



UNIVERSITÀ
POLITECNICA
DELLE MARCHE

FACOLTÀ DI INGEGNERIA
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA MECCANICA

Sviluppo di geometrie complesse realizzate tramite modelli computazionali per processi di metal additive manufacturing

**Development of complex geometries created using computational models
for metal additive manufacturing processes**

Candidato:
Luca Marozzi

Relatore:
Prof.ssa Eleonora Santecchia

Anno Accademico 2023-2024



UNIVERSITÀ
POLITECNICA
DELLE MARCHE

FACOLTÀ DI INGEGNERIA
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA MECCANICA

Sviluppo di geometrie complesse realizzate tramite modelli computazionali per processi di metal additive manufacturing

**Development of complex geometries created using computational models
for metal additive manufacturing processes**

Candidato:
Luca Marozzi

Relatore:
Prof.ssa Eleonora Santecchia

Anno Accademico 2023-2024

UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA MECCANICA
Via Brezze Bianche – 60131 Ancona (AN), Italy

*Alla mia famiglia
e a tutti quelli che mi hanno
accompagnato in questo percorso*

Sommario

La tecnologia avanza a un ritmo sorprendente al giorno d'oggi e per applicazioni che vanno dal fenomeno della mass customization che sta pian piano aumentando la varietà richiesta dai consumatori, alle elevate performances richieste dai settori aerospace e automotive, le tecnologie di Additive Manufacturing risultano essere una frontiera preponderante grazie alla loro capacità di realizzare geometrie particolarmente complesse in tempi relativamente brevi. In particolar modo per ottenere elevati risultati in termini prestazionali, i ricercatori e gli ingegneri di tutto il mondo si sono concentrati, nell'ultimo decennio, sui metamateriali. Con il termine metamateriale si intendono tutte quelle architetture geometriche ingegnerizzate che permettono il conseguimento di proprietà non raggiungibili da parte dei comuni materiali, di impiego nell'industria meccanica o presenti in natura. Tali strutture geometriche devono le loro proprietà alla morfologia caratteristica dell'unità cellulare di costruzione, piuttosto che alla composizione chimica degli elementi con cui vengono prodotti. Il presente lavoro si focalizza sullo sviluppo di geometrie complesse da realizzare in AISI 316L attraverso i processi di Metal Additive Manufacturing. L'attività è iniziata con lo studio delle superfici minime e delle Triply Periodic Minimal Surfaces (TPMS) per comprendere le loro proprietà matematiche e il potenziale applicativo. Dopo un'attenta ricerca bibliografica sui metamateriali altoperformanti a prove di compressione, sono state selezionate delle celle base con cui costruire i pattern cellulari. Si è proceduto poi alla valutazione di software di modellazione per strutture lattice e Metafold 3D è stato scelto per la sua capacità di generare superfici analitiche di alta qualità, pur se in data odierna ancora in una fase iniziale di sviluppo. La fase di modellazione CAD ha preso come base la geometria di un provino standard per la prova di Hopkinson, un parallelepipedo di dimensioni 25x25x25 mm, creando diversi campioni con reticoli di vario genere e con diverse proprietà. I modelli selezionati sono stati poi importati nel software NX per simulare il processo di Laser Powder Bed Fusion (LPBF), analizzando la fattibilità della stampa, gli stress e le deformazioni residue che ci si può aspettare dalla realizzazione dei provini. Infine, è stata utilizzata la simulazione meccanica tramite il software ANSYS per prevedere il comportamento meccanico dei provini in una prova di compressione. Nonostante le restrizioni temporali abbiano impedito la stampa e la caratterizzazione dei modelli CAD, il lavoro ha permesso di stabilire una base per la futura realizzazione e caratterizzazione di questi provini.

Indice

1	Concetti introduttivi	1
1.1	Perché le TPMS?	1
1.2	Dalle superfici minime alle TPMS	4
1.2.1	I primi approcci matematici	4
1.2.2	La svolta dell'approccio pratico	6
1.2.3	L'entrata in campo complesso	6
1.2.4	Rappresentazione implicita	8
1.3	Generalità sui processi di AM e processo LPBF	11
1.3.1	Aspetti generali	11
1.3.2	Laser Powder Bed Fusion	13
2	Strumenti e Metodi	19
2.1	Introduzione	19
2.2	La scelta dell'ambiente di modellazione	19
2.2.1	Metafold 3D	21
2.2.2	PicoGK	27
2.3	Altri software	30
2.3.1	Cenni sui software	31
3	Procedure sperimentali	35
3.1	Scelta dei campioni ottimali per la sperimentazione meccanica	35
3.1.1	Barra di Hopkinson	35
3.1.2	Scelta delle strutture	36
3.2	Processo di modellazione	36
3.3	Analisi geometrica	45
3.4	Processo di simulazione	55
3.4.1	Simulazione di stampa	55
3.4.2	Simulazione della prova meccanica	60
4	Analisi dei risultati	67
4.1	Analisi della simulazione di stampa	67
4.2	Analisi della simulazione meccanica	71
5	Conclusione	77
5.1	Considerazioni finali	77
5.2	Sviluppi futuri	78

Elenco delle figure

1.1	Catenodie, prima superficie minima matematicamente descritta. . .	5
1.2	Cella unitaria della prima superficie di Scherk	6
1.3	Cella unitaria della Schwarz P detta Primitive	7
1.4	Due metodi tipici di modellazione delle unità TPMS [1]	10
1.5	Schema costruttivo architettura macchine LPBF	13
1.6	strategie per il posizionamento dei pezzi sul job di stampa. (1) posizionamento angolato rispetto al recoater per minimizzare le forze di impatto, (2) evitare zone in cui il recoater possa incagliarsi e sollecitare a flessione la parte, (3) far incontrare al recoater contemporaneamente meno parti possibile, (4) evitare di posizionare i pezzi uno dietro l'altro	15
1.7	strategia di scansione per la riduzione dell'anisotropia del pezzo stampato con rotazione della direzione di scansione di 67° ogni layer	16
1.8	analisi dell'influenza della potenza del laser e della velocità di scansione, a parità di hatch spacing e spessore del layer, sulla realizzazione di tracce esenti da difetti	17
1.9	Micrografie SEM di (a) le travi della struttura a reticolo cellulare giroide con dimensione della cella di 2 mm e frazione volumetrica del 15% realizzata mediante il processo SLM, (b) la sezione trasversale di una trave, (c) particelle legate alle superfici delle travi e (d) una particella individuale sulla superficie della trave.	17
2.1	Il grafico (in basso, in rosso) della distanza con segno tra i punti del piano xy (in blu) e un disco fisso (rappresentato anche in alto, in grigio)	20
2.2	Lattice Strut e TPMS di base	23
2.3	Conformazione sferica e cilindrica	23
2.4	Grading su geometria ellittica	24
2.5	Grading di materiale	24
2.6	Geometria Strut Conformal	25
2.7	Geometria Giroide Conformal	25
2.8	Geometria Giroide come infill con il comando shell	26
2.9	Sfera Custom Shape di raggio 50 centrata nell'origine	26
2.10	Modello di una giroide Tubular con frazione di volume del 20%	30
3.1	Modello realizzato in Metafaold3D di una giroide skeletal con frazione di volume del 20%	39

Elenco delle figure

3.2	Modello realizzato in Metafaold3D di una giroide skeletal Graded con frazione di volume del 20%	40
3.3	Geometria Skeletal Double giroide con densità al 20%	42
3.4	Geometria Skeletal Double Connected giroide con densità al 20%	43
3.5	(Varianti) Geometria Skeletal Double Connected giroide con densità al 20%	44
3.6	Modello di una giroide Tubular con frazione di volume del 20%	46
3.7	Risultato del calcolo di densità di una giroide skeletal	49
3.8	L'albero di conversione (sulla sinistra) e il modello caricato su nTop della Skeletal Double Connected Horizontal (sulla destra)	57
3.9	Rifilo del piano inferiore	58
3.10	Parametri del volume della stampante (sulla sinistra) e volume di stampa (sulla destra)	58
3.11	Compente sul piatto di stampa, essendo molto piccolo si vede solo il piano verde	59
3.12	Mesh del provino SD Connected Horizontal e i parametri della Mesh	59
3.13	Box del modulo Explicit Dynamics	60
3.14	Modello dell'AISI 316L contenuto nel database universitario	61
3.15	Mesh realizzata da Ansys delle geometrie Tubular (a) e Graded (b)	62
3.16	Assieme provino-piatti importato su Ansys, la struttura mostrata è la Tubular	62
3.17	Definizione contatto tra piatto fisso e base del provino	63
3.18	Definizione contatto tra piatto fisso e base del provino	64
3.19	Albero della simulazione meccanica con tutte le voci necessarie assegnante	65
4.1	Risultato simulazione di stampa Giroide Skeltal di riferimento	68
4.2	Risultato simulazione di stampa Giroide Skeltal Graded	69
4.3	Risultato simulazione di stampa Giroide Skeltal Double	69
4.4	Risultato simulazione di stampa Giroide Tubular	70
4.5	Risultato simulazione di stampa Giroide Skeltal Double Connected	70
4.6	Risultato simulazione di stampa Giroide Skeltal Double Connected Vertical	71
4.7	Grafico sforzo-deformazione della struttura Skeletal di riferimento	72
4.8	Grafico sforzo-deformazione della struttura Skeletal Double	72
4.9	Grafico sforzo-deformazione della struttura Skeletal Double	73
4.10	Grafico sforzo-deformazione della struttura Skeletal Double	74
4.11	Grafico sforzo-deformazione della struttura Skeletal Double	75

Elenco delle tabelle

1.1	Espressione matematica di diverse unità TPMS	9
3.1	Tabella contenente i parametri del modello Johnson-Cook	61
4.1	Dati sull'assorbimento di energia SD	75
4.2	Dati sull'assorbimento di energia Skeletal di riferimento	76
4.3	Dati sull'assorbimento di energia Graded	76
4.4	Dati sull'assorbimento di energia Tubular	76

Capitolo 1

Concetti introduttivi

1.1 Perché le TPMS?

Affrontare una sfida ingegneristica implica l'aderenza a specifici parametri di progetto. Tra questi, le performance meccaniche rappresentano un vincolo cruciale che deve essere soddisfatto da un artefatto progettato per una determinata funzione. Per garantire le prestazioni meccaniche richieste a un componente, è possibile intervenire in tre modi principali:

- Modificando la composizione chimica del materiale, selezionandolo, quindi, in funzione prettamente delle sue caratteristiche tecniche;
- Alterando la microstruttura del materiale attraverso trattamenti termici mirati, il che si lega a doppio filo con la scelta del materiale dato che i trattamenti disponibili dipendono dal materiale stesso;
- Migliorando l'architettura topologica del materiale, rendendolo più efficiente e condizionandone profondamente le proprietà.

Proprio su quest'ultimo punto si basa la grande sfida ingegneristica degli ultimi decenni: la realizzazione di materiali che hanno proprietà uniche grazie alla loro struttura morfologica specifica e precisa. Ad oggi numerose strutture complesse vengono costruite e applicate in diversi campi dell'ingegneria. Indipendentemente dal fatto che queste strutture siano composte da materiali metallici, materiali inorganici non metallici, polimeri o altri materiali più complessi, la maggior parte delle strutture attualmente utilizzate nell'industria sono classificate come strutture solide. Pertanto, le porosità delle strutture solide possono essere considerate pari a zero. Infatti, i pori interni sono considerati difetti causati da una bassa qualità manifatturiera. Tuttavia, esistono in natura molte strutture porose con prestazioni eccezionali, come ossa, coralli, favi e legni. Ispirandosi a queste strutture naturali, si moltiplicano i tentativi di generare strutture porose artificiali.

Numerose schiume porose (foam) vengono generate tramite lisciviazione del sale (salt leaching) o schiumatura a gas. Le caratteristiche porose e le prestazioni delle schiume così generate non sono facili da controllare. Pertanto, sono state sviluppate strutture reticolari (lattice structures) composte da montanti e nodi (strut-based), che

possono essere considerate come un secondo tipo di strutture porose. Le geometrie e le prestazioni possono essere comodamente controllate regolando la lunghezza, il raggio dei montanti e la topologia delle connessioni dei montanti. A causa dei montanti interni intricati, la maggior parte delle strutture strut-based sono fabbricate mediante manifattura additiva. Tuttavia, la concentrazione di stress può apparire nelle connessioni dei montanti del reticolo. Basandosi su ciò, al fine di migliorare ulteriormente le prestazioni, il terzo tipo di strutture porose è progettato dalle triply periodic minimal surfaces (TPMS), che sono l'oggetto principale di ricerca di questo lavoro.

Le TPMS sono un tipo di superficie implicita periodica con curvatura media nulla. Pertanto, rispetto ad altri tipi di strutture, le strutture porose TPMS possiedono due significativi vantaggi. (a) L'intera struttura può essere espressa con precisione tramite funzioni matematiche. Le prestazioni di base, come la porosità o le aree superficiali specifiche del volume, possono essere controllate direttamente regolando i parametri della funzione. (b) Le superfici di TPMS sono molto lisce, senza spigoli vivi o giunzioni come le strutture strut-based. Inoltre, le strutture porose TPMS sono altamente interconnesse con pori non tortuosi, aspetto importante per le applicazioni. Grazie alle caratteristiche implicite, la posizione, la curvatura e il periodo di una TPMS possono essere facilmente controllati.

Sebbene le TPMS siano un argomento importante di ricerca in diverse aree, i suoi vantaggi non sono ancora completamente sfruttati. La ricerca attuale si è concentrata principalmente sulle prestazioni o le applicazioni di una singola disciplina, risultando in direzioni caotiche e frammentate. Studi interdisciplinari sono necessari per promuovere le applicazioni del TPMS. Ad esempio, possono essere progettate con forme complesse, porosità graduata e pori multiscale attraverso algoritmi CAD. Tuttavia, la maggior parte delle strutture TPMS adottate, sia in terminologia che in chimica, sono ancora standard con porosità uniforme e forme semplici.

L'attenzione alla ricerca sulle applicazioni delle TPMS è in crescita. Grazie alle superfici lisce e ai pori altamente interconnessi, le superfici TPMS sono state applicate con successo in aree multidisciplinari. Applicazioni rappresentative si trovano nei domini meccanico, termico, biologico, chimico, acustico e ottico.

Le prestazioni meccaniche di base delle strutture porose TPMS sono state studiate sistematicamente. Basandosi sulla lunga fase elastica lineare e simile ad altre strutture porose, le TPMS possono essere applicate come assorbitori di energia sotto compressione [2–5] o assorbitori di impatto [6]. Data la loro natura di strutture porose il peso delle TPMS è molto inferiore rispetto alla struttura solida per lo stesso volume. Sfruttando metodi di ottimizzazione, queste possono essere impiegate come strutture leggere per ridurre il consumo di materiali ed energia. Nel dominio dell'ingegneria aerospaziale o automobilistica, ad esempio, i pannelli sandwich sono ampiamente applicati. I pannelli sandwich tipici sono composti da due pannelli frontali con strutture solide e nuclei interni porosi. Negli ultimi decenni, la maggior parte

dei pannelli sandwich erano con nuclei strut-based. Recentemente, nuovi pannelli sandwich si basano sulle strutture TPMS [7], ottenendo proprietà di flessione elevate e capacità di assorbimento dell'energia. Inoltre, le strutture TPMS possono essere impiegate direttamente come componenti funzionali. Alkebsi et al hanno utilizzato le TPMS come pale di turbine [8]. I parametri delle TPMS possono essere regolati per ottenere la rigidità variabile lineare. Pan et al hanno generato cerniere a flessione basate sulle TPMS [9], dimostrando che la superficie P è la più adatta per le cerniere a flessione, migliorando sia la conformità che il rapporto di conformità rispetto alle tradizionali cerniere a foglia. Grazie ai bandgap di vibrazione meccanica, le TPMS possono anche essere utilizzate come isolatori di vibrazioni [10]. Questi esempi sono solo alcuni delle numerose applicazioni nel dominio meccanico.

Grazie alle aree superficiali specifiche di alto volume, le prestazioni di scambio del calore delle TPMS sono eccezionali, quindi possono essere utilizzate come dissipatori di calore [11–13] o scambiatori di calore (Attarzadeh et al [14]), che rappresentano le principali applicazioni nel dominio termico. I risultati sperimentali indicano che lo scambio di calore tra l'aria in movimento e la fonte di calore può essere più efficiente. Le prestazioni termiche dello scambiatore di calore possono essere migliorate con uno spessore della parete ridotto. La ricerca e le applicazioni delle TPMS nel dominio termico sono ancora agli inizi, con la maggior parte degli scambiatori di calore TPMS progettati con forme standard e porosità uniformi. Un problema interessante è come migliorare ulteriormente le prestazioni di trasferimento del calore degli scambiatori di calore TPMS con un volume minimo.

Le strutture TPMS sono ampiamente applicate anche nel dominio biologico. La geometria e la topologia delle TPMS sono simili alle strutture naturali, offrendo numerosi vantaggi per le applicazioni biologiche. Le impalcature per l'ingegneria dei tessuti e gli impianti medici sono applicazioni tipiche delle strutture porose TPMS. A differenza delle strutture strut-based o foam, le superfici lisce delle TPMS sono adatte all'adesione e alla crescita delle cellule. Le aree superficiali specifiche di alto volume e le architetture porose interconnesse forniscono spazio sufficiente per il trasporto di nutrimento e sostanze di scarto [15]. Le strutture porose TPMS sono impiegate nella tissue engineering [16–20] e nei dispositivi di impianto [21–25]. Alabort et al hanno verificato l'uso delle TPMS come ossa metalliche mediante stampa 3D [26]. Le TPMS sono state applicate anche come impalcature per diverse ossa naturali. Zhu et al hanno utilizzato le TPMS come impianti meniscali [21], prevenendo maggiori compressioni e stress di taglio sull'articolazione cartilaginea e mantenendo alcune caratteristiche di base. In generale, le TPMS sono state verificate come un candidato ideale per le applicazioni biologiche, con impalcature o impianti adatti alle esigenze dell'ambiente biologico umano.

Le architetture interconnesse e l'alto rapporto superficie volume delle TPMS offrono vantaggi anche in altri ambiti. In campo chimico, ad esempio, l'efficienza e la qualità delle reazioni chimiche possono essere migliorate con aree di contatto più elevate. Pertanto, le TPMS hanno un grande potenziale come catalizzatori o microreatori [27, 28] e membrane [29, 30]. I sistemi di stoccaggio dell'energia elettrica, come le batterie, sono fondamentali nell'ingegneria attuale. Tuttavia, migliorare l'efficienza di rilascio e stoccaggio dell'energia rimane una grande sfida. Werner et al hanno progettato nuove batterie basate sulle strutture porose TPMS [31], con anodo e catodo in diversi canali delle TPMS. Inoltre, le TPMS consentono di risparmiare più spazio interno. Per generare la stessa capacità di una batteria a tre strati con le stesse dimensioni e materiali dello strato, sarebbe necessario uno spazio 4700 volte più grande.

Le TPMS sono anche scelte ideali come assorbitori di suono [32, 33], assorbimento di microonde elettromagnetiche [34] e come cristalli fotonici [35].

1.2 Dalle superfici minime alle TPMS

Le superfici minime rappresentano un ambito di studio molto interessante della geometria differenziale. Queste sono definite, come già menzionato, come superfici in cui la curvatura media in ogni punto è nulla. In natura esempi di superfici minime si associano a problemi di minimizzazione dell'energia: ad esempio se si immerge un telaio in dell'acqua saponata la forma che assume il sapone, a meno di deviazioni dovute a interazione con forze esterne come la gravità, risulta essere una superficie minima. I quesiti matematici ispirati da situazioni quotidiane sono tra i più vecchi nella storia della matematica. Si dice che Archimede fu il primo a introdurre i concetti di lunghezza e area minima in geometria. Comprese che il segmento rettilineo è il cammino più breve tra due punti nello spazio e che, data una curva chiusa piana qualsiasi, la superficie con area minima che ha tale curva come bordo è la porzione di piano delimitata dalla curva.

1.2.1 I primi approcci matematici

Proprio nelle dimostrazioni di Archimede la teoria delle superfici minime affonda le proprie basi ossia la soluzione di **problemi di area minima**: date una o più curve chiuse nello spazio trovare la superficie avente area minima tra tutte quelle che hanno le curve indicate come bordo. I problemi di minimizzazione dell'area, quando si considerano più contorni chiusi nello spazio o un singolo contorno non piano, sono più complessi da risolvere rispetto al caso specifico trattato da Archimede e sono esempi classici del settore matematico noto come calcolo delle variazioni. Uno dei primi matematici a studiare il problema delle superfici di area minima fu Eulero che nel 1744 con suo lavoro "*Methodus inveniendi lineas curvas maximi minimive proprietate gaudentes*" affrontò il problema di determinare la superficie di

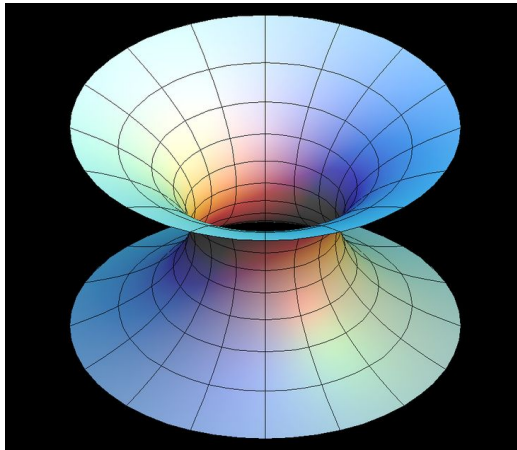


Figura 1.1: Catenodie, prima superficie minima matematicamente descritta.

area minima delimitata da due circonferenze su piani paralleli dove la retta passante per i due centri era ortogonale ai piani. Per intuito Eulero impostò il problema come la ricerca di una curva che genera una **superficie di rivoluzione**. Riuscì poi a dimostrare che la curva cercata era una catenaria chiamando la relativa superficie di rivoluzione **catenoide**.

Da questi problemi di minimizzazione dell'area emerse la teoria delle superfici minime. La data formale di nascita è il 1762, anno in cui Lagrange pubblicò il suo lavoro *"Essai d'une nouvelle méthode pour déterminer les maxima et les minima des formules intégrales indéfinies"*. In esso, Lagrange formulò l'equazione differenziale che deve essere soddisfatta da ogni punto di una superficie che minimizza l'area per un dato contorno nello spazio.

Questa equazione è nota come equazione di Eulero-Lagrange e, in riferimento al problema dell'area minima, Lagrange definì superficie minima ogni superficie che soddisfa tale equazione, da cui il nome equazione delle superfici minime.

Nel 1776, Meusnier scoprì il significato geometrico dell'equazione di Eulero-Lagrange, utilizzato ancora oggi: se una superficie soddisfa l'equazione in un punto, allora in quel punto la curvatura media è nulla. Di conseguenza, le superfici minime sono quelle con curvatura media nulla ovunque, e le superfici che risolvono il problema dell'area minima sono superfici minime. Meusnier trovò anche una nuova superficie minima tra le superfici rigate: l'elicoide retto. Per molto tempo, il piano, la catenoide e l'elicoide retto furono le uniche superfici minime conosciute.

Un altro contributo significativo alla teoria delle superfici minime fu dato da Monge nel 1783, che nella sua opera *"Sur une méthode d'intégrer les équations aux différences ordinaires"* stabilì un'importante relazione tra superfici minime e superfici di area minima: se una superficie che ha come bordo una curva chiusa semplice nello spazio è anche il grafico di una funzione differenziabile, allora essere una superficie minima è condizione necessaria e sufficiente per assicurare che la sua area sia inferiore a quella di qualsiasi altra superficie con lo stesso bordo. In altre parole, nel caso di superfici

che sono grafici di funzioni differenziabili, essere una superficie minima è sufficiente per essere la soluzione del problema dell'area minima per un dato contorno chiuso e semplice nello spazio.

1.2.2 La svolta dell'approccio pratico

Intorno alla metà del diciannovesimo secolo il fisico belga Joseph Plateau inizia lo studio delle forme assunte dalle superfici di sapone. Plateau, tramite l'utilizzo di telai rigidi, sfrutta le proprietà fisiche dell'acqua saponata e le forze di tensione superficiale per costruire numerosi modelli geometrici di superfici minime.

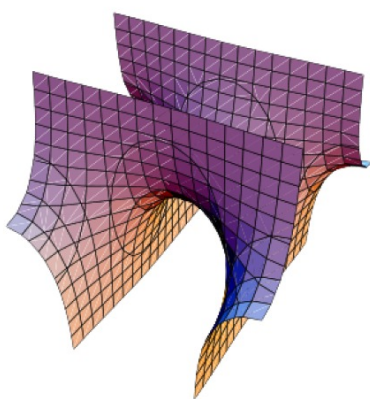


Figura 1.2: Cella unitaria della prima superficie di Scherk

Visto il successo degli esperimenti, da allora il problema di trovare la superficie di area minima avente come bordo un qualunque numero di curve chiuse nello spazio prende il nome di problema di Plateau. Tra il 1831 e il 1835, Scherk scoprì nuovi esempi di superfici minime, tra cui due famose superfici oggi note con il suo nome. Dal 1850 al 1880, la ricerca poi si concentrò sul problema di Plateau cercando soluzioni principalmente per contorni a forma di quadrilateri nello spazio.

Nel 1865, Schwarz risolse il problema nel caso di un contorno formato da quattro spigoli di un tetraedro. Identificò anche alcune regole di simmetria che regolano la forma delle superfici minime, oggi note come principio di riflessione di Schwarz, che gli permisero di costruire nuove

superfici minime illimitate e **periodiche**. Schwarz scoprì quindi le prime Triply Periodic Minimal Surfaces (TPMS), superfici minime illimitate, immerse nello spazio \mathbb{R}^3 . Le superfici di questo tipo hanno una notevole valenza non solo perché dividono lo spazio in **due volumi distinti**, cosa già nota per le superfici minime, ma soprattutto perché sono periodiche nelle tre direzioni dello spazio consentendo di essere utilizzate come cella di base per una struttura reticolare con proprietà specifiche, similmente ai **reticoli di Bravais**.

1.2.3 L'entrata in campo complesso

Nel 1866 Karl Weierstrass scoprì poi un'importante connessione tra le superfici minime e la teoria delle funzioni olomorfe, fornendo una formula che esprime una superficie minima semplicemente connessa in termini di due funzioni olomorfe (per la precisione una meromorfa e una analitica). In matematica, una funzione olomorfa è una funzione a valori complessi di una o più variabili complesse che è differenziabile in un intorno di ciascun punto in un dominio nello spazio di coordinate \mathbb{C}^n . Una funzione meromorfa, invece, è una funzione olomorfa eccetto in dei punti isolati, detti

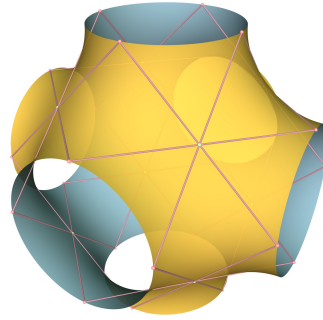


Figura 1.3: Cella unitaria della Schwarz P detta Primitive

poli della funzione. Considerando f e g due funzioni sull'intero piano complesso, dove g è meromorfa e f è analitica, tale che dove g ha un polo di ordine m , f ha uno zero di ordine $2m$ (o equivalentemente, in modo tale che il prodotto fg^2 è olomorfo) e lasciamo c_1, c_2, c_3 costanti, la superficie con le coordinate (x_1, x_2, x_3) è minima, dove le x_k sono definite utilizzando la parte reale di un integrale complesso come segue:

$$x_k(\zeta) = \operatorname{Re} \left\{ \int_0^\zeta \varphi_k(z) dz \right\} + c_k, \quad k = 1, 2, 3$$

$$\varphi_1 = f(1 - g^2)/2$$

$$\varphi_2 = if(1 + g^2)/2$$

$$\varphi_3 = fg$$

Questa scoperta portò alla formulazione della classificazione delle TPMS in **famiglie associate**: la famiglia associata (o famiglia di Bonnet) di una TPMS è un insieme di superfici minime che condividono gli stessi dati di Weierstrass e si distinguono tra loro per un unico parametro. In particolare si aggiunge un parametro polare alla definizione delle coordinate

$$x_k(\zeta, \theta) = \operatorname{Re} \left\{ e^{i\theta} \int_0^\zeta \varphi_k(z) dz \right\} + c_k \quad \theta \in [0, 2\pi]$$

Considerando che le TPMS sono oggetti in tre dimensioni e che sono soggette a vincoli, la loro rappresentazioni parametriche di Enneper–Weierstrass, come descritto in [15], sono

$$x = \operatorname{Re} \left(e^{i\theta} \int_{\omega_0}^\omega (1 - \tau^2)R(\tau) d\tau \right),$$

$$y = \operatorname{Re} \left(e^{i\theta} \int_{\omega_0}^\omega i(1 + \tau^2)R(\tau) d\tau \right),$$

$$z = \operatorname{Re} \left(e^{i\theta} \int_{\omega_0}^\omega 2\tau R(\tau) d\tau \right)$$

dove $i^2 = -1$, τ è una variabile complessa, θ è l'angolo di Bonnet definito prima. $R(\tau)$ è la funzione di Weierstrass di differenti tipi di TPMS. Per esempio la funzione di Weierstrass delle superfici Schwarz D, Schwarz P e Giroide (D, P e G) può essere espressa come:

$$R(\tau) = \frac{1}{\sqrt{\tau^8 - 14\tau^4 + 1}}$$

e gli angoli di Bonnet per le superfici D, P e G sono rispettivamente 0, 90 e 38.0147 gradi [15]. Tuttavia, con questo approccio è possibile generare solo pochi tipi di TPMS. Per ora la funzione Weierstrass è stata trovata solo per alcune superfici minime.

1.2.4 Rappresentazione implicita

Un'altra metodologia di rappresentazione per le TPMS è quella tramite equazioni algebriche in forma implicita, che coincide con una superficie di isolivello. Data una funzione di tre variabili $F(x, y, z)$ le sue superfici di livello sono rappresentate dall'equazione:

$$F(x, y, z) = c$$

dove c è una costante che definisce il valore del livello e, per le TPMS, è anche il parametro di curvatura. In particolare, per le superfici TPMS, si considera il caso in cui $c = 0$. Pertanto l'equazione diventerebbe




$$F(x, y, z) = 0$$

che corrisponderebbe alla forma implicita che meglio approssima la formulazione di Weierstrass della superficie TPMS. In particolare la forma implicita delle TPMS è combinazione di funzioni trigonometriche e, in forma generale, si rappresenta [36]

$$\phi(\mathbf{r}) = \sum_{k=1}^K A_k \cos \left[\frac{2\pi(\mathbf{h}_k \cdot \mathbf{r})}{\lambda_k} + P_k \right] = C$$

dove A_k è l'ampiezza, λ_k è il fattore di periodo e P_k è la fase della funzione. Sulla base di questo le TPMS P, G e D possono essere rappresentate come in tabella 1.1

Tabella 1.1: Espressione matematica di diverse unità TPMS

TPMS	Espressione matematica	Modello 3D
P	$F(x, y, z) = \cos(\omega_x x) + \cos(\omega_y y) + \cos(\omega_z z) = C$	
G	$F(x, y, z) = \sin(\omega_x x) \cos(\omega_y y) + \sin(\omega_z z) \cos(\omega_x x) + \sin(\omega_y y) \cos(\omega_z z) = C$	
D	$F(x, y, z) = \cos(\omega_x x) \cos(\omega_y y) \cos(\omega_z z) - \sin(\omega_x x) \sin(\omega_y y) \sin(\omega_z z) = C$	

In generale, per determinare se una superficie è una superficie minima, dobbiamo verificare che la curvatura media H sia nulla in ogni punto della superficie. La curvatura media è definita come la media aritmetica delle curvatures principali k_1 e k_2 , e si esprime come:

$$H = \frac{1}{2}(k_1 + k_2)$$

Una superficie è considerata minima se e solo se $H = 0$ in ogni suo punto, il che implica che la somma delle curvatures principali deve essere zero ovunque sulla superficie. Per analizzare se tale superficie è minima, dobbiamo calcolare le curvatures principali utilizzando il gradiente e l'Hessiano di F . Il gradiente di F , indicato con ∇F , è un vettore che punta nella direzione di massimo incremento della funzione e la cui norma indica la pendenza della funzione. In questo caso rappresenta la pendenza della funzione $F(x, y, z)$ nello spazio \mathbb{R}^4 . L'Hessiano di F , che è la matrice delle derivate parziali seconde, ci fornisce informazioni sulla concavità o convessità della funzione in un punto e deve essere ristretta al piano tangente alla superficie per poterne valutare le curvatures principali. Per valutare le curvatures principali in un punto $P = (x_0, y_0, z_0)$ dapprima si calcola il **gradiente** di F nel punto come:

$$\nabla F(P) = \left(\frac{\partial F}{\partial x}(P), \frac{\partial F}{\partial y}(P), \frac{\partial F}{\partial z}(P) \right)$$

Questo vettore è normale alla superficie in quel punto. Successivamente si trova una base per il piano tangente alla superficie, che è ortogonale al gradiente ∇F trovando due vettori indipendenti nel piano tangente che siano ortogonali a ∇F . Si forma poi la matrice Q , matrice di proiezione dell'Hessiano sul piano tangente, usando questi due vettori come colonne. La matrice Q sarà una matrice 3×2 . Successivamente si calcola l'Hessiano

$$H(F) = \begin{bmatrix} \frac{\partial F}{\partial x^2} & \frac{\partial F}{\partial x \partial y} & \frac{\partial F}{\partial x \partial z} \\ \frac{\partial F}{\partial y \partial x} & \frac{\partial F}{\partial y^2} & \frac{\partial F}{\partial y \partial z} \\ \frac{\partial F}{\partial z \partial x} & \frac{\partial F}{\partial z \partial y} & \frac{\partial F}{\partial z^2} \end{bmatrix}$$

e lo si valuta nel punto P . Si calcola l'Hessiano tangente H_{tangente} come $Q^T H Q$, dove H è l'Hessiano di F . Questo prodotto ci dà la matrice 2×2 che rappresenta l'Hessiano ristretto al piano tangente. In fine si calcolano gli autovalori di H_{tangente} che sono le curvatures principali della superficie nel punto considerato.

Questo procedimento è sempre valido per il calcolo della curvatura in un punto ma diventa estremamente complesso validare l'intera superficie: il procedimento richiederebbe la definizioni di **Equazioni alle Derivate Parziali** (EDP) la cui costruzione dipende dall'equazione stessa della superficie. Oltre questo la risoluzione delle EDP non lineari è uno dei rami più complessi della matematica e spesso richiede approcci computazionali perché risulta troppo complicato trovare le soluzioni analiticamente.

Per quanto riguarda la costruzione matematica dei solidi che hanno come base superfici TPMS questi possono essere realizzati principalmente seguendo due logiche: superficie Skeletal e superficie Double. Tutto sta nel come si manipola l'isovalore della superficie TPMS: prendendo la funzione dell'equazione implicita $F(x, y, z) = c$ si cambia il valore di c in funzione del tipo di struttura che si vuole ottenere. La Figura 1.4 rappresenta i processi di costruzione della geometria della cella unitaria. Come esempio di geometria di base è stata utilizzata la superficie Schwarz P. Nell'approccio skeletal alla geometria, si sceglie un isovalore che genera una isosuperficie che viene combinata con dei confini (boundary) per isolare uno dei due volumi. Questo processo dà vita a una struttura solida con una superficie esterna che corrisponde alla TPMS originale quando $c = 0$. Per valori di $c < 0$, la struttura risultante è più esile, mentre per $c > 0$, si ottiene una forma più robusta, sebbene questa variante sia meno comune. Nel-

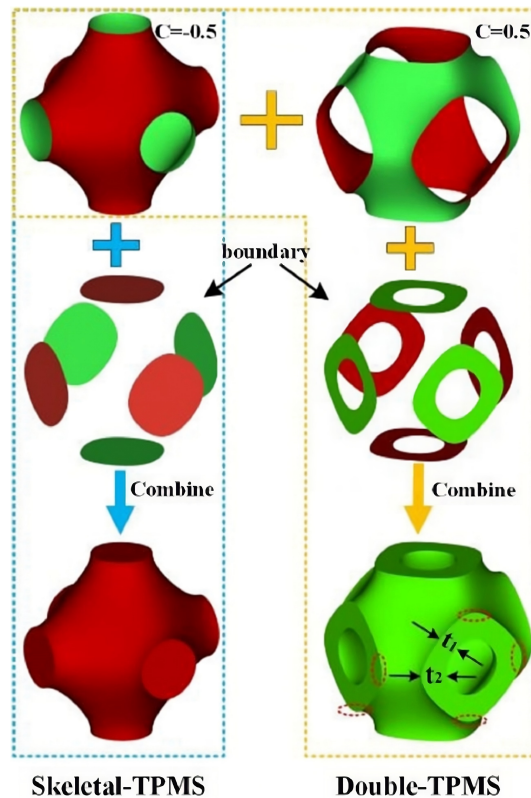


Figura 1.4: Due metodi tipici di modellazione delle unità TPMS [1]

l'approccio con geometria double, si procede a creare un ispessimento della TPMS originale: si scelgono due isovalori, generalmente opposti, per formare due superfici distinte. Queste due superfici, che racchiudono la TPMS iniziale, vengono poi chiuse per generare un volume.

1.3 Generalità sui processi di AM e processo LPBF

L'Additive Manufacturing, noto anche come fabbricazione additiva, rappresenta una frontiera innovativa nel campo della produzione industriale. L'AM è definita come "Processo di giunzione di materiale per costruire un oggetto 3D partendo dal modello CAD, generalmente strato su strato, in opposizione alle tradizionali tecnologie di manifattura sottrattiva". Questo perché si distingue per la sua capacità di creare oggetti aggiungendo materiale dove precedentemente assente, metodologia in netta contrapposizione ai classici metodi sottrattivi tradizionali, i quali rimuovono il materiale da un blocco pieno. La natura intrinsecamente flessibile dell'Additive Manufacturing si traduce in un elevatissimo grado di personalizzazione, consentendo la realizzazione di prodotti che si adattano perfettamente alle esigenze specifiche di ogni cliente, come nel caso delle protesi mediche o dei componenti aerospaziali.

1.3.1 Aspetti generali

Una delle caratteristiche più sorprendenti di questa tecnologia è la sua capacità di produrre geometrie complesse senza un incremento significativo dei costi. Questo aspetto si rivela particolarmente vantaggioso quando si considerano le forme complesse che sono spesso richieste in settori come l'aerospaziale e il biomedicale. Soprattutto spesso queste forme complesse sono state ottenute andando a ottimizzare la distribuzione del materiale nel componente attraverso processi di ottimizzazione topologica ottenendo una parte dalle prestazioni superiori. Inoltre, l'Additive Manufacturing promuove la sostenibilità attraverso la riduzione degli sprechi materiale, poiché esso viene depositato solo dove è necessario, e attraverso la completa riciclabilità dello stesso nella quasi totalità dei casi (sia che esso sia polimerico o metallino). La velocità con cui è possibile passare dal design alla produzione finale è un altro punto di forza dell'Additive Manufacturing. Questa rapidità è fondamentale per accorciare i cicli di sviluppo dei prodotti, permettendo alle aziende di rispondere prontamente alle esigenze del mercato con prototipi e parti finali realizzati in tempi brevi.

Oltre questo l'Additive Manufacturing risulta essere anche una tecnologia interamente legata alla digitalizzazione: per la realizzazione del componente abbiamo necessariamente bisogno del modello CAD del prodotto dal quale devono essere ricavate le caratteristiche geometriche di produzione. In particolare gli step che si seguono sono:

1. Il prototipo è disegnato al CAD, qui si definisce la forma che deve possedere il prodotto e solitamente si fa uso di metodi non convenzionali e che fanno uso di capacità di calcolo considerevoli come l'ottimizzazione topologica la distribuzione del materiale in maniera ottimale nel componente concordemente con la funzione che dovrà svolgere, oppure la realizzazione di componenti alleggeriti tramite un infill di geometria e distribuzione più o meno variabile. Tutto questo viene poi esportato in un file di superficie STL che prende il nome dalla prima tecnologia di AM mai creata ossia la Stereolitografia.
2. Il modello STL è elaborato dalla macchina per definire l'orientamento del pezzo, la direzione di accrescimento, i supporti. Questi sono tutti parametri preliminari fondamentali per la riuscita del processo perché definiscono se la geometria si troverà in sospensione durante il processo di creazione, quanto potrà quindi essere soggetta a deformazioni legate alla gravità o al surriscaldamento influenzando anche sul tempo di realizzazione.
3. Il modello virtuale è sottoposto allo slicing ossia la divisione in strati, che definirà poi lo spessore del materiale consolidato ad ogni layer. In questa fase si determinano anche:
 - precisione dimensionale (data dallo spessore degli strati e dal percorso della testa di deposizione)
 - Parametri di processo quali la potenza di erogazione della fonte di energia che porta al consolidamento oppure la velocità di movimento della traccia che disegna la geometria
 - Tempi e costi di produzione in maniera indiretta dato che tutte le specifiche sopracitate (e non solo) influenzano in maniera significativa la produttività
4. Fase di generazione del pezzo, in cui i parametri di processo vengono monitorati e viene valutato se i parametri di processo erano corretti. Per pezzi complessi, dipendentemente anche dal tipo di tecnologia che si utilizza, potrebbero essere necessarie alcune prove di stampa.
5. Termine del processo di stampa, rimozione degli elementi stampati dal piatto (con materiali metallici solitamente dopo un trattamento termico distensivo per evitare la deformazione del componente dovuta a tensioni residue). Eventuale post-trattamento per migliorare caratteristiche meccaniche ed estetiche come l'aumento di finitura superficiale (fattore che influenza sia la componente estetica che meccanica) oppure trattamenti di invecchiamento.

Nonostante i numerosi vantaggi, l'Additive Manufacturing presenta anche alcune sfide. La qualità superficiale delle parti prodotte può essere inferiore rispetto a quella ottenuta con metodi tradizionali, e questo, come detto, può richiedere ulteriori lavorazioni post-produzione (quindi è una problematica ovviabile ma non senza un

incremento dei costi). Inoltre, le dimensioni delle parti sono spesso limitate dalle dimensioni della macchina di stampa, e i costi dei materiali possono essere elevati, soprattutto per quanto riguarda quelli metallici, perché devono essere di qualità elevata per ottenere componenti avanzati e performanti.

Tra le tecnologie più note di Additive Manufacturing vi sono la stereolitografia (SLA), che utilizza un laser per polimerizzare resine fotosensibili, e la sinterizzazione/fusione selettiva a laser (SLS per i polimeri o SLM per i metalli), che fonde polveri plastiche o metalliche per creare parti solide. Queste tecnologie offrono diversi gradi di precisione e resistenza dei materiali, rendendole adatte a una varietà di applicazioni industriali.

1.3.2 Laser Powder Bed Fusion

Le tecnologie di fabbricazione additiva che si basano sulla strategia LPBF realizzano la fusione layer by layer di particelle di polvere, precedentemente depositate a formare un "letto", attraverso l'utilizzo di un laser ad elevata densità di energia. pertanto, il letto di polvere costituisce il piano x-y che viene scansionato selettivamente: depositando layer successivi di polvere si ottiene la costruzione del pezzo lungo l'asse z come è possibile visualizzare in Figura Figura 1.5

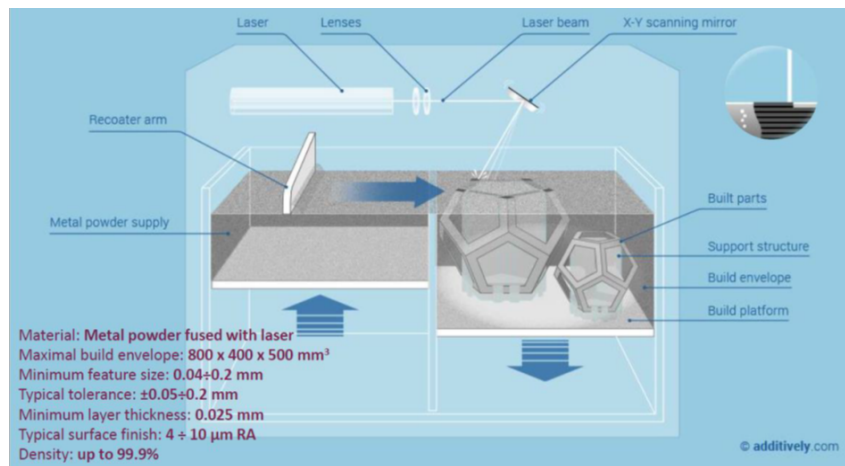


Figura 1.5: Schema costruttivo architettura macchine LPBF

La camera di stampa viene mantenuta sotto controllo da un gas inerte che consente di proteggere il materiale da fenomeni di ossidazione a caldo e per estrarre i vapori metallici che si formano nell'interazione tra laser e polvere che potrebbero rimanere in sospensione tra la testa e il punto del piano che sta venendo colpito oppure anche andare a depositarsi ad esempio sull'ugello del laser riducendone di fatto la potenza utile. I principali parametri di processo relativi alla tecnologia LPBF sono i seguenti:

- Specifiche e caratteristiche della polvere: la morfologia, lo shape factor ($\frac{S}{V}$, dove S = superficie e V = volume) e l'aspect ratio ($\frac{D_{MAX}}{d_{min}}$, dove D_{MAX} =

dimensione massima e d_{min} = dimensione minima) della polvere influiscono notevolmente sulla capacità di scorrere e di compattarsi della polvere formando un letto di stampa consono alla realizzazione di componenti conformi;

- Materiale lama del recoater: per il materiale della spatola, è necessario valutare un compromesso tra vari elementi come la densità ottenuta, il prezzo della spatola e i fenomeni che si verificano durante l'impatto con componenti emergenti sulla superficie di stampa. Le spatole in silicone hanno un costo vantaggioso e distribuiscono il materiale in maniera omogenea, tuttavia, risultano in una minore densificazione della polvere sulla superficie di stampa. Al contrario, le spatole in metallo producono una notevole pressione di densificazione (riducendo così lo spazio vuoto tra le particelle di polvere), il che favorisce la creazione di strutture a griglia e a cellule: questo è possibile perché sono estremamente flessibili e sottili, quindi con l'incremento della densificazione della superficie di stampa si ottiene una maggiore stabilità strutturale;
- Disposizione e allineamento dei pezzi sulla base di stampa: scegliere l'orientamento adeguato e la giusta collocazione previene la comparsa di inconvenienti nella fabbricazione dei componenti durante il processo di stampa, influenzando la consistenza del procedimento e le caratteristiche dei pezzi prodotti. È di particolare rilievo considerare come i livelli già stampati interagiscono con il recoater: nel momento in cui la spatola di copertura si muove verso l'area precedentemente fusa, se l'ampiezza della sezione fusa è considerevole, la pressione esercitata dal recoater può causare il distacco del componente dalla base di stampa, compromettendo l'esito dell'intera operazione di stampa. Per questo motivo bisogna cercare di posizionare i componenti all'interno dello stesso volume in modo tale che un eventuale impatto con il recoater sia graduale tra i componenti e comporti la minima deformazione con il singolo componente. Esempi sono mostrati in Figura Figura 1.6;
- Controllo del livello di ossigeno: mantenere un tenore ridotto di ossigeno è fondamentale durante la fase di stampa, poiché livelli elevati possono causare ossidazione durante il riscaldamento. Per abbassare la concentrazione di ossigeno, si ricorre all'impiego di gas nobili come l'azoto (N) e l'argon (Ar). L'uso di questi gas non solo previene l'ossidazione ma facilita anche la rimozione dei vapori metallici prodotti dall'interazione del laser con il letto di polvere, che altrimenti si depositerebbero all'interno dell'ambiente di stampa;
- La temperatura del piatto di stampa: è essenziale riscaldare la base entro un intervallo termico di 100°-200° perché, durante la deposizione dei primi strati, una base a temperatura troppo bassa causerebbe lo scorrimento del metallo liquido, portando a un fallimento anticipato dell'operazione di stampa. Oltre

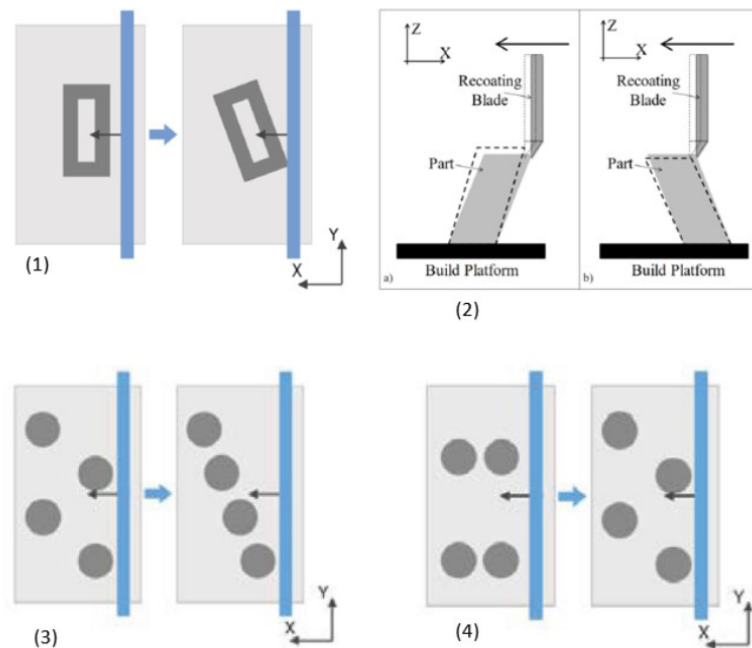


Figura 1.6: strategie per il posizionamento dei pezzi sul job di stampa. (1) posizionamento angolato rispetto al recoater per minimizzare le forze di impatto, (2) evitare zone in cui il recoater possa incagliarsi e sollecitare a flessione la parte, (3) far incontrare al recoater contemporaneamente meno parti possibile, (4) evitare di posizionare i pezzi uno dietro l'altro

questo un piatto riscaldato diminuisce la velocità di raffreddamento nei primi strati diminuendo anche le distorsioni;

- Materiale del piatto di stampa: la composizione del piatto di stampa deve corrispondere o essere altamente compatibile con quella della polvere da lavorare (per esempio, per le leghe di nichel si utilizza un piatto in acciaio per la similitudine termica). Questo perché è essenziale che la polvere e la superficie di stampa reagiscano in modo analogo alle variazioni di temperatura, garantendo così una buona adesione dei primi strati e prevenendo il rischio di deformazioni durante il processo di stampa;
- Strategia di scansione: la scelta di un'adeguata strategia di scansione è cruciale per minimizzare gli stress residui e per l'omogeneità dei pezzi prodotti. Per mitigare gli stress residui, si costruiscono strutture di supporto e si limita la lunghezza del percorso del laser; queste strutture di supporto sono essenziali anche per prevenire il cedimento di elementi con un'inclinazione superiore ai 45°. Per contrastare l'effetto dell'anisotropia causato dall'orientamento specifico delle linee mappate, si alterna l'angolazione di mappatura di 67° tra i diversi strati, come illustrato nella Figura Figura 1.7. Un ulteriore fattore significativo nella scelta della metodologia di mappatura è la distanza di sovrapposizione

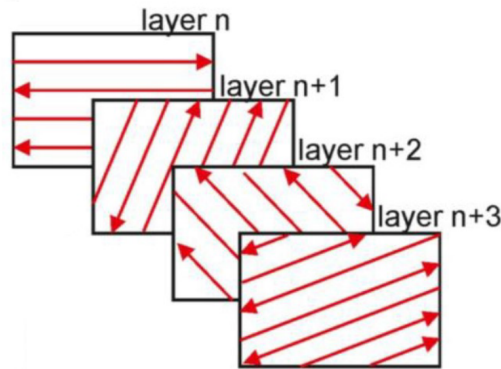


Figura 1.7: strategia di scansione per la riduzione dell'anisotropia del pezzo stampato con rotazione della direzione di scansione di 67° ogni layer

(hatch spacing), che indica il livello di copertura tra le linee mappate consecutive. Questo parametro varia in base alla densità desiderata del materiale e al grado accettabile di porosità;

- Spessore del layer: lo spessore del layer ha un impatto diretto sulla durata del processo di stampa e sulla texture superficiale dell'oggetto. In dettaglio, un incremento nella dimensione del livello porta a un miglioramento dell'efficienza produttiva, ma comporta una qualità superficiale inferiore. Al contrario, diminuendo la dimensione del livello, si verifica una riduzione dell'efficienza produttiva, ma si ottiene una migliore qualità nella finitura superficiale;
- Parametri del laser: Le variabili modificabili nelle impostazioni del laser includono la potenza e la dimensione dello spot. Queste sono strettamente correlate alla velocità di scansione; esiste una correlazione tale per cui, se si incrementa la potenza del laser, è necessario anche aumentare la velocità di scansione per evitare di danneggiare il metallo o di generare porosità nell'elemento prodotto.

I principali parametri di processo che vanno settati nel processo di stampa additiva LPBF possono essere riassunti con la VED (densità di energia volumetrica):

$$VED = \frac{P}{vht}$$

Dove P = potenza del fascio laser, v = velocità di scansione, h = hatch spacing e t = spessore del layer. Occorre trovare il giusto trade-off di questi parametri per poter realizzare delle tracce continue senza difetti (si mostrano gli effetti in Figura 1.8).

Va specificato che sebbene si possano ottimizzare i parametri di processo, ci sarà sempre una certa deviazione tra la geometria progettata su CAD e i provini realizzati in additive manufacturing. Si può visualizzare ciò andando a osservare la deviazione della geometria del modello CAD con quella effettivamente ottenuta con il processo di stampa come riportato da J.A. Harris e G.J. McShane [37].

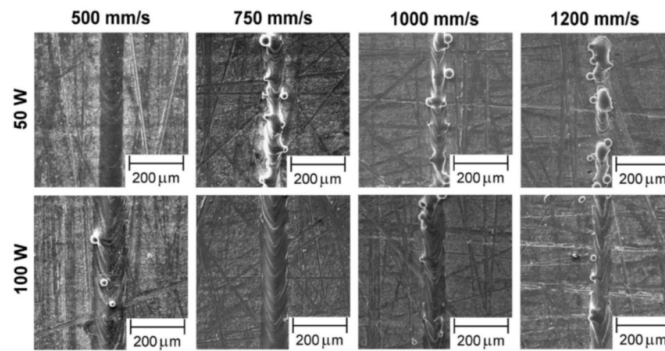


Figura 1.8: analisi dell'influenza della potenza del laser e della velocità di scansione, a parità di hatch spacing e spessore del layer, sulla realizzazione di tracce esenti da difetti

Facendo riferimento ad alcune ricerche specifiche, Yan et al hanno discusso la fabbricabilità dei TPMS utilizzando la tecnologia SLM con polvere di acciaio inossidabile 316 l [38]. Secondo i risultati sperimentali, tutte le dimensioni delle unità che vanno da 2 a 8 mm possono essere fabbricate tramite SLM senza difetti e strutture di supporto aggiuntive. Tuttavia, come mostrato nella figura Figura 1.9, ci sono molte particelle legate sulle superfici. La rugosità superficiale è aumentata da queste particelle legate. Fan et al hanno fabbricato TPMS graduati tramite SLM con polveri di Ti6Al4V [39].

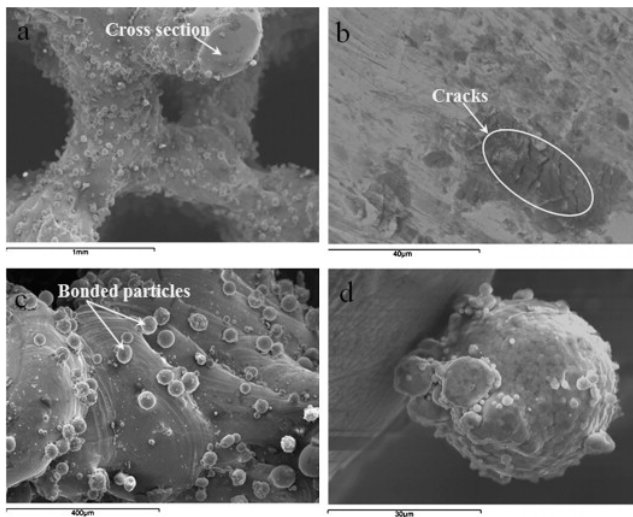


Figura 1.9: Micrografie SEM di (a) le travi della struttura a reticolo cellulare giroide con dimensione della cella di 2 mm e frazione volumetrica del 15% realizzata mediante il processo SLM, (b) la sezione trasversale di una trave, (c) particelle legate alle superfici delle travi e (d) una particella individuale sulla superficie della trave.

Esistono anche polveri semi-fuse sulle superfici. Rispetto alle strutture con spessore della parete uniforme, le pareti più sottili delle strutture graduate possono fornire meno supporto. I difetti sono diminuiti gradualmente lungo la direzione del gradiente. Pertanto, gli errori di spessore delle strutture graduate sono maggiori. Per le strutture TPMS porose, le maggiori deviazioni si troveranno sulle pareti interne superiori dei pori; inoltre, per migliorare ulteriormente la rugosità superficiale, il metodo sabbato è efficace per ridurre l'influenza dei gradienti e delle sovrapposizioni o delle particelle legate [40].

Capitolo 2

Strumenti e Metodi

2.1 Introduzione

Per la realizzazione di architetture complesse e avveniristiche c'è bisogno di sistemi che siano in grado di realizzare questi modelli. La geometria computazionale è una disciplina affascinante che si colloca all'incrocio tra matematica e informatica, dedicata allo studio e alla creazione di algoritmi per risolvere problemi geometrici. Questo campo si avvale di concetti provenienti da diverse aree come l'algebra lineare e la topologia, per sviluppare soluzioni efficienti e precise.

Uno dei principali vantaggi della geometria computazionale è la sua capacità di trattare problemi complessi in modo efficiente, garantendo un uso ottimale delle risorse di calcolo. La precisione è un altro punto di forza, poiché gli algoritmi sono progettati per ridurre al minimo gli errori di arrotondamento, che sono particolarmente critici in operazioni come la divisione e l'utilizzo di funzioni trigonometriche. Inoltre, la vasta applicabilità di questa disciplina la rende preziosa in numerosi campi, dalla robotica [41] alla grafica computerizzata, dai sistemi di informazione geografica alla progettazione assistita da computer.

Tuttavia, la geometria computazionale presenta anche delle sfide. La complessità di alcuni problemi può richiedere una profonda comprensione sia matematica che informatica. Inoltre, le limitazioni dell'hardware, come la precisione e la capacità di memoria, possono influenzare la rappresentazione e l'elaborazione degli oggetti geometrici. Un altro potenziale ostacolo è il rischio di overflow, specialmente quando si lavora con valori molto grandi o piccoli, che richiede una gestione attenta dei limiti numerici. Nonostante queste sfide, l'utilità della geometria computazionale è indiscutibile.

2.2 La scelta dell'ambiente di modellazione

Definito il campo di ricerca adesso si aveva la necessità di trovare l'ambiente di modellazione più adatto per la realizzazione delle TPMS. Un buon metodo di modellazione per la realizzazione di geometrie che possono essere descritte analiticamente è l'utilizzo delle Signed Distance Function ("Funzioni di distanza con

segno" o SDF): In matematica e nelle sue applicazioni, la funzione di distanza con segno (o funzione di distanza orientata) è la distanza ortogonale di un dato punto \boldsymbol{x} al confine di un insieme Ω in uno spazio metrico, con il segno determinato dal fatto che \boldsymbol{x} sia o meno all'interno di Ω . La funzione assume valori positivi nei punti \boldsymbol{x} all'interno di Ω , diminuisce di valore man mano che \boldsymbol{x} si avvicina al confine di Ω dove la funzione di distanza con segno è zero, e assume valori negativi all'esterno di Ω . Un esempio in R^3 è mostrato in Figura 2.1. Possiamo osservare in alto che la forma del dominio Ω (in grigio) è un cerchio, quindi l'equazione che lo descrive in questo caso è del tipo

$$x^2 + y^2 = R^2$$

Se descriviamo Ω sul piano xy (in blu) si osserva come, per $z = R$, questa altro non sia che la curva di isolivello di una **superficie cartesiana** di equazione:

$$x^2 + y^2 = z^2$$

ossia un cono (mostrato in rosso). Se noi applicassimo l'equazione del dominio a ogni punto P del piano xy a quota $z = R$, ossia la funzione $f(x, y) = x^2 + y^2 - R^2$, avremmo 3 situazioni:

- $f(P) = 0$: P appartiene alla curva di isolivello $z = R$ ed è quindi parte del **bordo** o **frontiera** di Ω ;
- $f(P) > 0$: P si trova a una distanza dal centro del dominio (ossia l'origine $O = (0, 0)$) maggiore del raggio R ($x^2 + y^2 > R^2$), quindi è un punto esterno al cerchio;
- $f(P) < 0$: P si trova a una distanza dal centro del dominio minore del raggio ($x^2 + y^2 < R^2$), quindi è un punto interno a cerchio.

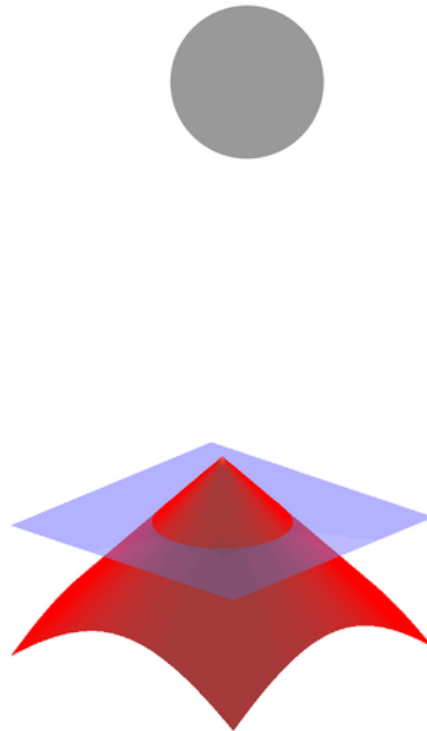


Figura 2.1: Il grafico (in basso, in rosso) della distanza con segno tra i punti del piano xy (in blu) e un disco fisso (rappresentato anche in alto, in grigio)

In questo modo è intuitivo capire che abbiamo definito due domini, uno interno e uno esterno, e quindi abbiamo rappresentato il cerchio come oggetto "fisico". La definizione matematica di una SDF, in realtà, non è esattamente questa: una SDF

per definizione dovrebbe dare come risultato la **distanza** con segno dal piano, cosa che l'esempio precedente non fa, il valore da solo un'idea di dove è collocato. Questo per la parte applicativa è più che sufficiente: infatti se andiamo a estendere il caso nello spazio \mathbb{R}^4 considerando un dominio Ω di 3 dimensioni (lo spazio xyz) e non più un piano, considerando poi una funzione di $F(x, y, z)$ e un valore costante C , ci troviamo proprio nelle condizioni di rappresentazione di una superficie TPMS in forma implicita con l'equazione già vista precedentemente, che adesso assume un significato geometrico:

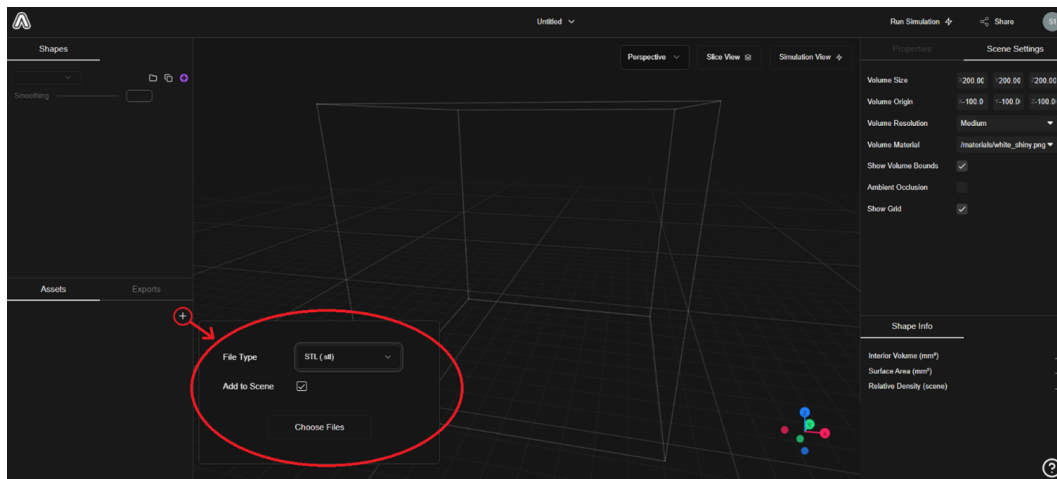
$$F(x, y, z) = C$$

Dell'utilizzo di questa modellazione ne verrà parlato più nel dettaglio in seguito.

I due software presi in considerazione, per trattare questo tipo di modellazione, sono stati Metafold 3D e PicoGK. Essi hanno l'obiettivo comune di offrire un servizio di computazione efficiente per l'ottimizzazione di componenti tramite strutture reticolari volti alla successiva realizzazione tramite manifattura additiva. Il software Metafold 3D presenta una web interface con cui è possibile eseguire le proprie ottimizzazioni con semplicità e praticità: per un completo utilizzo necessita di un abbonamento ma la versione gratuita offre le stesse funzionalità concedendo un numero di interazioni limitato (numero di progetti attivi nel proprio portfolio, progetti esportati o simulazioni avviate). Oltre questo offre la possibilità di realizzare internamente al programma superfici SDF tramite codice e un'API in Python per un processamento più rapido di numerose analisi. PicoGK invece è un programma opensource ora detenuto da Leap71: in questo caso non si ha una user interface e il lavoro viene realizzato tramite codice C# consentendo di lavorare con oggetti di vario tipo come mesh, volumi voxelizzati o superfici SDF e analizzare visivamente il risultato sul loro visualizzatore.

2.2.1 Metafold 3D

L'editor si presenta classicamente come uno spazio 3D con un volume di lavoro all'interno del quale le geometrie verranno computate. L'opzione più semplice per iniziare è quella di realizzare un modello su un software esterno e importarlo:



Per lo scopo di questa spiegazione si è deciso di utilizzare più componenti per osservare il comportamento del software.

Esistono due principali tipologie di oggetti in Metafold3D: da una parte si hanno le primitive, ossia i modelli di lavoro che posso esse importati oppure realizzati direttamente all'interno; dall'altra si hanno le strutture reticolari (lattice) che generalmente vengono ripetute per tutto il volume di lavoro. Queste possono interagire tra loro nelle 3 operazioni booleane fondamentali: unione, intersezione e sottrazione. Come informazione degna di nota esiste anche un tipo di superficie chiamata Custom Shape che consente di generare geometrie più complesse tramite codice, utilizzando le Signed Distance Function (SDF). Parlando delle strutture reticolari le macro tipologie sono le strutture a raggi (beam lattice o BL) e le TPMS (Triply Periodic Minimal Surface) che invece sfruttano le superfici minime, superfici matematiche in cui ogni punto ha curvatura media pari a 0. All'interno di ogni famiglia si può poi selezionare una cella elementare specifica, ad esempio la cubica a corpo centrato per le beam o la giroide per le TPMS. Inoltre è possibile importare celle beam personalizzate scritte in un file json. La scelta importante da fare prima di aggiungere al volume di lavoro una struttura reticolare è la matematica con la quale verrà generata: sia per le beam che le TPMS si hanno diverse opzioni di generazione che verranno esplorate nel dettaglio.

Basic

Queste non hanno nessuna dicitura particolare, riempiono semplicemente il volume di lavoro seguendo un reticolo cartesiano ortogonale. Le opzioni di personalizzazione in questo caso riguardano semplicemente il tipo di cella, il posizionamento del reticolo, la sua orientazione e la dimensione della cella. La metodologia più classica di lavoro è quella di intersecare in maniera booleana il volume del reticolo con una mesh desiderata, applicando di fatto la struttura reticolare al componente.

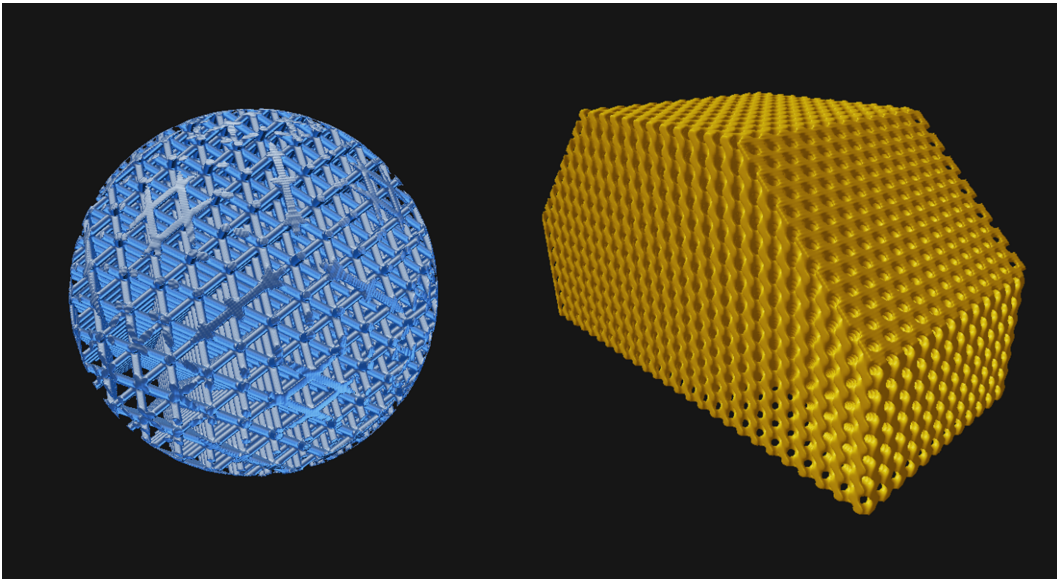


Figura 2.2: Lattice Strut e TPMS di base

Cylindrical/Spherical

In queste strutture il riferimento cartesiano è rispettivamente cilindrico o sferico. Così si cerca di sfruttare le potenzialità dell'ingegneria computazionale, deformando la cella per seguire un determinato pattern, differente da come la cella era stata concepita inizialmente. I parametri a disposizione sono come i precedenti, ovviamente con valori diversi in quanto si utilizza un sistema di coordinate diverso. E' interessante notare come la cella si meglio adatti alla struttura (particolarmente evidente dal confronto delle sfere)

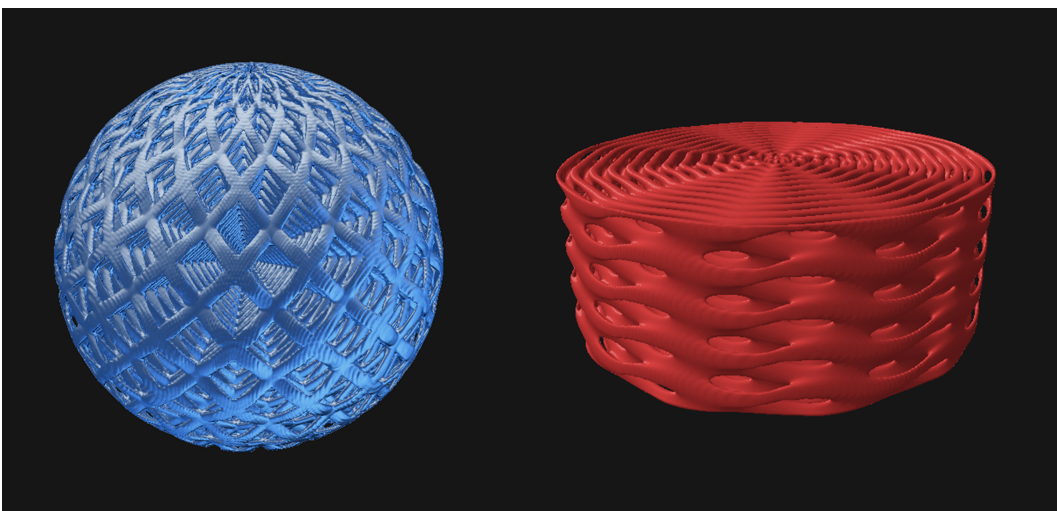


Figura 2.3: Conformazione sferica e cilindrica

Cell Size/Material Grading

Nelle varianti Grading si può modificare la gradazione di una grandezza seguendo una geometria specifica. Le opzioni sono, come deducibile dai nomi,

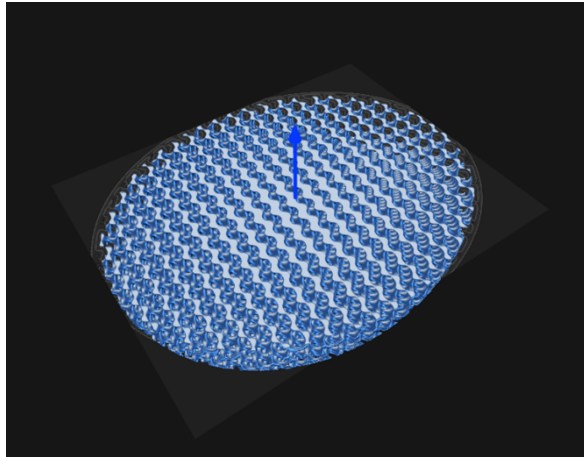


Figura 2.4: Grading su geometria ellittica

la dimensione della cella oppure il materiale (lo spessore del materiale della cella). In questo caso si porta il concetto stesso delle strutture reticolari su un livello più alto: poter regolare la distribuzione di materiale all'interno del componente significa poter ottimizzare la distribuzione della struttura. I parametri controllabili sono (oltre a quelli già menzionati) la forma di base da dare al gradiente, la sua posizione e orientazione nel sistema di riferimento, la sua grandezza caratteristica e la distribuzione del gradiente.

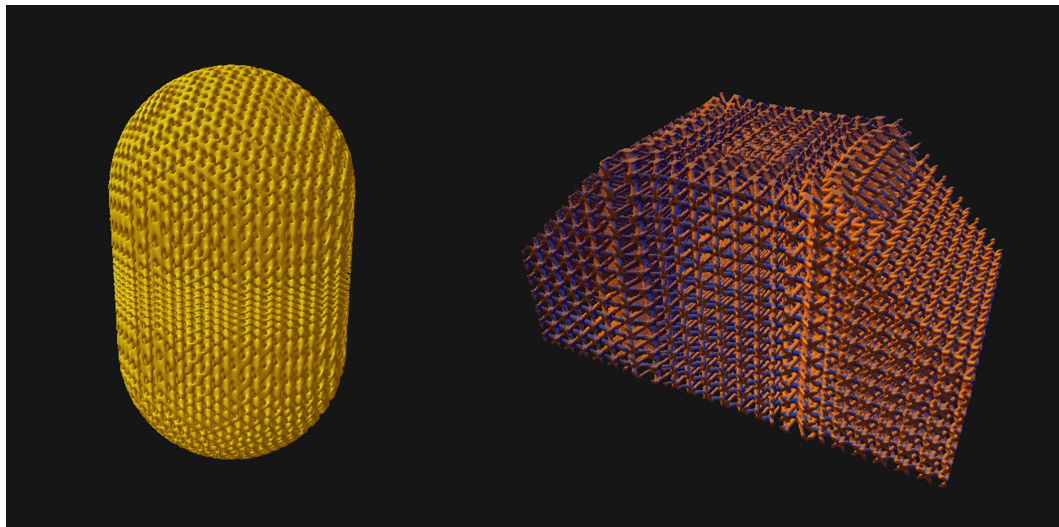


Figura 2.5: Grading di materiale

Conformal

Infine con l'ultima opzione si usa pienamente l'ingegneria computazionale eseguendo il Conformal Mapping. L'idea è quella di utilizzare la geometria del modello di cui si vuole generare l'infill come ambiente di modellazione per la disposizione spaziale della cella elementare, generando un reticolo ben adattato alla mesh di riferimento e che segue la sua forma. Per eseguire correttamente questa operazione

bisogna prima generare una parametrizzazione dell'asset desiderato. Ovviamente per fare questo non è obbligatorio utilizzare lo stesso oggetto come base, infatti è possibile utilizzarne una versione semplificata per rendere il calcolo più semplice, oppure parziale se non si vuole generare infill per tutto l'oggetto. Dopo aver generato la parametrizzazione basta aggiungere la mesh conformal (beam o surface) e selezionarla nella voce "Parametrization" tra quelle disponibili. Come esempio è stato realizzato un componente di forma più complessa rispetto alle primitive: questo sia per mostrare come il conformal mapping funzioni sia perché il software non crea una parametrizzazione se il modello è troppo semplice (in tal caso devono essere utilizzati gli altri metodi).

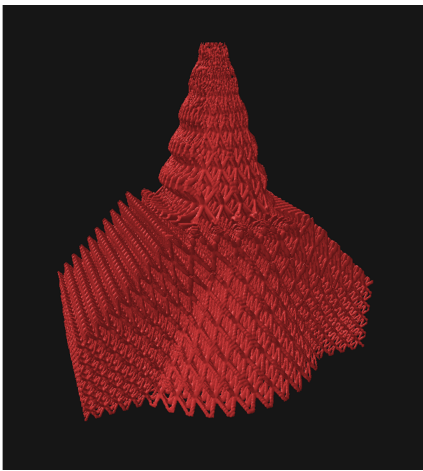


Figura 2.6: Geometria Strut Conformal

All'interno di ogni mesh ci sono anche le opzioni di redistance e shell: la prima serve per modificare la posizione della superficie esterna spostandola di un certo offset; la seconda serve (usata dopo la prima) per svuotare l'interno di una mesh e trattarla come un corpo cavo. La principale utilità di queste due funzioni è quella di poter generare un corpo al cui interno viene generato l'infill, opzione non normalmente disponibile con le operazioni booleane sopracitate. Di particolare utilità è l'opzione conformal che consente di unire la mesh cava con l'infill conformal (nelle altre strutture lattice si lavora per intersezione solitamente).

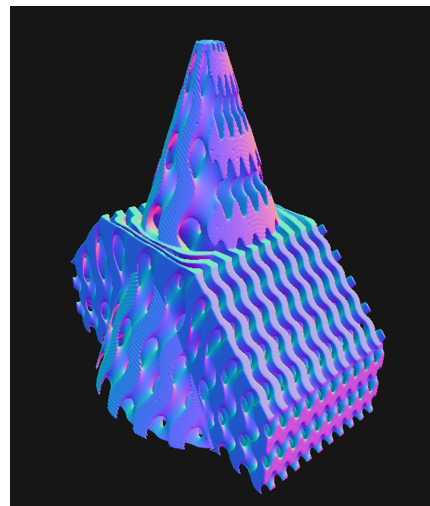


Figura 2.7: Geometria Giroide Conformal

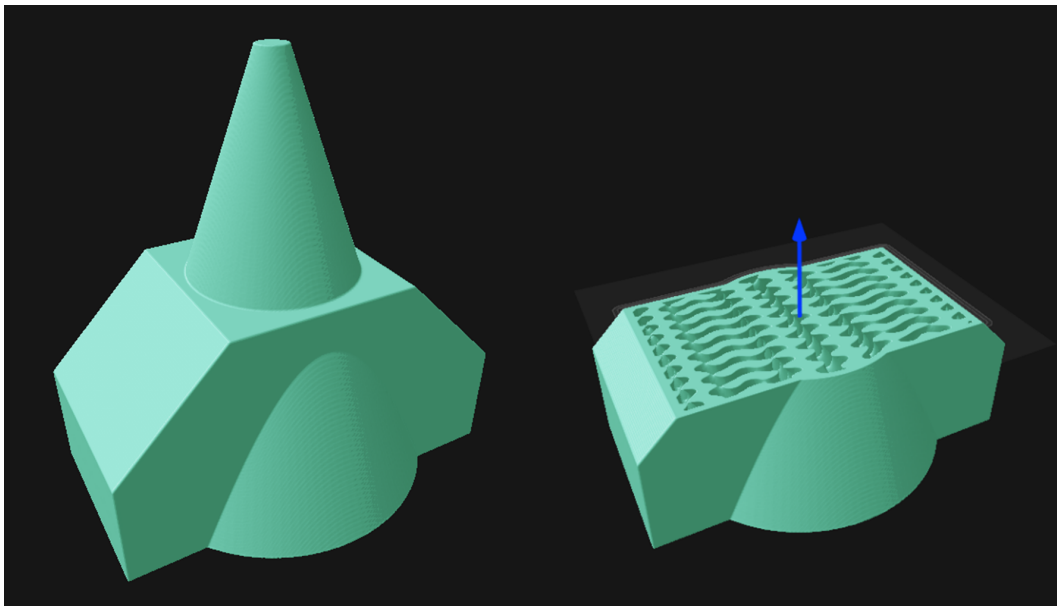


Figura 2.8: Geometria Giroide come infill con il comando shell

Custom Shape

Un oggetto molto utile della categoria primitive è Custom Shape: questo consente di realizzare tramite codice delle geometrie interpretando il risultato che la funzione principale del codice (chiamata semplicemente "surface") restituisce. In particolare la funzione viene applicata ad ogni punto del volume di lavoro e se il risultato è un numero negativo allora il punto apparterrà al volume solido, se invece il risultato è positivo allora il punto apparterrà alla frazione di vuoto (0 sarebbe la superficie dell'oggetto).

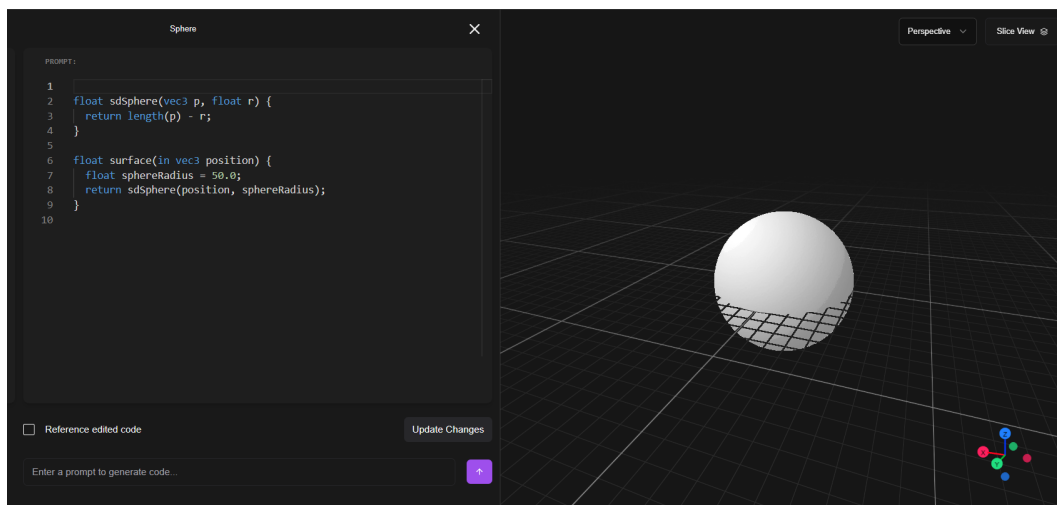


Figura 2.9: Sfera Custom Shape di raggio 50 centrata nell'origine

Nella figura Figura 2.9 possiamo vedere come la funzione principale "surface" che prende in input un vettore tridimensionale chiamato posizione (questa è una variabile dell'ambiente di modellazione che contiene le coordinate del generico punto) e dopo aver memorizzato nella variabile "sphereRadius" il raggio desiderato della sfera restituisce il valore della funzione definita sopra, ossia "sdSphere" dando come valori la posizione del punto e il raggio (come argomenti posizionali ossia position diventa p e sphereRadius diventa r); analizzando la funzione "sdSphere" vediamo che si comporta in modo simile alle SDF definite nella sezione 2 di questo capitolo: questa restituisce il valore della lunghezza del segmento tra punto p e l'origine, in pratica la distanza cartesiana, e la confronta con il raggio. Combinando queste funzioni con il funzionamento della "Custom Shape" definita prima, otteniamo una sfera solida di raggio 50 centrata nell'origine.

Altro aspetto interessante del software Metafold 3D è quello relativo alle simulazioni: per il momento si ha la possibilità di eseguire solo la simulazione di compressione per basse velocità di deformazione (questo anche un numero limitato di volte a seconda del proprio piano di utilizzo) ma, come già detto, il software è ancora in fase di sviluppo quindi a breve potrebbero presentarsi nuove possibilità. Infine è possibile esportare tutti gli oggetti visualizzati nel volume di lavoro, per il formato sono offerte diverse opzioni: si può esportare direttamente la mesh risultante (in formato STL, OBJ o PLY), il volume (VDB) oppure in stack di immagini in alta risoluzione (sarebbero tutte sezioni del modello). Quello che viene mostrato nel visualizzatore è solo un'anteprima: il prodotto finale avrà una qualità nettamente superiore.

2.2.2 PicoGK

PicoGK è un software opensource adesso detenuto dalla Leap71 che consente di approcciarsi all'ingegneria computazionale direttamente via codice. I programmi realizzati sono in codice C# mentre il kernel (il centro delle operazioni) è scritto in C++. Essendo un software opensource si ha anche la possibilità di creare il proprio kernel ma non essendo questo l'obiettivo principale di questo lavoro si utilizzerà quello realizzato da Leap71 (oltre a quello nativo di base di PicoGK). In questo caso la realizzazione di un progetto risulta più complicata ma più puntuale: si ha la possibilità di realizzare una forma da zero (spesso anche semplice) che poi viene modellata, distorta, duplicata o riposizionata nello spazio per realizzare la geometria desiderata. Per lo scopo di questo progetto non si è approfondito questo aspetto ma solo la possibilità di generare infill tramite librerie dedicate. Il metodo più veloce per iniziare è installare PicoGK e utilizzare come base il file di esempio fornito, ovviamente è anche possibile creare un progetto da capo ma dato che necessita di una struttura organizzata di file (similmente ad altri progetti C#) è più semplice con una disposizione già data. A livello di codice si lavora sempre con oggetti: il

metodo di lavoro consigliato (o quantomeno quello utilizzato negli esempi) è quello di realizzare una classe con un nome identificativo, creando poi un metodo chiamato Task all'interno del quale si eseguono tutte le operazioni; successivamente questo metodo verrà chiamato nella funzione principale all'interno di un try-catch. In questo ambiente si lavora sostanzialmente con le mesh e i volumi. In genere le mesh sono oggetti di dimensioni finite mentre i volumi (quando generati) possono avere anche dimensioni infinite (devono essere successivamente circoscritti ad una zona specifica). Si ha la possibilità di realizzare molte geometrie ma a titolo dimostrativo è stato realizzato un cubo con una cavità sferica al suo interno con un infill di geometria giroide.

Dapprima si realizza un cubo di determinata dimensione e lo costruisce come mesh:

```
Mesh mshCube1 = Utils.mshCreateCube(new Vector3(mainCubeScale));
```

dove la variabile mainCubeScale è una grandezza caratteristica (impostata 100). Poi si crea una sfera proporzionale al cubo con un fattore di scala (centrata nell'origine)

```
IImplicit sdfSphere = new ImplicitSphere(new Vector3(0), mainCubeScale  
    * 0.3f);
```

Viene creata come una superficie implicita (Signed distance function, sdf). Poi si crea una zona di interesse (bounding box) e si trasforma la geometria implicita in un volume:

```
BBox3 oBBox = new BBox3(  
    new Vector3(-fWorkDimension),  
    new Vector3(fWorkDimension));  
  
Voxels voxelShape = new Voxels(sdfSphere, oBBox);
```

Si trasforma la mesh (il cubo) in un volume:

```
Voxels voxelCube1 = new Voxels(mshCube1);
```

E si effettua la sottrazione booleana per scavare il cubo con la sfera:

```
voxelCube1.BoolSubtract(voxelShape);
```

Successivamente si crea la geometria implicita giroide:

```
IImplicit sdfPattern = new ImplicitGyroid(6,1);
```

Dove il primo numero è la dimensione della cella unitaria mentre il secondo è lo spessore delle pareti. Poi si trasforma la geometria in un volume e la si interseca con l'oggetto di lavoro:

```
Voxels voxLattice = new Voxels(sdfPattern, oBBox);  
Voxels voxLatticeCube = Sh.voxIntersect(voxelCube1,voxLattice);
```

In fine si salva l'oggetto come un file stl:

```
voxLatticeCube.mshAsMesh().SaveToStlFile(Path.Combine  
(Library.strLogFolder, "LatticeCube.stl"));
```

Per concludere si crea un secondo cubo spostato dall'origine e si visualizza l'oggetto tagliandolo con questo per visualizzare la geometria interna:

```
Mesh mshCube2 = Utils.mshCreateCube(new Vector3(mainCubeScale/2),  
    new Vector3(mainCubeScale*0.25f));  
Voxels voxelCube2 = new Voxels(mshCube2);  
  
voxLatticeCube.BoolSubtract(voxelCube2);  
Library.oViewer().Add(voxLatticeCube);
```

Poi si chiama il metodo task della classe creata (chiamata in questo caso Lattice Cube) nella funzione principale, con le dovute librerie:

```
try  
{  
    PicoGK.Library.Go(  
        0.4f,  
        MyShapes.CubeLattice.Task,  
        strOutputFolder  
    );  
}  
catch (Exception e)  
{  
    Console.WriteLine("Failed to run Task.");  
    Console.WriteLine(e.ToString());  
}
```

PicoGK.Library.Go è il metodo che avvia il processo di computazione; 0.4f è la dimensione dei voxel per la creazione del volume; MyShapes.CubeLattice.Task è il percorso che porta al metodo Task dove si trovano le operazioni sopra descritte; strOutputFolder è il percorso file dato per salvare il file stl. Inoltre il tutto è messo

all'interno di un try che gestisce eventuali errori senza far interrompere bruscamente il programma (in caso di errori si finisce nel catch). In Figura 2.10 è mostrato il risultato dell'output

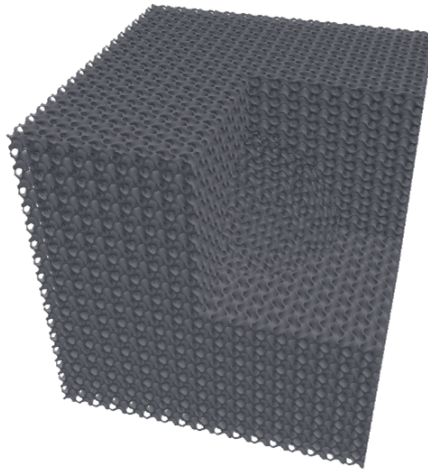


Figura 2.10: Modello di una giroide Tubular con frazione di volume del 20%

Facendo una valutazione complessiva si è deciso di usare Metafold3D come ambiente di modellazione: grazie alla sua interfaccia grafica risulta essere più user friendly e possiede comunque un modo per descrivere in maniera matematica la superfici tramite codice.

2.3 Altri software

Una volta modellati i provini si procederebbe alla loro simulazione. Questi oggetti verrebbero realizzati con la tecnologia Laser Powder Bed Fusion quindi c'è la necessità di simulare il loro comportamento nel processo di additive manufacturing dal punto di vista di dissipazione de calore durante il processo e quindi di eventuali deformazioni e tensioni residue che possono permanere alla fine di questo. Il passo successivo sarebbe quello di simulare i campioni meccanicamente per valutarne a priori le proprietà meccaniche, in rapporto alla loro densità. I software che possono essere utilizzati per questi scopi sono molti, uno molto potente a disposizione dell'Università Politecnica delle Marche è Ansys: questo software permette di fare molti tipi di simulazioni diverse tramite i suoi moduli implementativi ed è particolarmente apprezzato per le simulazioni meccaniche. Il problema principale riguarda la parte di simulazione di stampa: il modulo di processo di Ansys produce simulazioni molto pesanti (perché simula la parte termica in maniera dettagliata e conseguentemente la parte meccanica) richiedendo molti giorni di processamento; oltre a questo i dati ottenuti dal modulo termico sarebbero dovuti essere convertiti nel modulo meccanico per conservare le

informazioni sulle tensioni residue dei provini, altro processo complesso e oneroso in termini di tempo. Per questo motivo si è deciso di utilizzare un altro software con un modulo per le modellazioni termiche ossia Siemens NX. Questo avrebbe permesso di avere simulazioni (certamente un po' meno precise) ma più leggere. Questa scelta ha però portato un problema: utilizzare un software diverso per la simulazione del processo di stampa ci impone di dare ad Ansys un campione senza alcuno stress residuo. Se vogliamo considerare i campioni as built ("come costruiti", ossia usarli direttamente usciti dal processo di stampa, senza trattamenti termici ulteriori), cosa abbastanza comune in ambito di ricerca per capire quanto la forma sia compatibile con il processo, si deve dare la precedenza ai campioni che risultano avere i minori valori di stress residui per avere simulazioni il più precise possibili. Per la manipolazione dei campioni, una volta usciti dal software Metafold3D è stato utilizzato anche nTop, potente software di modellazione che consente di effettuare operazioni complesse tramite blocchi di codice logico.

2.3.1 Cenni sui software

In questa sezione si accenna alle capacità dei software utilizzati in generale, le loro applicazioni nel dettaglio di questo lavoro verranno discusse in seguito.

VisualStudioCode



Visual Studio Code

Il primo che voglio menzionare dopo Metafold3D è Visual Studio Code in quanto grazie a questo software sono stati fatti dei programmi in linguaggio Python per analizzare la geometria dei provini modellati. Molto utile per valutare il rispetto di certi requisiti.

Siemens NX



Siemens NX, generalmente denominato anche solo NX, è un software CAD che consente di modellare componenti tramite comandi al suo interno. Funziona particolarmente bene utilizzando comandi come Estrusione, Rivoluzione, Foro o Unisci abbinanti ad uno Sketch (schizzo). È molto utile per la modellazione di componenti "tradizionali" e dà la possibilità di utilizzare



diversi moduli. Nel caso specifico quelli che sono stati utilizzati sono: il modulo "Progetto di Additive Manufacturing", per valutare la stampabilità pre-simulazione nel quale il software va a valutare la direzione di stampa ottimale in funzione di parametri quali la distribuzione della massa, l'altezza e gli angoli di overhang; e il modulo di simulazione di stampa (che sarebbe uno specifico Tool) dal nome Additive manufacturing process simulation, ottenuto installando Simcenter3D ossia un modulo aggiuntivo di CAE (Computer Aided Engineering) che contiene tutti i programmi che consentono di utilizzare le analisi FEM sui prodotti Siemens.

Ntop



Il software nTop (chiamato precedentemente nTopology) è un programma di modellazione molto potente che permette di lavorare su modelli CAD complessi, anche componenti per Additive Manufacturing. Data la sua complessità, però, è stato utilizzato marginalmente per la ricostruzione e conversione dei modelli, come sarà illustrato più avanti.

Ansys



Ansys è un software di simulazione ingegneristica leader nel settore, ampiamente utilizzato per progettare e analizzare prodotti virtuali prima della loro realizzazione fisica. Questo strumento consente agli ingegneri di prevedere il comportamento di un prodotto in diverse condizioni, ottimizzandone le prestazioni e riducendo i costi e i tempi di sviluppo. Ansys offre una vasta gamma di moduli per simulare una moltitudine di fenomeni fisici, per citarne alcuni:

- Strutturale: Analisi di stress, deformazioni, vibrazioni, fatica e molto altro.
- Fluidodinamica computazionale (CFD): Simulazione di flussi di liquidi e gas, come ad esempio l'aerodinamica di un'auto o il flusso sanguigno in un'arteria.
- Termico: Analisi della conduzione, convezione e radiazione del calore.
- Elettromagnetico: Simulazione di campi elettrici e magnetici, utilizzati per progettare dispositivi elettronici e sistemi elettrici.

Gli utenti possono anche creare modelli 3D all'interno del software ma la procedura più utilizzata è quella di importare modelli precedentemente realizzati in formato

standard (STEP o stp). Successivamente si perfeziona il modello a livello fisico, includendo proprietà dei materiali, condizioni di carico realistiche e interazioni tra gli elementi del sistema. Ansys consente anche di eseguire numerose simulazioni variando i parametri del progetto, identificando la soluzione ottimale in termini di prestazioni, costi e altri criteri.

Capitolo 3

Procedure sperimentali

3.1 Scelta dei campioni ottimali per la sperimentazione meccanica

Per la definizione dei campioni ottimali bisogna tenere in considerazione due aspetti principali: la struttura TPMS e la prova meccanica che verrà eseguita sui campioni. La scelta delle strutture disponibili è ampia: sono molte infatti le trattazioni che studiano le proprietà meccaniche delle celle TPMS di varia geometria. La ricerca è stata fatta in funzione del test a cui i provini dovranno essere sottoposti ossia il test alla barra di Hopkinson.

3.1.1 Barra di Hopkinson

La prova con barra di Hopkinson, o prova con barra frazionata di Hopkinson, è una metodologia sperimentale avanzata per la caratterizzazione delle proprietà meccaniche dei materiali sotto carico dinamico. Questa tecnica è particolarmente adatta per studiare il comportamento dei materiali a velocità di deformazione elevate, che possono variare da 10^2 a $10^5 s^{-1}$.

L'apparato sperimentale consiste principalmente in due barre cilindriche lunghe e sottili, comunemente realizzate in materiali ad alta resistenza e bassa attenuazione come l'acciaio o l'alluminio. Le barre sono denominate come barra incidente e barra trasmittente. Un campione del materiale da testare è posizionato tra le due barre, assicurando un contatto perfetto per garantire la trasmissione ottimale dell'onda di sollecitazione. Le fasi principali della prova sono le seguenti:

- **Generazione e Propagazione dell'Onda di Sollecitazione:** Un percussore, spesso azionato pneumaticamente, colpisce l'estremità della barra incidente, generando un'onda di compressione che si propaga verso il campione. Al contatto con il campione, parte dell'onda viene riflessa, mentre la restante parte attraversa il campione e si propaga nella barra trasmittente.
- **Rilevamento e Misurazione:** Estensimetri a resistenza (strain gauges) sono posizionati strategicamente sulle barre per rilevare le deformazioni indotte dalle onde di sollecitazione. Le tensioni di deformazione registrate permettono di

determinare le onde incidenti, riflesse e trasmesse, le quali sono fondamentali per l'analisi dei dati.

- **Analisi dei Dati:** Le equazioni di conservazione della quantità di moto e dell'energia sono applicate per correlare le misure delle onde con le proprietà meccaniche del materiale. La tensione dinamica σ e la deformazione ε nel campione possono essere espresse come:

$$\sigma = \frac{E}{2L} \cdot (\varepsilon_i - \varepsilon_r + \varepsilon_t)$$

$$\varepsilon = \frac{1}{C_0} \cdot \int (\varepsilon_i - \varepsilon_r) dt$$

dove E è il modulo di Young del materiale della barra, L è la lunghezza della barra, ε_i , ε_r , e ε_t sono le deformazioni incidente, riflessa e trasmessa rispettivamente, e C_0 è la velocità del suono nel materiale della barra.

I campioni che verranno studiati avranno dimensioni adatte alla barra di Hopkinson disponibile presso i laboratori dell'Università Politecnica delle Marche, quindi avranno dimensioni nominali di $25 \times 25 \times 25$ mm.

3.1.2 Scelta delle strutture

La scelta delle strutture si è basata su un'indagine che cerca di mettere insieme varie conoscenze apprese dagli articoli di bibliografia. Esistono diverse tipologie di TPMS, ognuna con le proprie peculiarità e vantaggi e usate per molteplici applicazioni come citato in precedenza. La Schwarz D, ad esempio, è rinomata per la sua robustezza e per la capacità di creare strutture con ampie cavità ma generalmente presenta proprietà meccaniche inferiori, mentre le geometrie P e D sono apprezzate per le loro proprietà di flusso e scambio termico, rendendole particolarmente adatte per applicazioni che richiedono un efficiente trasferimento di calore o in fluidodinamica. Dopo aver esplorato l'ampio panorama delle TPMS, ho deciso di specializzarmi sulla Giroide per le sue caratteristiche uniche e la sua rilevanza nelle applicazioni pratiche. Infatti in diversi studi mostra buone proprietà sia con metodo di costruzione Skeletal sia con metodo di costruzione Double [1,42,43]. Date le numerose ricerche riguardanti metodologie di costruzione standard (come la Skeletal e la Double appunto), in questo lavoro ho cercato di realizzare nuove geometrie (sempre utilizzando come base la TPMS Giroide) per provare ad aprire nuovi interessanti spunti sulla ricerca.

3.2 Processo di modellazione

Utilizzando Metafold3D come software si è cercato inizialmente di sfruttare la caratteristica principale: la computazione in cloud. Grazie a questa infatti si ha la

possibilità di realizzare geometrie onerose a livello computazionale in relativamente poco tempo e poterle ispezionare tramite il visualizzatore. Il problema fondamentale è stato però che dalle strutture offerte da Metafold poche risultavano appetibili per un'effettiva innovazione: le strutture TPMS generate sono tutte di tipo Skeletal e al massimo è possibile cambiarne il sistema di coordinate (come la cilindrica o sferica) ma questo non sembrava di particolare interesse per una prova di compressione monoassiale. Anche l'oggetto conformal sarebbe risultato essenzialmente inutile, in quanto i campioni per la prova alla barra di hopkinson sono essenzialmente dei parallelepipedi (quindi non c'era motivo di creare un adattamento alla geometria). Così, dopo iniziali prove, si è utilizzato l'oggetto custom shape in maniera estensiva: partendo dall'equazione implicita della giroide

$$F(x, y, z) = \cos(\omega_x x) + \cos(\omega_y y) + \cos(\omega_z z) = C$$

si è generata una semplice geometria skeletal tramite il seguente codice

```

1 float sdGyroid(vec3 p, float scale, float isoValue) {
2     const float pi = 3.14159265359;
3     const float moveX = 0;
4     const float moveY = 0;
5     const float moveZ = 0;
6     float x = (p.x - moveX) * pi * 2 / scale;
7     float y = (p.y - moveY) * pi * 2 / scale;
8     float z = (p.z - moveZ) * pi * 2 / scale;
9
10    float value = sin(x) * cos(y) + sin(y) * cos(z) + cos(x) * sin(z);
11    return value - isoValue;
12 }
13
14
15 float surface(in vec3 position) {
16     float nCell = 2.2;
17     float scale = 13.0 / nCell;
18     float isoValue = -0.92; //-0.92 isoValue for 19.8%
19     return sdGyroid(position, scale, isoValue);
20 }

```

In questo script sono presenti numerosi parametri, ma di particolare importanza è la variabile isoValue dentro la funzione surface (quella più in basso, equivarrebbe alla funzione main in altri linguaggi di programmazione). Questa variabile controlla lo spessore della geometria skeletal indicando (in maniera qualitativa) il comportamento dello spessore della struttura. Come descritto nella sezione 2.2 infatti, le SDF sono definite come funzioni che danno come risultato la distanza con segno di un punto dello spazio di analisi dal bordo di un dominio: in questo caso la variabile isoValue non indica la distanza vera e propria ma l'isovalore di tutti i punti sulla superficie.

Il valore in questione fa riferimento alla funzione in tre variabili $F(x, y, z)$ che non sarebbe altro che la funzione utilizzata nell'equazione della giroide nella tabella 1.1 (anche precedentemente richiamata), l'isovalore lì è però impostato pari a 0 quando si descrive la superficie giroide skeletal "classica" che divide lo spazio in due regioni di stesso volume (quindi al 50%) che è anche la tipologia di TPMS generata dagli oggetti surface lattice di Metafold. Osservando il codice sopra riportato il valore che la funzione surface restituisce (quel return a riga 19) richiama la funzione `sdGyroid`: in questa funzione oltre a dei parametri di posizionamento del reticolo nello spazio (`moveX`, `moveY`, `moveZ`), utilizzati generalmente per fare piccoli aggiustamenti per rimuovere o limitare eventuali aberrazioni, vengono definite le tre variabili del periodo di ripetizione nello spazio (`x`, `y`, `z`) e poi messe, alla riga 10, nell'equazione che descrive la geometria giroide. Il valore risultante, salvato nella variabile `value`, è il valore che si ottiene applicando la funzione al punto preso in esame; questo valore quindi rappresenta la vicinanza alla TPMS, ottenendo 0 quando il punto appartiene alla superficie, un valore positivo se il punto è esterno al volume solido e ottenendo un valore negativo se il punto è in direzione opposta (interno al volume solido). Per ottenere una geometria più snella però entra in gioco l'isovalore: il valore che la funzione restituisce non è direttamente quello ottenuto applicando la funzione al punto, bensì viene diminuito sottraendogli la variabile `isoValue` (riga 11). Questa è la sintassi per la realizzazione di una geometria skeletal e in particolare in questo caso dato che la variabile `isoValue` è un numero negativo, ragionando sui valori restituiti questi saranno più positivi: tutti i valori sono aumentati di 0.92, facendo diventare punti di vuoto punti che prima appartenevano al solido; tutti i punti con isovalore -0.92 ad esempio (che prima erano interni), adesso sono esattamente sulla superficie. La geometria è poi stata regolata creando un parallelepipedo di lato 25x25x25 ossia le dimensioni scelte per i campioni; il blocco è stato semplicemente creato con l'oggetto `primitive` e poi è stata data alla struttura `Custom Shape` in questione (e a tutte quelle che seguiranno) l'interazione di volume booleana `Intersection` così da restringere la computazione al solo ingombro del provino. La struttura risultante è mostrata in figura 3.1. Come si può notare è stato anche aggiunto un piattino in fondo al provino: questo servirà a migliorare i contatti in fase di simulazione meccanica; l'idea iniziale era quella di metterlo anche superiormente ma avrebbe causato solo problemi di stampabilità tra overhang e accumuli eccessivi di calore. Il piattino in questione è stato realizzato semplicemente prendendo due parallelepipedi: uno di dimensioni nominali (25x25x25) e l'altro con dimensioni (25x25x24.4) in modo tale da avere uno spessore del piattino pari a 0.6. Anche quest'ultimo oggetto interagirà allo stesso modo con le successive strutture.

Modificando l'isovalore, quindi, si ha un diretto controllo sulla geometria dell'oggetto ma resta difficile valutare la quantità di materiale che lo compone. Per valutare questo aspetto si è avuto un primo riscontro con uno strumento messo a disposizione da Metafold3D: nell'angolo in basso a destra della schermata si possono trovare dei valori relativi all'oggetto nel volume di lavoro (Figura a lato);

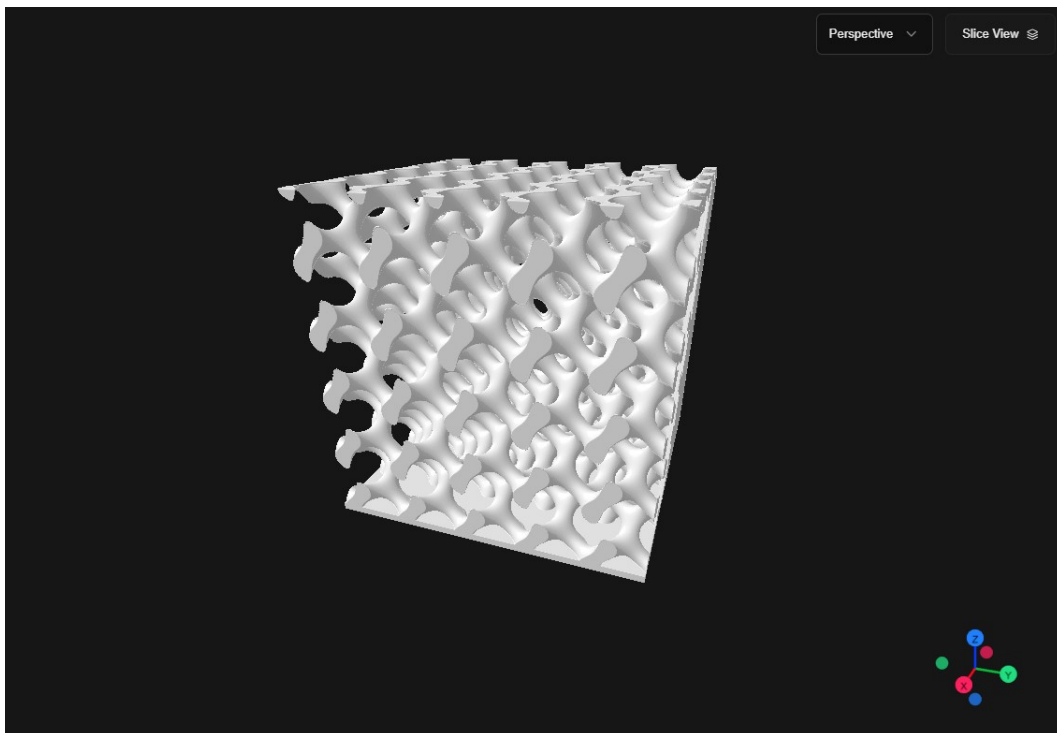


Figura 3.1: Modello realizzato in Metafaold3D di una giroide skeletal con frazione di volume del 20%

questi sono il volume di materiale (in mm^3), l'area della superficie esterna (in mm^2) e la frazione di densità di volume (con valore massimo 1.0). Grazie al valore di densità relativa, se il volume di lavoro viene stretto in modo tale da contenere perfettamente l'ingombro del provino, il valore di densità mostrato è

Shape Info	
Interior Volume (mm^3)	3063.32
Surface Area (mm^2)	8438.87
Relative Density (scene)	0.20

la frazione di volume di materiale rispetto a un provino delle stesse dimensioni ma pieno e, come si può vedere, con isovalore -0.92 la densità relativa scende a circa il 20% (dal 50% con 0). Lo scopo principale di questo provino è essere utilizzato come riferimento per la valutazione delle proprietà meccaniche di stampabilità.

Dopo aver ottenuto una procedura stabile per il controllo si è passati alla realizzazione delle idee, prima fra tutte quella di un campione di struttura giroide Graded (o gradata), ossia con una densità relativa di materiale variabile: in questo caso la direzione di variazione scelta è stata l'asse z in quanto è quella lungo la quale i carichi della prova di compressione verranno trasmessi. Per fare questo è stata aggiunta al codice grafico di Metafold una funzione per andare a gestire questa opzione.

```
float thicknessGrading(vec3 p,float moveZ){
```

```
float z = p.z;
float isoValue = -0.626; //-0.615 for 20%, -0.626 if use the palates
return isoValue -((z - moveZ) * 0.025);
}

float surface(in vec3 position) {
    ...
    float isoValue = thicknessGrading(position,move.z);
    return sdGyroid(position, scale, isoValue);
}
```

Come mostrato nel codice, all'isovalore non è dato un valore costante ma ha un valore variabile, dato dalla funzione `thicknessGrading`: questa funzione prende come variabile in ingresso la posizione del punto e (fissato un isovalore di partenza) si utilizza la coordinata `z` del punto per variare il valore di base con un certo passo. Anche in questo caso, data la complessità nel gestire la geometria, i valori inseriti sono stati ricavati manualmente tramite tentativi fino a che la struttura non risultasse sufficientemente regolare (dal punto di vista della gradazione) e non raggiungesse la percentuale di volume desiderata. La Figura 3.2 ne mostra il risultato. Successivamente mi è venuta

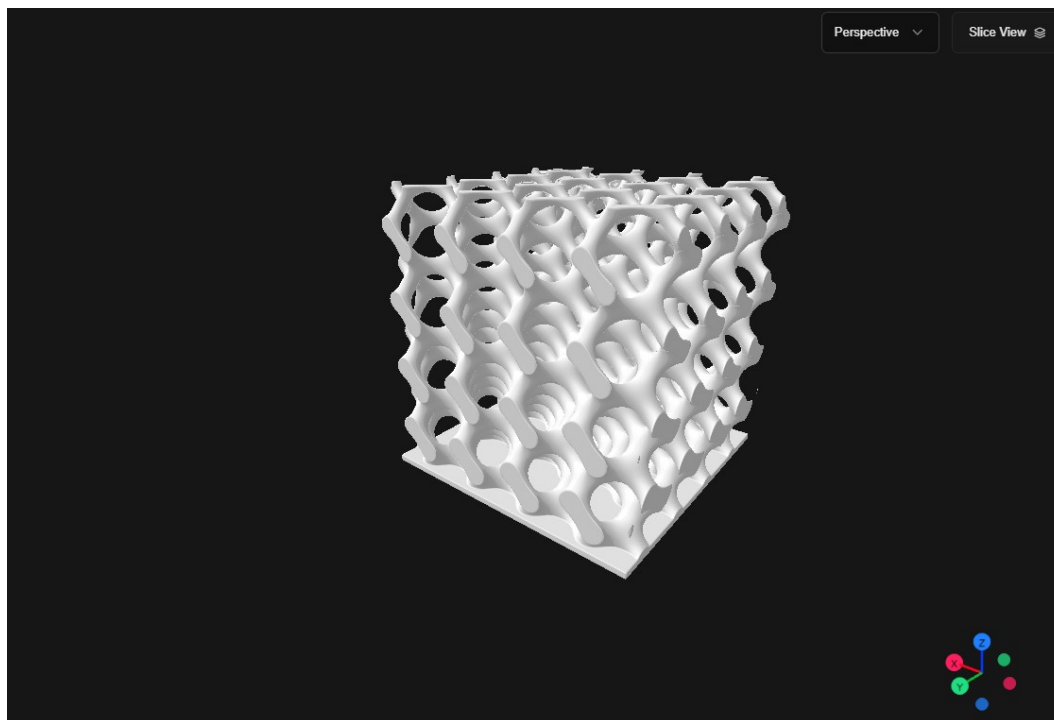


Figura 3.2: Modello realizzato in Metafaold3D di una giroide skeletal Graded con frazione di volume del 20%

un'idea interessante: dato che la giroide presenta una struttura definita e regolare nello spazio, e anche ben distribuita, sarebbe possibile creare una geometria per cui due strutture giroidi si intrecciano senza mai toccarsi. Facendo ricerche sul web pochi

studi si trovano nei quali si studia la stessa idea. Un nome ufficiale non è stato trovato quindi, per la prosecuzione di questo lavoro, questa struttura sarà chiamata Skeletal Double. Per realizzare questa struttura a livello matematico bisogna partire dal concetto stesso di TPMS Skeletal: si prende una porzione di volume chiusa attorno alla superficie e la si rende solida. secondo questa logica per descrivere al livello matematico la giroide skeletal double dobbiamo rendere negativi due insimi di punti, uno nella "geometria originale" e l'altro nella seconda metà di volume (considerando che la superficie giroide standard divida lo spazio al 50%). La soluzione che si è trovata è mostrata di seguito

```
float sdGyroid(vec3 p, float scale, float isoValue) {
    ...
    float value = sin(x) * cos(y) + sin(y) * cos(z) + cos(x) * sin(z);
    return (value - isoValue) * (-value - isoValue);
}

float surface(in vec3 position) {
    ...
    float isoValue = -1.2;
    return sdGyroid(position, scale, isoValue);
}
```

Come si può vedere dal codice la funzione che viene assegnata alla variabile `value` è sempre quella della giroide standard ma in questo caso il valore restituito viene manipolato in maniera diversa: il valore associato al punto non è altro che il prodotto di quello che avrebbe posseduto normalmente (ossia `value - isoValue`) per il valore del punto esattamente opposto rispetto alla superficie giroide centrale (infatti è stato messo `-value`); in sostanza i punti che hanno `value` negativo sono descritti dal primo fattore mentre quelli che hanno `value` positivo (e quindi sono nell'altra metà del volume) sono descritti dal secondo. In questo modo se un punto appartiene al primo dominio (cioè alla prima delle due geometrie skeletal) avrà solo il primo fattore negativo, se un punto appartiene al secondo dominio (cioè alla seconda geometria) avrà solo il secondo fattore negativo e se un punto non appartiene a nessuna delle due strutture avrà entrambi i valori positivi. L'ultimo caso in cui un punto appartiene sia al primo che al secondo dominio non è possibile perché la giroide essendo una TPMS divide lo spazio in due parti distinte senza autointersezioni. Con il prodotto dei due valori così ottenuti otterremo valore negativo quando il punto appartiene a uno dei due domini (quindi deve essere un punto solido) e valore positivo quando il punto non appartiene a nessuno dei due domini (quindi otterremmo un punto di vuoto). È interessante notare che se l'isovalore fosse impostato a 0 il valore restituito per ogni punto sarebbe $-\text{value}^2$ che è un numero mai positivo (al più zero) quindi si riempirebbero completamente entrambi i domini. Il modello ottenuto è mostrato in Figura 3.3 Un altro aspetto interessante da notare è la non linearità della variazione

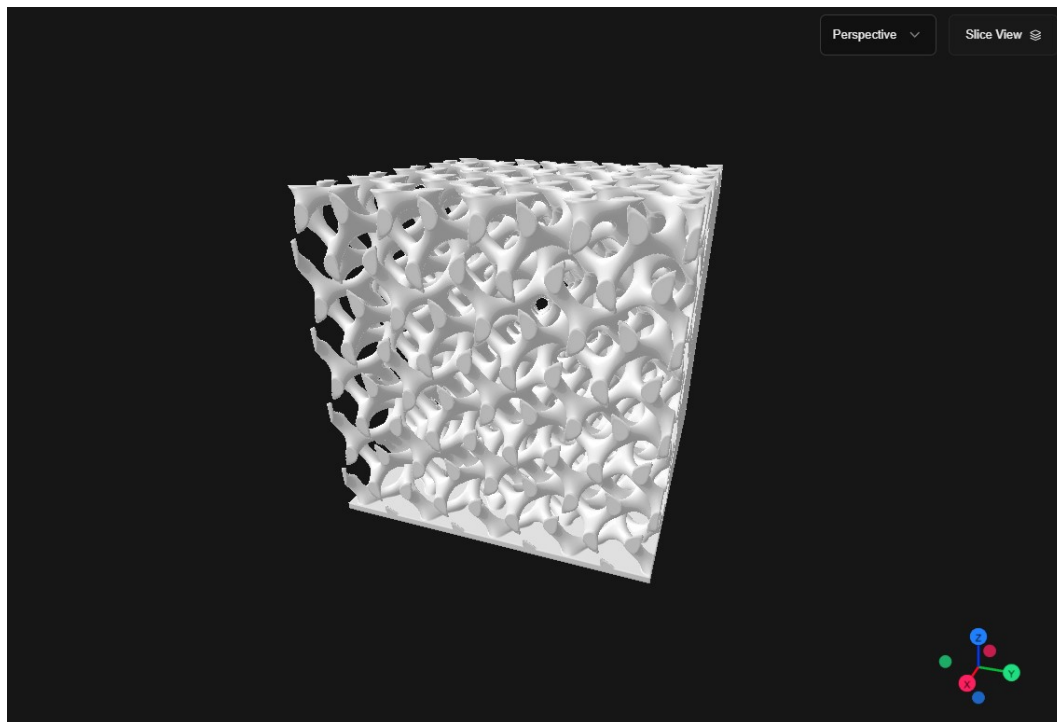


Figura 3.3: Geometria Skeletal Double giroide con densità al 20%

dell'isovalore in funzione della densità: sebbene la quantità di materiale sia la stessa (la frazione in volume è sempre il 20%) l'isovalore, il quale dovrebbe descrivere una sola metà della geometria, non è il doppio del precedente (ma -1.2).

Dopo la realizzazione della Skeletal Double ci si è chiesto se si potesse andare a realizzare una struttura simile dove però gli sforzi non rimangano separati nei due reticoli ma si trasmettano, quindi in qualche modo creare dei punti di giunzione tra i due. Gli approcci possibili potrebbero essere molteplici ma la situazione si complicherebbe dal punto di vista matematico. Si è pensato così di sfruttare l'opzione di creare due reticoli distinti e traslarli fino a farli sovrapporre. Questa variante è stata chiamata Skeletal Double Connected. Ecco dove i parametri `moveX`, `moveY`, `moveZ` sono risultati più utili. Sono stati creati quindi due oggetti Custom Shape, il primo con il seguente codice

```
float sdGyroid(vec3 p, float scale, float isoValue) {  
    const float pi = 3.14159265359;  
    const float moveX = 0 - 1.1;  
    const float moveY = 0 + 1.9;  
    const float moveZ = 0 + 0.5;  
    ...  
  
    float value = sin(x) * cos(y) + sin(y) * cos(z) + cos(x) * sin(z);  
    return (value - isoValue);  
}
```

```
float surface(in vec3 position) {
    ...
    float isoValue = -1.173;
    return sdGyroid(position, scale, isoValue);
}
```

mentre il secondo è stato spostato di un quarto della dimensione della cella e considera i punti simmetrici cambiando il segno del valore ottenuto

```
float sdGyroid(vec3 p, float scale, float isoValue) {
    const float pi = 3.14159265359;
    const float moveModule = scale/4;
    const float moveX = moveModule - 1.1;
    const float moveY = moveModule + 1.9;
    const float moveZ = moveModule + 0.5;
    ...
    float value = sin(x) * cos(y) + sin(y) * cos(z) + cos(x) * sin(z);
    return (-value - isoValue);
}
```

i numeri che si sommano alle variabili move servono, come già detto prima, a fare piccoli aggiustamenti per minimizzare le aberrazioni. Dopo aver creato questi oggetti li si è messi in un unico gruppo così da poterne controllare l'attivazione (tramite il piccolo occhio sulla sinistra) contemporaneamente. Il modello risultante è quindi quello mostrato in Figura 3.4 Successivamente si è pensato di creare ulteriori

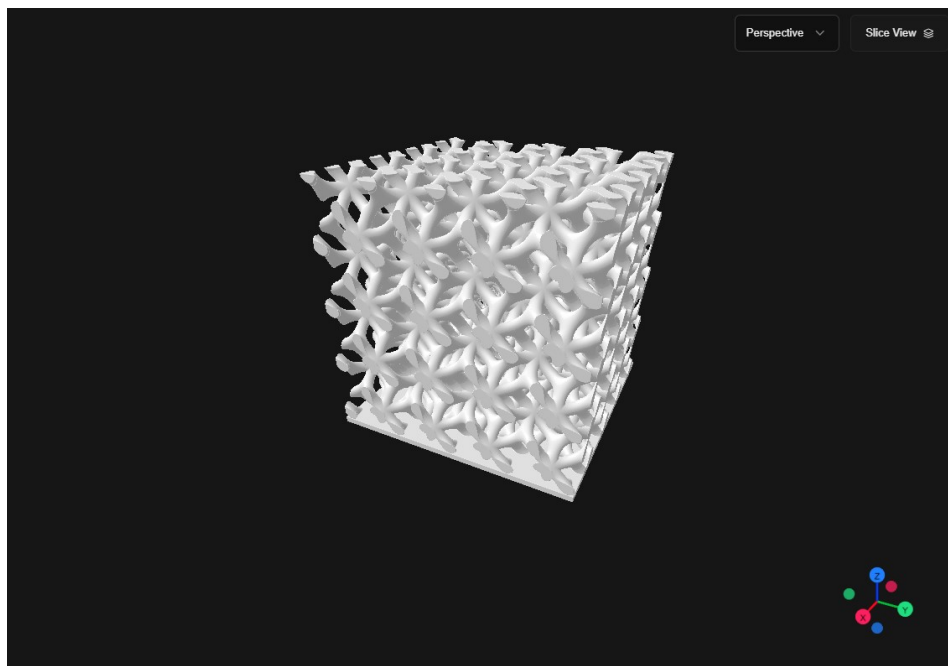


Figura 3.4: Geometria Skeletal Double Connected giroide con densità al 20%

varianti di questa struttura. Dato che i nodi di intersezione dei reticoli formano un'intersezione pressoché planare ma orientata in direzione obliqua, il collegamento può essere fatto facilmente: riuscire a ruotare il reticolo in modo tale che i nodi siano orientati orizzontalmente e verticalmente. Grazie alla possibilità di decidere posizione e rotazione degli oggetti in Metafold3D questa operazione diventa più facile perché, senza scomodare la rotazione matematica delle equazioni, è possibile ruotare direttamente il sistema di riferimento dell'oggetto. Purtroppo non avendo uno strumento che consentisse di misurare con precisione la direzione del vettore ortogonale al nodo anche in questo caso il risultato è stato ottenuto per tentativi: in particolare si hanno 45° di rotazione sull'asse z e rispettivamente 55° e -35° sull'asse x per la struttura a nodi orizzontali (chiamata Horizontal) e a nodi verticali (chiamata Vertical). I risultati sono mostrati nelle Figure 3.5a e 3.5b

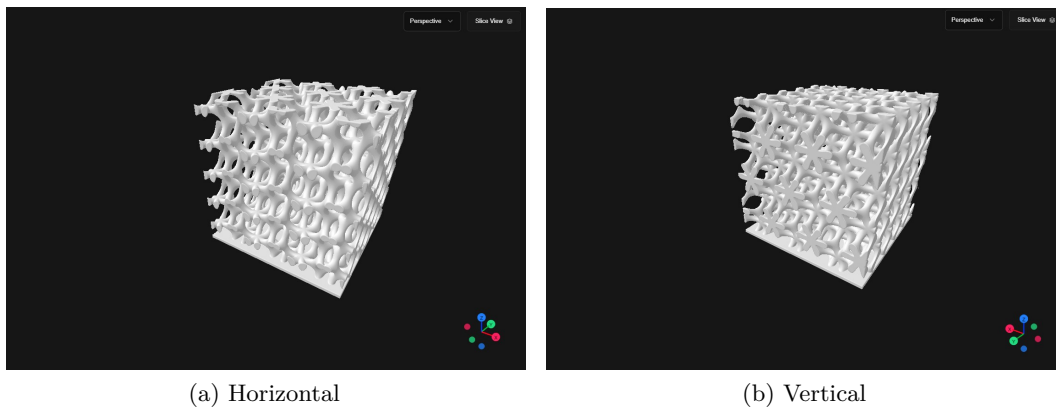


Figura 3.5: (Varianti) Geometria Skeletal Double Connected giroide con densità al 20%

Dopo aver prodotto già una buona diversità di campioni si è provato a sperimentare ulteriormente con la manipolazione dell'isovalore. La successiva ispirazione è venuta proprio pensando alla geometria Skeletal Double infatti in questo oggetto le due metà della struttura non si incontrano mai (si sta parlando di quella non connected ovviamente) e si va a descrivere il solido, come detto in precedenza, con 3 risultati possibili: punto solido della prima geometria, punto solido della seconda geometria e punto che non appartiene a nessuna delle due. Il caso di intersezione delle geometrie non deve essere gestito come detto, ma cosa succederebbe in caso ci fossero intersezioni? Se si dovesse conservare come operatore di combinazione in prodotto si avrebbe che nei punti di intersezione il risultato sarebbe positivo quindi i punti che appartengono a entrambe le geometrie risulterebbero punti di vuoto, cosa alquanto spiacevole nella maggior parte dei casi. In tutti probabilmente, tranne in uno: una struttura tubulare. E così nasce l'idea di una struttura Tubular, altra geometria molto poco studiata in letteratura. L'idea di realizzazione di questa geometria è realizzare due strutture skeletal normali che però si intersecano completamente, in

particolare una contiene completamente l'altra. Questa situazione (al contrario del caso precedente) può essere ottenuta matematicamente in maniera semplice. Di seguito il codice utilizzato per realizzarla

```
float sdGyroid(vec3 p, float scale, float isoValue) {
    ...
    float value = sin(x) * cos(y) + sin(y) * cos(z) + cos(x) * sin(z);
    return value - isoValue;
}

float surface(in vec3 position) {
    ...
    float isoValue = -0.4;
    float deltaIsoValue = 0.59;
    return sdGyroid(position, scale, isoValue) * sdGyroid(position, scale,
        isoValue - deltaIsoValue);
}
```

Come si può vedere dal codice questa volta la funzione `sdGyroid` è normale senza alcuna manipolazione particolare (rappresenta cioè una semplice Skeletal come nel provino di riferimento) ma la vera differenza sta nella definizione della variabile `deltaIsoValue`: questo non è altro che il valore di differenza tra l'isovalore della skeletal esterna (quindi più spessa e con isovalore maggiore) e della skeletal interna (quindi più sottile e con isovalore minore). Quello che la funzione poi restituisce non è nient'altro che il prodotto del punto applicato a entrambe le geometrie come nel caso della Skeletal Double, solo che questa volta viene richiamata due volte la stessa funzione prima con l'isovalore maggiore e poi con quello minore creando due geometrie identiche a meno dello spessore. Il risultato è mostrato in Figura ??

Aspetto interessante da notare è che la struttura somiglia molto alla geometria Sheet (o Double): infatti se si imposta un isovalore pari al doppio di `deltaIsoValue` si ottiene una struttura simmetrica rispetto alla superficie con isovalore 0 come se questa fosse inspessita, esattamente la definizione di Sheet TPMS.

3.3 Analisi geometrica

Durante la realizzazione dei modelli dei provini si è reso necessario l'analisi della loro geometria, non avendo un programma che esplicitamente ci consenta di controllarne i parametri in uscita. Un esempio è proprio la percentuale di volume, valutata solo in prima approssimazione con i valori forniti da Metafold3D. Dato che il restringimento del volume al solo ingombro del provino poteva in alcuni casi a creare artefatti grafici non consoni, si è deciso di calcolare la frazione di volume tramite codice. Nello specifico è stato realizzato un codice Python che, tramite le grandi capacità di calcolo numerico offerte dalla libreria Numpy, riesce a valutare con grande precisione la

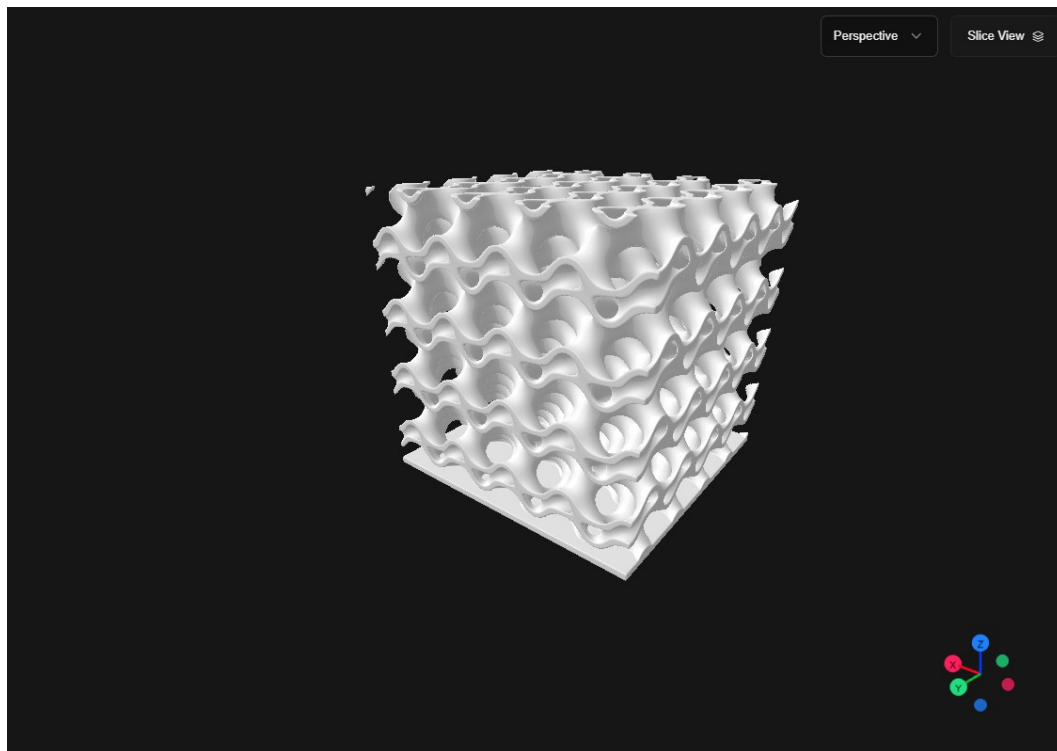


Figura 3.6: Modello di una giroide Tubular con frazione di volume del 20%

frazione del materiale all'interno di un volume. Nel dettaglio verranno spiegati gli elementi del codice.

Si inizia importando le librerie necessarie

```
import numpy as np
import sdfModule as sdfM
```

che in questo file sono solo due: Numpy per l'appunto e sdfModule, un file esterno creato appositamente che contiene tutte le funzioni per modellare le superfici SDF. È stato chiamato così perché non descrive semplicemente le superfici TPMS ma serve a ottenere gli isovalori di tutto lo spazio (i quali comunque non corrispondono alla distanza però il nome SDF è la cosa che più può rendere l'idea). Il codice è comunque stato scritto in maniera abbastanza grezza e finalizzata allo scopo: la formattazione non è delle migliori e non sono stati previsti input da parte dell'utente che ne fa uso; il codice viene eseguito direttamente nel software di editing (VisualStudioCode). Anticipato questo, vengono definite poi le variabili globali, parametri che vengono modificati manualmente e che regolano la geometria e il posizionamento della struttura

```
scale = 13/2.2
isoValue = -0.92

# Box parameters
x_shift,y_shift,z_shift = 0.2,0.15,0
```

```

x_min, y_min, z_min = 0 + x_shift, 0 + y_shift, 0.8 + z_shift

x_sample, y_sample, z_sample = 25, 25, 25

factor = 0

x_max, y_max, z_max = scale * factor + x_shift + x_sample, \
    scale * factor + y_shift + y_sample, scale * factor + z_shift +
    z_sample

# Grid resolution
resolution = 300 # Number of stitches per size

```

prima fra tutti è `scale` (la scala, la lunghezza della cella), lo stesso valore (ovviamente) utilizzato per i provini; poi c'è la variabile `isoValue`; poi si sono quelli chiamati "Box parameters" che non sono altro che una serie di valori che definiscono il volume di lavoro (valori massimi e minimi delle coordinate della scatola o anche la lunghezza del campione); in fine vediamo la variabile di risoluzione del volume di lavoro (`resolution`), la quale ci serve per definire con che grado di precisione si ottiene il risultato. Definite queste variabili inizia la funzione principale

```

def main():

    # Create a grid of points
    x = np.linspace(x_min, x_max, resolution)
    y = np.linspace(y_min, y_max, resolution)
    z = np.linspace(z_min, z_max, resolution)
    xx, yy, zz = np.meshgrid(x, y, z)

```

Il primo passaggio è quello di creare il nostro volume di lavoro. Per farlo, prima si creano 3 vettori lineari `x`, `y`, `z` tramite la funzione `linspace` di `numpy` in cui si crea una serie di valori di numerosità pari alla risoluzione, equidistanti e che vanno dalla coordinata `min` e la coordinata `max` di `x`, `y`, `z`; poi si combinano questi valori con la funzione `meshgrid` per creare lo spazio dei punti. Fondamentalmente si hanno 3 spazi di 3 dimensioni distinti (`xx`, `yy`, `zz`) nei quali ognuno contiene il valore della coordinata corrispondente del punto che si trova nella stessa posizione nel volume di lavoro. Questo offre una grande potenzialità che è quella di applicare una funzione a tutti i punti dello spazio. Se io per esempio andassi a sommare i tre spazi

```

sum_coordinates = xx + yy + zz

```

otterrei uno spazio delle stesse dimensioni di quelli precedenti dove a ogni punto è associata la somma delle proprie coordinate. Questo consente di sfruttare appieno tutta la potenza di `numpy` e svolgere operazioni cicliche in pochi istanti. Proseguendo

con il codice

```
sdfSurface = sdfM.GyroidSkeletal(scale,isoValue)

#grading = sdfM.isoVGradedZ(isoValue,0.04,zz)

# Calculate the SDF for each point in the grid
sdf_values = sdfSurface.getValue(xx, yy, zz)
```

viene generato un oggetto utilizzando la classe relativa alla geometria TPMS (in questo caso Gyroid Skeletal) assegnandogli i parametri di lunghezza di cella e isovalore. In seguito verrà esaminato il codice delle classi modellate in SDF module. Nella riga utile successiva si ha una funzione per il grading: questa genera uno spazio che contiene la variazione dell'isovalore partendo da un valore di passo dato; l'unica coordinata fornita è la `zz` perché la gradazione viene realizzata solo in senso verticale. In questo codice la funzione è commentata perché come esempio si sta usando la geometria Skeletal Gyroid non gradata. Successivamente si ha che viene creato (con il principio visto poc'anzi) uno spazio in cui a ogni punto è associato il suo valore della funzione SDF, utilizzando l'oggetto `sdfSurface`, generando una mappa della geometria. Per realizzare questa matrice viene utilizzato il metodo `getValue` dell'oggetto `sdfSurface` passandogli i valori delle coordinate. In questa parte di codice è stata utilizzata la classe `GyroidSkeletal` ma basta cambiare la classe per ottenere una mappa della geometria diversa. Per concludere si ha che

```
# Apply a mask to filter the points
solid_mask = sdf_values <= 0

# Calculate the percentage of the volume
volume_percentage = np.sum(solid_mask)/ (resolution**3) * 100

print(f'\nSolid Volume Percentage: {volume_percentage}%\n')
```

```
main()
```

viene dapprima applicata una maschera sullo spazio: questa non è altro che un'ulteriore matrice ottenuta dall'applicazione di una condizione a tutti i valori SDF dello spazio. In questo modo si ottiene un oggetto delle stesse dimensioni dove a ogni punto corrisponde un valore di verità (Vero o Falso). La condizione è semplice: si cercano tutti quei valori minori o uguale a zero per ottenere tutti i punti interni al solido. L'ultima operazione calcola direttamente la frazione di volume della struttura sommando algebricamente tutti i punti della maschera (così facendo si ottiene il numero di punti del solido, dato che algebricamente Vero corrisponde a 1 e Falso corrisponde a 0) e dividendo per i punti totali (la risoluzione elevata al cubo), moltiplicando il numero per 100 per avere il risultato percentuale. In fine il risultato

viene stampato a schermo con un print, come si può vedere in Figura 3.7.

Come si può vedere dall'immagine la densità non è esattamente del 20% ma un po'

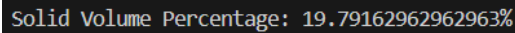


Figura 3.7: Risultato del calcolo di densità di una giroide skeletal

meno: questo era il modo migliore per far combaciare tutte le densità. Il risultato è anche congruo con quello di Metafold3D, dato che questo restituisce la densità in forma unitaria con solo due cifre dopo la virgola. Sono state fatte anche delle prove e si è visto come quel valore sia un'approssimazione. Parlando delle classi del modulo sdfModule, queste sono gestite con una struttura gerarchica: c'è una classe padre che modella genericamente una funzione SDF e poi viene ereditata dalle classi figlie specifiche. La classe padre è definita nel modo seguente

```
class sdfTPMS():
    def __init__(self,scale,isoValue) -> None:

        self._scale = scale
        self._isovalue = isoValue
        self._equation = None

    def getEquation(self):
        return self._equation

    def getGrad(self,values:np.ndarray = []):
        x, y, z = sp.symbols('x y z')
        grad = sp.Matrix([sp.diff(self._equation, var) for var in (x, y,
            z)])
        if len(values) > 0:
            return np.array([[float(expr.evalf(subs={x:point[0] , y:
                point[1] , z:point[2]})) for expr in grad] for point in
                values])
        return grad

    def setScale(self,new_scale):
        self._scale = new_scale

    def setIsovalue(self,new_isovalue):
        self._isovalue = new_isovalue

    def getScale(self):
        return self._scale

    def getIsovalue(self):
        return self._isovalue
```

```
def getValue(self,x: float | np.ndarray,y: float | np.ndarray,z: float
| np.ndarray) -> float | np.ndarray:
    value_function = sp.lambdify((x,y,z), self._equation,'numpy')
    return value_function(x,y,z)
```

tutte le funzioni definite (con def) sono funzioni interne alla classe e sono chiamati metodi. Il metodo `__init__` è detto anche costruttore della classe e definisce gli attributi che l'oggetto creato con la classe possiederà. In generale nella programmazione a oggetti le classi sono paragonabili a una sorta di stampo con cui gestire insiemi di dati complessi. Chiamare una classe nel codice (con ad esempio `NomeClasse()`) crea un oggetto di quella classe che avrà a disposizione tutte le sue proprietà e i suoi metodi. Gli attributi non sono altro che dei "sottovalori" associati all'oggetto principale, sono molto utili perché viaggiano con l'oggetto e sono facilmente accessibili dai suoi metodi. L'assegnamento di questi attributi viene fatto utilizzando il `self`: quando `self` viene inserito come prima variabile presa in input da un metodo, quello che questa variabile contiene è l'oggetto stesso. In questo modo associando a `self._scale` la variabile `scale` sto generando un attributo di scala (`_scale`) associato all'oggetto (`self`). Come si può notare gli attributi che vengono assegnati sono la scala, l'isovalore e l'equazione: in questo caso l'equazione non c'è perché ci troviamo nella classe padre generica, però è bene comunque definirla assegnandogli un valore di assenza (`None`). I metodi definiti successivamente sono tutti metodi che permettono di interagire con l'oggetto e sono chiamati getter e setter. I getter hanno il compito di restituire un valore associato all'oggetto e magari anche prima processarlo, i setter invece hanno il compito di impostare dei valori o attributi all'oggetto. I setter non sono importanti ai fini di questo utilizzo perché non andremo mai a modificare un oggetto già generato, però per completezza sono stati comunque inseriti. Parlando quindi dei getter abbiamo che `getEquation`, `getScale` e `getIsovale` sono abbastanza intuitivi e restituiscono i valori dei corrispettivi attributi. Sia `getGrad`, il quale restituisce il gradiente della funzione $F(x, y, z)$ che costituisce l'equazione, che `getValue` di questo costruttore includono l'utilizzo di una libreria per lavorare con equazioni simboliche. Sono un aspetto che verrà analizzato in seguito. Definita la classe padre possiamo ora passare a una classe figlio, come esempio prendiamo sempre la classe che modella la skeltal giroide

```
class GyroidSkeletal(sdfTPMS):
    def __init__(self, scale, isoValue) -> None:
        super().__init__(scale, isoValue)
        x, y, z = sp.symbols('x y z')
        self._equation = sp.sin(2 * x * sp.pi / self._scale) * sp.cos(2 * y
            * sp.pi / self._scale) + sp.sin(2 * y * sp.pi / self._scale) *
            sp.cos(2 * z * sp.pi / self._scale) + sp.cos(2 * x * sp.pi /
            self._scale) * sp.sin(2 * z * sp.pi / self._scale) -
            self._isoValue
```

```

def getValue(self,x: float | np.ndarray,y: float | np.ndarray,z: float
| np.ndarray,grading: np.ndarray = np.array([]) ) -> float |
np.ndarray:
    xScale,yScale,zScale = scaleValues(x,y,z,self._scale)
    value = np.sin(xScale) * np.cos(yScale) + np.sin(yScale) *
        np.cos(zScale) + np.cos(xScale) * np.sin(zScale)
    if not grading.size == 0: return value - grading
    else: return value - self._isovalue

```

Nel codice il costruttore della classe richiama a sua volta il costruttore del padre per associare gli stessi attributi (`super().__init__(scale, isoValue)`) fornendogli le variabili date a lui in ingresso. Successivamente tramite la libreria Simpy si definisce l'equazione in forma simbolica della giroide, ma come già detto questo verrà analizzato in seguito. Per il calcolo del volume quello che ci serve è il metodo `getValue`. Qui possiamo vedere definiti come argomenti, oltre a `self`, le coordinate `x,y,z` come argomenti necessari, le quali possono essere numeri float oppure vettori numpy (ossia le matrici che dobbiamo lavorare), e un argomento opzionale che è il `grading`. All'interno del metodo si vede prima una funzione per scalare i valori delle dimensioni sul periodo (come in `Metafold3D`), e poi la definizione dell'equazione della giroide con funzioni numpy: in questo modo sia che venga passato uno spazio a 3 coordinate oppure solo 3 coordinate singole le funzioni saranno applicate lo stesso. Poi è definito un `if`: l'array `grading` è impostato di default ad essere un array nullo quindi se non viene inserito in input (quindi la sua `size` fosse 0) si andrebbe sull'`else` e si restituirebbe lo spazio dei valori - l'attributo di isovalore dell'oggetto; se invece una matrice viene assegnata e la sua dimensione non è 0 questa viene sottratta allo spazio dei valori (lo spazio di `grading` viene realizzato a monte con la funzione `sdfM.isoVGradedZ` che prima era commentata).

Un altro aspetto importante da considerare è lo spessore minimo della struttura: dato che parliamo di oggetti piccoli e geometrie molto leggere (quindi anche molto sottili) bisogna cercare di valutare in qualche modo lo spessore minimo. Anche perché questo valore è limitato inferiormente dalla risoluzione del laser, in quanto piccoli dettagli possono perdere di qualità e una geometria priva di difetti come una TPMS ne risentirebbe molto. Per questo motivo è stato realizzato un altro codice per la valutazione dello spessore minimo. La logia che si è cercato di adottare è quella di ricerca del punto in direzione ortogonale alla superficie: prima si filtrano tutti i punti il cui valore spaziale (in modulo) è vicino allo zero, cioè punti vicini alla superficie (la funzione utilizzata per calcolarli è quella che definisce l'SDF, come su `Metafold`); successivamente si valuta il vettore gradiente della funzione in ogni punto trovato, in questo modo è possibile identificare una direzione ortogonale alla superficie passante per il punto; in fine si utilizza un metodo di bisezione cercando un altro punto sulla stessa direzione che sia prossimo alla superficie e la distanza tra i due punti ci darebbe lo spessore. Il procedimento di bisezione ovviamente deve essere applicato ad ogni singolo punto. Vediamo nel dettaglio la struttura del codice.

Anche qui si inizia importando le librerie necessarie

```
import sympy as sp
import numpy as np
import sdfModule as sdfM
```

In questo caso si aggiunge la libreria Sympy, una potente libreria che consente di utilizzare il calcolo simbolico. Si definiscono i parametri come in precedenza

```
scale = 13/2.2
isoValue = -0.4 - 0.59

# Box parameters
x_shift,y_shift,z_shift = 0,0,0
#x_shift,y_shift,z_shift = scale/2,scale/2,scale/2

x_min, y_min,z_min= 0 + x_shift, 0 + y_shift, 0 + z_shift

x_sample, y_sample, z_sample = 0,0,0

factor = 1

x_max,y_max,z_max = scale * factor + x_shift + x_sample, scale * factor
    + y_shift + y_sample, scale * factor + z_shift + z_sample

# Grid resolution
resolution = 400

default_tolerance = 1 * 10 ** -4
```

ma si può notare che c'è una nuova variabile ossia `default_tolerance`. Questa variabile serve per impostare un valore di tolleranza con la quale cercare i punti per misurare lo spessore. Si avvia allora la funzione `main`

```
def main(isoValue):

    max_tolerance = default_tolerance

    sdfSurface = sdfM.GyroidSkeletal(scale,isoValue)

    # Create a grid of points
    x = np.linspace(x_min, x_max, resolution)
    y = np.linspace(y_min, y_max, resolution)
    z = np.linspace(z_min, z_max, resolution)
    xx, yy, zz = np.meshgrid(x, y, z)

    sdf_values = sdfSurface.getValue(xx, yy, zz)
```

e si definiscono le stesse variabili viste nel codice precedente con anche quel valore di `max_tolerance`. A questo punto la prima fase della ricerca dello spessore minimo può iniziare. L'idea è quella intanto di definire un campione di punti sui quali lavorare perché la computazione comunque sarà abbastanza onerosa

```

MAXPOINTS = 200
MINPOINTS = 100
while True:
    solid_mask_max = np.abs(sdf_values) <= max_tolerance
    max_points = np.sum(solid_mask_max)
    if max_points >= MAXPOINTS: break
    max_tolerance = max_tolerance * 2

```

si definiscono così le costanti `MAXPOINTS` e `MINPOINTS` che determinano l'intervallo nel quale la numerosità di punti ricadrà. Per valutare una numerosità corretta si vuole utilizzare il metodo della bisezione. Dapprima si stabilisce il valore di tolleranza massima: si crea una maschera solida sui punti dello spazio il cui valore assoluto è più piccolo della tolleranza e si verifica quanti punti sono associati a quel valore di tolleranza (sommando la maschera e riducendola ad un numero intero); dopo di che se questi punti sono superiori al massimo possibile si può procedere (con il `break`) altrimenti si raddoppia la tolleranza massima e si ripete il ciclo. Questo perché volendo utilizzare il metodo della bisezione per determinare un valore di tolleranza corretto, necessitiamo di un valore inferiore di tolleranza (che sarebbe 0) e di un valore superiore (che sarebbe proprio `max_tolerance`): dobbiamo però essere certi che la soluzione che stiamo cercando sia all'interno del nostro intervallo quindi se il valore massimo di tolleranza trova più punti di quelli necessari la condizione è sicuramente soddisfatta. Proseguendo con il codice

```

up_value, low_value = max_tolerance, 0
while True:
    tolerance = (up_value + low_value)/2
    solid_mask = np.abs(sdf_values) <= tolerance
    use_points = np.sum(solid_mask)
    if use_points <= MAXPOINTS and use_points >= MINPOINTS: break
    elif use_points > MAXPOINTS: up_value = tolerance
    elif use_points < MINPOINTS: low_value = tolerance

```

inizia il ciclo di bisezione: vengono definiti a monte i valori estremi dell'intervallo (assegnati alle variabili `up_value` e `low_value`) e poi si entra nel ciclo; si definisce il valore di tolleranza da utilizzare come la media degli estremi e si cerca il numero di punti nello stesso modo precedente assegnando il valore di punti trovati alla variabile `use_points`; si controlla se i punti trovati sono nell'intervallo desiderato, nel qual caso si esce, mentre in caso contrario la variabile `tolerance` viene assegnata all'estremo superiore se si sono trovati più punti di quelli voluti oppure all'estremo inferiore se se ne sono trovati meno. L'ultima porzione del codice `main` consiste nel calcolo dello

spessore minimo usando i punti

```

surface_points = np.column_stack((xx[solid_mask], yy[solid_mask],
                                  zz[solid_mask]))
grad_list = sdfSurface.getGrad(surface_points)
pass
thickness_list = [thickness_evaluation(surface_points[i], grad_list[i],
                                     sdfSurface, tolerance) for i in range(len(grad_list))]
thickness = min(thickness_list)
print(f"\nThe minimum thickness is: {thickness}\n")

```

nella variabile `surface_points` vengono estratti tutti i punti dallo spazio che appartengono alla superficie: prima generiamo una tupla che contiene 3 elementi `xx[solid_mask], yy[solid_mask], zz[solid_mask]` che non sono altro che rispettivamente tutte le coordinate x, tutte le y e tutte le zeta dei punti della superficie. L'applicazione della maschera allo spazio infatti restituisce un array monodimensionale contenente tutti i valori in cui la maschera risulta vera. Successivamente con la funzione `column_stack` si va a ruotare l'oggetto scambiando le righe e le colonne: da un oggetto di dimensione $3 \times m$ (dove m è il numero di punti) otteniamo un oggetto di dimensioni $m \times 3$. Di fatto adesso abbiamo creato un oggetto che contiene m elementi ognuno di lunghezza 3, questi non sono altro che i punti.

Successivamente si trasforma la lista dei punti nella corrispondente lista dei vettori gradienti grazie al metodo `getGrad`. Per concludere si crea una lista di tutti gli spessori con i risultati della funzione `thickness_evaluation` eseguita su tutti i punti che abbiamo e si prende il minimo di questi come valore di spessore minimo. Ora un piccolo accenno al metodo `getGrad`. Questo metodo è assegnato agli stessi oggetti utilizzati nel codice per la valutazione del volume solo che viene utilizzato in questa circostanza. Prima in costruzione dell'oggetto si assegna una equazione simbolica all'attributo `_equation` (prendiamo l'esempio della Skeletal Giroide)

```

class GyroidSkeletal(sdfTPMS):
    def __init__(self, scale, isoValue) -> None:
        super().__init__(scale, isoValue)
        x, y, z = sp.symbols('x y z')
        self._equation = sp.sin(2 * x * sp.pi / self._scale) * sp.cos(2 * y
                               * sp.pi / self._scale) + sp.sin(2 * y * sp.pi / self._scale) *
                               sp.cos(2 * z * sp.pi / self._scale) + sp.cos(2 * x * sp.pi /
                               self._scale) * sp.sin(2 * z * sp.pi / self._scale) -
                               self._isoValue

```

si può notare che l'equazione è scritta esattamente allo stesso modo di quella utilizzata nel `getValue` ma le funzioni invece che `np` presentano il prefisso `sp`: questo perché non sono funzioni numpy in grado di fare potenti calcoli rapidamente ma sono funzioni sympy, ossia funzioni che vengono salvate come simboliche per poter essere manipolate matematicamente. Infatti sopra la definizione dell'equazione c'è la definizione delle

variabili x, y, z come variabili simboliche sympy. Guardando al metodo `getGrad`

```
def getGrad(self, values: np.ndarray = []):
    x, y, z = sp.symbols('x y z')
    grad = sp.Matrix([sp.diff(self._equation, var) for var in (x, y,
        z)])
    if len(values) > 0: return
        np.array([[float(expr.evalf(subs={x:point[0] , y: point[1] ,
            z:point[2]})) for expr in grad] for point in values])
    return grad
```

vediamo come l'argomento `values` è opzionale (perché se on viene assegnato ha già un valore assegnato de default): in questo modo se non viene passato nessun valore si restituisce soltanto l'equazione in forma simbolica del gradiente. Comunque si specifica al sistema nuovamente che le variabili simboliche sono x, y, z e si genera l'equazione del gradiente (che ovviamente risulta essere un vettore) facendo il differenziale (`diff`) rispetto alla variabile (`var`) con le variabili x, y, z e si mette il risultato in una matrice sympy. al passo successivo se la lunghezza della variabile `values` è 0 si restituisce direttamente l'equazione, altrimenti si entra nell'`if` dove ci sono un pò di operazioni annidate: partendo dalle parentesi più interne abbiamo che l'espressione (`expr`) viene valutata (`evalf`) assegnando a x, y, z rispettivamente i valori nelle posizioni 0,1,2 dell'iterabile `point`; questo per `expr in grad` (quindi `expr` rappresenta in maniera iterativa le tre componenti del gradiente) e per `point in values` che è la lista di punti che abbiamo dato al metodo; tutto questo viene poi convertito in array numpy e restituito.

3.4 Processo di simulazione

Come già detto precedentemente il processo di simulazione si è svolto in due fasi principali: la prima di simulazione del comportamento del provino durante la stampa LPBF; la seconda di simulazione meccanica con una prova di compressione dinamica.

3.4.1 Simulazione di stampa

Metafold3D, anche a detta degli stessi sviluppatori, punta a realizzare direttamente i campioni per la stampa. Per questo motivo, in particolar modo ora che il software è ancora in sviluppo, l'unica opzione di export file meglio utilizzabile è quella in formato `slt`. Questo però per questo lavoro si è rivelato essere un problema perché NX in particolare, ma anche ANSYS, non lavorano bene con file `stl`, sarebbe opportuno utilizzare dei file di volume in formato `step`. Per questo motivo un potente software di modellazione 3D è stato utilizzato per risolvere il problema: `nTop`. Dopo il passaggio su `nTop` il file è stato caricato su NX per la simulazione.

Preparazione dei campioni per la simulazione

In ambiente nTop è stata utilizzata una funzione molto potente chiamata Simplified Body Representation. Questa funzione consente di ricostruire il modello CAD di un oggetto per adattarlo a quello di un corpo implicito. Definiamo prima però alcune tipologie di oggetti tridimensionali:

- Una mesh (o superficie reticolata) è una rappresentazione geometrica di un oggetto tridimensionale composta da poligoni, solitamente triangoli o quadrilateri. Le mesh sono costituite da vertici, spigoli e facce: i vertici sono i punti nello spazio, gli spigoli sono le linee che collegano i vertici e le facce sono le superfici piane delimitate dagli spigoli. Le mesh sono molto flessibili e possono rappresentare forme complesse e dettagliate, motivo per cui sono ampiamente utilizzate in grafica 3D, animazione e stampa 3D. Possono essere facilmente modificate spostando i vertici, aggiungendo o rimuovendo facce, e applicando trasformazioni come scalatura e rotazione. Tuttavia, non sono ideali per rappresentare superfici lisce e continue, poiché sono approssimazioni poligonali. La qualità della rappresentazione dipende dalla densità della mesh: più poligoni ci sono, più dettagliata sarà la rappresentazione. Infatti i file stl generati da Metafold3D sono molto pesanti anche alla minima qualità possibile;
- Un implicit body (o corpo implicito) è una rappresentazione matematica di un oggetto tridimensionale definita da una funzione implicita. Questo significa che l'oggetto è descritto da un'equazione matematica che determina se un punto nello spazio appartiene o meno all'oggetto. Ad esempio, una sfera può essere definita dall'equazione $x^2 + y^2 + z^2 = R^2$, dove R è il raggio, oppure le geometrie modellate in questo capitolo lo sono. Gli implicit body, infatti, sono particolarmente utili per rappresentare superfici lisce e continue senza la necessità di approssimazioni poligonali. Sono molto utili per le operazioni booleane (unione, intersezione, differenza) tra solidi, poiché queste operazioni possono essere eseguite direttamente sulle equazioni che definiscono gli oggetti. Gli implicit body possono rappresentare dettagli molto fini e precisi, poiché non sono limitati dalla risoluzione di una mesh;
- Un CAD body è un oggetto più generico che si riferisce a qualsiasi geometria tridimensionale creata all'interno di un software CAD (Computer-Aided Design). I CAD body possono essere rappresentati in vari modi, tra cui mesh, superfici NURBS (Non-Uniform Rational B-Splines) e solidi parametrici. La rappresentazione più comune nei software CAD è quella dei solidi parametrici, dove l'oggetto è definito da una serie di parametri e relazioni geometriche. Questi parametri possono essere facilmente modificati per aggiornare il modello, rendendo i CAD body molto flessibili e adatti per la progettazione ingegneristica e la produzione.

Avendo definito questi oggetti possiamo parlare della conversione realizzata su nTop. La Figura 3.8 mostra sulla sinistra l'albero logico realizzato per blocchi delle funzioni utilizzate.

Nell'albero a blocchi le funzioni sono annidate una dentro l'altra e le prime ad essere

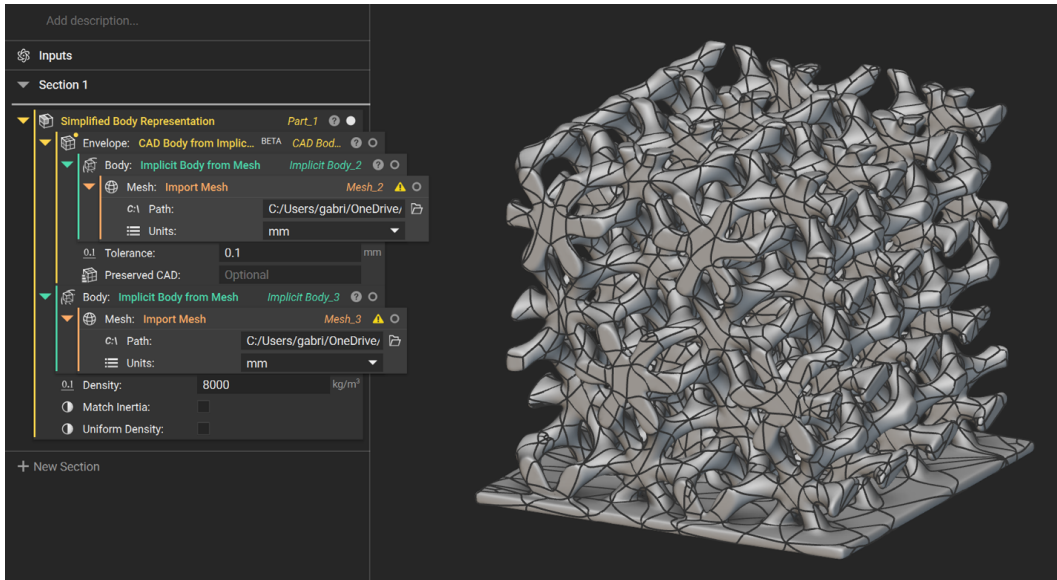


Figura 3.8: L'albero di conversione (sulla sinistra) e il modello caricato su nTop della Skeletal Double Connected Horizontal (sulla destra)

eseguite sono quelle più interne per passare via via verso quelle più esterne. Vediamo quindi che la Simplified Body Representation richiede un CAD Body e un Implicit Body a cui adattarsi. La CAD body from Implicit Body (funzione ancora in beta) richiede a sua volta un implicit body. Per questo motivo come input della funzione CAD Body e secondo input della funzione Simplified Body si usa lo stesso oggetto: in entrambi i casi la funzione Implicit Body richiede una Mesh come modello d'ingresso. Quindi quello che il sistema cerca di fare è creare un CAD Body (Con la Simplified Body) convertendo la Mesh di partenza direttamente in CAD Body e aggiustandone la geometria con un Implicit Body realizzato dalla stessa Mesh. In pratica è un doppio controllo. Quando il sistema è riuscito ad elaborare questa operazione (molto onerosa a livello computazionale dato che le mesh di partenza sono molto grandi) il modello CAD è pronto ma prima di scaricarlo il provino è stato anche tagliato: in questo modo le simulazioni possono andare più velocemente e possiamo trascurare gli effetti di bordo che vedremmo nella realtà. Le dimensioni di tutti i provini sono state ridotte da $25 \times 25 \times 25$ mm a $18 \times 19 \times 17,7$ mm (il taglio non è stato controllato alla perfezione ma è una cosa che è emersa a fine lavoro) tranne il modello Graded la cui altezza è rimasta a 25 mm così da sfruttare la variazione di spessore della cella lungo la direzione z. Adesso ogni modello è pronto e il suo file step può essere scaricato e importato su NX.

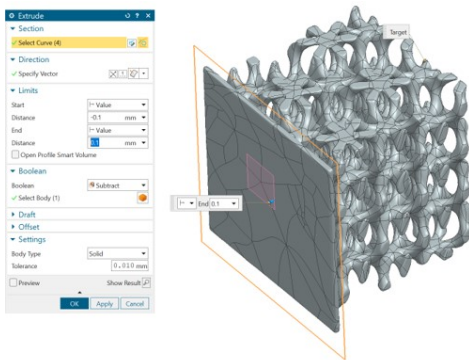


Figura 3.9: Rifilo del piano inferiore

Dopo aver importato il file step in NX si è proceduto al taglio del piattino del fondo (Figura 3.9) perché Metafold3D con gli oggetti che gestisce internamente (gli oggetti primitive) tende a raccordare le superfici, combinando questo aspetto al fatto che il modello è stato anche ricostruito da una mesh, il risultato è che il piatto inferiore non è perfettamente planare e non garantirebbe il corretto appoggio sul piatto di stampa, alterando la simulazione. Per risolvere questo

inconveniente è stato eseguito un comando di Estrusione con sottrazione booleana ottenendo una superficie correttamente ricostruita. Successivamente il file è stato salvato in .prt, ossia l'estensione nativa di NX, per facilitare la maneggevolezza nel software.

Impostazione della simulazione

Per prima cosa bisogna creare l'ambiente di stampa in cui avverrà la simulazione. Si crea quindi un nuovo file Additive Manufacturing e si impostano per prima cosa i parametri del volume di stampa (Figura 3.10) per poi posizionare il componente al suo interno vincolandolo al piano di stampa (Figura 3.11).

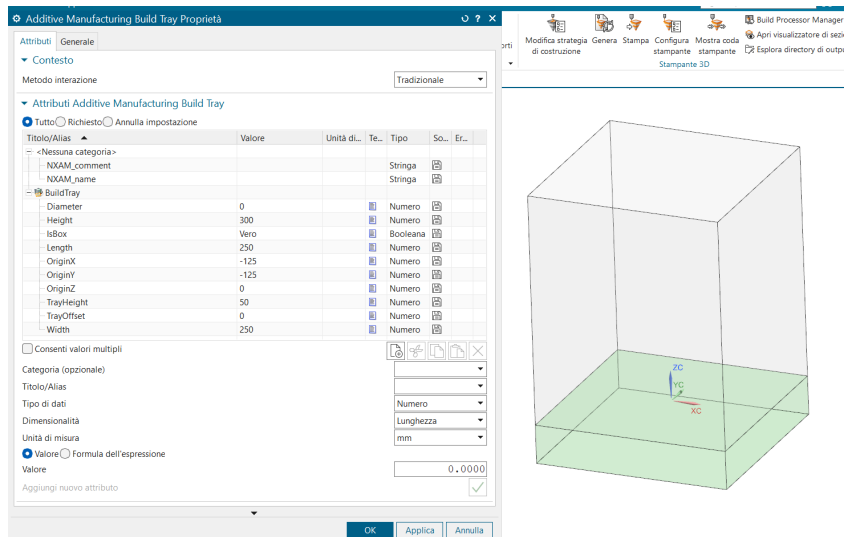


Figura 3.10: Parametri del volume della stampante (sulla sinistra) e volume di stampa (sulla destra)

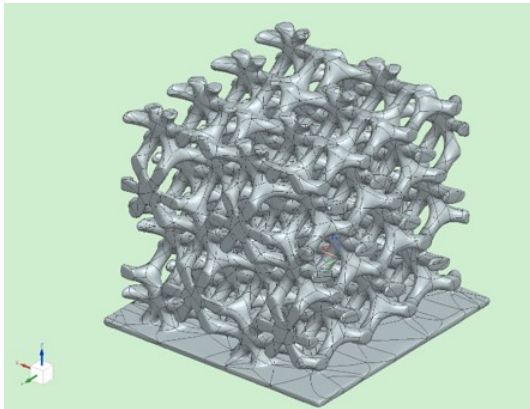


Figura 3.11: Componente sul piatto di stampa, essendo molto piccolo si vede solo il piano verde

Dopo il posizionamento del componente si definisce la tipologia di simulazione (termica più meccanica nel nostro caso) e si imposta il materiale con le sue proprietà (AISI 316L).

Successivamente bisogna far generare al sistema una mesh per la simulazione: in questo caso sono state impostate con mesh a Voxel. Con questo tipo di mesh il volume viene diviso in tanti piccoli cubi (l'equivalente 3D di un pixel, infatti Voxel deriva da volume pixel). La mesh è stata impostata in modo da avere una geometria che fosse il più attinente possibile con quella nominale.

Il metodo utilizzato da NX per i calcoli prende il nome di inherent strain: tale metodo prevede l'omogenizzazione dei del componente a livello delle singole slice impostate sul software. Ciò permette di valutare il componente in maniera discretizzata suddividendo il problema principale in un numero finito di problemi a minor complessità caratterizzati da vincoli bonded tra i vari layer in modo da considerare tutto il corpo come un continuum. In questo modo la simulazione risulta molto più veloce e se fatta con uno slicing corretto permette di prevedere correttamente il comportamento del provino simulato. Per tale sono state eseguite più prove per verificare lo slicing ottimale che non generasse delle condizioni al contorno gravose che alterassero i dati in output nelle zone tra una slice e un'altra, potendo essere utilizzato per tutte le simulazioni. Il valore ottimale è stato 0.4. In Figura 3.12 un esempio di mesh risultante con tanto di valore di slice utilizzato (e Voxel per slice).

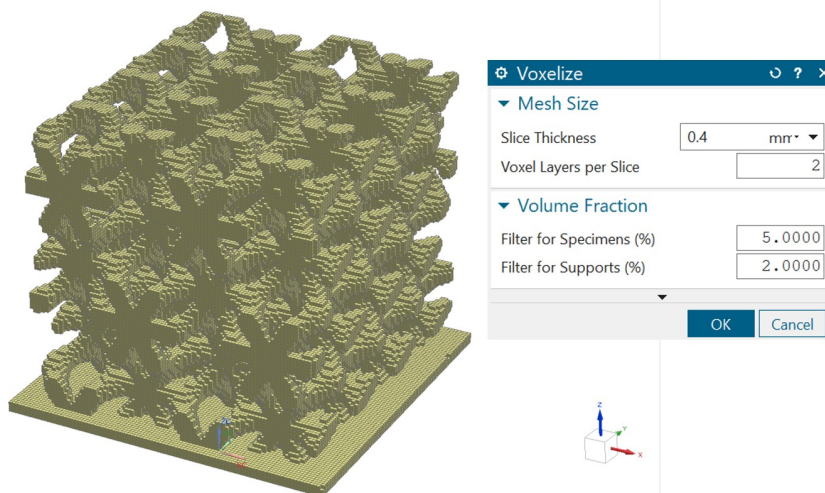


Figura 3.12: Mesh del provino SD Connected Horizontal e i parametri della Mesh

3.4.2 Simulazione della prova meccanica

In questa fase ci saremmo concentrati solo su i campioni che avrebbero presentato minori quantità di tensioni residue, come preannunciato nel capitolo 2.

Impostazione della simulazione

Per queste simulazioni è fatto uso del modulo Explicit Dynamics: questo modulo è caratterizzato da un solutore esplicito che permette di affrontare problemi caratterizzati da alta velocità di deformazione e grandi deformazioni. In Figura 3.13 è mostrato il box con tutti gli aspetti che costituiscono la simulazione. All'interno del software sono stati implementati i parametri relativi al modello del Johnson-Cook (JC) relativi al materiale di base. In particolare, nel luglio del precedente anno sono stati testati quasi-staticamente e dinamicamente dei provini cilindrici in 316L nelle condizioni as built, di diametro e altezza 5x5 mm. Da questi sono stati poi ricavati dal gruppo di costruzione di macchine i valori del modello del JC riportato e riportato nel seguente sistema di equazioni:

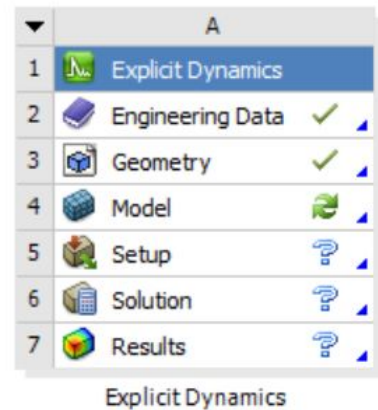


Figura 3.13: Box del modulo Explicit Dynamics

$$\sigma = (A + B\dot{\varepsilon}^n)(1 + C \ln \varepsilon^*)(1 - T^{*m}) \quad (3.1)$$

$$\dot{\varepsilon}^* = \frac{\dot{\varepsilon}}{\dot{\varepsilon}_{\text{ref}}} \quad (3.2)$$

$$T^* = \frac{(T - T_{\text{ref}})}{(T_m - T_{\text{ref}})} \quad (3.3)$$

Nello specifico, A corrisponde alla tensione di snervamento del materiale determinata con prova quasi-statica, B e n sono rispettivamente il coefficiente lineare e il coefficiente esponenziale relativi al fenomeno dell'incrudimento determinato con test quasi-statico, C è il coefficiente lineare legato al fenomeno dello strain-rate hardening determinato attraverso l'utilizzo di test dinamici ed infine m il coefficiente esponenziale termico legato ai fenomeni di ammorbidimento del materiale che si verificano all'aumentare della temperatura eseguendo test dinamici a differenti temperature. In Tab. 3.1 sono riportati i valori di tali coefficienti ottenuti dalle prove sperimentali. In Fig. 3.14 è possibile visualizzare i valori del modello implementati in Ansys e come lo strain rate hardening, pari all'incremento della tensione nella curva stress-strain, di tale materiale sia molto basso (e dunque efficiente per applicazioni riguardanti l'assorbimento di urti). Una nota importante è che non è stato impostato un modello di danneggiamento, con una semplificazione questo viene raggiunto quando gli elementi del campione arrivano a un valore di strain troppo elevato (1.5 delle

dimensioni originali). Purtroppo non si dispone al momento di un modello realizzato perché non si sono fatte le prove.

A	B	n	C	m
524	1583	0,8159	0,0071	0,7

Tabella 3.1: Tabella contenente i parametri del modello Johnson-Cook

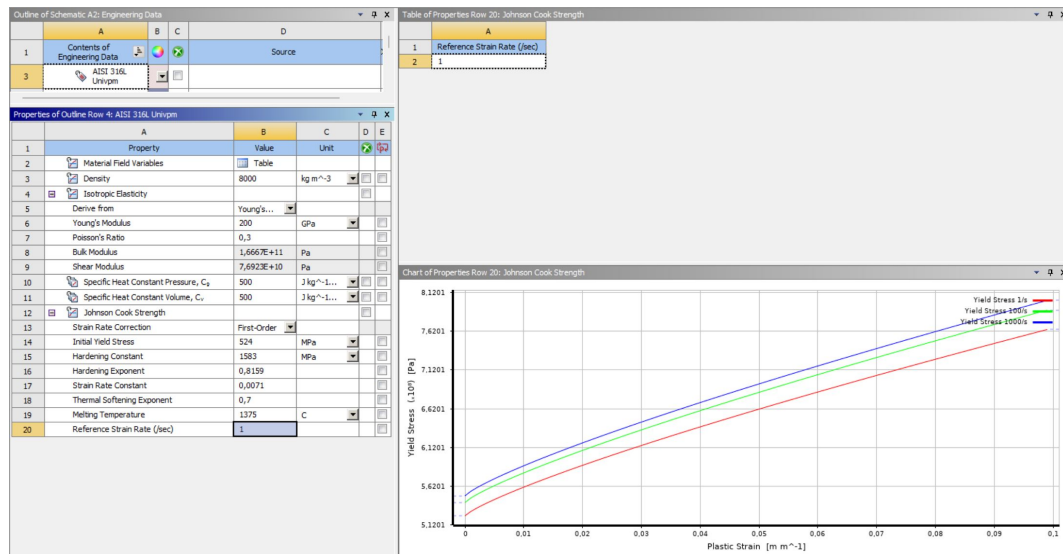


Figura 3.14: Modello dell'AISI 316L contenuto nel database universitario

Dopo aver impostato il modello del materiale si è generata la mesh utilizzata per la simulazione. La tipologia di mesh in questo caso è quella Tetraedrica. La differenza con la mesh a Voxel è che essendo il tetraedro un oggetto meno ingombrante è in grado di seguire meglio le superfici avendo una qualità migliore. A parità di dimensioni dell'elemento (ossia il lato del solido) la mesh tetraedrica è più pesante perché contiene più elementi e le interazioni tra questi elementi sono più complesse. Basti pensare che nella mesh Voxel questi sono orientati con i lati paralleli al sistema di riferimento, quindi tutti i contatti sono sugli stessi piani, quantomeno inizialmente. Il risultato ottenuto si può osservare in Figura 3.15 che mostra la mesh della struttura Tubular (a) e quella della struttura Graded (b).

È stato creato su NX il set-up necessario per implementare le simulazioni. Nello specifico questo prevede l'utilizzo di un piatto fisso e un piatto mobile caratterizzato da una determinata velocità di impatto, questi sono stati realizzati semplicemente come due cilindri messi a contatto con il provino come mostrato in Figura 3.16. Una volta creato l'assieme in NX e salvato questo come step, esso è stato importato su Ansys. All'interno di Ansys è stato impostato come materiale di base dei piatti un acciaio strutturale e come materiale di base del provino il modello del 316L precedentemente discusso. In questo modo il sistema dovrebbe dare in output valori

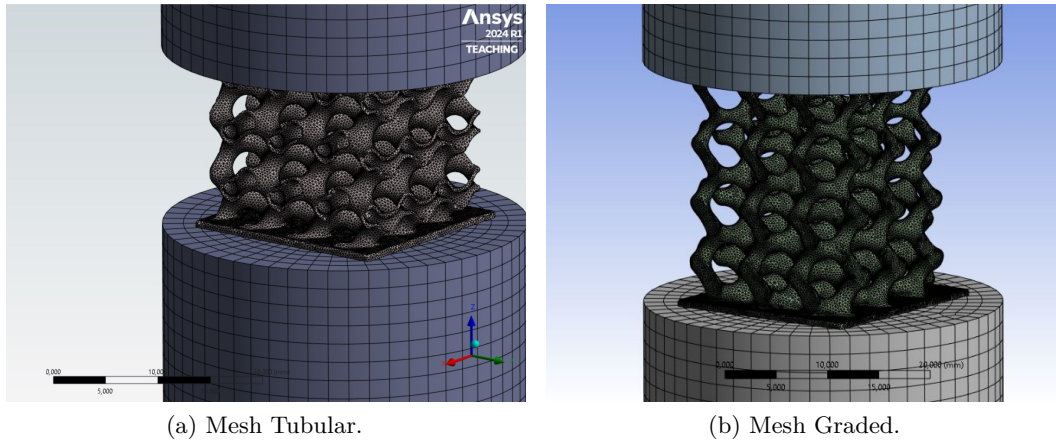


Figura 3.15: Mesh realizzata da Ansys delle geometrie Tubular (a) e Graded (b)

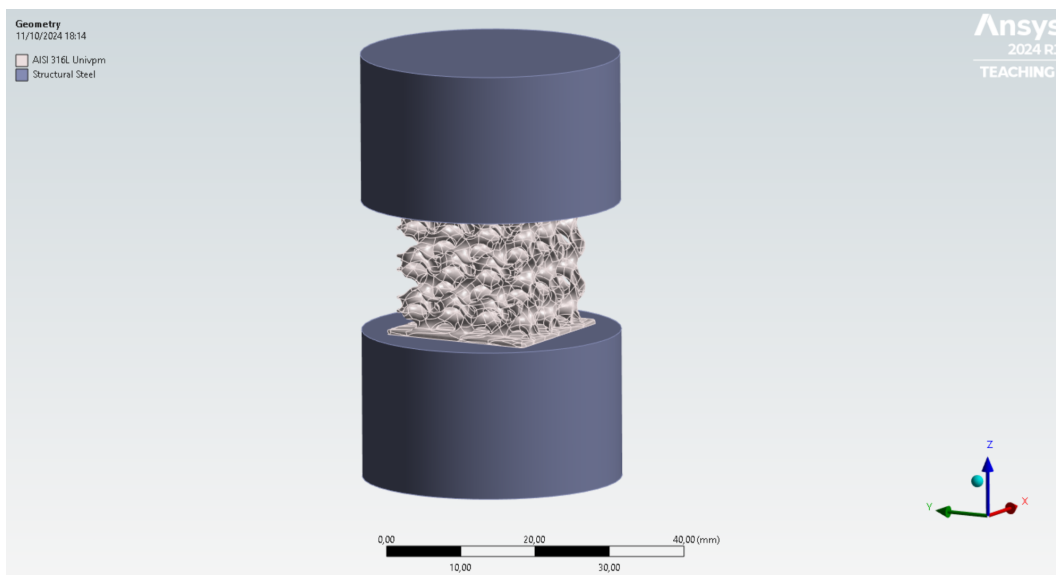


Figura 3.16: Assieme provino-piatti importato su Ansys, la struttura mostrata è la Tubular

quanto più verosimili ai fenomeni che avvengono durante il test reale. Per quanto riguarda i contatti, sulla base di simulazioni precedenti svolte (comparando i dati numerici con quelli sperimentali), sono stati impostati dei contatti di tipo Frictional tra le superfici dei piatti e le superfici del provino a contatto con essi (Figura 3.17). Tali contatti sono stati impostati in modo da avere un coefficiente di attrito pari a 0,2 sia per quanto riguarda l'attrito statico che dinamico. I coefficienti di attrito sono stati lasciati nulli per l'analisi del self-body-contact relativo all'interazione delle struts del provino durante la compressione (quando, cioè, le struts, implodendo, vanno a collidere le une con le altre). Questo consente di avere una simulazione meno complessa e la perdita di dati è minima dato che essendo le struts regolari, anche in caso di contatto difficilmente si muoverebbero le une rispetto alle ma continuerebbero a implodere.

Nella sezione Analysis Settings si è poi andato a impostare il tempo della simulazione.

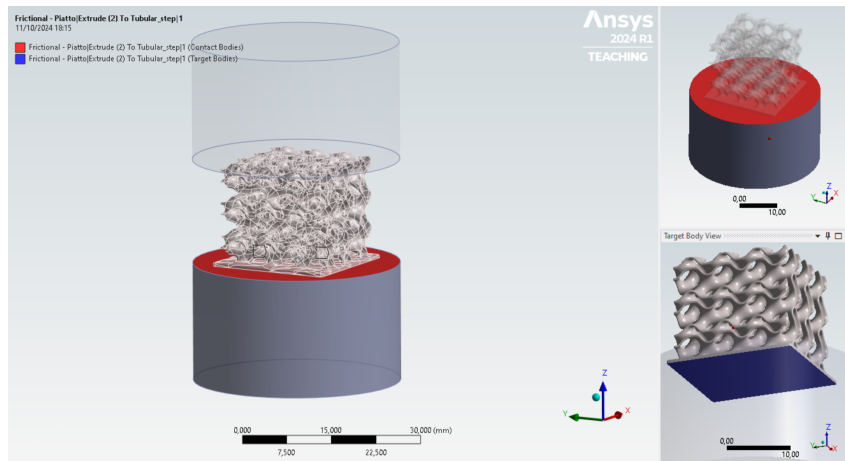


Figura 3.17: Definizione contatto tra piatto fisso e base del provino

Questo viene calcolato come S/v . Mentre v è la velocità di movimento del piatto mobile della prova (50 m/s) lo spostamento è dato dalla valutazione della densificazione del materiale, ossia quando il materiale viene schiacciato completamente fino ad aver raggiunto densità del 100%. Nel nostro caso l'altezza dei provini tagliati è 17,7 mm quindi, sapendo che la densità è del 20%, l'escursione massima sarebbe $17,7\text{mm} \cdot 0,8 = 14,16\text{ mm}$. Mantenendo una certa distanza dal valore limite (perché non ci interessa raggiungere il 100% di densificazione) il valore di escursione preso è 12 mm con un tempo di esecuzione risultante di $2,4 \cdot 10^{-4}$ s. Per quanto riguarda la Graded invece l'escursione massima risulta 20 mm, il valore preso in considerazione è stato 18 mm e il tempo risultante è di $3,6 \cdot 10^{-4}$ s.

Sono stati poi impostati i vincoli (Figura 3.18)e:

- Vincolo di Fixed support sulla superficie inferiore del piatto a contatto con la parte del provino caratterizzata da maggior densità (A);

- Velocità del piatto superiore a contatto con la parte reticolare del provino pari a 50 m/s per valutare il comportamento del provino ad alte prestazioni. Tale condizione è quella per cui le strutture TPMS metalliche vengono generalmente progettate (B);
- Vincolo di spostamento nullo eccetto lungo la direzione verticale per evitare traslazioni indesiderate del piatto mobile durante la simulazione (se per esempio le strutture dovessero cedere con una deformazione obliqua o a x);

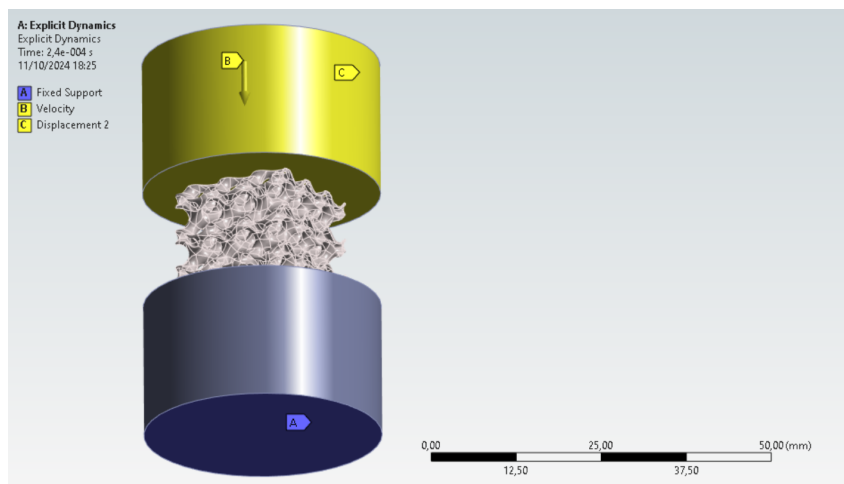


Figura 3.18: Definizione contatto tra piatto fisso e base del provino

In fine si apre la sezione di soluzione, in cui si va a scegliere cosa avere nei risultati. Le opzioni scelte sono:

- Deformazione direzionale in direzione Z del piatto mobile: utilizzata per valutare la corsa del pistone a velocità pari a 50 m/s e successivamente lo strain della prova;
- Valore della reazione vincolare (Force reaction) sul piatto superiore ed inferiore: ciò serve per valutare se il corpo è in una condizione di equilibrio o no ma soprattutto a valutare la forza esercitata sul provino durante la compressione (i dati più affidabili sono quelli del piatto mobile, quello fisso subisce degli effetti di ritardo inerziale);
- Distribuzione degli stress di Von-Mises per verificare dove si accumulano maggiormente gli stress durante la deformazione. Si potrebbe pensare di prendere direttamente questi valori, tuttavia ciò non è corretto in quanto i provini cellulari vanno analizzati sempre a partire dai valori di forza perché gli stress non si distribuiscono in maniera coerente sulle sezioni che tra l'altro tendono anche a variare nella direzione di carico. Pertanto si prendono i valori di stress come valori indicativi, per analizzare visivamente dove si concentra maggiormente

il carico e dove il materiale può raggiungere condizioni critiche che possono portare ad una rottura catastrofica o alla nucleazione e crescita di vuoti;

- Entità di deformazione elastica e plastica del provino. In questo modo, similmente a quanto detto per gli stress di Von-Mises, si ha un riscontro visivo della deformazione del provino e della distribuzione di questa.

La Figura 3.19 mostra l'albero completo della simulazione con tutte le voci necessarie assegnate.

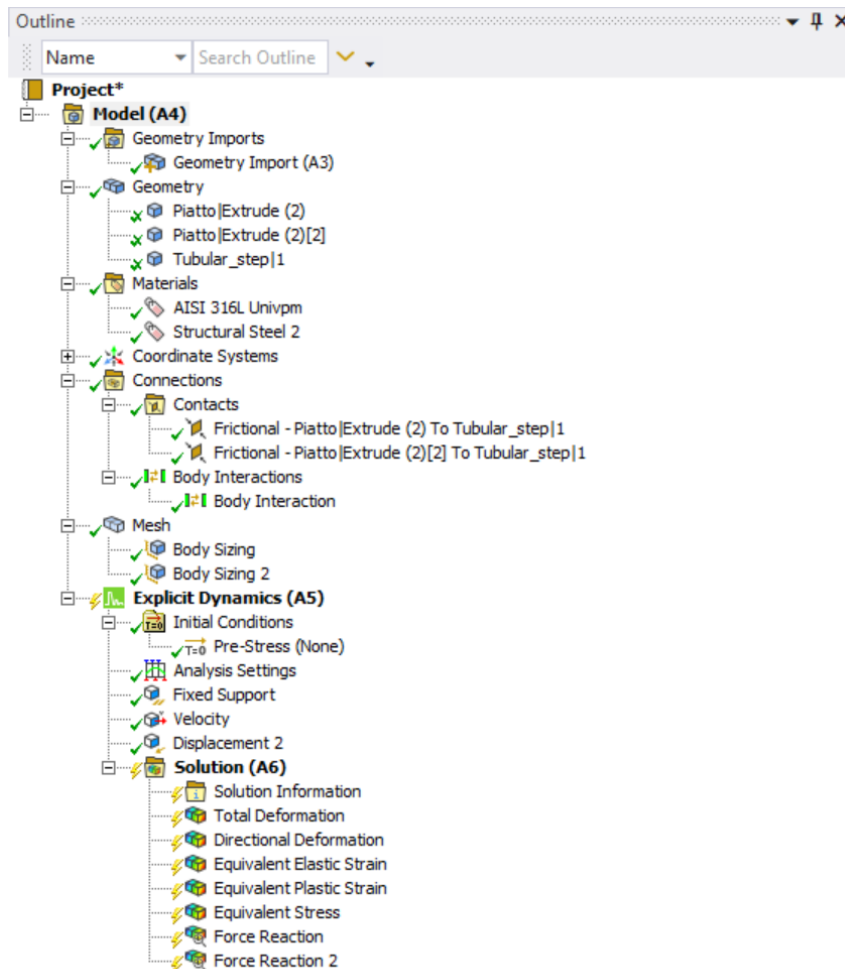


Figura 3.19: Albero della simulazione meccanica con tutte le voci necessarie assegnate

Capitolo 4

Analisi dei risultati

4.1 Analisi della simulazione di stampa

Dopo aver realizzato i modelli già a occhio si potevano osservare alcune problematiche di stampabilità su alcuni di essi. In particolare, aspetto confermato da una successiva analisi, la struttura SD Connected Horizontal presenta parecchi problemi di overhang: la presenza di quei nodi, relativamente grandi rispetto al resto della struttura, e il fatto che fossero appunto disposti orizzontalmente, hanno portato a decidere di non simulare in stampa questa struttura. Ovviamente dal punto di vista reale non è irrealizzabile: per esempio ruotando di 90° il provino, i nodi si troverebbero in posizione verticale come l'altra variante, la Vertical. Questo però aggiungerebbe un ulteriore fattore di complessità e variabilità dato che non è possibile (ad oggi) definire la direzione dei grani (dipendente dalla direzione di stampa) all'interno di una struttura 3D, però è cosa nota che la direzione di stampa influisce profondamente sulle proprietà meccaniche del provino (in special modo quelli lasciati as-built) perché l'elevato calore generato favorisce la ricristallizzazione dinamica dei grani cambiandone forma e proprietà (spesso questi si trovano accresciuti proprio nella direzione di stampa). Questo è un fattore che, tra gli altri campioni, può essere tralasciato con discreta approssimazione dato che si parla di geometria simile, ma non se la direzione di stampa dovesse essere diversa. In generale essendo queste geometrie simili perché tutte base giroide, se si considera la stessa direzione di stampa la conformazione dei grani dovrebbe essere abbastanza simile in forma, dimensioni e orientamento rispetto alla struttura.

Osservando i risultati sugli stress residui si può valutare quali geometrie presentano la condizione migliore as-built. Come spiegato precedentemente, per ottenere delle simulazioni meccaniche che potessero avvicinarsi il più possibile alla realtà si sarebbe dovuto prendere le strutture con la minor entità di stress residui dato che i dati su questi non possono essere importati in Ansys. In questo caso tutte le mappe di colore sono state scalate ad un valore massimo di 912 MPa, il quale viene raggiunto solo localmente da alcune in qualche punto (probabilmente è un valore anomalo) ma che serve come riferimento. Guardando alle mappe di colore si può notare come la distribuzione del materiale in due volumi abbia beneficiato la capacità di smaltimento del calore per la Skeletal Double (Figura 4.3); la Giroide

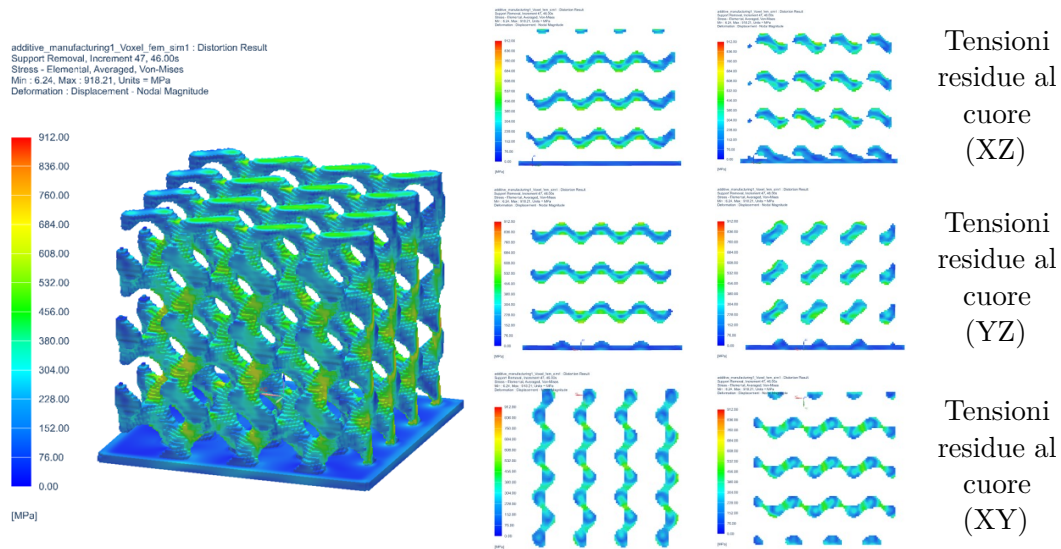


Figura 4.1: Risultato simulazione di stampa Giroide Skeltal di riferimento

Skeletal (Figura 4.1) standard si comporta discretamente bene (ma verrà importata a prescindere come riferimento); la Graded (Figura 4.2) presenta una distribuzione di tensioni residue variabile lungo la lunghezza, in maniera concorde con le differenze di spessore tra la SD e la Skeletal standard, può essere presa in considerazione perché più si scende con i livelli più le porzioni di struttura hanno meno tempo per lavorare in una prova di compressione quindi gli effetti della distorsione non saranno così marcati; la Tubular (Figura 4.4) sembra presentare diversi punti di concentrazione del calore ma questi non sono regolari: le pareti della struttura sono molto sottili ed avendo la tipica geometria a spirale della giroide forse è stata quella su cui si sono sentiti maggiormente gli effetti della simulazione in Voxel, quindi verrà presa in considerazione tenendo sempre a mente possibili divergenze dalla realtà; le due strutture SD Connected standard e variante Vertical simulate (rispettivamente in Figura 4.5 e 4.6) sono risultate le più inefficaci: l'accumulo di calore ai nodi abbastanza marcato unito questo al fatto che essi potrebbero rappresentare zone critiche perché punti di concentrazione degli sforzi (dove anche la continuità della geometria TPMS si interrompe) hanno fatto decidere di non simulare meccanicamente questi provini. Scelti quindi i provini si è proceduto a simularli come descritto precedentemente.

4.1 Analisi della simulazione di stampa

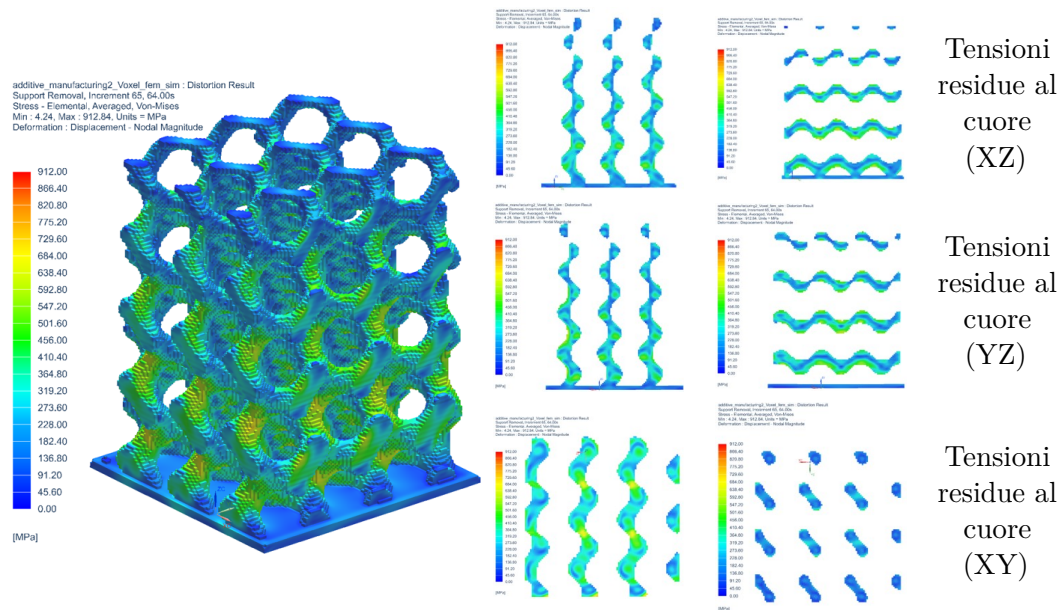


Figura 4.2: Risultato simulazione di stampa Giroide Skeltal Graded

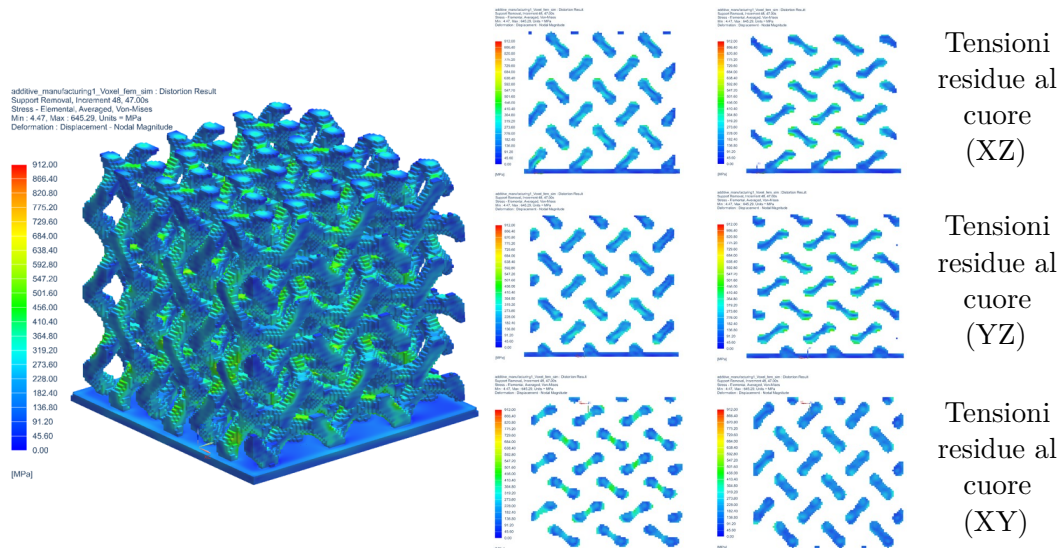


Figura 4.3: Risultato simulazione di stampa Giroide Skeltal Double

Capitolo 4 Analisi dei risultati

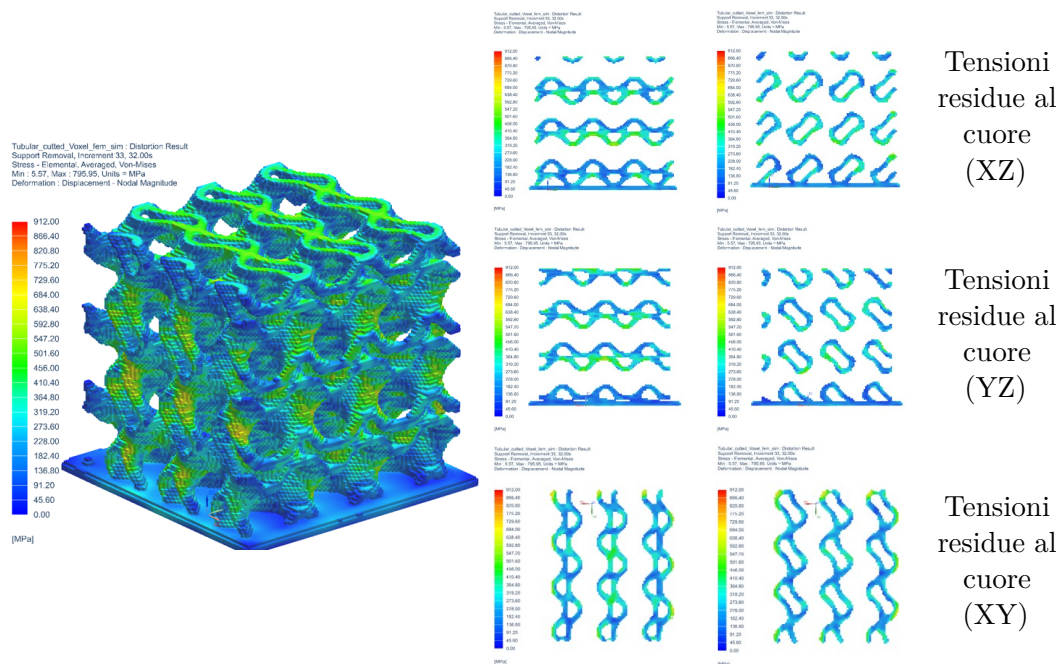


Figura 4.4: Risultato simulazione di stampa Giroide Tubular

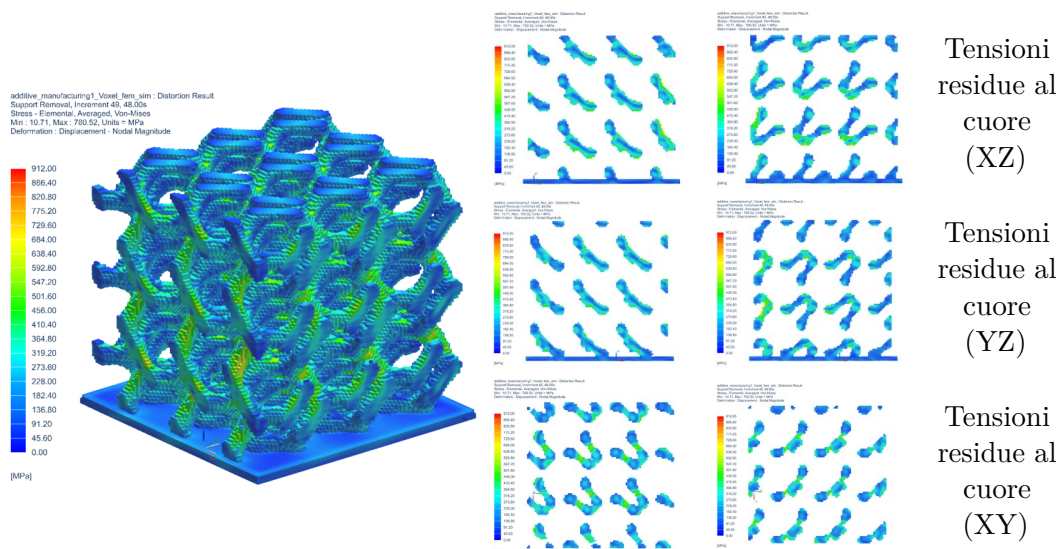


Figura 4.5: Risultato simulazione di stampa Giroide Skeltal Double Connected

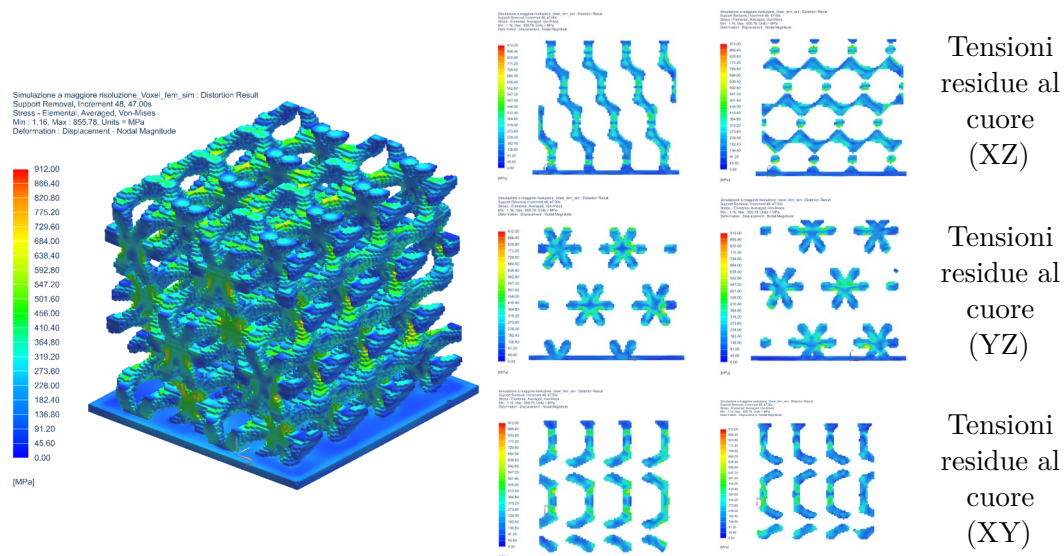


Figura 4.6: Risultato simulazione di stampa Giroide Skelatal Double Connected Vertical

4.2 Analisi della simulazione meccanica

Veniamo adesso all'analisi della simulazione, l'aspetto indubbiamente più importante di questo lavoro. Le strutture presentano proprietà diverse e spiccati comportamenti diversi. Partiamo con il definire come i dati ottenuti dalla simulazione sono stati utilizzati: i valori di forza e spostamento ottenuti, sono quelli del piatto mobile ovvero la sua traslazione verso il basso e la forza di reazione vincolare al contatto con il provino (dato che era vincolato a muoversi a 50m/s questa rappresenta la forza di contrasto del provino). Questi dati sono poi stati normalizzati (ne si è preso il valore assoluto) e successivamente trasformati in parametri ingegneristici: lo spostamento del pistone, essendo questo vincolato in contatto, è anche lo schiacciamento del provino quindi dividere lo spostamento per l'altezza del provino (che è quello tagliato di altezza 17,7 mm) ci da la deformazione relativa ϵ . La reazione vincolare poi, per lo stesso motivo precedente, è la forza di compressione del provino che ci consente di calcolare la tensione. In questo caso la geometria complessa del provino non ci interessa, questo viene trattato come un materiale unico e dalle caratteristiche peculiari (un meta materiale) e si studiano le sue proprietà "a scatola chiusa". Per questo motivo dividendo i valori di forza ottenuti per l'area di ingombro del provino (ossia $18 \cdot 19 = 342$) otteniamo i valori di sforzo σ . Definito come i valori sono stati ottenuti adesso osserviamo il comportamento delle varie strutture. Partiamo con l'analizzare la struttura Skeletal di riferimento. In Figura 4.7 abbiamo il grafico sforzo-deformazione ottenuto dai dati ricavati nei passaggi precedenti, il termine Velocity sta ad indicare che sono i valori di forza a cui è stato dato il vincolo velocity (ossia il piatto in movimento): quelli dell'altro piatto (fixed) a causa di fenomeni inerziali sul corpo seguono in ritardo quelli velocity oltre ad avere maggiori

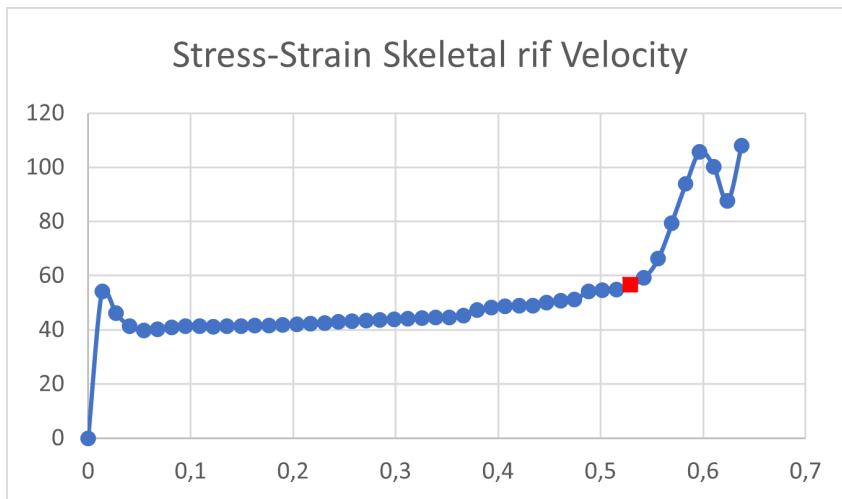


Figura 4.7: Grafico sforzo-deformazione della struttura Skeletal di riferimento

fluttuazioni, Quello che si può desumere dall'andamento grafico è che la struttura abbia un comportamento principalmente bend-dominated come ci si aspettava: il picco iniziale è dovuto solo a dei fenomeni inerziali di risposta tardiva che si creano perché la simulazione è stata fatta ad alta velocità, quindi in realtà quel picco non sarebbe presente normalmente (infatti dura solo pochi punti); l'aspetto che più la avvicina alla bend-dominated è il lungo plateau prima di raggiungere la densificazione. Da questa situazione possiamo definire una base comparativa per le altre strutture. Partiamo innanzitutto dalla Skeletal Double il cui grafico stress-strain è mostrato in Figura 4.8. Possiamo notare come l'andamento sia simile a quello della skeletal standard ma una cosa salta subito all'occhio, i valori. Infatti la cosa che non avevamo previsto era questa scarsa resistenza della struttura. D'altronde la quantità

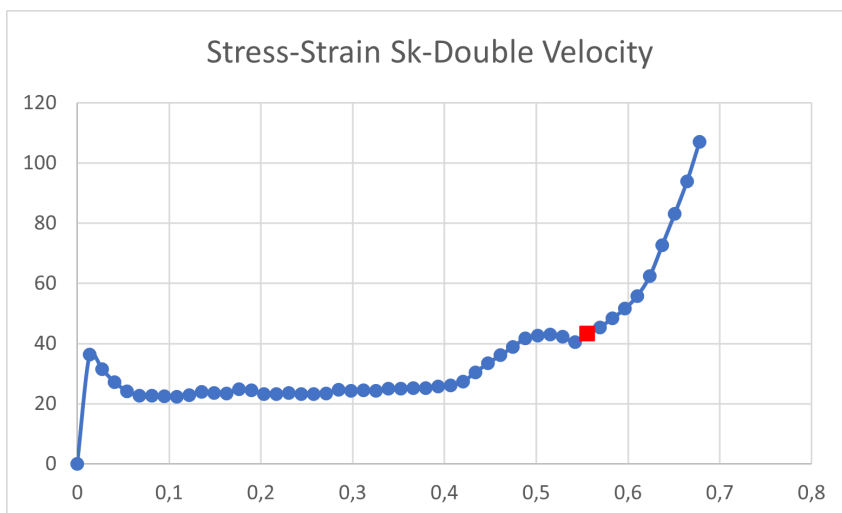


Figura 4.8: Grafico sforzo-deformazione della struttura Skeletal Double

di materiale è la stessa, la geometria di resistenza è la stessa (di fatto sono due

reticoli giroide classici che collaborano) però non avevamo messo in conto una cosa: il comportamento bend-dominated. Perché le strutture skeltal comportandosi in questo modo sono sottoposte molto a carichi di flessione e questo ha evidentemente mandato in instabilità molto facilmente le due strutture più fine rispetto a quella singola più grande. Non si conosce la natura di quella gobba prima dell'inizio della densificazione, dovrà essere confermata e valutata da prove empiriche. Passiamo poi alla struttura Graded il cui grafico è mostrato in Figura ??.

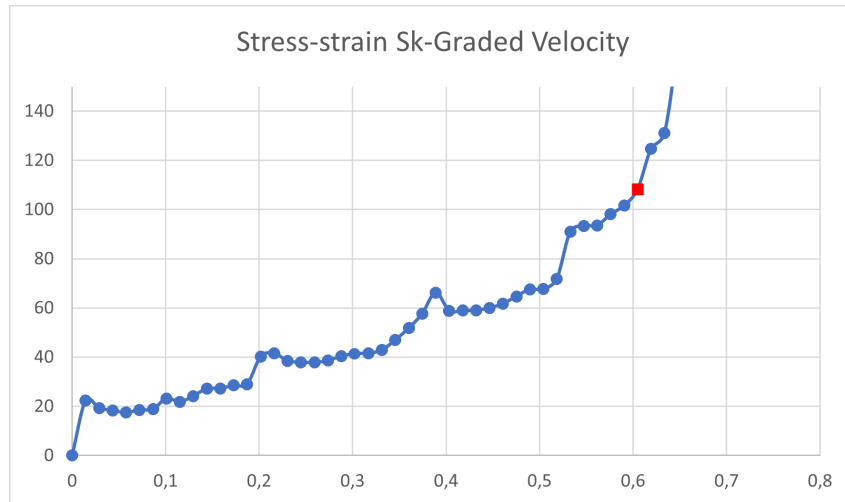


Figura 4.9: Grafico sforzo-deformazione della struttura Skeletal Double

ha un andamento alquanto singolare ma trova riscontro con altri studi trovati in letteratura. Questo comportamento è dato dal progressivo cedimento della gradazione da più sottile a più spessa: la struttura collassa in maniera irregolare tendendo quasi a dividersi in strati di densificazione piuttosto che ad avere un andamento regolare. Si può vedere infatti come ogni tratto non abbia un cedimento istantaneo ma aumenti la propria rigidità come in fase di densificazione.

In fine in Figura 4.10 è mostrato il grafico della struttura Tubular. Questa si nota subito essere la struttura più performante di tutte. Molto bassa risposta inerziale e poche fluttuazioni la rendono una struttura che può competere con quelle altoperformanti. La parte interessante è vedere come il plateau sia in effetti leggermente crescente. In realtà questa struttura è affine a livello geometrico alle Sheet infatti si è visto che la matematica che c'è dietro consente anche, per certi isovalori, di ottenere proprio una struttura Sheet. La cosa si può notare anche dalle sue sezioni mostrate nel capitolo precedente, la struttura in alcuni punti si tocca e si fonde con se stessa assumendo in parte il comportamento stretch-dominated. Oltre questa similitudine geometrica c'è un vantaggio importante: essendo comunque una struttura skeletal di principio il suo comportamento di base è quello di una bend-dominated, questo fa sì quindi che la struttura lavori per la maggior parte a flessione; essendo questa una struttura tubulare, però, avere la stessa densità delle altre, pur avendo una cavità al suo interno, implica che il diametro della geometria esterna sia maggiore comparato

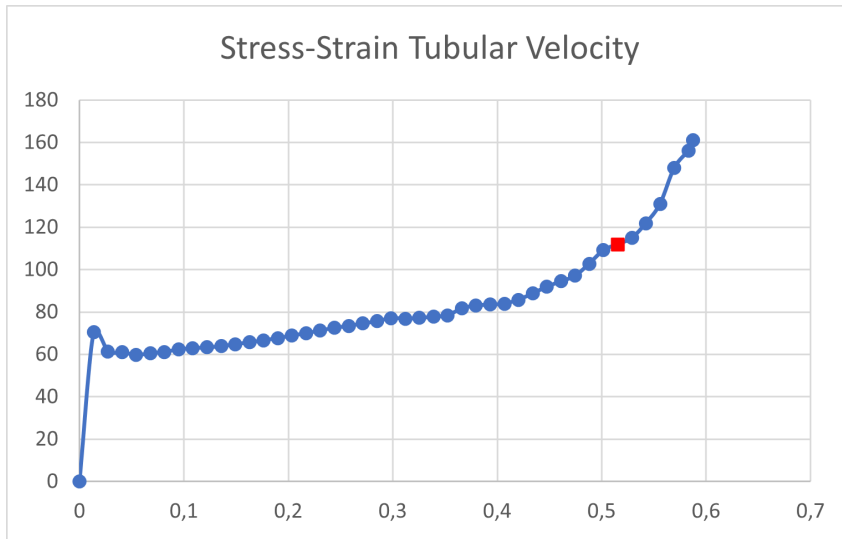


Figura 4.10: Grafico sforzo-deformazione della struttura Skeletal Double

con provini della stessa densità: questo comporta una maggiore capacità di resistenza a flessione.

Un aspetto interessante è come le curve siano arrivate a densificazione: nonostante il tempo inserito per la simulazione fosse inferiore a quello della densificazione vediamo comunque in tutte una crescita rapida della tensione, significando che la struttura è collassata su se stessa. Il perché di questo fenomeno anticipato è dato dalla conformazione della giroide stessa: quando la struttura collassa tende a schiacciare in due dimensioni il reticolo; questo fa sì che le varie strut rimangano stabili e collassino quai perfettamente su se stesse. Per questo motivo i campioni finali presentano comunque delle porosità, la struttura cioè si comporta come a densificazione prima di essere arrivata al 100%.

Con una densificazione prematura come questa bisogna mettere un punto per prendere i valori utili pre-collasso. Una metodologia valida è quella che utilizza l'energy absorption efficiency. Questa è definita come

$$\eta(\varepsilon) = \frac{1}{\sigma(\varepsilon)} \int_0^\varepsilon \sigma(\varepsilon) d\varepsilon$$

dove σ e ε sono lo stress di compressione e la deformazione nominale rispettivamente. Secondo questo approccio, la densificazione effettiva inizia (o finisce la fase di plateau) quando η raggiunge il massimo, ossia quando

$$\left. \frac{d\eta(\varepsilon)}{d\varepsilon} \right|_{\varepsilon=\varepsilon_{d0}}$$

Come esempio per questo lavoro possiamo osservare in Figura 4.11 il grafico della Skeletal giroide. Qui è possibile vedere chiaramente il punto di massimo e come dopo questo l'andamento cambi drasticamente. Il punto rosso rappresenta proprio

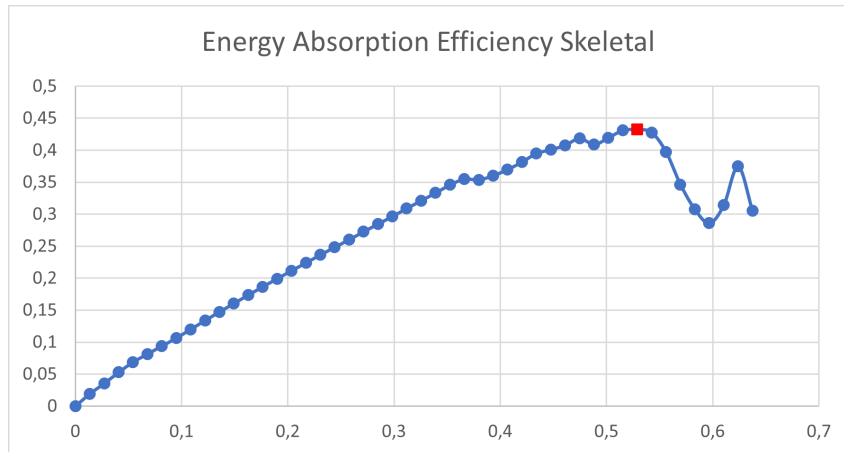


Figura 4.11: Grafico sforzo-deformazione della struttura Skeletal Double

il massimo ed è lo stesso che (concordemente per ogni struttura) è stato marcato su tutti gli altri grafici. Questo metodo funziona molto bene perché considera l'alto assorbimento energetico delle strutture lattice: dato che sono semi vuote queste presentano una deformazione molto elevata rispetto ad un provino pieno. Essendo l'energia assorbita per unità di volume l'area sottesa dalla curva stress-strain, quando questa area comincia a diminuire rispetto alla tensione (ossia si è raggiunto il massimo) vuol dire che il campione si sta iniziando a comportare come un provino pieno.

Un altro aspetto emerso da questa analisi è la capacità di assorbimento di energia: nelle tabelle da 4.1 a 4.4 sono mostrati i valori associati di energia per unità di volume e la SEA (Specific Energy Absorption). Da questi valori (in particolare il volume d'ingombro e quello del materiale fornito da Ansys) emerge come i provini con il piattino e tagliati per velocizzare la simulazione non abbiano più una densità del 20%, però queste essendo simulazioni devo comunque dare risultati orientativi che vanno poi verificati con prove sperimentali dove possono anche emergere molti effetti che nelle simulazioni non possono essere previsti.

volume ingombro	6.053,4	mm^3
volume di materiale	1377,83	mm^3
Energia assorbita	16,067	mJ/mm^3
	0,09726	kJ
massa	0,01102	kg
SEA	8,8237	kJ/kg
Densification strain	0,55593	mm/mm
Plateau stress	28,148	MPa

Tabella 4.1: Dati sull'assorbimento di energia SD

volume ingombro	6.053,4	mm^3
volume di materiale	1395,0	mm^3
Energia assorbita	24,463	mJ/mm^3
	0,14808	kJ
massa	0,01116	kg
SEA	13,2698	kJ/kg
Densification strain	0,5288	mm/mm
Plateau stress	45,273	MPa

Tabella 4.2: Dati sull'assorbimento di energia Skeletal di riferimento

volume ingombro	8.550,0	mm^3
volume di materiale	1799,5	mm^3
Energia assorbita	30,784	mJ/mm^3
	0,26320	kJ
massa	0,0143	kg
SEA	18,283	kJ/kg
Densification strain	0,6048	mm/mm
Plateau stress	22,786	MPa
	41,616	
	63,108	
	97,667	

Tabella 4.3: Dati sull'assorbimento di energia Graded

volume ingombro	6.053,4	mm^3
volume di materiale	1418,2	mm^3
Energia assorbita	40,080	mJ/mm^3
	0,2426	kJ
massa	0,01134	kg
SEA	21,384	kJ/kg
Densification strain	0,5152	mm/mm
Plateau stress	76,436	MPa

Tabella 4.4: Dati sull'assorbimento di energia Tubular

Capitolo 5

Conclusione

5.1 Considerazioni finali

Questo lavoro ha esplorato lo sviluppo di geometrie complesse tramite modelli computazionali, da stampare con tecnologie di Additive Manufacturing metallica utilizzando come materiale l'AISI 316L. L'analisi delle superfici minime e delle Triply Periodic Minimal Surfaces (TPMS) ha permesso di selezionare celle base ad alte prestazioni, successivamente modellate e simulate utilizzando software avanzati come Metafold 3D, Siemens NX e Ansys.

Le simulazioni di stampa hanno evidenziato diverse criticità e punti di forza delle geometrie studiate. La struttura Skeletal Double ha mostrato una buona capacità di smaltimento del calore grazie alla distribuzione del materiale in due volumi distinti, mentre la struttura Graded ha presentato una distribuzione delle tensioni residue variabile lungo la lunghezza, coerente con le differenze di spessore. La struttura Tubular, sebbene abbia mostrato punti di concentrazione del calore, ha evidenziato una buona stabilità complessiva.

Dal punto di vista meccanico, le simulazioni di compressione hanno rivelato comportamenti distinti tra le diverse geometrie. La struttura Skeletal di riferimento ha mostrato un comportamento principalmente bend-dominated, con un lungo plateau prima della densificazione. La Skeletal Double, pur avendo una geometria simile, ha evidenziato una resistenza inferiore, probabilmente a causa della maggiore instabilità delle strutture più sottili. La struttura Graded ha mostrato un comportamento irregolare, con cedimenti progressivi lungo la gradazione. Infine, la struttura Tubular si è distinta per le sue elevate prestazioni, con un plateau leggermente crescente e una buona resistenza a flessione, grazie alla sua conformazione tubolare.

Altre strutture, come la Skeletal Double Connected e la sua variante Vertical, sono state accantonate in corso d'opera a causa di problematiche di stampabilità e accumulo di calore ai nodi, che avrebbero potuto compromettere la loro integrità meccanica. La Skeletal Double Connected Horizontal, in particolare, ha presentato problemi di overhang significativi, rendendo la sua stampa non praticabile senza ulteriori modifiche.

5.2 Sviluppi futuri

L'applicazione di queste strutture e metodologie di realizzazione per le superfici TPMS può essere espansa non solo alla geometria giroide trattata in questo lavoro. I principi di modellazione, infatti, risultano gli stessi a prescindere dalla geometria di base che viene scelta. Oltre questo è possibile combinarli tra loro, per esempio realizzare una geometria Tubular e Skeletal Double allo stesso tempo creando due reticoli tubolari che si intrecciano senza incrociarsi, oppure è possibile creare una geometria Skeletal Double Graded che magari abbia una gradazione opposta in direzione nelle due metà della sua struttura. Queste metodologie possono essere anche applicate su strutture diverse e magari cambiare drasticamente le relazioni di performance tra di loro. Oltre questo altre strategie di manipolazione della struttura non sono state utilizzate, ad esempio la variazione delle dimensioni delle celle, realizzando un tipo diverso di gradazione (non con l'isovalore ma con la dimensione della cella appunto) oppure cambiando il rapporto d'aspetto della geometria (ad esempio facendo diventare la cella unitaria da cubo a parallelepipedo). In aggiunta alla realizzazione i provini potrebbero subire ulteriori alterazioni per migliorarne le proprietà meccaniche. Per esempio la struttura tubular una volta stampata potrebbe essere riempita con del materiale polimerico per darle più stabilità. Per di più sarebbe possibile cambiare anche ramo di ricerca, la geometria tubular potrebbe per esempio essere valutata come scambiatore di calore. In conclusione, studio ha dimostrato come l'integrazione di tecniche di modellazione computazionale e simulazioni avanzate possa ottimizzare significativamente i processi di additive manufacturing realizzando geometrie varie sia in forma che in proprietà. Le geometrie complesse sviluppate rispondono alle esigenze di personalizzazione e alte prestazioni richieste in settori come l'aerospaziale e l'automotive, aprendo nuove possibilità per l'innovazione nei metamateriali.

Bibliografia

- [1] Xiaojie Fan, Qian Tang, Qixiang Feng, Shuai Ma, Jun Song, Mengxia Jin, Fuyu Guo, and Peng Jin. Design, mechanical properties and energy absorption capability of graded-thickness triply periodic minimal surface structures fabricated by selective laser melting. *International Journal of Mechanical Sciences*, 204:106586, 2021.
- [2] I. Maskery, N.T. Aboulkhair, A.O. Aremu, C.J. Tuck, and I.A. Ashcroft. Compressive failure modes and energy absorption in additively manufactured double gyroid lattices. *Additive Manufacturing*, 16:24–29, 2017.
- [3] Lei Zhang, Stefanie Feih, Stephen Daynes, Shuai Chang, Michael Yu Wang, Jun Wei, and Wen Feng Lu. Energy absorption characteristics of metallic triply periodic minimal surface sheet structures under compressive loading. *Additive Manufacturing*, 23:505–515, 2018.
- [4] Essam Ibrahim Sohaib Z. Khan, S. H. Masood and Zaini Ahmad. Compressive behaviour of neovius triply periodic minimal surface cellular structure manufactured by fused deposition modelling. *Virtual and Physical Prototyping*, 14(4):360–370, 2019.
- [5] Amanda Rossi de Oliveira, Anibal de Andrade Mendes Filho, Mohammad Masoumi, and Erik Gustavo Del Conte. Compression and energy absorption of maraging steel primitive scaffolds produced by powder bed fusion. *The International Journal of Advanced Manufacturing Technology*, 116(3):1271–1283, Sep 2021.
- [6] Hanfeng Yin, Xianjun Zheng, Guilin Wen, Chao Zhang, and Zhantao Wu. Design optimization of a novel bio-inspired 3d porous structure for crashworthiness. *Composite Structures*, 255:112897, 2021.
- [7] Chenxi Peng, Kate Fox, Ma Qian, H. Nguyen-Xuan, and Phuong Tran. 3d printed sandwich beams with bioinspired cores: Mechanical performance and modelling. *Thin-Walled Structures*, 161:107471, 2021.
- [8] Toufik Outtas Ebrahim Ahmed Ali Alkebsi, Hacene Ameddah and Abdallah Almutawakel. Design of graded lattice structures in turbine blades using topology optimization. *International Journal of Computer Integrated Manufacturing*, 34(4):370–384, 2021.

Bibliografia

- [9] A. Jiansheng Pan, B. Jianwei Wu, C. Yin Zhang, D. Hui Wang, and E. Jiubin Tan. Design and analyze of flexure hinges based on triply periodic minimal surface lattice. *Precision Engineering*, 68:338–350, 2021.
- [10] Waiel Elmadih, Wahyudin P. Syam, Ian Maskery, Dimitrios Chronopoulos, and Richard Leach. Mechanical vibration bandgaps in surface-based lattices. *Additive Manufacturing*, 25:421–429, 2019.
- [11] Zahid Ahmed Qureshi, Emad Elnajjar, Oraib Al-Ketan, Rashid Abu Al-Rub, and Salah Burhan Al-Omari. Heat transfer performance of a finned metal foam-phase change material (fmf-pcm) system incorporating triply periodic minimal surfaces (tpms). *International Journal of Heat and Mass Transfer*, 170:121001, 2021.
- [12] Inderjot Kaur and Prashant Singh. Flow and thermal transport characteristics of triply-periodic minimal surface (tpms)-based gyroid and schwarz-p cellular materials. *Numerical Heat Transfer, Part A: Applications*, 79(8):553–569, 2021.
- [13] Oraib Al-Ketan, Mohamed Ali, Mohamad Khalil, Reza Rowshan, Kamran A Khan, and Rashid K Abu Al-Rub. Forced convection computational fluid dynamics analysis of architected and three-dimensional printable heat sinks based on triply periodic minimal surfaces. *J. Therm. Sci. Eng. Appl.*, 13(2):1–33, April 2021.
- [14] Reza Attarzadeh, Marc Rovira, and Christophe Duwig. Design analysis of the “schwartz d” based heat exchanger: Anumerical study. *International Journal of Heat and Mass Transfer*, 177:121415, 2021.
- [15] Srinivasan Rajagopalan and Richard A. Robb. Schwarz meets schwann: Design and fabrication of biomorphic and durataxic tissue engineering scaffolds. *Medical Image Analysis*, 10(5):693–712, 2006. The Eighth International Conference on Medical Imaging and Computer Assisted Intervention – MICCAI 2005.
- [16] Z Gorgin Karaji, M Speirs, S Dadbakhsh, J-P Kruth, H Weinans, A A Zadpoor, and S Amin Yavari. Additively manufactured and surface biofunctionalized porous nitinol. *ACS Appl. Mater. Interfaces*, 9(2):1293–1304, January 2017.
- [17] Davar Ali. Effect of scaffold architecture on cell seeding efficiency: A discrete phase model cfd analysis. *Computers in Biology and Medicine*, 109:62–69, 2019.
- [18] Elham Davoodi, Hossein Montazerian, Ali Khademhosseini, and Ehsan Toyserkani. Sacrificial 3d printing of shrinkable silicone elastomers for enhanced feature resolution in flexible tissue scaffolds. *Acta Biomaterialia*, 117:261–272, 2020.

- [19] E F Lehder, I A Ashcroft, R D Wildman, L A Ruiz-Cantu, and I Maskery. A multiscale optimisation method for bone growth scaffolds based on triply periodic minimal surfaces. *Biomech. Model. Mechanobiol.*, 20(6):2085–2096, December 2021.
- [20] Ihsan UIIah, Lei Cao, Wei Cui, Qian Xu, Rui Yang, Kang lai Tang, and Xing Zhang. Stereolithography printing of bone scaffolds using biofunctional calcium phosphate nanoparticles. *Journal of Materials Science & Technology*, 88:99–108, 2021.
- [21] Li-Ya Zhu, Lan Li, Zong-An Li, Jian-Ping Shi, Wen-Lai Tang, Ji-Quan Yang, and Qing Jiang. Design and biomechanical characteristics of porous meniscal implant structures using triply periodic minimal surfaces. *J. Transl. Med.*, 17(1):89, March 2019.
- [22] Jorge Corona-Castuera, Daniela Rodriguez-Delgado, John Henao, Juan Carlos Castro-Sandoval, and Carlos A Poblano-Salas. Design and fabrication of a customized partial hip prosthesis employing CT-scan data and lattice porous structures. *ACS Omega*, 6(10):6902–6913, March 2021.
- [23] Meng-Ting Hsieh, Matthew R. Begley, and Lorenzo Valdevit. Architected implant designs for long bones: Advantages of minimal surface-based topologies. *Materials & Design*, 207:109838, 2021.
- [24] Dalia Mahmoud, Kassim S. Al-Rubaie, and Mohamed A. Elbestawi. The influence of selective laser melting defects on the fatigue properties of ti6al4v porosity graded gyroids for bone implants. *International Journal of Mechanical Sciences*, 193:106180, 2021.
- [25] Elham Davoodi, Hossein Montazerian, Reza Esmailizadeh, Ali Ch Darabi, Armin Rashidi, Javad Kadkhodapour, Hamid Jahed, Mina Hoorfar, Abbas S Milani, Paul S Weiss, Ali Khademhosseini, and Ehsan Toyserkani. Additively manufactured gradient porous Ti-6Al-4V hip replacement implants embedded with cell-laden gelatin methacryloyl hydrogels. *ACS Appl. Mater. Interfaces*, 13(19):22110–22123, May 2021.
- [26] Enrique Alabort, Daniel Barba, and Roger C. Reed. Design of metallic bone by additive manufacturing. *Scripta Materialia*, 164:110–114, 2019.
- [27] Hong-Yuan Lei, Jing-Rong Li, Qing-Hui Wang, Zhi-Jia Xu, Wei Zhou, Chang-Lin Yu, and Tian-Qing Zheng. Feasibility of preparing additive manufactured porous stainless steel felts with mathematical micro pore structure as novel catalyst support for hydrogen production via methanol steam reforming. *International Journal of Hydrogen Energy*, 44(45):24782–24791, 2019.

Bibliografía

- [28] Francisco M Baena-Moreno, Miriam González-Castaño, Juan Carlos Navarro de Miguel, Kamal U M Miah, Ralf Ossenbrink, Jose Antonio Odriozola, and Harvey Arellano-García. Stepping toward efficient microreactors for CO₂ methanation: 3d-printed gyroid geometry. *ACS Sustain. Chem. Eng.*, 9(24):8198–8206, June 2021.
- [29] Tim Femmer, Alexander J.C. Kuehne, and Matthias Wessling. Estimation of the structure dependent performance of 3-d rapid prototyped membranes. *Chemical Engineering Journal*, 273:438–445, 2015.
- [30] Felix Hesselmann, Nils Scherenberg, Patrick Bongartz, Suzana Djeljadini, Matthias Wessling, Christian Cornelissen, Thomas Schmitz-Rode, Ulrich Steinseifer, Sebastian V. Jansen, and Jutta Arens. Structure-dependent gas transfer performance of 3d-membranes for artificial membrane lungs. *Journal of Membrane Science*, 634:119371, 2021.
- [31] J. G. Werner, G. G. Rodríguez-Calero, H. D. Abruña, and U. Wiesner. Block copolymer derived 3-d interpenetrating multifunctional gyroidal nanohybrids for electrical energy storage. *Energy Environ. Sci.*, 11:1261–1270, 2018.
- [32] Kahyun Hur, Richard G Hennig, and Ulrich Wiesner. Exploring periodic bicontinuous cubic network structures with complete phononic bandgaps. *J. Phys. Chem. C Nanomater. Interfaces*, 121(40):22347–22352, October 2017.
- [33] Chee Kai Chua Wenjing Yang, Jia An and Kun Zhou. Acoustic absorptions of multifunctional polymeric cellular structures based on triply periodic minimal surfaces fabricated by stereolithography. *Virtual and Physical Prototyping*, 15(2):242–249, 2020.
- [34] Yurun Feng, Xue Guo, Kai Huang, Hamada Elsayed, Giorgia Franchin, Hongyu Gong, and Paolo Colombo. Enhanced electromagnetic microwave absorption of sic ceramics targeting the integration of structure and function. *Journal of the European Ceramic Society*, 41(13):6393–6405, 2021.
- [35] Caroline Pouya, Johannes T B Overvelde, Mathias Kolle, Joanna Aizenberg, Katia Bertoldi, James C Weaver, and Pete Vukusic. Characterization of a mechanically tunable gyroid photonic crystal inspired by the butterfly *Parides sesostris*. *Adv. Opt. Mater.*, 4(1):99–105, January 2016.
- [36] Dong-Jin Yoo. Computer-aided porous scaffold design for tissue engineering using triply periodic minimal surfaces. *Int. J. Precis. Eng. Manuf.*, 12(1):61–71, February 2011.
- [37] J.A. Harris and G.J. McShane. Metallic stacked origami cellular materials: Additive manufacturing, properties, and modelling. *International Journal of Solids and Structures*, 185-186:448–466, 2020.

- [38] Chunze Yan, Liang Hao, Ahmed Hussein, and David Raymont. Evaluations of cellular lattice structures manufactured using selective laser melting. *International Journal of Machine Tools and Manufacture*, 62:32–38, 2012.
- [39] Xiaojie Fan, Qian Tang, Qixiang Feng, Shuai Ma, Jun Song, Mengxia Jin, Fuyu Guo, and Peng Jin. Design, mechanical properties and energy absorption capability of graded-thickness triply periodic minimal surface structures fabricated by selective laser melting. *International Journal of Mechanical Sciences*, 204:106586, 2021.
- [40] Lei Yang, Chunze Yan, Wenchao Cao, Zhufeng Liu, Bo Song, Shifeng Wen, Cong Zhang, Yusheng Shi, and Shoufeng Yang. Compression–compression fatigue behaviour of gyroid-type triply periodic minimal surface porous structures fabricated by selective laser melting. *Acta Materialia*, 181:49–66, 2019.
- [41] Ehsan Latif and Ramviyas Parasuraman. On the intersection of computational geometry algorithms with mobile robot path planning. *Algorithms*, 16(11), 2023.
- [42] Oraib Al-Ketan, Reza Rowshan, and Rashid K. Abu Al-Rub. Topology-mechanical property relationship of 3d printed strut, skeletal, and sheet based periodic metallic cellular materials. *Additive Manufacturing*, 19:167–183, 2018.
- [43] I. Maskery, N.T. Aboulkhair, A.O. Aremu, C.J. Tuck, and I.A. Ashcroft. Compressive failure modes and energy absorption in additively manufactured double gyroid lattices. *Additive Manufacturing*, 16:24–29, 2017.