IL PROBLEMA SSI NELLA MODELLAZIONE SOLIDA CON SUPERFICI NURBS

Nota di GIULIO CASCIOLA, SERENA MORIGI *

Presentata da ILIO GALLIGANI, Accademico Corrispondente Residente

nella seduta del 22 Febbraio 1995

Sommario

Abstract. This note presents a modified version of the marching method proposed in [1], for NURBS surface/surface intersection.
The balance between robustness and efficiency, an open question in SSI algorithms, is resolved by a clear trick. Sperimental analysis on the set of tolerances which control robustness, accuracy and efficiency shows a good behaviour of our proposal.

1. Introduzione

I più moderni sistemi di modellazione solida permettono di progettare, visualizzare ed analizzare al calcolatore oggetti complessi definiti, mediante rappresentazione boundary, da superfici sculturate (o a forma libera) prima di passare alla fase di produzione.
Uno dei problemi di ricerca nel progettare tali sistemi di modellazione solida è quello di estendere il dominio geometrico da superfici piane a tratti e quadriche a superfici a forma libera. In questo senso le curve e superfici NURBS (Non Uniform Rational B-Splines) sono a tutt'oggi lo strumento più idoneo per risolvere tale problema senza dover costruire un sistema ibrido; infatti essendo una generalizzazione delle spline hanno tutte le possibilità per generare una grande varietà di forme ed essendo delle razionali possono rappresentare esattamente le coniche e le quadriche.
Uno dei maggiori problemi computazionali nel progettare i sistemi di modellazione a forma libera consiste nella difficoltà di risolvere i problemi di intersezione fra superfici (problema SSI).

*Questa ricerca è stata finanziata dal CNR-Italy, Contratto n.94.00132.CT01

L'intersezione è l'operazione base per effettuare operazioni Booleane come unione, differenza e intersezione, che servono nella fase di modellazione di oggetti complessi costruiti a partire da semplici primitive.
In letteratura sono presenti molte proposte per il problema SSI [1, 2, 8]. Indipendentemente dall'approccio usato un buon algoritmo deve soddisfare tre proprietà: accuratezza, robustezza ed efficienza.
La curva di intersezione calcolata deve essere una buona approssimazione, nella tolleranza richiesta, della vera curva di intersezione.
L'algoritmo non deve essere soggetto a fallimenti e deve trovare tutte le porzioni di una curva di intersezione che può essere formata da più rami disgiunti.
Per la maggior parte delle applicazioni, l'algoritmo deve essere efficiente così da consentire una interazione in tempo reale fra il progettista ed il sistema.
Queste richieste presentano un serio dilemma. Un algoritmo robusto ed accurato può essere molto lento, mentre un algoritmo efficiente può fallire in certi casi. Nessuno degli algoritmi noti in letteratura soddisfa tutte e tre queste proprietà. Affinchè un algoritmo sia di uso pratico si devono introdurre significativi compromessi fra le sue proprietà.
In questo lavoro si vuole presentare una versione modificata della proposta di Barnhill ed altri [1] per l'intersezione di superfici in forma parametrica, per l'intersezione di superfici NURBS.
Il compromesso robustezza-efficienza viene qui risolto introducendo un particolare accorgimento nell'algoritmo che garantisce la massima robustezza senza perdere in efficienza a fronte di una piccola limitazione di applicabilità del metodo.

2. Superfici NURBS trimmate

Una superficie spline relativa ad una partizione non uniforme di nodi del dominio $U \times V = [0,1] \times [0,1]$ può essere definita come prodotto tensoriale di funzioni spline e rappresentata da

$$s(u,v) = \sum_{i=1}^{n+H} \sum_{j=1}^{m+K} c_{ij} N_{i,n}(u) N_{j,m}(v)$$

dove $N_{i,n}(u)$ e $N_{j,m}(v)$ sono le B-spline Normalizzate di ordine n ed m rispettivamente definite sui vettori dei nodi:

$$(0, ..., 0, u_{n+1}, ..., u_{n+H}, 1, ..., 1)$$

$$(0, ..., 0, v_{m+1}, ..., v_{m+K}, 1, ..., 1).$$

con nodi agli estremi aventi molteplicità n ed m rispettivamente.

Una superficie NURBS viene definita come il rapporto fra superfici spline appartenenti allo stesso spazio prodotto tensoriale; una superficie NURBS in forma parametrica è data nella forma seguente:

$$\mathbf{r}(u,v) = \frac{\sum_{i=1}^{n+H}\sum_{j=1}^{m+K} w_{ij}\mathbf{P}_{ij}N_{i,n}(u)N_{j,m}(v)}{\sum_{i=1}^{n+H}\sum_{j=1}^{m+K} w_{ij}N_{i,n}(u)N_{j,m}(v)} \tag{1}$$

dove i w_{ij} sono numeri reali positivi detti pesi, e i $\mathbf{P}_{ij}$ punti 3D di controllo che definiscono una griglia detta di controllo.
Nonostante sia ottenuta a partire da superfici prodotto tensoriale, una superficie NURBS non è una superficie prodotto tensoriale.
Introducendo le funzioni B-spline razionali normalizzate

$$R_{i,n;j,m}(u,v) = \frac{w_{ij}N_{i,n}(u)N_{j,m}(v)}{\sum_{i=1}^{n+H}\sum_{j=1}^{m+K} w_{ij}N_{i,n}(u)N_{j,m}(v)}$$

la (1) può essere scritta nella forma:

$$\mathbf{r}(u,v) = \sum_{i=1}^{n+H}\sum_{j=1}^{m+K} w_{ij}\mathbf{P}_{ij}R_{i,n;j,m}(u,v)$$

Le superfici così definite rappresentano uno schema bidirezionale di curve 3D NURBS; se infatti si fissa uno dei parametri, per esempio $u_0 \in [0,1]$, allora permettendo a v di variare in $[0,1]$ si ottiene una sezione della superficie che altro non è che una curva 3D NURBS, cioè

$$\mathbf{c}(v) = \mathbf{r}(u_0,v) = \sum_{j=1}^{m+K} \mathbf{Q}_j(u_0)R_{j,m}(v)$$

I $\mathbf{Q}_j(u_0)$ sono gli $m+K$ punti di controllo 3D per la curva $\mathbf{c}(v)$ [5].
Si definisce *trimmed surface* la restrizione di $\mathbf{r}(u,v)$ ad un sottodominio $D \subset U \times V$, dello spazio parametrico, chiamato *trimming region*. Il dominio D è definito come l'insieme di regioni di $U \times V$ limitate da rami di curve chiuse orientate e non intersecantesi, dette *trimming curve*.

3. Solidi NURBS

Seguendo la rappresentazione boundary di un solido, si può definire *solido primitivo* ogni solido avente come boundary una sola superficie NURBS che banalmente deve essere chiusa per dividere lo spazio in due parti una delle quali limitata.
Tra i solidi primitivi che rispondono a questa definizione si trovano per esempio sfere, tori, superfici chiuse di rotazione e superfici chiuse a forma libera.

I solidi primitivi possono poi essere composti mediante operazioni booleane, quali intersezione, unione e differenza, per dar luogo ad un *solido composto*.
Il costo e le difficoltà incontrate nella combinazione booleana di solidi primitivi a forma libera sono state in parte superate grazie all'introduzione delle trimmed surface che permettono una rappresentazione completa del boundary di un solido attraverso l'unione di superfici ristrette ad opportuni domini. Il boundary di un solido composto S è quindi

$$bS = \bigcup_{i=1}^{k} \mathbf{r}_i(D_i)$$

dove D_i è la trimming region associata alla superficie $\mathbf{r}_i$
Il boundary del solido composto è nuovamente una superficie chiusa.
Dati due solidi A e B definiti dalle rispettive superfici di bordo, il problema della composizione consiste nel determinare i bordi dei solidi $b(A \bigcup B)$, $b(A \bigcap B)$ e $b(A-B)$ a partire da

$$bA = \bigcup_{i=1}^{k} \mathbf{r}_i(D_i) \; e \; bB = \bigcup_{j=1}^{h} \mathbf{s}_j(E_j)$$

dove $\mathbf{r}_i$ ed $\mathbf{s}_j$ sono trimmed surface definite sulle trimming region D_i e E_j.
Alla base di tali operazioni vi è l'intersezione fra trimmed surface che risulta un problema molto complicato; seguendo le proposte di [4] e [6] tale problema può essere riformulato eseguendo l'intersezione solo tra le superfici non trimmate (definite cioè sull'intero dominio parametrico).

4. Il problema SSI

Tra i diversi metodi proposti in letteratura per la risoluzione del problema SSI, i metodi geometrico e numerico sono da considerarsi le tecniche che presentano maggior interesse da un punto di vista computazionale. Il metodo geometrico, robusto ma lento, è basato sulla decomposizione ricorsiva del problema in problemi più semplici fino alla risoluzione di intersezioni di sottosuperfici elementari (triangolo/piano). Mentre alla base del metodo numerico è la generazione di una sequenza di starting point, almeno uno per ogni ramo di curva di intersezione, dai quali risalire alle curve richieste che saranno poi raccordate nella fase di ordinamento. L'approccio numerico si avvale di tecniche iterative, è veloce ma meno affidabile.
Nell'ottica di trarre vantaggio dalla robustezza del metodo geometrico e nel contempo mantenere l'efficienza del metodo numerico, nasce il metodo ibrido. Tecniche geometriche vengono utilizzate per localizzare gli starting point che sono poi raffinati mediante tecniche numeriche fino ad ottenere le esatte curve di intersezione.

Seguendo la proposta di Barnhill ed altri [1], si vuole mettere a punto un algoritmo ibrido che si avvalga delle potenzialità delle superfici NURBS per ottenere significativi risultati.
Il risultato dell'intersezione di due superfici può essere: 1) un insieme vuoto, 2) un insieme di punti, 3) un insieme di curve, 4) un insieme di superfici o 5) una combinazione di 2), 3) e 4).
Due superfici sono dette essere in posizione generale se la loro intersezione è di tipo 1) o 3). L'algoritmo proposto nel presente lavoro tratta solo superfici in posizione generale che vengono approssimate da curve lineari a tratti su domini parametrici.
Per determinare un giusto compromesso tra accuratezza, robustezza ed efficienza, si prendono in considerazione i seguenti parametri di tolleranza:
1. Same Point Tolerance (SPT): se due punti 3D sono ad una distanza inferiore ad SPT allora sono da considerarsi un unico punto;
2. Search Refinement Tolerance (SRT): regola la qualità della suddivisione adattiva della superficie;
3. Curve Refinement Tolerance (CRT): se due punti appartenenti ad una curva di intersezione sono ad una distanza inferiore a CRT, il segmento che li unisce è preso come approssimazione del tratto di curva che li separa.
Prima di passare ad una descrizione dettagliata dell'algoritmo se ne illustrano brevemente le quattro fasi principali. La prima fase (Generazione griglia adattiva) consiste nella approssimazione di ciascuna delle superfici mediante una griglia adattiva di curve isoparametriche approssimate da lineari a tratti seguendo la tolleranza SRT. Da una tale grigliatura si può ottenere banalmente una triangolarizzazione della superficie. Nella seconda fase (Generazione starting point) si utilizzano intersezioni elementari (segmento di una superficie/triangolo dell'altra) per ottenere almeno uno starting point per ciascuna curva di intersezione. A partire da questi punti il passo successivo (Generazione curve di intersezione) si propone di determinare gli altri punti sulle varie curve di intersezione. Nella fase conclusiva (Ordinamento) si procede al raccordo dei rami di curva trovati fino alla determinazione delle curve di intersezione 3D.

5. Generazione griglia adattiva

Per poter ricondurre il problema SSI ad un insieme di intersezioni fra triangoli e segmenti, si devono approssimare le due superfici curve con triangoli piani. Questo si ottiene mediante un procedimento ricorsivo che, a partire da un insieme di curve isoparametriche (in genere tre) in entrambe le direzioni u e v, suddivide ogni tratto di curva nel punto medio, nel caso la distanza tra il tratto di curva in esame e il segmento sotteso superi la tolleranza SRT. Dalla grigliatura rettangolare ottenuta si passa facilmente ad una triangolarizzazione inserendo le diagonali. Per rendere meno onerosa questa fase si sono sfruttate alcune caratteristiche delle curve NURBS.

Il problema del controllo della linearità di un tratto di curva NURBS rispetto ad una tolleranza SRT, si può infatti ricondurre, per la proprietà del guscio convesso di cui godono le curve NURBS, ad un più semplice test sulla distanza massima tra i punti di controllo ed il segmento rettilineo congiungente gli estremi. Per poter riapplicare il test sulla distanza massima ai singoli tratti di curva si sfrutta la possibilità di suddividere una curva in due curve NURBS su intervalli parametrici adiacenti utilizzando la tecnica della knot-insertion per inserire nel punto medio un nodo $\hat{u}$ (o $\hat{v}$) di molteplicità pari all'ordine della spline. Il procedimento iterato porta alla convergenza del poligono di controllo alla curva.

6. Generazione Starting Point

Alla fase di approssimazione lineare delle superfici segue la ricerca dei punti di intersezione iniziali detti starting point dai quali partire, nella fase successiva, per determinare le curve di intersezione di cui fanno parte. Una prima approssimazione degli starting point si calcola dalle intersezioni tra le approssimazioni delle curve isoparametriche della prima superficie e la approssimazione bivariata lineare a tratti della seconda superficie. Se l'intersezione non è vuota si ottiene un punto che, in generale, non appartiene ad alcuna delle superfici poichè generato dall'intersezione delle loro approssimazioni. I due valori nei domini parametrici relativi al punto trovato, individuano due punti, uno su ciascuna superficie, vicini ad un vero punto di intersezione. Quest'ultimo è calcolato con l'ausilio del metodo iterativo di Newton-Raphson a partire dai due punti nei domini che vanno a costituire l'iterato iniziale. Il metodo risolve il sistema di tre equazioni non lineari in quattro incognite

$$\mathbf{r}_2(u,v) - \mathbf{r}_1(s,t) = \mathbf{0} \tag{2}$$

dove viene fissata l'incognita relativa alla linea parametrica in esame.
Se il metodo non converge si suddividono ulteriormente entrambe le approssimazioni considerate per migliorare l'iterato iniziale e si riapplica il procedimento finchè, o non si ha più l'intersezione, o si ottiene la convergenza ad uno starting point; questo può poi essere trascurato se risulta contenuto in una curva già trovata. Per trovare tutti gli starting point è necessario considerare le intersezioni con tutte le linee isoparametriche della prima superficie quindi scambiare i ruoli delle due superfici.
La fase di generazione degli starting point è la fase più onerosa dell'algoritmo; ciò è dovuto in particolare alle numerose intersezioni segmento/triangolo la maggior parte delle quali non hanno successo. Per contenere il numero delle mancate intersezioni è necessario eliminare a priori le coppie segmento/triangolo che non presentano intersezione.
Un primo test di esclusione suggerito e adottato in questa implementazione è basato sull'utilizzo di bounding box, costruite facilmente a partire dai punti di controllo delle intere superfici, per individuare, se esiste, una bounding box di intersezione e

considerare solo quei segmenti che hanno almeno uno degli estremi in essa.
Un secondo test, basato sulle bounding box delle coppie segmento/triangolo, evita inutili intersezioni nel caso di bounding box disgiunte.
Un nuovo ed ulteriore accorgimento è dato dall'eliminazione delle coppie segmento/triangolo (primo dominio/secondo dominio) se il segmento è attraversato da una curva di intersezione già trovata e il triangolo è quello che ha permesso di calcolare lo starting point. Vengono eliminati inoltre i segmenti costituenti il triangolo qualora accoppiati a triangoli nel primo dominio attraversati da una curva. Quest'ultimo 'filtro' impone però l'aggiunta di una ipotesi sulle curve di intersezione per mantenere la robustezza dell'algoritmo. Condizione necessaria per trovare tutti i rami di una curva è che ognuno di questi si differenzi dagli altri in almeno un dominio parametrico, per almeno un segmento attraversato; in questo modo sarà possibile trovare un altro starting point su quella curva. Il problema di perdita di un ramo può essere risolto diminuendo la tolleranza SRT per ottenere una approssimazione più accurata. Come si farà vedere nel seguito una piccola tolleranza SRT non comporta, grazie a quest'ultimo accorgimento, un aggravio dal punto di vista computazionale mentre risulta essenziale per la robustezza del metodo.

7. Generazione curve di intersezione

Trovato uno starting point, si costruisce per passi la curva di intersezione a cui questo appartiene. Si cerca inizialmente una stima del punto successivo che viene poi raffinata per determinare un vero punto di intersezione; quindi si raffina il segmento che unisce i due punti fino a raggiungere una distanza CRT tra due punti consecutivi ed infine si ottimizza il risultato.
Cominciamo col determinare un nuovo punto sulla curva; chiameremo il punto corrente sulla curva di intersezione $\mathbf{P}_0$ e consideriamo un vettore $\mathbf{V}$ nella direzione tangenziale alla curva di intersezione per ottenere un punto $\mathbf{P}_1$.
Nel caso generale, $\mathbf{V}$ si può determinare intersecando i piani tangenti alle due superfici in $\mathbf{P}_0$ e scalando questo vettore per essere di lunghezza proporzionale a CRT. Poi, mediante inversione dello Jacobiano in $\mathbf{P}_0$, si ottengono i punti corrispondenti di $\mathbf{P}_1$ nel dominio parametrico di ciascuna superficie (s_1, t_1, u_1, v_1). Si useranno questi punti nei domini come iterati iniziali per un procedimento iterativo superficie/superficie che cerca di trovare un punto vero di intersezione.
La routine iterativa superficie/superficie risolve le tre equazioni non lineari (2) nelle quattro incognite s, t, u, v. Si può utilizzare la forma linearizzata di (2), che può essere interpretata geometricamente come l'intersezione fra il piano tangente ad $\mathbf{r}_1$ in (s_1, t_1) e il piano tangente ad $\mathbf{r}_2$ in (u_1, v_1). Il successivo iterato viene scelto come il punto medio M_2 delle proiezioni ortogonali A_1 di $\mathbf{r}_1(s_1, t_1)$ e B_1 di $\mathbf{r}_2(u_1, v_1)$ su questa linea di intersezione (vedi Figura 1).
Invertendo nuovamente lo Jacobiano al precedente iterato si ottiene il nuovo ite-

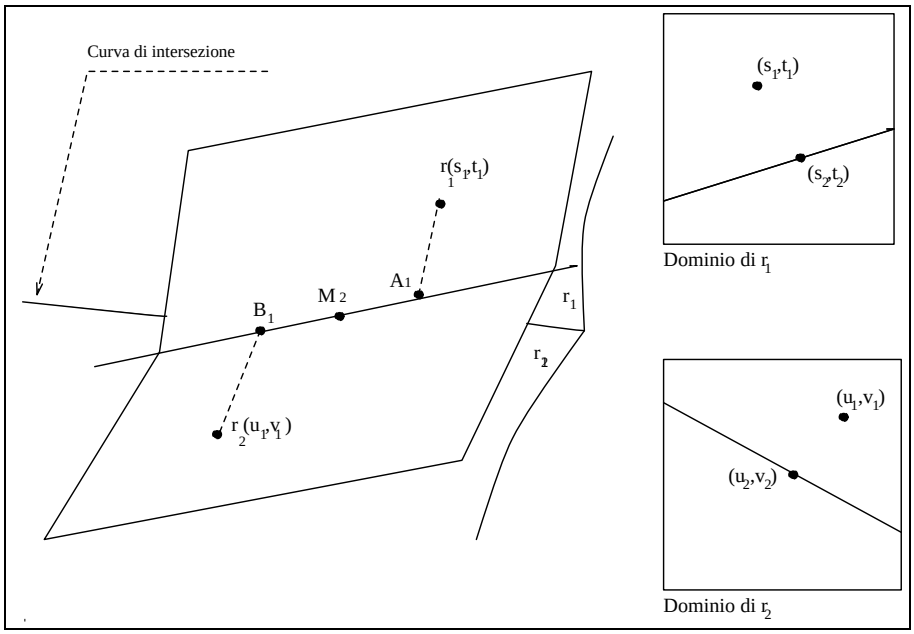

Figura 1: Significato geometrico del metodo

rato (s_2, t_2, u_2, v_2) su ciascun dominio (si rimanda all'Appendice per la descrizione matematica di queste fasi).
Una volta trovata una nuova vera intersezione, chiamiamola ancora $\mathbf{P}_1$, si considera l'intervallo fra $\mathbf{P}_0$ e $\mathbf{P}_1$. Se questo segmento è più lungo di CRT si raffina ed ottimizza l'approssimazione alla curva di intersezione.
L'algoritmo termina la ricerca di ulteriori punti sulla curva di intersezione per uno dei seguenti motivi:
1. la curva incontra un bordo o incontra se stessa;
2. la curva incontra un punto finale di una curva precedentemente trovata;
3. la fase iterativa non ha trovato un punto successivo;
4. la curva incontra una intersezione fra una linea isoparametrica ed una curva precedentemente trovata.

Al termine di questa fase si ottiene una lista dei rami di curve 3D trovati e relative coordinate parametriche nei rispettivi domini, che verranno ordinati e raccordati nella fase successiva.
Durante la fase di tracciamento della curva può capitare che il punto di intersezione trovato si trovi fuori dal dominio parametrico. E' necessario in questo caso seguire una strategia di estrapolazione dei punti sul bordo. Questi punti infatti risultano fondamentali per effettuare operazioni booleane fra le superfici intersecate. L'estrapolazione determina un punto sul bordo del dominio parametrico come l'intersezione fra la retta congiungente l'ultimo e il penultimo dei punti della curva e la linea isoparametrica di bordo. Dai due punti trovati, uno su ciascuna super-

ficie, si ricerca il vero punto sulla curva di intersezione mediante la procedura di Newton-Raphson (curva/superficie). Il successo del procedimento è strettamente legato alla tolleranza CRT poichè l'andamento della curva è stimato attraverso la retta congiungente gli ultimi punti trovati.

8. Ordinamento

Trovati tutti i rami di curva si procede al loro ordinamento e raccordo al fine di ottenere curve chiuse.
L'algoritmo considera rami di curve nello spazio, facendo coincidere in 3D, attraverso le funzioni corrispondenti alle due superfici, i rami di intersezione trovati sui diversi domini parametrici.
La connessione di due rami di curva avviene quando i due estremi sono a distanza minore di CRT nello spazio o quando gli estremi si trovano su bordi opposti nel dominio parametrico. Un punto isolato non è considerato curva chiusa ed è così raccordato ad un vicino ramo di curva.

9. Analisi sperimentale delle tolleranze SRT e CRT

Gli esempi presentati vogliono essere un'analisi dei parametri di tolleranza SRT e CRT che influenzano le proprietà di robustezza, accuratezza e velocità dell' algoritmo. Obiettivo della sperimentazione effettuata è quindi quantificare il concetto di compromesso tra queste proprietà, in termini dei valori da associare ai due parametri per ottenerlo grazie anche all'introduzione di opportuni accorgimenti implementativi.
Le superfici test utilizzate provengono dalla modellazione solida e sono tutte boundary di solidi primitivi, come definito nel paragrafo 3. Vengono prese in considerazione intersezioni fra superfici sculturate, fra superfici naturali (quadriche e tori) e fra sculturate e naturali. Le Figure 2-8 mostrano le intersezioni test (vengono riportate anche quelle nascoste), le superfici che le generano (rappresentate con eliminazione delle parti nascoste) e le curve di intersezione nei rispettivi domini parametrici insieme alla grigliatura ottenuta con l'SRT utilizzato (i segmenti di griglia attraversati dalle curve sono stati cancellati seguendo la tecnica dei segmenti annullati introdotta). La Figura 9 mostra il risultato dell'algoritmo per un problema di intersezione fra più superfici. I tempi presentati sono relativi alle quattro fasi precedentemente descritte e sono rilevati in millesecondi. L'elaborazione è stata realizzata su una workstation Silicon Graphics IRIS Indigo XZ.
La tolleranza SRT guida la suddivisione adattiva della superficie controllando la qualità dell'approssimazione lineare iniziale e influenzando di conseguenza il numero di intersezioni per la ricerca di starting point sulla curva di intersezione. L'SRT deve essere quindi tale da garantire almeno uno starting point per ogni ramo di curva affinchè la successiva fase di tracciamento della curva possa aver successo. La scelta

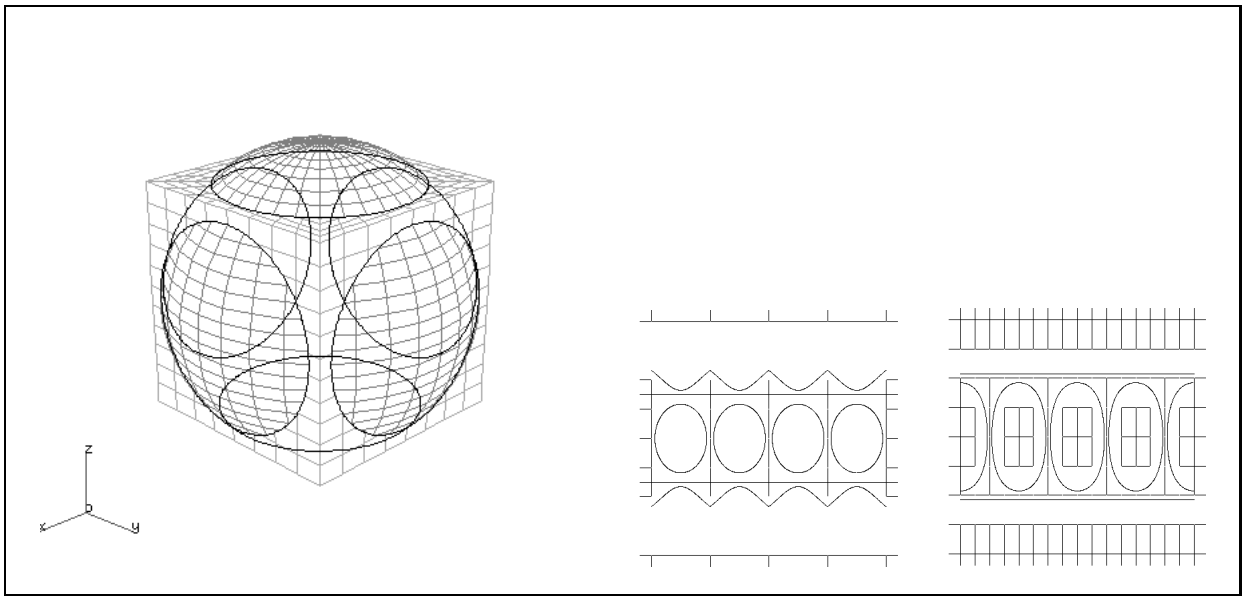

Figura 2: Intersezione fra cubo (SRT=0.12) e sfera (SRT=0.12) con CRT=0.01. A sinistra dominio cubo, a destra dominio sfera

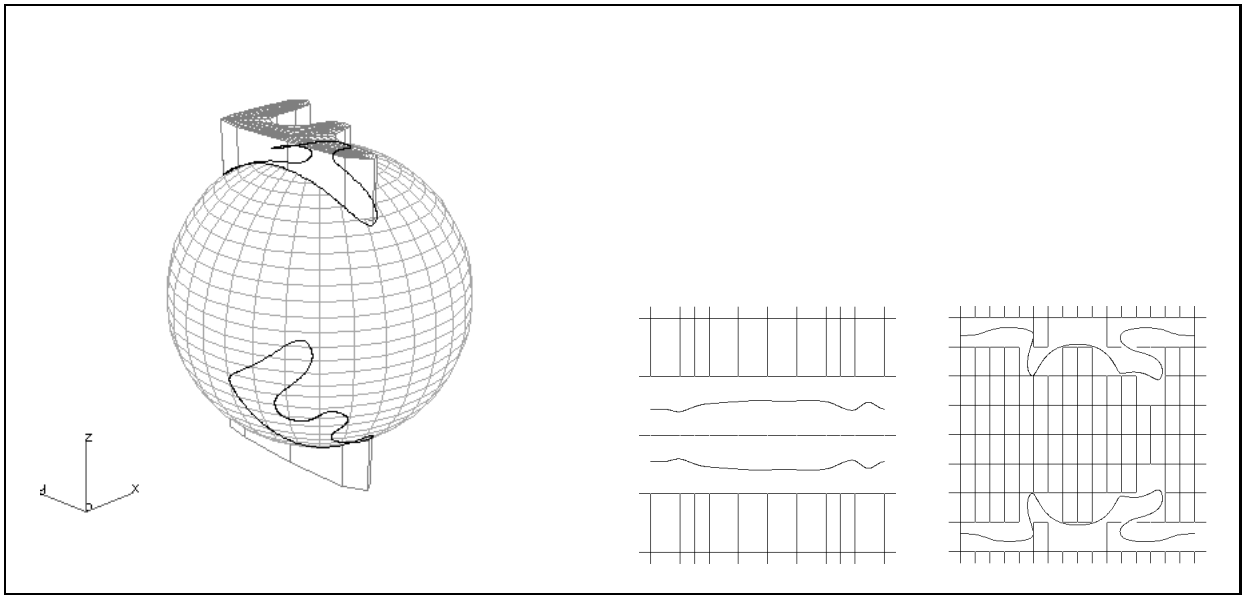

Figura 3: Intersezione fra una superficie a forma libera (SRT=0.12) e una sfera (SRT=0.12); CRT=0.01. A sinistra dominio della superficie a forma libera, a destra dominio della sfera.

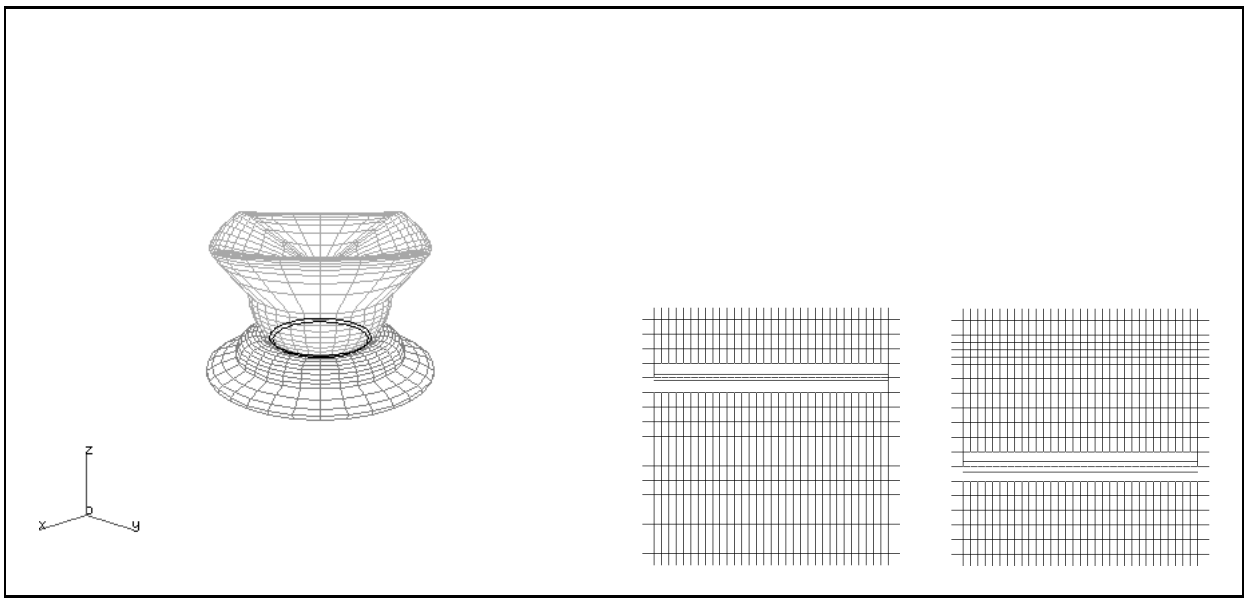

Figura 4: Intersezione fra superfici a forma libera (entrambe con SRT=0.02); CRT=0.02. A sinistra dominio della superficie in basso, a destra dominio della superficie in alto.

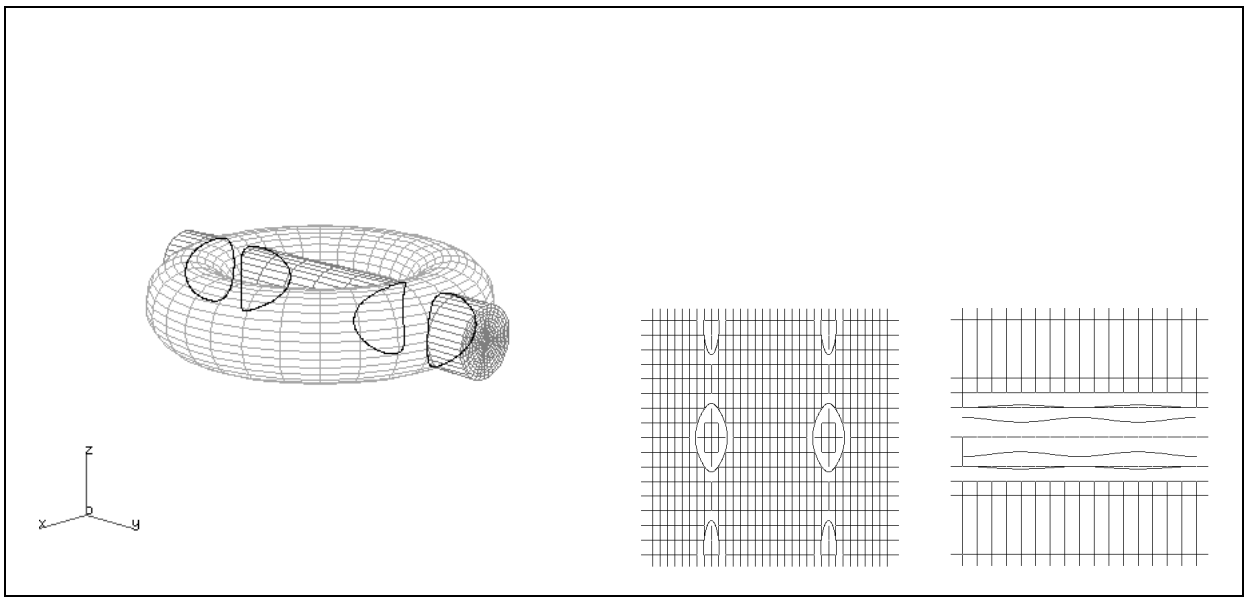

Figura 5: Intersezione fra toro (SRT=0.03) e cilindro (SRT=0.03) con CRT=0.01. A sinistra dominio toro, a destra dominio cilindro.

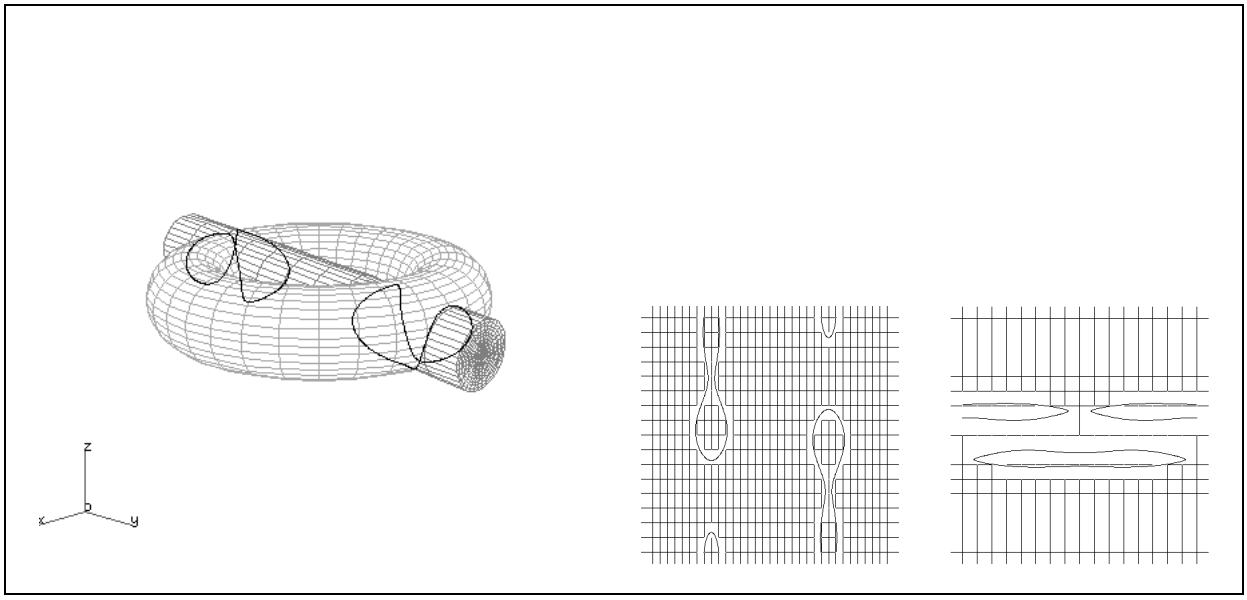

Figura 6: Intersezione fra toro (SRT=0.02) e cilindro (SRT=0.02) con CRT=0.01. A sinistra dominio toro, a destra dominio cilindro.

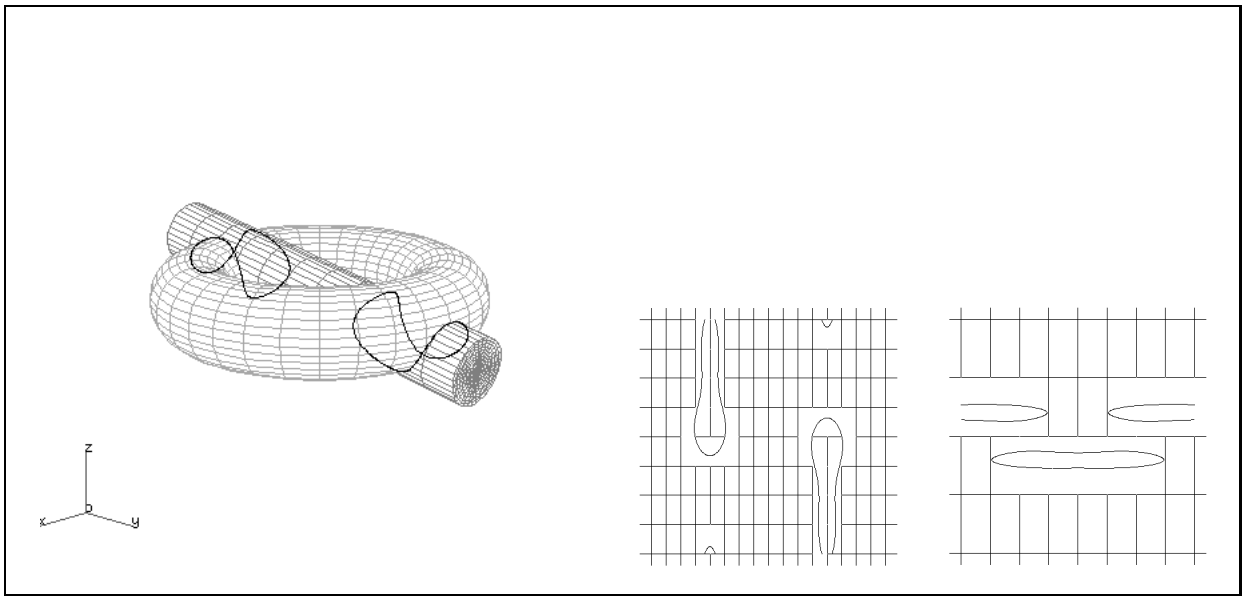

Figura 7: Intersezione fra toro (SRT=0.12) e cilindro (SRT=0.12) con CRT=0.01. A sinistra dominio toro, a destra dominio cilindro.

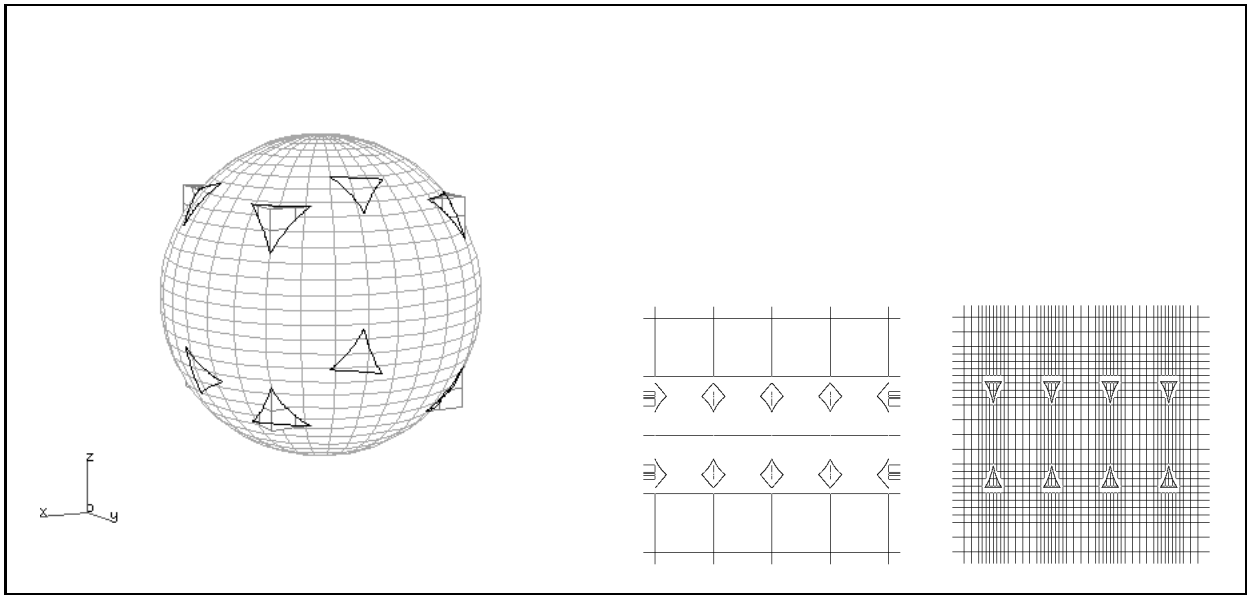

Figura 8: Intersezione fra cubo (SRT=0.01) e sfera (SRT=0.01) con CRT=0.003. A sinistra dominio cubo, a destra dominio sfera.

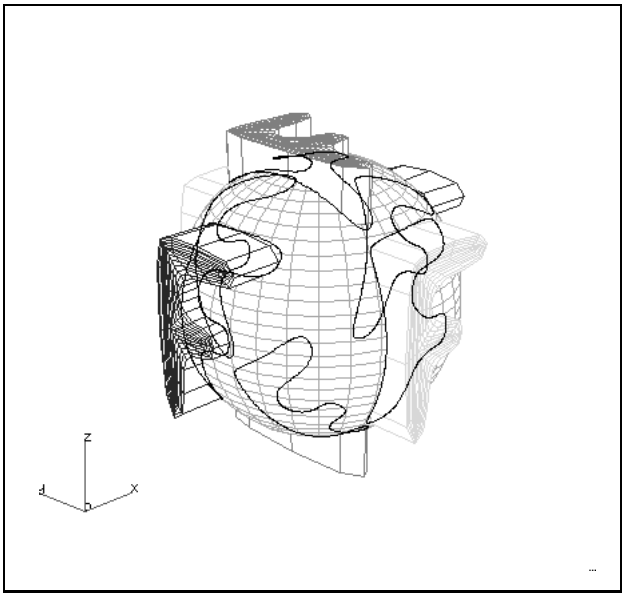

Figura 9: Intersezione multipla fra tre superfici a forma libera (SRT=0.12) e una sfera (SRT=0.12); CRT=0.01.

di CRT, infine, influenzando il numero di punti per ramo di curva determina il successo e l'efficienza della fase di tracing.
Allo stato dell'arte non esiste alcun metodo per calcolare a priori l'esatta combinazione di entrambi i parametri che possa rispondere contemporaneamente alle richieste di efficienza, robustezza e accuratezza, nè esiste una stima degli stessi che possa garantire anche solo il ritrovamento di tutte le soluzioni.
Poichè i due parametri in analisi, pur essendo dipendenti l'uno dall' altro, influenzano fasi diverse dell'algoritmo, si è preferito prenderli in considerazione separatamente mantenendone costante uno e considerando variazioni dell'altro a partire da una opportuna stima iniziale. La tolleranza SPT rimane invece costante ad un valore pari a 1.0e-5.

9.1. Analisi della tolleranza SRT

L'impossibilità di avere un parametro SRT che assicuri il ritrovamento di tutti i rami di curva in ogni situazione spinge alla ricerca di una stima consistente con le superfici in oggetto, ma poco costosa in termini computazionali. Poichè il parametro SRT deve variare in funzione della dimensione della superficie, nell'implementazione da noi realizzata si usa un valore pari ad 1/10 del raggio della sfera contenente il poliedro di controllo della superficie che deriva dalla considerazione di prendere come riferimento una superficie sferica. Queste stime iniziali per SRT si sono dimostrate sperimentalmente valide in quanto segnano, nella maggioranza dei casi, la soglia massima oltre la quale non si riescono più a determinare tutti gli starting point necessari per ritrovare tutti i rami di curva. Nei casi in cui una particolare curvatura delle superfici o la loro posizione relativa siano tali da portare all'insuccesso della stima iniziale, può essere necessario ridurre l'SRT fino ad 1/10 della stima iniziale eseguendo una serie di tentativi intermedi. Trattando solo superfici chiuse è infatti possibile verificare l'insuccesso dell'algoritmo e innescare di conseguenza una procedura iterativa automatica basata sul parametro SRT.
Come già in precedenza rimarcato un piccolo SRT porta ad alti costi delle fasi di suddivisione adattiva e ricerca degli starting point. Per superare questo problema e poter quindi volutamente spingere sull'SRT senza preoccuparsi del peggioramento della prestazioni, sono stati introdotti degli accorgimenti per ottimizzare l'algoritmo. L'ottimizzazione ha permesso di eliminare il calcolo degli starting point superflui tramite il meccanismo dei segmenti annullati, il controllo preliminare mediante bounding box della superficie di intersezione e bounding box delle intersezioni elementari segmento/triangolo. Si può quindi diminuire il valore di SRT ottenendo suddivisioni più raffinate senza che questo comporti un decadimento delle prestazioni. I risultati in Tabella 1 e 2 relativi agli esempi in Figura 2 e 3, evidenziano l'aumento del costo delle fasi di splitting e ricerca starting point in relazione al tempo totale dell'algoritmo al variare del parametro SRT dal valore consigliato ad 1/10 dello

SRT	$TEMPI(msec)$					$GRIGLIA$
	$Splitting$	$St.Point$	$Tracing$	$Sorting$	$Totale$	
0.12	60	40	2180	120	2400	9×17 9×5
0.10	60	40	2180	120	2400	9×17 9×5
0.08	60	40	2180	120	2400	9×17 9×5
0.06	60	50	2290	120	2520	9×17 11×5
0.04	130	120	2340	130	2400	13×25 11×5
0.02	260	170	2280	130	2840	17×33 13×5
0.012	280	200	2380	130	2990	17×33 15×5

Tabella 1: andamento dei tempi al variare del parametro SRT (superfici in Figura 2); CRT=0.01.

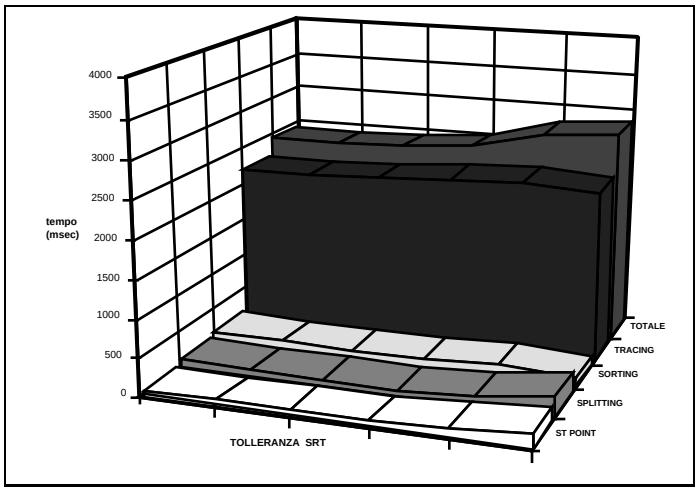

Figura 10: grafico dei tempi di Tabella 1

stesso. L'andamento è messo in risalto dal grafico in Figura 10 relativo alla Tabella 1.
La prima delle ottimizzazioni menzionate può creare problemi nel caso ci siano curve di intersezione che passano più volte per lo stesso segmento. Il problema della perdita del ramo può essere risolto abbassando l'SRT in modo da avere una grigliatura più fine. Il carico di lavoro che ne consegue è illustrato nella Tabella 3 relativa all'intersezione di Figura 4 che presenta questo caso limite di curve ravvicinate.
Per come è stata realizzata la suddivisione in griglie adattive delle superfici, il forzare SRT verso valori molto piccoli può presentare problemi numerici nel caso in cui la superficie sia lineare. Negli esempi di Figura 5, 6 e 7 infatti, il cilindro tende ad essere grigliato eccessivamente in una direzione rispetto all'altra, con conseguente determinazione di triangoli 3D stretti e lunghi. Questo porta a problemi di intersezione segmento-triangolo mal condizionati che potrebbero essere superati gestendo sepa-

SRT	$TEMPI(msec)$					$GRIGLIA$
	$Splitting$	$St.Point$	$Tracing$	$Sorting$	$Totale$	
0.12	80	30	720	60	890	5×11 9×17
0.10	80	30	730	60	890	5×12 9×17
0.08	80	40	720	60	900	5×13 9×17
0.06	100	40	730	60	930	5×17 9×17
0.04	140	90	740	60	1030	9×23 9×17
0.02	360	460	750	50	1620	9×29 17×33
0.012	750	620	780	70	2220	9×40 17×33

Tabella 2: andamento dei tempi al variare del parametro SRT (superfici in Figura 3); CRT=0.01.

SRT	$TEMPI(msec)$					$GRIGLIA$
	$Splitting$	$St.Point$	$Tracing$	$Sorting$	$Totale$	
0.1	110	110	170	10	400	7×17 9×17
0.05	100	140	160	10	410	8×17 12×17
0.02	500	1170	360	30	2060	14×33 19×33

Tabella 3: Comportamento dell'algoritmo al variare del parametro SRT (superfici in Figura 4): con SRT=0.1 e SRT=0.05 trova solo una delle due curve di intersezione, con SRT=0.02 le trova entrambe; CRT=0.02.

CRT	*TEMPI*					*PUNTI*
	Splitting	*St.Point*	*Tracing*	*Sorting*	*Totale*	
0.011	80	30	720	50	880	1141
0.010	80	40	740	60	910	1254
0.008	80	30	900	80	1090	1582
0.006	80	30	1170	60	1340	2120
0.004	80	20	1730	160	1990	3194

Tabella 4: andamento dei tempi al variare del parametro CRT (superfici in Figura 3); già con CRT=0.011 vengono trovate tutte le curve di intersezione; SRT=0.12.

ratamente la grigliatura nelle due direzioni, o evitando di diminuire troppo l'SRT nei casi di superfici lineari.

9.2. Analisi della tolleranza CRT

E' possibile guidare la qualità delle curve di intersezione per bilanciare accuratezza ed efficienza, sia tramite la tolleranza SRT, influendo perciò sulla approssimazione lineare e il numero di starting point da cui partire, sia mediante la tolleranza CRT al fine di limitare il numero di punti nella fase di tracciamento della curva. Al diminuire di questa tolleranza si ottiene una approssimazione più raffinata della curva, poichè si trova un maggior numero di punti di intersezione a scapito dell' efficienza dell' algoritmo. Un metodo per ottenere buone tolleranze è calcolarle in modo inversamente proporzionale alla massima curvatura della superficie, valore spesso non noto. Al diminuire di questo valore le tolleranze dovrebbero decrescere e viceversa. In questa implementazione si è optato per una stima di CRT che consentisse di mantenere curve di intersezione rettilinee di al più 100 punti. A tal fine si è stimato il valore di CRT iniziale come 1/100 della lunghezza della diagonale del parallelepipedo intersezione.
Il parametro SRT negli esempi che seguono è quello suggerito come stima iniziale. Mantenendo perciò l'SRT a valori appena sufficienti a trovare uno starting point per ogni ramo di curva, è necessario forzare il parametro CRT a valori piccoli affinchè non abbandoni il tracing del ramo poichè non avrebbe più altri punti sul ramo stesso da cui ripartire. Ci si è chiesto quindi quanto costa il dover spingere il CRT dal valore consigliato a valori pari ad un decimo del CRT iniziale. I valori riportati in Tabella 4 relativi al test in Figura 3 mettono in evidenza come l'utilizzare una tolleranza CRT più o meno piccola, che agisce sul tempo di tracing, influenzi il tempo complessivo proporzionalmente al numero di punti trovati. L'andamento è riportato graficamente in Figura 11.
L'utilizzo di tolleranze CRT più spinte è talvolta motivato anche dal desiderio di ottenere un'approssimazione più precisa ed accurata della curva. L'accuratezza, ai fini

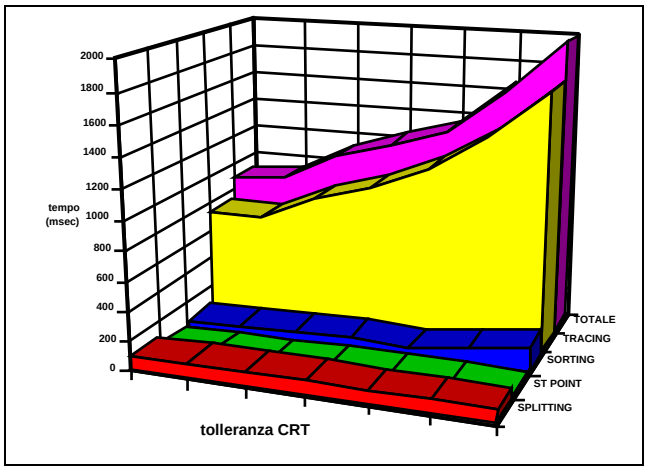

Figura 11: grafico dei tempi di Tabella 4

CRT	$TEMPI$				
	$Splitting$	$St.Point$	$Tracing$	$Sorting$	$Totale$
0.02	740	380	870	40	2010
0.01	740	330	1390	50	2500
0.009	740	410	1430	60	2630
0.007	740	400	1710	80	2920
0.005	740	320	2140	100	3300
0.003	740	310	3540	170	4740

Tabella 5: andamento dei tempi al variare del parametro CRT (superfici in Figura 8); SRT=0.01.

della modellazione solida, è ottenuta inoltre mediante il raffinamento in prossimità dei bordi del dominio parametrico per avere curve che delimitano zone chiuse.
Sebbene la convergenza quadratica del metodo sia garantita solo per superfici di classe C^2 (vedi Appendice), l'algoritmo è stato proposto per superfici almeno C^1. L'esempio seguente, illustrato in Figura 8, mostra come, per opportune scelte dei parametri SRT e CRT, l'algoritmo sia in grado di trattare correttamente anche superfici che siano solo di classe G^0. L'esempio permette di osservare il comportamento dell'algoritmo in presenza di superfici con spigoli vivi. I risultati ottenuti (vedi Tabella 5) evidenziano come in questo caso un SRT molto piccolo (0.01) sia condizione necessaria per trovare almeno uno starting point per ogni ramo G^1, poichè il tracing fallisce in presenza di singolarità, e come sia necessario un CRT sufficientemente piccolo per raggiungere gli spigoli vivi nella fase di tracing. Ovviamente la scelta di tali parametri porta ad un aumento sensibile del costo computazionale.

10. Conclusioni

In questo lavoro viene presentata una versione modificata dell'algoritmo di Barnhill ed altri [1] per l'intersezione di superfici NURBS. La particolarizzazione a tale classe di superfici ha permesso una serie di ottimizzazioni e accorgimenti che portano ad un buon compromesso robustezza-efficienza. Sono allo studio differenti tecniche per la fase di splitting e generazione starting point nonchè una generalizzazione a superfici solo G^0. Un altro obiettivo futuro è la realizzazione di un sistema prototipo per la costruzione di solidi da primitive NURBS mediante operazioni booleane basato sull'algoritmo presentato.

Riferimenti

[1] Barnhill R.E.,Farin G.,Jordan M. e Piper B.R., (1987) Surface/surface intersection, Computer Aided Geometric Design, 4,3-16

[2] Barnhill R.E. e Kersey S.N., (1990) A marching method for parametric surface/surface intersection, Computer Aided Geometric Design, 7, 257-280

[3] Casale M.S., (1987) Free form solid modeling with trimmed patch, IEEE Computer Graphics and Applications, Jan., 33-43

[4] Casale M.S. e Borrow J.E., (1989) A set operation algorithm for sculptured solids modeled with trimmed patches, Computer Aided Geometric Design, 6, 235-247

[5] Casciola G., (1989) Rappresentazione di oggetti a forma libera, Workshop CNR-GNIM Conoscenza per Immagini

[6] Casciola G. e Quaquarelli B., (1990) Primitive solide, operazioni Booleane e NURBS, Atti del convegno ICO-GRAPHICS

[7] Faux I.D. e Pratt M.J., (1979) Computational Geometry for Design and Manufacture, Ellis Horwood, Chichest U.K.

[8] Houghton E.G., Emnet R.F., Factor J.D. e Sabharwal C.L., (1985) Implementation of a divide and conquer method for intersection of parametric surface, Computer Aided Geometric Design, 2, 173-183

[9] Müllenheim G., (1990) Convergence of a surface/surface intersection algorithm, Computer Aided Geometric Design, 7, 415-423

Appendice

Siano $\mathbf{f}$ e $\mathbf{g}$ due superfici parametriche rispettivamente definite dalle funzioni vettoriali:

$$\mathbf{f} : D_1 = [u_a, u_b] \times [v_a, v_b] \rightarrow R^3$$

$$\mathbf{g} : D_2 = [s_a, s_b] \times [t_a, t_b] \rightarrow R^3$$

differenziabili almeno fino alla derivata prima.
Siano $(u_0, v_0) \in D_1$ e $(s_0, t_0) \in D_2$ i punti iniziali del metodo, che si propone di generare due successioni $\{(u_\nu, v_\nu)\}_{\nu \in N}$, $\{(s_\nu, t_\nu)\}_{\nu \in N}$ che convergano ai punti (u^*, v^*) e (s^*, t^*) con $\mathbf{f}(u^*, v^*) = \mathbf{g}(s^*, t^*)$.
Siano $Jf(u, v)$ e $Jg(s, t)$ le matrici Jacobiane di $\mathbf{f}$ e $\mathbf{g}$. Dai due iterati iniziali $(u_0, v_0) \in D_1$ e $(s_0, t_0) \in D_2$ si definiscono le linearizzazioni $\mathbf{\Phi}$ di $\mathbf{f}$ e $\mathbf{\Gamma}$ di $\mathbf{g}$ mediante sviluppo di Taylor

$$\mathbf{\Phi}(u, v) = \mathbf{f}(u_0, v_0) + Jf(u_0, v_0) \begin{pmatrix} u - u_0 \\ v - v_0 \end{pmatrix}$$

$$\mathbf{\Gamma}(s, t) = \mathbf{g}(s_0, t_0) + Jg(s_0, t_0) \begin{pmatrix} s - s_0 \\ t - t_0 \end{pmatrix}$$

con

$$Jf(u_0, v_0) = \begin{pmatrix} \frac{\partial f_1}{\partial u}\Big|_{(u_0,v_0)} & \frac{\partial f_1}{\partial v}\Big|_{(u_0,v_0)} \\ \frac{\partial f_2}{\partial u}\Big|_{(u_0,v_0)} & \frac{\partial f_2}{\partial v}\Big|_{(u_0,v_0)} \\ \frac{\partial f_3}{\partial u}\Big|_{(u_0,v_0)} & \frac{\partial f_3}{\partial v}\Big|_{(u_0,v_0)} \end{pmatrix} \quad Jg(s_0, t_0) = \begin{pmatrix} \frac{\partial g_1}{\partial s}\Big|_{(s_0,t_0)} & \frac{\partial g_1}{\partial t}\Big|_{(s_0,t_0)} \\ \frac{\partial g_2}{\partial s}\Big|_{(s_0,t_0)} & \frac{\partial g_2}{\partial t}\Big|_{(s_0,t_0)} \\ \frac{\partial g_3}{\partial s}\Big|_{(s_0,t_0)} & \frac{\partial g_3}{\partial t}\Big|_{(s_0,t_0)} \end{pmatrix}$$

Si noti che i vettori sono vettori colonna.
Le linearizzazioni $\mathbf{\Phi}$ e $\mathbf{\Gamma}$ descrivono rispettivamente il piano tangente alla superficie $\mathbf{f}$ in (u_0, v_0) e alla superficie $\mathbf{g}$ in (s_0, t_0).
Nelle ipotesi di continuità del piano tangente la matrice Jacobiana non ha rango deficiente in quanto il piano tangente è generato dalle sue colonne.
Per ogni coppia di punti $(u, v) \in D_1$ e $(s, t) \in D_2$ si possono definire le normali alle superfici

$$\mathbf{n_1}(u, v) = \frac{d\mathbf{f}}{du}(u, v) \times \frac{d\mathbf{f}}{dv}(u, v)$$

$$\mathbf{n_2}(s, t) = \frac{d\mathbf{g}}{ds}(s, t) \times \frac{d\mathbf{g}}{dt}(s, t)$$

Se entrambi i vettori $\mathbf{n_1}(u_0, v_0)$ e $\mathbf{n_2}(s_0, t_0)$ sono non nulli e non paralleli allora esiste una retta di intersezione fra i piani $\mathbf{\Phi}$ e $\mathbf{\Gamma}$. Questa retta viene a sua volta intersecata da un terzo piano non parallelo per definire univocamente un nuovo punto 3D $\mathbf{P}_1$. In Barnhill ed altri [1] l'iterato successivo viene preso come il punto medio delle proiezioni dell'iterato precedente sulla retta di intersezione dei due piani tangenti. Se definiamo $\mathbf{n_3} = \mathbf{n_1} \times \mathbf{n_2}$ la normale al terzo piano, l'iterato $\mathbf{P}_1$ usato in [1] può essere valutato da

$$\begin{pmatrix} \mathbf{n_1}^T \\ \mathbf{n_2}^T \\ \mathbf{n_3}^T \end{pmatrix} \mathbf{P}_1 = \begin{pmatrix} \mathbf{n_1} \cdot \mathbf{f}(u_0, v_0) \\ \mathbf{n_2} \cdot \mathbf{g}(s_0, t_0) \\ \mathbf{n_3} \cdot \frac{\mathbf{f}(u_0,v_0)+\mathbf{g}(s_0,t_0)}{2} \end{pmatrix} \tag{3}$$

Un metodo diretto per ottenere l'intersezione di tre piani è suggerito da Faux e Pratt in [7]. Date le tre normali, e definendo gli elementi del lato destro di (3) come b_1, b_2 e b_3 il nuovo punto è

$$\mathbf{P}_1 = \frac{b_1(\mathbf{n_2} \times \mathbf{n_3}) + b_2(\mathbf{n_3} \times \mathbf{n_1}) + b_3(\mathbf{n_1} \times \mathbf{n_2})}{\mathbf{n_1} \cdot (\mathbf{n_2} \times \mathbf{n_3})}$$

In Figura 1 è illustrato questo primo passo del metodo.
Per iterare si determinano le preimmagini di $\mathbf{P}_1$ sui domini parametrici risolvendo i sistemi:

$$\mathbf{\Phi}(u_1, v_1) \begin{pmatrix} u_1 \\ v_1 \end{pmatrix} = \mathbf{P}_1 \qquad \mathbf{\Gamma}(s_1, t_1) \begin{pmatrix} s_1 \\ t_1 \end{pmatrix} = \mathbf{P}_1$$

Poichè questi sistemi sono ridondanti possono essere estratti tre sottosistemi 2×2 da ogni sistema e, per quanto detto, almeno uno non sarà singolare.
Se $(u_1, v_1) \in D_1$ e $(s_1, t_1) \in D_2$ si può ripetere il procedimento utilizzando la nuova coppia di iterati (u_1, v_1) e (s_1, t_1).
Sotto determinate ipotesi le successioni ottenute dal metodo convergono e, inoltre, in [9] è stato dimostrato che se le superfici sono C^2 la convergenza è quadratica.
Trovato un punto $\mathbf{f}(u^*, v^*) = \mathbf{g}(s^*, t^*)$ sulla curva di intersezione ci si muove lungo la stessa iterando il metodo con un nuovo iterato iniziale. Si ottiene in questo modo una successione di punti della curva di intersezione.
Dato un iterato iniziale per trovare un punto di intersezione fra le due superfici si può fissare una variabile (diciamo u) dell'equazione

$$\mathbf{f}(u, v) - \mathbf{g}(s, t) = \mathbf{0} \tag{4}$$

ottenendo e potendo applicare uno schema iterativo di Newton-Raphson al sistema 3×3. Questo metodo è molto meno costoso, ma non è utile per procedere lungo la curva. Barnhill ed altri [1] suggeriscono questa procedura per calcolare il primo punto (starting point) della curva di intersezione e nella nostra implementazione

viene usata anche per determinare i punti delle curve di intersezione sui bordi dei domini parametrici.
Un'altra possibilità definita passo costretto, differisce dal metodo visto nell' impostazione del sistema (4). Viene infatti aggiunta una quarta equazione che definisce il cosiddetto passo costretto e rende determinato il sistema:

$$(\mathbf{f}(u_0, v_0) + c\mathbf{E} - \mathbf{f}(u_i, v_i))\mathbf{E} = \mathbf{0}$$

dove
$\mathbf{f}(u_0, v_0)$ è il punto corrente sulla curva di intersezione;
$\mathbf{E}$ è il versore tangente alla curva di intersezione in $\mathbf{f}(u_0, v_0)$;
c è la distanza del piano di avanzamento dal punto corrente (proporzionale a CRT);
$\mathbf{f}(u_0, v_0) + c\mathbf{E}$ è l'iterato iniziale.
Il significato della quarta equazione è quello di fissare la lunghezza del passo lungo la curva di intersezione. Geometricamente si richiede che il nuovo punto generato dall'algoritmo non solo appartenga alla curva di intersezione, ma giaccia anche su di un piano, chiamato piano di avanzamento, distante c dal punto corrente e perpendicolare al vettore tangente alla curva di intersezione in tale punto.