

Realizzazione di un'interfaccia grafica per la visualizzazione del comportamento degli Algoritmi di De Casteljau e De Boor

Studente: Massimiliano Sirgiovanni

Matricola: 7077251

Email: massimiliano.sirgiovanni@stud.unifi.it

L'obiettivo principale di questo progetto è la **realizzazione di un'interfaccia grafica**, nella quale è possibile **osservare il comportamento degli algoritmi di De Casteljau e De Boor**, modificando, eventualmente, i parametri in input.

Per prima cosa, è stato necessario **implementare gli algoritmi** sopracitati, nel caso delle curve e delle superfici.

Successivamente, è stato creato un ambiente di visualizzazione e modifica del procedimento algoritmico.

L'**algoritmo di De Casteljau** si pone l'obiettivo di valutare le **curve di Bezier** in un determinato punto, specificato come parametro dell'algoritmo.

Si tratta di un metodo ricorsivo che consente di calcolare i punti delle curve di Bezier, passando come input esclusivamente i punti di controllo ed un parametro t , compreso nell'intervallo $[0, 1]$.

L'algoritmo può essere poi **esteso al caso delle superfici**, prendendo, in questo caso, oltre ai punti, due parametri, u e v , sempre compresi tra zero ed uno.

L'**algoritmo di De Boor** è simile a quello di De Casteljau, con la differenza che viene applicato alle **curve di B-Spline** (o *patch B-Spline nel caso delle superfici*).

I parametri presi in input, in questo caso, sono i punti di controllo, il vettore esteso dei nodi ed un parametro, t , questa volta compreso tra $[a, b]$, corrispondenti al primo ed all'ultimo valore dei nodi interni nel vettore esteso dei nodi.

Nel **caso delle superfici**, come in precedenza, i parametri da passare all'algoritmo sono due, u e v , anch'essi compresi nell'intervallo $[a, b]$.

Gli algoritmi sono stati implementati seguendo la teoria vista a lezione, per lo più, e dunque sfruttando massivamente il **principio di località**.

Inoltre, si è deciso di **rendere visibili all'utente i vari step degli algoritmi**, tramite l'utilizzo di istruzioni "pause" ed il plotting delle fasi intermedie.

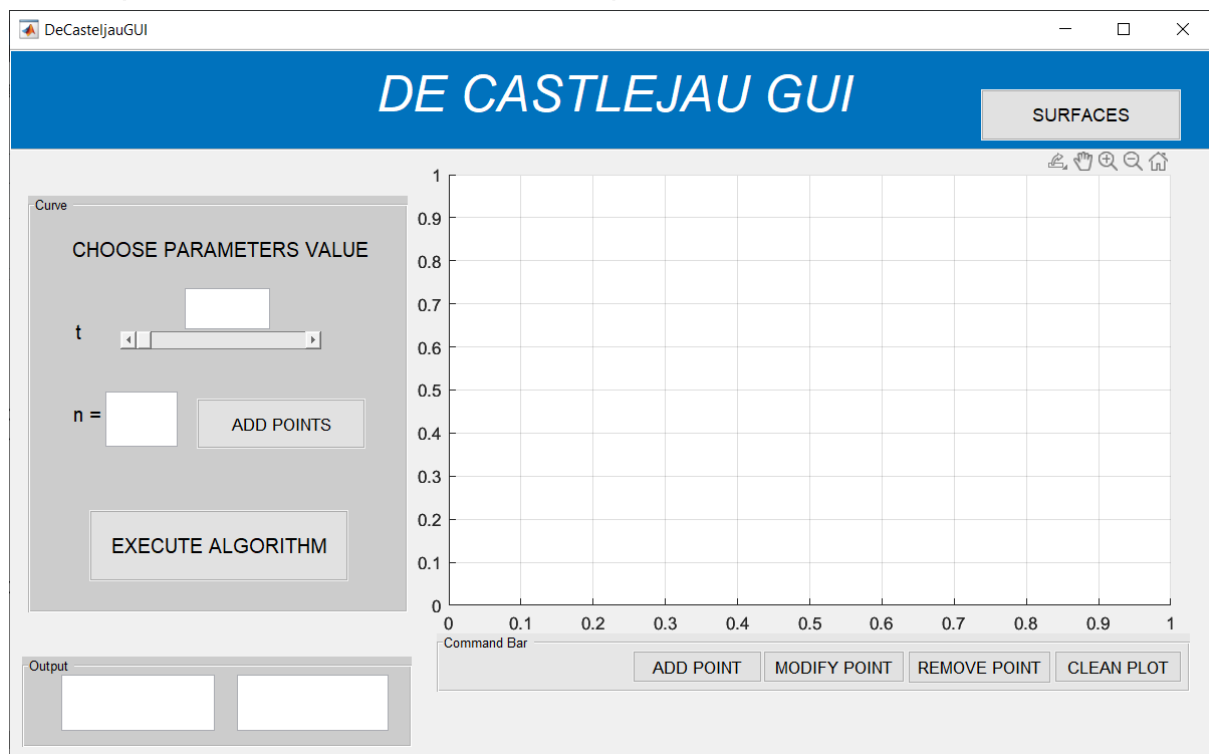
Interfaccia Grafica

Una volta implementati gli algoritmi, è stato necessario costruire un ambiente che permettesse all'utente di **inserire i parametri** desiderati, *in maniera semplice ed intuitiva*, e che permettesse una **visualizzazione**, *dal punto di vista grafico*, del **procedimento algoritmico** sottostante e dei **risultati ottenuti**.

Per modellare tale interfaccia, è stata utilizzata una funzionalità di Matlab che consente di realizzare una **GUI** (*Graphical User Interfaces*).

Per accedere all'editor è sufficiente digitare, sulla console matlab, il comando **guide**.

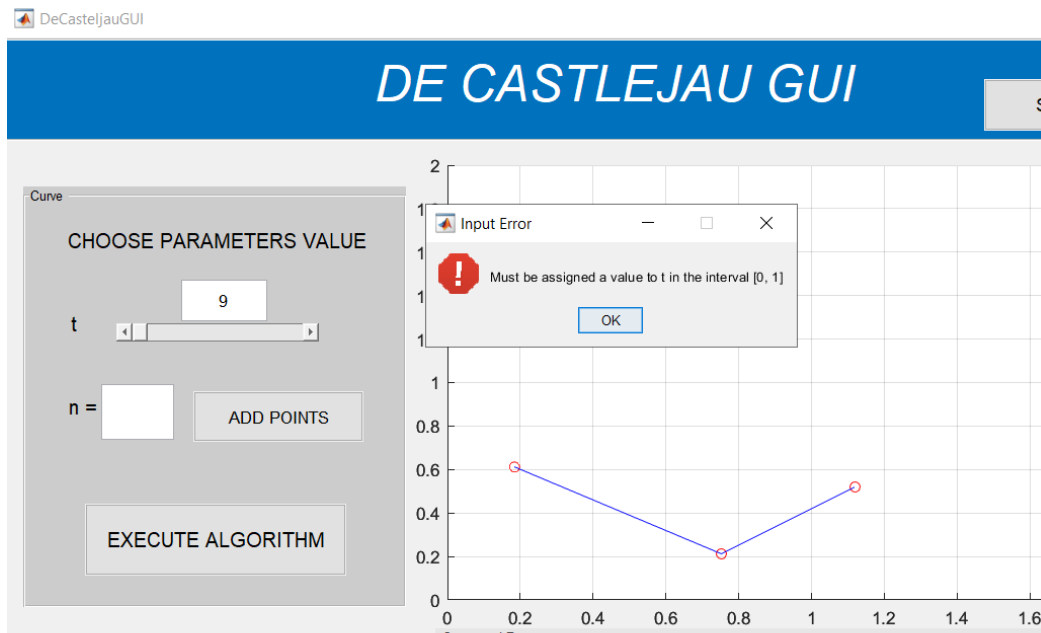
Sono state create **due diverse interfacce**, una per l'algoritmo di De Casteljau ed una per De Boor. Le due interfacce, però, hanno una struttura molto simile.



Nell'immagine di sopra è possibile osservare **due sezioni**, un pannello contenente dei blocchi per l'inserimento del testo, utile per l'**inizializzazione dei parametri** dell'algoritmo da parte dell'utente, ed una zona adibita alla **visualizzazione del plot**.

Per l'esecuzione dell'algoritmo si deve settare un **parametro t**, *compreso tra 0 e 1*. Per impostare tale parametro si può usare lo **slider** posto accanto alla stringa "t" oppure si può direttamente scrivere il valore di t nel box.

Utilizzando lo slider si rimane, *necessariamente*, in quest'intervallo, mentre inserendo il valore manualmente si potrebbe porre un valore non idoneo per l'algoritmo. In questo secondo caso, viene restituito un **messaggio d'errore**.

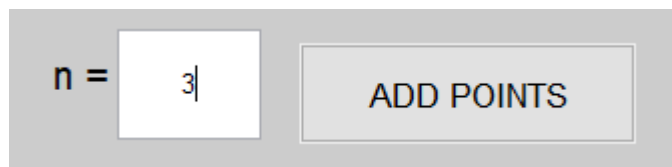


Un messaggio d'errore compare sullo schermo anche se non viene selezionato nessun punto, prima dell'esecuzione dell'algoritmo.

Selezione dei punti

Per **selezionare i punti** da passare all'algoritmo, si possono usare diverse opzioni.

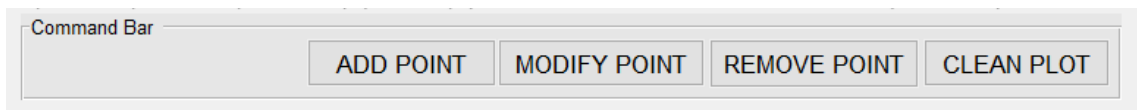
Si possono aggiungere più punti assieme, *tramite l'opzione visibile nella figura a destra*, specificando prima il numero di punti che si desidera aggiungere. In questo modo, verranno eseguite tante ***gininput()*** quanto è il valore inserito dall'utente.



Per rendere l'inserimento dei punti **visivamente più gradevole**, è stato realizzato un ciclo for che richiede i punti all'utente uno per volta e vengono rappresentati graficamente i punti fino ad ora scelti.

```
for i=1:n
    new_point = ginput(1);
    b = [b; new_point];
    plot(b(:, 1), b(:, 2), 'ro');
    plot(b(:, 1), b(:, 2), 'b');
```

D'altra parte, è anche possibile usare i comandi presenti nella zona sottostante al plot.



Tramite i seguenti comandi è possibile selezionare un nuovo punto, da aggiungere all'insieme dei punti da usare per l'algoritmo, modificare la posizione di un punto, rimuovere un punto ed eliminare tutti i punti fino ad ora selezionati.

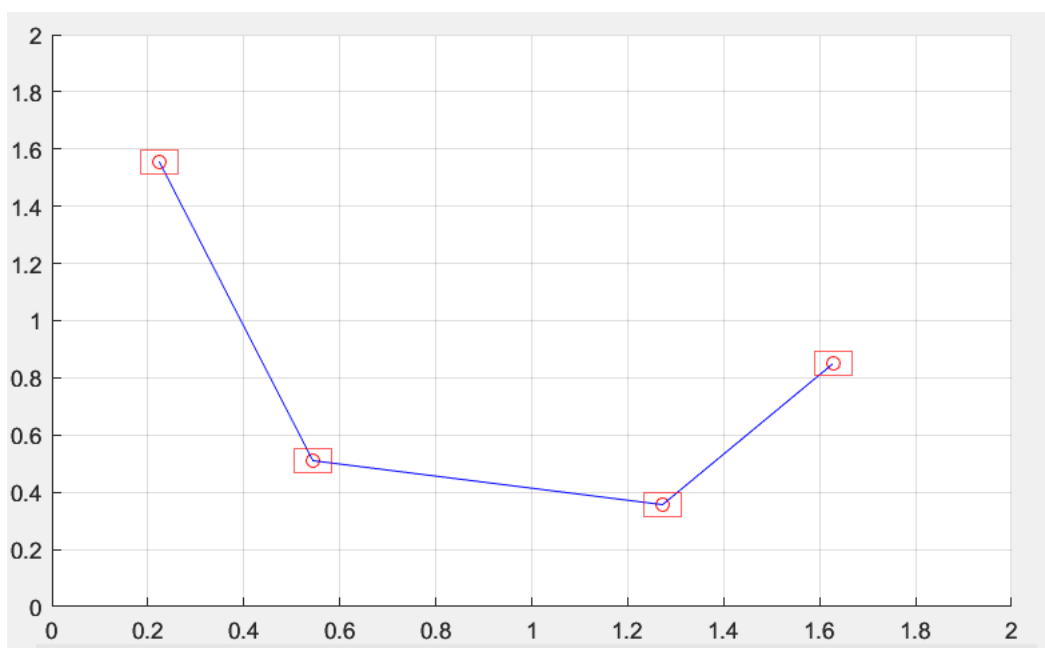
Per l'aggiunta di un punto, viene semplicemente utilizzata una ***ginput()*** ed il punto selezionato tra gli assi viene aggiunto alla lista dei punti.

Rimozione e Modifica dei punti

Per poter **cancellare o modificare uno dei punti** precedentemente selezionati, è **prima necessario individuarlo**.

Per farlo, si è deciso di utilizzare un sistema simile ad una tecnica spesso usata nell'ambito videoludico, le **hitbox**.

Ovvero, si è deciso di definire un **quadrato attorno ai vari punti**, in modo tale che un singolo punto possa essere **rappresentato da un'area più grande** nel piano cartesiano.



Ovviamente, tali riquadri saranno poi **invisibili all'utente**, il quale dovrà solo cercare di cliccare sui punti (*definiti dai cerchi rossi*), ma gli viene lasciato un discreto **margin di errore**.

La grandezza di questi riquadri può essere modificata all'interno del codice, aumentandone o diminuendone le dimensioni.

C'è da tenere in considerazione che, **aumentando troppo la dimensione** dei riquadri, si rischia di creare più facilmente **problemi dovuti alla sovrapposizione dei riquadri**.

Diminuendo eccessivamente le dimensioni, si rischia di **non rendere agevole all'utente la selezione dei punti**.

Per selezionare un punto, dunque, viene poi usata una ***ginput()***, e, tramite codice, si verifica se il punto selezionato è contenuto nell'area appartenente a quel punto. Se così è, il punto viene selezionato, altrimenti, apparirà a schermo un messaggio di errore.

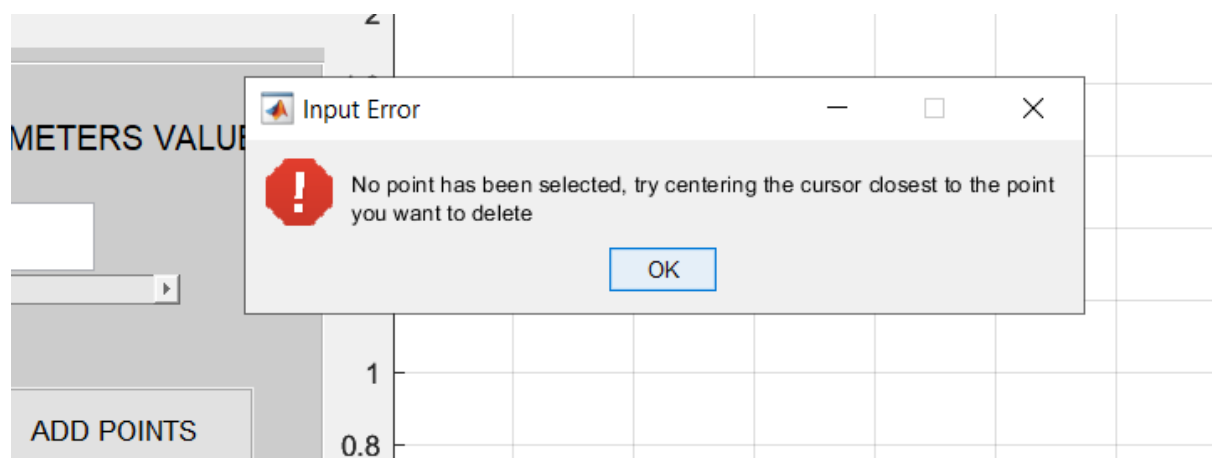
```
d = 0.04;
rmv = ginput(1);
set(handles.pushbutton6, 'visible', 'off');
rmved = 0;
for i=length(b):-1:1
    if rmv(1) >= b(i, 1)-d && rmv(2) >= b(i, 2)-d && rmv(1) <= b(i, 1)+d && rmv(2) <= b(i, 2)+d
        b(i, :) = [];
        rmved = 1;
        break
    end
end
```

Nel codice sopra riportato, si può osservare il fulcro della procedura di selezione e rimozione del punto.

La **grandezza dei riquadri** viene definita dalla variabile “*d*”.

La selezione del punto avviene sotto forma di ***ginput()***, rimandando all'utente la selezione. Infine, si verifica se il punto è contenuto all'interno del riquadro relativo all'*i*-esimo punto, con *i* compreso tra 1 ed il numero di punti totali.

Nel caso in cui il **punto non venga identificato**, come anticipato, viene stampato il **messaggio di errore**, visibile nella *Figura sottostante*, e la procedura di eliminazione/modifica viene interrotta.

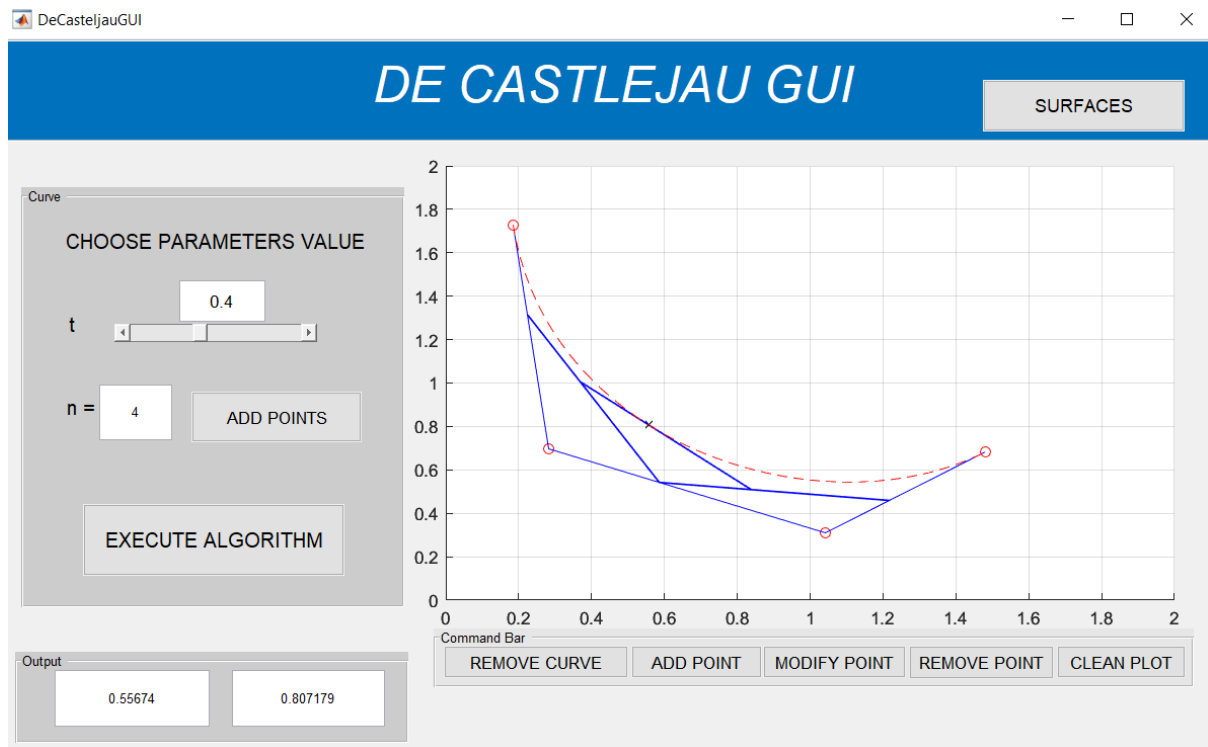
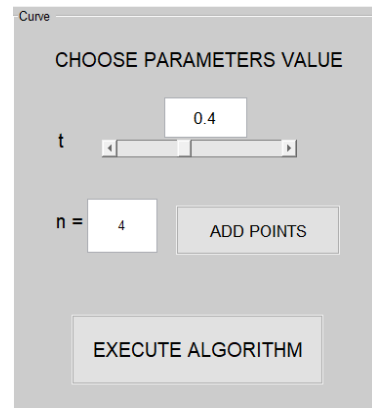


Esecuzione dell'Algoritmo

Una volta ottenuti tutti i parametri richiesti, è sufficiente cliccare sul pulsante “**EXECUTE ALGORITHM**”, presente nel pannello nella parte sinistra dell'interfaccia, per proseguire con l'esecuzione dei passi dell'algoritmo.

Una volta eseguito, è possibile osservare, graficamente, **tutti i passi ricorsivi dell'algoritmo di De Casteljau**.

Una volta terminata l'esecuzione dell'algoritmo, è anche possibile inserire all'interno del grafico la **curva di Bezier** che sta venendo valutata dall'algoritmo, ed ottenuta considerando i punti di controllo selezionati dall'utente.

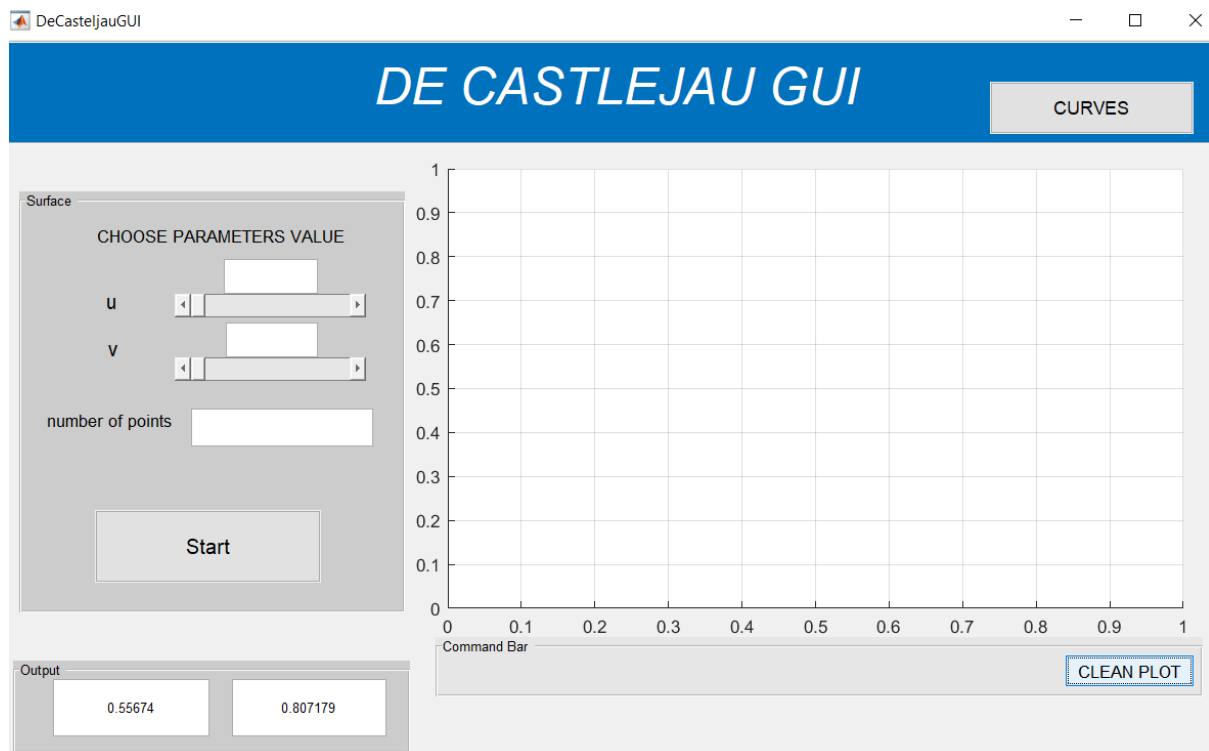
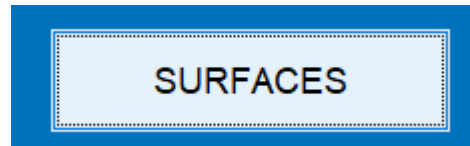


Il punto trovato dall'algoritmo di De Casteljau è contrassegnato da una “x” grigia nel plot, come si vede nella Figura posta sopra. Inoltre, **le coordinate del punto** vengono inserite in basso a sinistra, nel pannello denominato “Output”.

Le Superfici

Fino a questo momento, si è osservato il funzionamento dell'algoritmo di De Casteljau nel caso delle curve di Bezier. Tuttavia, è stato affrontato anche il caso delle superfici.

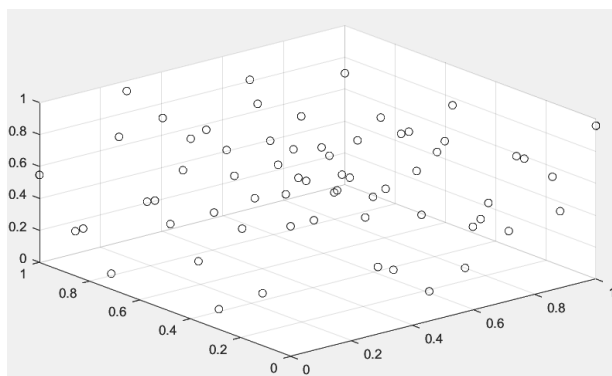
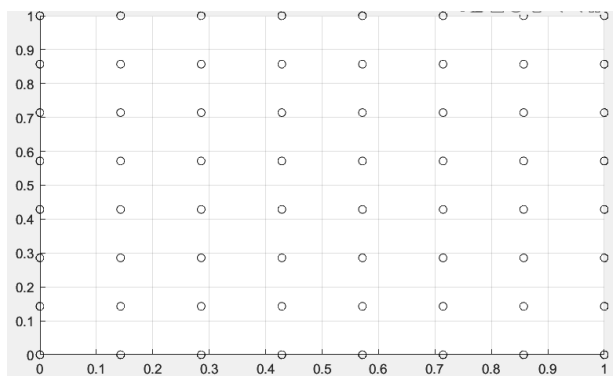
Per poter passare dallo studio delle curve allo studio delle superfici, non sarà necessario aprire una nuova interfaccia. Infatti, è sufficiente cliccare sul pulsante in alto a destra, nella finestra, con la scritta **"SURFACES"** (sarà poi possibile anche tornare allo studio delle curve in maniera simile).



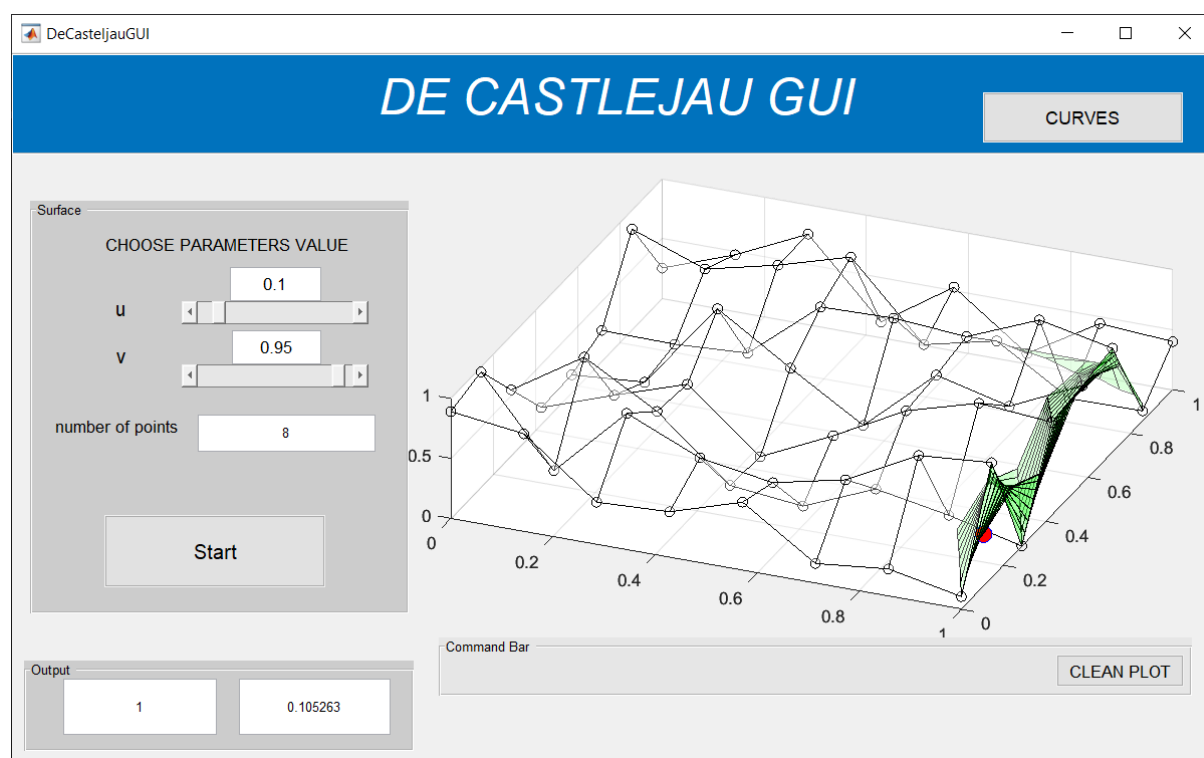
Non vi sono molte differenze nella configurazione dell'interfaccia, se non che è possibile solo inserire il numero di punti che si desidera selezionare e **non vi è una selezione diretta** come nel caso delle curve.

Questo poiché **non è possibile utilizzare una ginput in ambiente 3D**, o per lo meno non si è trovato la maniera per farlo.

Dunque, si è deciso di dare la possibilità all'**utente di scegliere il numero dei punti di controllo** che definiranno la superficie di Bézier, ma la loro posizione è predefinita. Infatti, con i punti di controllo viene creata una griglia con i punti equidistanti sulle assi X ed Y. D'altra parte, **la coordinata sull'asse Z viene assegnata randomicamente ad ogni punto**.

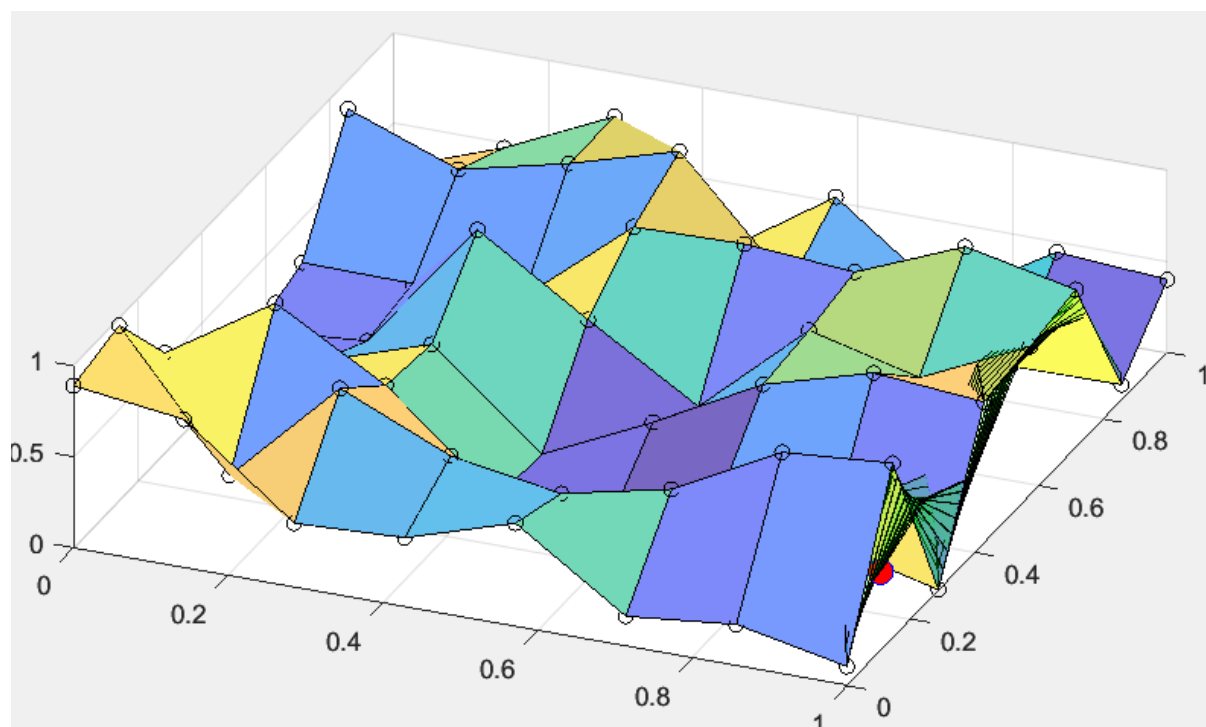


Una volta scelti i punti ed i valori dei due parametri, “ v ” ed “ u ” (compresi tra 0 e 1), si può applicare l’algoritmo di De Casteljau per le superfici. Il **punto valutato** grazie all’algoritmo è visibile nel plot con l’aspetto di un **punto rosso**. Inoltre, come nel caso delle curve, le sue coordinate sono visibili nella sezione in basso a sinistra, dove è presente il pannello “Output”.



In questo caso non compaiono i comandi per l’aggiunta, la cancellazione o la modifica di un punto, in quanto, trovandosi in un ambiente 3D, la modifica diretta dei punti non è di facile accesso.

Tuttavia, **rimane la funzionalità che consente di disegnare**, in questo caso, **la superficie di Bézier definita dai punti di controllo selezionati**.

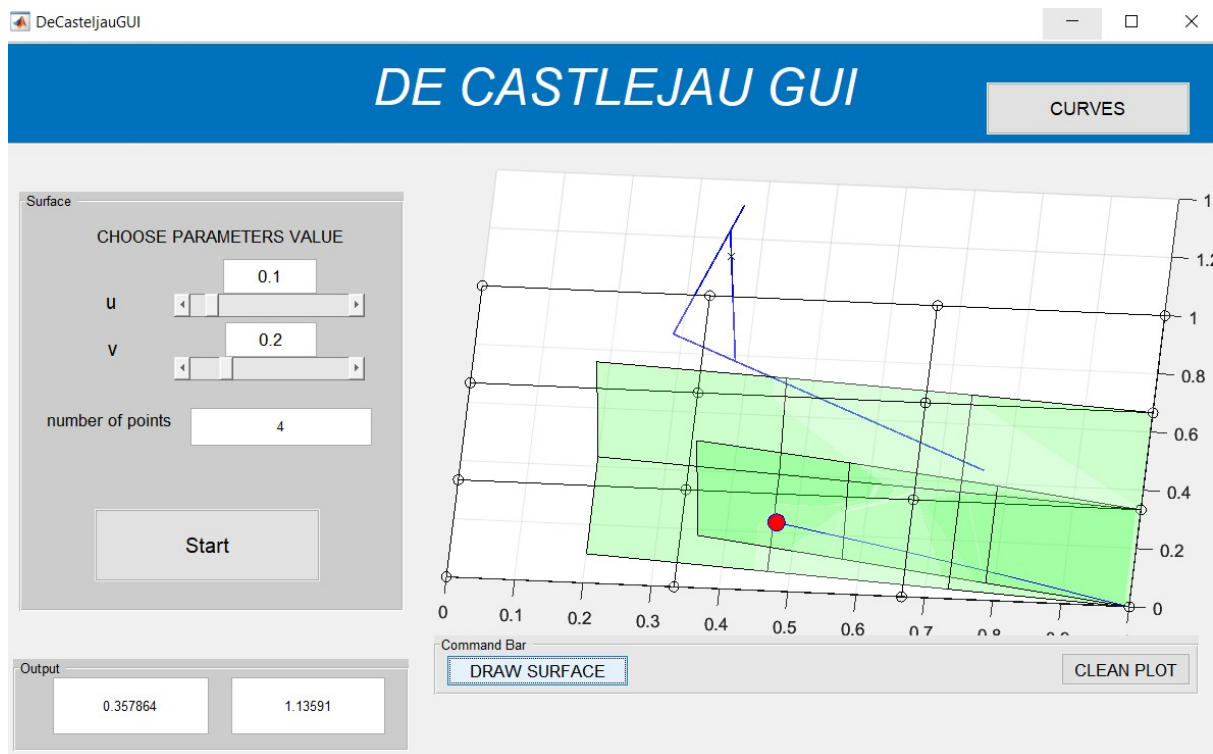


Overlapping delle figure

Un **problema** riscontrato all'interno dell'interfaccia è la possibilità che i **plot si sovrappongono tra loro**.

Questo è dovuto alla presenza di alcune **istruzioni "pause"** nel codice, che mettono in pausa l'esecuzione di un insieme di istruzioni, finché l'utente non clicca il tasto *"Invio"*.

Dunque, se il **flusso di una funzione dovesse rimanere in pausa e**, nel mentre, **un utente dovesse eseguire un'altra operazione**, si potrebbe verificare una situazione simile a quella presente nell'immagine sottostante.



Tuttavia, **questo problema dell'interfaccia è stato risolto**, tramite l'utilizzo di una **variabile globale** chiamata **"interruption"**.

Nel momento in cui viene **premuto un pulsante** dell'interfaccia, e *dunque eseguita una procedura*, questa **variabile viene posta a uno**.

Al termine della procedura, alla variabile viene **assegnato il valore zero**.

L'**overlapping dei plot può avvenire solo in presenza di una "pause"**, altrimenti l'esecuzione procede senza problemi.

Dunque, si è posto un **controllo dopo ogni pause**, per **verificare se è stata eseguita un'altra operazione** durante il periodo in cui il flusso della funzione è stato messo in pausa.

Se **non dovesse essere stato eseguito nulla**, allora l'**esecuzione procede senza problemi** fino al successivo step.

In **caso contrario**, il **flusso della funzione viene interrotto**.

Per fare questo, è sufficiente **verificare se, dopo l'esecuzione della pause, la variabile globale iteration è posta a zero o ad uno.**

Se dovesse essere posta ad uno, implicherebbe che non si è arrivati al termine di nessun'istruzione, e dunque non è stata eseguita alcuna operazione durante l'interruzione del flusso della funzione.

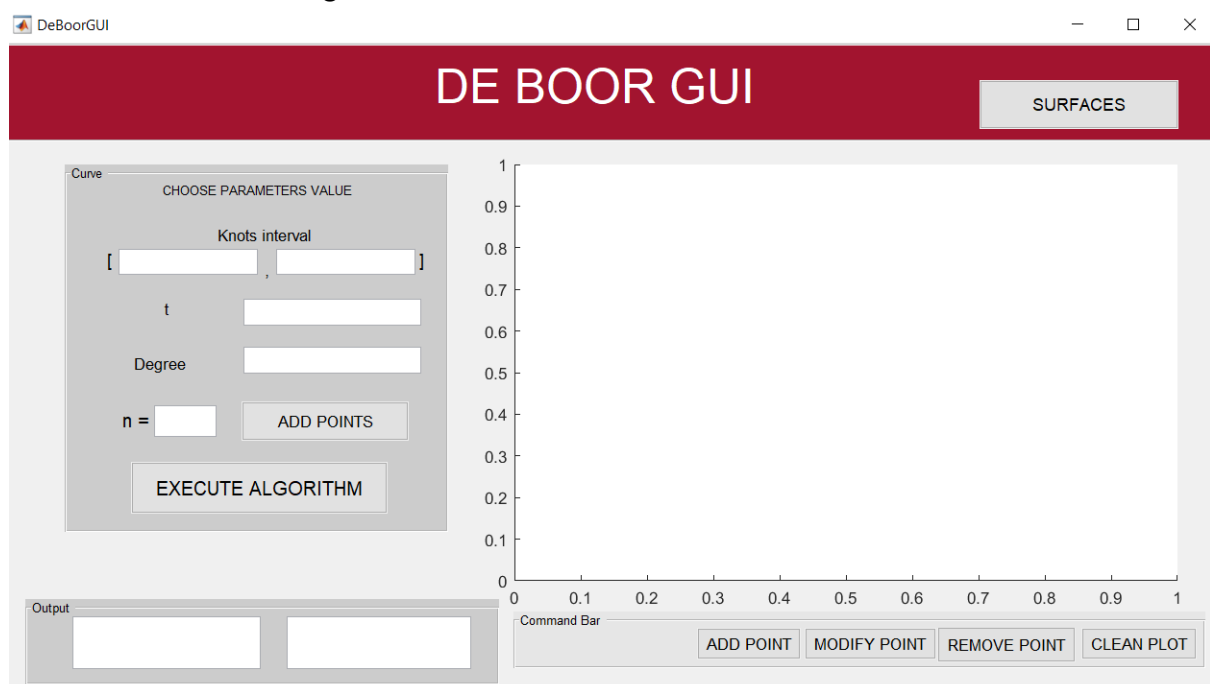
Al contrario, se dovesse essere pari a zero, significherebbe che un'altra operazione è stata eseguita, poiché iteration viene posto a zero solo al termine di tutte le funzioni relative alla gestione dei bottoni.

Interfaccia per l'algoritmo di De Boor

Fino ad ora, si è posto l'accento sull'interfaccia per l'applicazione dell'algoritmo di De Casteljau, senza considerare la seconda interfaccia implementata.

Tuttavia, le caratteristiche della seconda interfaccia, *che gestisce l'algoritmo di De Boor*, **sono molto simili a quelle già viste nelle precedenti pagine.**

Infatti, nonostante vi siano più parametri da definire, le funzionalità alla base dell'interfaccia rimangono le medesime.



La principale differenza si ritrova nel fatto che per De Boor è necessario **inserire un intervallo** per il **vettore dei nodi interni**, che verrà poi inserito nel vettore esteso dei nodi.

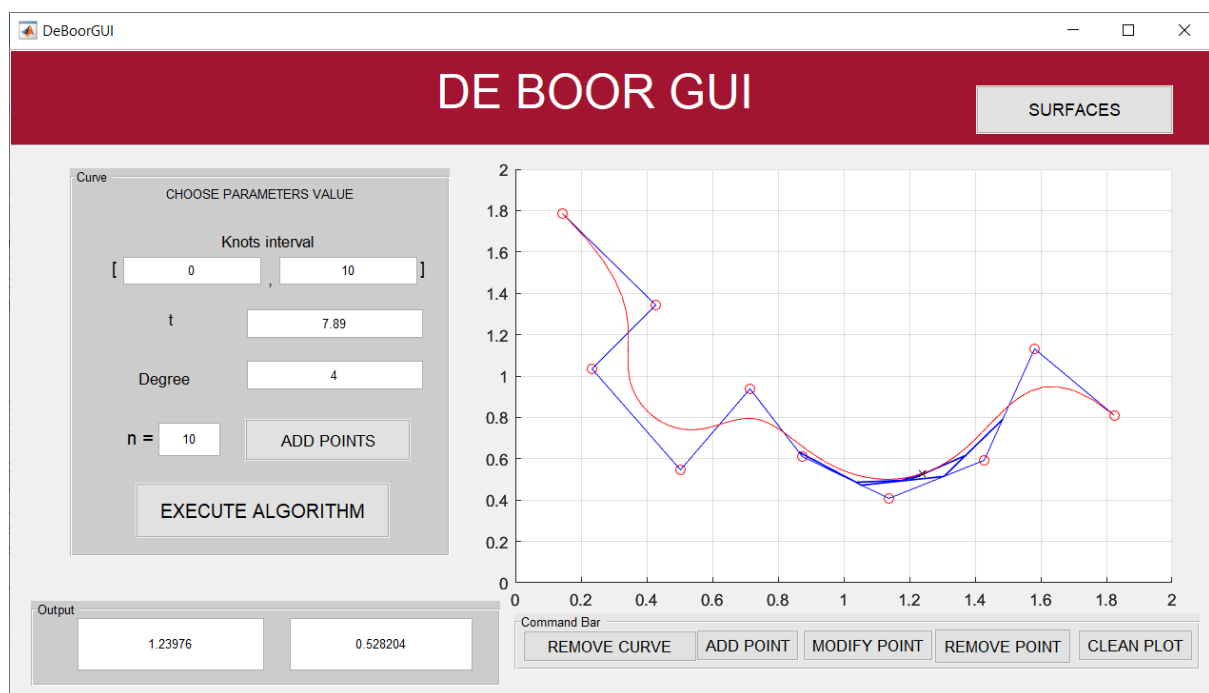
Infatti, in questo caso ci si sposta nel mondo delle **curve di B-Spline** ed è necessario definire tale vettore.

Per semplicità, il vettore **viene poi costruito con un semplice *linspace()***, prendendo dei nodi equidistanti definiti sull'intervallo passato in input.

In questo caso, il **parametro *t*** deve essere definito **all'interno dell'intervallo del vettore dei nodi interni, $[a, b]$** , per tale motivo non è stato possibile inserire uno slider, dato che l'intervallo viene passato in maniera dinamica dall'utente.

Per comodità, il **vettore esteso dei nodi è stato definito come un vettore *Clamped***.

Il risultato dell'applicazione dell'algoritmo è visibile nella seguente figura:



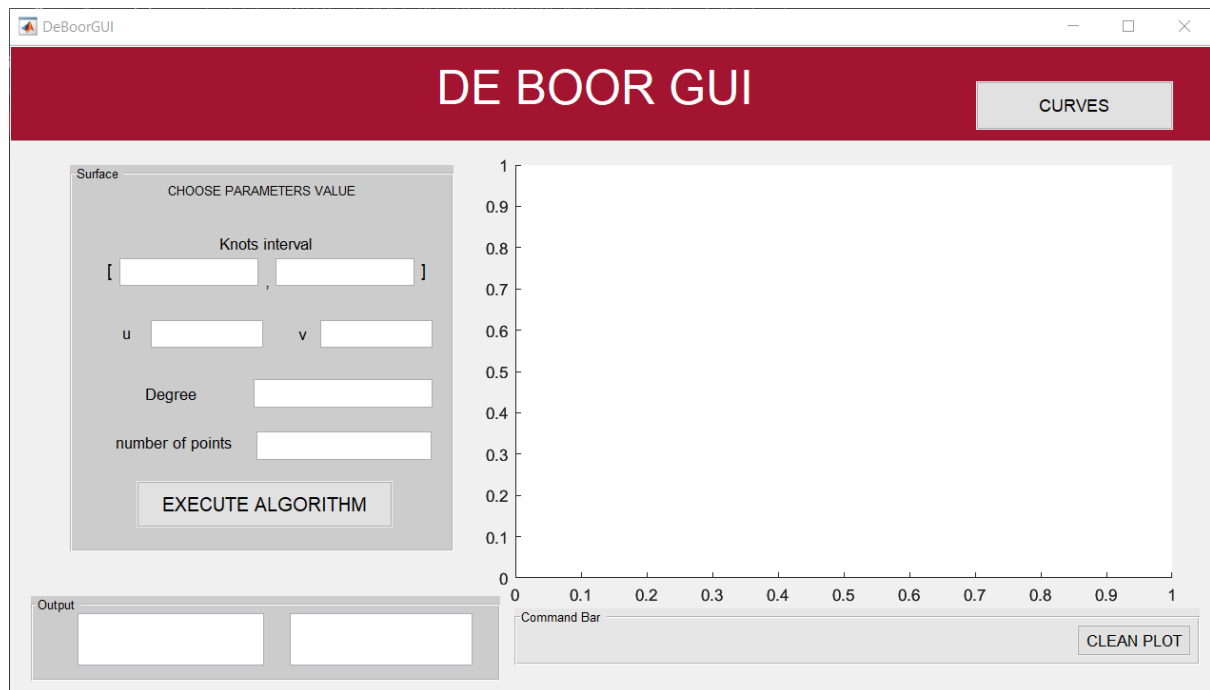
Come nel caso di De Casteljau, è possibile rimuovere la curva dal grafo e reinserirla.

Se si desidera resettare l'interfaccia e inserire nuovi punti di controllo, eliminando quelli fino ad ora selezionati, è sufficiente premere il tasto **“CLEAN PLOT”**. Una volta rimossi i punti non vi è modo di tornare indietro se non riselezionandoli manualmente.

Le Superfici (De Boor)

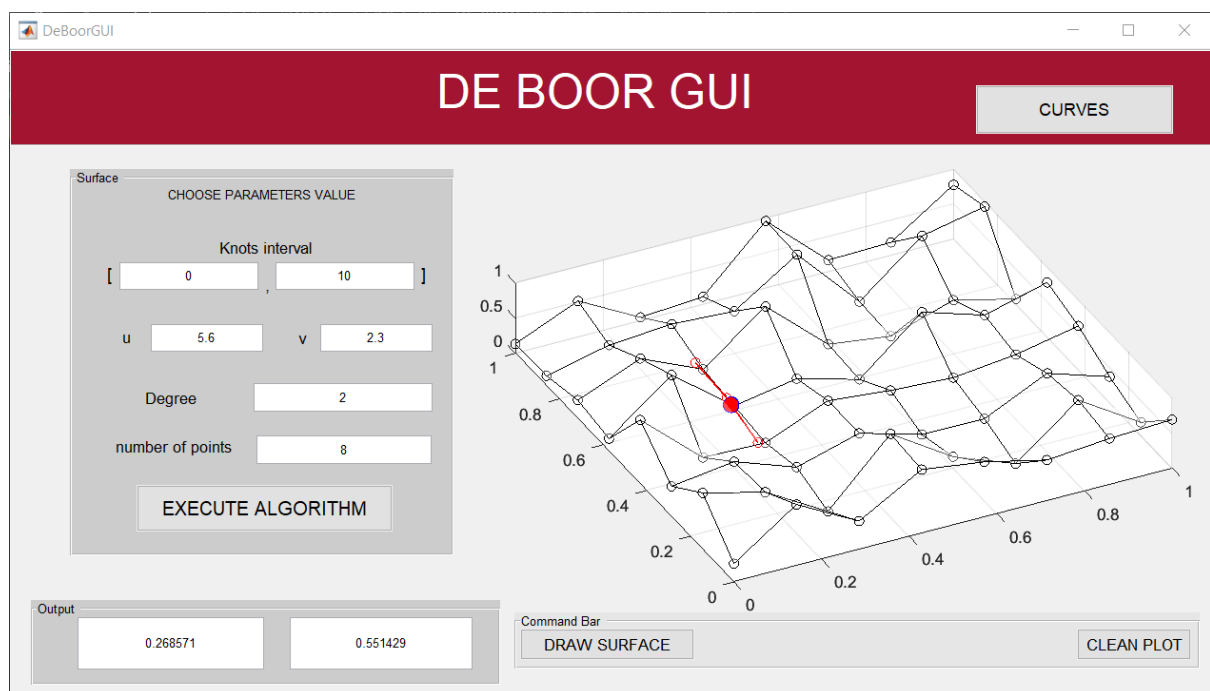
Il funzionamento dell'interfaccia per le **superfici nel caso dell'algoritmo di De Boor**, come per le curve, ha una struttura molto simile a quella vista per l'algoritmo di De Casteljau.

Differentemente alla precedente interfaccia, trattando **patch B-Spline** e non di Bezier, è necessario specificare **l'intervallo in cui sono compresi i nodi interni** del vettore esteso dei nodi ed il **grado della superficie**.



I restanti parametri e le restanti funzionalità sono equivalenti al caso di De Castlejau.

Cambia, ovviamente, l'algoritmo che viene applicato, trattandosi dell'algoritmo di De Boor.



Anche in questo caso è possibile disegnare la superficie, nonostante questo disegno non sia estremamente preciso.