

POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Meccanica

Tesi di Laurea Magistrale

Sviluppo di un tool per la generazione di una mesh parametrica per ingranaggi



Relatore
prof. Carlo Rosso

Candidato
Giancarlo RIVA GAN

Supervisore aziendale
GE Avio Aero
dott. Luca Ronchiato

Settembre 2018

The first principle is that you must not fool yourself, and you are the easiest person to fool.

– *Richard Feynman*

Ringraziamenti

Desidero ringraziare il Prof. Carlo Rosso, relatore di questa tesi, il mio tutor dott. Luca Ronchiato e il dott. Paride Mesaglio Chittaro, per avermi seguito e supportato durante l'attività di tesi.

Ringrazio il dott. Michele Gravina per la disponibilità nello spiegarmi gli aspetti più complicati della geometria delle ruote dentate.

Ringrazio l'intero team System Design Technologies perchè ho trovato un ambiente di lavoro positivo e stimolante.

Il ringraziamento più grande va ai miei genitori, per tutto ciò che mi hanno trasmesso, per il loro esempio, e alla mia famiglia, a cui dedico questa tesi, che nelle fasi più complicate del percorso mi ha sempre supportato, e sopportato. Vi voglio bene.

Infine, una dedica speciale ai miei amici e alla mia ragazza, che ogni giorno hanno condiviso con me gioie, sacrifici e successi, senza voltarmi mai le spalle. L'affetto e il sostegno che mi hanno dimostrato rendono questo traguardo ancora più prezioso.

Indice

Elenco delle tabelle	VI
Elenco delle figure	VII
I Prima Parte	1
1 Introduzione generale	3
1.1 Introduzione	3
1.2 Scopo della mesh parametrica	3
1.2.1 Applicazioni nell'industria	3
1.2.2 Altre applicazioni	4
1.3 Obiettivi preposti all'inizio della tesi	4
1.4 Ordine di presentazione degli argomenti	4
2 Topologia	7
2.1 Introduzione	7
2.2 Connessione degli spazi euclidei	7
2.2.1 Definizione di connessione e semplice connessione	7
2.2.2 La necessità della semplice connessione	9
2.3 Insiemi convessi	9
2.3.1 Definizione	9
2.3.2 Esempi	10
2.3.3 Utilizzo di questa definizione nel tool	11
2.3.4 Gruppi fondamentali	12
2.4 Conclusioni	13
3 Mesh	15
3.1 Introduzione	15
3.2 La mesh: utilizzo, definizione, tipologie	15
3.2.1 Soluzione dei problemi ingegneristici	15
3.2.2 Definizione	15
3.2.3 Tipologie	17
3.2.4 Topologia di una mesh	19
3.3 Conclusioni	20

II	Seconda Parte	21
4	Ruote a denti dritti	23
4.1	Introduzione	23
4.2	Parametri in input	24
4.3	Geometria del vano	24
4.4	Mesh 2D - introduzione	26
4.4.1	Anatomia della mesh	26
4.4.2	Mesh di riferimento	26
4.4.3	Strutture della mesh	29
4.4.4	Sequenza di operazioni per ottenere la mesh della ruota	29
4.5	Mesh 2D - metà sezione	30
4.5.1	Implementazione - modelli a 2, 3, 5 zone	30
4.5.2	Main nodes	33
4.5.3	Vincoli sui main nodes	35
4.5.4	Algoritmi di calcolo della matrice dei nodi	36
4.5.5	Algoritmi di calcolo degli elementi 1D, 2D	42
4.6	Mesh 2D - intera sezione	43
4.7	Mesh 3D	45
4.8	Risultati	46
4.9	Conclusioni	58
5	Ruote a denti elicoidali	59
5.1	Introduzione	59
5.2	Realizzazione	59
III	Terza Parte	65
6	Ruote spiroconiche	67
6.1	Introduzione	67
6.2	Il tool preesistente	67
6.3	Algoritmi di intersezione	68
6.3.1	Metodo con proiezioni ortogonali	69
6.4	Metodo "Curve points check"	72
6.4.1	Introduzione	72
6.4.2	Principio	73
6.5	Algoritmo di mesh 3D	73
6.5.1	Ottimizzazione	73
6.5.2	Prestazioni	74
6.6	Risultati	75
7	Esportazione dei risultati	81
8	Conclusioni	83
	Bibliografia	85

Elenco delle tabelle

Elenco delle figure

2.1	Insieme non semplicemente connesso	8
2.2	Insieme semplicemente connesso	8
2.3	Insieme semplicemente connesso, meshabile	9
2.4	Insieme convesso	10
2.5	Insieme concavo	10
2.6	Paraboloide	11
2.7	Ellissoide	11
2.8	Laccio giacente sulla superficie toroidale	12
2.9	Lacci "a" e "b" giacenti sulla superficie toroidale	12
3.1	Tipologie di elementi	17
3.2	Mesh strutturata cartesiana	17
3.3	Mesh strutturata curvilinea	17
3.4	Mesh non strutturata	18
3.5	Mesh ibrida	18
3.6	Mesh Single-Block	19
3.7	Mesh Multi-Block	19
4.1	Ruota a denti dritti	23
4.2	Vano full fillet	25
4.3	Vano flat fillet	25
4.4	Mesh di metà dente - Anatomia	26
4.5	Template mesh 1	27
4.6	Template mesh 2	27
4.7	Template mesh 3	28
4.8	Template mesh 3 - zone	29
4.9	Modello a 2 zone	30
4.10	Modello a 3 zone	31
4.11	Modello a 3 zone - criticità intersezione	32
4.12	Modello a 5 zone	32
4.13	Modello a 5 zone - caso rim basso	33
4.14	Modello a 5 zone - caso rim alto	33
4.15	Modello a 5 zone - main nodes	34
4.16	Modello a 3 zone - main nodes	34
4.17	Esempio 1 - struttura di elementi quadrati	35
4.18	Esempio 2 - struttura di quadrilateri	36
4.19	Zone della mesh, main nodes, e sistemi di riferimento	37
4.20	Nodi e punti di integrazione in un elemento "quad"	38
4.21	Nodi - set 1	39
4.22	Nodi - set 2	39
4.23	Nodi - set 3	40

4.24	Nodi - set 4	40
4.25	Convenzione elementi 1D	42
4.26	Convenzione elementi 2D	42
4.27	Simmetria del dente	43
4.28	Topologia dei nodi della mesh sinistra di una dentatura a denti dritti	44
4.29	Topologia dei nodi della mesh sinistra di una dentatura spiroconica	44
4.30	Rappresentazione 3D della struttura dei nodi e degli elementi 2D	45
4.31	Mesh 3D - dente esterno - raccordo full fillet - set di parametri 1	46
4.32	Mesh 3D - dente esterno - raccordo full fillet - set di parametri 2	47
4.33	Mesh 3D - dente esterno - raccordo full fillet - set di parametri 3	48
4.34	Mesh 3D - dente esterno - raccordo flat root - set di parametri 1	49
4.35	Mesh 3D - dente esterno - raccordo flat root - set di parametri 1	50
4.36	Mesh 3D - dente esterno - raccordo flat root - set di parametri 2	51
4.37	Mesh 3D - dente esterno - raccordo flat root - set di parametri 3	52
4.38	Mesh 3D - dente interno	53
4.39	Mesh 3D - ruota a denti esterni	54
4.40	Mesh 3D - ruota a denti esterni, dettaglio	55
4.41	Mesh 3D - ruota a denti interni	56
4.42	Mesh 3D - ruota a denti interni	56
4.43	Mesh 3D - ruota a denti interni	57
4.44	Mesh 3D - ruota a denti interni	58
5.1	Mesh 3D - dente	59
5.2	Mesh 3D - ruota a denti esterni elicoidale - set 1	60
5.3	Mesh 3D - ruota a denti interni elicoidale - set 1	61
5.4	Mesh 3D - ruota a denti interni elicoidale	62
5.5	Mesh 3D - ruota a denti interni elicoidale, dettaglio	63
6.1	Output del tool modificato	68
6.2	Punti di intersezione tra fascio di spline e superfici coniche	69
6.3	Schema di base	70
6.4	Schema qualitativo delle iterazioni	71
6.5	Precisione algoritmo	72
6.6	Sistemi di riferimento coinvolti	73
6.7	Sistemi di riferimento coinvolti	74
6.8	Dente - set parametri 1	75
6.9	Dente - set parametri 2	76
6.10	Dente - set parametri 3	77
6.11	Ruota spiroconica - test case 1	78
6.12	Ruota spiroconica - test case 2	79
7.1	Esempio di mesh generata dal tool aperta in Hypermesh	81

Parte I
Prima Parte

Capitolo 1

Introduzione generale

1.1 Introduzione

In questo capitolo si descrive quale sia lo scopo per cui viene realizzata una mesh parametrica, quali sono stati gli obiettivi preposti all'inizio della tesi e per quale motivo.

Infine, si descrive come e perchè gli argomenti sono presentati in questo ordine, in modo che sia visibile il filo logico che collega le varie sezioni.

1.2 Scopo della mesh parametrica

La realizzazione di mesh in tre dimensioni di componenti meccanici è materia ben nota, sono presenti vari software commerciali per i quali, realizzare una mesh automaticamente richiede pochi secondi.

Nel caso delle trasmissioni meccaniche, in cui siano presenti ruote dentate, è necessario però valutare con precisione, specialmente in campo aeronautico, stress e deformazioni.

Una mesh automatica, in questo caso, non ha le caratteristiche adatte per questi scopi e dunque deve essere rifinita manualmente.

La possibilità di generare automaticamente la mesh di una dentatura partendo dai suoi parametri caratteristici quali, ad esempio, modulo e numero di denti, determina un significativo risparmio di tempo e di ore-uomo. L'aumento di produttività è dunque evidente.

Sono presenti codici in commercio che effettuano questa operazione di mesh delle dentature, tuttavia, si tratta di codici non open source, dunque difficili da personalizzare nel caso si voglia concentrare l'attenzione su una zona piuttosto che un'altra del dente.

Questa mancanza, in termini di produttività è penalizzante poichè, essendo vincolati a dei template prefissati, per realizzare analisi dettagliate soltanto in alcune zone del dente, si aumenta inutilmente il costo computazionale ed il tempo di calcolo conseguentemente.

L'obiettivo che si propone la tesi è la realizzazione di un tool scritto in linguaggio MATLAB che realizzi la mesh velocemente e con una certa robustezza al variare dei parametri inseriti.

La scelta di MATLAB come linguaggio di programmazione, permette di sfruttare la vettorizzazione, che come sarà esposto più avanti, è determinante al fine di avere un codice performante.

1.2.1 Applicazioni nell'industria

Le applicazioni di un tool di questo tipo nell'industria sono nelle analisi general purpose, ad esempio statica e dinamica delle trasmissioni meccaniche che sfruttano ruote dentate a denti dritti, elicoidali,

spiroconiche.

La possibilità di settare il numero di elementi è una caratteristica che rende il tool versatile.

Per quanto riguarda la dinamica, è possibile, svolgere sia analisi sugli spostamenti che analisi modale.

Il tool realizzato non è adatto per la Tooth Contact Analysis (TCA), il motivo è che non si tiene conto delle microcorrezioni a cui le dentature reali sono soggette, rispetto a quelle ideali.

Le motivazioni per cui nascono le microcorrezioni sono varie, alcune si riscontrano nell'ambito della distribuzione degli stress, che si preferisce concentrare nel centro della larghezza di fascia piuttosto che sui bordi, oppure nell'ambito della cinematica; è importante che in funzionamento non si verifichino interferenze e le ruote trovino quindi forte difficoltà a ruotare.

1.2.2 Altre applicazioni

L'algoritmo scritto per la realizzazione di mesh di dentature, è di carattere generale, dunque può essere modificato al fine di ottenere anche mesh di palette di turbina, sempre in 3D, piuttosto che altri componenti assialsimmetrici come alberi di trasmissione, flange e parti di cuscinetti.

1.3 Obiettivi preposti all'inizio della tesi

L'evoluzione di un percorso di tesi non è mai noto, nei dettagli, a priori. Negli incontri iniziali con il tutor aziendale si è stabilita una linea da seguire al fine di arrivare a realizzare mesh di ruote dentate in 3D.

Sono presentati, dunque, gli step intermedi:

1. Realizzazione del profilo di metà del dente: avendo come input i parametri geometrici della dentatura, il primo step consiste nel calcolo corretto del profilo evolvente, del raccordo, di eventuali giunzioni e del tratto di troncatura esterna o tip.
2. Realizzazione della mesh 2D di metà del dente: si sfruttano non solo i parametri della dentatura ma anche le caratteristiche che deve avere la mesh per poter essere accoppiata alla mesh dell'albero, come, ad esempio, il diametro interno.
In questo step interviene l'algoritmo di realizzazione della mesh 2D, che deve calcolare gli elementi finiti 1D, 2D.
3. Realizzazione della mesh 3D per una dentatura a denti dritti
4. Realizzazione della mesh 3D per una dentatura elicoidale
5. Realizzazione della mesh 3D per una dentatura spiroconica: le ruote spiroconiche hanno una geometria più complessa e quindi si è deciso di trattarle come un argomento a parte, sia come generazione del profilo, che come mesh 2D, piuttosto che come mesh 3D.
6. Calcolo delle informazioni relative alla dentatura, e al suo ingranamento con un'altra. Ad esempio, sono stati calcolati: HPSTC, LPSTC, lunghezza della linea di azione, roll angles.

1.4 Ordine di presentazione degli argomenti

Il primo capitolo è teorico, tratta degli argomenti di topologia. Essi sono esposti al fine rendere chiare quali sono le caratteristiche di questo tool, i suoi limiti, e spiegare i motivi della sua flessibilità rispetto alle applicazioni. Questi concetti saranno ripresi nella trattazione nei capitoli successivi.

Dopo i capitoli introduttivi, al fine di presentare con chiarezza l'elaborato, si è scelto di presentare gli argomenti in ordine di difficoltà di realizzazione.

Il capitolo successivo tratta il significato della parola "mesh" e introduce brevemente alcuni dei metodi che possono essere utilizzati per generarla, al fine di spiegare la scelta del metodo di generazione utilizzato nel tool.

Dopo un capitolo iniziale, si trattano le ruote dentate a denti dritti, successivamente le elicoidali e infine le spiroconiche.

Il capitolo relativo alle ruote a denti dritti introduce l'algoritmo di mesh 2D che verrà utilizzato successivamente anche per le ruote elicoidali, e il cui principio è lo stesso delle ruote spiroconiche, anche se è stato opportunamente modificato.

In ognuno dei capitoli dedicati alle ruote, oltre al modo in cui è stato caratterizzato il codice, sono presentati anche degli esempi di ruote reali per dimostrare la coerenza dell'algoritmo, dei suoi risultati e il rispetto della precisione richiesti.

Nel capitolo relativo alle ruote a denti dritti è posta l'attenzione anche sulla geometria del profilo del dente, in questo tipo di ruota e nelle elicoidali, il tipo raccordo a fondo dente si rivela una caratteristica importante nella realizzazione della mesh.

Le prestazioni di un codice sono un altro elemento di interesse, soprattutto per dimostrare che una singola stesura può non essere sufficiente per ottenere un codice performante a livello ottimale, e per porre l'attenzione sul fatto che l'utilizzo di un codice poco performante può influire negativamente sulla qualità della gestione del tempo e infine dei risultati a lungo termine.

Un'altra considerazione riguarda la correttezza dei risultati: la maggior parte delle rappresentazioni grafiche delle ruote in questa tesi di laurea, non sono il prodotto di CAD ma sono la rappresentazione delle matrici ottenute come risultato dell'algoritmo, eventuali errori di rappresentazione ne determinerebbero dunque la non accettabilità o la non robustezza.

Le ruote spiroconiche sono da considerarsi quasi un argomento separato dalle ruote elicoidali e a denti dritti, che invece hanno molte più affinità.

La trattazione delle spiroconiche parte dalla spiegazione delle variazioni dell'algoritmo di mesh delle sezioni, seguono gli algoritmi di intersezione, infine si descrivono le caratteristiche dei risultati e le prestazioni del codice.

Capitolo 2

Topologia

2.1 Introduzione

L'obbiettivo di questo capitolo è delineare i principali concetti di topologia che sono sfruttati nel tool, in modo da descrivere con discreta accuratezza quali sono le caratteristiche attuali e quelle che potrebbe assumere. Il primo argomento è inerente la connessione degli spazi euclidei. Si motiva perchè le zone, o blocchi, della superficie da meshare debbano essere semplicemente connessi.

Segue la spiegazione della definizione di insieme convesso, importante nella geometria del blocco generico della mesh, con annessi esempi ed infine la caratteristica che deve rispettare ogni blocco del dominio su cui effettuare l'operazione di mesh.

L'ultimo argomento trattato è l'omotopia, essa insieme al concetto di parametrizzazione di una curva, permette di comprendere perchè la matrice dei nodi permette un disaccoppiamento molto efficace tra coordinate dei punti di un elemento finito e le connessioni tra i punti che definiscono l'elemento.

2.2 Connessione degli spazi euclidei

2.2.1 Definizione di connessione e semplice connessione

"In topologia, uno spazio topologico è semplicemente connesso se è connesso per archi e il suo gruppo fondamentale è il gruppo banale, ovvero se ogni curva chiusa può essere deformata fino a ridursi a un singolo punto"

Il significato di questa affermazione, in maniera assolutamente non rigorosa, nel campo bidimensionale si può tradurre nel " non avere buchi ".

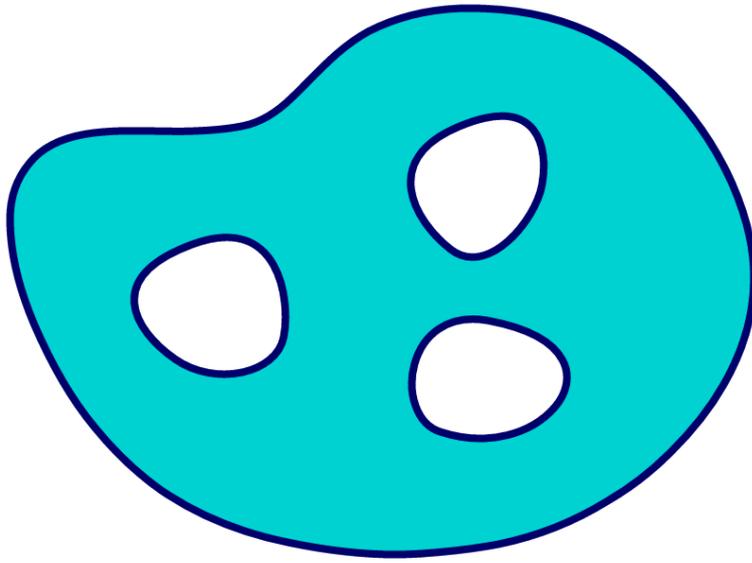


Figura 2.1. Insieme non semplicemente connesso

Il dominio in Fig.2.1 si può sezionare in modo da ottenere regioni semplicemente connesse. Il fatto che siano semplicemente connesse non implica che debbano essere convesse. Le regioni possono essere concave.

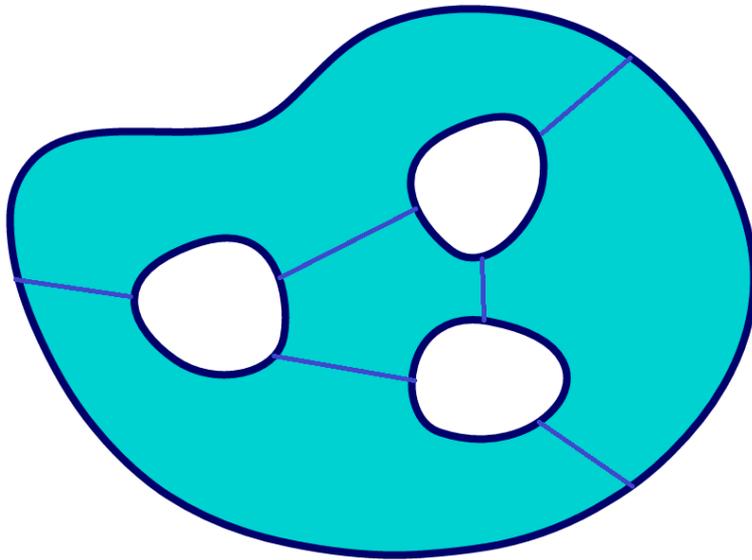


Figura 2.2. Insieme semplicemente connesso

Il tool attuale, se adattato, riuscirebbe a realizzare la mesh se le regioni avessero quattro bordi ciascuna. Il tool, lavorando con geometrie semplici si rivela flessibile, il lato negativo è che fare le modifiche richiederebbe tempo. Non si tratta di un generatore di mesh automatico general purpose quanto di un tool specifico per l'applicazione " dentature di ruote dentate ".

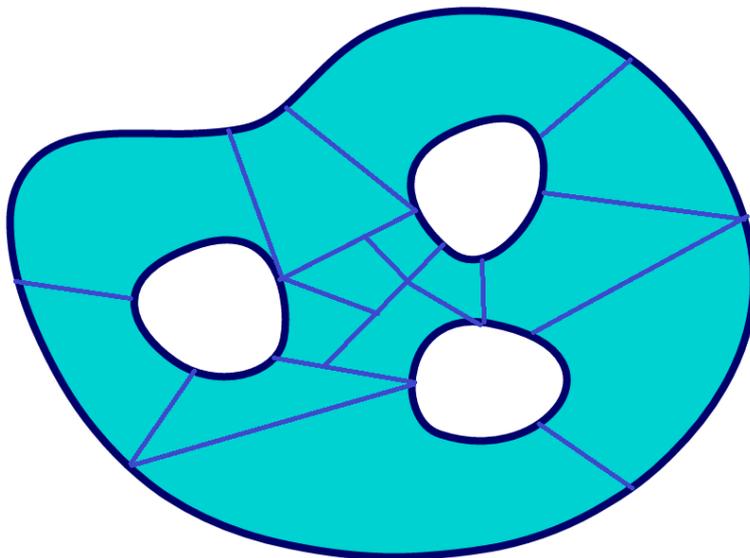


Figura 2.3. Insieme semplicemente connesso, meshabile

2.2.2 La necessità della semplice connessione

Lo script non è in grado di realizzare la mesh di un dominio che non è semplicemente connesso. Il motivo è il seguente: usando logiche di calcolo dei punti molto semplici, basandosi su blocchi con solo e solamente quattro lati, non è possibile passare alla funzione che calcola i punti interni alle zone più di quattro lati. Il risultato è che è possibile passare insiemi convessi, in qualche caso particolare concavi ma non domini non semplicemente connessi.

2.3 Insiemi convessi

2.3.1 Definizione

Le definizione di insiemi convessi è la seguente: "In uno spazio euclideo un insieme convesso è un insieme nel quale, per ogni coppia di punti, il segmento che li congiunge è interamente contenuto nell'insieme" Se ne deduce che sono insiemi convessi tutte quelle figure che non hanno concavità come gli archi di circonferenze nel piano e i tori nello spazio.

2.3.2 Esempi

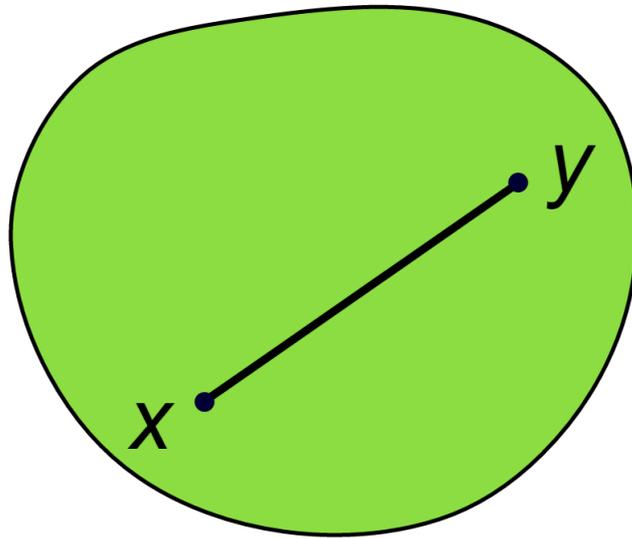


Figura 2.4. Insieme convesso

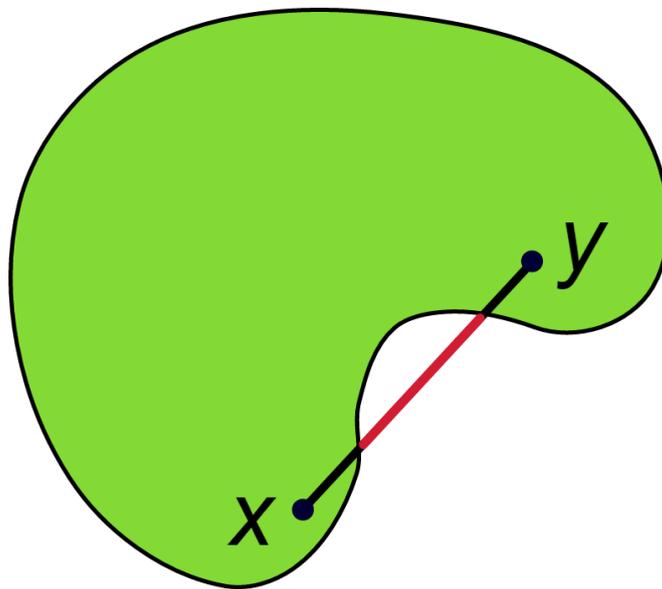


Figura 2.5. Insieme concavo

In Fig.2.4 e Fig.2.5 sono presentati due esempi di figure concava e convessa per quanto concerne figure appartenenti al piano.
Nello spazio sono convessi insiemi come il paraboloide e l'ellissoide.

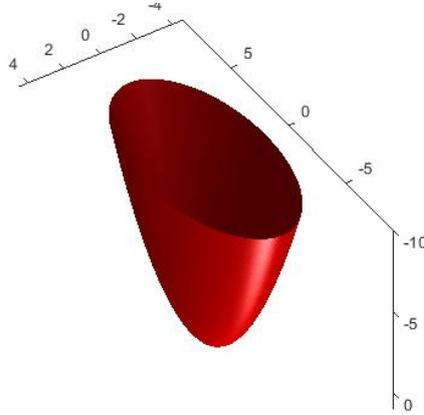


Figura 2.6. Paraboloide

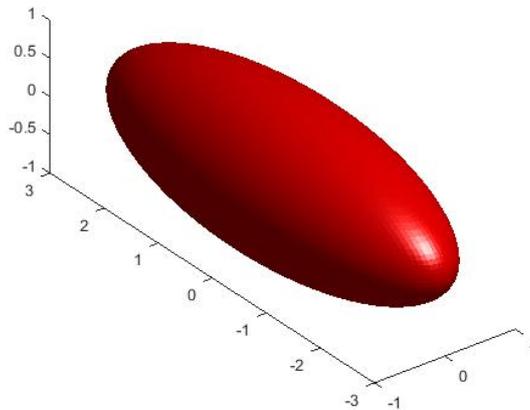


Figura 2.7. Ellissoide

2.3.3 Utilizzo di questa definizione nel tool

La definizione di dominio convesso/concavo è importante nella realizzazione della mesh in un singolo blocco del dominio da meshare. Sostanzialmente, sono ammessi domini in cui è presente un

bordo non rettilineo che determina la regione sia concava. Nel caso della mesh di un dente si tratta di quella che nei capitoli successivi sarà chiamata "zona 4", cioè quella in cui è presente come bordo il raccordo del dente. Se il bordo non rettilineo rende la regione convessa non esistono complicazioni.

In questo tool, nello script dedicato alle ruote spiroconiche, i punti delle sezioni del dente appartengono a superfici coniche posizionate come loro sostegno. Le zone che definiscono questi punti sono convesse, a causa della superficie su cui poggiano. Applicando una trasformazione lineare che trasporta questi punti della superficie in un piano definito dai due parametri della superficie, si potrà osservare che la zona 4 sarà ancora concava però. Un risultato utile di questa considerazione è che non è rilevante che le zone poggino su una superficie conica o iperbolica; anche nel caso iperbolico, che non è convesso, non ci sarebbero problemi di rappresentazione.

2.3.4 Gruppi fondamentali

Avendo uno spazio topologico da analizzare, quale può essere un sottoinsieme dello spazio, o del piano, tramite il gruppo fondamentale, è possibile tradurre questo oggetto in forma algebrica.

A livello intuitivo, prendendo la superficie di un toro, si può riconoscere un gruppo fondamentale considerando dei lacci poggiati sulla sua superficie.

Sapendo che il toro è un solido di rivoluzione senza punti appartenenti all'asse di rotazione, si deduce che al centro vi sia una cavità. Questa è la caratteristica principale di questa superficie, un gruppo deve riuscire a rappresentarla.

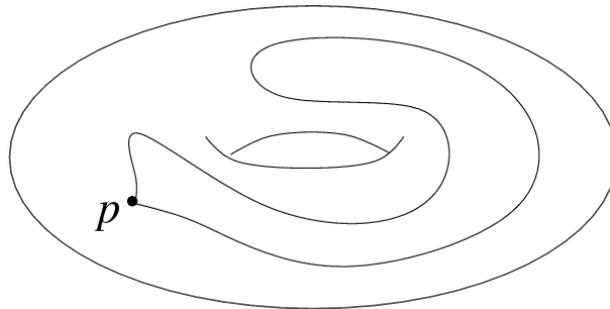


Figura 2.8. Laccio giacente sulla superficie toroidale

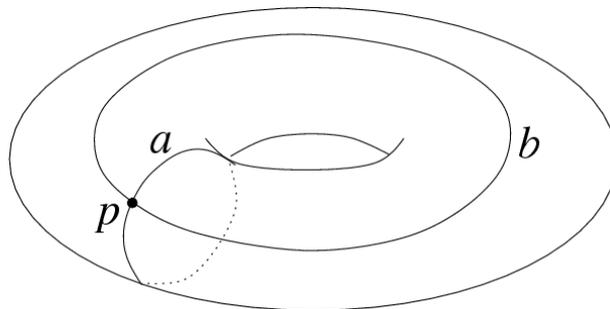


Figura 2.9. Lacci "a" e "b" giacenti sulla superficie toroidale

In Fig.2.8 è evidente come tramite omotopia il laccio possa essere ridotto fino ad un punto, avendo sempre punti che giacciono sulla superficie. La stessa cosa non si può dire dei lacci "a" e "b", in Fig.2.9, che sono vincolati nel caso di omotopie. Se vengono applicate delle deformazioni, i punti del laccio "a" abbandonano la superficie toroidale e quindi si individua una caratteristica del solido che il laccio "p" non poteva descrivere.

L'insieme dei due lacci "a" e "b" è sufficiente a descrivere il solido. Essi insieme compongono un oggetto che sta alla base dell'algebra: il gruppo.

Per definizione, quando uno spazio topologico ha un solo gruppo, si definisce " semplicemente connesso ", e il gruppo si definisce " gruppo banale ".

Nell'ambito di questa tesi di laurea tutte i blocchi di cui sarà costituita la mesh saranno semplicemente connessi.

2.4 Conclusioni

In questo capitolo son stati presentati alcuni argomenti teorici, utili nei prossimi capitoli ad attribuire aggettivi formalmente corretti agli oggetti che costituiscono la mesh. Alcuni riferimenti sono stati anticipati per rendere questo capitolo meno slegato dalla trattazione.

Capitolo 3

Mesh

3.1 Introduzione

Lo scopo di questo capitolo è fornire una panoramica di quali sono i principali tipi di mesh e da quali elementi una mesh generalmente è composta. Questa classificazione è utile al lettore, che potrà comprendere meglio i motivi per cui è stato scelto un tipo di mesh strutturato in questa tesi di laurea.

3.2 La mesh: utilizzo, definizione, tipologie

3.2.1 Soluzione dei problemi ingegneristici

La soluzione di molti problemi dell'ingegneria richiede la risoluzione di equazioni alle derivate parziali su un certo dominio e con delle precise condizioni al contorno.

La discretizzazione del dominio è un elemento essenziale, qui intervengono le tecniche di mesh generation.

Nel caso di problemi di analisi lineare elastica statica, per esempio, il problema si riduce ad un sistema lineare del tipo:

$$Au = f$$

dove A è la matrice di rigidezza del sistema, u il vettore degli spostamenti incogniti, f il vettore dei termini noti.

I coefficienti della matrice A sono calcolati conoscendo la formulazione variazionale del problema, gli elementi finiti che costituiscono la mesh e le caratteristiche del materiale.

3.2.2 Definizione

A livello formale una mesh M può essere definita come una tupla (V, Q) dove $V = v_i \in R^3 = 1 \dots N_v$ 'è l'insieme dei vertici del modello, appartenenti allo spazio, e Q contiene l'adjacency information, cioè quali sono le connessioni tra i vertici che formano gli elementi della mesh: spigoli e facce.

La mesh composta da un singolo elemento quadrilatero ("quad"), si esprime come:

$$M = (V, Q) = (\{v_0, v_1, v_2, v_3\}, \{\{v_0, v_1\}, \{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_0\}, \{v_0, v_1, v_2, v_3\}\})$$

in cui sono elencati in sequenza i quattro vertici, i quattro spigoli, il quadrilatero.

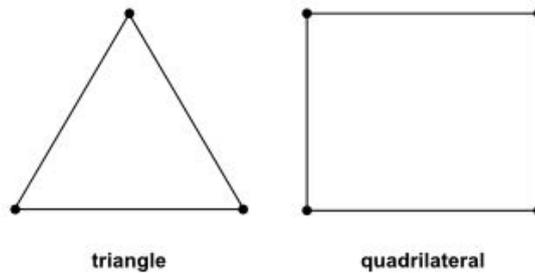
A livello intuitivo, si definisce mesh la partizione di una superficie continua in celle poligonali. Esse possono essere ad esempio, triangoli e quadrilateri. Questo assunto 'è valido anche per una

superficie nello spazio, nella quale bisogna anche considerare la presenza di elementi finiti in 3 dimensioni.

Nel caso tridimensionale gli elementi classici che si riconoscono in una mesh sono tetraedri, piramidi a base quadrata, prismi a base triangolare ed esaedri. Tutti hanno facce triangolari o quadrangolari. Nel caso di mesh 3D ottenute per estrusione di una superficie discretizzata con questi elementi, si ottengono prismi a base triangolare ed esaedri.

Esistono anche elementi 2D con pi' u di quattro lati, ad esempio gli elementi a 5 lati si possono ottenere come combinazione di un elemento triangolare ed uno quadrilaterale.

2D CELL TYPES



3D CELL TYPES

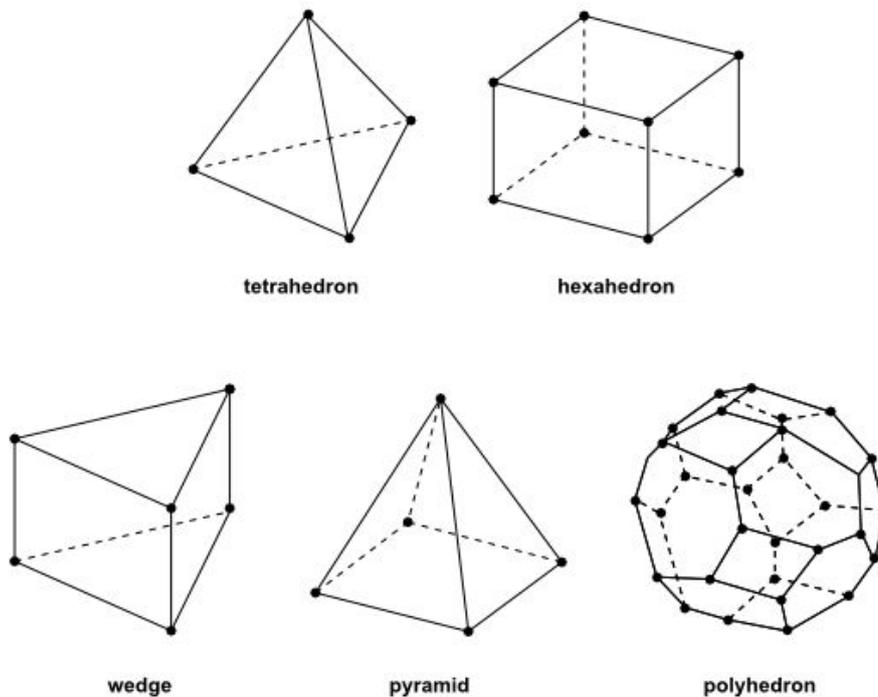


Figura 3.1. Tipologie di elementi

3.2.3 Tipologie

Le tipologie di mesh sono principalmente due: strutturate e non strutturate.

La differenza è nella connettività: le mesh strutturate hanno una connettività regolare.

Questa caratteristica determina una maggiore velocità di individuazione di un elemento della mesh, se si conosce l'algoritmo con cui sono stati numerati gli elementi.

Sono esempio di mesh regolari:

- cartesiana
- curvilinea

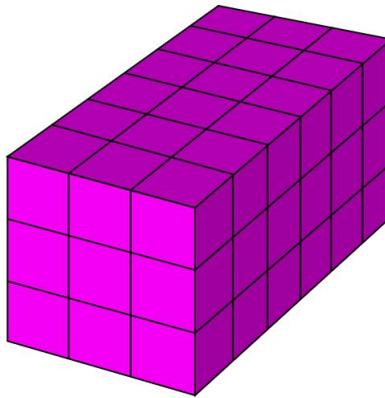


Figura 3.2. Mesh strutturata cartesiana

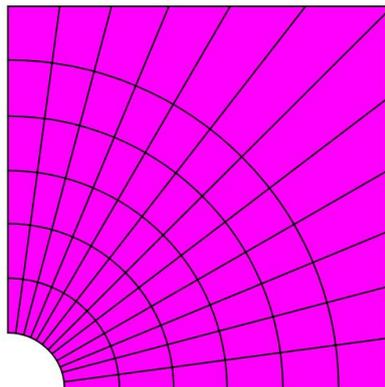


Figura 3.3. Mesh strutturata curvilinea

Nelle mesh non strutturate la connettività è irregolare, questo complica il riconoscimento degli elementi, inoltre, l'utilizzo della memoria per salvare le informazioni relative alla mesh, è superiore e di conseguenza più inefficiente.

Sono possibili anche soluzioni ibride in cui alcune parti del dominio sono discretizzate con una mesh strutturata e altre parti con una mesh non strutturata. L'applicazione determina la scelta del tipo di mesh.

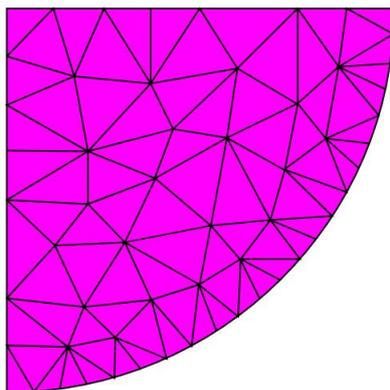


Figura 3.4. Mesh non strutturata

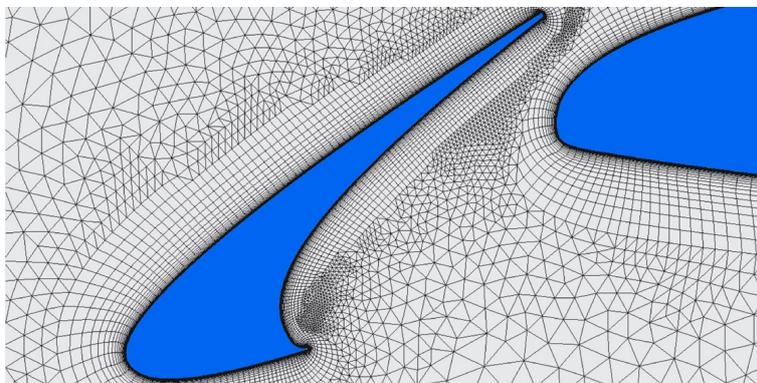


Figura 3.5. Mesh ibrida

Il vantaggio dell'utilizzo di mesh non strutturate risiede nel riuscire ad approssimare meglio il dominio, rispetto a quelle strutturate. Questo succede frequentemente quando i domini hanno una forma complessa. La combinazione di mesh strutturate e non si rivela uno strumento molto versatile nell'ambito di discretizzazione di un dominio.

3.2.4 Topologia di una mesh

La topologia di una mesh può essere "Single-Block" (SB) oppure "Multi-Block" (MB). La topologia MB ha il vantaggio rispetto a quella SB, in alcuni domini, di produrre elementi di qualità superiore, inoltre è più flessibile.

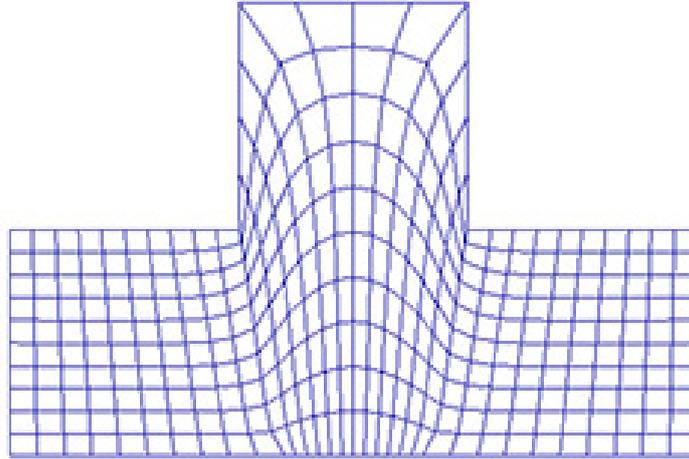


Figura 3.6. Mesh Single-Block

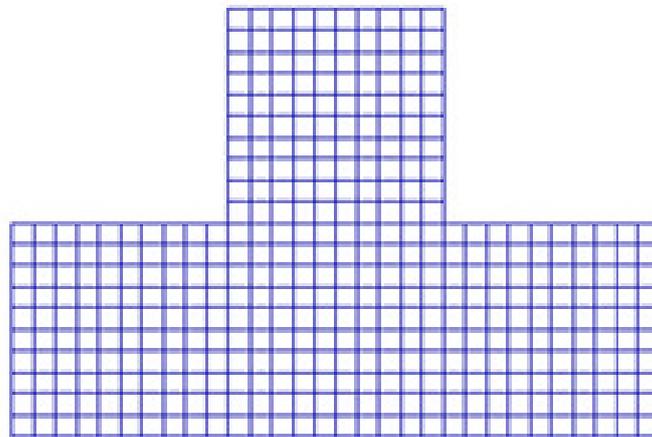


Figura 3.7. Mesh Multi-Block

Confrontando le figure Fig.3.6 e Fig.3.7 si osserva come una mesh MB permetta di avere elementi molto più regolari, la discretizzazione risulta meno approssimata nella parte superiore del dominio.

Una gestione attenta delle singolarità che si manifestano in una mesh, può essere sfruttata per realizzare mesh MB ben ordinate, e robuste, al variare dei parametri, nel caso di dentature.

3.3 Conclusioni

La geometria delle dentature è definita da parametri geometrici, i denti non assumeranno mai forme eccessivamente diverse tra loro. Partendo da questo presupposto si deduce che il dominio, al variare della dentatura non vedrà eccessive deformazioni.

Riuscire a realizzare una mesh strutturata per una configurazione di dentatura risulta dunque sufficiente al fine di meshare una grande quantità di configurazioni, la maggior parte di quelle usate nell'industria.

I motivi per cui si è scelto di utilizzare una mesh strutturata verranno esplicitati meglio nei capitoli successivi.

Per quanto concerne l'utilizzo di una mesh SB o MB, la scelta è stata sostanzialmente obbligata ad una mesh MB, poichè non sarebbe stato possibile raggiungere gli obiettivi preposti con una SB; anche questa scelta verrà motivata meglio nei capitoli successivi.

Parte II
Seconda Parte

Capitolo 4

Ruote a denti dritti

4.1 Introduzione

L'obiettivo di questo capitolo è presentare il metodo con cui si è raggiunta la realizzazione di una mesh 2D per ruote a denti dritti.

La prima sezione spiega quali sono gli input del programma e come si ottiene la geometria del profilo del dente.

La sezione successiva spiega come si è impostato l'algoritmo di mesh e perchè si è scelto di utilizzare una mesh strutturata. Successivamente si presenta come si ottiene la mesh 2D e infine 3D.

Nell'ultima sezione si presentano dei risultati del codice assieme alle prestazioni del programma.



Figura 4.1. Ruota a denti dritti

4.2 Parametri in input

Per caratterizzare una dentatura, è necessario inserire i parametri della dentatura per ottenere la geometria del dente, ed i parametri della mesh.

I parametri della mesh sono ciò che caratterizza questo codice rispetto al codice commerciale di riferimento.

Parametri della dentatura:

- numero di denti
- modulo
- angolo di pressione
- diametro di troncatura esterna
- diametro di radice
- informazioni relative all'utensile
- tipo di raccordo del dente
- spessore del dente alla circonferenza primitiva
- larghezza di fascia
- diametro interno/esterno a seconda che sia una ruota esterna o interna

Parametri della mesh:

- numero di elementi sul profilo evolvente
- numero di elementi sul raccordo
- numero di elementi sul tratto di rim

Il significato dei parametri della mesh verrà esplicitato successivamente. Con questi parametri la geometria della dentatura è completamente definita.

La prima sezione del codice elabora questi dati al fine di ottenere la geometria del profilo del dente, costituita da: tratto di raccordo e dal tratto di evolvente.

Una caratteristica che si è rivelata importante nella generazione della mesh, è il tipo di raccordo del dente. Le tipologie sono due e vengono definite "full fillet" e "flat fillet".

4.3 Geometria del vano

Calcolare un profilo evolvente, nota l'equazione è un'operazione abbastanza semplice, come pure ruotarlo opportunamente al fine di ottenere un dente. Questo discorso vale per il caso ideale in cui non ci sono microcorrezioni.

La parte più critica di una dentatura in questi casi, è il raccordo. Per svincolare l'attività di tesi da una attività di studio approfondito della geometria delle ruote dentate, che avrebbe richiesto molto tempo, si è scelto di utilizzare un tool preesistente in grado di calcolare la geometria del vano di una ruota dentata a denti dritti.

L'output di questo tool è stato opportunamente modificato dopo una sua lettura approfondita al fine di essere sicuri di non intaccare il calcolo della geometria. Le modifiche hanno riguardato il calcolo del numero di elementi e l'estensione del vano dente. Il tool calcola solo metà del profilo

precedente, i motivi risiedono nella facilità di gestire una struttura di questo tipo. La mesh di metà del dente sarebbe impossibile utilizzando tutto il vano.

Di seguito, sono presentati due esempi di output della funzione che calcola la geometria del vano al variare del tipo di raccordo.

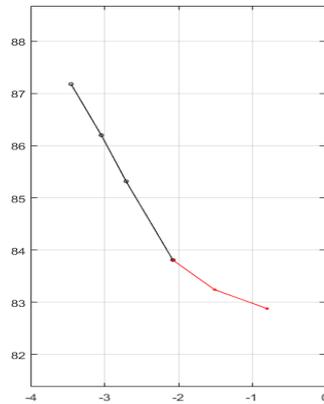


Figura 4.2. Vano full fillet

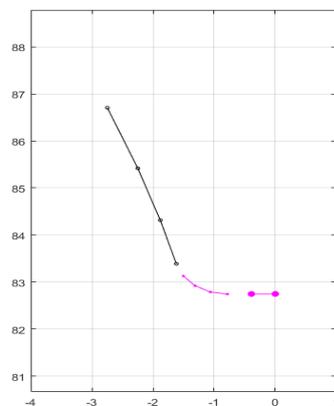


Figura 4.3. Vano flat fillet

Nel caso di un raccordo flat fillet si può osservare che è simile al full fillet ma sia presente un tratto "piatto" aggiuntivo. Il tool gestisce il tipo di raccordo assegnando di default il modello a 5 zone al caso flat root e il modello a 3 zone al caso full fillet.

L'assegnazione di default non è definitiva per il caso flat root, il tool realizza un controllo sulla dimensione degli elementi che sarebbero generati, basandosi sulla dimensione relativa rispetto agli elementi adiacenti del raccordo e ad un fattore di discriminazione del modello che può essere imposto dall'utente, nel caso che voglia sbilanciare la scelta del modello verso il 3 oppure il 5 zone.

4.4 Mesh 2D - introduzione

4.4.1 Anatomia della mesh

In figura, è presentata la mesh di metà dente, sono identificati i principali bordi che la caratterizzano: rim, fillet, profile, tip, inner.

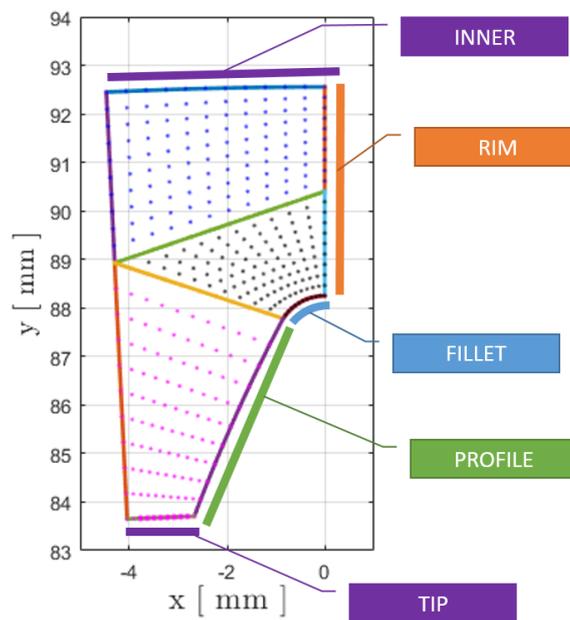


Figura 4.4. Mesh di metà dente - Anatomia

4.4.2 Mesh di riferimento

L'obiettivo della generazione di una mesh parametrica è ottenere una mesh in cui è facile variare il numero di elementi sui tratti di raccordo, sul tratto evolvente, sul tratto di rim.

Il riferimento iniziale fornito è stato il template di un programma che realizza mesh di ruote dentate. Questo codice commerciale sfrutta solamente tre template, che sono presentati di seguito:

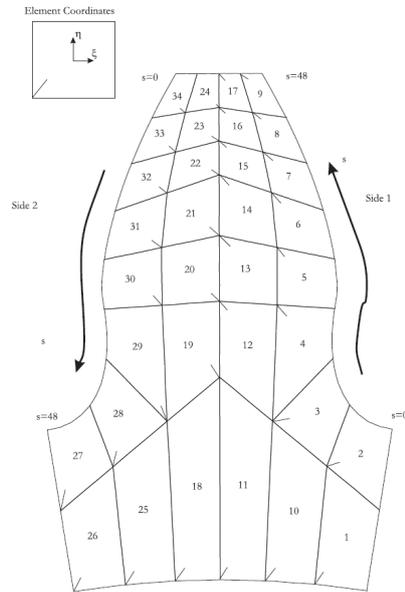


Figura 4.5. Template mesh 1

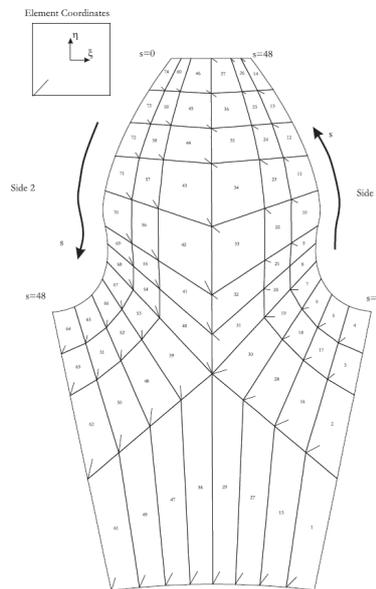


Figura 4.6. Template mesh 2

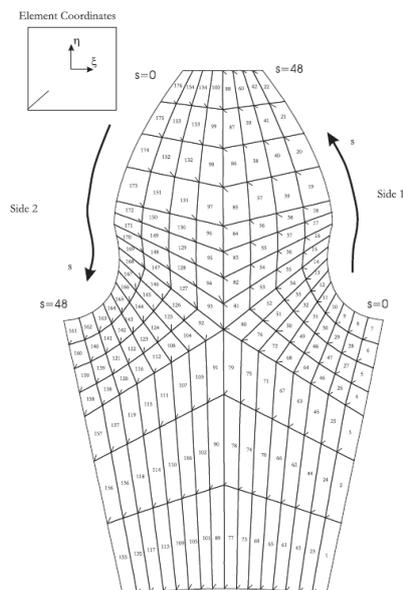


Figura 4.7. Template mesh 3

Come si può osservare nelle figure, i vari template si distinguono per il numero di elementi generati.

Una caratteristica importante è la presenza dell'elemento di esempio in alto a sinistra. Esso ha dei nodi ma possiede anche un proprio sistema di riferimento. Osservando il template, è chiaro che tutti gli elementi "quadrilaterali" sono deformazioni di questo elemento di base.

Dalle figure dei template emerge quella che, forse, è la caratteristica più importante: l'ordine. La mesh è strutturata ma gli elementi sono numerati in sequenza, in base alla posizione. L'obiettivo dell'attività di tesi non è stato la realizzazione di qualcosa di completamente nuovo, quanto la rielaborazione di idee già presenti. Va specificato che oltre a questi template non era disponibile altro, né codici, né documentazione che spiegasse come fossero stati ottenuti.

Un'altra informazione che si ricava guardando il template, è la presenza di una coordinata curvilinea "s" che ha origine in un punto del raccordo e vede l'ultimo punto nel tratto evolvente.

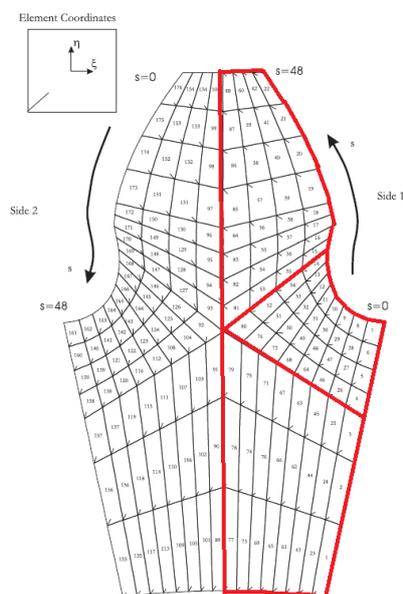


Figura 4.8. Template mesh 3 - zone

L'ultima caratteristica che si può estrarre confrontando i tre template è la presenza di tre zone, o blocchi. Questo pattern ha permesso di vedere la simmetria nell'organizzazione della mesh nelle due metà del dente, inoltre ha fornito le basi per formulare il concetto di "main nodes".

L'utilizzo di questi template è limitante, soprattutto se si considera che per effettuare analisi agli elementi finiti è necessario combinare la mesh della dentatura con quella dell'albero.

A meno di usare tecniche particolari, presenti nei maggiori software commerciali, avere mesh conformi è una caratteristica importante, nella soluzione di un problema FEM.

Questi template si rivelano limitanti quindi per il numero di elementi che presentano al lato "inner" ma anche per il fatto, come in Fig.4.8, di inserire un numero di elementi elevato in zone in cui non è necessario. Il risultato è che si aumenta il costo computazionale senza avere vantaggi.

4.4.3 Strutture della mesh

A livello pratico, realizzare una mesh consiste nella creazione di strutture matriciali contenenti informazioni sui nodi e sugli elementi. Gli elementi nel caso di una dentatura in tre dimensioni, sono di tipo 1D, 2D, 3D. Ogni nodo è qualificato da un indice e dalle sue coordinate, in uno o più sistemi di riferimento.

Sono stati inoltre calcolati i gruppi, essi sono oggetti composti da elementi 2D aventi la caratteristica di collegare lo stesso tipo di bordo della sezione tra due sezioni del dente diverse.

Le rappresentazioni grafiche, in questa tesi, sfruttano principalmente gli elementi 2D della mesh, e la matrice dei nodi globale.

4.4.4 Sequenza di operazioni per ottenere la mesh della ruota

La mesh della ruota avviene per fasi:

1. Mesh 2D di metà sezione del dente
2. Mesh 2D dell'intera sezione del dente
3. Mesh 3D manipolando la mesh 2D

Queste fasi non sono indipendenti tra di loro, quindi nella stesura dello script è stato necessario raggiungere non solo l'obiettivo di calcolare la mesh di metà sezione del dente, ma anche ottenere delle strutture che, al variare dei parametri di mesh, non compromettessero la realizzazione della mesh 2D dell'intera sezione del dente.

Una nota a parte è l'attributo "2D", nel caso di ruote spiroconiche, la mesh giace su un cono, quindi nel sistema di riferimento cartesiano i punti hanno tre coordinate. Nel sistema di riferimento dei parametri della superficie, che sono solo due, le coordinate invece sono due e sono sufficienti. Il passaggio da un sistema di riferimento all'altro avviene tramite le equazioni parametriche della superficie conica.

4.5 Mesh 2D - metà sezione

4.5.1 Implementazione - modelli a 2, 3, 5 zone

Sono stati realizzati tre modelli di mesh. Il tipo di mesh è Multi-Block poichè non sarebbe possibile con un solo blocco riuscire a imporre il numero di elementi sul raccordo, sul profilo evolvente e sul rim.

Inizialmente, il modello con cui si voleva realizzare la mesh era a due zone, quindi il dominio veniva suddiviso in due blocchi, come si vede in figura:

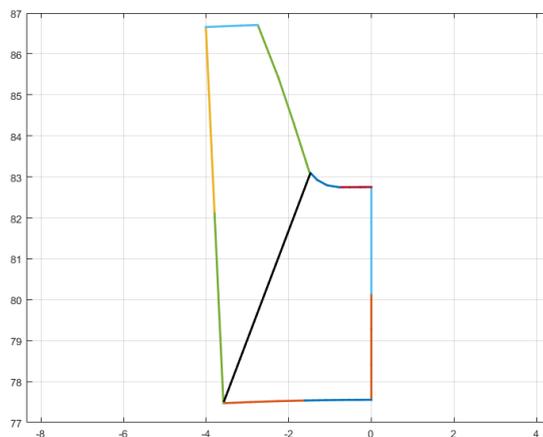


Figura 4.9. Modello a 2 zone

La criticità di questo modello, che è la ragione per cui è stato abbandonato, è la bassa qualità degli elementi che vengono generati specialmente nella parte in basso a sinistra e vicino all'inizio del raccordo.

Si tratta di un modello molto semplice da gestire però: la complessità di realizzazione di un algoritmo di questo tipo è inferiore rispetto ai modelli con 3 e 5 zone.

Questi modelli sono stati sviluppati successivamente ispirandosi ai template del codice commerciale di riferimento. Il modello a 3 zone è stato pensato per le dentature con raccordo full fillet, mentre il modello a 5 zone per il modello flat fillet. La gestione del dominio è diversa rispetto al modello a due zone, ma riserva delle similarità.

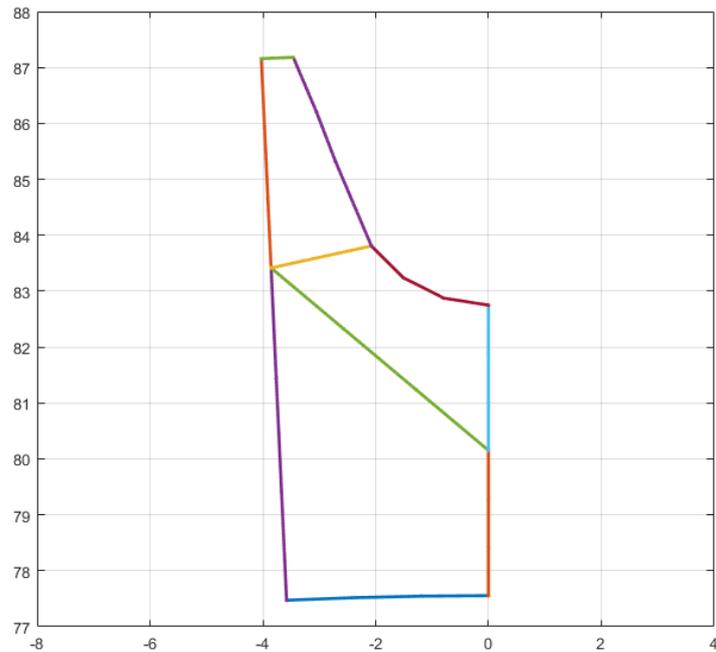


Figura 4.10. Modello a 3 zone

Come si può osservare in Fig.4.10, il modello a 3 zone è un modello a 2 zone a cui è stata aggiunta una zona nella parte inferiore.

Il risultato è che si elimina il difetto del modello a due zone separando il rim in due lati: "top" e "bottom". Il numero di elementi del lato "rim top" viene mantenuto fino al lato "tip". Il numero di elementi del profilo evolvente è pari a quello del lato sinistro superiore, mentre quello del raccordo è pari al numero di elementi sul lato "inner". Infine, il numero di elementi del lato "rim bottom" è pari a quello del lato sinistro inferiore.

Il modello a 5 zone è stato sviluppato per il tipo di raccordo "flat root", per non trascurare questo tratto del raccordo che altrimenti sarebbe male approssimato da un modello a 3 zone. Sostanzialmente si tratta di un modello a 3 zone a cui vengono aggiunte due zone a destra.

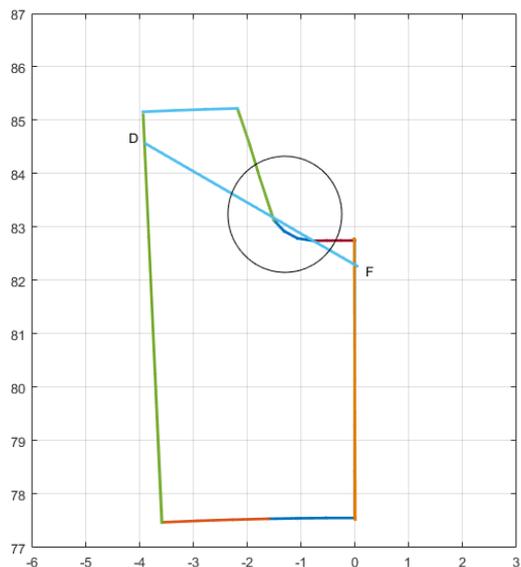


Figura 4.11. Modello a 3 zone - criticità intersezione

La caratteristica di questo modello è che il punto D è meno vincolato rispetto alla dimensione del lato rim, quindi gli elementi mantengono una qualità accettabile e, soprattutto, non si verificano intersezioni tra i tratti DF e ABC del dominio. Il risultato è un dominio con zone da meshare molto più robusto, che non diventa instabile in casi limite di rim molto bassi o alti.

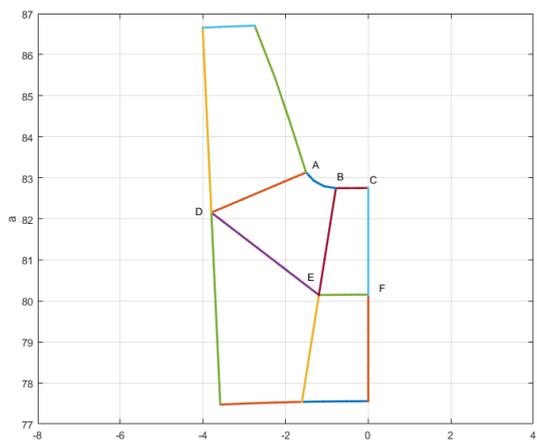


Figura 4.12. Modello a 5 zone

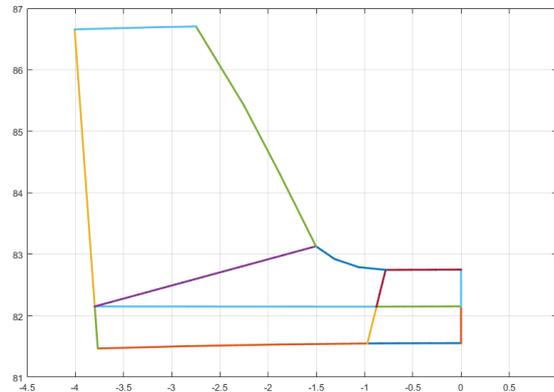


Figura 4.13. Modello a 5 zone - caso rim basso

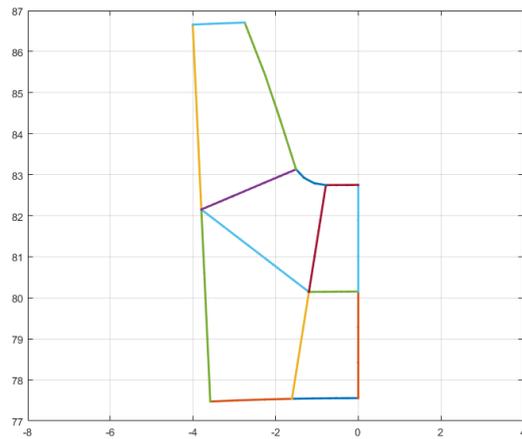


Figura 4.14. Modello a 5 zone - caso rim alto

4.5.2 Main nodes

Lo step successivo alla generazione della geometria del vano e la scelta del tipo di modello, è la generazione degli altri punti del dominio. Questi punti sono i cardini su cui viene effettuata la mesh. La creazione di una struttura di questo tipo permette di disaccoppiare l'azione della funzione che genera la mesh dall'oggetto che si vuole meshare.

Nella funzione che realizza la mesh è necessario, ovviamente, passare anche i punti dei bordi del profilo, del raccordo, dell'eventuale tratto di root. potrebbero anche essere dei bordi, definiti per punti, anche di una turbina piuttosto che di una sezione di una trave, piuttosto che la sezione di un componente che si ottiene per estrusione di una sezione o sweep, come nel caso delle spiroconiche

ad esempio.
 Seguono alcuni esempi di applicazioni:

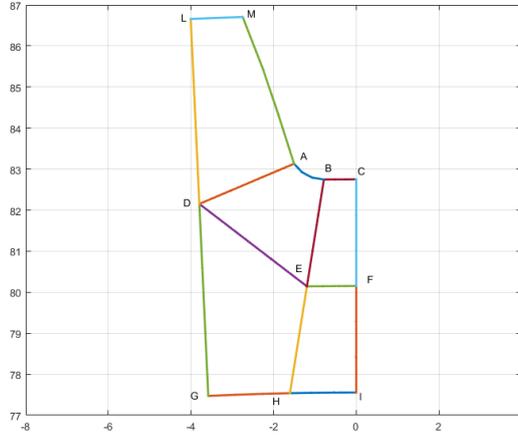


Figura 4.15. Modello a 5 zone - main nodes

La denominazione dei nodi generale è riferita al modello a 5 zone, di cui il modello a 3 zone è una semplificazione.

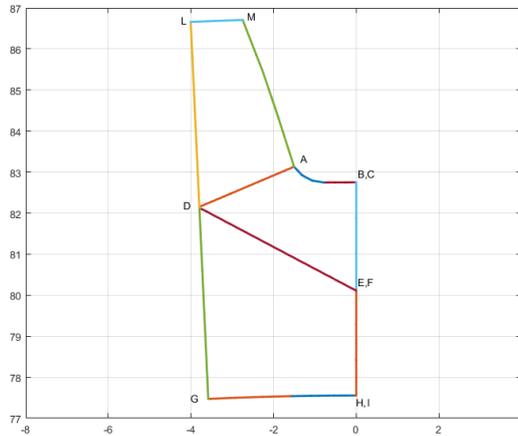


Figura 4.16. Modello a 3 zone - main nodes

Nelle figure precedenti si osserva che le forme delle zone possono avere variabilità elevata. Per questo script, ci sono tuttavia dei limiti. La realizzazione della mesh è garantita quando le zone sono tutte convesse, o più elegantemente il dominio della zona risulta semplicemente connesso.

I main nodes sono importanti perchè è a partire da questi che vengono generati i bordi delle zone, per definizione sono situati ai vertici di ogni zona. Questa caratteristica determina che l'algoritmo che realizza il calcolo dei nodi interni ad ogni zona, non sia dipendente dal dominio. In questa applicazione, ovviamente, sono passati anche i punti calcolati di profilo evolvente, raccordo, lato inner, lato tip, altrimenti si avrebbe una mesh costituita da zone delimitate da segmenti. Allo stato attuale, come detto, solo i punti del profilo evolvente, del raccordo, del lato tip, del lato inner sono passati in input alla funzione che calcola i nodi interni ad ogni zona. Questo non significa che ad esempio anche i tratti DE, AD, ad esempio, non possano essere passati in input, facendo le opportune modifiche al codice. Ne risulta quindi un prodotto molto flessibile. Il codice potrebbe essere modificato per impostare manualmente i nodi D, E, F, ma si è preferito, nell'ambito di tesi, di realizzare un tool almeno inizialmente, automatico. Le possibilità di personalizzazione sono, dunque, numerose. Questa caratteristica sarà evidente nei capitoli relativi alle ruote spiroconiche.

4.5.3 Vincoli sui main nodes

La robustezza è una caratteristica importante per questo tool. I main nodes, devono rispettare sempre un certo pattern in termini di posizione topologica. Se questo vincolo non è rispettato, la mesh nella maggior parte dei casi è realizzata ma non si ottiene il dente. Dei vincoli sulla posizione topologica, sono stati dunque imposti, e nel caso non fossero rispettati, il punto viene automaticamente riposizionato. Questa caratteristica non influisce sulla customizzabilità dei main nodes della mesh.

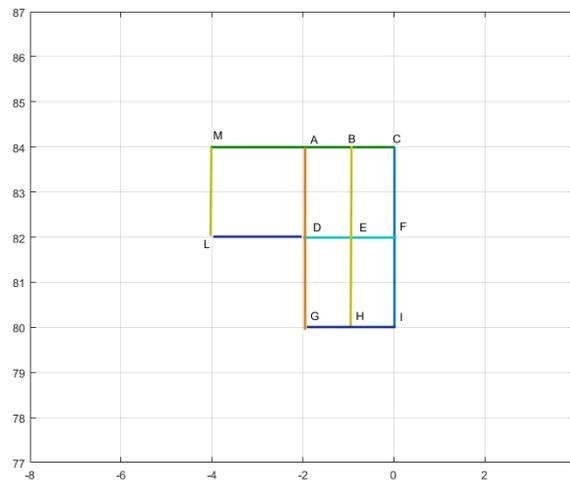


Figura 4.17. Esempio 1 - struttura di elementi quadrati

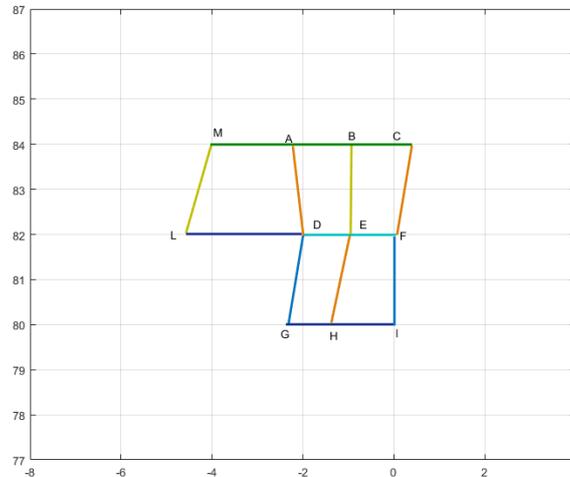


Figura 4.18. Esempio 2 - struttura di quadrilateri

Nelle figure precedenti, sono visibili esempi di strutture che il tool può meshare. La caratteristica comune è la topologia e la denominazione dei main nodes.

4.5.4 Algoritmi di calcolo della matrice dei nodi

A livello cronologico, è stata sviluppata inizialmente la parte del tool per la realizzazione della mesh parametrica di ruote dentate a denti dritti, successivamente la parte relativa alle elicoidali, infine la parte relativa alle spiroconiche.

Andare in ordine di difficoltà è stato importante perchè le dentature spiroconiche hanno una complessità geometrica che richiede l'utilizzo di più strumenti di calcolo rispetto alle dentature a denti dritti. Approcciarle subito, significherebbe trovarsi a dover risolvere problemi complicati non solo in termini di algoritmi di mesh ma anche di gestione delle strutture geometriche.

Approcciando le dentature a denti dritti per prime, ha permesso di focalizzarsi sulle caratteristiche della mesh 2D nel piano cartesiano (x,y) , Una volta trovato un algoritmo funzionante è stato possibile studiare alcune delle sue proprietà, che durante la stesura non erano balzate subito all'occhio ma si sono rivelate determinanti nella stesura di un algoritmo di nuova concezione per le ruote spiroconiche.

La scoperta di alcune relazioni tra i nodi della mesh 2D ha permesso di riformare, nel caso delle dentature spiroconiche, il codice per il calcolo della matrice dei nodi e di una parte degli elementi finiti. Il risultato è stato un codice: `fz`

- Più snello: la dimensione delle strutture dati coinvolte è inferiore
- Più leggibile: il codice è più breve e sfrutta funzioni semplici
- Meno complesso: l'assegnazione di alcuni nodi alle strutture di elementi finiti può richiedere la definizione di molti casi particolari, caratteristica superata.

più snello in termini di dimensioni delle matrici utilizzate,

Nel capitolo relativo alle dentature spiroconiche questi tratti saranno presentati in maniera più specifica. In conclusione, lo sviluppo di un algoritmo in un ambiente più semplice, ha permesso l'estrazione di alcune caratteristiche che sono state determinanti nella realizzazione di un nuovo algoritmo.

Il motivi della generazione di un nuovo algoritmo sono stati essenzialmente pratici. Attualmente, il codice di mesh che calcola l'intera mesh delle dentature spiroconiche ha un tempo di esecuzione confrontabile, se non inferiore, rispetto a quello delle dentature a denti dritti od elicoidali, nonostante compia molte più operazioni a causa della geometria più complessa.

Algoritmo di calcolo della matrice dei nodi - non ottimizzato

La logica di funzionamento è la seguente: nel dominio sono presenti 3 o 5 zone, l'algoritmo cicla sulle zone e per ognuna di esse calcola una parte della matrice dei nodi. Le matrici di ogni zona vengono assemblate opportunamente e si ottiene la matrice dei nodi del dominio costituito da metà del dente.

Al fine di rendere robusto il codice, è stata studiata una logica per i cicli, che rende irrilevante il numero di elementi desiderati su ognuno dei bordi della mesh. Le caratteristiche di questa logica sono essenzialmente la definizione di sistemi di riferimento a livello locale e globale sia a livello topologico che cartesiano.

Quando il codice cicla su una zona, gli indici scorrono opportunamente sia sul sistema di riferimento cartesiano che topologico.

Questo tipo di logica è molto simile a quella che compierebbe una persona che volesse compilare la matrice dei nodi guardando un dominio in 2D disegnato su un foglio di carta, che è ciò che è realmente successo visto che la letteratura in termini di mesh generation non è esattamente ampia.

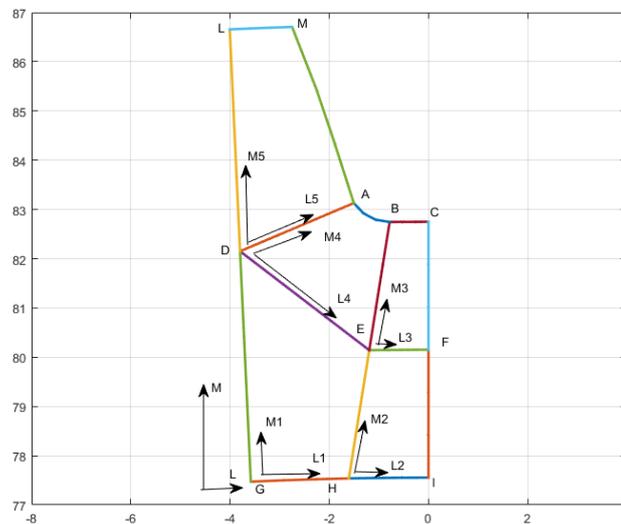


Figura 4.19. Zone della mesh, main nodes, e sistemi di riferimento

Come si vede in Fig.4.19, il dominio su cui calcolare la matrice dei nodi, è una metà del dente. La regione è divisa in zone. Tutti i punti del dominio possono essere espressi in un sistema di riferimento cartesiano (x,y) ma anche in un sistema di riferimento globale (l,m) basato sulla topologia della discretizzazione, che dipende direttamente dal numero di elementi richiesti sui vari bordi del dominio. Ogni zona, inoltre, ha un proprio sistema di riferimento locale topologico. La versione non ottimizzata dello script sfrutta dei cicli for aventi indici che scorrono sui vari assi dei vari sistemi di riferimento. L'azione avviene in maniera sinergica e in contemporanea per ogni zona. Oltre a calcolare le matrici dei nodi, questa versione, calcola anche la matrice contenente i nodi che stanno sulla frontiera della regione, che saranno utili nel calcolo degli elementi finiti 1D. La distribuzione dei punti è equispaziata, ma anche qui lo script è altamente personalizzabile. Avendo l'obiettivo di creare un tool automatico si è optato per la distribuzione equispaziata poichè è la più stabile ma volendo migliorare il determinante jacobiano degli elementi della mesh si può arrivare a posizionare i punti da tastiera. La definizione di determinante dello jacobiano di un elemento è poco chiara in questa trattazione. Le interpretazioni sono molteplici, i vari software commerciali utilizzano definizioni diverse. Ad esempio, Hypermesh, per jacobiano intende il rapporto tra il minimo ed il massimo dei determinanti calcolati in particolari punti dell'elemento: i punti di integrazione, altrimenti noti come Gauss points.

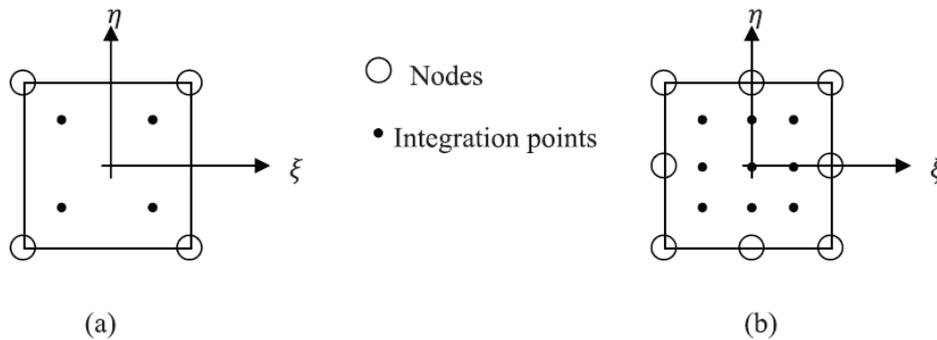


Figura 4.20. Nodi e punti di integrazione in un elemento "quad"

In Fig.4.26 sono presenti due tipi di distribuzioni di punti di integrazione. Il sistema di riferimento è quello ideale, questi elementi nel sistema di riferimento cartesiano, ad esempio, potrebbero risultare deformati, con il risultato di avere un determinante diverso da uno. Lo stadio successivo del tool, oltre a comprendere geometrie più complesse, potrebbe anche introdurre calcoli di questo tipo, ma anche la valutazione della distribuzione delle aree degli elementi.

$$J^{elem} = \frac{\max(J^{GP})}{\min(J^{GP})}$$

Per come è definito lo jacobiano di un elemento, risulta evidente che in Hypermesh gli estremi a cui può arrivare questo indicatore siano -1 e 1.

Dopo questa digressione sono presentati alcuni risultati dell'algoritmo di calcolo dei nodi per dentature a denti dritti esterne ed interne al variare del numero degli elementi richiesti ma anche dei parametri delle dentature.

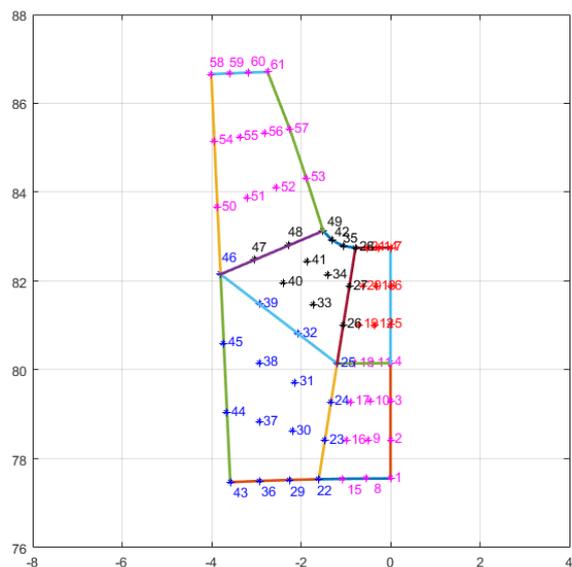


Figura 4.21. Nodi - set 1

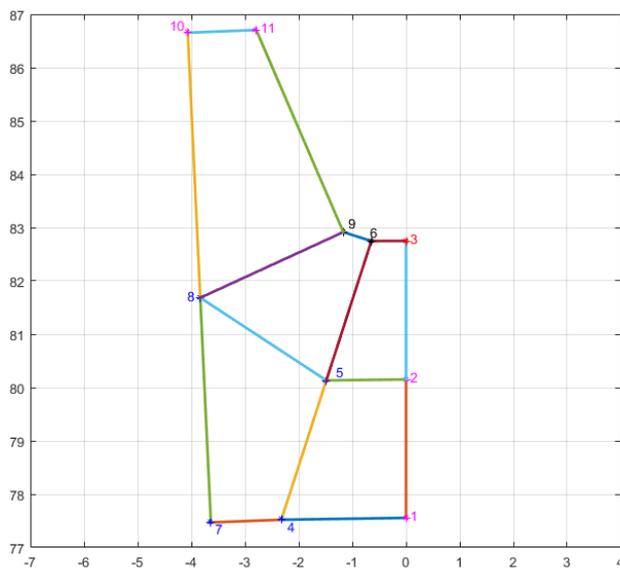


Figura 4.22. Nodi - set 2

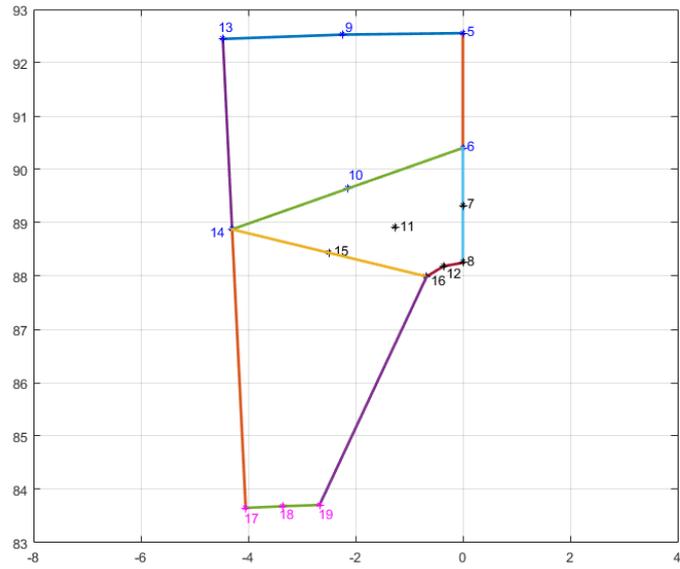


Figura 4.23. Nodi - set 3

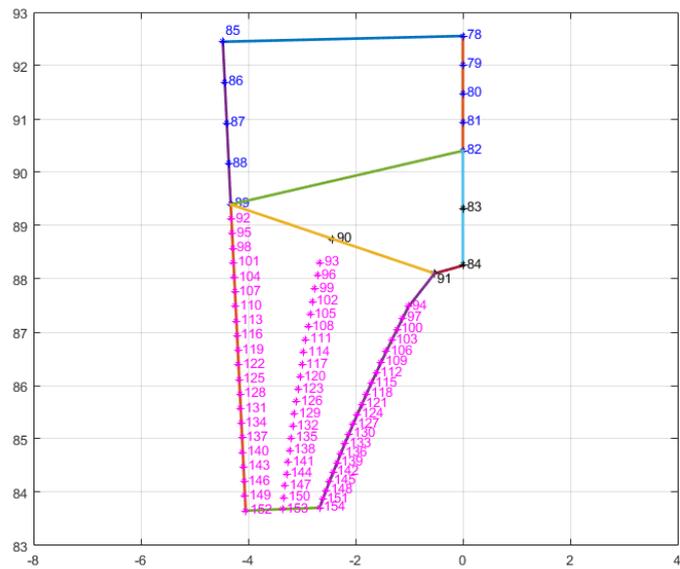


Figura 4.24. Nodi - set 4

Tempi di sviluppo

Si è ritenuto che per un'applicazione così specifica, fosse più veloce in termini di tempo di realizzazione, creare un proprio algoritmo piuttosto che ispirarsi ad algoritmi da altri che sarebbero poi dovuti essere adattati.

In termini di time management, prima dello sviluppo dell'algoritmo, si è ritenuta vera la relazione:

$$T_{ricerca} + (T_{comprensione} + T_{adattamento})N + T_{confronto\ prestazioni} > T_{sviluppo\ di\ soluzione\ dedicata}$$

dove:

N = numero di algoritmi esterni presi in considerazione

Non ci sono controprove a sostegno di questa tesi, ma dato il tipo di applicazione, a posteriori si ritiene ancora sensata la relazione, per i seguenti motivi:

- Lo sviluppo di una soluzione dedicata ha permesso di ottimizzare il codice.
Nel caso della versione ottimizzata nello script per le dentature spiroconiche, la riduzione di tempo in alcune parti del codice è di almeno un ordine di grandezza.
- Le prestazioni in termini di tempo di calcolo sono buone, l'ordine è dei secondi.
Una considerazione sul tempo di esecuzione: i tempi di calcolo di una mesh non strutturata, anche per un dominio semplice, possono arrivare a superare il minuto.
Non è frequente trovare codici pronti da eseguire in MATLAB, oltre a comprendere la logica si dovrebbe poi implementarli con sguardo critico.

Prestazioni dell'algoritmo

Variare il numero di elementi di un bordo della regione, piuttosto che di un altro, non influisce molto sul tempo computazionale richiesto dalla funzione che calcola i nodi interni, se il numero totale di nodi si mantiene circa costante.

Un'osservazione interessante riguardo all'algoritmo è il fatto che il tempo di calcolo necessario, nonostante la preallocazione delle matrici, variasse in modo tale che al crescere dei lanci su MATLAB, il tempo di esecuzione calasse.

La motivazione trovata per questo comportamento apparentemente "magico", risiede nel fatto che MATLAB sfrutta la memoria occupata dalle variabili del workspace del lancio precedente per velocizzare il salvataggio delle nuove. Utilizzando il comando per pulire totalmente il workspace, questo fenomeno non si ripeté.

Le prestazioni dell'algoritmo sono state presentate in questo punto della trattazione poiché la funzione di calcolo dei nodi interni è quella più dispendiosa in termini di tempo, e quindi la controlla. Nonostante la vettorizzazione anche la parte di rappresentazione è dispendiosa in termini di tempo, ma con un numero di nodi adatto alle applicazioni previste, lo script conclude tutte le operazioni in meno di 10 secondi. Per la funzione di calcolo dei nodi interni il tempo si attese intorno ai 2 secondi mediamente.

I tempi segnalati in questo paragrafo sono poco significativi se non si introduce il fatto che le macchine che effettuano i calcoli hanno delle ottime caratteristiche in termini di hardware. Non è irragionevole ritenere però che lo script possa essere eseguito anche su computer meno potenti senza essere penalizzato eccessivamente.

Questo discorso sulla penalizzazione è meno significativo nel caso del codice ottimizzato tramite vettorizzazione, poiché non ci sono più cicli for.

4.5.5 Algoritmi di calcolo degli elementi 1D, 2D

Un ulteriore motivo per la scelta di una mesh strutturata, oltre alla maggiore snellezza computazionale, è la possibilità di calcolare agevolmente gli elementi della mesh.

Come precisato nei paragrafi precedenti, la versione non ottimizzata dell'algoritmo di calcolo della matrice dei nodi, è in grado di calcolare anche i nodi sui bordi del dominio da meshare.

Avendo queste due strutture dati e potendo visualizzare i nodi nella mesh, è facile vedere un pattern. La generazione delle matrici degli elementi 1D e 2D si ottiene per manipolazione delle matrici dei nodi e dei nodi sui bordi. Questo tipo di soluzione è particolarmente efficiente su MATLAB, quindi a livello di tempo di esecuzione si sta facilmente sotto il decimo di secondo.

Questa caratteristica è stata ricercata in fase di progettazione dell'algoritmo, quella svolta su carta.



Figura 4.25. Convenzione elementi 1D

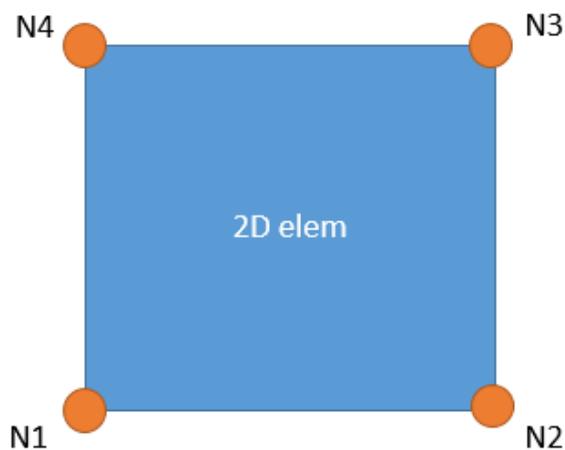


Figura 4.26. Convenzione elementi 2D

Le matrici di output seguono delle convenzioni per quanto concerne la sequenza dei nodi. Se così non fosse, se la mesh fosse esportata in un programma in grado di visualizzarla, gli elementi assumerebbero forme improbabili e i gli indici sui jacobiani assumerebbero valori anomali.

4.6 Mesh 2D - intera sezione

La realizzazione della mesh 2D dell'intera sezione è materia più complessa di ciò che si potrebbe immaginare, per il fatto che ci sono dei nodi in comune tra le due metà del dominio, si tratta dei nodi sul lato meridiano della sezione del dente.

Le conseguenze impattano sul calcolo delle matrici dei nodi e degli elementi finiti. Non è sufficiente, per generare la mesh di questa metà del dente, richiamare l'algoritmo che ha generato nodi ed elementi finiti della prima metà del dente.

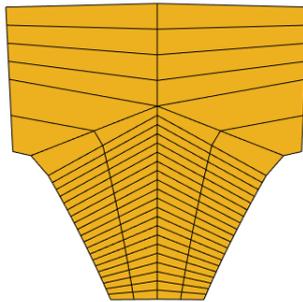


Figura 4.27. Simmetria del dente

A livello metodologico, si è scelto di non modificare l'algoritmo che ha generato la prima metà della mesh. Analizzando la struttura di un dente appartenente ad una ruota a denti dritti è facile osservare che ci sia una simmetria rispetto all'asse meridiano. Per conoscere i nodi dell'altra metà della sezione, è sufficiente dunque manipolare la matrice dei nodi: la si duplica e si cambia il segno dei valori contenuti nella colonna contenente le coordinate x .

Ovviamente, a questo punto, non è sufficiente concatenare le matrici dei nodi per ottenere quella definitiva. Facendo così i nodi sull'interfaccia sarebbero doppi. Un metodo per cancellare i nodi dell'interfaccia duplicati, sarebbe filtrare la matrice controllando i valori delle coordinate x e y . Questo metodo è stato valutato come efficace ma poco robusto, visto che si dovrebbe scegliere un grado di approssimazione. Si è scelto di implementare un nuovo metodo basato sulla numerazione degli indici, più veloce e robusto. Sostanzialmente si sfrutta l'ordine con cui sono memorizzati i nodi nella matrice dei nodi.

Per quanto concerne gli elementi 1D, anche qui la materia è complicata dai nodi duplicati. La matrice dei nodi sui bordi è stata manipolata al fine di escluderli. Successivamente, è stato implementato un algoritmo di concatenazione che permette di salvare in una matrice tutti i nodi sui bordi, in essa, ogni bordo è rappresentato da una riga.

Per quanto concerne gli elementi 2D, è stato sviluppato un algoritmo per manipolare la matrice della prima metà escludendo i nodi duplicati. Il metodo, anche qui, si basa sulle proprietà della topologia della mesh strutturata. Sostanzialmente anche qui si è derivato su carta un pattern per un certo set di parametri di mesh, successivamente il metodo è stato generalizzato al variare del numero di elementi richiesti sui bordi di evolvente, raccordo, rim.

La robustezza di queste manipolazioni è stata testata nelle fasi di rappresentazione della mesh 3D. Non si sono mai viste irregolarità nella mesh.

Una caratteristica importante di questo approccio è che è quasi generale. La stessa funzione viene

utilizzata anche per le dentature elicoidali. Per quanto riguarda le dentature spiroconiche, invece, l'algoritmo di generazione della mesh di una sezione della dentatura è diverso poichè essa non è simmetrica. Nonostante ciò, le proprietà topologiche della mesh sono le stesse di una mesh a denti dritti, quindi alcune parti dell'algoritmo sono state riproposte anche lì.

Per chiarire quali siano le proprietà topologiche, si presenta uno schema, da cui è facile dedurre, che la non simmetria della sezione della dentatura, influisca soltanto sulle coordinate dei nodi della mesh, ma non sulle relazioni tra i nodi.

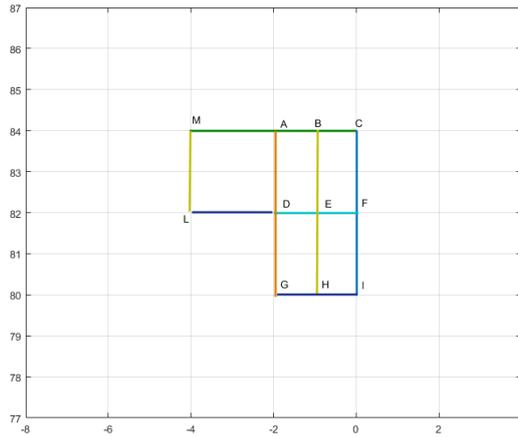


Figura 4.28. Topologia dei nodi della mesh sinistra di una dentatura a denti dritti

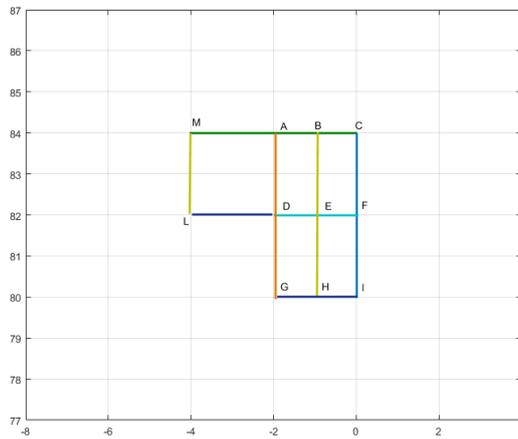


Figura 4.29. Topologia dei nodi della mesh sinistra di una dentatura spiroconica

Come si può osservare, l'immagine è la stessa, perchè non ci sono differenze su questo piano.

4.7 Mesh 3D

La generazione della mesh 3d è caratterizzata da varie sezioni 2D della dentatura inoltre è costituita da elementi 1D, 2D, 3D. Il processo di calcolo dei nodi della mesh 3D, è fondato sull'operazione di estrusione. Sostanzialmente, dopo aver replicato la matrice dei nodi della mesh2D, si modificano le coordinate z della matrice dei nodi in maniera opportuna, in modo da ottenere il numero di elementi voluti lungo la larghezza di fascia della ruota.

Per quanto riguarda il calcolo degli elementi 3D e 2D laterali, si considera la relazione tra gli indici dei nodi di due sezioni 2D diverse. Gli ID dei nodi di tutte le sezioni a diversa z, sono sequenziali. Per calcolarli è sufficiente usare:

$$ID_i = ID_0 + iN_{face\ nodes}$$

dove:

i = indice della sezione estrusa i-esima

ID_0 = ID del nodo generico sulla prima sezione dell'estrusione

$N_{face\ nodes}$ = numero di nodi su una sezione della dentatura ortogonale all'asse di estrusione

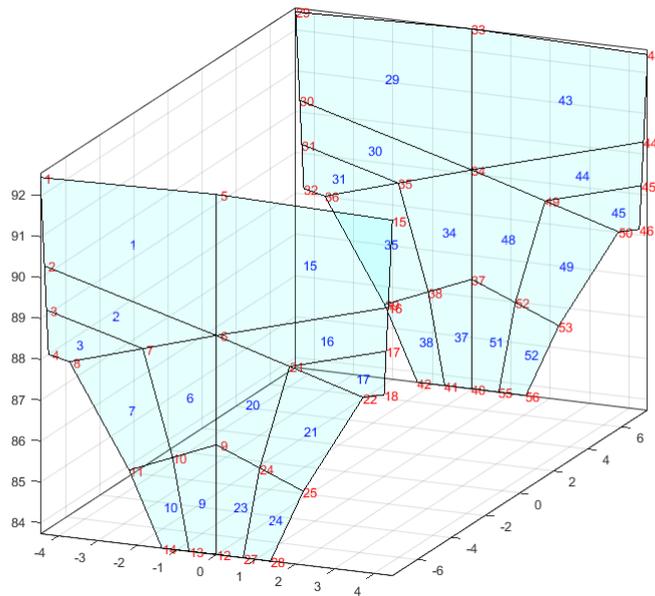


Figura 4.30. Rappresentazione 3D della struttura dei nodi e degli elementi 2D

Per quanto concerne il calcolo degli elementi di superficie laterali, è facile dallo schema estrarre un pattern. Sfruttando la matrice degli elementi 1D della sezione, anche qui basta manipolare opportunamente le matrici, senza fare calcoli od altre operazioni. Il calcolo, anche qui, in termini di tempo e risorse, è molto veloce.

4.8 Risultati

In questa sezione sono rappresentati i risultati di alcuni esempi di dentature.

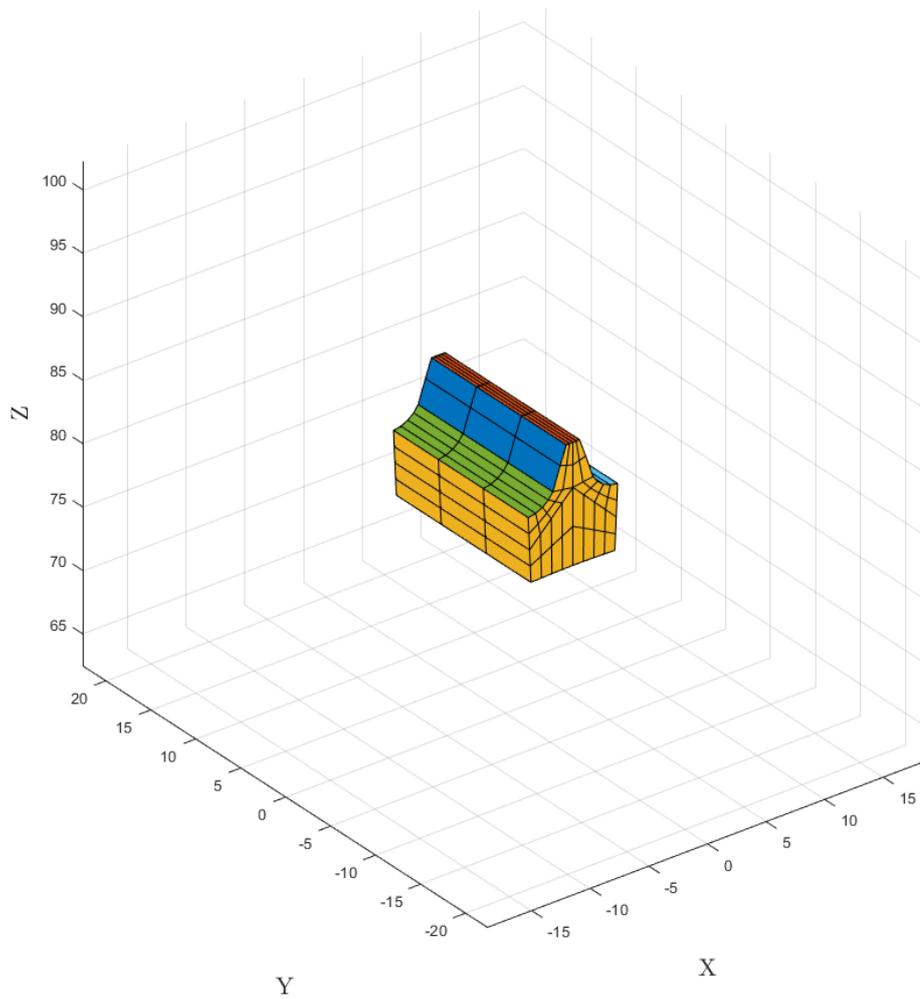


Figura 4.31. Mesh 3D - dente esterno - raccordo full fillet - set di parametri 1

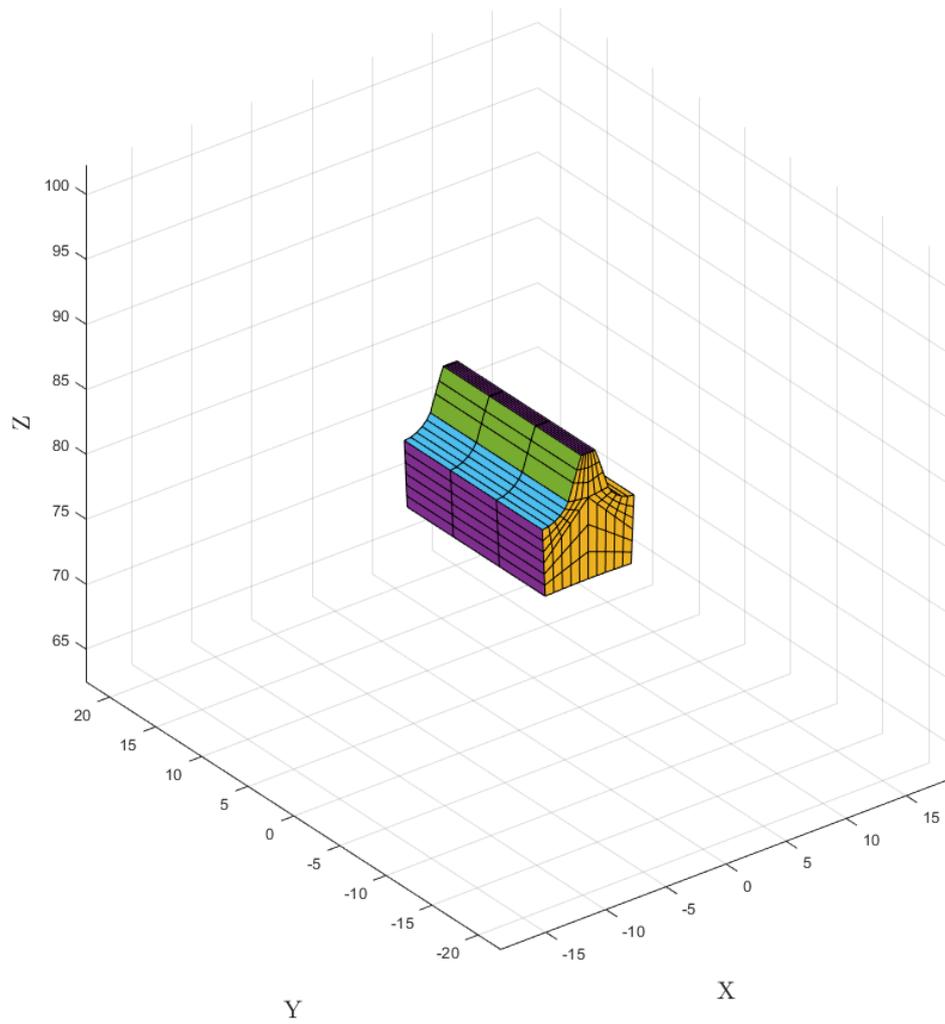


Figura 4.32. Mesh 3D - dente esterno - raccordo full fillet - set di parametri 2

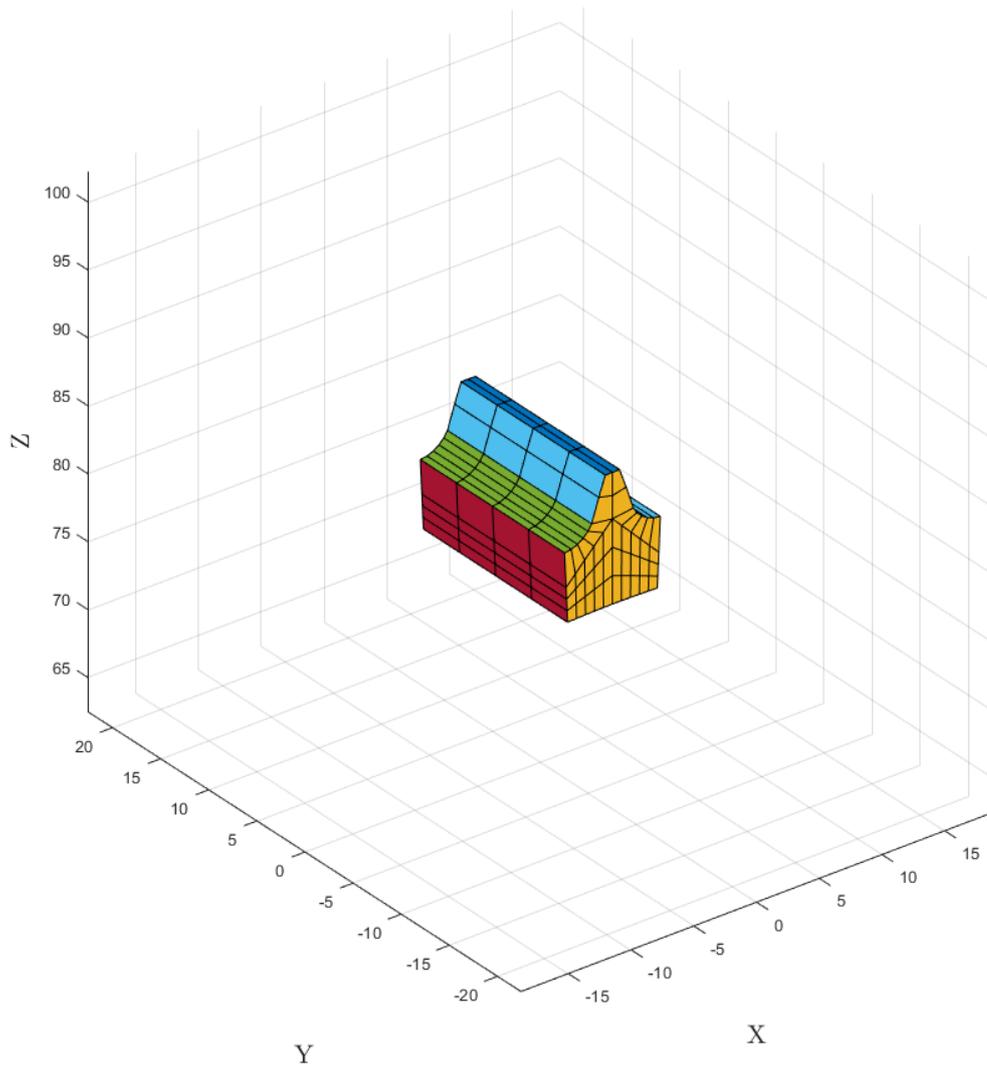


Figura 4.33. Mesh 3D - dente esterno - raccordo full fillet - set di parametri 3

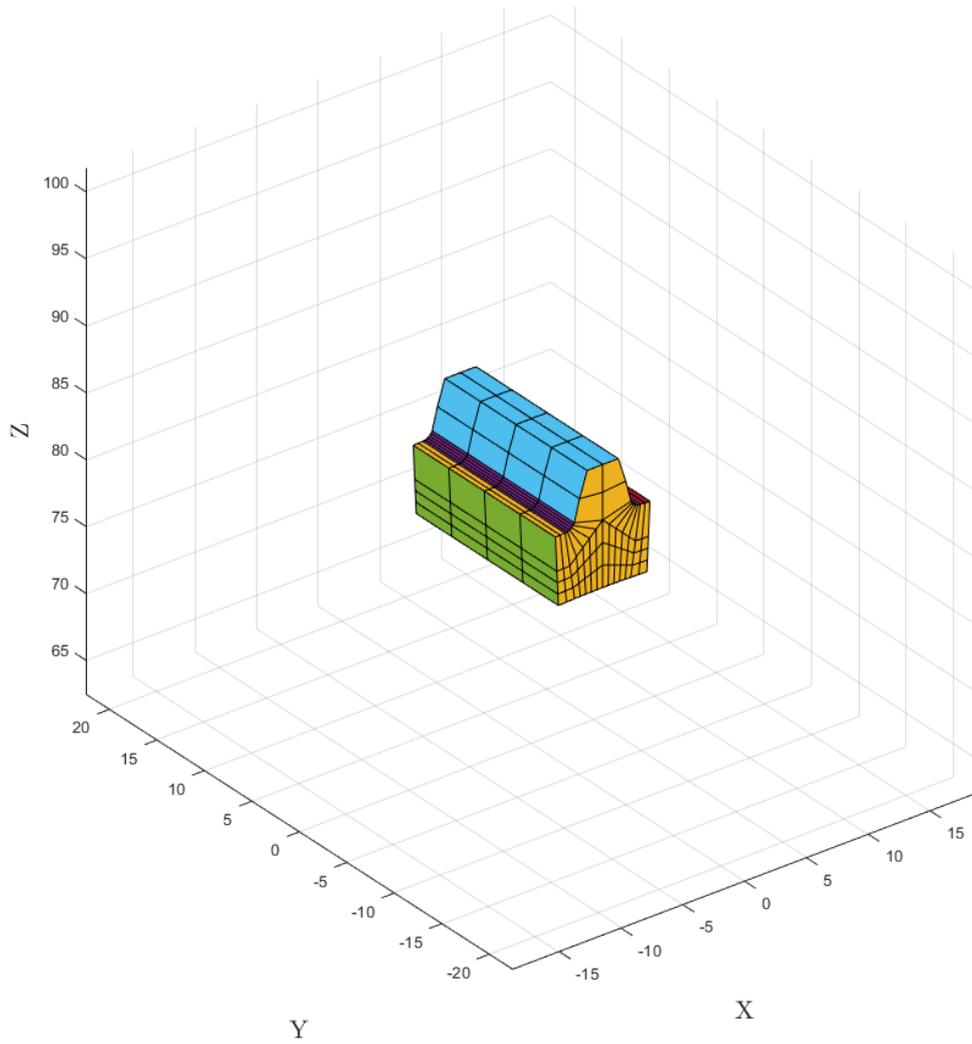


Figura 4.34. Mesh 3D - dente esterno - raccordo flat root - set di parametri 1

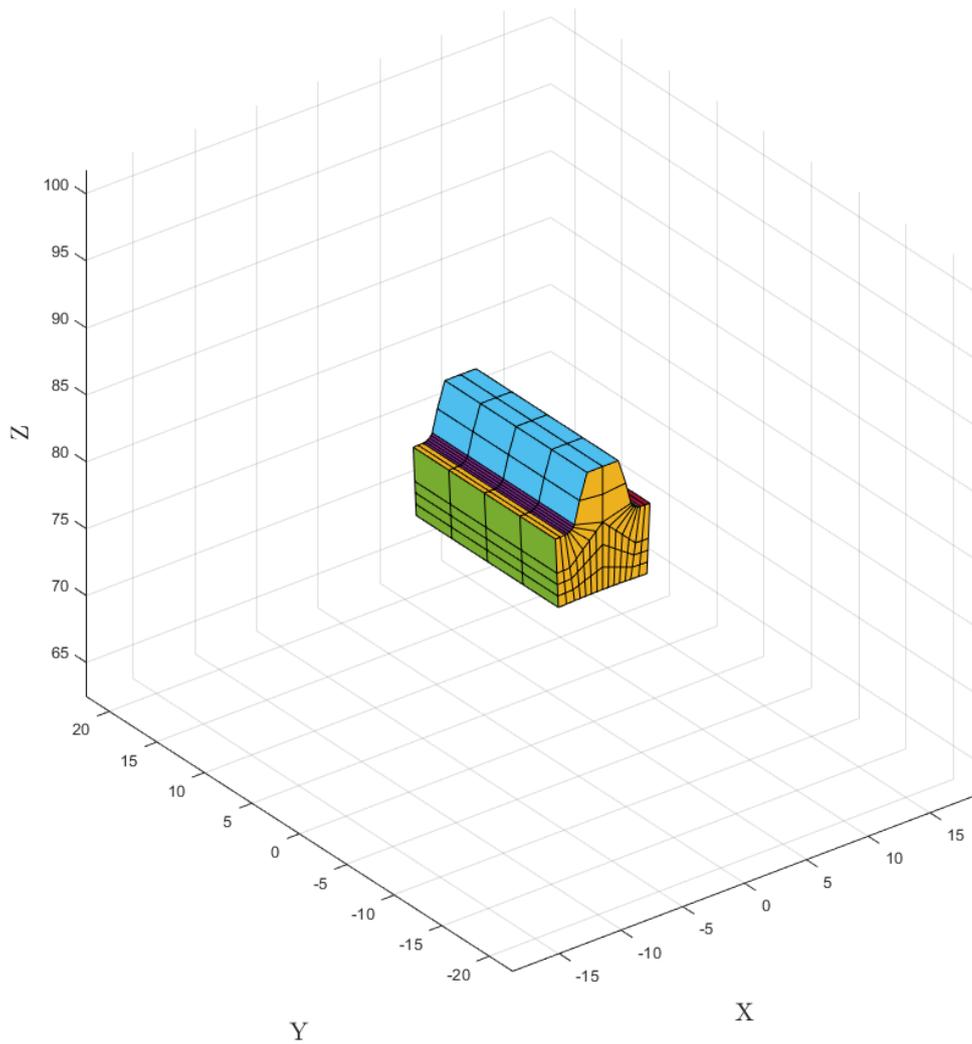


Figura 4.35. Mesh 3D - dente esterno - raccordo flat root - set di parametri 1

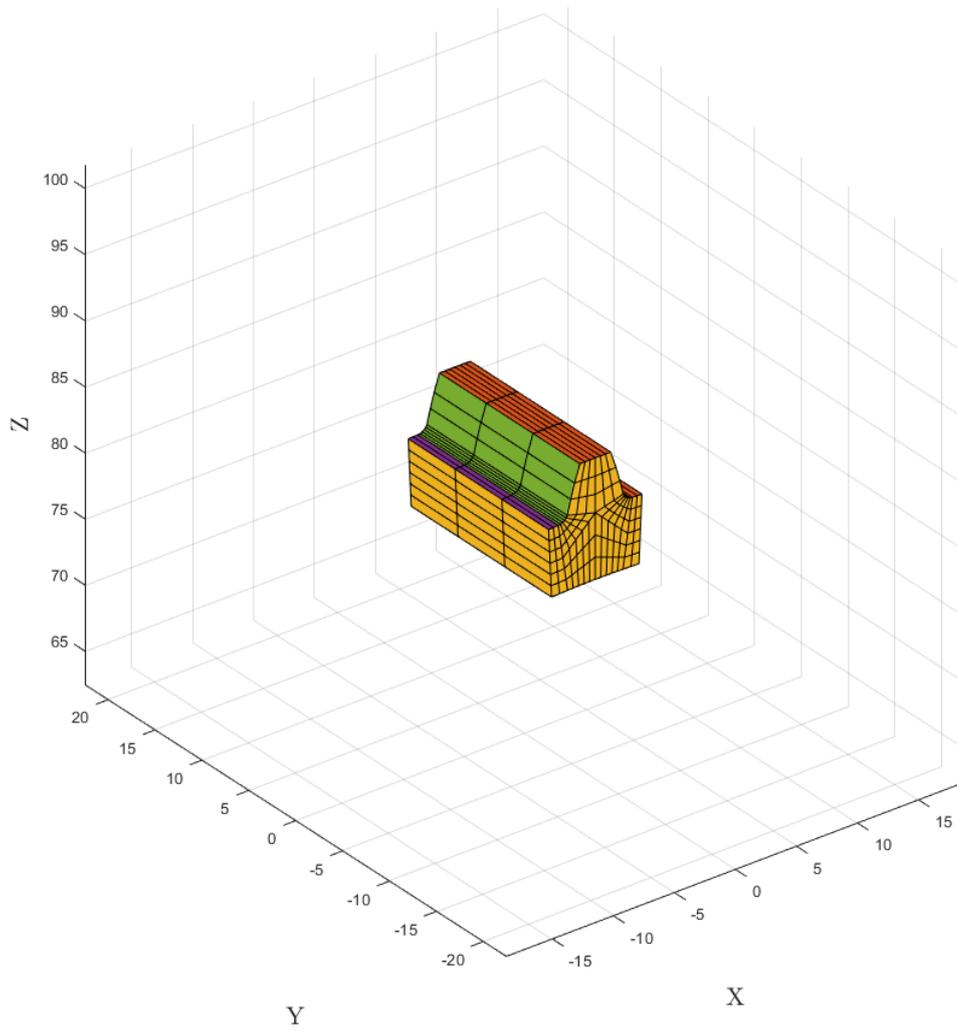


Figura 4.36. Mesh 3D - dente esterno - raccordo flat root - set di parametri 2

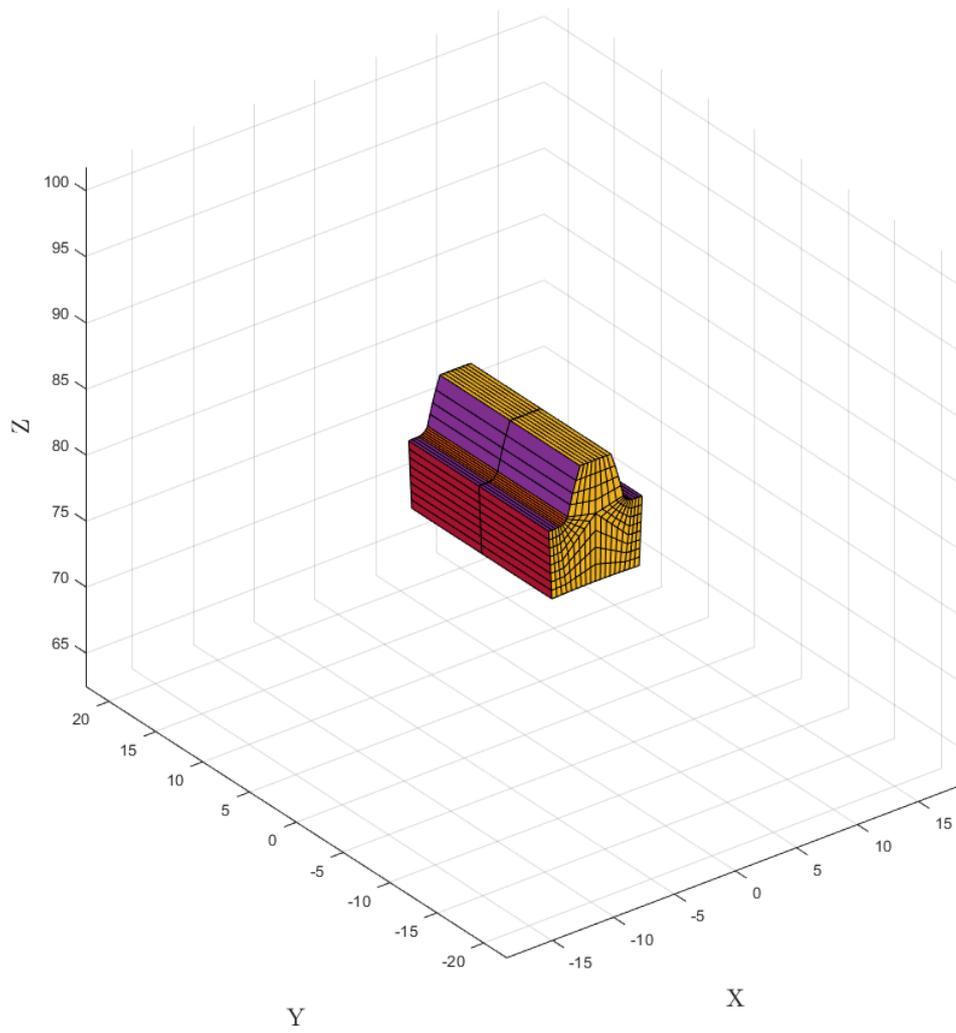


Figura 4.37. Mesh 3D - dente esterno - raccordo flat root - set di parametri 3

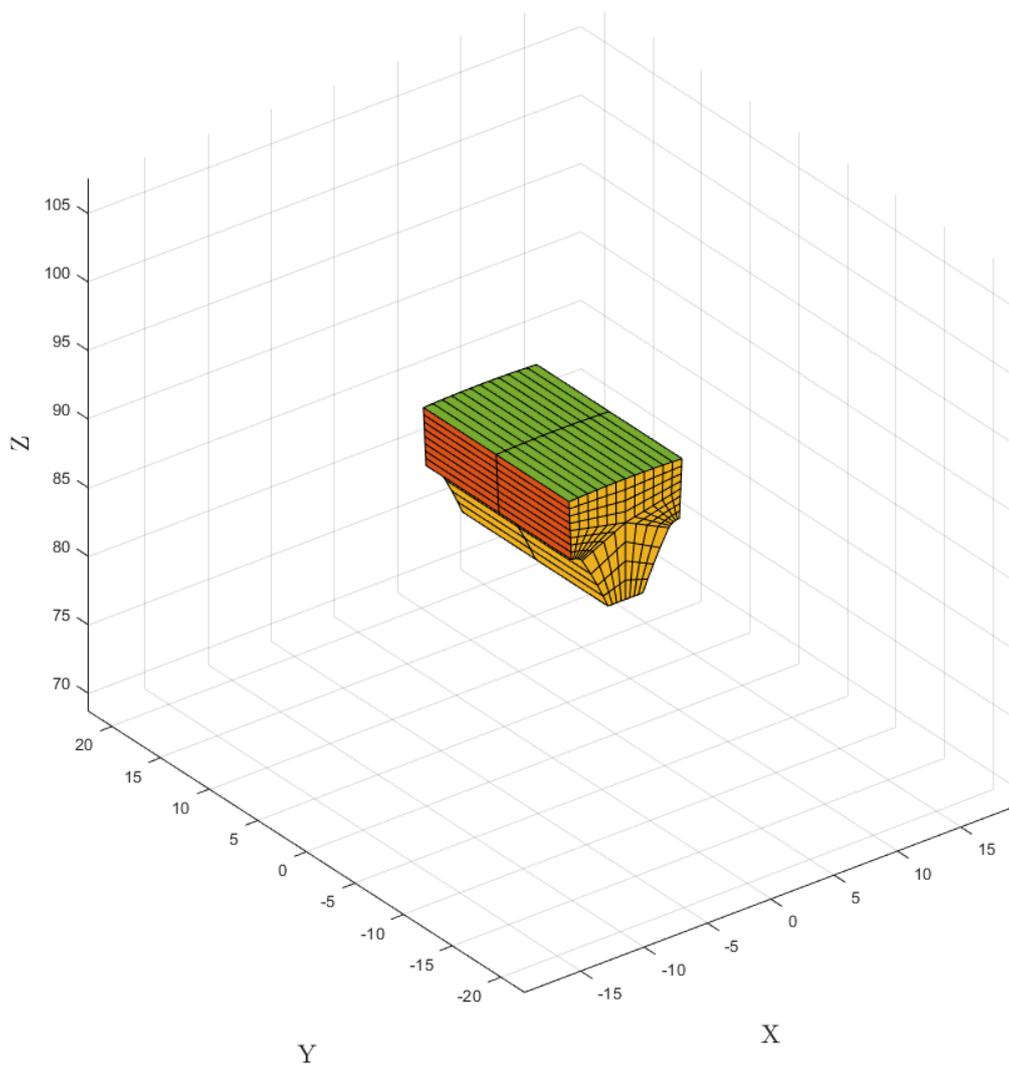


Figura 4.38. Mesh 3D - dente interno

Si può osservare che in queste immagini è stato variato il numero di elementi lungo il profilo del dente ma anche lungo la larghezza di fascia. Segue la rappresentazione delle ruote, non si tratta di una vera e propria mesh ma è un feedback importante per quanto concerne la correttezza di realizzazione della geometria.

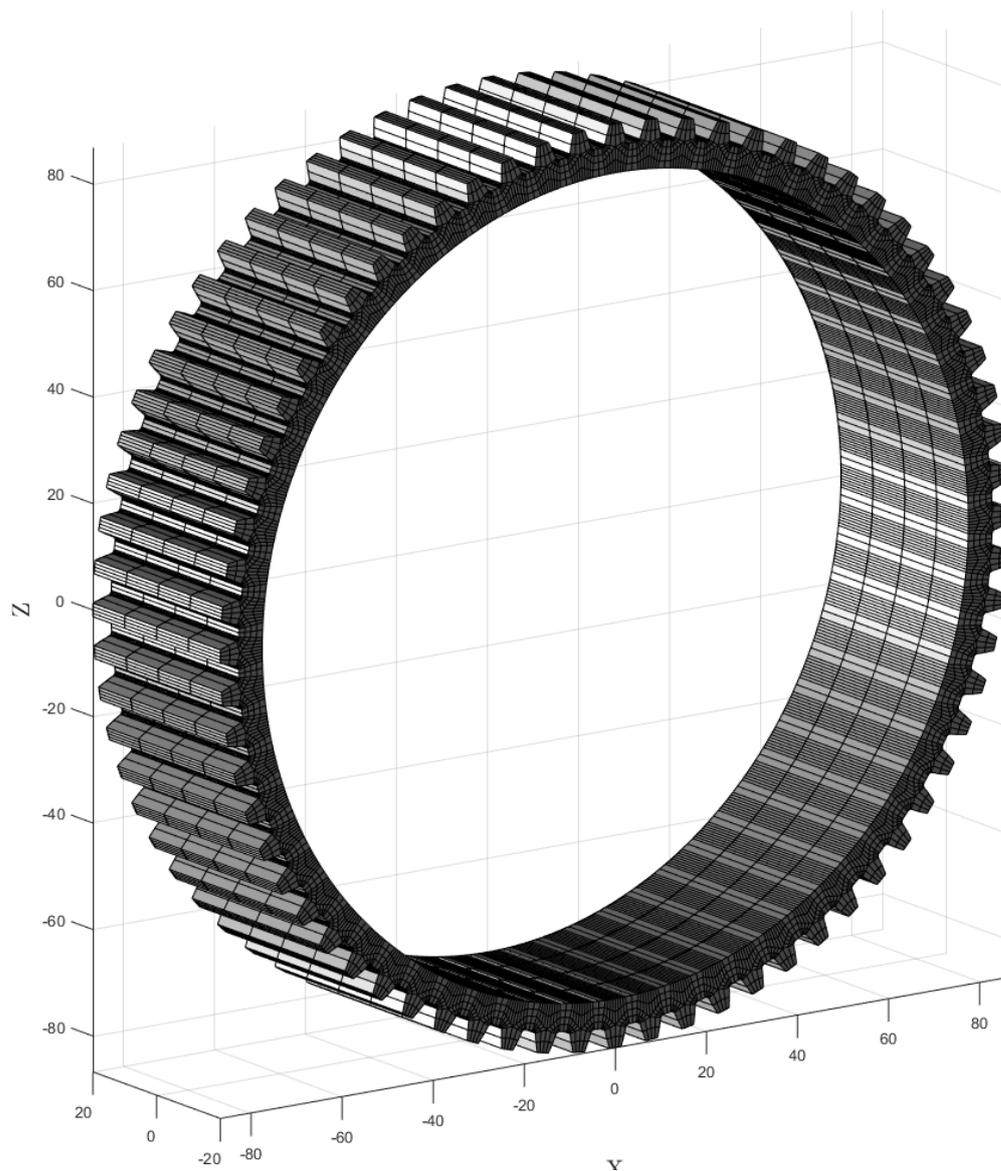


Figura 4.39. Mesh 3D - ruota a denti esterni

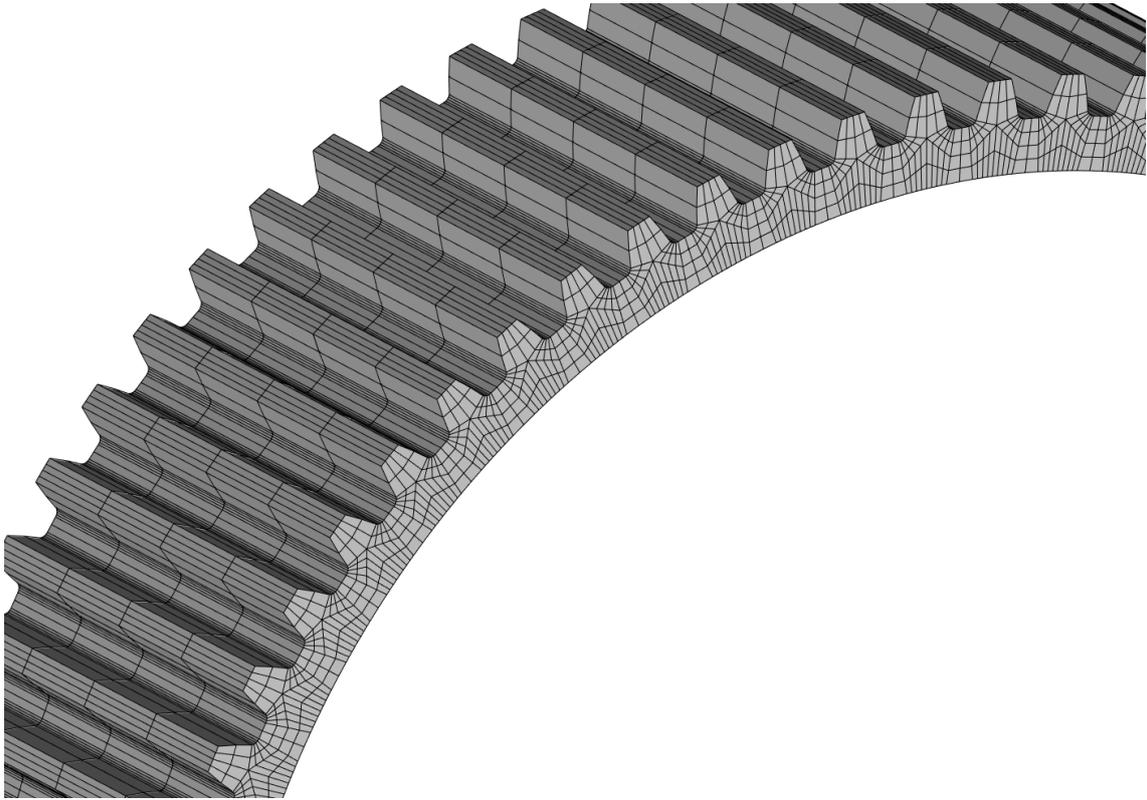


Figura 4.40. Mesh 3D - ruota a denti esterni, dettaglio

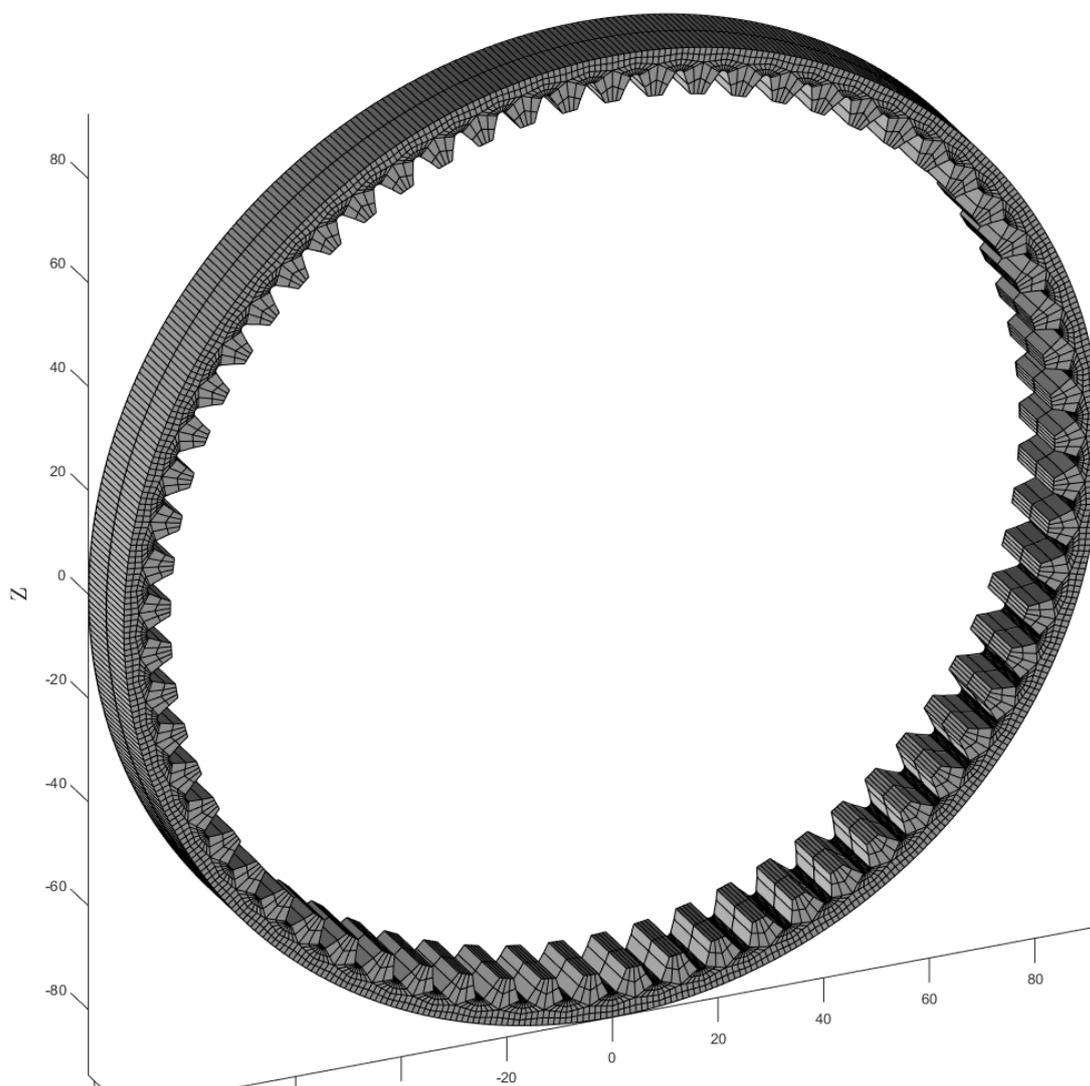


Figura 4.41. Mesh 3D - ruota a denti interni

Figura 4.42. Mesh 3D - ruota a denti interni

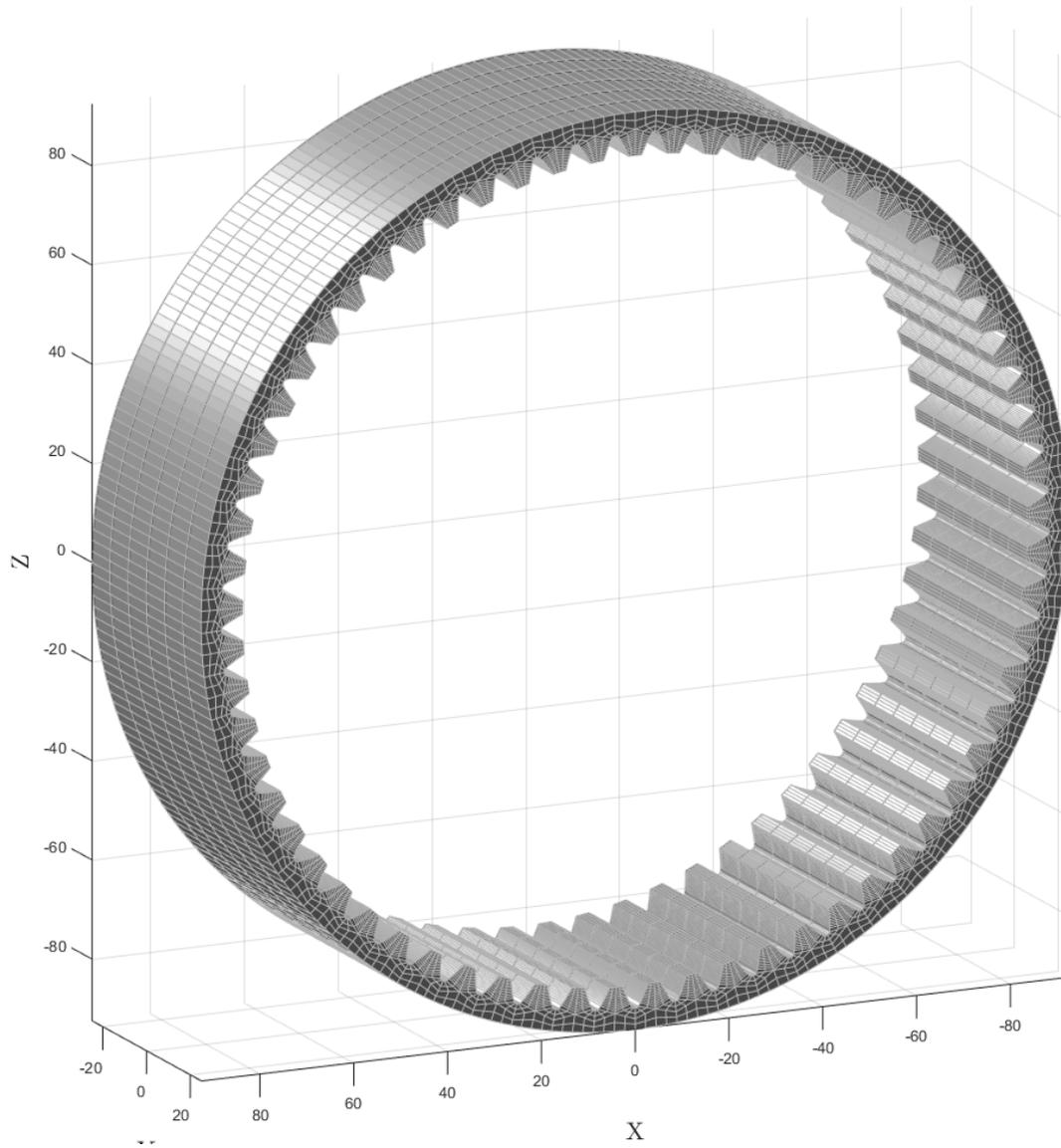


Figura 4.43. Mesh 3D - ruota a denti interni

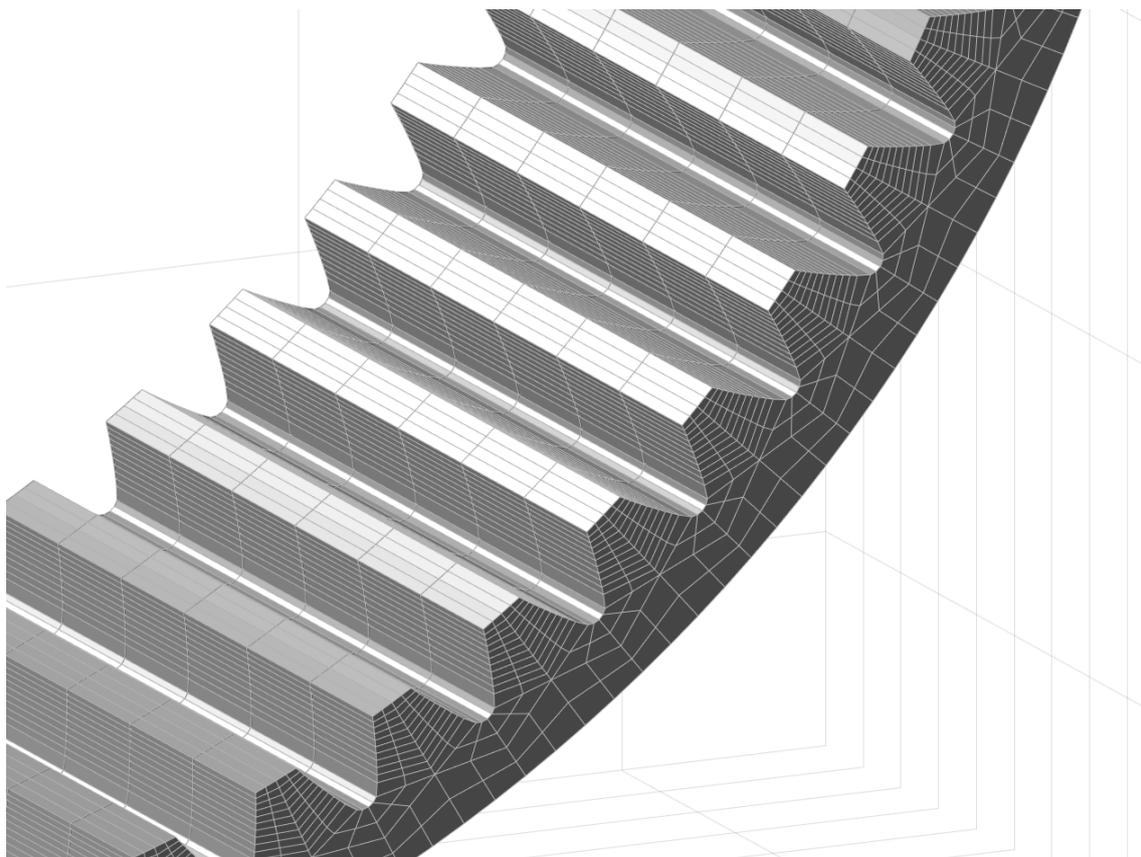


Figura 4.44. Mesh 3D - ruota a denti interni

4.9 Conclusioni

In questo capitolo è stato illustrato un metodo per realizzare le strutture che compongono la mesh di una ruota dentata a denti dritti. Dopo aver generato il profilo evolvente, il raccordo, si procede alla definizione del dominio 2D. Da questo tramite opportune funzioni si riesce a calcolare la matrice dei nodi per ogni zona della mesh e dei nodi sul bordo. Manipolando la matrice dei nodi si ottengono gli elementi 2D. Il passaggio alla mesh 3D si ottiene replicando la mesh della sezione 2D nello spazio, cambiando opportunamente gli indici dei nodi sulla sezione e usando logiche simili a quelle del calcolo degli elementi 2D sulla sezione piana per il calcolo degli elementi 2D laterali. Il risultato è un set di matrici di nodi, elementi 1D, elementi 2D, elementi 3D. La matrice dei nodi è caratterizzata da indice, coordinate nello spazio. Le matrici degli elementi sono caratterizzate da indice dell'elemento e nodi dell'elemento. I nodi sono ordinati secondo una convenzione. Questo metodo di generazione di mesh si rivela snello in termini di tempo e costi computazionali perchè sfrutta con criterio le caratteristiche dell'ambiente MATLAB. Il tempo di calcolo di una mesh di questo tipo è inferiore, normalmente, ai 5 secondi.

Capitolo 5

Ruote a denti elicoidali

5.1 Introduzione

In questo capitolo si presenta la realizzazione della mesh parametrica di ruote a denti elicoidali.

5.2 Realizzazione

L'algoritmo è sostanzialmente uguale a quello delle ruote a denti dritti. La rotazione della sezione dovuta all'elica è linearizzata. Si applica una rotazione alle sezioni intorno all'asse z . Seguono i risultati:

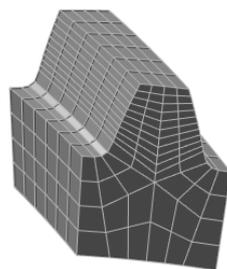


Figura 5.1. Mesh 3D - dente

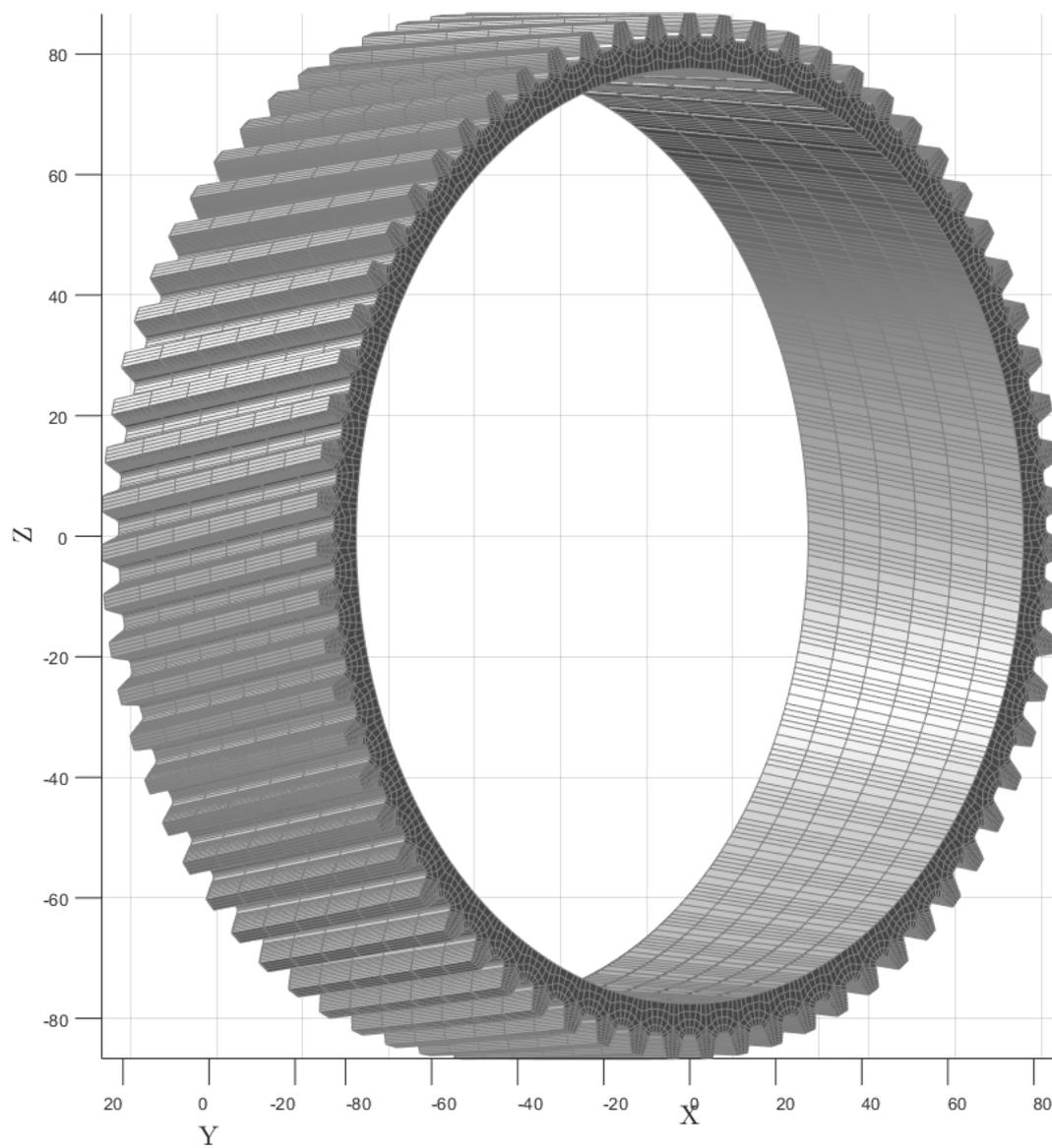


Figura 5.2. Mesh 3D - ruota a denti esterni elicoidale - set 1

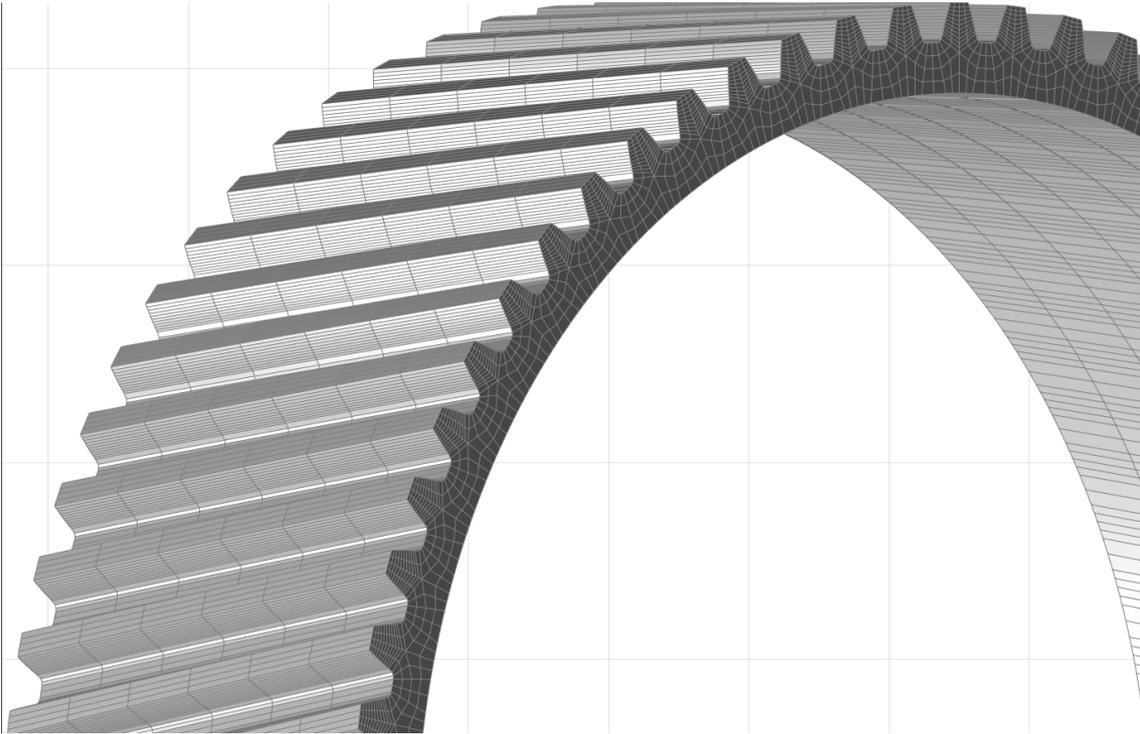


Figura 5.3. Mesh 3D - ruota a denti interni elicoidale - set 1

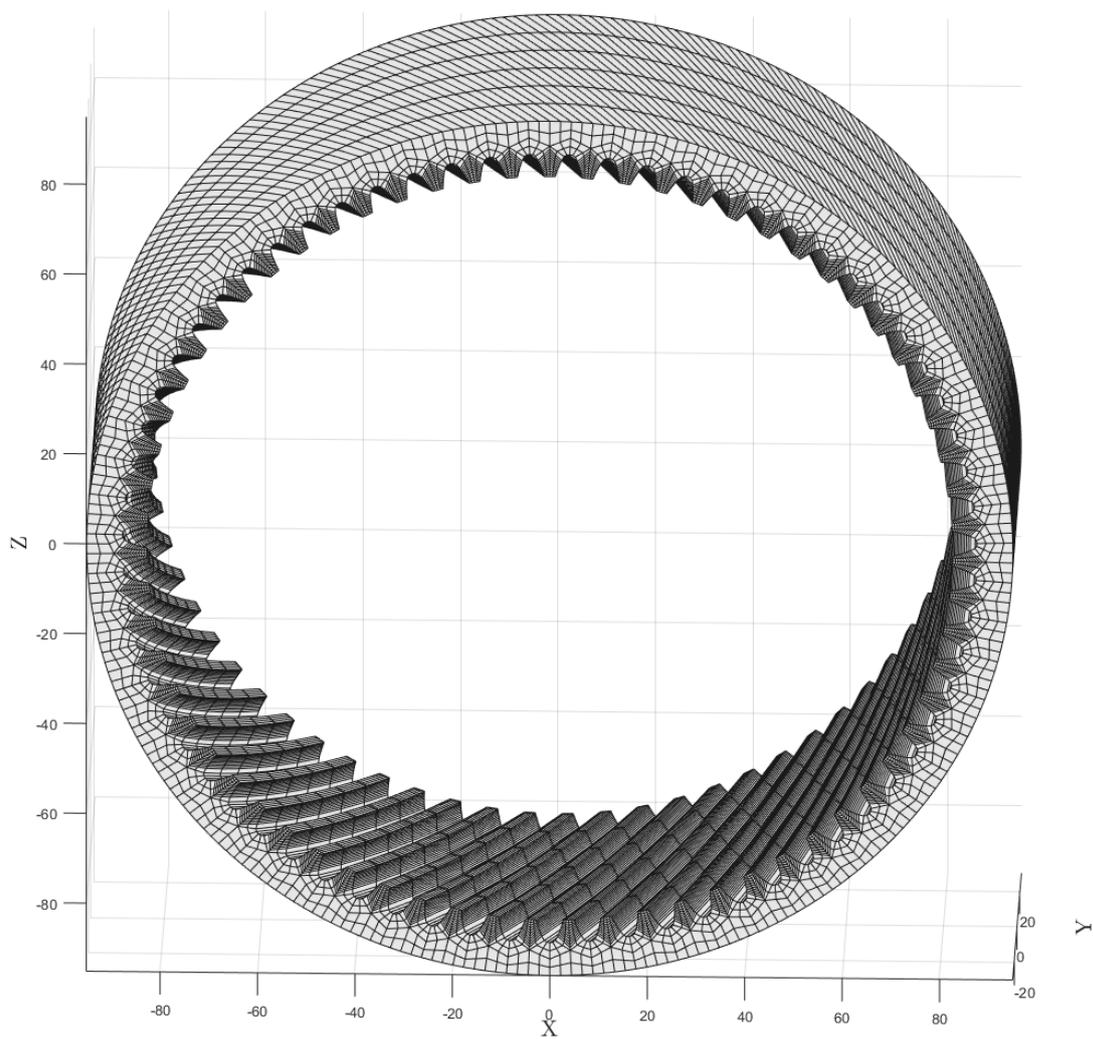


Figura 5.4. Mesh 3D - ruota a denti interni elicoidale

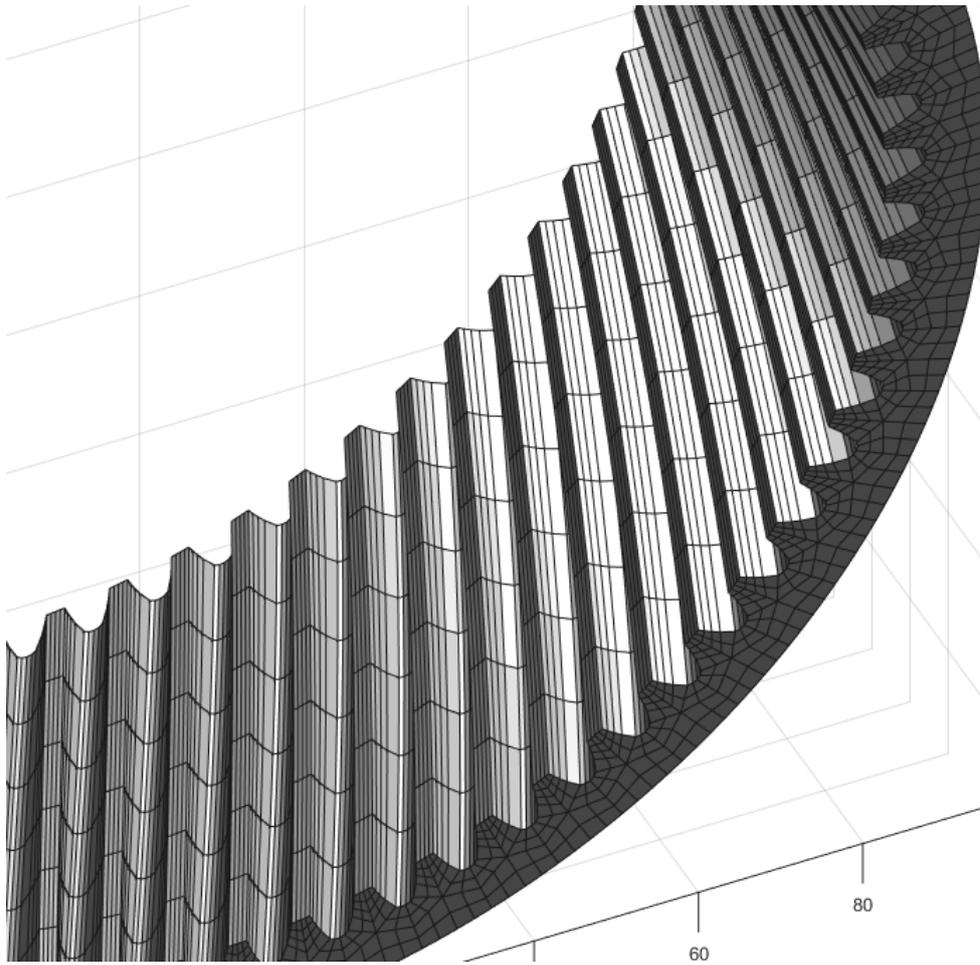


Figura 5.5. Mesh 3D - ruota a denti interni elicoidale, dettaglio

Parte III
Terza Parte

Capitolo 6

Ruote spiroconiche

6.1 Introduzione

In questa parte della tesi si descrive la realizzazione della parte del tool relativa alle ruote spiroconiche.

Il primo passo è la descrizione dei dati di partenza e dell'output del tool preesistente.

Successivamente si tratta il metodo con cui si è scelto di meshare la ruota e quindi si presentano le azioni necessarie per arrivare al risultato. Si descrivono i due algoritmi per trovare i punti di intersezione tra una curva e una superficie nello spazio e si confrontano i risultati.

Successivamente si presenta il principio dell'algoritmo di mesh 3D dei denti. Infine sono presentate le prestazioni dell'algoritmo e i risultati.

6.2 Il tool preesistente

Per la realizzazione della mesh delle ruote spiroconiche non è stato necessario inserire le equazioni di evolvente, raccordo per il calcolo del vano. Un tool preesistente è stato sfruttato per questo scopo. Esso oltre a calcolare la geometria dei vani, li posiziona correttamente nello spazio, ispirandosi al processo di realizzazione manifatturiero.

Segue che l'output di questo tool è una superficie definita per punti, che rappresenta uno dei vani della ruota spiroconica i cui parametri geometrici sono stati immessi come input.

Questo tool è stato modificato al fine di ottenere, come per le dentature elicoidali e a denti dritti, il numero di elementi imposto dall'utente sui tratti del profilo evolvente, del raccordo, del rim.

Altre modifiche sono state fatte nella fase di ottimizzazione della precisione dell'algoritmo di calcolo dei punti di intersezione.

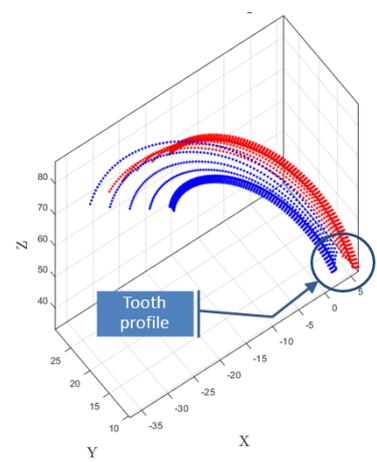


Figura 6.1. Output del tool modificato

Questa struttura viene interpretata come un fascio di spline. La spline collega, quindi, lo stesso punto topologico di sezioni diverse. Queste spline, sono le curve che vengono utilizzate in input nell'algoritmo di intersezione.

6.3 Algoritmi di intersezione

Gli obiettivi da raggiungere per la realizzazione di questa mesh, hanno imposto di non utilizzare i punti calcolati dal tool preesistente come sezioni della dentatura. Al fine di ottenere un tool più flessibile, si è optato per imporre automaticamente le sezioni della ruota, su cui calcolare i punti di intersezione tra il fascio di spline e le superfici sulle quali si vuole ottenere la mesh.

Per essere più chiari: le sezioni della dentatura non sono più piane ma giacciono su varie superfici coniche. I punti sono calcolati come punti di intersezione tra il fascio di spline e le superfici coniche.

Il motivo per cui si è optato per superfici coniche, risiede nella necessità di una dentatura ciclosimmetrica. Se replicata, non ci devono essere errori di visualizzazione o geometria.

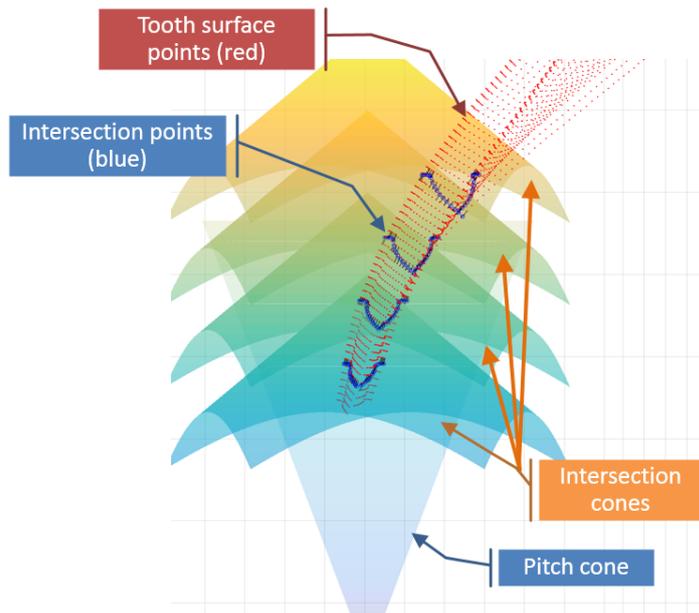


Figura 6.2. Punti di intersezione tra fascio di spline e superfici coniche

Sono stati sviluppati due algoritmi per il calcolo di questi punti di intersezione. Il primo si basa su una pubblicazione, il secondo è stato sviluppato dal tesista. Il primo algoritmo si basa sulle proiezioni ortogonali, il secondo su un metodo denominato "Curve points check".

L'algoritmo di intersezione è una parte importante del metodo di mesh di ruote spiroconiche. Sostanzialmente, quando su carta è stato verificato che l'algoritmo ottimizzato di calcolo della matrice dei nodi sulle varie zone del dominio era molto efficiente, si è capito che in termini di prestazioni dell'algoritmo, il collo di bottiglia sarebbe stato rappresentato dall'algoritmo di intersezione.

Per il problema dell'intersezione, non sono state individuate soluzioni alternative a quelle della creazione di un algoritmo. Trovare una soluzione in forma chiusa è un problema di non immediata soluzione. Una superficie conica si può definire tramite due parametri, una curva nota per punti può essere approssimata da una curva di Bezier nel caso più semplice. L'approssimazione con curve di Bezier è stata realizzata, ma il problema è che queste curve sono definite da equazioni che dipendono da un parametro diverso da quelli usati per la definizione della superficie conica. In teoria, è possibile relazionare i parametri, tuttavia, non è detto che anche riuscendoci, la soluzione sia robusta per tutti i tipi di ruote spiroconiche che il tool si propone di realizzare. Sotto questo punto di vista, gli algoritmi di intersezione sviluppati, invece, sono valutati come più robusti.

Segue la presentazione del principio del metodo sfruttato per trovare la proiezione di un punto nello spazio su una curva.

6.3.1 Metodo con proiezioni ortogonali

Il primo algoritmo sfrutta le proiezioni ortogonali. La scelta di questo algoritmo si concilia bene con le proprietà della curva e della superficie da intersecare. Una di queste risiede nel fatto che la curva attraversa sempre la superficie conica, non esistono situazioni in cui può essere tangente, che

sono valutate come critiche per questo tipo di algoritmo.

Questo algoritmo è stato trovato in letteratura ma solo in forma descrittiva. L'implementazione è stata realizzata dal tesista.

subsubsectionLogica dell'algoritmo: proiezione ortogonale

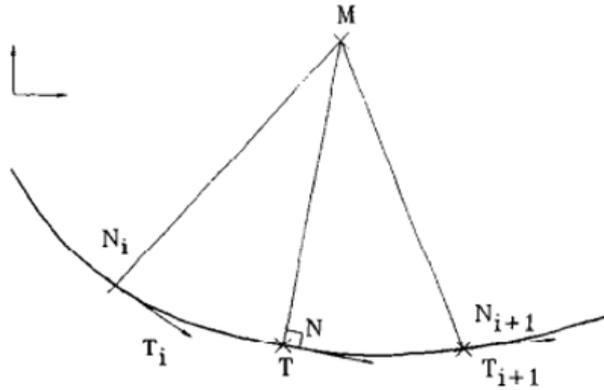


Figura 6.3. Schema di base

Come si può vedere in figura, il prodotto scalare tra il segmento che collega il punto da proiettare e il segmento tangente alla curva, è nullo in corrispondenza del punto di proiezione ortogonale. Prima e dopo il punto ortogonale, si evidenzia che i segni del prodotto scalare sono diversi. Il principio del cambio di segno permette di capire il tratto di curva in cui è presente il punto di proiezione. Non è possibile trovare la soluzione in forma chiusa, però è possibile trovare il punto con la precisione desiderata, reiterando l'algoritmo su tratti di curva sempre più piccoli.

Logica dell'algoritmo di intersezione

Nell'algoritmo di intersezione, il metodo delle proiezioni viene sfruttato per proiettare un punto della curva sulla superficie conica.

Più precisamente, il punto viene proiettato su una curva appartenente alla superficie conica. Sostanzialmente, si impone uno dei due parametri che descrivono la superficie conica come costante. Sono state utilizzate due iterazioni per trovare la proiezione di un punto sulla superficie, questo perché si è visto che erano sufficienti per portare a convergenza l'algoritmo. Il punto proiettato sulla superficie viene successivamente proiettato ortogonalmente sulla curva. Si vede che questo punto, è più vicino al punto di intersezione.

Questo meccanismo è ripetuto monitorando la riduzione relativa della quota z ad ogni iterazione. Nei casi presi in esame, si sono rivelate necessarie meno di venti iterazioni.

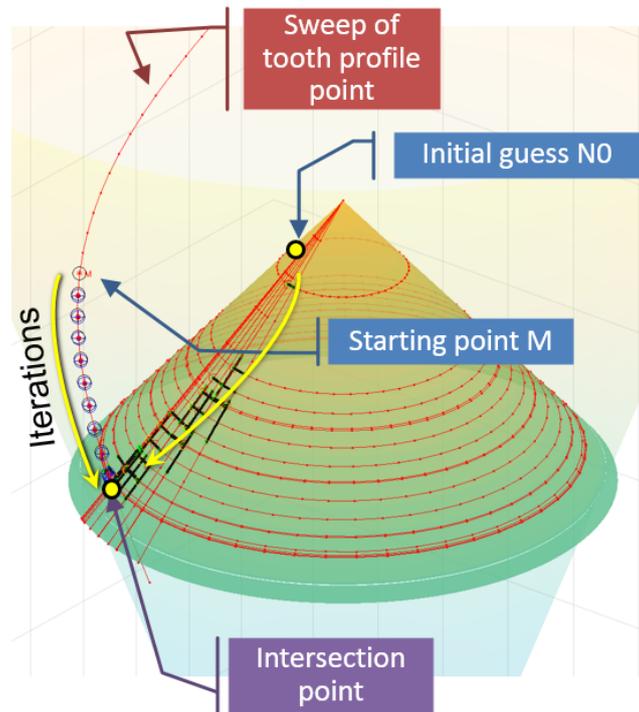


Figura 6.4. Schema qualitativo delle iterazioni

Limiti dell'algoritmo

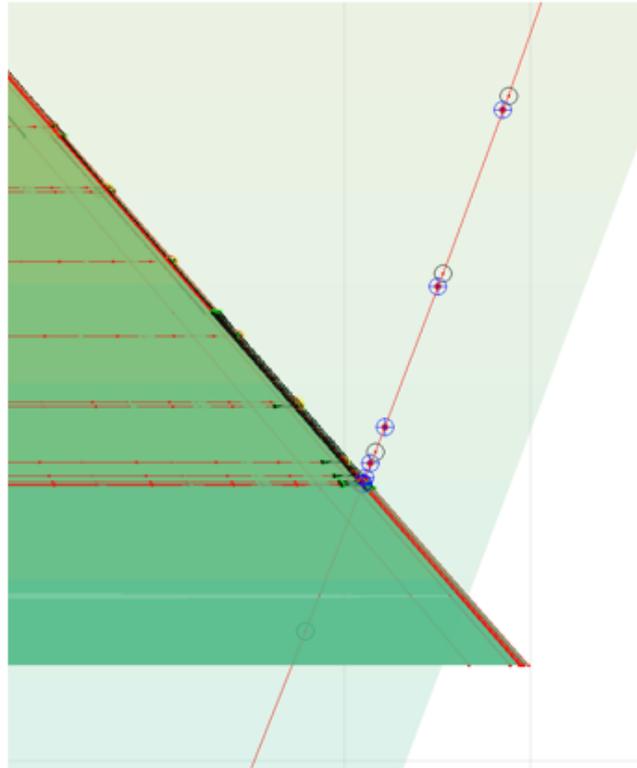


Figura 6.5. Precisione algoritmo

Questo algoritmo non si può vettorizzare, di conseguenza le performance in ambiente MATLAB, nonostante degli incoraggianti miglioramenti non sono state giudicate accettabili. In un problema geometrico come quello che ci si propone di risolvere, i punti da calcolare non sono pochi. L'ordine, è delle decine. L'algoritmo arriva a convergenza dopo 0.6 secondi per ogni punto. Per trovare 10 punti, il costo in termini di tempo è dell'ordine di 5-6 secondi.

Per questo motivo, è stato implementato un altro algoritmo, quello denominato "Curve point check".

6.4 Metodo "Curve points check"

6.4.1 Introduzione

Questo metodo, di semplice comprensione, è stato sviluppato dopo aver analizzato i motivi per cui il precedente falliva.

Si è deciso di creare un algoritmo che fosse vettorizzabile. L'algoritmo cerca i punti della curva che si trovano sopra e sotto la superficie conica.

6.4.2 Principio

La superficie conica, è la frontiera di un cono. Il punto di intersezione tra la spline di interesse ed il cono, è lo stesso di quello calcolato come intersezione della curva con la superficie conica. L'equazione cartesiana del cono è nota. Sono noti anche i punti della curva che si vuole intersecare con esso. L'idea è che introducendo le coordinate dei punti della curva nell'equazione del cono implicita, il segno del risultato cambia a seconda che i punti si trovino dentro o fuori dal cono. Il tratto di curva in cui due punti successivi, quando sostituiti nell'equazione, danno risultati con segni diversi, è quello in cui si trova il punto di intersezione.

L'algoritmo dopo il primo step, approssima la curva con una retta, o in alternativa con una curva di Bezier, localmente.

In questo modo si può procedere con altre iterazioni. Dopo un certo numero di test, si è trovato che due iterazioni sono sufficienti ad ottenere una precisione di $1e-5$.

I parametri di questo algoritmo sono comunque modificabili dall'utente, nel caso fosse necessaria una maggiore o minore precisione.

Performance dell'algoritmo

Questo algoritmo, una volta vettorizzato, permette di trovare un punto di intersezione in circa 0.0001 secondi, quindi il match tra performance richieste e realizzate è molto buono.

6.5 Algoritmo di mesh 3D

6.5.1 Ottimizzazione

L'algoritmo per il calcolo della matrice dei nodi di ogni zona è stato ottimizzato.

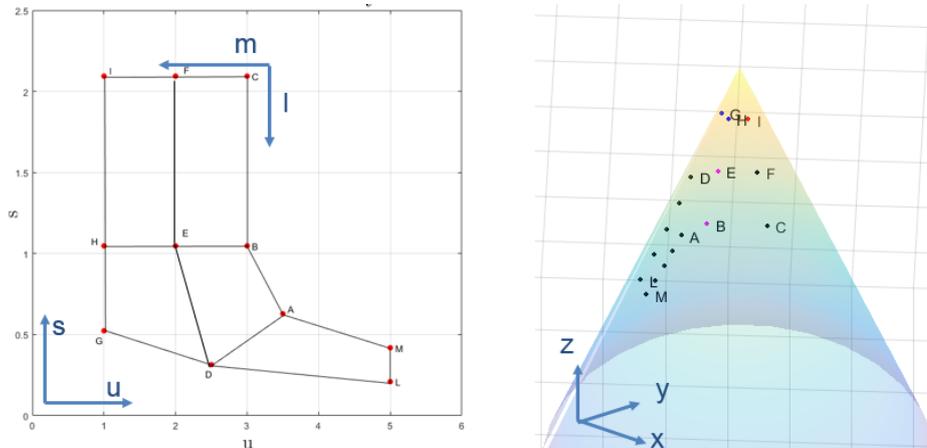


Figura 6.6. Sistemi di riferimento coinvolti

Nella figura sono presentati i main nodes di metà dente, prima nel sistema di riferimento dei parametri della superficie conica (u,s) e poi nel sistema di riferimento cartesiano. Una caratteristica importante è anche il sistema di riferimento topologico globale (l,m). Ogni zona ha poi un proprio sistema topologico locale.

La logica che è stata seguita per generare la mesh è stata la seguente:

1. Generazione della mesh per la prima metà di tutte le sezioni
2. Generazione della mesh per l'altra metà di tutte le sezioni
3. Fusione dei due tipi di mesh precedenti

La matrice dei nodi viene calcolata con un nuovo algoritmo, che permette di calcolare nella stessa funzione anche gli elementi 1D, 2D.

Le informazioni delle due metà sono integrate usando una funzione che è simile a quella che era stata utilizzata per le ruote a denti dritti.

Principio dell'algoritmo

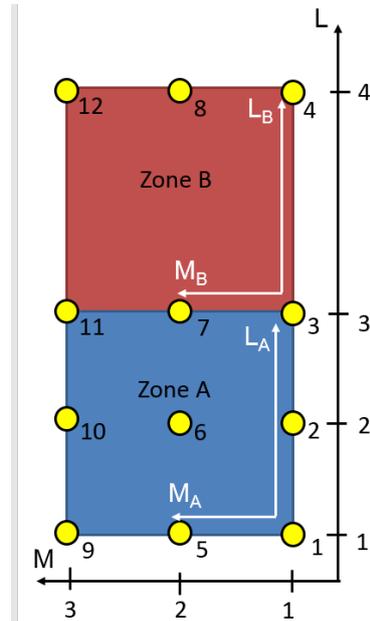


Figura 6.7. Sistemi di riferimento coinvolti

In figura sono rappresentate due zone del dominio. Sono presenti i sistemi di riferimento topologici locali e quello globale. Tramite relazioni matematiche è possibile ricondursi agli indici globali dei nodi. Nella stessa matrice sono introdotti i vettori dei punti in coordinate (u,s) . Questi valori, tramite le equazioni del cono di intersezione, sono trasformati nelle coordinate x,y,z del sistema di riferimento cartesiano. Tramite questa tecnica, è possibile relazionare gli indici globali con le coordinate x,y,z in maniera univoca.

A livello computazionale, il calcolatore si trova a dover compilare una matrice, inserendoci degli array noti ed estraendo le colonne di interesse, cioè: ID globale, coordinate del sistema di riferimento cartesiano.

6.5.2 Prestazioni

La matrice dei nodi viene calcolata prima per ogni zona, poi tutte le matrici sono integrate in quella globale per quella sezione.

Si deduce che le dimensioni delle matrici sono piccole, quindi molto agevoli da gestire per MATLAB. I tempi di computazione, non avendo più cicli for sono molto ridotti. Questo algoritmo è stato dedotto osservando i risultati della versione non ottimizzata, estraendo dei pattern dalla matrice globale dei nodi.

6.6 Risultati

Per quanto concerne i risultati, sono presentati in questa sezione prima i denti al variare dei parametri in input e successivamente due test case di ruote spiroconiche nella loro interezza.

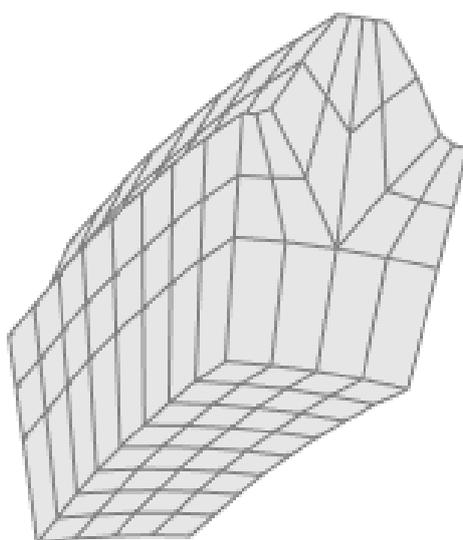


Figura 6.8. Dente - set parametri 1

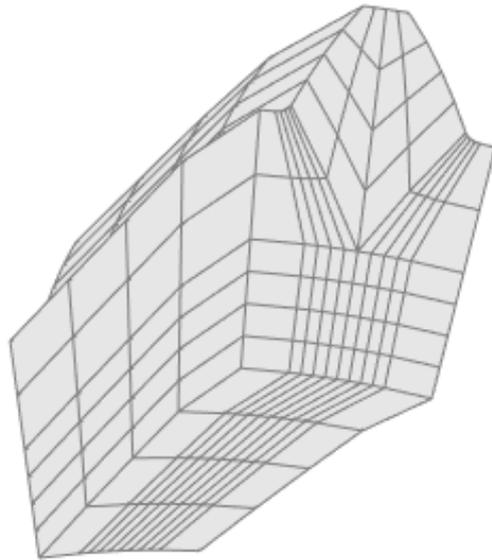


Figura 6.9. Dente - set parametri 2

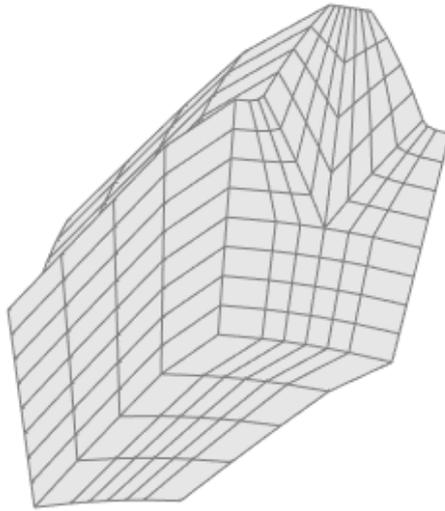


Figura 6.10. Dente - set parametri 3

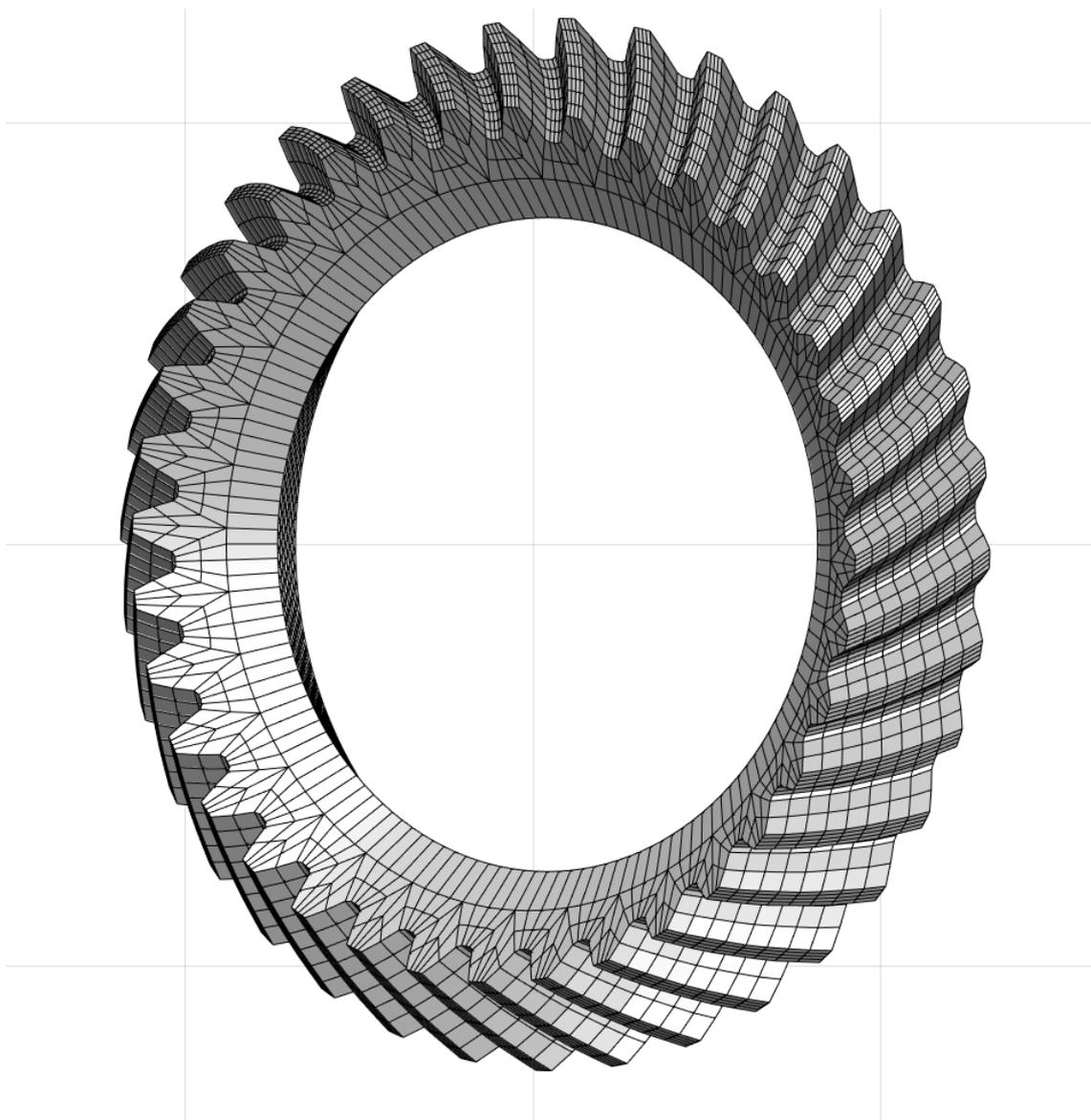


Figura 6.11. Ruota spiroconica - test case 1

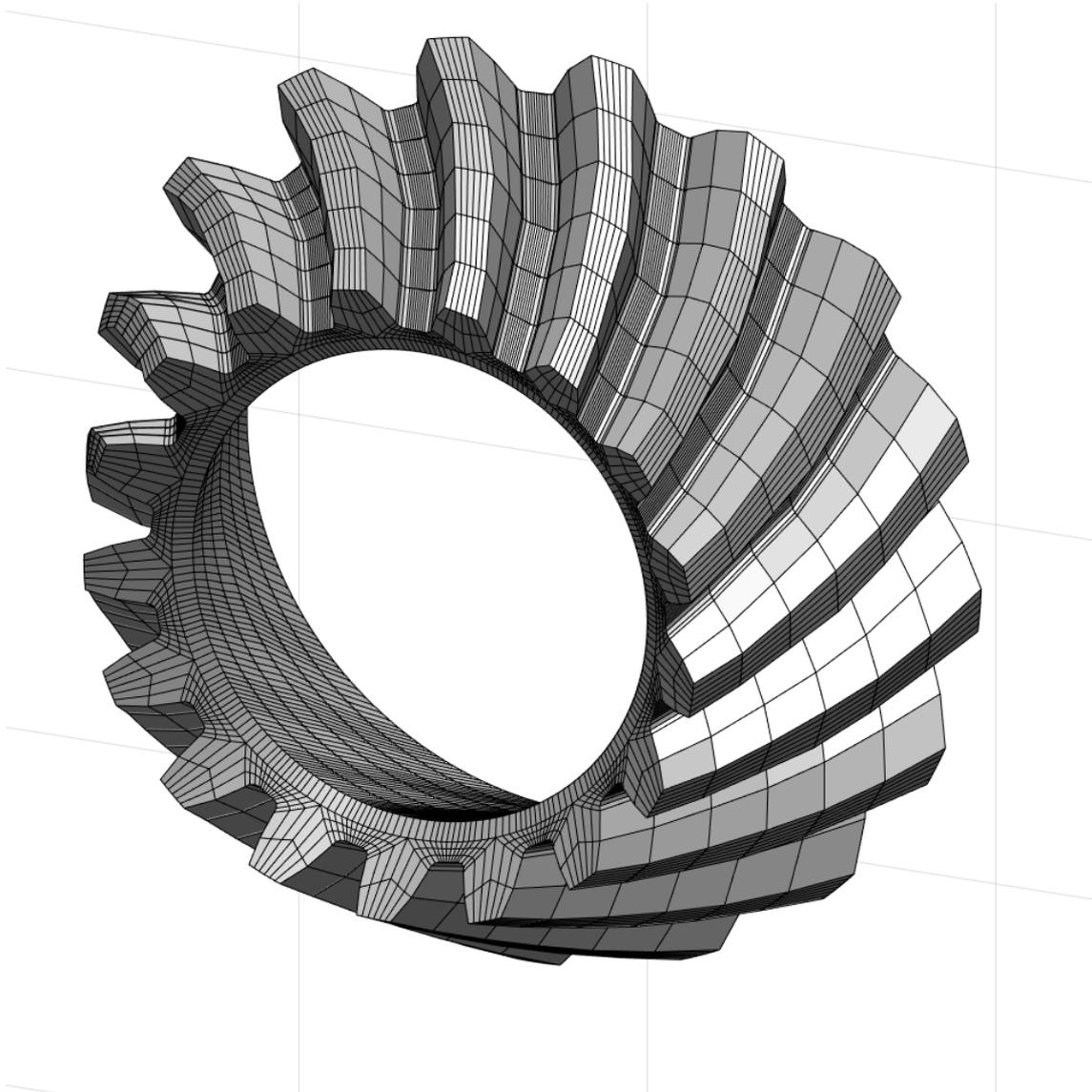


Figura 6.12. Ruota spiroconica - test case 2

Capitolo 7

Esportazione dei risultati

L'output del tool, come detto precedentemente, è un insieme di matrici che forniscono informazioni sui nodi, gli elementi, i gruppi del dente meshato.

Siccome l'obiettivo di questo tool è sostituire l'operazione di mesh in Hypermesh di una dentatura, ma non il risultato, si deve ottenere un file analogo a quello generato da Hypermesh.

Per questo scopo, uno script MATLAB è stato sviluppato al fine di convertire le informazioni delle matrici nel formato corretto.

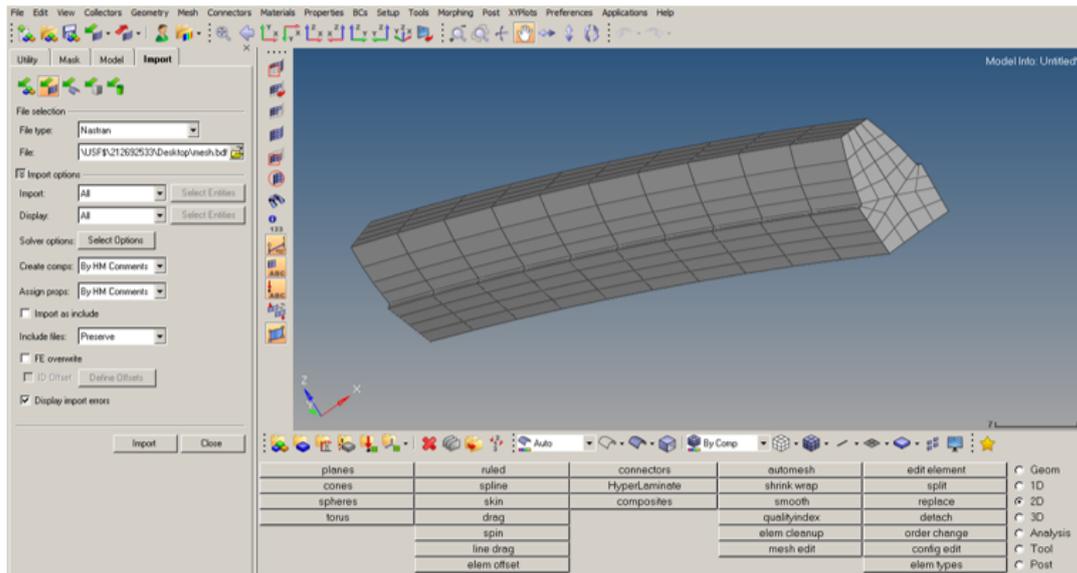


Figura 7.1. Esempio di mesh generata dal tool aperta in Hypermesh

In questo modo si ottiene lo stesso risultato della mesh manuale e non ci sono ulteriori variazioni nelle operazioni successive per realizzare le analisi agli elementi finiti.

Capitolo 8

Conclusioni

L'esperienza di tesi è stata molto formativa, gli obiettivi preposti sono stati raggiunti e il tool sembra sufficientemente robusto per tutte le categorie di ruote. La realizzazione di un codice così complesso ha richiesto un innalzamento della qualità della programmazione sia in termini di organizzazione in script e funzioni, ma anche a livello di efficienza per minimizzare i tempi di esecuzione. La possibilità di accedere a codici inerenti la geometria delle ruote dentate è stata importante, soprattutto in termini di acquisizione di know-how. Alcuni di questi aspetti hanno avuto una influenza importante nella fase di progettazione su carta degli algoritmi.

Bibliografia

- [1] Stephen P.Radzevich, *Dudley's - Handbook of practical gear design and manufacture* , Second Edition
- [2] Bern, Marshall; Plassmann, (2000), *Mesh Generation, Handbook of Computational Geometry*, Elsevier Science
- [3] F. Thompson, B.K.Soni, N.P. Weatherill, *Handbook of Grid Generation* ,CRC-Press, 1998, ISBN 978-0-8493-2687-5
- [4] Adeeb, S. (2018), *Introduction to Solid Mechanics and Finite Element Analysis*, Retrieved from <https://sameradeeb-new.srv.ualberta.ca>
- [5] Anis Limaiem, François Trochu, (1995), *Geometric algorithms for the intersection of curves and surfaces* , Elsevier Science
- [6] Litvin, Faydor L., Alfonso Fuentes, *Gear geometry and applied theory*, Cambridge University Press, 2004.
- [7] Claudio Canuto, *Appunti per il Corso di Modelli e Metodi Numerici*
- [8] Hooman Abdoli, *A finite element approach to spur, straight bevel and hypoid gear design*, University of Oslo
- [9] Tom M.Apostol, *Unwrapping Curves from Cylinders and Cones*, The Mathematical Association of America
- [10] Vick, *An Introduction to Algebraic Topology*, 2^a ed., New York, Springer, 1994
- [11] Marco Manetti, *Topologia*, Springer, 2008
- [12] Andre Bakker, *Meshing,Lecture 7*, Retrieved 2012-11-10
- [13] Immagini:
Copyright © 2015XYZ *Scientific Applications*, Inc., 2255 Morello Ave. Suite 220, Pleasant Hill CA. 94523, U.S.A. All rights reserved.
- [14] © SAS IP, Inc. All rights reserved.
- [15] CC BY-SA 3.0
- [16] Creative Commons licence