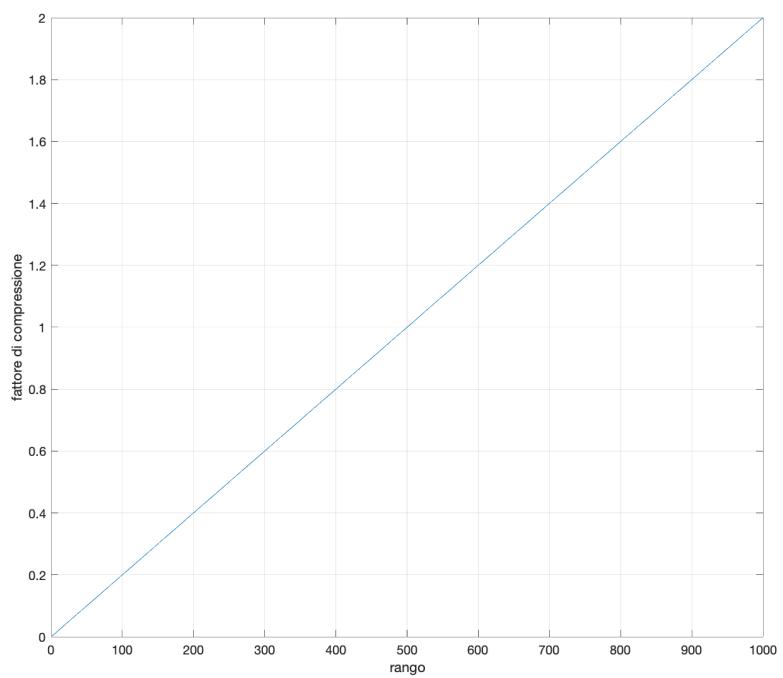


Figure 8: Fattore di compressione

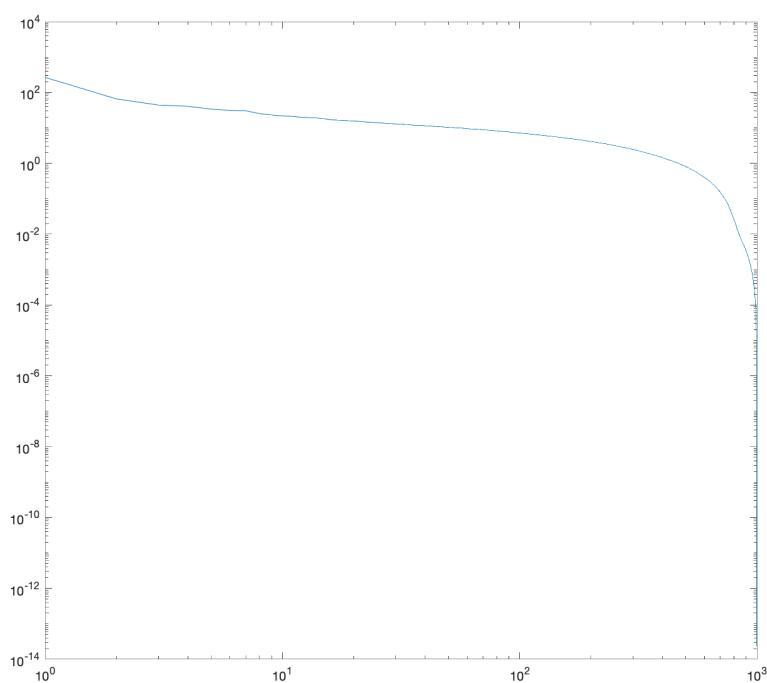


Figure 9: Grafico dei valori singolari

Immagine	errore < 1%	errore < 5%	errore < 10%
buzz.jpg	285	27	7
venice.jpg	106	14	5
wolf.jpg	32	17	9

Altri due esempi sono le immagini *venice.jpg* e *wolf.jpg*. Da notare come per l'ultima di queste, sia sufficiente un numero molto piccolo di diadi per avere una buona rappresentazione per via dei pochi dettagli che presenta.



Figure 10: venice.jpg

Nell'ultimo esercizio veniva chiesto di risolvere il problema dei **Minimi Quadrati Lineari** sia risolvendo il *sistema delle equazioni normali* sia utilizzando la *fattorizzazione SVD* di A. I risultati ottenuti non evidenziano differenze nel risultato usando i due metodi sopracitati; i polinomi generati sono infatti uguali. Le seguenti immagini mostrano il polinomio che approssima i dati del dataset indicato nella consegna dell'esercitazione per differenti valori di n (grado del polinomio).



Figure 11: venice.jpg compressa ad un quarto della dimensione



Figure 12: wolf.jpg

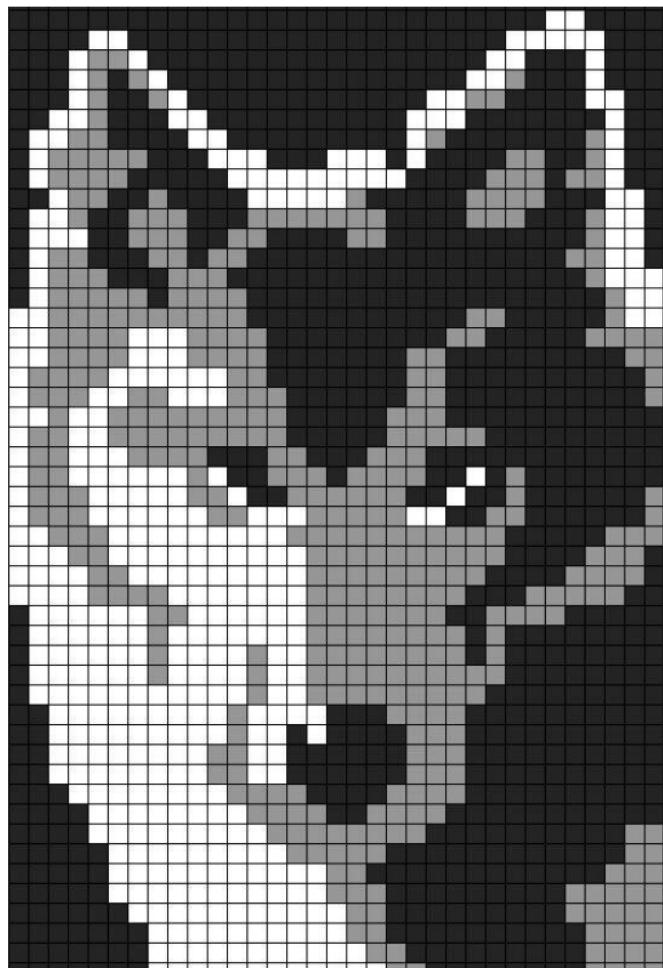


Figure 13: wolf.jpg compressa usando 60 diadi

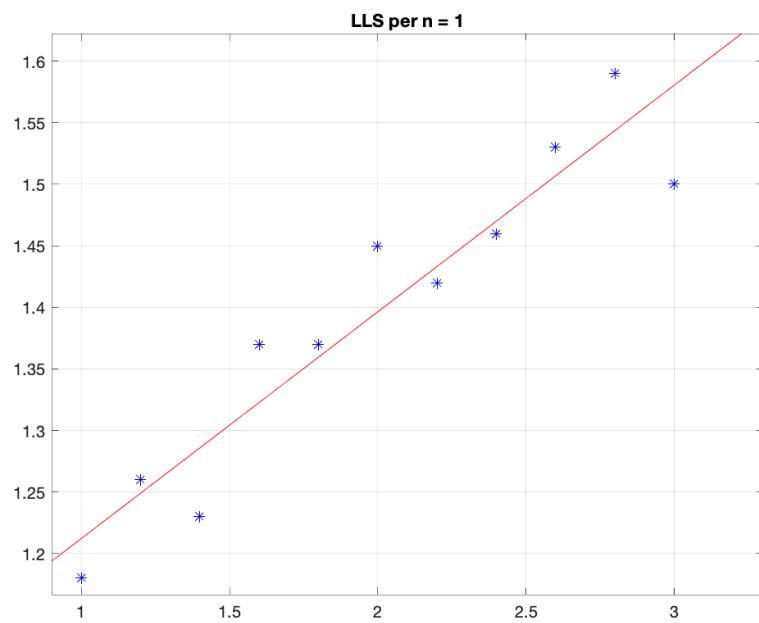


Figure 14: LLS per n = 1

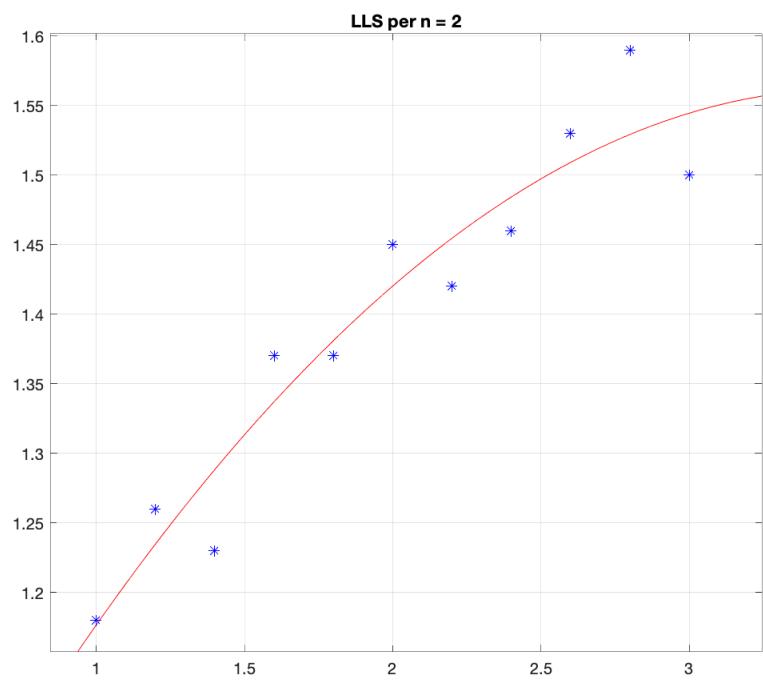


Figure 15: LLS per n = 1

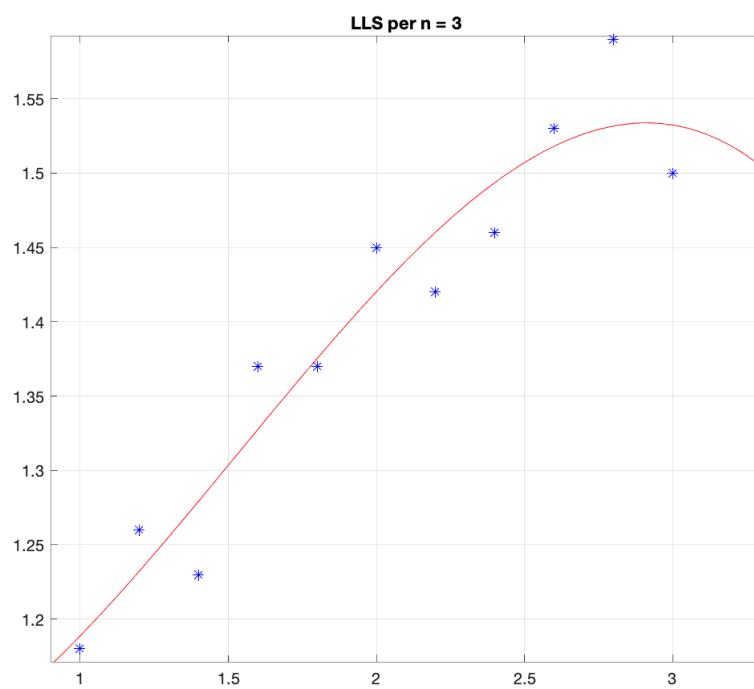


Figure 16: LLS per $n = 1$

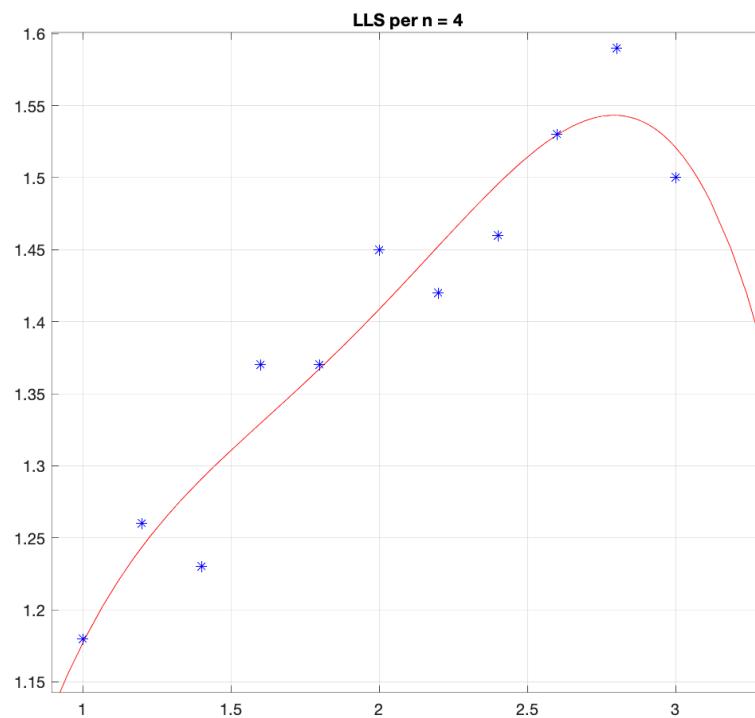


Figure 17: LLS per n = 1

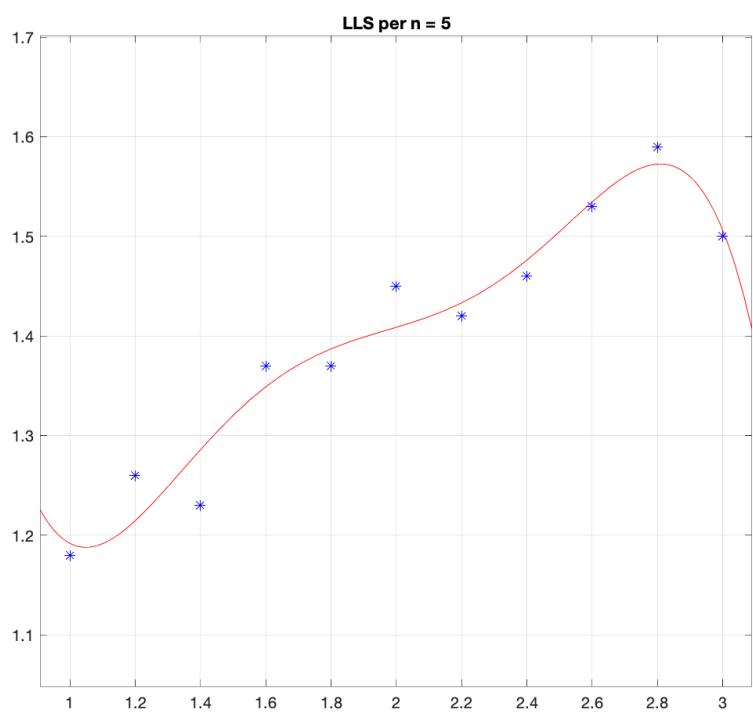


Figure 18: LLS per $n = 1$