



Università Degli Studi di Pavia

FACOLTA' DI INGEGNERIA DIPARTIMENTO DI INGEGNERIA
DELL'INFORMAZIONE

**CORSO DI LAUREA MAGISTRALE IN
BIOINGENERIA**

TITOLO

**Ricostruzione Tridimensionale di Distretti
Vascolari Patient-Specific a Partire da
Centerlines: dai Dati Geometrici al File CAD**

Candidato: Virna Puchooa

Relatore: Prof. Stefania Marconi

Correlatore: Dott. Alberto Cattenone

Anno Accademico 2023-2024

Abstract

La presente tesi sviluppa una metodologia integrata per la modellazione 3D accurata di strutture vascolari, cruciale in ingegneria biomedica per la simulazione del flusso sanguigno, la pianificazione chirurgica e la progettazione di dispositivi medici.

L'obiettivo è di superare le limitazioni dei formati di file tradizionali come STL (Stereolithography), in cui le superfici 3D vengono rappresentate tramite una tessellazione, ovvero una rete di triangoli. Questo metodo introduce approssimazioni geometriche che possono compromettere l'accuratezza delle simulazioni. Inoltre, la mancanza di informazioni fisiche e materiali nel formato STL rappresenta un'ulteriore sfida per la creazione di modelli affidabili per la simulazione.

Per superare queste limitazioni, questa tesi sviluppa una metodologia articolata in diverse fasi. Inizialmente, viene presentato il software Autodesk Inventor, ambiente CAD 3D leader, e vengono descritte le principali tecniche di modellazione utilizzate per la creazione di geometrie complesse. Successivamente, viene introdotto il formato STEP, evidenziandone le potenzialità e le difficoltà legate alla sua gestione diretta, soprattutto per quanto riguarda le spline. Si passa quindi all'introduzione di VBA (Visual Basic for Applications), linguaggio di programmazione integrato in Inventor, che consente di automatizzare compiti e personalizzare il software.

Il cuore della tesi è rappresentato dall'integrazione tra Matlab e Inventor. I dati geometrici, ottenuti da un file di testo pre-esistente (derivato da un software come Materialise), vengono elaborati in Matlab per estrarre le informazioni rilevanti. I risultati vengono poi trasferiti in ambiente Inventor tramite codice VBA, dove vengono generati workpoints e superfici per la creazione del modello 3D finale.

La metodologia sviluppata produce modelli 3D vascolari accurati, superando le limitazioni dell' STL e abilitando nuove analisi. I risultati ottenuti vengono validati attraverso specifici metodi di confronto e analisi. Infine, vengono discusse le conclusioni della ricerca, evidenziando i limiti e le potenzialità della metodologia proposta, nonché i possibili sviluppi futuri.

Indice

Abstract	1
1 Introduzione	6
1.1 Stato dell'Arte: Tecniche avanzate per la creazione di geometrie per la medicina personalizzata	8
1.2 Obiettivo della tesi	12
2 Autodesk Inventor	13
2.1 Presentazione di Inventor	13
2.2 Interfaccia di Inventor	14
2.3 Funzionalità di modellazione	15
2.4 Funzionalità di Inventor utilizzate nel progetto	16
2.5 Il formato .ipt : elementi costitutivi delle parti in Inventor . .	18
2.5.1 .ipt vs .stl: due formati a confronto	18
3 Approccio mediante modifica diretta di file STEP	19
3.1 Introduzione	19
3.2 Utilizzo di file STEP	19
3.2.1 Primo approccio con file STEP	19
3.2.2 Composizione di uno STEP file	20
3.2.3 Funzione SPLINE	28
3.2.4 Limitazioni nell'editing della spline tramite il formato STEP	34
4 Automazione mediante VBA	36
4.1 Automazione del processo di disegno con VBA Autodesk Inventor supportata da Matlab	41
4.1.1 Flusso di lavoro di Materialise	41
4.1.2 Estrazione e formattazione dei dati con Matlab	48
4.2 Implementazione del codice VBA	55

4.2.1	Routine principale Congiunzione_Biforcazioni	55
4.2.2	Subroutine <i>LeggiDatiExcel</i>	57
4.2.3	Subroutine CreaSolidoPerMarker	60
5	Risultati	65
5.0.1	Creazione dei punti di riferimento:	65
5.0.2	Creazione delle linee e Sketch circonferenze:	66
5.0.3	Creazione del loft:	67
5.0.4	Modello 3D completo:	68
5.0.5	Validazione del modello attraverso confronto di geometria	69
6	Discussione	71
6.1	Riepilogo dei risultati principali	71
6.2	Valutazione dell'efficacia del metodo	73
6.3	Analisi delle scelte progettuali	74
6.4	Discussione delle limitazioni	75
7	Conclusione	78
8	Appendice: Codici completi Matlab e VBA	80
8.1	Codice Matlab:	80
8.2	Codice VBA:	86

Elenco delle figure

1.1	Passaggi per l'ottenimento di geometrie Patient Specific	9
1.2	Fasi della tesi	12
2.1	Informazioni presenti nell'interfaccia di Inventor	14
2.2	Elementi di disegno per la creazione dello schizzo 2D	16
2.3	Funzionalità per lo sviluppo del modello 3D	17
3.1	creare un nuovo file di parte	21
3.2	Visualizzazione dei diversi piani su cui iniziare il disegno	21
3.3	Schizzo 2D iniziale	22
3.4	Modello 3D finale, con sweep	23
3.5	come esportare il file .ipt in file STEP	24
3.6	Esempio STEP file	25
3.7	Rappresentazione del disegno 2D ottenuto realizzando due sketch: il primo la spline di interpolazione, con 5 punti definiti e il secondo la circonferenza di raggio 2.5.	31
3.8	Modello 3D finale, ottenuto mediante l'estrusione <i>sweep</i> del profilo circolare lungo la traiettoria definita dalla curva	31
3.9	Passaggi di costruzione modello 3D.	31
4.1	Icona della Macro, da qui è possibile accedere all'interfaccia di scrittura di VBA da Inventor.	37
4.2	Finestra di dialogo <i>Macro</i> , in Inventor, dove si possono creare, modificare o eliminare i moduli.	38
4.3	Come arrivare ad aggiungere la libreria degli oggetti di Excel.	39
4.4	Finestra di dialogo <i>Riferimenti- ProgettoApplicazione</i> dove bi- sogna spuntare la libreria relativa ad Excel.	40
4.5	Main Branch Set, Branch Segment 1, vengono definiti tutti i parametri del primo segmento della struttura centrale.	45
4.6	Branch Set 1 sono tutti i parametri relativi alla prima ra- mificazione. Branch Segment 9 invece indica la numerazione continua di tutti i segmenti.	45

4.7	Diagramma di flusso della logica di elaborazione delle branch e assegnazione dei marker	52
4.8	File generati all'interno della cartella DataPoints	54
4.9	Organizzazione del file Excel generato dal codice Matlab, Colonne A-C coordinate x y z, colonna D i diametri Dfit, colonna E Marker del branch	54
4.10	Diagramma di flusso della subroutine principale VBA, che illustra il ciclo di lettura dei file Excel basato sui marcatori.	56
4.11	Diagramma di flusso della subroutine VBA per l'importazione di coordinate, diametri e marcatori da file Excel	57
4.12	Rappresentazione grafica del flusso di lavoro del codice VBA, che automatizza la generazione del modello 3D a partire dai dati di input.	61
5.1	La subroutine CreaSolidoPerMarker crea punti di riferimento per marker, utilizza le coordinate lette dal file Excel branch_1	65
5.2	Le linee vengono create per definire l'ase del lume della ramificazione e contemporaneamente vengono generati i piani di lavoro normali alla linea per disegnare la circonferenza ad ogni punto.	66
5.3	La funzione LoftFeatures.Add viene utilizzata per creare il solido 3D della ramificazione, collegando i profili 2D. In oltre possiamo vedere che il ciclo è andato avanti nella lettura del file branch_2 con la generazione dei punti di lavoro e lo sketch di linee e circonferenze.	67
5.4	Loft completi di diverse ramificazioni.	67
5.5	Il codice VBA genera il modello 3D completo del vaso sanguigno, combinando i risultati delle subroutine.	68
5.6	Nel Browser Modello sono visibili i diversi solidi generati dal codice VBA	68
5.7	Confronto tra i due modelli allineati.	69
6.1	Dimensioni dei diversi file, in ordine: STL, CAD, STEP.	71
6.2	Mancanza dei dati Dfit, nel file txt di Materialise	75
6.3	Sovrapposizione dei profili 2D (circonferenze viola) che impedisce la creazione del loft	76
6.4	La mancanza di una definizione univoca dei punti: Branch 2 definito con andamento dall'interno verso esterno.	76
6.5	La mancanza di una definizione univoca dei punti: Branch 4 definito con andamento dall'esterno verso l'interno.	77

Capitolo 1

Introduzione

Nel settore biomedicale, la creazione di modelli 3D personalizzati, noti come geometrie patient-specific, è diventata cruciale per la simulazione e la pianificazione di interventi chirurgici [1, 2]. Questi modelli, basati su dati individuali del paziente ottenuti tramite tecniche di imaging come la tomografia computerizzata (TC) o la risonanza magnetica (RM), permettono di simulare il comportamento di tessuti e organi, ottimizzando la progettazione di dispositivi medici e la pianificazione di interventi chirurgici.

La possibilità di visualizzare e analizzare l'anatomia del paziente in 3D consente ai medici di prendere decisioni più informate e di personalizzare i trattamenti, migliorando l'efficacia delle terapie e riducendo i rischi per il paziente [3].

Il processo di creazione di geometrie patient-specific inizia con l'acquisizione di dati di imaging medicali, che rappresentano l'anatomia del paziente come una serie di sezioni bidimensionali. Questi dati vengono quindi elaborati e segmentati per identificare le strutture anatomiche di interesse. Successivamente, le superfici evidenziate vengono ricostruite e convertite in modelli 3D.

La tecnica più comunemente utilizzata per la creazione di mesh di volume a partire da questi modelli 3D è la conversione in formato STL (Stereolithography). L'STL è diventato uno standard per la sua semplicità e compatibilità con una vasta gamma di software di modellazione 3D e di stampa 3D.

Tuttavia, nonostante la sua diffusione, l'STL presenta limitazioni significative che ne ostacolano l'utilizzo in contesti che richiedono alta precisione e modificabilità. La perdita di informazioni parametriche durante la conversione dei dati in formato STL rende difficile la modifica e l'ottimizzazione dei modelli 3D.

Inoltre, la rappresentazione della geometria attraverso una mesh grid, ossia tramite la suddivisione della superficie in unità infinitesime triangolari con

un lato in comune può introdurre artefatti e imprecisioni, compromettendo l'accuratezza delle simulazioni.

La mancanza di informazioni sulla disposizione e sull'organizzazione interna dei tessuti rende difficile l'utilizzo di tecniche di simulazione avanzate, come l'analisi agli elementi finiti (FEA).

Pertanto, è necessario esplorare approcci alternativi per la gestione di geometrie patient-specific, che permettano di superare le limitazioni dell' STL e di generare modelli 3D ottimizzati per la simulazione. Questi approcci dovrebbero consentire di conservare le informazioni parametriche, di generare mesh di volume di alta qualità e di facilitare l'applicazione di tecniche di simulazione avanzate.

1.1 Stato dell'Arte: Tecniche avanzate per la creazione di geometrie per la medicina personalizzata

La creazione di geometrie patient-specific, ovvero modelli 3D personalizzati basati su dati individuali del paziente, può essere una grande risorsa in ambito biomedicale per la simulazione e la pianificazione di interventi chirurgici. Le tecniche all'avanguardia in questo campo si basano su una combinazione di segmentazione, modellazione 3D e meshing, con un ruolo centrale del CAD.

Il processo inizia con la segmentazione di immagini mediche (TC, RM), fase cruciale per isolare le strutture anatomiche di interesse. Software specializzati come **Mimics Innovation Suite** [4], **3D Slicer** [5] e **ITK-SNAP** [6] offrono strumenti avanzati per questa fase, combinando algoritmi automatici con strumenti di editing manuale per garantire accuratezza e affidabilità.

A partire dai file STL, ottenuti da segmentazioni di immagini mediche, il **CAD (Computer-Aided Design)** gioca un ruolo centrale nella ricostruzione delle superfici delle strutture anatomiche. Questa tecnologia, che consente di creare, modificare, analizzare e documentare modelli 2D e 3D di oggetti, è particolarmente importante in un contesto di modellazione patient-specific.

La modellazione parametrica, una funzionalità chiave dei moderni software CAD, permette di creare modelli 3D in cui le modifiche a un parametro si propagano automaticamente all'intera geometria, facilitando l'adattamento dei modelli a diverse configurazioni anatomiche. Software CAD avanzati come **Materialise 3-Matic** [7] e **Amira** [8] e **Autodesk Inventor** [9], offrono strumenti potenti per la modellazione 3D a partire da dati di imaging.

La generazione di mesh di volume a partire dai modelli 3D creati con CAD è essenziale per la simulazione del comportamento di tessuti e organi. Software di meshing automatico come **ANSYS Meshing** [10], **HyperMesh** [11] e **TetGen** [12] consentono di generare mesh di volume con diverse tipologie di elementi, come tetraedri e esaedri. Le mesh esaedriche offrono una maggiore precisione per le simulazioni, ma sono più difficili da generare rispetto alle mesh tetraedriche. Vengono utilizzate tecniche di ottimizzazione della mesh per migliorarne la qualità e ridurre gli errori di discretizzazione, garantendo la convergenza e l'accuratezza delle simulazioni.

La presente tesi propone una metodologia innovativa per la modellazione 3D di strutture vascolari a partire da dati geometrici pre-esistenti, utilizzando Autodesk Inventor come software CAD. L'obiettivo è di superare le limitazioni delle tecniche esistenti e di contribuire al progresso della modellazione patient-specific in ambito biomedicale.

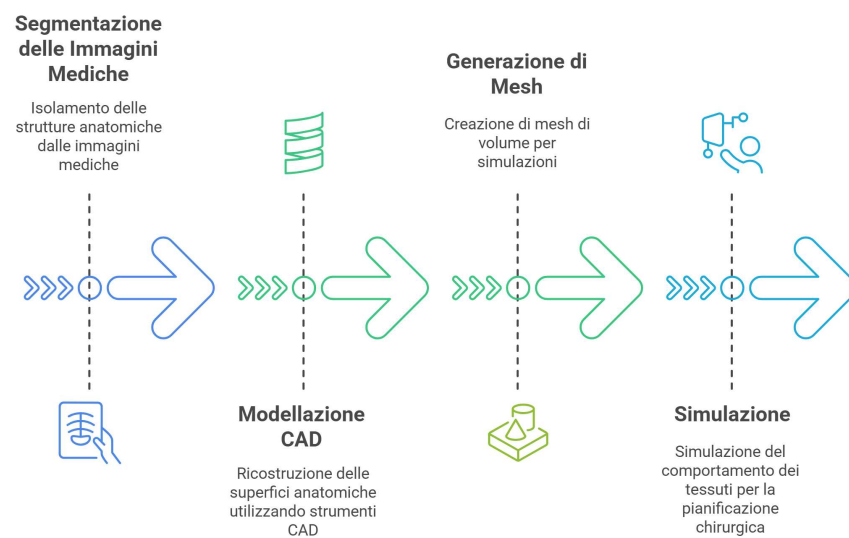


Figura 1.1: Passaggi per l'ottenimento di geometrie Patient Specific

Per identificare le metodologie più avanzate nella modellazione di reti vascolari, con particolare attenzione alla generazione di modelli 3D patient-specific e alla loro applicazione, ho condotto una ricerca approfondita in letteratura. Questa ricerca ha evidenziato come la modellazione patient-specific sia diventata uno strumento cruciale in ambito biomedico, consentendo la diagnosi, la pianificazione di interventi e la simulazione ad esempio del flusso sanguigno. In questo contesto, la generazione di modelli 3D accurati e affidabili è fondamentale per ottenere risultati clinicamente rilevanti.

In particolare Kretschmer et al.(2013) [13] si sono concentrati sulla modellazione interattiva di strutture vascolari, proponendo un metodo basato su superfici di sweep implicite per superare le limitazioni delle tecniche di segmentazione automatica. Riconoscendo che queste ultime non sempre producono modelli perfetti, specialmente in presenza di patologie, il loro approccio consente correzioni manuali efficienti, riducendo la necessità di rielaborare le segmentazioni. L'obiettivo finale è fornire modelli accurati per la simulazione del flusso sanguigno (CFD), sottolineando l'importanza di modelli di alta precisione per ottenere risultati affidabili nelle simulazioni emodinamiche.

Taylor et al. (2023) [14] hanno dimostrato l'importanza dei modelli computazionali paziente-specifici del flusso sanguigno coronarico, un'area di grande rilevanza clinica data l'alta incidenza delle malattie cardiovascolari. Il loro lavoro fornisce una panoramica completa delle diverse tecniche di modellazione utilizzate, dai modelli a parametri concentrati (0D) ai modelli di propagazione dell'onda (1D) e ai modelli tridimensionali (3D), con particolare enfasi sulla FFRCT, una tecnica non invasiva per valutare il significato funzionale delle stenosi coronariche. Lo studio evidenzia come la modellazione paziente-specifica del flusso sanguigno coronarico migliori la precisione diagnostica e offra potenziali applicazioni nella pianificazione del trattamento e nella previsione del rischio di rottura delle placche aterosclerotiche.

Invece, Decroocq et al. (2023)[15] hanno affrontato la sfida della generazione di mesh esaedriche, particolarmente adatte per la simulazione del flusso sanguigno, proponendo un metodo automatico basato sulle linee centrali delle reti vascolari. Il loro approccio si distingue per la robustezza ai difetti delle linee centrali e la flessibilità nel modificare il modello vascolare, utile per studiare l'impatto della geometria e della topologia sull'emodinamica. La natura automatica del metodo e il rilascio pubblico del codice sorgente favoriscono la riproducibilità e l'ulteriore sviluppo della tecnica.

In oltre l'articolo di Wu et al.(2011)[16] offre un contributo significativo, presentando un approccio innovativo per la segmentazione e la ricostruzione di strutture vascolari. Il loro metodo si articola in diverse fasi, ciascuna mirata a superare le limitazioni delle tecniche convenzionali. Il metodo proposto si basa sull'estrazione della centerline dei vasi sanguigni, ottenuta mediante scheletrizzazione della struttura segmentata. La centerline rappresenta l'asse longitudinale del vaso e fornisce un'informazione geometrica essenziale per la ricostruzione.

Oltre alla centerline, vengono stimate le sezioni trasversali del vaso, approssimando la loro forma con cerchi o ellissi. Questa stima può essere effettuata utilizzando diverse tecniche, come l'adattamento di cerchi o ellissi ai contorni del vaso segmentato. La ricostruzione della superficie vascolare viene eseguita a partire dalla centerline e dalle sezioni trasversali stimate.

Wu et al. (2011) propongono un approccio che preserva la topologia del vaso e genera patch quadrilatero, ovvero superfici quadrate o rettangolari che approssimano la superficie del vaso.

La ricerca ha inoltre evidenziato l'importanza dell'automazione per ridurre i tempi e gli sforzi nella modellazione 3D, nonché la necessità di modelli accurati per la simulazione del flusso sanguigno.

1.2 Obiettivo della tesi

Per superare le limitazioni del formato STL, questa tesi si propone di sviluppare una metodologia integrata e innovativa per la modellazione tridimensionale di strutture vascolari a partire da dati geometrici pre-esistenti, forniti in un file di testo. Questo approccio innovativo, basato sull'integrazione di Matlab, Inventor e VBA, permette di superare le difficoltà legate alla gestione manuale di grandi quantità di informazioni geometriche.

Nello specifico, la ricerca si concentra sull'elaborazione dei dati geometrici, quali le coordinate dei punti che definiscono la linea centrale e i diametri corrispondenti del vaso. Questi dati, ottenuti tramite un processo di elaborazione preliminare svolto con un software specifico (in questo caso Materialise), vengono importati in ambiente Matlab per l'estrazione delle informazioni necessarie alla creazione del modello 3D. Successivamente, attraverso l'utilizzo di codice VBA in ambiente Inventor, le coordinate vengono usate come workpoints per creare la center line del vaso e le circonferenze, poste normali alla curva, permettono la generazione tridimensionale del vaso.

La tesi si propone di sviluppare un flusso di lavoro efficiente e automatizzato per la modellazione 3D di strutture vascolari a partire da dati pre-esistenti, superando le difficoltà legate alla gestione manuale di grandi quantità di informazioni geometriche. Si mira ad ottenere modelli 3D accurati e affidabili, che possano essere utilizzati per diverse applicazioni in ambito biomedico, quali la simulazione del flusso sanguigno per la diagnosi di aneurismi, la pianificazione di interventi chirurgici per la riparazione di malformazioni vascolari o la progettazione di stent personalizzati.



Figura 1.2: Fasi della tesi

Capitolo 2

Autodesk Inventor

2.1 Presentazione di Inventor

Autodesk Inventor è un software CAD 3D leader nel settore, sviluppato da Autodesk, ampiamente utilizzato in diversi settori industriali, tra cui l'ingegneria meccanica, l'automotive, l'aerospaziale e il biomedicale [17].

Si distingue per la sua capacità di offrire un ambiente di lavoro integrato per la progettazione, la simulazione, la visualizzazione e la documentazione di modelli 3D.

Inventor consente ai progettisti di creare modelli parametrici, ovvero modelli le cui caratteristiche geometriche sono definite da parametri legati da relazioni logiche. La modifica di un parametro attiva una catena di aggiornamenti automatici, adattando tutti i parametri dipendenti e mantenendo l'integrità del modello. Questo approccio consente di automatizzare il processo di progettazione e di esplorare diverse varianti di un modello in modo rapido ed efficiente.

2.2 Interfaccia di Inventor

L'interfaccia utente di Inventor è intuitiva e personalizzabile. La finestra principale è suddivisa in diverse aree, tra cui la barra degli strumenti, il browser del modello, l'area di lavoro e la barra di stato.

La barra degli strumenti contiene i comandi per la creazione e la modifica di modelli 3D, mentre il browser del modello ne mostra la struttura gerarchica del modello, consentendo di visualizzare e gestire le diverse feature che lo compongono.

L'area di lavoro è lo spazio in cui vengono create e manipolate le geometrie 3D, mentre la barra di stato fornisce informazioni sul comando attivo e sullo stato del modello.

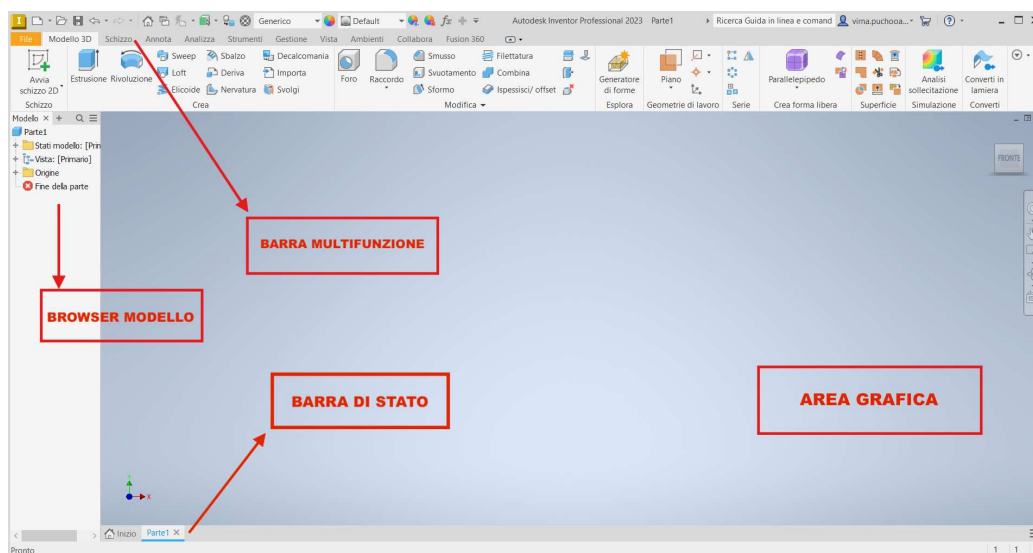


Figura 2.1: Informazioni presenti nell'interfaccia di Inventor

2.3 Funzionalità di modellazione

Inventor offre una vasta gamma di funzionalità per la modellazione 3D, tra cui:

- Creazione di solidi e superfici: Inventor permette di creare solidi 3D utilizzando diverse tecniche, come l'estrusione, la rivoluzione, lo sweep e il loft. È anche possibile creare superfici complesse utilizzando spline e superfici di Bézier.
- Feature: Le feature sono operazioni di modellazione predefinite che consentono di creare elementi geometrici specifici, come fori, raccordi e smussi.
- Vincoli e parametri: Inventor permette di definire vincoli geometrici e dimensionali tra le diverse parti di un modello, consentendo di adattarsi automaticamente alle modifiche.
- Assiemi: Inventor permette di creare assiemi complessi, combinando diverse parti in un unico modello. È possibile definire vincoli tra le parti per simulare il comportamento meccanico dell'assieme.

2.4 Funzionalità di Inventor utilizzate nel progetto

Per la realizzazione dei modelli 3D di strutture vascolari, sono state utilizzate diverse funzionalità di Autodesk Inventor, scelte in base alle specifiche esigenze del progetto. Di seguito vengono descritte le principali:



Figura 2.2: Elementi di disegno per la creazione dello schizzo 2D

- **Linea:** La funzione *Linea* è stata utilizzata per creare i tratti rettilinei che compongono la struttura vascolare. Inventor offre due modalità di creazione di linee:
 - *Linea 2D:* Consente di disegnare linee rette su un piano 2D.
 - *Linea 3D:* Permette di creare linee nello spazio 3D, definendo i punti di inizio e fine.

Per il progetto, sono state utilizzate entrambe le modalità, a seconda del contesto specifico.

- **Cerchio:** La funzione *Cerchio* è stata utilizzata per creare le sezioni trasversali dei vasi sanguigni. È possibile definire il cerchio specificando il centro e il raggio, oppure il centro e un punto sulla circonferenza.

Per la generazione di geometrie 3D a partire dal disegno, sono state selezionate due funzioni specifiche, considerate più adatte allo scopo:

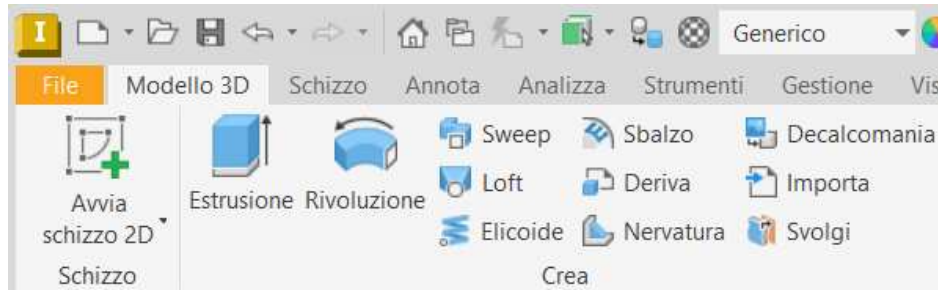


Figura 2.3: Funzionalità per lo sviluppo del modello 3D

- **Sweep:** La funzione *Sweep* consente di creare un solido 3D, muovendo un profilo 2D (sezione trasversale) lungo un percorso definito (traiettoria).
- **Loft:** La funzione *Loft* permette di creare un solido 3D a partire da due o più profili 2D che ne definiscono le sezioni trasversali, interpolando autonomamente i volumi che le connettono..

2.5 Il formato .ipt : elementi costitutivi delle parti in Inventor

In Autodesk Inventor, le parti 3D sono create e vengono archiviate nel formato nativo *.ipt*. Questi file racchiudono ogni dettaglio del modello: dalla geometria solida alle feature applicate (estrusioni, raccordi, smussi), dagli schizzi 2D che ne definiscono le forme ai vincoli che ne regolano le relazioni, fino all'intera storia di costruzione.

La struttura parametrica dei file *.ipt* offre una notevole flessibilità. Infatti, sono il punto di partenza per la creazione di assiemi complessi (*.iam*), ovvero la combinazione di più parti, vincolate da relazioni spaziali, in un'unica struttura che simula il comportamento del prodotto finito.

2.5.1 .ipt vs .stl: due formati a confronto

Per condividere i modelli con altri software CAD o per prepararli alla fabbricazione tramite stampa 3D, è necessario convertirli in formati standard. Uno dei più diffusi è il formato *.stl*. Pur rappresentando entrambi modelli 3D, *.ipt* e *.stl* sono profondamente diversi.

Un file *.ipt* è un modello parametrico completo, che include la geometria 3D, la storia di costruzione, gli schizzi, i vincoli e le feature.

Inoltre, i file *.ipt* tendono ad essere significativamente meno pesanti in termini di dimensioni rispetto ai file *.stl*.

Un file *.stl*, invece, è una rappresentazione semplificata della geometria 3D, costituita da una mesh di triangoli. Questa mesh è ideale per la stampa 3D, ma sacrifica la ricchezza di informazioni del file *.ipt*. A differenza di un file *.ipt*, un file *.stl* non conserva la storia di costruzione, i vincoli parametrici o le feature originali.

Di conseguenza, le modifiche dirette a un file *.stl* sono complesse e imprecise, richiedendo software specifici per la manipolazione di mesh.

In sintesi, i file *.ipt* sono lo strumento ideale per la progettazione 3D e la creazione di modelli parametrici mentre i file *.stl*, pur essendo essenziali per la stampa 3D, possono introdurre imprecisioni.

Capitolo 3

Approccio mediante modifica diretta di file STEP

3.1 Introduzione

In questo capitolo vengono illustrate le prime metodologie di modellazione 3D che sono state considerate e implementate nel corso della presente ricerca. Verrà fornita una panoramica delle principali tecniche di modellazione, con un focus particolare sulla modellazione di superfici complesse, che risultano particolarmente rilevanti nel contesto delle strutture vascolari.

3.2 Utilizzo di file STEP

Il formato STEP (Standard for the Exchange of Product Model Data) è uno standard ISO per lo scambio di dati di modelli 3D tra diversi software CAD [18]. La sua struttura modulare consente di rappresentare sia la geometria che le informazioni topologiche di un oggetto.

3.2.1 Primo approccio con file STEP

Inizialmente, l'approccio alla modellazione 3D si è basato sull'utilizzo diretto di file STEP. L'idea di base era quella di creare un modello 3D semplificato in ambiente CAD e di esportarlo in formato STEP, per poi modificarlo e adattarlo alle specifiche esigenze.

Questo approccio si basava sul presupposto che il file STEP, essendo un file di testo, potesse essere facilmente manipolato per modificare i dati geometrici. Inoltre, il formato STEP presenta il vantaggio di essere più leggero rispetto

ad altri formati, come STL, e di essere apribile con software di visualizzazione gratuiti.

3.2.2 Composizione di uno STEP file

L'informazione contenuta in un file STEP dipende dal disegno eseguito all'interno di Inventor.

L'obiettivo è fornire al lettore una comprensione di base della struttura di un file STEP e delle informazioni geometriche che contiene.

Creazione della geometria di partenza in Inventor

Come primo passo, viene creata una geometria 3D semplice in ambiente Autodesk Inventor. Questa geometria è composta da due elementi di base:

- Una linea retta orizzontale, disposta lungo l'asse X.
- Una linea retta inclinata, che si interseca con la linea orizzontale.

Questi due elementi rappresentano i componenti fondamentali della struttura vascolare: un tratto rettilineo e una biforcazione.

Passaggi per la creazione della geometria

1. **Nuovo file di parte:** Apro Autodesk Inventor e creo un nuovo file di parte (.ipt).

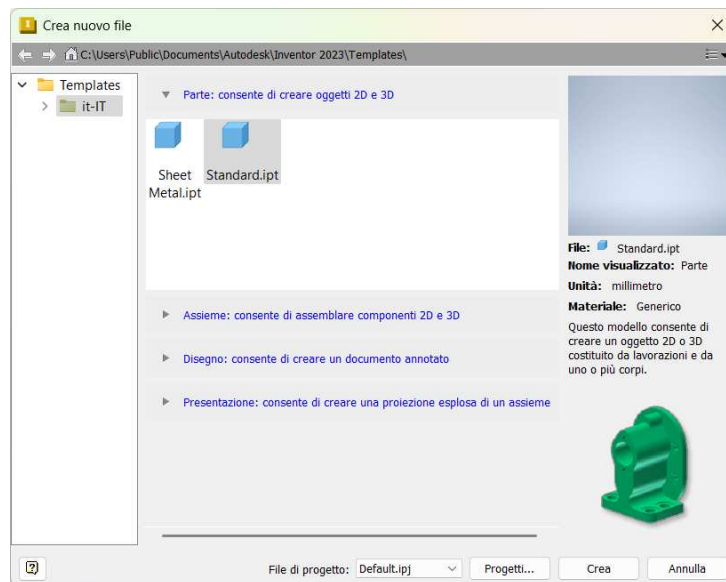


Figura 3.1: creare un nuovo file di parte

2. **Origine e piani:** Selezionando *Avvia Schizzo 2D*, come si vede nella 2.3 si visualizzerà l'origine e i piani di riferimento (XY, YZ, XZ).

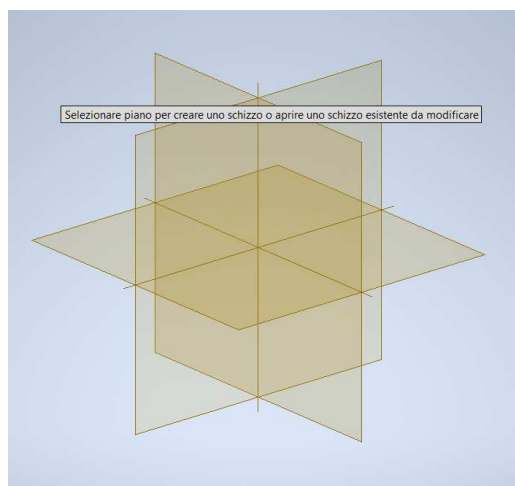


Figura 3.2: Visualizzazione dei diversi piani su cui iniziare il disegno

3. Schizzo 1 (linee):

- (a) Seleziono il piano XY e creo un nuovo schizzo 2D su di esso.
- (b) Disegno una linea orizzontale lungo l'asse X della lunghezza desiderata.
- (c) Partendo da un'estremità della linea orizzontale, disegno una linea diagonale verso l'alto, sempre sul piano XY.
- (d) Concludo lo schizzo.

4. **Creazione piano di lavoro:** Creo un piano di lavoro perpendicolare alle due linee create nello schizzo 1 e passante per il punto di congiunzione tra le due linee. Posso usare diversi metodi per creare questo piano, ad esempio selezionando le due linee e scegliendo l'opzione *Piano normale a due linee*.

5. Schizzo 2 (cerchio):

- (a) Creo un nuovo schizzo 2D sul piano di lavoro appena creato.
- (b) Disegno un cerchio con il diametro desiderato nel punto di intersezione tra il piano di lavoro e le linee (o l'estrusione delle linee se create).
- (c) Concludo lo schizzo.

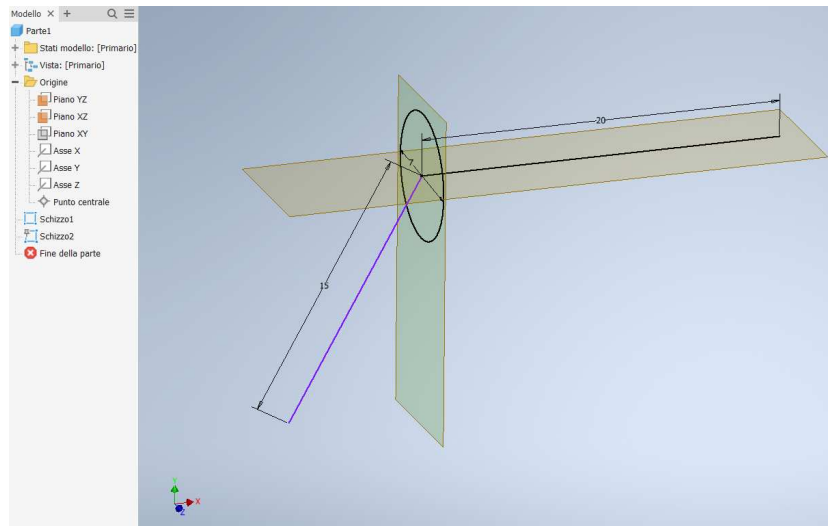


Figura 3.3: Schizzo 2D iniziale

6. Sweep:

- (a) Utilizzo la funzione *Sweep* per creare la parte finale.
- (b) Selezione il profilo (cerchio) e il percorso (combinazione delle due linee dello schizzo 1) per eseguire la sweep. Inventor dovrebbe automaticamente interpolare fra le due linee per creare il percorso.

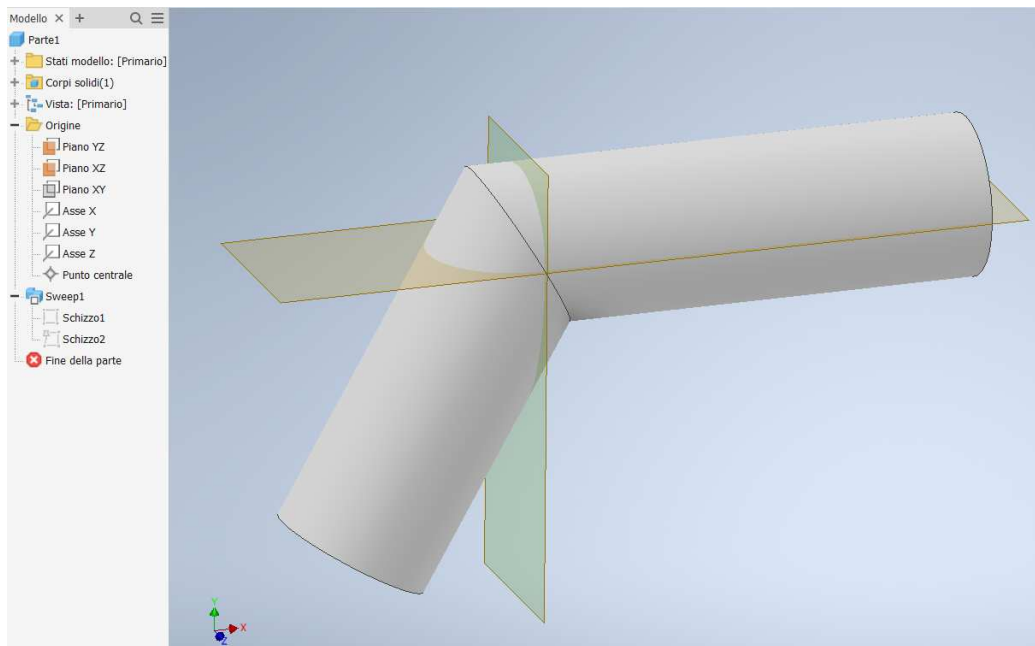


Figura 3.4: Modello 3D finale, con sweep

7. **Salvataggio:** Salvo il file di parte (.ipt).

Esportazione in formato STEP

Una volta completata la geometria 3D, il modello viene salvato e poi esportato in formato STEP. Questo formato è uno standard ISO per lo scambio di dati di modelli 3D tra diversi software CAD.

Per esportare un disegno di Inventor in formato STEP, si può procedere come segue:

- **1:** Dalla barra in alto, nella sezione *Annota*, si trova l'opzione *Formato CAD*. Cliccando su di essa, si apre una nuova finestra.
- **2:** Questa finestra permette di esportare il file in diversi formati. In particolare, selezionare *file STEP*.

L'esportazione in formato STEP produce un file di testo che contiene la descrizione geometrica del modello 3D appena disegnato.

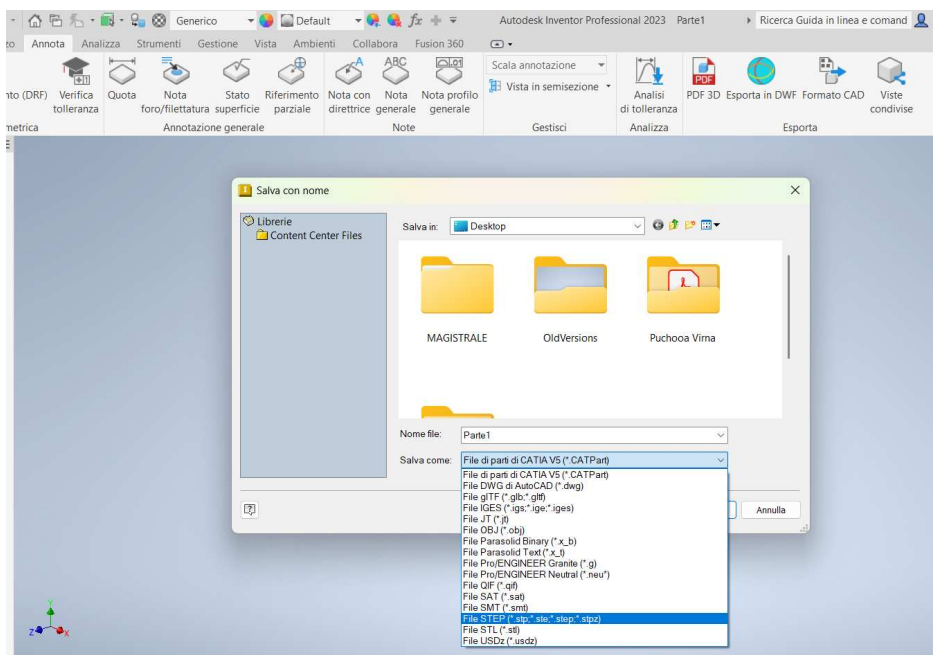


Figura 3.5: come esportare il file .ipt in file STEP

Analisi del file STEP

Il file STEP generato viene aperto con un editor di testo per analizzarne il contenuto. Il file è strutturato in una serie di entità, ognuna delle quali rappresenta un elemento geometrico o topologico del modello 3D.

L'analisi del file STEP consente di comprendere come vengono rappresentate le informazioni geometriche, come le coordinate dei punti, le equazioni delle linee e delle superfici, e le relazioni topologiche tra gli elementi.

```
ISO-10303-21;
HEADER;
/* Generated by software containing ST-Developer
 * from STEP Tools, Inc. (www.steptools.com)
 */

FILE_DESCRIPTION(
/* description */ (''),
/* implementation_level */ '2;1');

FILE_NAME(
/* name */ 'Parte3.stp',
/* time_stamp */ '2024-06-21T12:58:10+02:00',
/* author */ ('Utente'),
/* organization */ (''),|
/* preprocessor_version */ 'ST-DEVELOPER v19.2',
/* originating_system */ 'Autodesk Inventor 2023',
/* authorisation */ '');

FILE_SCHEMA (('AUTOMOTIVE_DESIGN { 1 0 10303 214 3 1 1 }'));
ENDSEC;

DATA;
#10=MECHANICAL_DESIGN_GEOMETRIC_PRESENTATION_REPRESENTATION('',(#14),#46);
#11=SHAPE_REPRESENTATION_RELATIONSHIP('SRR','None',#53,#12);
#12=GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATION('',(#13),#45);
#13=GEOMETRIC_CURVE_SET('Schizzo 3D1',(#15));
#14=STYLED_ITEM('',(#63),#15);
#15=POLYLINE('',(#41,#42));
#16=DERIVED_UNIT_ELEMENT(#19,1.);
#17=DERIVED_UNIT_ELEMENT(#48,-3.);
#18=DIMENSIONAL_EXPONENTS(0.,1.,0.,0.,0.,0.,0.);
#19=(
CONVERSION_BASED_UNIT('gram',#21)
MASS_UNIT()
```

Figura 3.6: Esempio STEP file

Di seguito viene riportato il contenuto del file STEP analizzato, relativo al modello 3D di riferimento. Il file è di tipo ISO-10303-21 e descrive un modello 3D come l'esempio riportato in figura 3.6.

Troviamo informazioni relative alla geometria, alle proprietà dei materiali, alle unità di misura e ad altri aspetti del componente, in conformità con la serie di standard internazionali *ISO-1030* che definiscono un formato di dati neutro per rappresentare e scambiare informazioni relative ai prodotti durante il loro ciclo di vita.

Il file si apre con una sezione **HEADER** che fornisce informazioni generali sul contenuto, tra cui:

- **FILE_DESCRIPTION**: Descrizione del file e livello di implementazione (2.1).
- **FILE_NAME**: Nome del file (*Parte3.stp*), data di creazione (21 giugno 2024) e altre informazioni come autore e software utilizzato (Autodesk Inventor 2023).
- **FILE_SCHEMA**: Schema utilizzato (*AUTOMOTIVE_DESIGN*), specifico per il design automobilistico.

La sezione **DATA** contiene la parte principale del file, che descrive la geometria dell'oggetto e altre proprietà. La rappresentazione geometrica del solido è definita da

#10 **MECHANICAL_DESIGN_GEOMETRIC_PRESENTATION_REPRESENTATION**, collegata a un oggetto geometrico più complesso, #14 **MANIFOLD_SOLID_BREP**, che descrive un solido 3D tramite una rappresentazione di confine (BREP) e denominato *Solido1*. Il solido è un *guscio chiuso* #66 **CLOSED_SHELL**, costituito da facce connesse.

Gli elementi geometrici includono #25 **ELLIPSE**, #32 **CIRCLE**, #28 **LINE** (ellissi, cerchi, linee e curve di bordo #39 **EDGE_CURVE**). Le facce e i loop sono definiti da #17 **FACE_OUTER_BOUND** (bordo esterno di una faccia associata a un **EDGE_LOOP** come #21) e #62 **ADVANCED_FACE** (faccia avanzata legata a una superficie cilindrica).

Il posizionamento è definito da #88 **AXIS2_PLACEMENT_3D** (posizionamenti 3D per sistema di riferimento e posizione dell'oggetto). Le proprietà dei materiali includono #79 **DESCRIPTIVE_REPRESENTATION_ITEM** (materiale generico) e #74 **MEASURE_REPRESENTATION_ITEM** (misura della densità). Le proprietà del prodotto #149 **PRODUCT_DEFINITION** (*Primo_disegno*) sono legate a materiale e densità.

Il contesto applicativo è definito da

#152 APPLICATION_PROTOCOL_DEFINITION (design automobilistico). La data di creazione #84 DATE_AND_TIME è il 13 novembre 2024.

Gli stili di superficie e colore sono definiti da

#156 PRESENTATION_STYLE_ASSIGNMENT (stili di presentazione) e

#164 COLOUR_RGB (colore grigio chiaro).

3.2.3 Funzione SPLINE

Le spline sono curve matematiche utilizzate per rappresentare forme complesse in modo flessibile e accurato. In ambito CAD, le spline vengono spesso utilizzate per la modellazione di superfici, raccordi e altre geometrie complesse.

La loro versatilità le rende uno strumento fondamentale in diversi ambiti, tra cui la progettazione industriale, la grafica computerizzata.

Definizione e tipologie di spline

Formalmente, una spline è una curva definita da una serie di punti di controllo (o nodi) e da una funzione matematica che interpola o approssima questi punti. Esistono diverse tipologie di spline, ognuna con caratteristiche specifiche:

- **Spline di Bézier:** Definite da un insieme di punti di controllo e utilizzate per creare curve lisce e uniformi.
- **Spline B-spline:** Generalizzazione delle spline di Bézier, offrono un maggiore controllo sulla forma della curva e sono meno sensibili alla posizione dei punti di controllo.
- **Spline NURBS (Non-Uniform Rational B-Splines):** Rappresentazione unificata di curve e superfici, ampiamente utilizzate in ambito CAD per la loro flessibilità e capacità di rappresentare forme complesse.

Utilizzo delle spline in Autodesk Inventor

Autodesk Inventor offre diverse funzionalità per la creazione e la modifica di spline, consentendo ai progettisti di modellare curve complesse in modo intuitivo. In particolare, Inventor supporta due modalità principali di disegno di spline:

- **Spline di interpolazione:** Questa modalità consente di creare una spline che passa *esattamente* attraverso una serie di punti definiti dall'utente. È possibile specificare i punti di controllo uno per uno, oppure importarli da un file esterno. Le spline di interpolazione sono utili quando si desidera creare una curva che rispetti precisi vincoli geometrici.
- **Spline di vertici di controllo:** In questa modalità, la spline *si avvicina* a una serie di punti, senza necessariamente passarci attraverso. L'utente può specificare il grado di vicinanza della curva ai punti, ottenendo un maggiore controllo sulla forma complessiva della spline. Le spline di approssimazione sono utili quando si desidera creare una curva liscia e uniforme, senza la necessità di rispettare vincoli geometrici precisi.

Modellazione di strutture vascolari con spline

Nel contesto della modellazione 3D di strutture vascolari, è conveniente usare le spline in quanto funzioni che permettono di controllare la tortuosità della curva che si vuole disegnare. Essendo le strutture vascolari caratterizzate da un andamento non sempre uniforme, ma variabile e tortuoso, le spline si presentano come lo strumento ideale per la modellazione di tali geometrie. La loro flessibilità consente di modellare sia tratti rettilinei che curve complesse, come biforcazioni o anastomosi, connessione o unione tra due strutture anatomiche, come vasi sanguigni o nervi.

In oltre la combinazione di spline e altre funzionalità di Inventor, come Sweep e Loft, consente di creare modelli 3D complessi di strutture vascolari, che possono essere utilizzati per diverse applicazioni, tra cui la simulazione del flusso sanguigno, la pianificazione di interventi chirurgici o la progettazione di dispositivi medici.

La SPLINE nello STEP FILE

In generale, le spline vengono rappresentate nel formato STEP attraverso entità specifiche, che ne definiscono la geometria e le proprietà. Tra le entità più comuni troviamo:

- **POLYLINE**: questa entità definisce una polilinea, ovvero una sequenza di segmenti di linea connessi. Le polilinee possono essere utilizzate per approssimare curve più complesse, come le spline. Ogni segmento della polilinea è definito da due punti, le cui coordinate sono specificate all'interno dell'entità.
- **BSPLINE_CURVE**: questa entità rappresenta una curva B-spline, una tipologia di spline molto utilizzata in ambito CAD. La definizione di una curva B-spline include i punti di controllo, il grado della curva e i pesi dei punti di controllo. Questi parametri determinano la forma della curva.
- **NURBS_CURVE**: questa entità rappresenta una curva NURBS (Non-Uniform Rational B-Spline), una generalizzazione delle curve B-spline. Le curve NURBS sono in grado di rappresentare una vasta gamma di forme, incluse le curve coniche e le superfici.

Esempio Spline di interpolazione

Segue un esempio pratico dell'utilizzo della spline con punti di interpolazione.

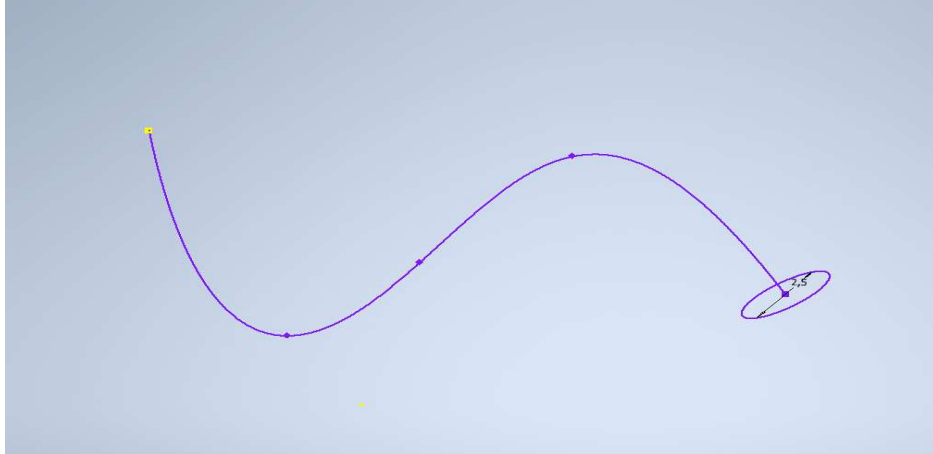


Figura 3.7: Rappresentazione del disegno 2D ottenuto realizzando due sketch: il primo la spline di interpolazione, con 5 punti definiti e il secondo la circonferenza di raggio 2.5.

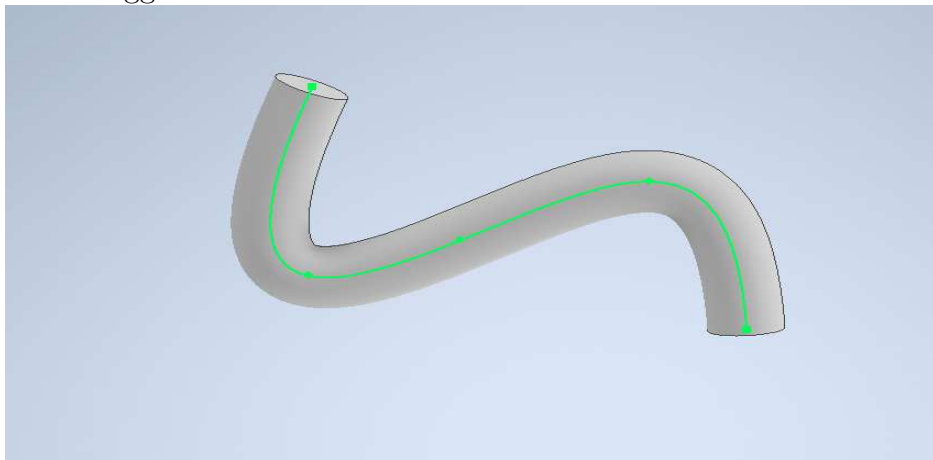


Figura 3.8: Modello 3D finale, ottenuto mediante l'estrusione *sweep* del profilo circolare lungo la traiettoria definita dalla curva

Figura 3.9: Passaggi di costruzione modello 3D.

indicano a che punto del cursore corrisponde ogni punto della curva e sono cruciali per la sua forma finale. La presenza di così tanti attributi sottolinea la complessità del formato STEP e la difficoltà di modificare direttamente le spline.

3.2.4 Limitazioni nell'editing della spline tramite il formato STEP

Come abbiamo visto finora, l'utilizzo di spline per la modellazione di geometrie, seppur efficace e versatile, presenta delle sfide significative, soprattutto quando si considera l'interoperabilità e la modifica dei modelli. Questo aspetto particolarmente importante in quanto il disegno di partenza è solo una geometria semplificata del modello.

Analizziamo ora perché la gestione di spline e file STEP, in particolare, non si adatta in modo ottimale ai nostri obiettivi specifici.

La definizione matematica delle spline può risultare complessa, e la loro implementazione in ambiente CAD richiede competenze specifiche [19].

Questa complessità si manifesta in modo particolare quando si considera la rappresentazione delle spline nel formato STEP.

Come abbiamo visto, il file STEP contiene una miriade di informazioni relative alla spline: non solo i punti di controllo originali, ma anche una fitta approssimazione della curva, tolleranze, tipo di raccordo, sistemi di coordinate e altre proprietà geometriche e topologiche.

Questa ricchezza di informazioni, se da un lato garantisce l'interoperabilità e la riproducibilità del modello in qualsiasi software CAD, dall'altro rende estremamente complessa la modifica della spline.

Sia il formato STEP che l'ambiente di lavoro Inventor, infatti, non forniscono una *formula matematica* esplicita per la spline.

La curva è definita attraverso una serie di punti e parametri interconnessi, ma non è possibile risalire facilmente all'equazione che la descrive.

Di conseguenza, modificare la spline non è un semplice aggiustamento di parametri, ma un processo laborioso che richiede la ricalibrazione di numerosi dati. Aggiungere anche un solo punto di controllo, o alterare la posizione di uno esistente, innesca una cascata di modifiche per preservare la coerenza geometrica.

Questa complessità intrinseca del formato STEP evidenzia una sfida importante nella gestione di modelli 3D. Sebbene il formato STEP sia fondamentale per lo scambio di dati, la sua struttura rende le modifiche precise e mirate un compito arduo, che richiede una profonda comprensione del formato e l'utilizzo di strumenti software specializzati.

Inoltre, l'obiettivo di inserire ulteriori punti che facciano parte della spline di partenza si scontra con la difficoltà di reperire e interpretare le numerose informazioni necessarie per una modifica coerente della spline all'interno del file STEP. Pertanto, per raggiungere i nostri obiettivi di modellazione, dobbiamo considerare approcci alternativi che offrano un maggiore controllo e flessibilità nella gestione delle spline.

Capitolo 4

Automazione mediante VBA

VBA (Visual Basic for Applications) è un linguaggio di programmazione sviluppato da Microsoft che consente di automatizzare e personalizzare applicazioni appartenenti alla suite Microsoft Office, come Excel, Word, Access e Outlook, nonché altre applicazioni che supportano l'integrazione con VBA, tra cui Autodesk Inventor. VBA si basa sul **linguaggio di programmazione Visual Basic** e consente agli utenti di scrivere **macro** (sequenze di istruzioni) per eseguire operazioni ripetitive, manipolare dati, interagire con l'interfaccia utente e personalizzare comportamenti delle applicazioni.

VBA in Autodesk Inventor

In Autodesk Inventor, VBA permette di controllare il software, interagire con i modelli 3D, automatizzare processi di disegno, analisi e gestione dei dati. Grazie a VBA, è possibile creare script che eseguono automaticamente operazioni complesse, come la generazione di geometrie, la modifica di parametri, l'esecuzione di analisi agli elementi finiti o la creazione di report personalizzati.

Configurazione di Inventor, VBA per il primo utilizzo

Per poter iniziare a scrivere un codice VBA all'interno di Inventor, è necessario innanzitutto creare una parte, poi selezionare strumenti dalla barra multifunzione.

1. **Aprire un documento parte:** Mi sono assicurata di avere un documento parte attivo in Inventor.
2. **Accedere alla scheda *Strumenti*:** Nella barra multifunzione (Ribbon) di Inventor, facendo clic sulla scheda *Strumenti*.
3. **Trovare il pannello *Opzioni*:** All'interno della scheda *Strumenti*, cerco il pannello *Opzioni*.
4. **Fare clic sull'icona *Macro*:** Nel pannello *Opzioni*, si trova l'icona *Macro* (evidenziata in un cerchio rosso nell'immagine4.1). Ho cliccato su questa icona.
5. **Apertura dell'editor di Visual Basic:** Facendo clic sull'icona *Macro*, si aprirà l'editor di Visual Basic, dove è possibile iniziare a scrivere e modificare il codice VBA.

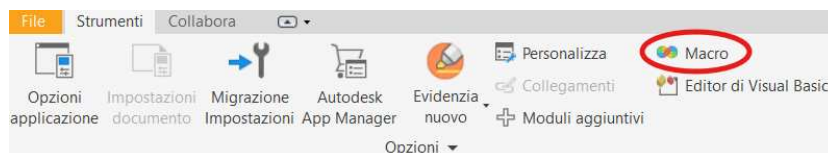


Figura 4.1: Icona della Macro, da qui è possibile accedere all'interfaccia di scrittura di VBA da Inventor.

L'icona *Macro* funge da punto di accesso all'ambiente VBA in Inventor. Cliccando su di essa, si avvia l'editor di Visual Basic, che consente di:

- Scrivere e modificare il codice VBA.
- Creare e gestire macro.
- Accedere all'API (Application Programming Interface) di Inventor, che permette di controllare e automatizzare le funzionalità del software.

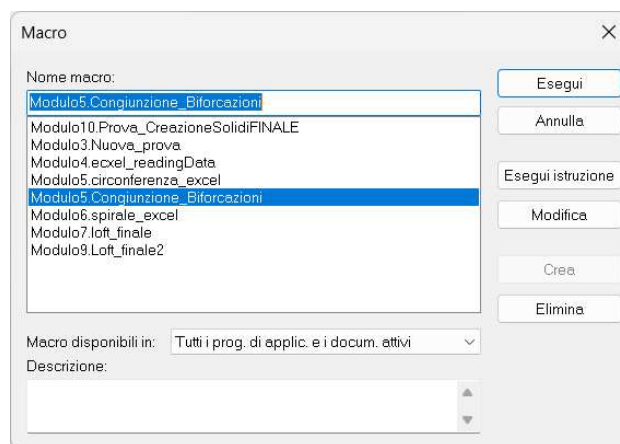


Figura 4.2: Finestra di dialogo *Macro*, in Inventor, dove si possono creare, modificare o eliminare i moduli.

Interazione VBA Excel

Siccome il progetto necessita l'uso di Excel per la lettura e la manipolazione dei dati, è necessario eseguire un'ultima configurazione prima di poter eseguire correttamente il codice VBA in Inventor.

Questa configurazione consiste nell'aggiungere un riferimento alla libreria degli oggetti di Excel all'interno dell'ambiente VBA di Inventor.

Questo passaggio è fondamentale per poter utilizzare le funzioni e gli oggetti di Excel all'interno del codice VBA.

In 4.4 è riportata la finestra di dialogo *Riferimenti - ProgettoApplicazione*, che permette di gestire i riferimenti alle librerie degli oggetti disponibili. In questo caso, è stata selezionata la libreria *Microsoft Excel 16.0 Object Library*, che consente al codice VBA di accedere alle funzionalità di Excel. La selezione di questa libreria abilita l'utilizzo di oggetti come *Excel.Application*, *Workbook*, *Worksheet* e altri, che sono essenziali per leggere e manipolare i dati nei file Excel.

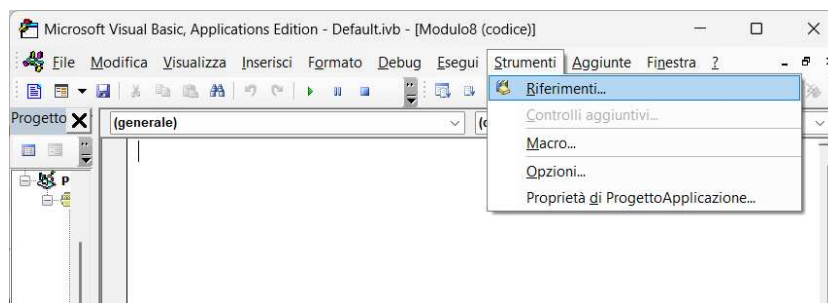


Figura 4.3: Come arrivare ad aggiungere la libreria degli oggetti di Excel.

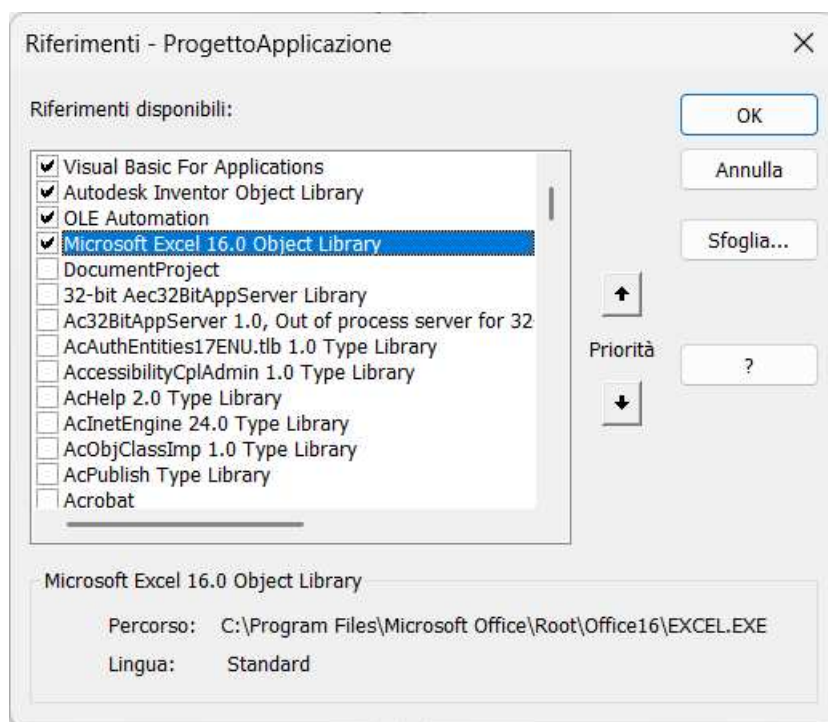


Figura 4.4: Finestra di dialogo *Riferimenti- ProgettoApplicazione* dove bisogna spuntare la libreria relativa ad Excel.

4.1 Automazione del processo di disegno con VBA Autodesk Inventor supportata da Matlab

L'obiettivo di questo sottocapitolo è descrivere il processo di automazione del disegno in Autodesk Inventor, utilizzando dati generati dal software Materialise e elaborati con Matlab.

Questa automazione consente di ridurre significativamente i tempi di progettazione e di migliorare la precisione, soprattutto quando si lavora con modelli 3D complessi derivati da immagini mediche. È importante notare che in questo lavoro è stato utilizzato Materialise come software ma che esistono diverse altre soluzioni di segmentazione e ricostruzione 3D che possono fornire dati di input per il processo di automazione descritto.

4.1.1 Flusso di lavoro di Materialise

Il file contenente le informazioni necessarie per la progettazione del modello è stato ottenuto utilizzando il software Materialise [20].

In generale, Materialise offre una suite di strumenti software per l'elaborazione di immagini mediche e la preparazione di dati per la produzione additiva.

Questo processo si articola nei seguenti passaggi principali:

1. **Acquisizione delle immagini mediche:** Si parte dall'acquisizione di immagini mediche, come quelle ottenute da CT o MRI. Queste immagini forniscono una serie di slice bidimensionali del corpo umano o di un'area specifica di interesse.
2. **Importazione in Materialise Mimics:** Le immagini CT/MRI vengono importate nel software Materialise Mimics [21]. Mimics è specializzato nell'elaborazione di immagini mediche e consente di visualizzare e manipolare i dati 3D derivati dalle immagini 2D.
3. **Segmentazione:** La segmentazione è un passaggio cruciale. In questa fase, l'utente identifica e separa le diverse strutture anatomiche presenti nelle immagini (ad esempio ossa, tessuti molli, organi). Mimics offre strumenti avanzati per la segmentazione manuale, semi-automatica e automatica.

4. **Creazione del modello 3D:** Una volta completata la segmentazione, Mimics ricostruisce un modello 3D della struttura anatomica di interesse. Questo modello 3D è una rappresentazione digitale della geometria della struttura.
5. **Esportazione in Materialise 3-Matic:** Il modello 3D creato in Mimics può essere esportato e importato in Materialise 3-Matic [7]. 3-Matic è un software per la preparazione dei dati per la stampa 3D e offre strumenti per l'analisi, la riparazione e l'ottimizzazione dei modelli 3D.
6. **Analisi e misurazione in 3-Matic:** In 3-Matic, è possibile eseguire diverse analisi e misurazioni sul modello 3D. Ad esempio, è possibile:
 - Misurare distanze, angoli e volumi.
 - Analizzare la geometria di fori e canali interni.
 - Calcolare diametri di circonferenze.
 - Generare supporti per la stampa 3D.

In particolare, 3-Matic permette di analizzare le dimensioni interne e la geometria di fori e canali utilizzando la *tecnica della sfera inscritta*.

7. **Generazione del file di dati numerici:** 3-Matic genera un file di testo contenente le coordinate dei punti 3D e altre informazioni geometriche.

La tecnica della sfera inscritta, permette di analizzare le dimensioni interne e la geometria di fori e canali. Questa tecnica prevede:

1. **Posizionamento iniziale:** Il software posiziona una piccola sfera virtuale all'interno del volume del modello 3D.
2. **Ingrandimento progressivo:** La sfera viene progressivamente ingrandita, mantenendo il suo centro all'interno del volume.
3. **Rilevamento delle collisioni:** Man mano che la sfera si ingrandisce, il software monitora le collisioni tra la superficie della sfera e le pareti interne del modello 3D.
4. **Arresto e misurazione:** L'ingrandimento della sfera si arresta quando la sua superficie tocca le pareti interne. Il software misura il diametro della sfera, che rappresenta la dimensione massima del volume.

5. **Analisi dei risultati:** Il diametro fornisce informazioni sulle dimensioni interne e sulla geometria, utili per verificare la stampabilità, ottimizzare la progettazione, analizzare porosità e densità, e ottenere dati per l'automazione.

Quindi, il file di testo che Materialise costruisce alla fine contiene diverse grandezze geometriche.

Il file è strutturato con una legenda che descrive brevemente il significato di ciascun parametro:

Legend

=====

Order: curve order

Px,Py,Pz: coordinates of point P on the centerline

Tx,Ty,Tz: coordinates of the direction of the Tangent vector to point P

Nx,Ny,Nz: coordinates of the direction of the Normal vector to point P

BNx,BNy,BNz: coordinates of the direction of the Binormal vector to point P

Dfit:diameter of the best fit circle in point P

Dmin:diameter of the inscribing circle in point P

Dmax:diameter of the subscribing circle in point P

C:curvature in point P

Dh:hydraulic diameter in point P

Kh:hydraulic ratio in point P

Scf:distance of circumference in point P

Area:sectional area in point P

E:ellipticity in point P

Per il progetto specifico sono stati scelti solo alcuni di questi parametri: P_x , P_y , P_z e D_{fit} .

Le coordinate rappresentano la posizione di ciascun punto sulla linea centrale del modello 3D, fornendo le informazioni necessarie per ricostruire le *center lines*, ovvero la linea centrale che attraversa la lunghezza del vaso sanguigno, fornendo una rappresentazione schematica della sua traiettoria, in Inventor. Il parametro D_{fit} , invece, rappresenta il diametro del cerchio di migliore approssimazione nel punto P, fornendo una stima delle dimensioni locali del lume della struttura vascolare in esame.

Questi parametri sono stati selezionati perché sono fondamentali per la costruzione del modello 3D del vaso in Inventor, permettendoci di congiungere le centerlines e i diametri per ottenere una rappresentazione accurata della sua geometria.

Avendo identificato i parametri Px, Py, Pz e Dfit come essenziali per il nostro progetto, possiamo ora esaminare come questi vengono strutturati e rappresentati nel file di testo.

```
[Main Branch Set] Branch Segment 1:
```

Point (10.5359, -177.9185, -405.0381) connects to:

Branch Segment 13
Branch Segment 2

	Px	Py	Pz	Tx	Ty	Tz	Nx	Ny	Nz	BNx	BNy	BNz	Dfit
41.0846	-134.7597	-466.2466	-0.3986	0.0214	0.9169	-0.3532	-0.9262	-0.1319	0.8464	-0.3764	0.3767	8.2688	
38.8208	-134.7809	-461.1769	-0.4175	-0.0324	0.9081	-0.3059	-0.9360	-0.1740	0.8557	-0.3504	0.3809	8.0557	
36.3745	-135.1718	-456.0374	-0.4402	-0.1081	0.8914	-0.2581	-0.9356	-0.2409	0.8600	-0.3362	0.3839	7.9443	
33.5133	-136.1981	-450.5186	-0.4677	-0.2249	0.8548	-0.1821	-0.9218	-0.3421	0.8649	-0.3157	0.3901	7.8767	
30.4913	-138.1113	-445.3502	-0.4906	-0.3925	0.7780	-0.0677	-0.8729	-0.4831	0.8688	-0.2897	0.4017	9.1612	
27.5368	-141.0525	-441.0539	-0.4878	-0.5920	0.6416	0.0866	-0.7642	-0.6392	0.8687	-0.2562	0.4240	9.2653	
24.8219	-145.0044	-437.8433	-0.4440	-0.7652	0.4662	0.2627	-0.6086	-0.7487	0.8566	-0.2099	0.4713	9.0367	
22.4452	-149.7797	-435.6027	-0.3755	-0.8700	0.3196	0.4760	-0.4769	-0.7390	0.7953	-0.1253	0.5931	10.3212	
20.4299	-155.0476	-433.9473	-0.3130	-0.9133	0.2605	0.9273	-0.3532	-0.1240	0.2053	0.2028	0.9575	12.0822	
18.7357	-160.3927	-432.3445	-0.2714	-0.9123	0.3067	0.2977	0.2235	0.9281	-0.9153	0.3431	0.2109	11.7384	
17.2808	-165.3923	-430.2601	-0.2493	-0.8564	0.4521	0.0738	0.4487	0.8906	-0.9656	0.2554	-0.0487	10.6841	
15.9680	-169.6944	-427.2917	-0.2375	-0.7207	0.6513	0.0375	0.6631	0.7476	-0.9707	0.2020	-0.1305	11.4481	
14.7076	-173.0774	-423.2543	-0.2266	-0.5212	0.8228	0.0415	0.8389	0.5428	-0.9731	0.1571	-0.1684	12.0564	
13.4358	-175.4808	-418.2010	-0.2164	-0.3249	0.9206	0.0407	0.9392	0.3410	-0.9754	0.1113	-0.1900	11.4940	

Figura 4.5: Main Branch Set, Branch Segment 1, vengono definiti tutti i parametri del primo segmento della struttura centrale.

```
[Branch Set 1] Branch Segment 9:
```

Point (22.4512, -161.0416, -105.8606) connects from:

Branch Segment 6
Branch Segment 7

	Px	Py	Pz	Tx	Ty	Tz	Nx	Ny	Nz	BNx	BNy	BNz	Dfit
22.4512	-161.0416	-105.8606	-0.3586	-0.1069	0.9273	0.8275	-0.4962	0.2628	0.4321	0.8616	0.2664	27.0156	
21.6817	-161.4213	-103.2647	-0.1921	-0.1623	0.9679	0.9742	-0.1505	0.1681	0.1184	0.9752	0.1870	61.6910	
21.4617	-161.7536	-101.3451	-0.0247	-0.1719	0.9848	0.9994	0.0216	0.0288	-0.0262	0.9849	0.1712	59.6331	
21.5646	-162.1156	-99.1542	0.0752	-0.1474	0.9862	-0.5115	0.8433	0.1650	-0.8560	-0.5169	-0.0120	49.1072	
21.7348	-162.4353	-96.5386	0.0691	-0.0951	0.9931	0.6259	0.7793	0.0311	-0.7769	0.6194	0.1133	47.1258	
21.9899	-162.6213	-93.6971	0.1056	-0.0325	0.9939	0.3378	0.9412	-0.0051	-0.9353	0.3363	0.1103	46.6471	
22.3409	-162.6061	-90.8128	0.1384	0.0450	0.9894	0.4311	0.8967	-0.1011	-0.8916	0.4405	0.1047	47.8192	
22.7901	-162.3595	-88.0291	0.1796	0.1308	0.9750	0.4231	0.8845	-0.1965	-0.8881	0.4478	0.1035	9.9444	
23.3302	-161.8931	-85.4336	0.2221	0.2151	0.9510	0.4490	0.8432	-0.2956	-0.8655	0.4927	0.0907	9.3197	
23.9457	-161.2559	-83.0537	0.2619	0.2834	0.9225	0.5046	0.7746	-0.3813	-0.8227	0.5654	0.0599	9.1313	
24.6159	-160.5204	-80.8626	0.2935	0.3223	0.9000	0.6945	0.5751	-0.4324	-0.6570	0.7519	-0.0551	9.2801	
25.3187	-159.7652	-78.7949	0.3125	0.3250	0.8926	0.6092	-0.7895	0.0742	0.7288	0.5206	-0.4448	9.2163	
26.0343	-159.0577	-76.7669	0.3171	0.2952	0.9013	-0.0660	-0.9411	0.3315	0.9461	-0.1646	-0.2789	9.0845	
26.7476	-158.4400	-74.6979	0.3082	0.2465	0.9188	-0.2508	-0.9106	0.3284	0.9177	-0.3317	-0.2189	9.1769	
27.4492	-157.9236	-72.5275	0.2910	0.1969	0.9362	-0.3822	-0.8732	0.3024	0.8771	-0.4458	-0.1788	9.4706	

Figura 4.6: Branch Set 1 sono tutti i parametri relativi alla prima ramificazione. Branch Segment 9 invece indica la numerazione continua di tutti i segmenti.

Oltre alla definizione dei parametri, è possibile notare, in 4.5 e 4.6, la presenza di sezioni denominate *Main Branch Set*, *Branch Set* e *Branch Segment*, seguite da un numero.

Queste sezioni indicano la struttura gerarchica dei dati: i *Main Branch Set* rappresentano l'intera struttura vascolare centrale, i *Branch Set* rappresentano l'insieme delle ramificazioni che possono generarsi *dalla* branch centrale oppure *verso* la branch centrale, mentre i **Branch Segment** rappresentano la numerazione dei segmenti che permette di identificare univocamente pezzi di branch all'interno del modello.

Altro dato importante sono le congiunzioni tra i diversi segmenti che sono definite attraverso la presenza di righe che indicano il punto di congiunzione.

Queste righe, che iniziano con *Point (x, y, z) connects from/to* specificano le coordinate del punto in cui un segmento si connette ad altri segmenti.

Ad esempio, la riga

```
[Main Branch Set] Branch Segment 1:
```

```
Point (10.5359, -177.9185, -405.0381) connects to:  
    Branch Segment 13  
    Branch Segment 2
```

indica che il punto con coordinate (10.5359, -177.9185, -405.0381) è il punto di congiunzione tra i *Branch Segment 13* e *Branch Segment 2*.

In modo simile,

```
[Branch Set 1] Branch Segment 9:
```

```
Point (22.4512, -161.0416, -105.8606) connects from:  
    Branch Segment 6  
    Branch Segment 7
```

indica che il punto con coordinate (22.4512, -161.0416, -105.8606) è il punto di congiunzione tra il *Branch Segment 6* e il *Branch Segment 7*.

Particolari del file txt di Materialise da considerare

Dal file di testo generato da Materialise, si evince la seguente struttura dei dati:

```
[Main Branch Set] Branch Segment 1  
[Main Branch Set] Branch Segment 2  
[Main Branch Set] Branch Segment 3  
[Main Branch Set] Branch Segment 4  
[Main Branch Set] Branch Segment 5  
[Main Branch Set] Branch Segment 6  
[Main Branch Set] Branch Segment 7  
[Main Branch Set] Branch Segment 8
```

La ramificazione centrale, denominata Main Branch Set, è suddivisa in otto segmenti distinti, come evidenziato dalla presenza di otto Branch Segment numerati. Questi segmenti sono connessi seguendo la logica della ripetizione dell'ultimo punto di un Branch Segment come primo punto del Branch Segment successivo.

Per quanto riguarda le ramificazioni secondarie, queste sono organizzate in Branch Set separati, anch'essi numerati in modo univoco. Il numero dei Branch Set può variare, e alcuni numeri possono ripetersi, indicando la presenza di più ramificazioni con la stessa identificazione.

```
[Branch Set 1] Branch Segment 9  
[Branch Set 2] Branch Segment 10  
[Branch Set 2] Branch Segment 11  
[Branch Set 2] Branch Segment 12  
[Branch Set 3] Branch Segment 13
```

Questa struttura gerarchica dei dati sarà fondamentale per l'implementazione della logica di elaborazione in MATLAB, che ci permetterà di estrarre e formattare i dati necessari per la costruzione del modello 3D in Inventor.

4.1.2 Estrazione e formattazione dei dati con Matlab

Data la complessità e l'articolazione del file di testo generato da Materialise, è necessario procedere con una fase di formattazione ed esportazione dei dati rilevanti. Per questa fase è stato utilizzato come programma Matlab [22]. Inoltre, l'utilizzo di Matlab offre una notevole flessibilità nel caso di lavoro con diversi formati di centerline. Infatti, il codice Matlab può essere facilmente adattato per gestire cambiamenti nel formato del file, mantenendo invariata l'intera procedura di automazione. Questa adattabilità consente di utilizzare la stessa metodologia con dati provenienti da diverse fonti, non limitandosi esclusivamente a Materialise. Il codice MATLAB 8.1 implementato esegue le seguenti operazioni:

1. **Creazione della cartella *DataPoints*:** Se la cartella non esiste, viene creata per salvare i file Excel generati.
2. **Lettura del file di testo:** Il file di testo generato da Materialise viene letto riga per riga.
3. **Identificazione delle branch e assegnazione dei Marker:** Il codice identifica le sezioni *Main Branch Set* e *Branch Set* per determinare l'appartenenza dei punti alle diverse branch e assegna un valore univoco (Marker) a ciascuna branch.
4. **Estrazione dei dati:** Per ciascun punto, vengono estratti i valori di P_x , P_y , P_z e D_{fit} .
5. **Gestione dei punti ripetuti:** Viene implementata una logica per eliminare i punti ripetuti, che potrebbero causare errori nella costruzione del modello 3D.
6. **Gestione dei valori mancanti (NaN):** I valori mancanti di D_{fit} vengono sostituiti con la media dei valori adiacenti, per garantire la continuità dei dati.
7. **Creazione dei file Excel:** Per ciascuna branch, viene creato un file Excel contenente le coordinate P_x , P_y , P_z , il diametro D_{fit} e l'identificatore della branch (Marker). In particolare è stata applicata una riduzione di fattore 10 alle coordinate (P_x , P_y , P_z) e al diametro (D_{fit}) prima di salvare i dati nei file Excel. Questa riduzione è necessaria per compensare la conversione delle unità di misura che avviene in automatico in VBA.

8. **Creazione del file `marker_count.txt`:** Viene creato un file di testo contenente il numero totale di branch (Marker) presenti nel modello.

Comprendere il funzionamento generale del codice è essenziale, ma è altrettanto importante focalizzarsi su una sezione specifica, cruciale per l'elaborazione accurata dei dati e la successiva generazione del modello 3D in Inventor.

Analizziamo quindi nel dettaglio la logica implementata per la suddivisione delle branch e l'assegnazione dei marker, esaminando il seguente blocco di codice

```
1 % Gestione marker
2 if contains(line, [Main Branch Set])
3     currentMarker = 1;
4     lastBranchSetNumber = -1; %resetto il lastbranchsetnumber
   per il main branch
5 elseif contains(line, [Branch Set])
6     tokens = regexp(line, \[Branch Set (\d+)\], tokens);
7     if ~isempty(tokens)
8         branchSetNumber = str2double(tokens{1}{1});
9         if branchSetNumber ~= lastBranchSetNumber
10            currentMarker = branchSetNumber + 1;
11            lastBranchSetNumber = branchSetNumber;
12        end
13    end
14 end
```

Questo segmento di codice gioca un ruolo chiave nell'organizzazione dei dati, attribuendo un identificativo unico (marker) a ciascuna branch. La logica applicata è la seguente:

- **if contains(line,[Main Branch Set]):** Questa condizione controlla se la riga corrente letta dal file di testo include la stringa *[Main Branch Set]*. In caso affermativo, si tratta dell'inizio dei dati relativi alla branch principale. Di conseguenza, la variabile *currentMarker* è impostata a 1, assegnando il marker 1 alla branch principale. Inoltre, la variabile *lastBranchSetNumber* è reimpostata a -1. Questa variabile serve a memorizzare il numero del Branch Set precedente e viene usata per gestire correttamente l'assegnazione dei marker alle branch secondarie.
- **elseif contains(line, [Branch Set]):** Questa condizione verifica se la riga corrente contiene la stringa *[Branch Set]*, indicante l'inizio di una ramificazione. Se la condizione è soddisfatta, il numero del Branch Set è estratto tramite l'espressione regolare *[Branch Set (d+)]*. L'espressione cattura il numero tra parentesi quadre e lo converte in un valore numerico (*branchSetNumber*). Successivamente, si verifica se *branchSetNumber* è diverso da *lastBranchSetNumber*. Questo controllo previene l'assegnazione di altri marker a segmenti dello stesso branch, che possono ripetersi nel file di testo, come mostrato in 4.1.1. Se *branchSetNumber* è diverso da *lastBranchSetNumber*, un nuovo marker è assegnato alla branch secondaria, incrementando il valore di *currentMarker*, e *lastBranchSetNumber* è aggiornato con il nuovo valore. Se

branchSetNumber è uguale a lastBranchSetNumber, il marker corrente (*currentMarker*) rimane invariato, mantenendo l'identificativo assegnato alla branch corrente. In questo caso, *lastBranchSetNumber* non viene modificato.

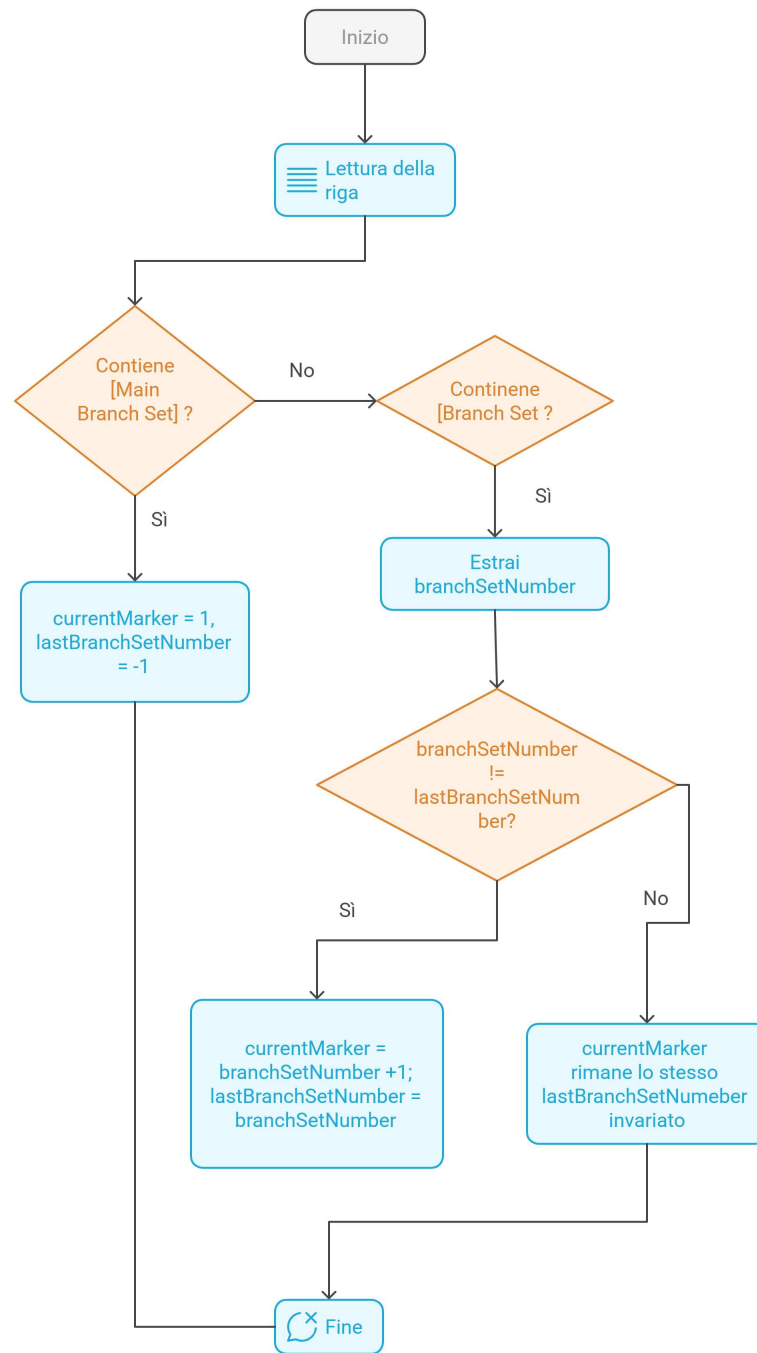


Figura 4.7: Diagramma di flusso della logica di elaborazione delle branch e assegnazione dei marker

Risultato dell'Algoritmo MATLAB

Una volta eseguito il codice MATLAB, il programma genera una cartella denominata *DataPoints*. Questa cartella contiene i risultati dell'elaborazione dei dati e include i seguenti file:

1. **marker_count.txt**: Questo file di testo contiene un singolo numero intero, che rappresenta il numero totale di marker identificati nel file di testo originale. In altre parole, questo numero indica quante ramificazioni distinte sono state individuate nel modello 3D. Nel nostro caso specifico, il file *marker_count.txt* conterrà il numero 11, indicando che il modello 3D è composto da 11 ramificazioni.
2. **File Excel branch_X.xlsx**: Per ciascun marker identificato, MATLAB genera un file Excel separato, denominato *branch_X.xlsx*, dove X è il numero del marker (da 1 a 11 nel nostro caso). Ogni file Excel contiene i dati relativi a una specifica ramificazione del modello 3D. Nel nostro caso specifico, verranno generati i seguenti file Excel: *branch_1.xlsx*, *branch_2.xlsx*, ..., *branch_11.xlsx*.

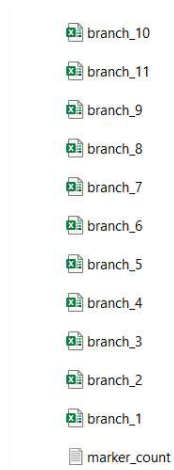


Figura 4.8: File generati all'interno della cartella DataPoints

A screenshot of an Excel spreadsheet titled 'branch_1 - Excel'. The spreadsheet displays a table with 5 columns (A-F) and 12 rows (22-35). The columns represent coordinates x, y, z, diameter Dfit, and marker E.

	A	B	C	D	E	F
22	0,42724	-18,0064	-37,0642	3,8671	1	
23	0,39327	-18,0129	-36,6117	3,97656	1	
24	0,37271	-17,974	-36,0343	4,05647	1	
25	0,37022	-17,8743	-35,4413	3,8691	1	
26	0,38099	-17,7079	-34,8468	3,32286	1	
27	0,39888	-17,4784	-34,2679	2,59225	1	
28	0,41699	-17,1979	-33,7193	2,22228	1	
29	0,42845	-16,8837	-33,2075	2,26208	1	
30	0,42741	-16,5552	-32,729	2,49958	1	
31	0,40995	-16,2298	-32,2712	2,59421	1	
32	0,37477	-15,9204	-31,8157	2,32527	1	
33	0,32348	-15,6339	-31,3442	2,07446	1	
34	0,2779	-15,4448	-30,9828	2,21652	1	
35	0,19144	-15,1315	-30,3084	2,73935	1	

Figura 4.9: Organizzazione del file Excel generato dal codice Matlab, Colonne A-C coordinate x y z, colonna D i diametri Dfit, colonna E Marker del branch

4.2 Implementazione del codice VBA

Dopo aver elaborato i dati in MATLAB e averli organizzati in file Excel, il passo successivo consiste nel creare il modello 3D in Inventor. Per automatizzare questo processo, è stato sviluppato un codice VBA che legge i dati dai file Excel e genera il modello 3D.

Il codice VBA è strutturato in una routine principale (*Congiunzione_Biforcazioni*) 8.2 che coordina l'esecuzione di due subroutine private:

LeggiDatiExcel e ***CreaSolidoPerMarker***.

La routine principale si occupa di leggere il numero di marker dal file *marker_count.txt* e di ciclare attraverso i file excel *branch_X.xlsx* per generare il modello 3D di ogni ramificazione.

4.2.1 Routine principale Congiunzione_Biforcazioni

Il processo ha inizio con la dichiarazione e l'inizializzazione delle variabili necessarie per l'esecuzione del codice. Queste variabili includono oggetti Inventor, come il documento parte attivo e la definizione del componente parte, nonché variabili per memorizzare i dati estratti dai file Excel e per controllare il flusso del codice. Successivamente, la routine principale si occupa di leggere il numero totale di marcatori (marker) dal file *marker_count.txt*.

Questo file, generato in precedenza dal codice MATLAB, contiene un singolo valore numerico che indica il numero di ramificazioni presenti nel modello 3D. La lettura di questo valore è cruciale, in quanto determina il numero di iterazioni del ciclo principale.

Il cuore della routine è un ciclo For che itera attraverso i marcatori, da 1 al numero totale di marcatori.

Un aspetto importante di questa routine è la gestione degli errori. Se il file *marker_count.txt* non viene trovato, la routine visualizza un messaggio di errore e termina l'esecuzione. Inoltre, la routine include controlli per verificare l'esistenza dei file Excel corrispondenti a ciascun marcatore, visualizzando un messaggio di errore se un file non viene trovato.

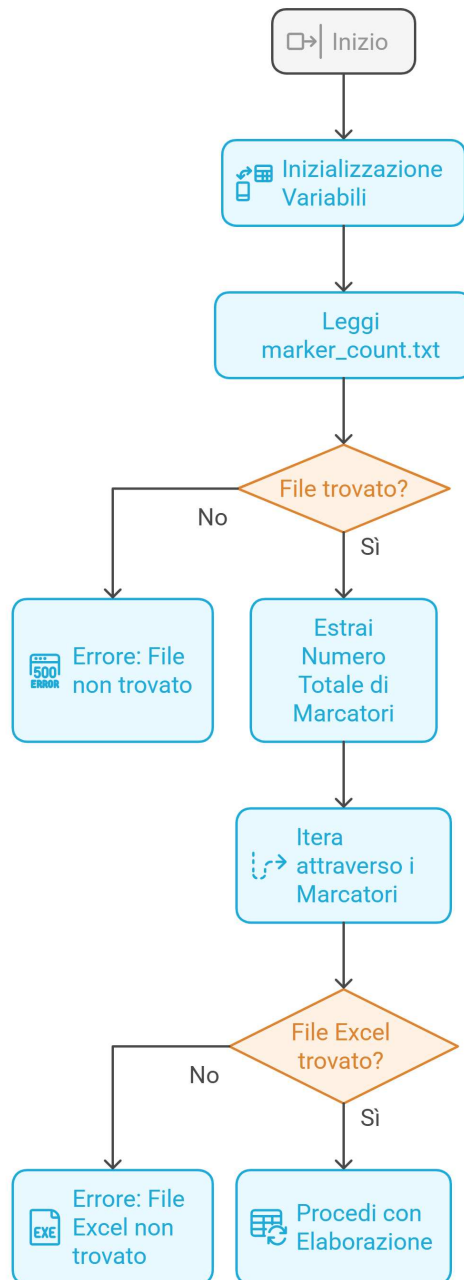


Figura 4.10: Diagramma di flusso della subroutine principale VBA, che illustra il ciclo di lettura dei file Excel basato sui marcatori.

4.2.2 Subroutine *LeggiDatiExcel*

Questa subroutine è progettata per estrarre dati numerici specifici da un file Excel, preparandoli per l'utilizzo in un contesto di modellazione 3D in Inventor tramite VBA. La sua funzione principale è quella di leggere le coordinate (x, y, z), i diametri (dfit) e i marcatori (marker) da un file Excel e memorizzarli in array, che vengono poi passati per riferimento alla routine chiamante.

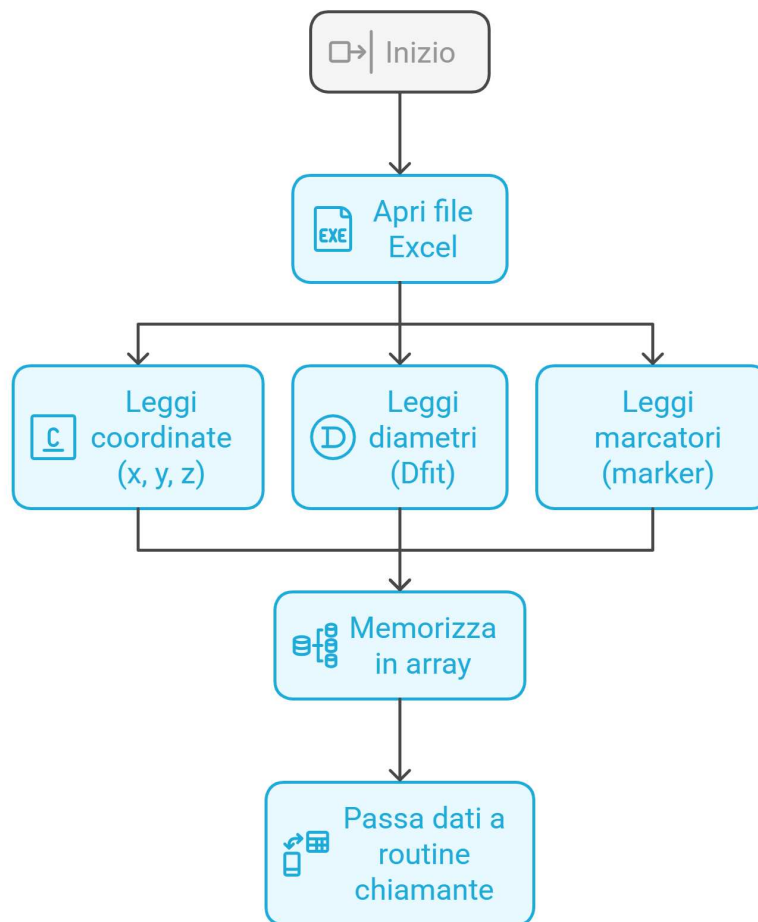


Figura 4.11: Diagramma di flusso della subroutine VBA per l'importazione di coordinate, diametri e marcatori da file Excel

Di seguito si analizzano alcune parti importanti.

```
1 Set oExcelApp = CreateObject("Excel.Application")
```

Questa linea crea un'istanza dell'applicazione Excel.

```
1 oExcelApp.Visible = False
```

Qui si imposta la visibilità dell'applicazione Excel a `False`, il che significa che l'utente non vedrà l'interfaccia di Excel durante l'esecuzione del codice. Questo è utile per eseguire il codice in background senza interrompere il flusso di lavoro dell'utente.

Apertura del file Excel

```
1 Set oWorkbook = oExcelApp.Workbooks.Open(ExcelFilePath)
```

Apri il file Excel specificato nel percorso `ExcelFilePath`. È importante che il percorso del file sia corretto, altrimenti il codice genererà un errore.

```
1 Set oWorksheet = oWorkbook.Sheets(1)
```

Impostazione del primo foglio di lavoro del file Excel come foglio di lavoro attivo.

Lettura dei dati

```
1 RowLimit = oWorksheet.Cells(oWorksheet.Rows.Count, 1).End(xlUp).Row
```

Questa linea determina l'ultima riga contenente dati nella prima colonna del foglio di lavoro. Questo viene fatto spostandosi dalla fine della colonna verso l'alto fino a trovare la prima cella non vuota.

```
1 ReDim x(1 To RowLimit, 1 To 1)
```

Ridimensionamento dell'array `x` per contenere i dati letti dal file Excel. Il numero di righe dell'array è determinato da `RowLimit`, mentre il numero di colonne è 1. Gli array `y`, `z`, `dfit` e `marker` vengono ridimensionati in modo simile.

```
1 x(GenericCounter, 1) = oWorksheet.Cells(GenericCounter, 1).Value2
```

Legge il valore della cella nella riga `GenericCounter` e nella prima colonna del foglio di lavoro e lo memorizza nell'array `x`. La proprietà `Value2` viene utilizzata per leggere il valore della cella come tipo di dati appropriato.

```
1 CDbl(x(GenericCounter, 1))
```

Conversione esplicita del valore nell'array `x` in un tipo di dati `Double`. Questo è importante per garantire che i dati numerici vengano elaborati correttamente.

Chiusura e rilascio degli oggetti Excel

```
1 oWorkbook.Close False
```

Chiusura del file Excel senza salvare le modifiche.

```
1 oExcelApp.Quit
```

Chiusura dell'applicazione Excel.

```
1 Set oWorksheet = Nothing
```

Questa linea rilascia l'oggetto `oWorksheet` dalla memoria. Questo è importante per liberare risorse di sistema. Gli oggetti `oWorkbook` e `oExcelApp` vengono rilasciati in modo simile.

Gestione degli errori

```
1 On Error GoTo ErrHandler
```

```
2 MsgBox("Unexpected error while reading excel file: " & Err.  
Description, vbOKOnly, "Excel File Error")
```

Gestione nel caso di errori in particolare si visualizzerà una finestra di messaggio di errore che descrive l'errore che si è verificato. Le linee successive nel gestore di errori rilasciano gli oggetti Excel per evitare perdite di memoria.

4.2.3 Subroutine CreaSolidoPerMarker

La subroutine *CreaSolidoPerMarker* è il fulcro della creazione del modello 3D in Inventor, responsabile della trasformazione dei dati numerici estratti dai file Excel in entità geometriche tridimensionali. Il suo obiettivo principale è quello di generare una ramificazione alla volta del modello 3D, utilizzando le coordinate, i diametri e i marcatori forniti come input.

Il processo inizia con la creazione di un nuovo schizzo 3D nel documento parte attivo di Inventor.

Questo schizzo 3D funge da spazio di lavoro tridimensionale in cui verranno create le entità geometriche che definiscono la ramificazione.

I punti di dati vengono filtrati in base al marcatore (MarkerIndex) fornito come input. Questo passaggio è cruciale per garantire che vengano considerati solo i punti appartenenti alla ramificazione corrente. I punti filtrati vengono quindi utilizzati per creare punti di riferimento (WorkPoints) all'interno dello schizzo 3D.

Una volta creati i punti di riferimento, la subroutine procede con la creazione di linee che collegano i punti in sequenza. Queste linee definiscono il percorso centrale della ramificazione.

Per ciascuna linea, la subroutine crea un piano di lavoro (WorkPlane) perpendicolare alla linea e passante per il punto di riferimento iniziale. Questo piano di lavoro funge da superficie su cui verrà creato il profilo 2D della ramificazione.

Successivamente, viene creato uno schizzo 2D sul piano di lavoro, proiettando il punto di riferimento iniziale sullo schizzo. Questo punto proiettato viene quindi utilizzato come centro per la creazione di un cerchio, il cui raggio è determinato dal diametro (dfit) corrispondente al punto di riferimento.

Il cerchio creato rappresenta il profilo 2D della ramificazione in quel punto. Questo profilo viene quindi aggiunto a una collezione di profili, che verrà utilizzata in seguito per creare un loft.

Una volta creati tutti i profili 2D, si dovrebbe passare alla creazione 3D della ramificazione.

Come discusso in 2.4, sono disponibili due funzioni per la creazione del modello 3D: in Inventor: Sweep o Loft e in VBA: *SweepFeatures.Add* e *LoftFeatures.Add*.

Inizialmente, è stata considerata l'opzione di utilizzare la funzione *SweepFeatures.Add*, che consente di creare un solido 3D estrudendo un profilo lungo un percorso. Tuttavia, questa funzione si è rivelata inadeguata per il questo caso d'uso, in quanto non permette di gestire variazioni di diametro lungo il percorso di estrusione.

I dati estratti dai file Excel generati da MATLAB indicano che il diametro (Dfit) delle ramificazioni non è costante, ma varia lungo il percorso. In particolare, il diametro può variare in modo significativo tra punti consecutivi, rendendo impossibile l'utilizzo della funzione *SweepFeatures.Add*.

La funzione *LoftFeatures.Add*, invece, consente di creare un solido 3D collegando una serie di profili 2D, quindi il loft viene creato collegando i profili 2D in sequenza, creando una forma tridimensionale che segue il percorso definito dalle linee.

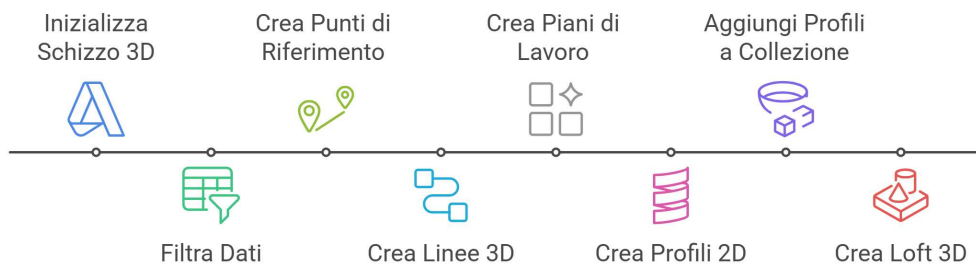


Figura 4.12: Rappresentazione grafica del flusso di lavoro del codice VBA, che automatizza la generazione del modello 3D a partire dai dati di input.

Creazione dello schizzo 3D

```
1 Set oSketch3D = oPartComponentDefinition.Sketches3D.Add
```

Crea un nuovo schizzo 3D nel documento parte attivo. Gli schizzi 3D vengono utilizzati per creare geometrie 3D complesse.

Filtraggio dei punti

```
1 Set FilteredWorkPoints = New Collection
```

Questa linea crea una nuova collezione che verrà utilizzata per memorizzare i punti di riferimento filtrati. Le collezioni sono utili per memorizzare insiemi di oggetti.

```
1 If marker(j, 1) = MarkerIndex Then
```

Controllo: se il valore nell'array `marker` per la riga corrente è uguale al `MarkerIndex` specificato. Questo viene utilizzato per filtrare i punti in base al `marker`.

```
1 Set oWorkPoint = oPartComponentDefinition.WorkPoints.AddFixed  
  (ThisApplication.TransientGeometry.CreatePoint(CDb1(x(j,  
  1)), CDb1(y(j, 1)), CDb1(z(j, 1))))
```

Questa linea crea un nuovo punto di riferimento in Inventor utilizzando le coordinate lette dal file Excel. La funzione `CDb1` viene utilizzata per convertire i valori delle coordinate in numeri a doppia precisione.

```
1 FilteredWorkPoints.Add oWorkPoint
```

Questa linea aggiunge il punto di riferimento alla collezione `FilteredWorkPoints`.

Creazione delle linee

```
1 Set oSketchLine3D = oSketch3D.SketchLines3D.AddByTwoPoints(  
  FilteredWorkPoints.Item(GenericCounter),  
  FilteredWorkPoints.Item(GenericCounter + 1), False, 1)
```

Crea una nuova linea 3D che collega due punti di riferimento consecutivi nella collezione `FilteredWorkPoints`.

Creazione dei profili

```
1 Set oWorkPlane = oPartComponentDefinition.WorkPlanes.  
  AddByNormalToCurve(oSketchLine3D, FilteredWorkPoints.Item(  
  GenericCounter))
```

Creazione del nuovo piano di lavoro perpendicolare alla linea 3D e passante per il punto di riferimento corrente.

```
1 oWorkPlane.Visible = False
```

Questa linea imposta la visibilità del piano di lavoro a **False**. Questo è utile per nascondere i piani di lavoro che non sono necessari per la visualizzazione del modello 3D.

```
1 Set oSketch = oPartComponentDefinition.Sketches.Add(  
    oWorkPlane)
```

Preparazione di un nuovo schizzo 2D sul piano di lavoro.

```
1 Set oSketchPoint = oSketch.AddByProjectingEntity(  
    FilteredWorkPoints.Item(GenericCounter))
```

Questa linea proietta il punto di riferimento corrente sullo schizzo 2D.

```
1 Set oSketchCircle = oSketch.SketchCircles.AddByCenterRadius(  
    oSketchPoint, CDb1(dfit(GenericCounter, 1)) / 2)
```

Questa linea crea un cerchio nello schizzo 2D con centro nel punto proiettato e raggio pari alla metà del diametro letto dal file Excel.

```
1 Set oProfile = oSketch.Profiles.AddForSolid
```

Creazione di un profilo dal cerchio. I profili sono utilizzati per creare solidi 3D.

```
1 oProfileCollection.Add(oProfile)
```

Questa linea aggiunge tutti i profili creati a una collection.

Creazione del loft

```
1 Set oLoftDefinition = oPartComponentDefinition.Features.  
    LoftFeatures.CreateLoftDefinition(oProfileCollection,  
    kNewBodyOperation)
```

Questa linea crea la loft definition usando la collection di profili creata in precedenza. In particolare l'uso della funzione *kNewBodyOperation* permette oltre a generare il loft anche di definirlo come singolo solido, per ogni branch si avrà la creazione di un solido indipendente. Questo permette di avere un disegno 3D dinamico in cui si può scegliere quale ramificazione visualizzare.

```
1 oPartComponentDefinition.Features.LoftFeatures.Add(  
    oLoftDefinition)
```

Questa linea aggiunge la loft feature al modello 3D.

Definizione delle variabili

Le variabili vengono dichiarate con tipi di dati specifici per garantire che i dati vengano elaborati correttamente. Gli oggetti Inventor vengono dichiarati come oggetti specifici (ad esempio, PartDocument, PartComponentDefinition, Sketch3D) per consentire l'accesso alle loro proprietà e metodi.

Letture del numero di marker

```
1 Set FSO = CreateObject("Scripting.FileSystemObject")
```

Crea un'istanza dell'oggetto FileSystemObject, che viene utilizzato per accedere ai file e alle cartelle del file system.

```
1 If FSO.FileExists(FolderPath & MarkerCountFile) Then
```

Controlla se il file marker_count.txt esiste nella cartella specificata.

```
1 Open FolderPath & MarkerCountFile For Input As #1
```

Apri il file marker_count.txt per la lettura.

Chiamata alle subroutine.

```
1 ' Legge i dati dal file Excel
2     Call LeggiDatiExcel(ExcelFilePath, x, y, z, dfit,
3         marker)
4 ' Crea un solido per il marker
5     Call CreaSolidoPerMarker(oPartDoc,
6         oPartComponentDefinition, oWorkPoints, x, y, z, dfit,
7         marker, i)
```

Capitolo 5

Risultati

Questo capitolo fornisce una rappresentazione visiva del processo di creazione del modello 3D in Inventor, illustrando i passaggi chiave descritti nei capitoli precedenti. Le immagini mostrano l'interfaccia di Inventor e il modello 3D in diverse fasi di sviluppo, offrendo una visione d'insieme del flusso di lavoro.

5.0.1 Creazione dei punti di riferimento:



Figura 5.1: La subroutine CreaSolidoPerMarker crea punti di riferimento per marker, utilizza le coordinate lette dal file Excel branch_1

5.0.2 Creazione delle linee e Sketch circonferenze:

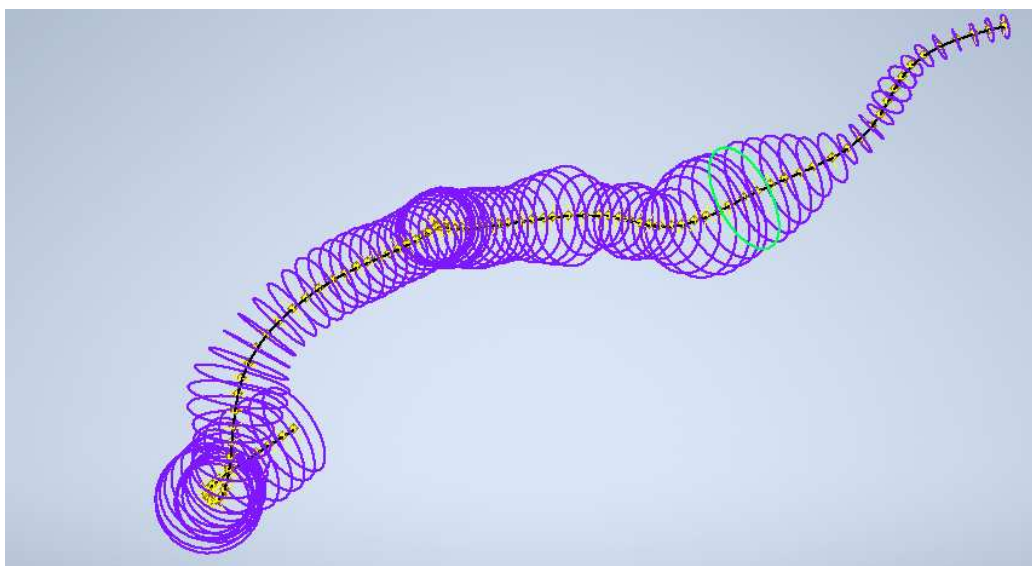


Figura 5.2: Le linee vengono create per definire l'ase del lume della ramificazione e contemporaneamente vengono generati i piani di lavoro normali alla linea per disegnare la circonferenza ad ogni punto.

5.0.3 Creazione del loft:

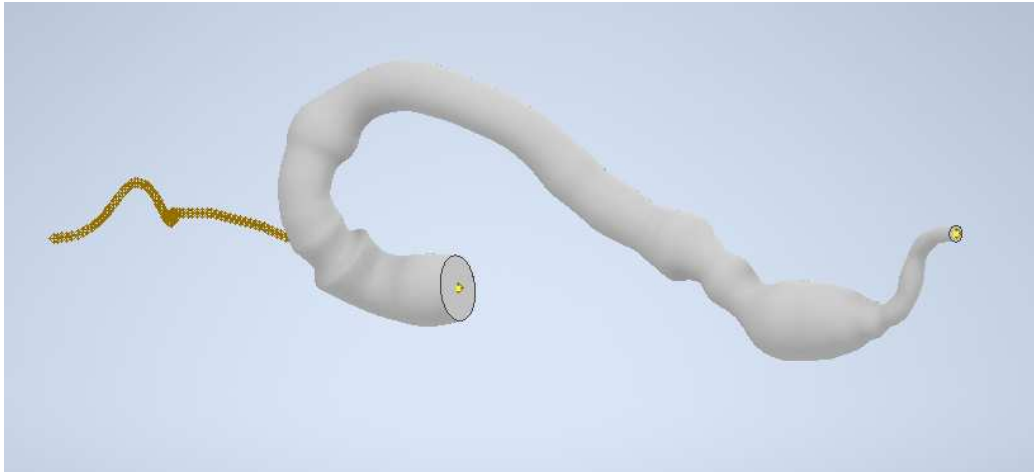


Figura 5.3: La funzione `LoftFeatures.Add` viene utilizzata per creare il solido 3D della ramificazione, collegando i profili 2D. In oltre possiamo vedere che il ciclo è andato avanti nella lettura del file `branch_2` con la generazione dei punti di lavoro e lo sketch di linee e circonferenze.



Figura 5.4: Loft completi di diverse ramificazioni.

5.0.4 Modello 3D completo:

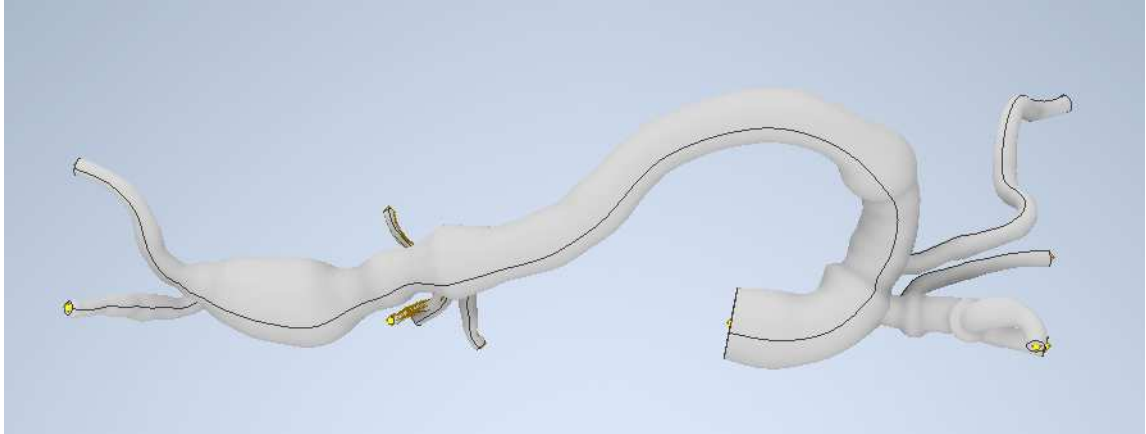


Figura 5.5: Il codice VBA genera il modello 3D completo del vaso sanguigno, combinando i risultati delle subroutine.

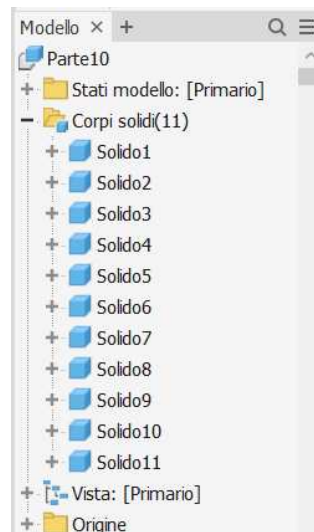


Figura 5.6: Nel Browser Modello sono visibili i diversi solidi generati dal codice VBA

5.0.5 Validazione del modello attraverso confronto di geometria

Al fine di quantificare le discrepanze geometriche tra i modelli, procediamo con un'analisi comparativa in *Netfabb*[23] un potente software sviluppato da Autodesk, specificamente progettato per ottimizzare i flussi di lavoro nella produzione additiva. Questo strumento consente agli utenti di importare, riparare, analizzare e preparare modelli 3D per la stampa, garantendo risultati di alta qualità e riducendo al minimo gli errori.

Tra le sue numerose funzionalità, *Netfabb* offre la possibilità di confrontare modelli 3D in diversi formati, come STL e CAD, per identificare eventuali discrepanze geometriche e ottimizzare i modelli per la stampa.

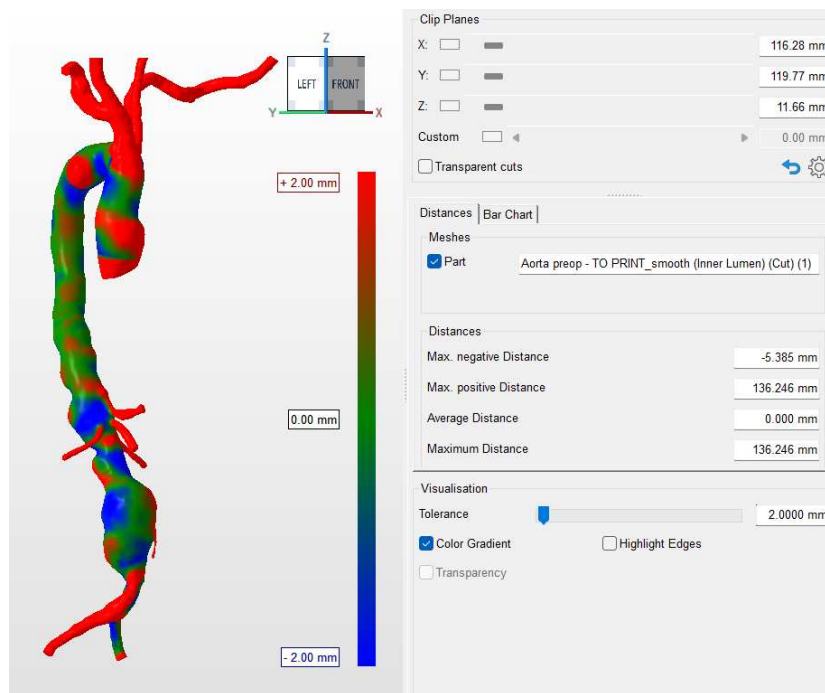


Figura 5.7: Confronto tra i due modelli allineati.

In 5.7 possiamo osservare il risultato di un'analisi comparativa dettagliata tra il modello STL originale e il modello CAD che ho generato.

La scala cromatica laterale ci fornisce una chiara indicazione dell'entità delle deviazioni, con valori numerici che vanno da -2.00 mm a $+2.00\text{ mm}$. Questo significa che le aree colorate in rosso indicano zone in cui il modello STL si discosta positivamente dal CAD fino a 2.00 mm , mentre le aree blu mostrano deviazioni negative

La tabella a destra dell'immagine ci fornisce ulteriori dati numerici cruciali. Ad esempio, la massima deviazione negativa riscontrata è di -5.385 mm, mentre la massima deviazione positiva è di 136.246 mm. Questi valori ci permettono di quantificare con precisione le differenze tra i modelli e di identificare le aree critiche in cui le deviazioni sono più significative. Tuttavia, è importante notare che il valore di massima deviazione positiva, 136.246 mm, è estremamente elevato e molto probabilmente dovuto alla presenza di un piccolo triangolo isolato all'interno del file STL, situato lontano dalla geometria principale del vaso sanguigno. Questo triangolo, pur non influenzando significativamente la geometria complessiva del modello, ha un impatto notevole sul calcolo della massima deviazione.

Osservando la superficie del modello, possiamo notare che le aree con colori più intensi, come il rosso e il blu, corrispondono alle zone in cui la geometria dei due modelli differisce maggiormente. Questo ci permette di identificare visivamente le aree in cui la mesh STL potrebbe necessitare di ottimizzazione o di correzioni.

Inoltre, la presenza della tolleranza di 2.00 mm ci consente di valutare se le deviazioni rientrano nei limiti accettabili. Se le deviazioni superano questa tolleranza, come nel caso della massima deviazione positiva di 136.246 mm, ciò indica che è necessario intervenire per correggere i modelli.

La tolleranza di $\pm 2.00\text{ mm}$ è un elemento cruciale e non arbitrario, scelto per una ragione ben precisa: riteniamo che un errore di questa entità sia accettabile, soprattutto in previsione di una possibile *simulazione agli elementi finiti (FEM)*, tecnica di analisi numerica utilizzata per prevedere il comportamento di un oggetto o di una struttura sottoposta a carichi o sollecitazioni. In pratica, la FEM suddivide l'oggetto in un numero elevato di elementi discreti, collegati tra loro da nodi. Successivamente, vengono applicate le equazioni matematiche che descrivono il comportamento del materiale e le condizioni al contorno (carichi, vincoli, ecc.). Il software FEM, ad esempio potrebbe essere utilizzato *Ansys*[24][10] che risolve queste equazioni per ciascun elemento, fornendo una stima del comportamento dell'intera struttura

Capitolo 6

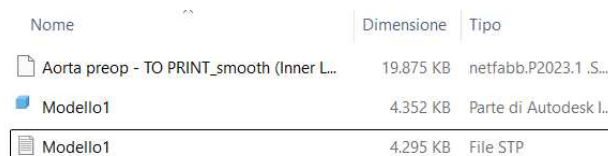
Discussione

6.1 Riepilogo dei risultati principali

Il presente lavoro ha dimostrato la fattibilità di un approccio automatizzato per la creazione di modelli 3D patient-specific di vasi sanguigni a partire da dati di imaging medico.

L'elaborazione dei dati in MATLAB ha permesso di estrarre informazioni cruciali sulla geometria delle ramificazioni, mentre il codice VBA in Inventor ha automatizzato la costruzione del modello 3D. I risultati ottenuti, in particolare la creazione di un modello 3D accurato e dettagliato, confermano la validità del metodo proposto. La capacità di gestire variazioni di diametro lungo le ramificazioni, ottenuta tramite l'utilizzo della funzione *LoftFeatures.Add*, rappresenta un punto di forza significativo del nostro approccio patient-specific.

L'ulteriore analisi condotta con Netfabb ha permesso di quantificare le deviazioni geometriche tra il modello CAD generato e l'STL originale. I risultati hanno evidenziato deviazioni maggiori soprattutto nelle ramificazioni, sottolineando l'importanza di considerare queste discrepanze nelle successive fasi di analisi.



Nome	Dimensione	Tipo
Aorta preop - TO PRINT_smooth (Inner L...	19.875 KB	netfabb.P2023.1 .S...
Modello1	4.352 KB	Parte di Autodesk I...
Modello1	4.295 KB	File STP

Figura 6.1: Dimensioni dei diversi file, in ordine: STL, CAD, STEP.

Inoltre come discusso in 2.5.1 e come si evince dall'immagine 6.1, possiamo vedere che il file STL è decisamente più pesante sia del CAD (modello1 .ipt)

che del file STEP generato dal CAD. Questo significa che, nonostante il formato STL sia una rappresentazione semplificata della geometria 3D, priva della ricchezza di informazioni parametriche del CAD, risulta paradossalmente più pesante di circa 15 MB. Questa differenza sostanziale di dimensioni è un aspetto cruciale da considerare.

6.2 Valutazione dell'efficacia del metodo

Il metodo proposto si è dimostrato efficace nel trasformare dati di imaging medico in modelli 3D patient-specific di vasi sanguigni. L'approccio automatizzato riduce significativamente il tempo e lo sforzo necessari per la modellazione 3D, rispetto a metodi manuali. In particolare, l'intero processo di automazione richiede circa 2 minuti e 30 secondi. Tuttavia, la precisione del modello 3D dipende dalla qualità dei dati di input, evidenziando l'importanza di dati patient-specific di alta qualità. Eventuali errori o artefatti nei dati di imaging possono influire sulla precisione del modello 3D. L'efficacia del metodo è stata ulteriormente confermata dall'analisi comparativa in Netfabb, che ha permesso di quantificare le deviazioni geometriche tra il modello CAD e l'STL. Questo passaggio ha evidenziato l'importanza di un controllo di qualità accurato, soprattutto in applicazioni mediche dove la precisione è fondamentale. La capacità di Netfabb di visualizzare e quantificare le deviazioni ha fornito informazioni preziose per valutare l'affidabilità del modello generato.

In futuro, sarebbe utile esplorare tecniche di pre-elaborazione dei dati per migliorare la qualità dei dati di input patient-specific e utilizzare la FEM per simulare il comportamento del vaso sanguigno sotto pressione o flusso sanguigno. Questo ci permetterebbe di:

- Valutare le tensioni e le deformazioni all'interno del vaso.
- Prevedere il rischio di rottura o di formazione di aneurismi.
- Ottimizzare la geometria del vaso per migliorare la sua resistenza.

6.3 Analisi delle scelte progettuali

La creazione di un modello 3D patient-specific di vasi sanguigni a partire da dati di imaging medico richiede una serie di scelte progettuali che influenzano significativamente il risultato finale. In questo lavoro, è stato adottato un approccio basato sull'utilizzo del software Inventor e del linguaggio VBA, combinato con l'elaborazione dei dati in MATLAB.

Una delle scelte progettuali più critiche è stata la selezione delle linee spezzate per collegare i punti di riferimento, al posto delle spline. La spline, pur offrendo una rappresentazione più accurata delle curve, presenta una complessità analitica elevata, specialmente in ambiente VBA. La definizione e la manipolazione di spline in VBA richiederebbero un notevole sforzo computazionale e di sviluppo. Abbiamo quindi optato per l'utilizzo di linee spezzate, che collegano i punti di riferimento tramite segmenti lineari. Questa scelta è stata motivata dalla densità dei punti di riferimento, che, essendo molto vicini tra loro, consentono di approssimare con precisione la forma delle ramificazioni utilizzando segmenti lineari. Inoltre, la creazione di linee spezzate in VBA è semplice e diretta, utilizzando la funzione *Sketch3D.CreateLine* per collegare due punti consecutivi.

Un'altra scelta progettuale significativa è stata la selezione della funzione *LoftFeatures.Add* per la creazione del solido 3D, in contrapposizione alla funzione *SweepFeatures.Add*. L'estrusione tramite sweep sarebbe stata una scelta valida se i diametri delle ramificazioni fossero stati costanti. Tuttavia, i dati di input mostrano variazioni significative nei diametri, rendendo il loft la soluzione più appropriata. Il loft consente di collegare profili 2D di dimensioni variabili lungo il decorso del vaso, adattandosi perfettamente alla geometria complessa dei vasi sanguigni e alle variazioni di diametro.

La scelta di gestire i punti di riferimento e le ramificazioni utilizzando marcatori (marker) e file Excel separati è stata un'altra decisione progettuale importante. L'utilizzo di marcatori consente di definire un ordine gerarchico tra le ramificazioni, facilitando l'organizzazione e la gestione dei dati. La suddivisione delle informazioni in file Excel separati, uno per ciascuna ramificazione, consente di modificare i dati in modo indipendente, senza dover modificare il file di input originale fornito da Materialise. Questo approccio modulare facilita l'editing dei dati, anche per utenti che non hanno familiarità con la struttura del file di input originale.

6.4 Discussione delle limitazioni

Il presente lavoro, pur avendo dimostrato la fattibilità di un approccio automatizzato per la creazione di modelli 3D patient-specific, presenta alcune limitazioni che meritano di essere discusse. In primo luogo, la qualità del modello 3D patient-specific dipende fortemente dalla qualità dei dati di input forniti da MATLAB. Durante il processo di modellazione, abbiamo riscontrato diverse lacune nei dati forniti dal file di Materialise.

Ad esempio, alcuni valori di diametro (Dfit) erano mancanti (NaN), richiedendo l'implementazione di un ciclo di controllo in MATLAB per la loro gestione. Questo ha introdotto un elemento di approssimazione, potenzialmente riducendo il livello di specificità del modello.

```
Dfit
23.0121
n/a
n/a
n/a
n/a
49.0995
47.6697
49.7798
42.8374
8.8224
8.3675
8.2955
```

Figura 6.2: Mancanza dei dati Dfit, nel file txt di Materialise

Inoltre, come si può osservare nella 6.2, alcuni valori di diametro, in particolare quelli in prossimità delle biforcazioni, presentavano valori eccessivamente grandi rispetto all'andamento della ramificazione.

Questo è dovuto alla metodologia di iscrizione della sfera utilizzata da Materialise, che tende a sovrastimare i diametri in regioni di curvatura elevata, come le biforcazioni, come discusso in 4.1.1. La presenza di queste circonferenze causa intersezioni indesiderate tra i piani di lavoro, rendendo impossibile la creazione del loft e compromettendo l'integrità del modello 3D.

Per risolvere questo problema, abbiamo dovuto sostituire manualmente questi valori anomali con diametri accettabili, introducendo un ulteriore elemento di soggettività.

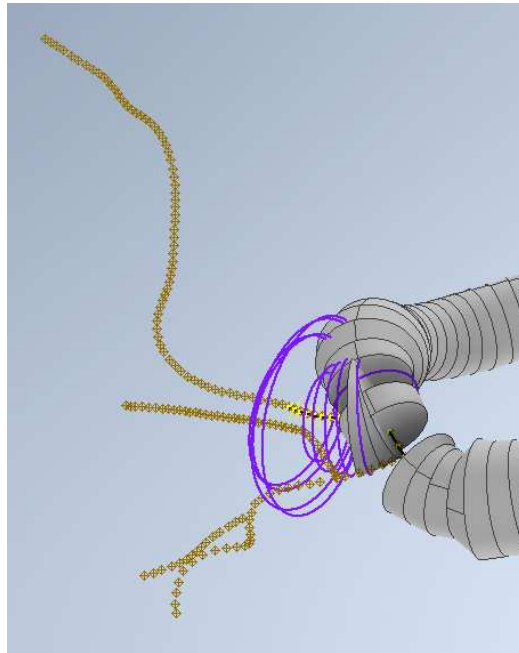


Figura 6.3: Sovrapposizione dei profili 2D (circonferenze viola) che impedisce la creazione del loft

Un'ulteriore limitazione emersa durante il processo di modellazione riguarda la mancanza di una definizione univoca nell'ordine di presentazione delle ramificazioni (branch) all'interno del file di input.

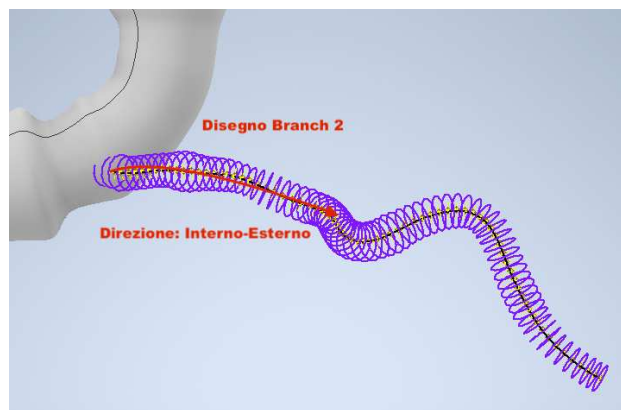


Figura 6.4: La mancanza di una definizione univoca dei punti: Branch 2 definito con andamento dall'interno verso esterno.

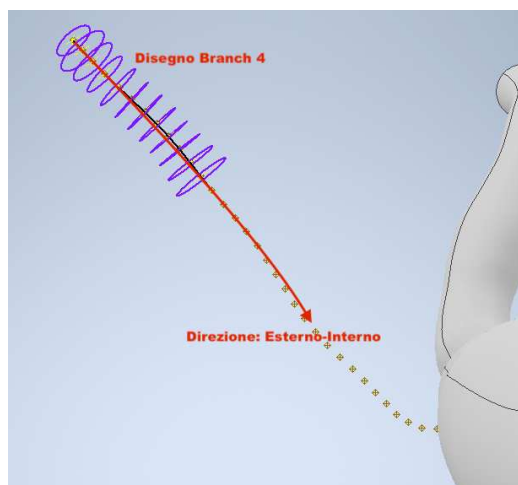


Figura 6.5: La mancanza di una definizione univoca dei punti: Branch 4 definito con andamento dall'esterno verso l'interno.

Questa incoerenza si manifesta nella direzione di definizione dei punti: alcune ramificazioni sono definite dall'esterno verso l'interno, mentre altre seguono la direzione opposta.

Questa ambiguità ha un impatto significativo sulla capacità di automatizzare completamente il processo di modellazione. Ad esempio, tornando al problema dei diametri (Dfit) anomali che causavano errori nella creazione del loft, l'assenza di un 'verso unico' nella definizione dei punti impedisce l'implementazione di cicli di controllo automatici per identificare e correggere tali anomalie. Non è possibile stabilire a priori se i punti anomali si trovino all'inizio o alla fine della sequenza di punti, poiché l'ordine può variare tra le diverse ramificazioni.

Di conseguenza, è stato necessario ricorrere a un'analisi manuale e a una mappatura preliminare dei punti nell'ambiente di lavoro di Inventor, al fine di identificare e correggere i diametri anomali. Questo ha aumentato la complessità del processo e ha limitato la possibilità di creare un flusso di lavoro completamente automatizzato e indipendente dall'intervento umano.

Queste limitazioni evidenziano l'importanza di dati di input di alta qualità e coerenti per la creazione di modelli 3D patient-specific accurati. In futuro, sarebbe utile esplorare tecniche di pre-elaborazione dei dati più avanzate per mitigare questi problemi e ridurre la necessità di interventi manuali.

Capitolo 7

Conclusione

In conclusione, questa tesi ha sviluppato una metodologia integrata per la creazione di modelli 3D patient-specific di reti vascolari, automatizzando il processo tramite VBA e gestendo le variazioni di diametro con loft.

Questo approccio ha dimostrato efficacia nel ridurre i tempi di progettazione e migliorare la precisione, cruciale per modelli complessi da immagini mediche.

Inoltre, l'utilizzo di un file *.ipt* che può essere poi convertito in formato STEP, rispetto all'STL si rivela vantaggioso, grazie alla sua minore dimensione del file e alla possibilità di essere gestito e modificato direttamente in un ambiente CAD.

La ricerca in letteratura ha evidenziato l'importanza di metodologie complementari per superare le limitazioni delle tecniche di segmentazione automatica. In particolare, Kretschmer et al. [13] propongono un metodo basato su superfici di sweep implicite, che consente di generare e correggere modelli lisci di alta qualità in modo intuitivo. Un approccio ibrido, che integri la nostra metodologia con quella di Kretschmer et al., potrebbe ottimizzare ulteriormente la modellazione vascolare, rendendola più accessibile alla pratica clinica quotidiana.

I modelli 3D patient-specific sviluppati trovano applicazioni cliniche nella pianificazione chirurgica e nella progettazione di stent personalizzati. L'integrazione con simulazioni del flusso sanguigno, come descritto da Taylor et al. [14], permetterebbe il calcolo di parametri emodinamici cruciali, supportando diagnosi e gestione di malattie cardiache.

Per quanto riguarda gli sviluppi futuri, un'analisi FEM dettagliata si rivela fondamentale, in quanto permetterebbe di ottenere informazioni più approfondite sulla geometria del vaso sanguigno, ottimizzandone la struttura e valutando potenziali rischi di rottura o di formazione di aneurismi.

La traduzione di queste potenzialità in applicazioni cliniche concrete richiede collaborazione interdisciplinare e validazione clinica. In futuro, l'intelligenza artificiale potrebbe automatizzare ulteriormente la modellazione e migliorare la precisione delle simulazioni, aprendo nuove frontiere nella medicina personalizzata.

Capitolo 8

Appendice: Codici completi Matlab e VBA

8.1 Codice Matlab:

```
1
2 %% Creazione di una cartella con dentro:
3 % file txt con il numero totale di markers
4 % file excel per marker denominati come:
   branch_numeroMarker
5 % all'interno dei file excel troviamo Px Py
   Pz Dfit Marker
6 clc
7 close all
8 % Creazione della cartella DataPoints se non
   esiste gia
9 if ~exist(DataPoints, 'dir')
10     mkdir(DataPoints);
11 end
12 fileID = fopen('n_a_Centerline.txt', 'r');
13 if fileID == -1
14     error('File not found or could not be
   opened.');
```

```

16 allPx = [];
17 allPy = [];
18 allPz = [];
19 allDfit = [];
20 allMarker = [];
21 puntiUnici = containers.Map();
22 indicePunti = 1;
23 currentMarker = 1;
24 lastBranchSetNumber = -1; % Inizializza con
    un valore che non corrisponde a nessun
    Branch Set
25 while true
26     line = fgetl(fileID);
27     if ~ischar(line)
28         break;
29     end
30     % Gestione marker
31     if contains(line, [Main Branch Set])
32         currentMarker = 1;
33         lastBranchSetNumber = -1; %resetto
    il lastbranchsetnumber per il main branch
34     elseif contains(line, [Branch Set])
35         tokens = regexp(line, \[Branch Set
    (\d+)\], tokens);
36         if ~isempty(tokens)
37             branchSetNumber = str2double(
    tokens{1}{1});
38             if branchSetNumber ~=
    lastBranchSetNumber
39                 currentMarker =
    branchSetNumber + 1;
40                 lastBranchSetNumber =
    branchSetNumber;

```

```

41         end
42     end
43 end
44
45     if contains(line, Px) || contains(line,
Point ()
46         if contains(line, Point ()
47             fgetl(fileID);
48             fgetl(fileID);
49         end
50         formatSpec = %f %f %f %*f %*f %*f %*
f %*f %*f %*f %*f %*f %f %*f %*f %*f %*f
%*f %*f %*f;
51         while true
52             dataLine = fgetl(fileID);
53             if ~ischar(dataLine) || isempty(
dataLine) || contains(dataLine, Point ()
|| contains(dataLine, Px) || contains(
dataLine, [])
54                 break;
55             end
56             data = textscan(dataLine,
formatSpec, 1, CollectOutput, true,
EmptyValue, NaN);
57             if ~isempty(data{1})
58                 pointData = data{1};
59                 chiavePunto = sprintf(%f_%f_
%f, pointData(1)/10, pointData(2)/10,
pointData(3)/10);
60                 % condizione di unicita
applicata a tutti i branch
61                 if ~puntiUnici.isKey(
chiavePunto)

```

```

62         puntiUnici(chiavePunto)
= indicePunti;
63         indicePunti =
indicePunti + 1;
64         allPx = [allPx;
pointData(1)/10];
65         allPy = [allPy;
pointData(2)/10];
66         allPz = [allPz;
pointData(3)/10];
67         allDfit = [allDfit;
pointData(4)/10];
68         allMarker = [allMarker;
currentMarker(:)]; % Correzione qui
69         end
70     end
71 end
72 end
73 end
74 fclose(fileID);
75 % Gestione NaN con media dei valori
    adiacenti e tabella di controllo
76 i = 1;
77 while i <= length(allDfit)
78     if isnan(allDfit(i))
79         % Cerca valori validi prima e dopo
    il NaN
80         j = i + 1;
81         while j <= length(allDfit) && isnan(
allDfit(j))
82             j = j + 1;
83         end

```

```

84         if i > 1 && j <= length(allDfit) %
Valori prima e dopo il NaN
85             media = (allDfit(i-1) + allDfit(
j)) / 2;
86             allDfit(i:j-1) = media;
87         end
88         i = j; % Salta i NaN appena gestiti
89     else
90         i = i + 1;
91     end
92 end
93 %%
94 % Creazione dei file Excel per ogni marker
95 uniqueMarkers = unique(allMarker);
96 for marker = uniqueMarkers
97     % Verifica le dimensioni degli array
98     if length(allPx) == length(allMarker) &&
length(allPy) == length(allMarker) &&
length(allPz) == length(allMarker) &&
length(allDfit) == length(allMarker)
99         markerData = [allPx(allMarker ==
marker), allPy(allMarker == marker), allPz
(allMarker == marker), allDfit(allMarker
== marker), allMarker(allMarker == marker)
];
100         filename = fullfile(DataPoints,
sprintf(branch_%d.xlsx, marker));
101         T = table(markerData(:, 1),
markerData(:, 2), markerData(:, 3),
markerData(:, 4), markerData(:, 5),
VariableNames, {Px, Py, Pz, Dfit, Marker});
102         writematrix(markerData, filename);

```

```

103         fprintf(File "%s" creato con
           successo.\n, filename);
104     end
105 end
106 % Scrittura del numero totale di marker nel
           file di testo
107 numMarkers = length(uniqueMarkers);
108 txtFilename = fullfile(DataPoints,
           marker_count.txt);
109 fileID_txt = fopen(txtFilename, w);
110 fprintf(fileID_txt, %d, numMarkers);
111 fclose(fileID_txt);
112 fprintf(File "%s" creato con successo con il
           numero di marker.\n, txtFilename);
113 disp(Dati estratti e salvati nei file Excel
           .);

```

8.2 Codice VBA:

```
1 Sub Congiunzione_Biforcazioni()  
2  
3     Dim FSO As Object  
4     Dim FolderPath As String  
5     Dim MarkerCountFile As String  
6     Dim MarkerCount As Integer  
7     Dim i As Integer  
8     Dim ExcelFilePath As String  
9     Dim oPartDoc As PartDocument  
10    Dim oPartComponentDefinition As  
11    PartComponentDefinition  
12    Dim x, y, z, dfit, marker As Variant  
13    Dim j As Long  
14    Dim GenericCounter As Long  
15  
16    Dim oWorkPlane As WorkPlane  
17    Dim oSketch As PlanarSketch  
18    Dim oSketchPoint As SketchPoint  
19    Dim oSketchLine3D As SketchLine3D  
20    Dim oProfileCollection As  
21    ObjectCollection  
22    Dim oProfile As Profile  
23    Dim oLoftDefinition As LoftDefinition  
24    Dim oProfileCounter As Integer  
25  
26    ' Percorso della cartella DataPoints  
    FolderPath = "C:\Users\Utente\Desktop\  
DataPoints2\" ' Sostituisci con il  
percorso reale
```

```

27     ' Nome del file con il conteggio dei
marker
28     MarkerCountFile = "marker_count.txt"
29
30     ' Crea un oggetto FileSystemObject per
gestire i file
31     Set FSO = CreateObject("Scripting.
FileSystemObject")
32
33     ' Legge il numero di marker dal file
34     If FSO.FileExists(FolderPath &
MarkerCountFile) Then
35         Open FolderPath & MarkerCountFile
For Input As #1
36         Input #1, MarkerCount
37         Close #1
38     Else
39         MsgBox "File " & MarkerCountFile & "
non trovato."
40         Exit Sub
41     End If
42
43     ' Ottiene il documento Part attivo
44     Set oPartDoc = ThisApplication.
ActiveDocument
45     Set oPartComponentDefinition = oPartDoc.
ComponentDefinition
46
47
48     ' Ciclo per ogni marker
49     For i = 1 To MarkerCount
50

```

```

51         ' Crea una nuova raccolta di
WorkPoints per ogni marker
52         Dim oWorkPoints As WorkPoints
53         Set oWorkPoints =
oPartComponentDefinition.WorkPoints
54
55
56         ' Crea un nuovo oggetto Sketch3D per
ogni marker
57         Dim oSketch3D As Sketch3D
58         Set oSketch3D =
oPartComponentDefinition.Sketches3D.Add
59
60         ' Crea un nuovo oggetto SketchCircle
per ogni marker
61         Dim oSketchCircle As SketchCircle
62
63         ' Costruisce il percorso del file
Excel corrente
64         ExcelFilePath = FolderPath & "
branch_" & i & ".xlsx"
65
66         ' Verifica se il file Excel esiste
67         If FSO.FileExists(ExcelFilePath)
Then
68
69             ' Legge i dati dal file Excel
70             Call LeggiDatiExcel(
ExcelFilePath, x, y, z, dfit, marker)
71
72             ' Crea un solido per il marker
73             Call CreaSolidoPerMarker(
oPartDoc, oPartComponentDefinition,

```

```

oWorkPoints, x, y, z, dfit, marker, i)
74     Else
75         MsgBox "File " & ExcelFilePath &
" non trovato. Si passa al marker
successivo."
76     End If
77
78     Next i
79
80 End Sub
81
82 Private Sub CreaSolidoPerMarker(oPartDoc As
PartDocument, oPartComponentDefinition As
PartComponentDefinition, oWorkPoints As
WorkPoints, x As Variant, y As Variant, z
As Variant, dfit As Variant, marker As
Variant, MarkerIndex As Integer)
83
84     Dim j As Long
85     Dim GenericCounter As Long
86     Dim oSketch3D As Sketch3D
87     Dim oSketchLine3D As SketchLine3D
88     Dim oWorkPlane As WorkPlane
89     Dim oSketch As PlanarSketch
90     Dim oSketchPoint As SketchPoint
91     Dim oSketchCircle As SketchCircle
92     Dim oProfileCollection As
ObjectCollection
93     Dim oProfile As Profile
94     Dim oLoftDefinition As LoftDefinition
95     Dim oWorkPoint As WorkPoint
96
97

```

```

98
99     Set oSketch3D = oPartComponentDefinition
    .Sketches3D.Add

100
101     ' Filtra i punti in base al MarkerIndex
    Dim FilteredWorkPoints As Collection
102     Set FilteredWorkPoints = New Collection
103
104
105     ' Crea WorkPoints e filtra
    For j = 1 To UBound(x, 1)
106         If marker(j, 1) = MarkerIndex Then
107             Set oWorkPoint =
108                 oPartComponentDefinition.WorkPoints.
                AddFixed(ThisApplication.TransientGeometry
                .CreatePoint(CDbl(x(j, 1)), CDbl(y(j, 1)),
                CDbl(z(j, 1))))
109                 FilteredWorkPoints.Add
                oWorkPoint
110             End If
111         Next j
112
113     Set oProfileCollection = ThisApplication
    .TransientObjects.CreateObjectCollection '
    Crea la collezione fuori dal ciclo
114
115     For GenericCounter = 1 To
    FilteredWorkPoints.Count
116         ' Rimuovi la condizione If (
    GenericCounter) < FilteredWorkPoints.Count
117         Set oSketchLine3D = oSketch3D.
        SketchLines3D.AddByTwoPoints(
        FilteredWorkPoints.Item(GenericCounter),
        FilteredWorkPoints.Item(GenericCounter +

```

```

1) , False , 1)
118
119         Set oWorkPlane =
oPartComponentDefinition.WorkPlanes.
AddByNormalToCurve(oSketchLine3D ,
FilteredWorkPoints.Item(GenericCounter))
120         oWorkPlane.Visible = False
121         oWorkPlane.Grounded = True
122         Set oSketch =
oPartComponentDefinition.Sketches.Add(
oWorkPlane)
123         Set oSketchPoint = oSketch.
AddByProjectingEntity(FilteredWorkPoints.
Item(GenericCounter))
124         Set oSketchCircle = oSketch.
SketchCircles.AddByCenterRadius(
oSketchPoint , CDb1(df1t(GenericCounter , 1)
) / 2)
125         Set oProfile = oSketch.Profiles.
AddForSolid
126
127         Call oProfileCollection.Add(oProfile
) ' Aggiungi il profilo alla collezione
128
129     ' ' Aggiungi una condizione per
evitare errori all'ultima iterazione
130     '         If GenericCounter =
FilteredWorkPoints.Count Then
131     '             Exit For
132     '         End If
133
134     Next GenericCounter
135

```

```

136     ' Crea il LoftDefinition solo dopo che
tutti i profili sono stati aggiunti
137     Set oLoftDefinition =
oPartComponentDefinition.Features.
LoftFeatures.CreateLoftDefinition(
oProfileCollection, kNewBodyOperation)
138     Call oPartComponentDefinition.Features.
LoftFeatures.Add(oLoftDefinition)
139
140 End Sub
141 Private Sub LeggiDatiExcel(ExcelFilePath As
String, ByRef x As Variant, ByRef y As
Variant, ByRef z As Variant, ByRef dfit As
Variant, ByRef marker As Variant)
142
143     Dim oExcelApp As Object
144     Dim oWorkbook As Object
145     Dim oWorksheet As Object
146     Dim RowLimit As Long
147     Dim GenericCounter As Long
148
149     On Error GoTo ErrHandler:
150
151     ' Crea un'istanza di Excel
152     Set oExcelApp = CreateObject("Excel.
Application")
153     oExcelApp.Visible = False
154
155     ' Apre il file Excel
156     Set oWorkbook = oExcelApp.Workbooks.Open
(ExcelFilePath)
157     Set oWorksheet = oWorkbook.Sheets(1)
158

```

```

159     ' Determina l'ultima riga con dati
160     RowLimit = oWorksheet.Cells(oWorksheet.
Rows.Count, 1).End(xlUp).Row
161
162     ' Dimensiona le varianti per contenere i
dati
163     ReDim x(1 To RowLimit, 1 To 1)
164     ReDim y(1 To RowLimit, 1 To 1)
165     ReDim z(1 To RowLimit, 1 To 1)
166     ReDim dfit(1 To RowLimit, 1 To 1)
167     ReDim marker(1 To RowLimit, 1 To 1)
168
169     ' Legge i dati dal file Excel
170     For GenericCounter = 1 To RowLimit
171         x(GenericCounter, 1) = oWorksheet.
Cells(GenericCounter, 1).Value2 ' Colonna
X
172         y(GenericCounter, 1) = oWorksheet.
Cells(GenericCounter, 2).Value2 ' Colonna
Y
173         z(GenericCounter, 1) = oWorksheet.
Cells(GenericCounter, 3).Value2 ' Colonna
Z
174         dfit(GenericCounter, 1) = oWorksheet
.Cells(GenericCounter, 4).Value2 ' Colonna
Dfit
175         marker(GenericCounter, 1) =
oWorksheet.Cells(GenericCounter, 5).Value2
' Colonna Marker
176     Next GenericCounter
177
178     ' Conversione esplicita in Double
179     For GenericCounter = 1 To RowLimit

```

```

180         x(GenericCounter, 1) = Cdbl(x(
GenericCounter, 1))
181         y(GenericCounter, 1) = Cdbl(y(
GenericCounter, 1))
182         z(GenericCounter, 1) = Cdbl(z(
GenericCounter, 1))
183         dfit(GenericCounter, 1) = Cdbl(dfit(
GenericCounter, 1))
184         marker(GenericCounter, 1) = Cdbl(
marker(GenericCounter, 1))
185     Next GenericCounter
186
187     ' Chiude e rilascia gli oggetti Excel
188     oWorkbook.Close False
189     oExcelApp.Quit
190     Set oWorksheet = Nothing
191     Set oWorkbook = Nothing
192     Set oExcelApp = Nothing
193
194     Exit Sub
195
196 ErrHandler:
197     Call MsgBox("Unexpected error while
reading excel file: " & Err.Description,
vbOKOnly, "Excel File Error")
198     If Not oWorksheet Is Nothing Then
199         Set oWorksheet = Nothing
200     If Not oWorkbook Is Nothing Then
201         Set oWorkbook = Nothing
202     If Not oExcelApp Is Nothing Then
203         oExcelApp.Quit
204     End If
205     End If

```

```
206     End If
```

```
207
```

```
208 End Sub
```

Bibliografia

- [1] Mikell P. Groover. *Fundamentals of Modern Manufacturing*. John Wiley & Sons, 2007.
- [2] Michael E. Mortenson. *Geometric Modeling*. Industrial Press Inc., 2006.
- [3] Nicolette Birbara, James Otton, and Nalini Pather. 3d modelling and printing technology to produce patient-specific 3d models. *Heart, Lung and Circulation*, 28, 11 2017.
- [4] Materialise NV. Mimics innovation suite. <https://www.materialise.com/en/medical/software/mimics-innovation-suite>.
- [5] 3D Slicer Community. 3d slicer. <https://www.slicer.org/>.
- [6] Paul A. Yushkevich, Joseph Piven, Heather C. Hazlett, Richard G. Smith, Susan Ho, James C. Gee, and Guido Gerig. Itk-snap: An interactive tool for semi-automatic segmentation of multi-modality 3d medical images. <http://www.itksnap.org/>.
- [7] Materialise NV. Materialise 3-matic Industrial | 3D Data Optimization Software.
- [8] Thermo Fisher Scientific. Amira software. <https://www.thermofisher.com/it/it/home/industrial/electron-microscopy/software-em-industrial/amira-software.html>.
- [9] Autodesk. Autodesk - official website, 2025.
- [10] ANSYS, Inc. Ansys meshing. <https://www.ansys.com/products/fluids/ansys-meshing>.
- [11] Altair Engineering, Inc. Altair hypermesh. <https://altair.com/hypermesh>.

- [12] Hang Si. Tetgen: A delaunay-based quality tetrahedral mesh generator. <http://tetgen.org/>.
- [13] Jan Kretschmer, Christian Godenschwager, Bernhard Preim, and Marc Stamminger. Interactive patient-specific vascular modeling with sweep surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2828–2837, 2013.
- [14] Charles A. Taylor, Kersten Petersen, Nan Xiao, Matthew Sinclair, Ying Bai, Sabrina R. Lynch, Adam UpdeGraff, and Michiel Schaap. Patient-specific modeling of blood flow in the coronary arteries. *Comput. Methods Appl. Mech. Engrg*, 417:116414, 2023.
- [15] Méghane Decroocq, Carole Frindel, Pierre Rougé, Makoto Ohta, and Guillaume Lavoué. Modeling and hexahedral meshing of cerebral arterial networks from centerlines. *Medical Image Analysis*, 84:102685, 2023.
- [16] Xunlei Wu, Vincent Luboz, Karl Krissian, Stephane Cotin, and Steve Dawson. Segmentation and reconstruction of vascular structures for 3d real-time simulation. *Medical image analysis*, 15(1):22–34, 2011.
- [17] Autodesk inventor: Guida dell’utente, 2023. Disponibile online: <https://help.autodesk.com/view/INVNTOR/2017/FRA/?guid=GUID-16291103-FB93-4351-9671-E2A5F6CEBEFC>.
- [18] Industrial automation systems and integration – product data representation and exchange – part 21: Implementation methods: Clear text encoding of the exchange structure, 2001.
- [19] Gerald Farin. *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann, 5 edition, 2002.
- [20] Materialise NV. Materialise website.
- [21] Materialise NV. Materialise mimics.
- [22] MathWorks. Matlab documentation.
- [23] Autodesk Inc. Autodesk netfabb, Anno di pubblicazione o aggiornamento, se disponibile. Software per la produzione additiva.
- [24] Ansys Inc. Ansys, Anno di pubblicazione o aggiornamento, se disponibile. Software per la simulazione multifisica.

Ringraziamenti

Vorrei innanzitutto ringraziare la mia relattrice, Prof. Stefania Marconi, per la sua disponibilità, pazienza e per i suoi consigli costruttivi che mi hanno guidato nel completamento di questa tesi.

Un grazie sincero va anche al mio correlatore, Dott. Alberto Cattenone, che mi ha aiutato con le sue competenze pratiche, offrendo soluzioni e spunti che hanno arricchito la qualità del mio lavoro.

Un ringraziamento speciale va al mio Papà, che mi ha spronato e sostenuto in ogni fase di questo percorso. Il suo amore incondizionato, la sua pazienza e il suo incoraggiamento costante non sono mai venuti meno, rendendo ogni difficoltà più affrontabile. Rimarrò l'ingegnere biomedico che non sa come funziona uno scaldabagno, ma almeno ora posso dire: 'Ingegnere Bis!'

Grazie alla mia migliore amica, Fra, che nonostante i 16.249 km di distanza e un fuso orario che confonde tutti, è sempre stata pronta ad ascoltarmi, consolarmi nei momenti di dubbio e a gioire insieme a me, urlando di felicità ogni volta che un esame era finalmente superato.

Un grazie di cuore a Saratonin, Marghe e Marco, per avermi sopportato durante la ricerca e la stesura di questa tesi. So di aver dato segni di pazzia lungo il percorso, ma voi siete stati sempre lì per tirarmi su e darmi la forza di andare avanti.

Un grazie speciale a Charith, il mago di VBA, che mi ha insegnato a utilizzare un software da zero in meno di due mesi. La sua pazienza e competenza sono state fondamentali.

Grazie a tutti i miei 'collegi di magistrale': Luca, Giulia, Lavi, Ale, Vassy, Ema, Saratonin, Marco, Marghe e Chiara, che mi avete accettata e accolta

nonostante non conoscessi nessuno all'inizio di questo percorso. E anche per i nostri super pranzi pieni di risate, ma anche di discussioni intense ma non serie, che hanno reso ogni giornata più leggera e divertente.

Un grazie speciale a Sara B, che mi ha aiutata in tutto, e anche un po' convinta a intraprendere questa avventura della magistrale. Le nostre conversazioni, che partono sempre da un punto e poi, magicamente, arrivano da qualche altra parte, sono state una fonte inesauribile di supporto e risate.

Grazie alla mia BiBi, che mi accompagna ogni giorno con il suo amore incondizionato, che mi obbliga a uscire per andare al parco e rende ogni momento più gioioso.

Dedico questa laurea **alla mia mamma**, la mia forever cheerleader, che mi ha sempre sostenuta con amore e forza in ogni passo del mio cammino. Anche se non è qui fisicamente, sono sicura che sarebbe molto felice non solo per il traguardo che ho raggiunto, ma anche per il fatto che sono circondata da persone che mi vogliono bene.

Grazie a tutti.