Elaborato Calcolo Numerico

Elaborato realizzato da:
Francesco Erminio di Fruscio (M63/1004)
Antimo Barbato (M63/1079)

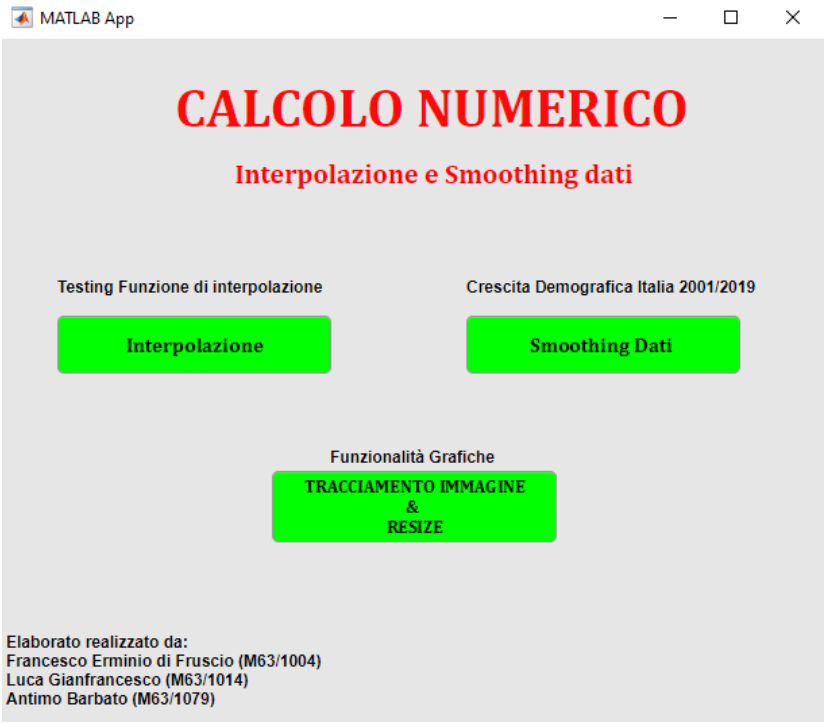
Introduzione

Una parte significativa del lavoro degli scienziati consiste nel raccogliere dati sperimentali.
Questi valori, che si traducono in punti su di un piano cartesiano, devono essere collegati fra loro attraverso una relazione matematica che dia luogo ad una unica curva che meglio “spieghi” questi dati.
La procedura che porta all'unione dei punti sul piano cartesiano identificando quindi l'insieme dei dati mediante una curva è definito fitting.
Di base in base al numero dei dati e alla qualità dei dati possiamo usare varie tecniche per generare la curva caratteristica dell'insieme dei dati che prendiamo in esame.
Tecniche che analizzeremo di seguito saranno basate sull’interpolazione e sullo smoothing.

Capitolo 1 : Interpolazione

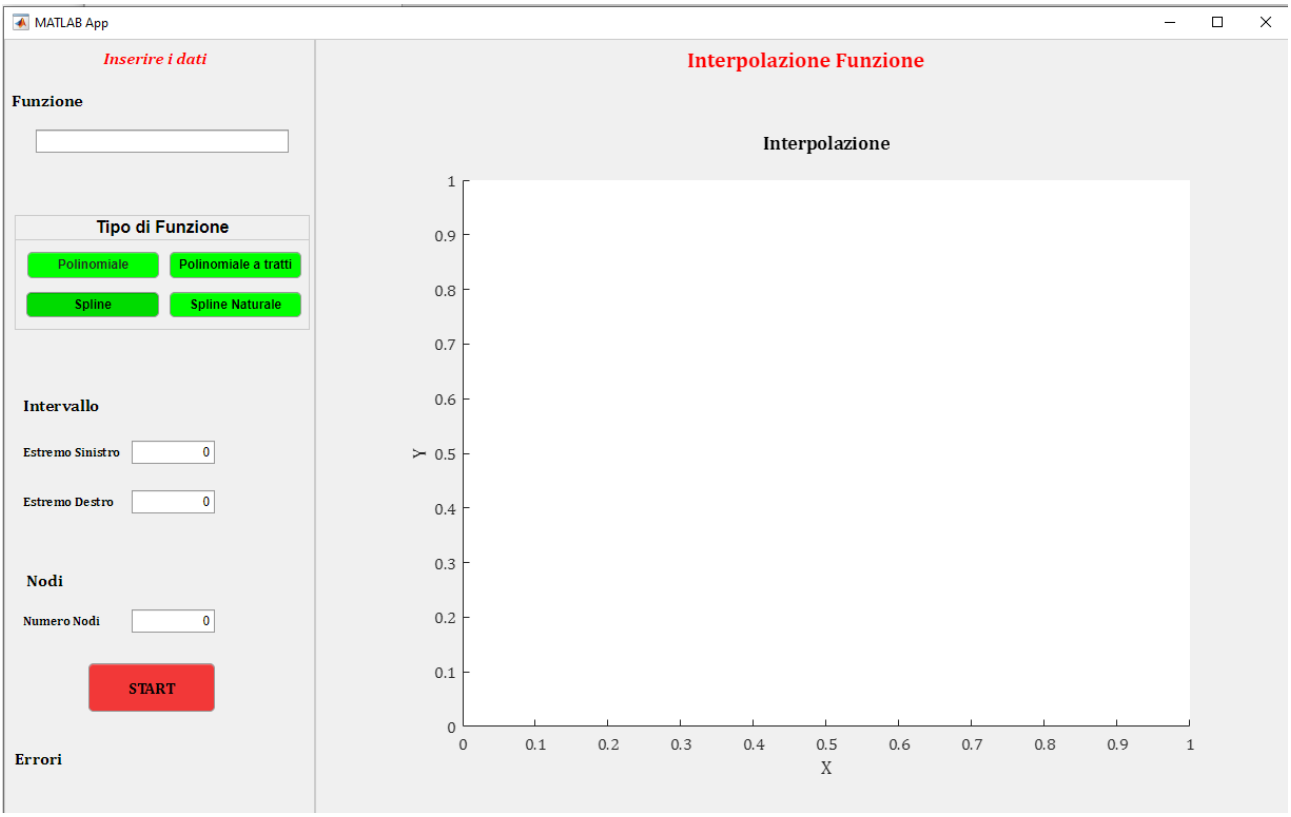
1.1 Menù

Abbiamo realizzato un applicazione sfruttando l’editor di matlab.
All’avvio dell’applicazione ci troveremo un menù che ci permetterà di accedere a 3 funzionalità:
1) Interpolazione: In questa sezione potremo testare le varie funzioni di interpolazione messe a disposizione da matlab e valutarne l’effettiva efficienza.
2) Tracciamento immagine & miglioramento: in tale sezione andremo ad approfondire gli aspetti grafici che ci offre la spline.
3) Smoothing Dati: in tale sezione siamo andati a valutare la crescita demografica italiana dal 2011 al 2019



1.2 Interpolazione

Alla selezione della funzionalità ‘’interpolazione’’ si aprirà la seguente finestra:



Con l’interpolazione andiamo a definire una funzione (di vario tipo) che ci permette , dato un insieme di dati esatti , di definire una curva passante per tali punti.
Tipi di funzioni :
-Polinomiale: ci permette di definire un polinomio su un intervallo [a,b] che andrà a rappresentare mediante una curva l’insieme di punti che vorremo collegare.
Problema => Al crescere dei nodi e quindi al crescere del grado , si genereranno delle oscillazioni.
Soluzione => Usiamo la polinomiale a tratti
-Polinomiale a tratti: ci permette di dividere l’intervallo [a,b] in sotto-intervalli , ognuno dei quali viene rappresentato con un polinomio di grado non troppo elevato in modo tale da evitare le oscillazioni.
Problema => Il polinomio non è regolare (ha delle discontinuità al suo interno)
Soluzione => Spline
- Spline => La spline è una polinomiale a tratti , quindi definisce per ogni sotto-intervallo un polinomio. Le derivate dei vari polinomi sono continue fino alla derivata di ordine m-1 dove m è il grado del polinomio.
Di solito la più usata è quella cubica (che usa matlab) , che di base avendo una derivata prima e seconda continue ci permette di rappresenta velocità e accelerazione continue.
-Spline cubica naturale : La spline cubica naturale si ha quando si definiscono le condizioni al contorno, dove gli estremi della derivata seconda saranno nulli. In tal modo minimizzo i fenomeni di oscillazione.

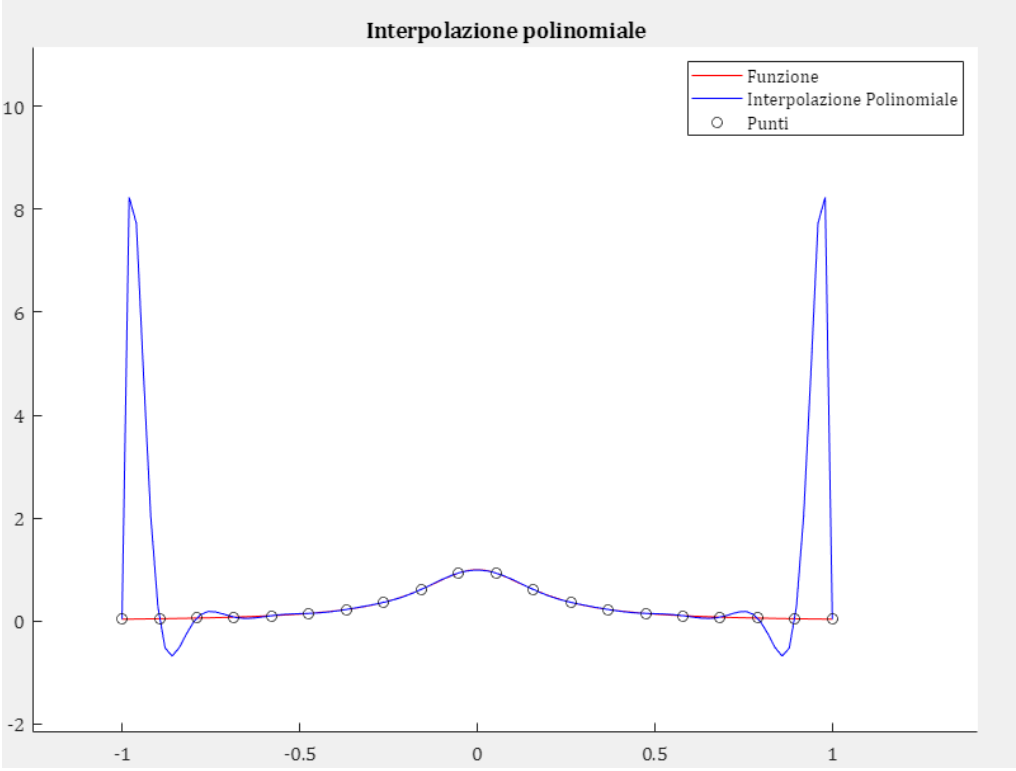
Di seguito riportiamo le funzioni usate in matlab:

-Polyfit : prende come ingressi i punti (x , y) che voglio andare ad interpolare e n che è il grado del polinomio. Restituisce i coefficienti per costruire il polinomio.
-Polyval : prende in ingresso i coefficienti calcolati con polyfit e un vettore ***t*** calcolato con linspace che spazia l'intervallo [a,b]. Restituisce il valore del polinomio nei singoli punti di ***t***
t ci serve poiché ci permette di considerare un numero maggiore di punti per rappresentare il grafico e quindi potremo andare a rappresentare una curva continua dal polinomio calcolato.
-Interp1 : prende in ingresso i punti (x,y) che vogliamo andare ad interpolare e il vettore ***t*** che spazia l'intervallo [a,b].
Quindi calcola i coefficienti del polinomio e ritrova i valori del polinomio nei punti del vettore t. (collega tra loro 2 punti).
-Spline : in matlab con tale funzione usiamo una spline cubica , quindi adottiamo un polinomio di terzo grado.
-Csape : restituisce i punti per una spline naturale (imponendo le condizioni di contorno) che poi verranno collegati mediante Ppval

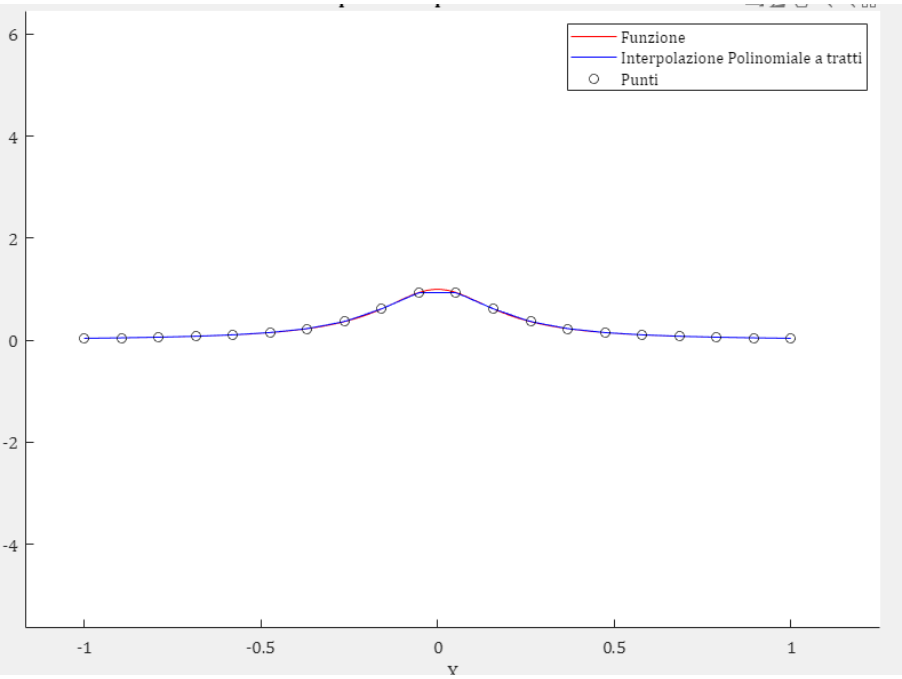
1.2 Casi di test

I casi di test sono stati effettuati tutti con i seguenti parametri:
-Intervallo [-1 , 1]
-Numero di nodi : 20

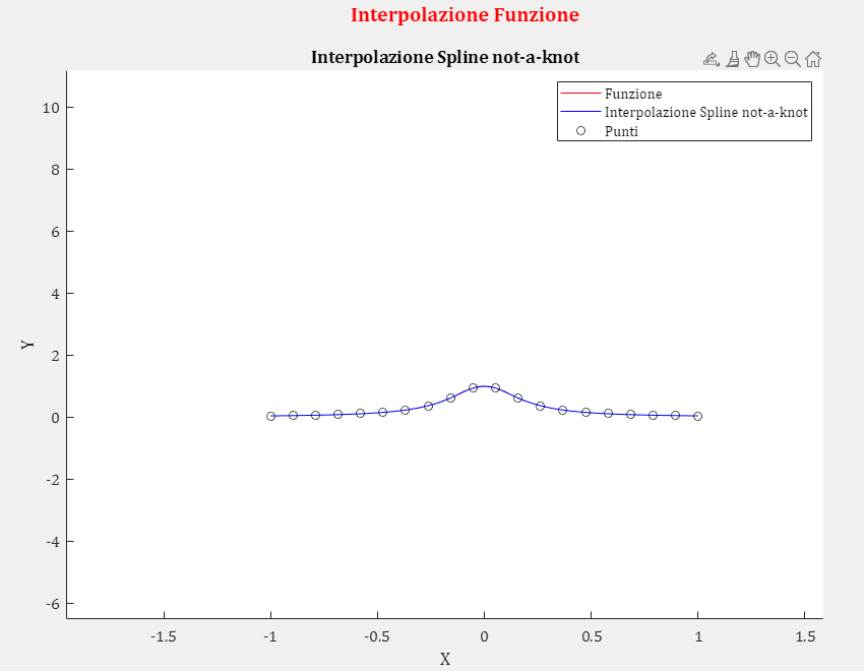
Nel primo grafico è stata utilizzata la Polinomiale e come si può notare ci sono delle oscillazioni causate dal grado elevato del polinomio



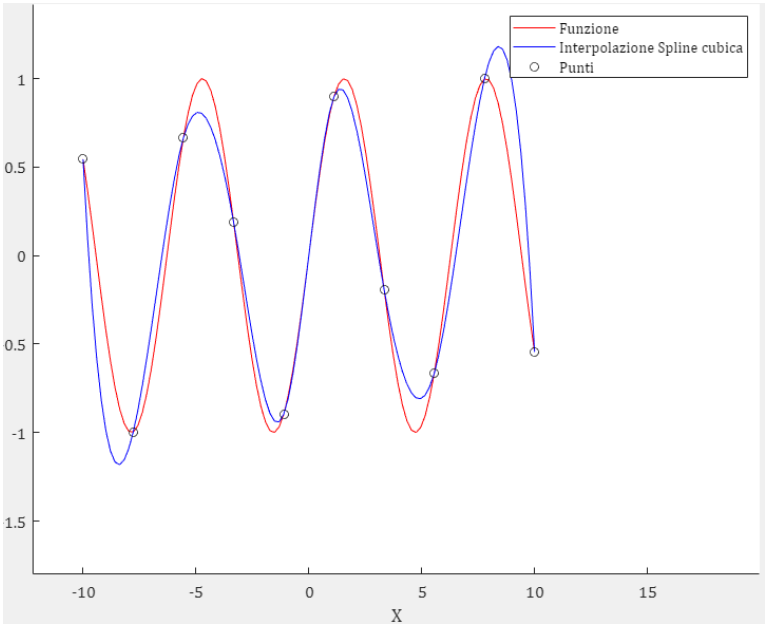
Nel secondo grafico è stata usata la polinomiale a tratti , non si hanno delle oscillazioni ma, a causa dei problemi derivanti dalla mancanza di continuità delle derivate, la curva è in realtà composta da tratti che non riescono a seguire perfettamente l'andamento della funzione.



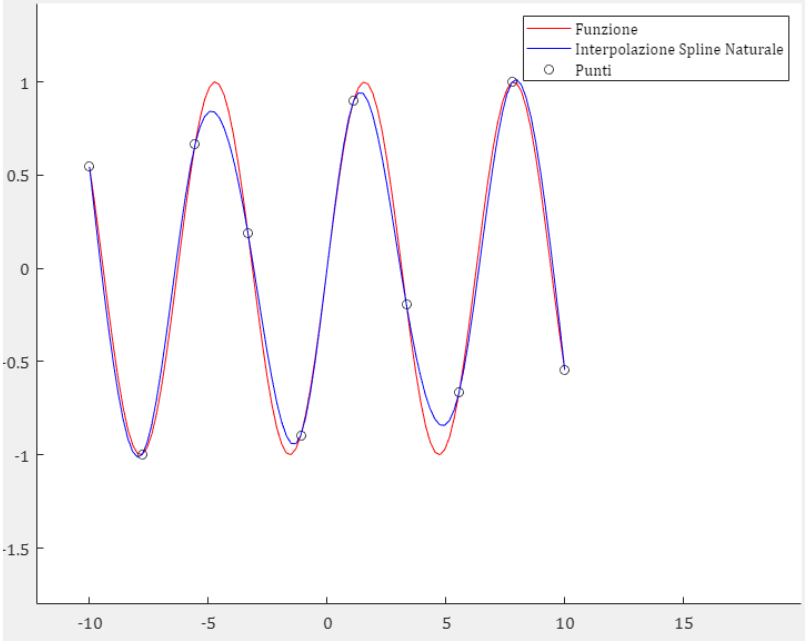
Nel grafico successivo usiamo la funzione di spline mediante la quale otteniamo effettivamente un risultato ottimale
Si otterrà un grafico pressoché identico anche per la spline naturale



Si forniscono ulteriori casi di test per la funzione di spline cubica e spline cubica naturale per evidenziare differenze nel processo di tracciamento della curva.
La funzione presa in esame in questo caso è sin(x) , valutata in un intervallo [-10,10] con numero di nodi pari a 10
Nel primo grafico , è riportato il grafico della funzione di spline cubica che fornisce un errore pari a 5.86*10^-1



Il secondo grafico invece è tracciato con spline cubica naturale che fornisce un errore pari a 1.77*10^-1.



Di conseguenza , valutando l'errore si nota che la spline cubica naturale fornisce un errore minore rispetto alla spline cubica.

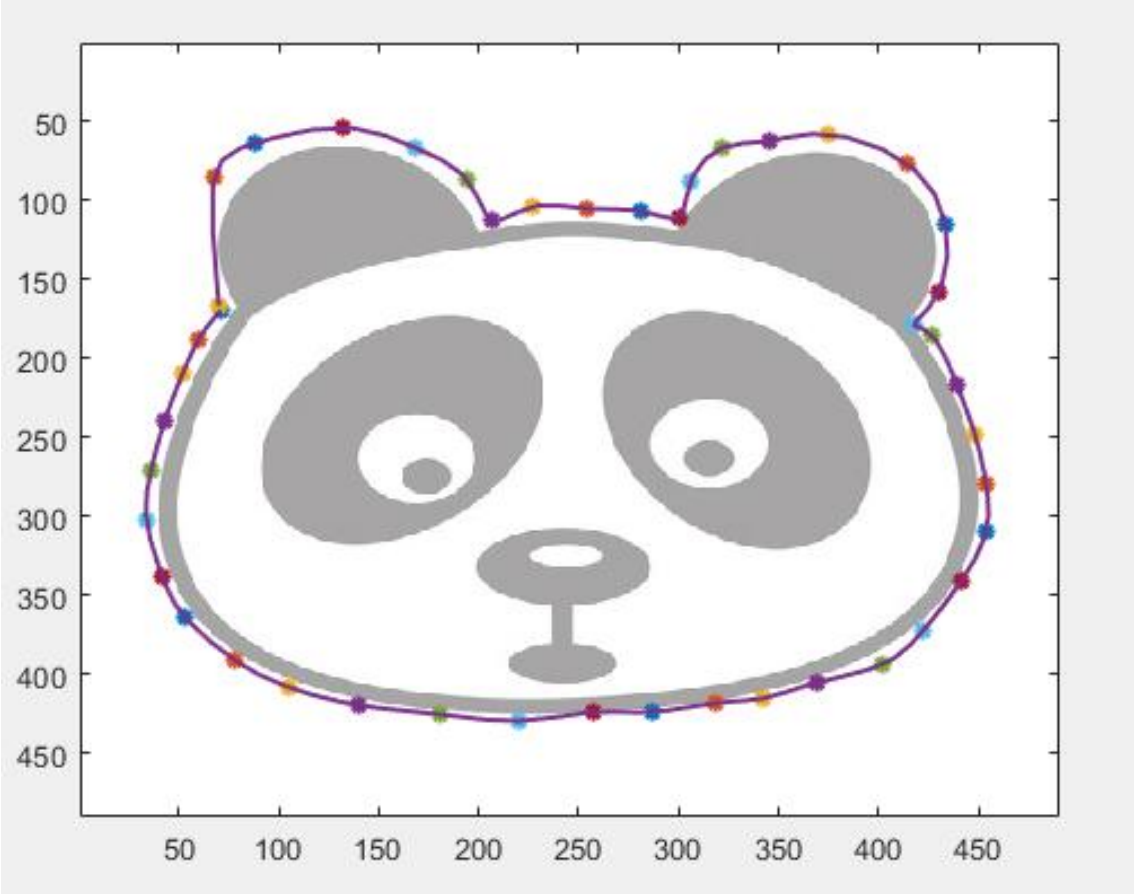
Capitolo 2 : Tracciamento e Resize immagine

In tale sezione si mostreranno le potenzialità nell'ambito della grafica 2D delle funzioni di interpolazione e quindi più in generale si vedranno alcune applicazioni pratiche per quanto riguarda l'interpolazione.

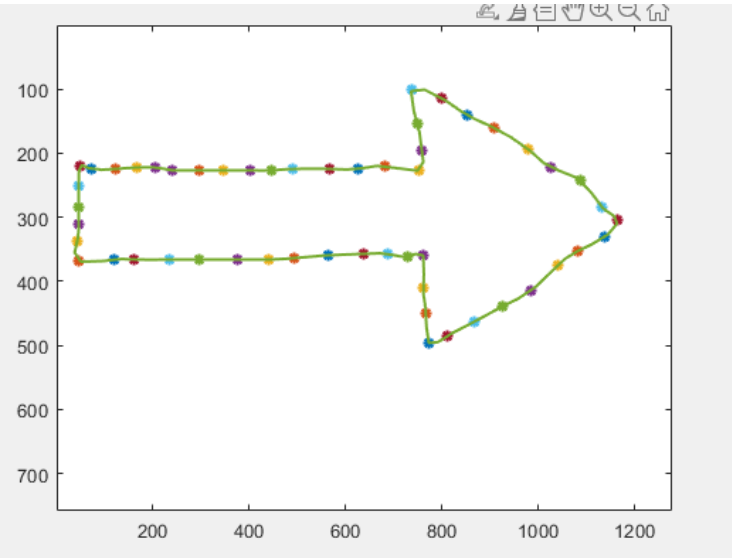
2.1 Tracciamento immagine

Nell'applicazione matlab cliccando sulla funzionalità di ''Tracciamento Immagine & Resize" accediamo ad una nuova finestra che ci mette a disposizione 2 funzionalità.
La prima traccerà l'immagine con un numero prefissato di punti di interpolazioni , tracciati manualmente dall'utente mentre la seconda ci permetterà di effettuare un resize dell'immagine.
In matlab per effettuare il tracciamento sfruttiamo la funzione ginput , inserita in un apposito ciclo while che ciclerà fintanto che non avremo disegnato il numero di nodi definiti nell'interfaccia applicativa.

Nell’immagine che segue abbiamo tracciato il contorno di un panda con circa 25 nodi di interpolazione

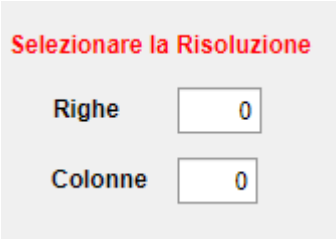


Nell’immagine che segue abbiamo realizzato il contorno di una freccia su sfondo bianco con circa 55 nodi di interpolazione



2.2 Dimensionamento immagine

In tale sezione si andrà ad effettuare il ridimensionamento di un immagine mediante la funzione imresize() Usando imresize() si può specificare la dimensione dell'immagine in output , si può specificare il metodo di interpolazione utilizzato e si può specificare anche il filtro da utilizzare per prevenire l'aliasing (non utilizzato nel nostro caso).
Utilizzando la funzione imresize() si possono specificare le dimensioni dell'immagine in uscita mediante un vettore che contiene il numero di righe e colonne , tale vettore viene trasmesso in input mediante l'interfaccia.



Come detto in precedenza , si può specificare il metodo di interpolazione da utilizzare per la realizzazione dell'immagine.
Si possono usare 3 tipi di funzioni di interpolazione per lavorare con i pixel:
- Nearest – neighbor (intorno del punto) : al pixel della nuova immagine viene assegnato il valore del pixel più vicino dell'immagine originale.
- Bilineare: al pixel della nuova immagine viene assegnata una media pesata dei valori dei pixel nell'intorno 2x2 dell'immagine originale. La media pesata è basata sulla distanza , i pixel più vicini hanno un peso maggiore dei pixel più distanti (ciò vale anche per la bicubica)
- Bicubica : al pixel della nuova immagine viene assegnata una media pesata dei valori dei pixel nell'intorno 4x4 dell'immagine originale.
La funzione di interpolazione viene utilizzata in modo tale da mappare l'immagine originale su un immagine dotata di un numero diverso di pixel (o righe e colonne).
Si specifica che maggiore è il numero di pixel considerato migliore sarà la qualità dell'immagine risultante , a discapito del tempo computazionale necessario per l'interpolazione.
Questo implica che il tempo di risoluzione di un interpolazione nearest-neighbor risulterà minore di quello della bilineare, mentre la bicubica avrà un tempo di risoluzione superiore agli altri due metodi.

Di seguito , si riportano i risultati di un esempio basato su uno screenshot del codice.

```
bilineare = imresize(immagine , [app.RigheEditField.Value , app.ColonneEditField.Value] , 'bilinear');
imwrite(bilineare , 'bilineare.jpg')

bicubica = imresize(immagine , [app.RigheEditField.Value , app.ColonneEditField.Value] , 'bicubic');
imwrite(bicubica , 'bicubica.jpg')

nearest = imresize(immagine , [app.RigheEditField.Value , app.ColonneEditField.Value] , 'nearest');
imwrite(nearest , 'nearest.jpg')
```

Figura 1 : Immagine Originale

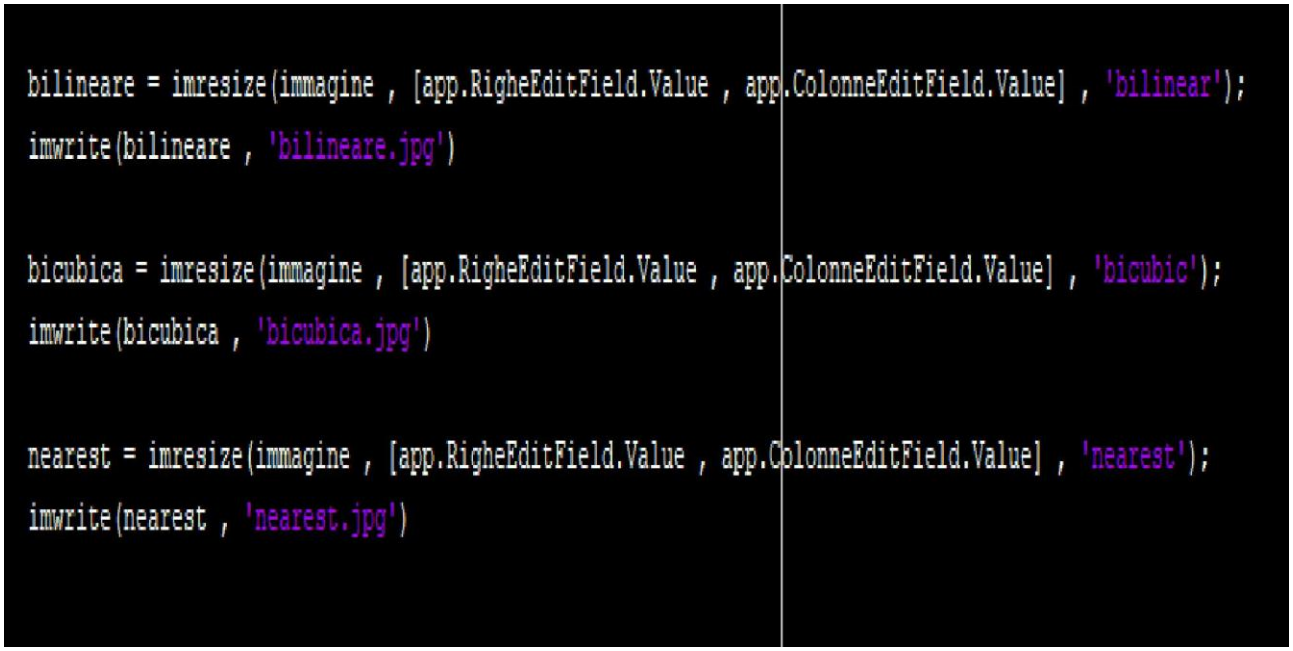


Figura 2: Immagine con interpolazione nearest

```
bilineare = imresize(immagine , [app.RigheEditField.Value , app.ColonneEditField.Value] , 'bilinear');
imwrite(bilineare , 'bilineare.jpg')

bicubica = imresize(immagine , [app.RigheEditField.Value , app.ColonneEditField.Value] , 'bicubic');
imwrite(bicubica , 'bicubica.jpg')

nearest = imresize(immagine , [app.RigheEditField.Value , app.ColonneEditField.Value] , 'nearest');
imwrite(nearest , 'nearest.jpg')
```

Figura 3: Immagine con interpolazione bilineare


```
bilineare = imresize(immagine , [app.RigheEditField.Value , app.ColonneEditField.Value] , 'bilinear');
imwrite(bilineare , 'bilineare.jpg')

bicubica = imresize(immagine , [app.RigheEditField.Value , app.ColonneEditField.Value] , 'bicubic');
imwrite(bicubica , 'bicubica.jpg')

nearest = imresize(immagine , [app.RigheEditField.Value , app.ColonneEditField.Value] , 'nearest');
imwrite(nearest , 'nearest.jpg')
```


Figura 4: Immagine con interpolazione bicubica

Si può notare come da un punto di vista puramente visivo , la nearest fornisce dei risultati peggiori rispetto alla bicubica e alla lineare.

Capitolo 3

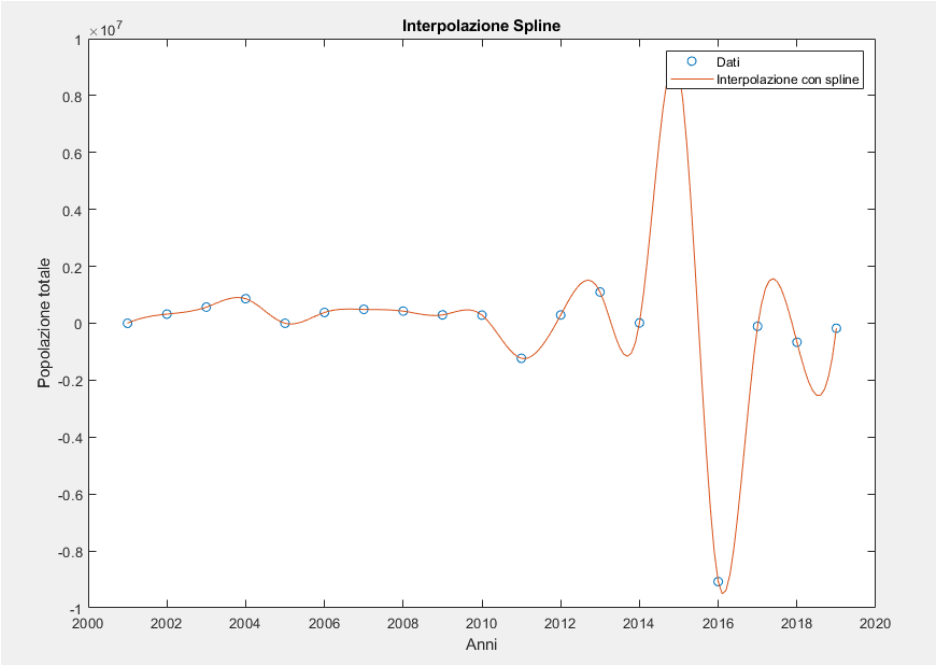
Per quanto riguarda questo capitolo andremo a trattare il fitting dei dati che può essere realizzato mediante due tecniche : Interpolazione e Smoothing.
Di base può essere può conveniente in alcuni casi avere una curva più approssimata per vedere con maggior chiarezza l'andamento del fenomeno di studio e in tal senso si usa lo smoothing.

3.1 Crescita Demografica in Italia dal 2001 al 2019

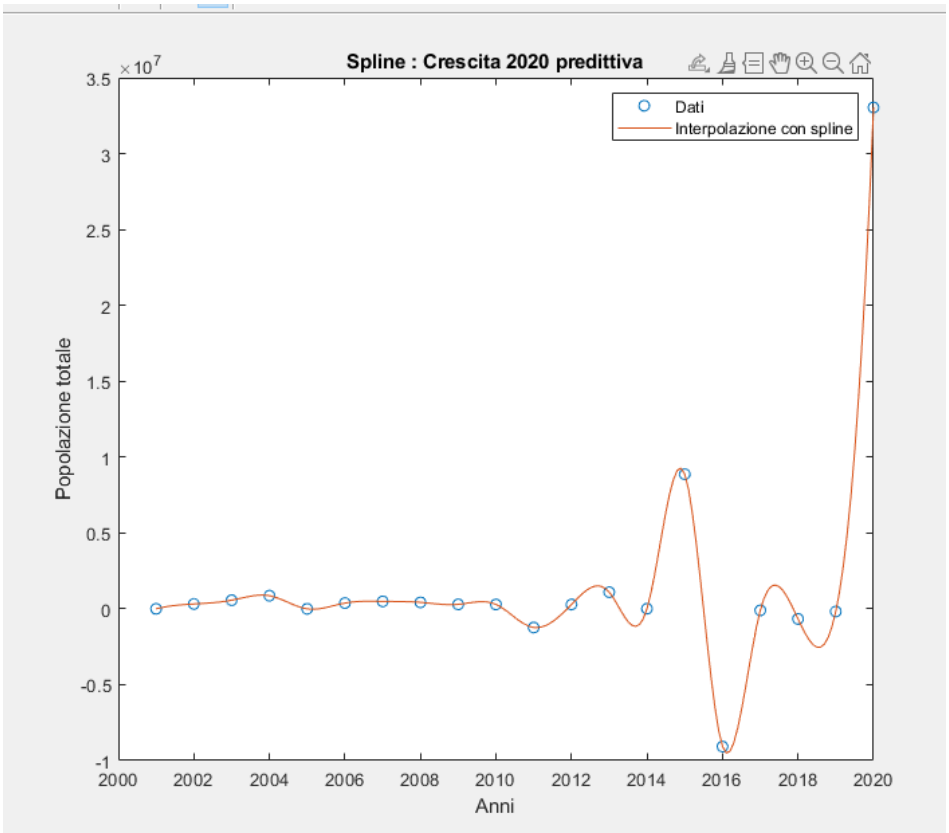
Alla pressione del bottone del menù principale ci apparirà tale finestra

In generale potremo effettuare varie valutazioni in base al tipo di funzione usata.
Anzitutto mediante il pulsante ‘’Grafico spline’’ usando l’apposita funzione di spline definita da matlab andremo a definire la curva caratteristica dato l’insieme di dati analizzato.
Andiamo quindi a rappresentare la crescita (o anche decrescita) annua.

Il grafico di spline è mostrato nell’immagine successiva

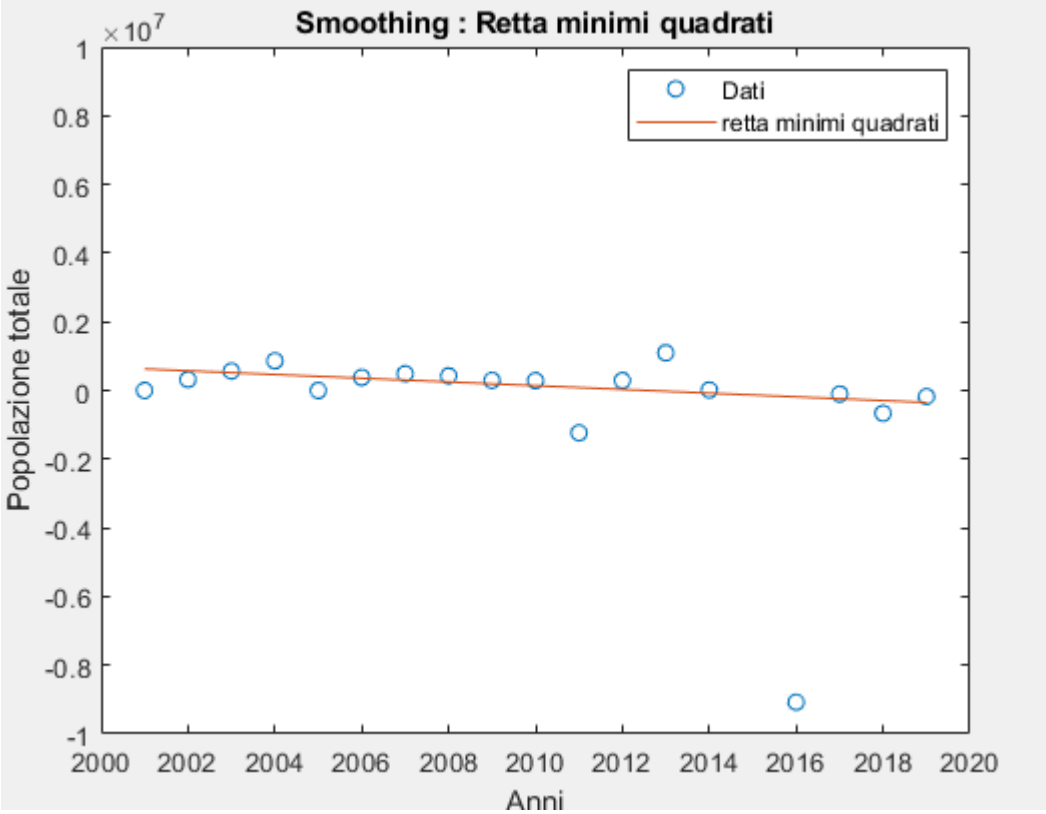


In generale poi sfruttando le proprietà predittive della spline siamo in grado anche di valutare l'eventuale crescita o decrescita data nell'anno successivo cioè il 2020.



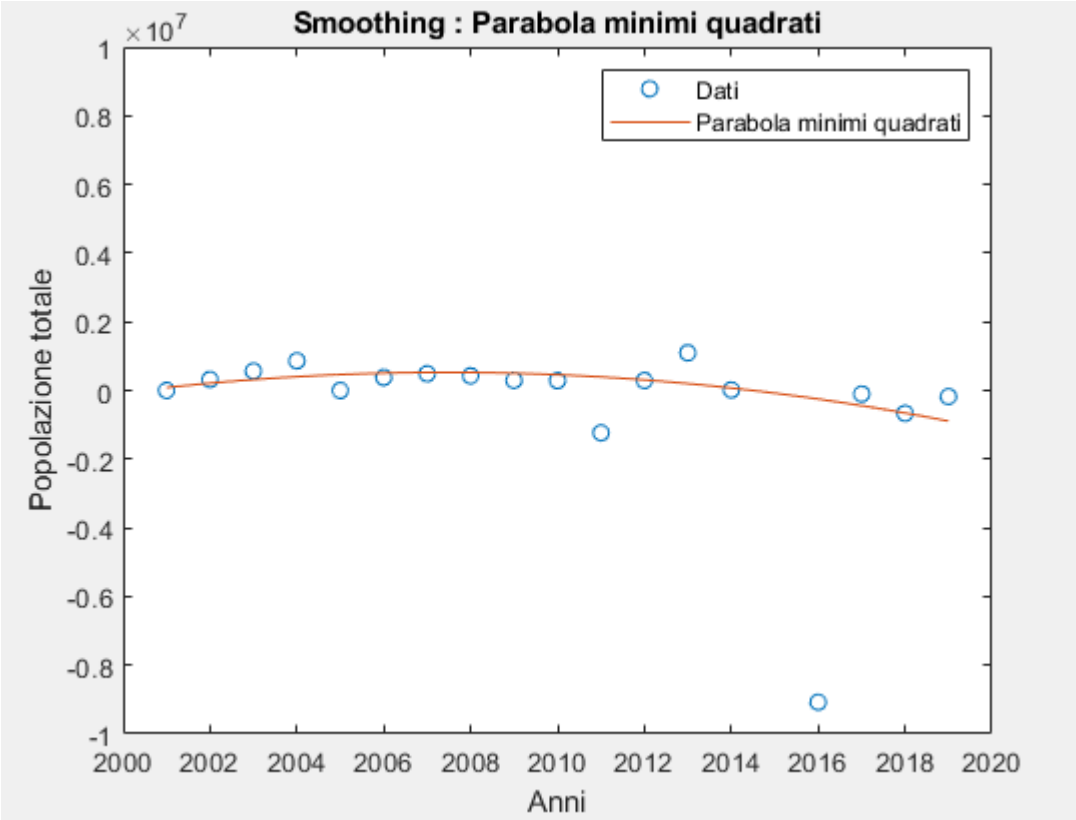
Si può vedere che nel 2020 dovrebbe esserci una crescita della popolazione.

Con le funzioni di smoothing invece siamo in grado di definire un grafico più approssimato ma che ci permetterà di definire l'andamento complessivo della crescita demografica in italia.



Nella figura sovrastante , abbiamo usato la retta dei minimi quadrati per rappresentare la crescita demografica annua della popolazione.

Di base notiamo che la retta ha andamento decrescente , di conseguenza dal 2001 al 2019 c'è stato complessivamente una decrescita della popolazione.



L'altro metodo utilizzato è quello della ''parabola dei minimi quadrati'' , la quale conferma già le osservazioni definite precedentemente