



**UNIVERSITÀ
DEGLI STUDI
DI PADOVA**



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

**CORSO DI LAUREA MAGISTRALE IN
BIOINGEGNERIA INDUSTRIALE**

**MODELLAZIONE VOLUMETRICA DI STRUTTURE RETICOLARI A
GRADIENTE DI POROSITÀ PER APPLICAZIONI BIOMEDICHE**

**Relatore: Prof. Gianpaolo Savio
Correlatori: Prof. ssa Francesca Uccheddu
Ing. Luca Grigolato**

Laureanda: Silvia Canton

ANNO ACCADEMICO 2021 – 2022

Data di laurea 13 dicembre 2022

Abstract

In questo lavoro di tesi si presentano delle strutture innovative nel panorama delle applicazioni biomediche: si tratta delle strutture reticolari a gradiente di porosità.

Studiando gli approcci di modellazione dei software esistenti, che per la maggior parte si basano sulla rappresentazione dei confini, si esplorano *plug-in* che permettono di estendere le loro funzionalità, consentendo la modellazione volumetrica di queste strutture che, grazie alla loro architettura in grado di distribuire all'interno di tutto il volume le proprietà meccaniche, fisiche e chimiche, offrono numerosissimi vantaggi per superare i limiti dei materiali tradizionali.

L'obiettivo del presente elaborato è proporre un metodo per creare strutture reticolari con gradiente di porosità, raggiungibile attraverso algoritmi di modellazione volumetrici.

The work presented in this thesis consists in the introduction of innovative structures in the biomedical field: it is about functionally graded lattice structures with porosity gradient.

By studying different modelling approaches, on which today's software are based, plug-ins that allow volumetric modelling are chosen, enabling the volumetric modelling of these structures which show many advantages to overcome the limits of traditional materials, thanks to their architecture capable of distributing the mechanical, physical and chemical properties within their entire volume.

The aim of this paper is to propose a method to create lattice structures with porosity gradient, through volumetric modelling algorithms.

Indice

INTRODUZIONE	7
CAPITOLO 1 – Introduzione ai Functionally Graded Materials	11
1.1 Cenni storici.....	12
1.2 Functionally Graded Materials in natura	12
1.3 Tipi di FGMs	14
1.4 Applicazioni biomediche	17
1.5 Strutture reticolari.....	20
1.6 Approcci di modellazione.....	28
1.6.1 Boundary-Representation (B-Rep)	28
1.6.2 Constructive Solid Geometry (CSG).....	30
1.6.3 Volumetric-Representation (V-Rep).....	31
CAPITOLO 2 – Strumenti e metodi per la progettazione di Functionally Graded Lattice Structures	39
2.1 Strumenti: plug-in di Grasshopper per la modellazione volumetrica	40
2.1.1 Dendro	41
2.1.2 Axolotl	45
2.1.3 Monolith.....	49
2.2 Metodo di progettazione delle FGLSs	53
CAPITOLO 3 – Risultati e discussione	85
3.1 FGLSs ottenute	86
3.2 Limiti riscontrati	94
CONCLUSIONI E DIREZIONI FUTURE	99
APPENDICE A	102
APPENDICE B	106
BIBLIOGRAFIA	114

Introduzione

Le strutture reticolari sono conosciute per le loro prestazioni superiori in termini di riduzione del peso, rigidità e producibilità, ma quando i materiali devono essere adattivi, multifunzionali e con proprietà regolabili, le strutture reticolari a gradiente funzionale, note come *Functionally Graded Lattice Structures* (FGLS), svolgono un ruolo cruciale in molti settori. In ambito biomedicale il settore che trae maggior vantaggio dall'utilizzo di queste strutture è quello degli impianti ortopedici e dentali.

Questi materiali, grazie alla loro particolare architettura, permettono di replicare in modo fedele le caratteristiche strutturali e meccaniche dei tessuti nativi e offrono quindi un grande potenziale per superare i limiti che portano al fallimento degli attuali impianti ortopedici e migliorare allo stesso tempo l'osteointegrazione e la resistenza all'usura.

La maggior parte dei software di modellazione esistenti si basa sulla rappresentazione dei confini degli oggetti 3D modellati. Si tratta di un metodo consolidato, ma alla cui base vi è un'assunzione fondamentale: il materiale presente all'interno è omogeneo. Oggi è quantomai necessario essere in grado di esplorare le caratteristiche dell'intero volume dell'oggetto, superando i limiti dei sistemi CAD tradizionali.

È per questo motivo che in questo lavoro di tesi si vuole presentare un metodo per la realizzazione di strutture reticolari graduate attraverso l'implementazione di algoritmi di modellazione volumetrica. Si mostrerà come la modellazione volumetrica è in grado di fornire le informazioni necessarie a rappresentare un oggetto che occupa uno spazio tridimensionale, fornendo la descrizione sia della superficie, sia della parte interna. Verrà illustrato come si sviluppa un flusso di lavoro di progettazione, a partire dalla scelta della cella unitaria e dalla definizione dei parametri della struttura, caratteristiche che influenzano in maniera significativa il risultato e la performance della struttura finale.

In un contesto di limitatezza di programmi predisposti per questo tipo di applicazioni volumetriche, si esplora un avanzato software di modellazione algoritmica per la generazione e il controllo di forme semplici e complesse: Grasshopper, *plug-in* direttamente integrato nel software CAD commerciale per la modellazione 3D Rhinoceros 7. Ciò che lo rende un linguaggio di programmazione estremamente versatile è la possibilità di usufruire di *plug-in* gratuiti, disponibili sulla piattaforma di applicativi per *Rhinoceros* e *Grasshopper*

“*food4Rhino*”. Di interesse per lo scopo di questa tesi sono i *plug-in* recentemente introdotti per la modellazione volumetrica, che verranno presentati e utilizzati nella progettazione delle FGLSs.

La realizzazione del gradiente avverrà attraverso varie strategie: le strutture implementate saranno tutte a gradiente di porosità, ovvero variando la densità relativa del materiale costituente.

Il metodo di progettazione che verrà messo a punto ben si presta alle moderne tecniche di produzione additiva in 3D, che permettono la fabbricazione di strutture complesse con una distribuzione precisa e graduale del materiale nelle tre dimensioni dello spazio.

Il presente elaborato è stato strutturato in tre sezioni principali, a partire dalla comprensione del concetto di materiale a gradiente funzionale, per poter giungere alla presentazione del metodo atto a creare queste strutture. La parte applicativa nasce dalla volontà di investigare le nuove potenzialità offerte dalla modellazione volumetrica e di affrontare la sfida di creare materiali che abbiano una distribuzione graduale delle loro proprietà.

Il primo capitolo introduce i FGMs, da quando è stato sviluppato originariamente il concetto alla loro evoluzione nei vari tipi, con focus sulle caratteristiche dei materiali porosi, in quanto i più utilizzati nell’ambito biomedico. Si approfondisce la tassonomia delle strutture reticolari e gli approcci alla loro modellazione: si tratta di un passaggio introduttivo necessario per comprendere il metodo proposto. Infatti, l’approccio di modellazione scelto deriva dall’analisi della panoramica degli approcci esistenti e su cui si basano i software di modellazione, per comprenderne gli aspetti negativi e i limiti, e giustificare l’adozione di un approccio di tipo volumetrico come quello più adatto a raggiungere l’obiettivo prefissato.

Successivamente, nel secondo capitolo, si mostra il *workflow* del metodo che porta alla progettazione di strutture reticolari a gradiente di porosità, a seguito della presentazione degli strumenti software a disposizione per metterlo in pratica. In questo capitolo si determina la densità relativa di tutte le celle unitarie che costituiranno gli elementi base delle strutture reticolari che si andranno a realizzare. Il tipo di cella e la porosità svolgono ruoli cruciali nel determinare la rigidità e la resistenza strutturali, e hanno un ruolo importante anche nell’idoneità del reticolo per la stampa 3D, per conoscere quali sono i limiti di stampabilità.

Nel terzo capitolo si mostrano e si discutono i risultati ottenuti dall’applicazione del metodo agli algoritmi di modellazione, portando diversi esempi ottenuti dalla messa a punto di varie

strategie per la generazione del gradiente e sollevando alcuni problemi emersi in fase di progettazione.

Sebbene vi siano moltissimi aspetti benefici, vi è ancora una notevole mancanza di standard e linee guida per l'introduzione delle FGLSs nella pratica clinica. Nelle conclusioni si accenna a quelle che sono le sfide ancora aperte affinché si possa sfruttare al meglio il potenziale di queste strutture reticolari a gradiente di porosità.

Capitolo 1

Introduzione ai Functionally Graded Materials

I *Functionally Graded Materials*, noti con l'acronimo FGMs, sono materiali ingegneristici avanzati che contengono una variazione spaziale della composizione, della microstruttura o della porosità allo scopo specifico di controllare le variazioni delle proprietà termiche, strutturali o funzionali. Come indica il termine stesso, si fa riferimento non solo alla possibilità di dare una funzionalità al materiale, ma anche al fatto che tale funzionalità deve essere gradualmente variabile al suo interno.

Questa caratteristica li vantaggiosi da diversi punti di vista, rispetto ai materiali tradizionali. La Figura 1.1 mostra la differenza di composizione e proprietà tra un materiale composito convenzionale e un FGM. Nel primo caso esiste un'interfaccia distinta tra i due materiali che compongono la struttura, ma non nel secondo. Questa differenza corrisponde alla distribuzione delle proprietà fisiche, chimiche e meccaniche. Un materiale composito ordinario presenta una brusca variazione delle proprietà all'interfaccia, mentre un FGM presenta un cambiamento graduale al suo interno. Questa è la chiave per poter esplorare le potenzialità di questi materiali.

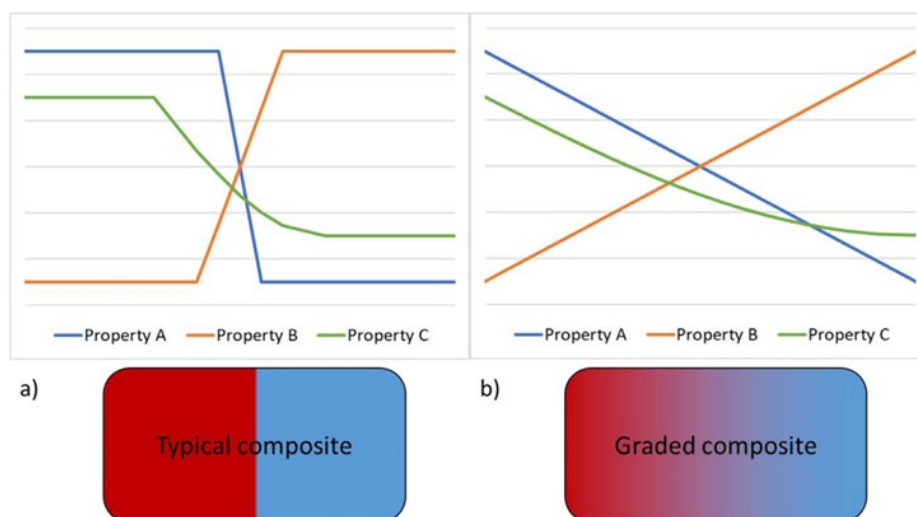


Figura 1.1. *Variazione delle proprietà tra un materiale composito tradizionale e graduato (dos Reis et al., 2020)*

Il concetto originale di FGM si proponeva di ridurre lo stress termico, ma ben presto è stato ampliato e oggi si riferisce a materiali con un cambiamento graduale nelle funzioni al loro interno, comprendendo non solo materiali resistenti al calore, ma anche materiali strutturali, biomateriali e semiconduttori.

1.1 Cenni storici

Il concetto di *Functionally Graded Material* è nato all'inizio degli anni '80 in Giappone. I ricercatori giapponesi si sono trovati ad affrontare un problema durante un progetto di veicolo spaziale riguardo a un tipo di materiale composito che fosse in grado di resistere a differenze di temperature molto alte. L'applicazione richiedeva che da un lato il materiale fosse sottoposto a una temperatura di circa 2000 K e che tale temperatura non dovesse essere trasferita all'altra parte. Il corpo del veicolo aveva dunque bisogno di un materiale composito che sarebbe stato esposto a un gradiente di temperatura di circa 1000 K tra l'interno e l'esterno.

I tradizionali materiali compositi non erano in grado di resistere a condizioni di lavoro così estreme. Per tale applicazione sono stati provati materiali compositi laminati convenzionali, ma senza successo. Ogni volta che il materiale composito tradizionale falliva, i punti di rottura si trovavano solitamente nello stesso sito e le modalità di guasto sono sempre risultate simili. Ciò avveniva per delaminazione, cioè separazione dei materiali compositi laminati dal sito in cui due materiali venivano uniti: questa è un'area ad elevato fattore di concentrazione delle sollecitazioni. La mancata corrispondenza delle proprietà dei due materiali, ovvero la discontinuità di questa interfaccia, era responsabile del cedimento. I ricercatori sapevano che se si fosse eliminata l'interfaccia brusca tra i due materiali, il problema si sarebbe risolto. Introducendo gradualmente il secondo materiale nel primo, il composito che si era così sviluppato era in grado di resistere per l'applicazione prevista: oggi questo tipo di materiale è noto come *Functionally Graded Material*.

Negli ultimi decenni sono stati creati diversi tipi di FGMs, tra cui a gradiente di materiale, microstrutturale, di porosità. Costituiscono infatti un importante campo di ricerca, con aree di applicazione in continua espansione.

1.2 Functionally Graded Materials in natura

Capita che l'ingegneria si rivolga con successo alla natura per cercare risposte a tutta una serie di domande quando cerca di risolvere problemi.

La maggior parte dei materiali prodotti dalla natura non sono monolitici, ma presentano strutture a gradiente, e sono ottimizzati per le condizioni di carico e gli ambienti di lavoro a cui

sono sottoposti. Questi materiali devono essere in grado di svolgere una varietà di funzioni, anche per un lungo periodo di tempo. Esistono infatti materiali come il legno, il bambù, le ossa e i denti che sono strutture a variazione graduale di proprietà.

La maggior parte dei tessuti e degli organi del corpo umano sono costituiti da FGMs. Esempi includono la pelle umana, complesso sistema strutturale multistrato costituito dall'epidermide, dal derma e dall'ipoderma. Ogni livello infatti ha proprietà diverse e svolge anche funzioni diverse. I denti hanno una superficie costituita da smalto duro, con cristalliti di idrossiapatite e un nucleo interno costituito da dentina. Viene infatti richiesta un'elevata resistenza all'usura nella parte esterna e una struttura interna duttile per ragioni di affaticamento e fragilità (Knoppers et al., 2004). Il passaggio dallo smalto alla dentina è fornito da strati la cui composizione cambia gradualmente. Il disco intervertebrale umano è un altro sofisticato esempio di sistema graduale nel corpo umano: si possono definire tre regioni all'interno del disco, *l'annulus fibrosus*, il nucleo polposo e le placche terminali della cartilagine (Pattappa et al., 2012). Un ulteriore esempio è fornito dalla giunzione osteocondrale, complessa struttura da cartilagine a osso che mostra gradazione nella matrice extracellulare, portando a variazioni nelle proprietà biomeccaniche e viscoelastiche insieme allo spessore per la transizione graduale dei carichi. Il tessuto osteocondrale degli adulti ha uno spessore di 3 mm. È costituito da cartilagine, uno strato di cartilagine calcificata e osso subcondrale. L'osso e la cartilagine sono due diversi tessuti con caratteristiche diverse su scale più piccole per quanto riguarda le proprietà chimiche, meccaniche, e strutturali. Il carico viene trasferito dalla cartilagine all'osso subcondrale e l'osso subcondrale sostiene il supporto meccanico per la cartilagine articolare.

Il tessuto osseo stesso può essere considerato un sistema funzionalmente graduato (Haïat et al., 2009). Infatti, nel corpo umano, le ossa possono essere classificate in vari gruppi, ma tutte presentano complicate strutture graduate. Un tipico esempio è dato dalle ossa lunghe, che sono caratterizzate da un lungo fusto con uno strato osseo corticale compatto che racchiude una cavità centrale contenente osso trabecolare, midollo e grasso. Se si considera la sezione trasversale di un osso lungo, non vi è alcuna interruzione evidente o brusca interfaccia tra l'osso corticale esterno e l'osso trabecolare interno, ma è possibile osservare una variazione della dimensione dei pori e della distribuzione della porosità (Figura 1.2).

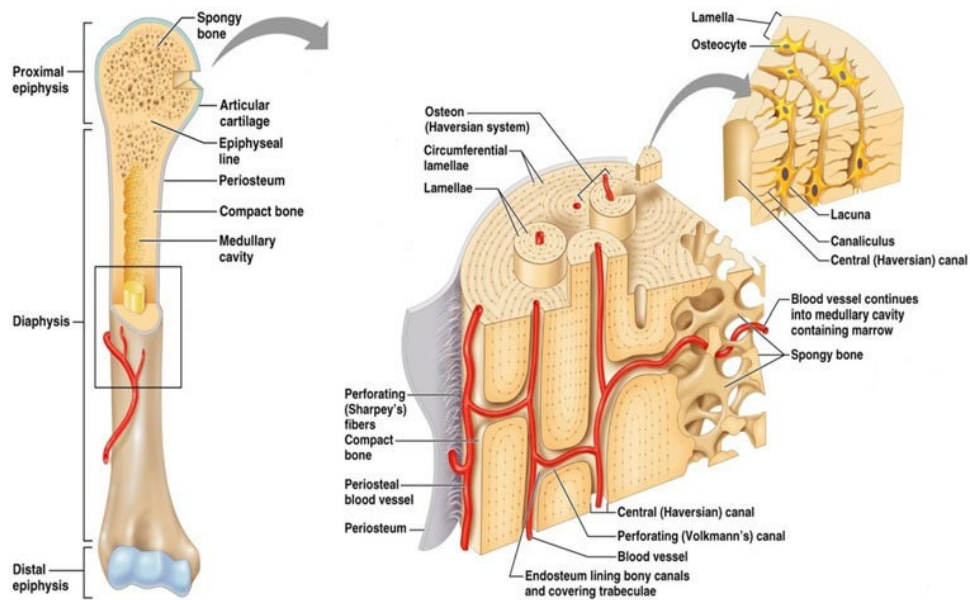


Figura 1.2. Sezione trasversale dell'osso, che mostra la sua struttura gerarchica e multistrato, con differenti livelli di porosità (Chahal et al., 2019)

Dal punto di vista compositivo, il tessuto osseo è costituito principalmente da matrice extracellulare mineralizzata (ECM), contenente fasi sia organiche, sia inorganiche: collagene di tipo I e idrossiapatite, $\text{Ca}_{10}(\text{PO}_4)_6(\text{OH})_2$. Lo strato esterno, ovvero l'osso corticale, è liscio, continuo e denso (circa $1,85 \text{ g/cm}^3$). L'interno, ovvero l'osso trabecolare, attraverso cui scorrono vasi sanguigni e midollo osseo, ha una porosità media del 75-95% e una densità media di 0.3 g/cm^3 (Karageorgiou & Kaplan, 2005).

1.3 Tipi di FGMs

Il potenziale di questi materiali viene oggi sfruttato in molti settori, anche diversi da quello iniziale per il quale erano stati inventati. Il tipo di applicazione prevista generalmente determina il tipo di FGM da utilizzare. Nell'applicazione biomedica, ad esempio, molti impianti devono mimare l'organo umano che intendono sostituire o riparare, affinché possano funzionare correttamente senza distruggere i tessuti circostanti e rimanere in servizio il più a lungo possibile. Questo è il motivo per cui c'è un così alto interesse di ricerca nella realizzazione di impianti biomedicali con materiali a variazione graduale di proprietà: gli innesti migliori sono quelli che si avvicinano di più al tessuto nativo.

I FGMs che vengono ora progettati includono FGMs a gradiente di composizione chimica, a gradiente microstrutturale e a gradiente di porosità (Figura 1.3). Ciascuno di questi verrà qui di seguito illustrato.

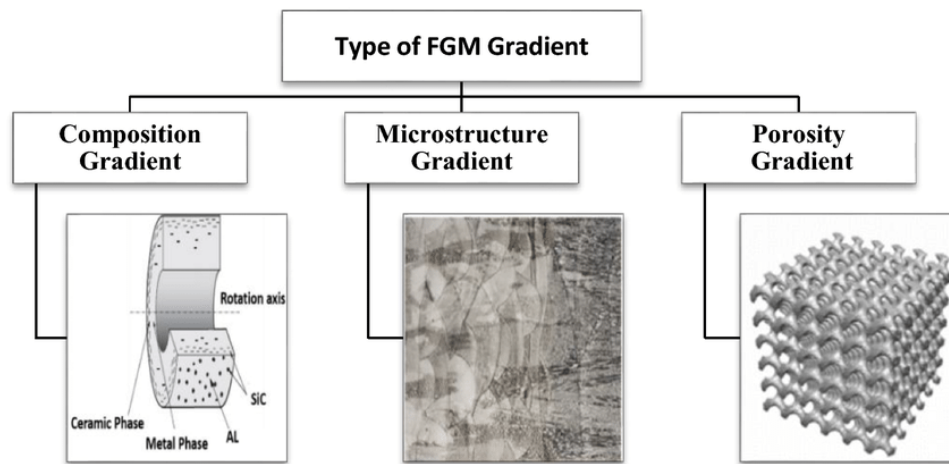


Figura 1.3. *Tipi di Functionally Graded Materials* (El-Galy et al., 2019)

- FGMs a gradiente di composizione chimica

Questo è il tipo di FGM in cui la composizione chimica viene gradualmente variata in base alla posizione spaziale del materiale. Ciò può avvenire sotto forma di una singola fase o in più fasi. Nel primo caso il materiale viene prodotto come risultato della solubilità degli elementi chimici di una fase nell'altra. Questo si verifica di solito durante il processo di sinterizzazione (Bharti et al., 2013). Il graduale cambiamento nella distribuzione degli elementi chimici nella singola fase porta alla formazione del materiale desiderato.

I FGMs più comunemente progettati e utilizzati sono però quelli multifase (R. M. Mahamood & Akinlabi, 2015). Le fasi e la composizione chimica vengono fatte variare attraverso tutto il volume del materiale. Poiché la composizione del materiale varia da un materiale all'altro, si creano fasi diverse con diverse composizioni chimiche mirate a raggiungere l'applicazione prevista. Le diverse fasi che vengono prodotte dipendono dalla quantità compositiva del materiale di rinforzo e dalle condizioni di fabbricazione, come la velocità di raffreddamento e il trattamento termico effettuato sul materiale.

Nella metallurgia delle polveri, il metodo di produzione consiste nell'inserire la composizione di polvere richiesta strato per strato, a cui segue la compattazione della polvere e quindi la sinterizzazione. Durante il processo di sinterizzazione, alcune polveri metalliche reagiranno per formare diversi composti chimici e fasi, che variano a seconda della composizione spaziale.

- FGMs a gradiente di microstruttura

I FGMs a gradiente di microstruttura costituiscono un altro tipo di materiale in cui la microstruttura è adattata in modo da produrre microstrutture diverse nel materiale, che viene fatto variare gradualmente. La gradazione microstrutturale può essere ottenuta durante il processo di solidificazione, in modo tale che la superficie del materiale venga temprata, ad

esempio, quando si vuole ottenere una proprietà superficiale molto dura. Il nucleo dello stesso materiale viene lasciato raffreddare lentamente, il che aiuterebbe a produrre microstrutture diverse da quelle sulla superficie fino alla parte più interna. Inoltre, la gradazione microstrutturale può essere ottenuta attraverso un processo di trattamento termico controllato.

- FGMs a gradiente di porosità

I FGMs a gradiente di porosità sono materiali la cui porosità viene fatta variare con il cambiamento della posizione spaziale. La forma e la dimensione del poro vengono progettate e variate a seconda delle proprietà che si vogliono conferire. Il gradiente di porosità si può ottenere per variazione della densità o della dimensione dei pori.

Questo tipo di materiale è importante nel settore biomedico, in quanto il materiale naturale che si intende sostituire è caratterizzato da un gradiente di porosità. La presenza di porosità è fondamentale per una corretta integrazione coi tessuti circostanti: promuove la guarigione della ferita e l'irrorazione dei tessuti. Aiuta anche a ridurre il peso complessivo dell'impianto e a regolare il modulo elastico in modo che sia il più paragonabile possibile a quello del tessuto naturale (Miao & Sun, 2009). La quantità e la distribuzione dimensionale della porosità, sia a livello macroscopico, sia microscopico, sono proprietà morfologiche importanti: sebbene i meccanismi di crescita ossea dipendano dal biomateriale stesso, è comunemente riconosciuto che i pori relativamente più grandi favoriscono l'osteogenesi diretta (attraverso la vascolarizzazione e l'elevata ossigenazione) e l'osteointegrazione, mentre i pori più piccoli determinano l'ossificazione osteocondrale (Karageorgiou & Kaplan, 2005).

I materiali a gradiente di porosità possono essere prodotti dalla deposizione di polvere con una miscela variabile di particelle di diverse forme e dimensioni, ma con scarso controllo sui parametri della struttura. È anche per questo che oggi si predilige la loro costruzione tramite la stampa 3D. L'utilizzo della manifattura additiva consente infatti di sfruttare a pieno i solidi cellulari nella progettazione di strutture reticolari per l'ingegneria tissutale, dove l'architettura della struttura FGM risulta fondamentale per ottenere una combinazione favorevole di proprietà meccaniche e biologiche per l'osteointegrazione.

I progressi dei metodi di fabbricazione digitale, in particolare la manifattura additiva (AM), hanno aperto una gamma completamente nuova di possibilità. Esiste una pleora di diversi processi, configurazioni meccaniche e materiali e loro combinazioni. Ciò che hanno in comune è l'aumento dei gradi di libertà geometrica che offrono. Non è più necessario che le parti siano costituite da un materiale solido, ma possono essere dettagliate in tutto il volume e rese selettivamente porose, per perseguire scopi diversi.

1.4 Applicazioni biomediche

Negli Stati Uniti, ogni anno vengono eseguiti quasi due milioni di interventi chirurgici per riparare ossa fratturate o danneggiate. La capacità dell'osso di autoripararsi nel caso di grandi difetti può essere limitata a causa di carenze di afflusso di sangue o in presenza di malattie sistemiche: i grandi difetti ossei, che sono comunemente causati da traumi, artrosi o tumori, sono sempre stati un problema impegnativo e costoso nella medicina moderna. Studi recenti hanno dimostrato che la domanda globale di sostituti ossei e innesti ha raggiunto i 3,4 miliardi di dollari USA nel 2022 con un tasso di crescita del 4,8%, rendendolo un problema molto costoso per molte nazioni e territori in tutto il mondo. Il processo di rigenerazione ossea prevede quattro diverse fasi: formazione dell'ematoma, formazione della fibrocartilagine, formazione del callo osseo e rimodellamento osseo. Durante questo processo complesso e lungo che può richiedere fino a mesi o addirittura anni, una varietà di citochine e fattori di crescita sono coinvolte per guidare molti tipi di cellule a riparare adeguatamente il difetto (Figura 1.4).



Figura 1.4. Esempio di difetto osseo che rende inadeguato il posizionamento di un impianto (R. Zhao et al., 2021)

Di conseguenza, per migliorare il processo e ottenere un migliore risultato di guarigione, dovrebbe essere fornito un ambiente meccanicamente stabile per ospitare tutte le molecole di segnalazione e consentire a quelle cellule essenziali di proliferare. Le capacità di penetrazione, proliferazione, differenziazione e migrazione delle cellule del rivestimento osseo sono infatti influenzate dalle dimensioni e dalla geometria dei pori della struttura.

In base al tempo necessario per la guarigione ossea, gli impianti ortopedici possono essere classificati in due diversi gruppi:

- protesi articolari permanenti (ad es. caviglia, anca, ginocchio e spalla artificiali, che dovrebbero funzionare per tutta la durata della vita);

- dispositivi temporanei per il fissaggio di fratture (ad es. perni intramidollari, placche, fili e viti, necessari per riparare l'osso fratturato e che dovrebbero durare per una durata relativamente breve).

Si rende pertanto necessario uno sforzo maggiore nello sviluppo di impianti ortopedici per riparare efficacemente le ossa danneggiate.

I requisiti delle proprietà meccaniche del gradiente e della biocompatibilità hanno portato alla concettualizzazione dei FGMs nell'ingegneria ortopedica per superare le principali limitazioni associate ai materiali monolitici. Per migliorare la biocompatibilità e le proprietà meccaniche dei materiali monolitici, sono state studiate diverse altre tecniche, tra cui il rivestimento superficiale. Tuttavia, la scarsa sopravvivenza e il distacco degli strati dalla superficie dell'impianto sono stati i limiti significativi della tecnologia di rivestimento.

La strategia di variazione graduata della porosità consente di realizzare potenziali materiali ad alte prestazioni, perché le proprietà che li caratterizzano sono molto simili a quelle del tessuto osseo nativo (Xiong et al., 2020). Sfortunatamente, un biomateriale altamente poroso si traduce in scarse proprietà meccaniche. Per ottimizzare le proprietà meccaniche e biologiche, sono stati sviluppati biomateriali a gradiente di porosità, che presentano molti vantaggi rispetto ai biomateriali densi e ai biomateriali porosi uniformi o omogenei. Le superfici dei pori interni dei biomateriali porosi graduati possono anche essere modificate con rivestimenti organici, inorganici o biologici e gli stessi pori interni possono anche essere riempiti con materiali biocompatibili e biodegradabili o cellule viventi.

Negli impianti di sostituzione scheletrica permanente, per esempio, il gradiente di porosità aiuta a minimizzare il fenomeno dello *stress shielding*, che si verifica quando gli impianti metallici, come placche ossee e viti, vengono utilizzati per riparare le fratture o nella chirurgia sostitutiva dell'articolazione (Thieme et al., 2001) (Figura 1.5). Sebbene le placche metalliche rigide stabilizzino il sito della frattura, aiutino a mantenere il contatto tra i frammenti ossei e consentano il carico precoce e la mobilità del paziente, la maggiore rigidità dell'impianto, generalmente realizzato in leghe di titanio e cromo o acciaio inossidabile, provoca la perdita ossea come risultato del ridotto carico fisiologico.

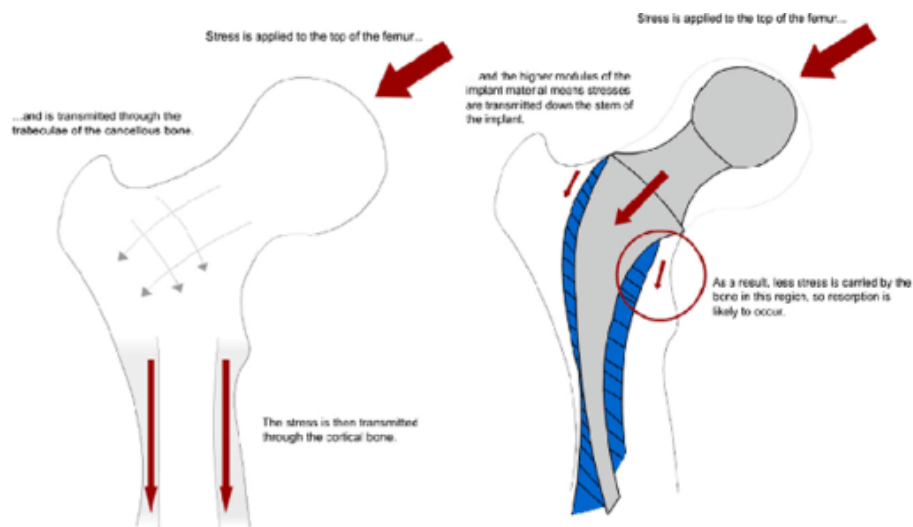


Figura 1.5. Fenomeno dello stress shielding applicato al caso di una protesi d'anca: nell'osso sano la sollecitazione è applicata alla testa del femore e trasmessa attraverso le trabecole dell'osso trabecolare all'osso corticale, mentre in presenza di impianto, l'alto modulo di elasticità del materiale porta a sollecitazioni che vengono trasmesse fino allo stelo, con risultato di una minore sollecitazione sull'osso, e conseguente riassorbimento osseo (Arifin et al., 2014)

Riassumendo, in campo biomedico si sfruttano i vantaggi legati porosità della struttura reticolare, e dalla gradazione di questa proprietà.

Dal fatto che si crea una struttura reticolare si ottiene:

- Alleggerimento, ovvero riduzione del peso delle parti, pur mantenendo l'integrità strutturale. Il reticolo consente infatti di ridurre la massa solida senza compromettere le prestazioni. Meno materiale riduce anche i costi di produzione, rendendo economicamente sostenibile la produzione con la manifattura additiva. Un altro vantaggio delle strutture reticolari nelle applicazioni di alleggerimento è che possono essere altamente resistenti ai danni. Questa proprietà le rende adatte anche per componenti critici dove la parte deve sopportare carichi significativi, come nel caso dell'esempio sopra riportato;
- Ammortizzazione, nel caso di strutture reticolari polimeriche che possono deformarsi e tornare alla loro forma originale, comportandosi come schiume. Le schiume stampate in 3D sono superiori a quelle tradizionali perché la loro forma e rigidità possono essere facilmente personalizzate o variate in tutta la parte. È possibile utilizzare queste schiume per stampare in 3D parti interne di dispositivi protesici;
- Assorbimento di energia, in quanto i reticoli possono migliorare le proprietà di assorbimento degli urti.

Il gradiente permette:

- Miglioramento della biocompatibilità degli impianti;
- Riduzione dell'effetto di *stress shielding*;
- Prevenzione del guasto meccanico dei *coatings* sugli impianti metallici;
- Miglioramento del requisito di stabilità biomeccanica.

1.5 Strutture reticolari

Secondo la definizione proposta da Gibson and Ashby, un solido a struttura cellulare è un solido costituito dall'assemblaggio nello spazio di singole unità dette celle; dove per cella si intende un elemento spaziale cavo delimitato da spigoli e facce.

Le caratteristiche geometriche e i parametri del reticolo sono gli elementi costitutivi di base da cui è possibile progettare reticoli a gradiente funzionale. Selezionando i parametri più adeguati e manipolandoli localmente per raggiungere gli obiettivi prefissati, un reticolo omogeneo può essere convertito in una struttura multifunzionale, altamente personalizzabile e modificabile.

In letteratura si possono trovare numerose tassonomie delle strutture cellulari, che vanno dalla morfologia, ai metodi di produzione, alla funzionalità. Solitamente vengono presentate sotto forma di albero decisionale.

Bhate et al. hanno proposto uno schema di classificazione del materiale cellulare basato su quattro decisioni prese durante la fase di progettazione (Bhate et al., 2019) (Figura 1.6):

- la selezione della dimensione della cella unitaria,
- la variazione spaziale della cella unitaria,
- i parametri della cella,
- l'integrazione nel dominio di progettazione.

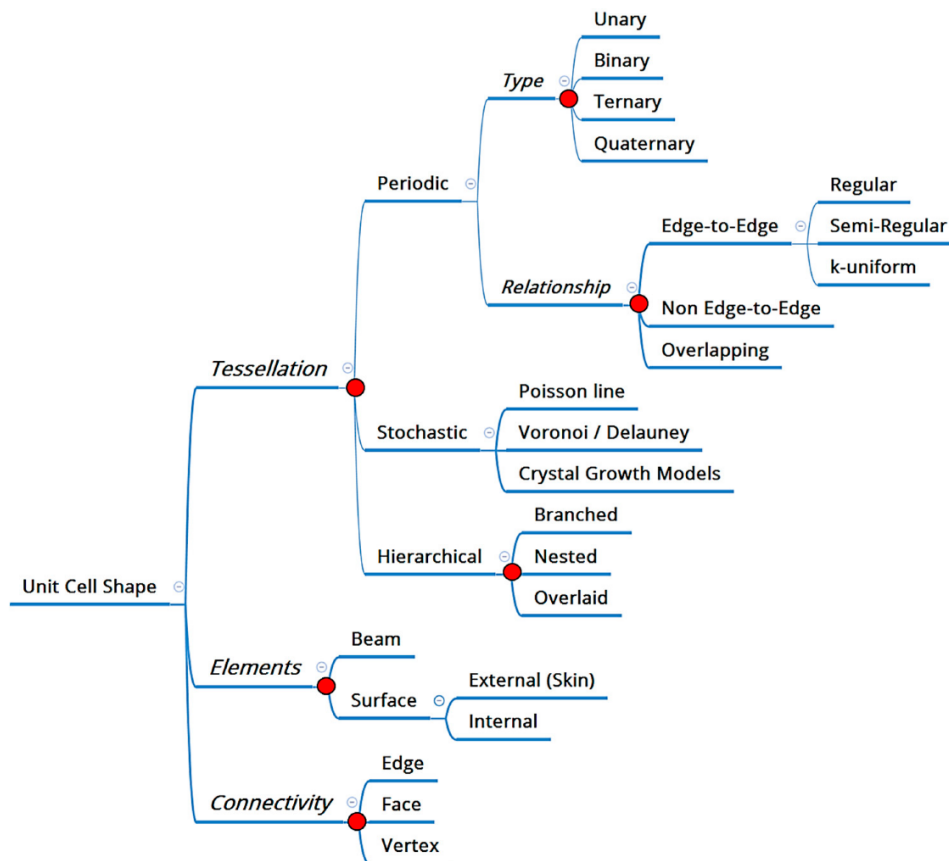


Figura 1.6. *Classificazione della cella unitaria dei materiali cellulari basata su un processo decisionale a tre livelli (Bhate et al., 2019)*

Tamburrino et al. hanno diviso le strutture reticolari in due categorie principali, sulla base del modo in cui la cella unitaria è disposta nello spazio (Tamburrino et al., 2018):

- stocastiche, come le schiume aperte e chiuse;
- non stocastiche, caratterizzate da un'organizzazione controllata delle loro celle unitarie.

La disposizione delle celle unitarie può essere sia bidimensionale, sia tridimensionale (Figura 1.7).

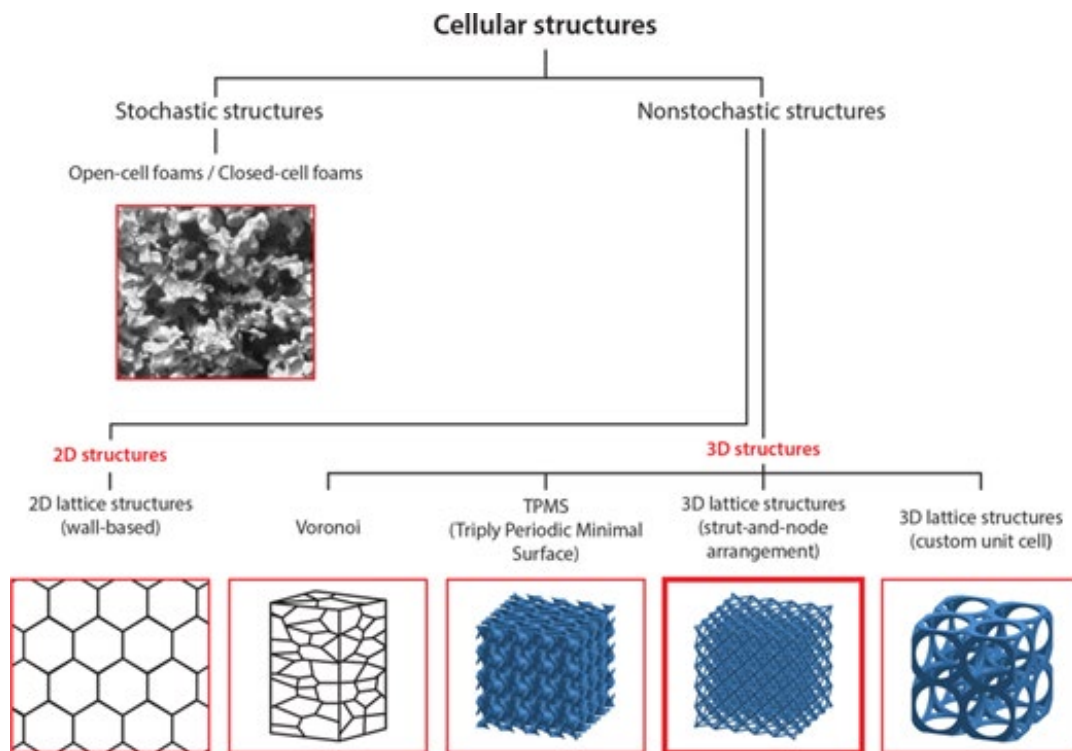


Figura 1.7. *Classificazione della cella unitaria dei materiali cellulari in base alle strutture stocastiche e non stocastiche (Tamburrino et al., 2018)*

Savio et al. hanno proposto uno schema di classificazione basato sulla geometria dei materiali cellulari secondo quattro criteri principali: le dimensioni dell'elemento cella, la geometria della cella, la topologia della cella e la forma della struttura (Savio et al., 2018) (Figura 1.8).

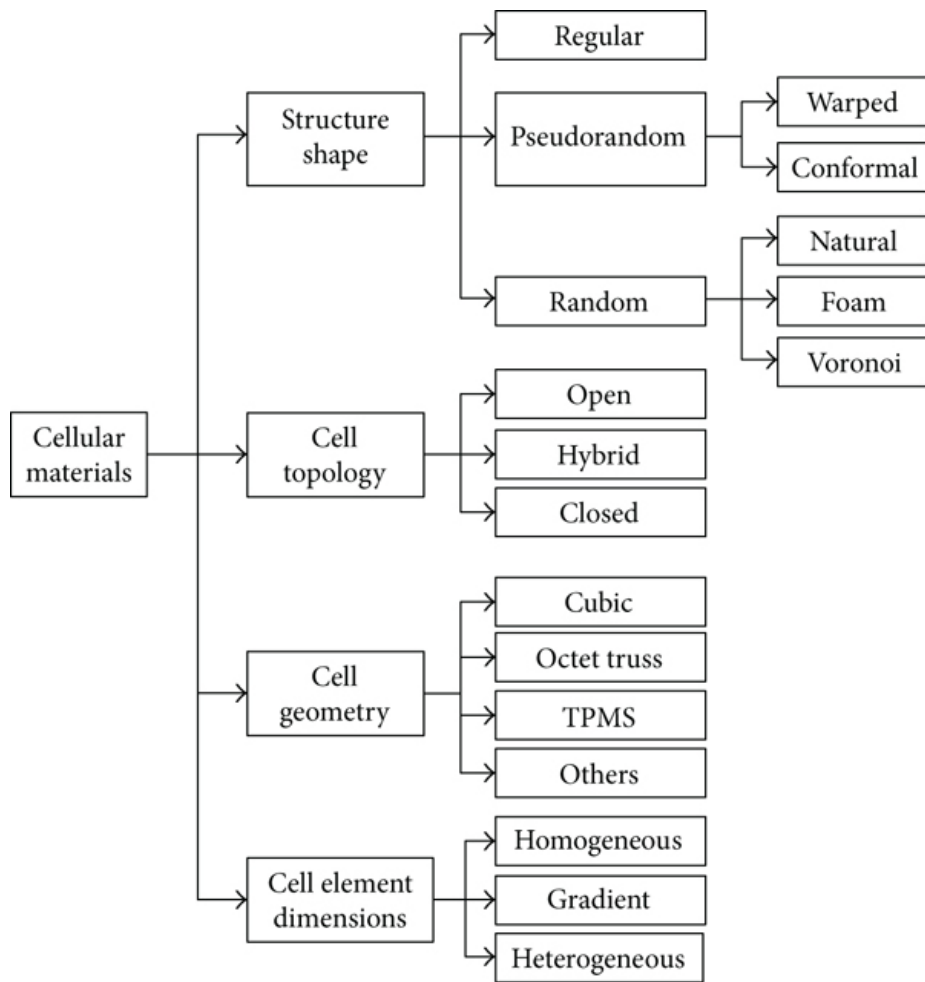


Figura 1.8. *Classificazione della cella unitaria dei materiali cellulari in base alla geometria* (Savio et al., 2018)

Secondo questa suddivisione, che verrà scelta come riferimento, a livello di forma della struttura, i materiali cellulari possono essere suddivisi in:

- Strutture regolari

Consistono in una semplice ripetizione della cella unitaria nell'intero volume di progetto;

- Strutture pseudocasuali

Si ottengono mantenendo la topologia e variando sia le dimensioni, sia la geometria. Questi materiali cellulari possono essere ulteriormente suddivisi in strutture deformate e conformi. Le strutture deformate sono realizzate deformando la cella unitaria, mantenendo la topologia originale. Nelle strutture conformi, la geometria e le dimensioni di ciascuna cella sono diverse per adattarsi (cioè conformarsi) alla forma esterna del modello. Rispetto ai normali materiali cellulari, quelli conformi non presentano mai cellule interrotte o incomplete; questa caratteristica elimina la debolezza ai confini e fornisce rigidità e resistenza all'intero modello (Figura 1.9).

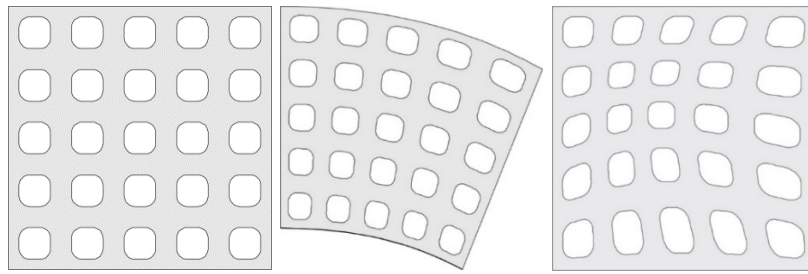


Figura 1.9. *Esemplificazione di un materiale cellulare regolare, conforme (la struttura è piegata sulla curva spessa) e deformato*

- Strutture casuali

Presentano un'organizzazione casuale delle celle, e sia la geometria, sia la dimensione delle celle variano in modo casuale.

In natura si possono trovare diversi tipi di strutture casuali, come l'osso corticale e trabecolare. Le strutture in schiuma si ottengono utilizzando metodi di fabbricazione tradizionali come l'iniezione di gas in metallo fuso, la deposizione di vapore o schiumatura spray; la forma e la dimensione dei pori possono essere parzialmente controllate modificando i parametri di processo. Strutture in schiuma possono essere ottenute anche con tecnologie di produzione additiva utilizzando funzioni matematiche o operazioni booleane che consentono un pieno controllo dei pori.

Inoltre, materiali cellulari casuali possono essere modellati adottando un diagramma Voronoi, posizionando casualmente un insieme di punti all'interno del volume di progetto, suddividendo lo spazio all'interno regioni in base alla distanza tra i punti e poi assegnando uno spessore ai bordi delle regioni. Un esempio di diagramma di Voronoi 2D è mostrato nella Figura 1.10, dove ogni cella è il sottoinsieme nel piano contenente i punti che sono più vicini a un seme specifico che a qualsiasi altro.

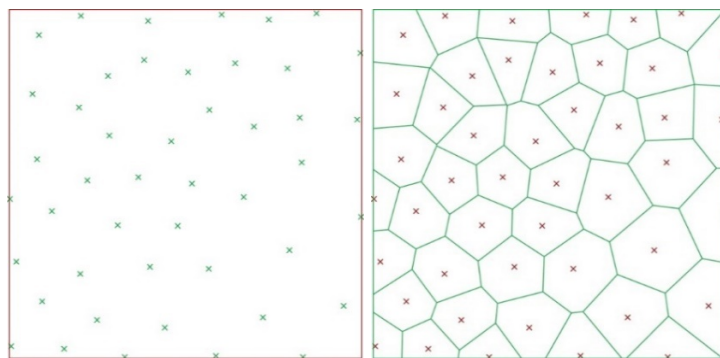


Figura 1.10. *Materiale cellulare casuale in 2D: punti e partizionamenti della regione*

Indipendentemente dalla forma della struttura, i materiali cellulari possono essere classificati in base alla topologia cellulare:

- A cella aperta

Presentano solo celle a struttura porosa aperta, il che significa che i pori sono accessibili da parte di un fluido. Sono preferiti nelle applicazioni biomediche, dove le connessioni tra celle sono necessarie per consentire lo scambio di liquidi e la ricrescita dei tessuti;

- A cella chiusa

I pori sono inaccessibili. Rispetto a quelle a cella aperta, questa struttura offre maggiore rigidità ma, allo stesso tempo, ostacola lo scambio di fluidi e impedisce lo svuotamento del materiale nel processo di produzione additiva; quindi, sono adatte a scopi strutturali;

- Ibrido

Il modello presenta parti a celle aperte e parti a celle chiuse.

I materiali cellulari possono differire nella geometria della cella unitaria:

- cubico semplice (SC), cubico a corpo centrato (BCC) e cubico a corpo centrato rinforzato (RBCC) provengono tutti dalla stessa cella cubica, con un numero crescente di raggi. La cella *octet-truss* (OT) proviene da una cella cubica a faccia centrata. Altre celle usate di frequente sono le celle modificate Gibson-Ashby (GA) e le celle modificate di Wallach-Gibson (WG) (Figura 1.11).

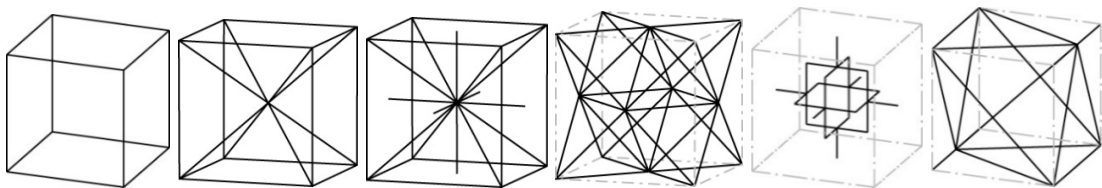


Figura 1.11. *Tipi di cella: SC, BCC, RBCC, OT, GA, WG*

- *Triply Periodic Minimal Surfaces* (TPMS), ovvero superfici minime periodiche triple, che sono superfici 3D caratterizzate da curvatura media nulla in ogni punto, che riducono al minimo l'area superficiale per determinate condizioni al contorno. Possono essere definite da una semplice funzione matematica. Tra le più studiate vi sono:

1. *Schwarz P*

La struttura reticolare *Schwarz primitive* viene generata come isosuperficie della funzione indicata nella seguente equazione:

$$F(x, y, z) = \cos\left(\frac{2\pi x}{L}\right) + \cos\left(\frac{2\pi y}{L}\right) + \cos\left(\frac{2\pi z}{L}\right) = k$$

dove F è una superficie calcolata all'isovalore k , ovvero il valore di *threshold* che si intende imporre, e L è la lunghezza della cella (Figura 1.12).

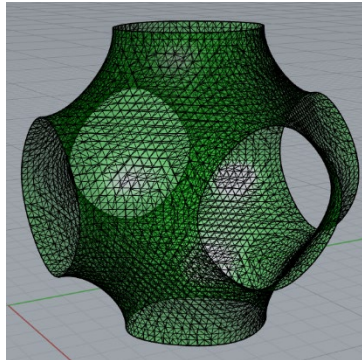


Figura 1.12. *Primitiva di Schwarz*

2. *Giroide di Schoen*

Il giroide è una TPMS infinitamente connessa scoperta da Alan Schoen nel 1970 (Figura 1.13). Può essere espressa dalla seguente equazione:

$$F(x, y, z) = \cos\left(\frac{2\pi x}{L}\right) \sin\left(\frac{2\pi y}{L}\right) + \cos\left(\frac{2\pi y}{L}\right) \sin\left(\frac{2\pi z}{L}\right) + \cos\left(\frac{2\pi z}{L}\right) \sin\left(\frac{2\pi x}{L}\right) = k$$

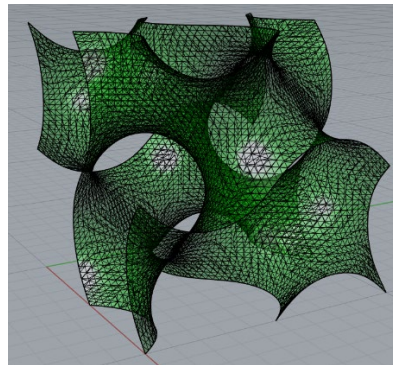


Figura 1.13. *Giroide di Schoen*

3. *Schwarz Diamond*

La superficie *Schwarz D* è un semplice esempio di TPMS con simmetria diedrica locale (Figura 1.14). È rappresentata dall'equazione:

$$F(x, y, z) = \sin(2\pi x/L) \sin(2\pi y/L) \sin(2\pi z/L) + \sin(2\pi x/L) \cos(2\pi y/L) \cos(2\pi z/L) +$$

$$\cos(2\pi x/L) \sin(2\pi y/L) \cos(2\pi z/L) + \cos(2\pi x/L) \cos(2\pi y/L) \sin(2\pi z/L) = k$$

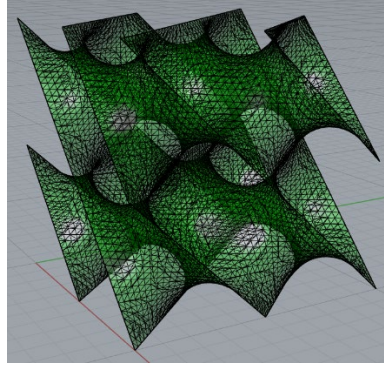


Figura 1.14. *Schwarz Diamond*

In letteratura sono stati riportati diversi metodi per convertire le funzioni TPMS in modelli fisici. Un primo metodo consiste nell'ispessimento della *shell*, mentre un secondo metodo crea una TPMS di tipo *network*, separando la cella unitaria in una fase solida e una fase vuota (Figura 1.15). La regione solida risultante sarà il volume con un valore inferiore all'isovalore (Jones et al., 2022):

$$F(x, y, z) \leq k$$

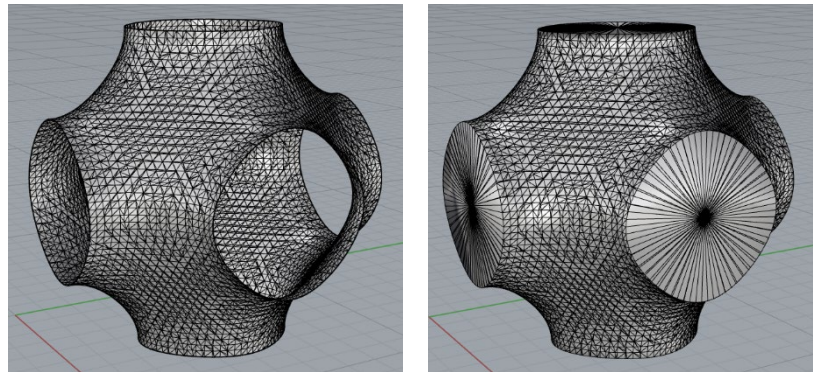


Figura 1.15. *Esempio di struttura Schwarz P ottenuta da a) ispessimento della shell; b) utilizzando la superficie come confine tra fase solida e fase vuota*

Le strutture che derivano dal primo tipo di cella vengono chiamate *truss-based*, in quanto formate da aste e nodi, mentre le seconde sono *shell-based*, ovvero costituite da superfici.

Infine, concentrandosi sulla dimensione (cioè lo spessore di una TPMS o il diametro del raggio di una struttura *truss-based*) degli elementi cellulari, viene presentato un altro ordinamento:

- Cella omogenea, se tutti gli elementi della cella hanno lo stesso spessore;
- Cella eterogenea, se lo spessore delle aste è diverso.

- Cella a gradiente, se lo spessore degli elementi cellulari varia gradualmente secondo un modello.

1.6 Approcci di modellazione

La modellazione geometrica utilizzata dai sistemi di Computer Aided Design (CAD), spesso indicata anche come *Computational geometry*, è la rappresentazione matematica della geometria di un oggetto. In letteratura sussistono tre classi principali in cui si possono classificare gli approcci di modellazione geometrica solida: Boundary Representation (B-Rep), Constructive Solid Geometry (CSG), Volumetric Representation (V-Rep) (Figura 1.16).

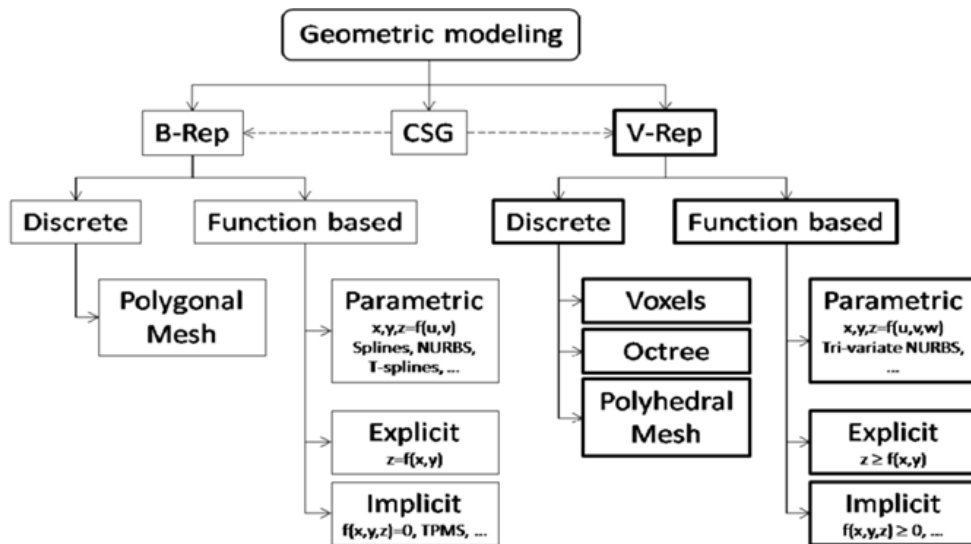


Figura 1.16. Schematizzazione degli approcci di modellazione geometrica (Grigolato et al., 2019)

1.6.1 Boundary-Representation (B-Rep)

Storicamente, la *Boundary Representation*, ovvero la rappresentazione dei contorni, è stata una delle prime rappresentazioni computerizzate ad essere utilizzate nella descrizione di oggetti poliedrici tridimensionali (Shapiro, 2002). Costituisce un approccio consolidato, in quanto la maggior parte dei software CAD presenti in commercio si basano su questo e sull'assunzione che il materiale costituente l'oggetto, quindi presente all'interno dei contorni, sia omogeneo.

Descrive la forma come un insieme di superfici che separano l'ambiente interno da quello esterno. I principali elementi topologici che consentono di rappresentare un oggetto sono facce, spigoli e vertici (Figura 1.17).

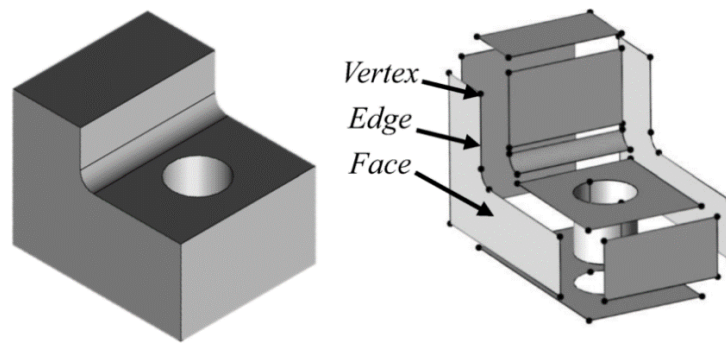


Figura 1.18. *Composizione di un modello B-Rep*

Per rappresentare curve e superfici si utilizzano tre tipi di funzioni: esplicita, implicita e parametrica. Le espressioni della forma $y = f(x)$ o $z = f(x, y)$ sono chiamate rappresentazioni esplicite perché esprimono una variabile in modo esplicito in termini di altre variabili. Non tutte le curve e le superfici possono essere catturate prontamente da una singola espressione di questo tipo. Le rappresentazioni implicite sono più generali. La curva esplicita $y = f(x)$ è la stessa della curva implicita $y - f(x) = 0$, ma non è sempre semplice convertire una curva implicita in un'unica formula esplicita. Inoltre, le equazioni implicite possono essere utilizzate per definire curve e superfici chiuse o curve e superfici che si autointersecano, forme impossibili da rappresentare con funzioni esplicite.

Tuttavia, le rappresentazioni implicite hanno anche i loro svantaggi. Data una rappresentazione esplicita $y = f(x)$, possiamo facilmente trovare molti punti sulla curva $(x, f(x))$ selezionando valori per x e calcolando $f(x)$. Se le nostre funzioni $f(x)$ sono limitate a funzioni elementari come i polinomi, allora per ogni x esiste una y unica e facilmente calcolabile. Quindi è semplice rappresentare graficamente la curva $y = f(x)$. D'altra parte, potrebbe non essere così facile trovare punti sulla curva $f(x, y) = 0$. Per molti valori di x potrebbe non esserci affatto y , oppure potrebbero esserci diversi valori di y . Trovare punti su superfici implicite $f(x, y, z) = 0$ può essere ancora più complicato. Pertanto, può essere difficile eseguire il rendering di curve e superfici definite in modo implicito.

Esiste un altro modo standard per rappresentare curve e superfici che è più generale della forma esplicita ed è ancora facile da renderizzare. Possiamo esprimere curve e superfici in forma parametrica, rappresentando ciascuna coordinata tramite un'equazione esplicita in un nuovo insieme di parametri. Per le curve planari poniamo $x = x(t)$ e $y = y(t)$; per superfici in spazi 3D poniamo $x = x(s, t)$, $y = y(s, t)$ e $z = z(s, t)$. Le funzioni parametriche, come le *Spline* e le *NURBS* (non-uniform rational basis-splines), sono supportate da numerosi algoritmi.

La rappresentazione parametrica presenta diversi vantaggi. Come per la rappresentazione esplicita, per la rappresentazione parametrica basta valutare le funzioni di coordinate a vari valori dei parametri. Così come le equazioni implicite, anche le equazioni parametriche possono essere utilizzate per rappresentare curve e superfici chiuse, nonché curve e superfici che si autointersecano. Inoltre, la rappresentazione parametrica ha un altro vantaggio: è facilmente estendibile a dimensioni superiori. Per esempio, se vogliamo rappresentare una curva in 3D, tutto ciò che dobbiamo fare è introdurre un'equazione aggiuntiva $z = z(t)$.

La rappresentazione parametrica ha le sue idiosincrasie. La rappresentazione esplicita di una curva è unica: il grafico di $y = g(x)$ è lo stesso del grafico di $y = f(x)$ se e solo se $g(x) = f(x)$. Allo stesso modo, se ci limitiamo alle funzioni polinomiali, la rappresentazione implicita $f(x, y) = 0$ è essenzialmente unica. Infatti, se $f(x, y)$ e $g(x, y)$ sono polinomi, allora $g(x, y) = 0$ rappresenta la stessa curva di $f(x, y) = 0$ sui numeri complessi se e solo se $g(x, y)$ è una costante moltiplicata per una potenza di $f(x, y)$. Tuttavia, la rappresentazione parametrica di una curva non è univoca. Inoltre, se restringiamo la nostra attenzione alle parametrizzazioni polinomiali o razionali, allora è noto che ogni curva o superficie parametrica si trova su una curva o superficie polinomiale implicita. Il contrario, però, non è vero. Esistono curve polinomiali implicite e superfici che non hanno parametrizzazione polinomiale o razionale. Pertanto, la forma polinomiale implicita è più generale della forma parametrica razionale. Le rappresentazioni in forma parametrica non sono adatte alla rappresentazione di forme estremamente complesse, per le quali è conveniente adoperare modelli basati sulle *mesh*, ossia una discretizzazione della geometria in un insieme di poligoni chiusi adiacenti.

1.6.2 Constructive Solid Geometry (CSG)

L'approccio di Geometria Solida Costruttiva combina il volume occupato dalla sovrapposizione di oggetti 3D utilizzando un set di operazioni booleane. Ogni primitiva è definita come una combinazione di semispazi. Tra le primitive standard tipiche vi sono cono, cilindro, sfera, toro, blocco, superficie, spline chiusa. Queste primitive possono essere omogenee o eterogenee. Nel secondo caso, le operazioni booleane dovrebbero essere ridefinite per gestire primitive eterogenee (Savio et al., 2019; Shin & Dutta, 2001). Tramite le operazioni di unione, intersezione e differenza delle primitive è possibile ottenere la geometria voluta. Spesso il procedimento per ottenere l'oggetto finale viene rappresentato da un albero, come mostrato nella Figura 1.19.

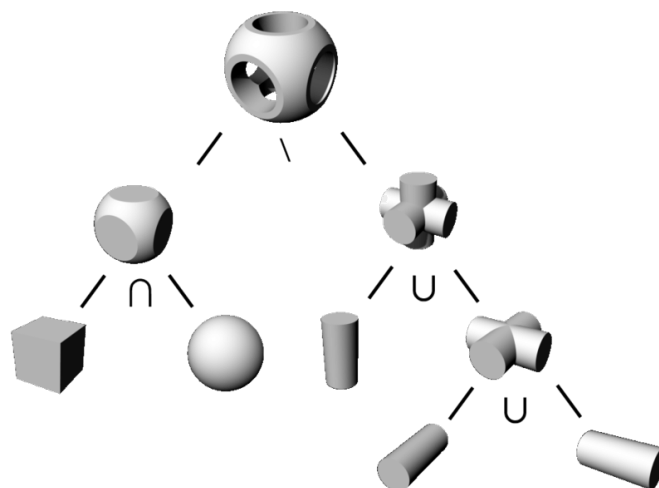


Figura 1.19. *Albero CSG con le tre operazioni booleane*

La geometria solida costruttiva ha una serie di usi pratici. Viene utilizzata nei casi in cui si desiderano oggetti geometrici semplici in cui l'accuratezza matematica è importante (Hughes et al., 1996).

Uno dei vantaggi della CSG è che può facilmente assicurare che gli oggetti sono "solidi" o a tenuta stagna se tutte le forme primitive sono a tenuta stagna (Rossen & Baranowski, 2011). Questo può essere importante per alcune applicazioni di calcolo di produzione. In confronto, quando si crea una geometria basata su B-Rep, sono necessari dati topologici aggiuntivi o devono essere eseguiti controlli di coerenza per garantire che la descrizione del contorno fornisca un oggetto solido valido. Una comoda proprietà delle forme CSG è che è facile classificare punti arbitrari come interni o esterni alla forma creata. Il punto viene semplicemente classificato rispetto a tutte le primitive sottostanti e viene valutata l'espressione booleana risultante (Glassner, 1989). Questa tecnica può inoltre essere facilmente convertita in B-Rep. Per visualizzare e realizzare il modello, a seconda delle primitive, è possibile passare a una B-Rep o a una V-Rep.

Uno svantaggio è costituito dal fatto che sono consentite solo operazioni booleane nel processo di modellazione: il range di forme da modellare è limitato. Oltre a questo, richiede una gran quantità di calcolo per ottenere informazioni su bordi e facce, importanti per la visualizzazione e manipolazione del solido.

1.6.3 Volumetric Representation (V-Rep)

La modellazione volumetrica consente di fornire le informazioni necessarie a rappresentare un oggetto che occupa uno spazio tridimensionale. Consente la descrizione sia della superficie, sia

della parte interna, e di conseguenza la rappresentazione della distribuzione delle proprietà interne e non solo dei confini (Grigolato et al., 2019).

La necessità di uno stretto accoppiamento tra le fasi di progettazione e analisi è stata riconosciuta come cruciale fin da quando è stata concepita la modellazione geometrica. Per circa mezzo secolo, le superfici NURBS (Non Uniform Rational B-spline) sono state la B-Rep scelta praticamente da tutta l'industria della modellazione geometrica. Tuttavia, le esigenze delle moderne tecnologie di produzione additiva non possono più essere soddisfatte dalla B-Rep, poiché richiedono la rappresentazione e la manipolazione di campi e proprietà interni.

Come si è visto, la B-Rep può essere rappresentata da modelli discreti o da funzioni. Questo è possibile anche per la V-Rep. Nel caso di V-rep discreta, la scomposizione spaziale può essere raggiunta da voxel, *octree* e *mesh* poliedriche. La tipica rappresentazione discreta può essere trovata nella rappresentazione dell'immagine di tomografia computerizzata (CT), in forma di voxel o nel metodo degli elementi finiti (FE), come *mesh* poliedrica. Oltre ai metodi di rappresentazione discreti, nel software CAD vengono spesso utilizzate funzioni parametriche bivariate o trivariate per modificare intuitivamente la forma spostando i punti di controllo. Di solito, le funzioni parametriche bivariate vengono utilizzate per la rappresentazione della superficie, mentre le trivariate vengono implementate per il *morphing* degli oggetti. Funzioni implicite ed esplicite possono essere implementate in software CAD e possono essere utili nella rappresentazione di forme particolari come superfici minime, ma non consentono una facile modifica della forma come nel caso delle funzioni parametriche.

1.6.3.1 Rappresentazione volumetrica discreta

I modelli volumetrici discreti sono spesso classificati come modelli valutati (Kou & Tan, 2007), poiché le informazioni volumetriche memorizzate nel modello sono direttamente disponibili per ulteriori applicazioni, come l'analisi numerica e la simulazione (Sharma & Gurumoorthy, 2019).

Il modo più semplice per discretizzare un volume è suddividerlo in cubi piccoli e uguali (cioè voxel). Un voxel (*volumetric picture element*) è un'unità di misura del volume. Rappresenta la controparte tridimensionale del pixel bidimensionale (che rappresenta l'unità dell'area), e perciò il volume buffer (un ampio array 3D di voxel) dei voxel può essere considerato come la controparte tridimensionale del frame buffer bidimensionale dei pixel ("Volume Graphics," 2000) (Figura 1.20). Il valore di un voxel può rappresentare diverse proprietà. Nelle scansioni CT, i valori sono espressi in unità basate sulla scala Hounsfield, fornendo l'opacità del materiale ai raggi X (Novelline & Squire, 1997). Altre proprietà rappresentate possono essere il colore,

la densità, il materiale. Diversi tipi di valore vengono acquisiti dalla risonanza magnetica o dagli ultrasuoni. L'aggregato dei voxel, che tassellano il volume buffer, forma il dataset volumetrico, che è rappresentato come una griglia tridimensionale regolare discreta.

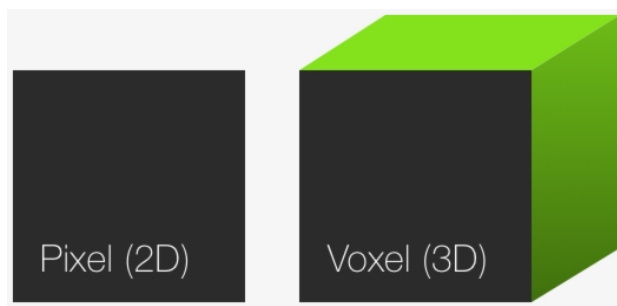


Figura 1.20. *Voxel, l'equivalente 3D del pixel*

La rappresentazione discreta può utilizzare anche un *octree* come struttura dati per rappresentare le informazioni. L'*octree* consiste nella partizione di uno spazio tridimensionale suddividendo ricorsivamente un cubo in otto cubi; ogni cubo può trovarsi all'interno, all'esterno o nel confine di un oggetto; per ogni cubo nel bordo si ripete la suddivisione fino ad ottenere la risoluzione voluta, come mostrato nella Figura 1.21. Ogni nodo interno può avere al massimo otto figli, a ciascuno dei quali viene assegnato un valore variabile a seconda che il cubo sia completamente contenuto, completamente esterno o parzialmente contenuto nel solido.

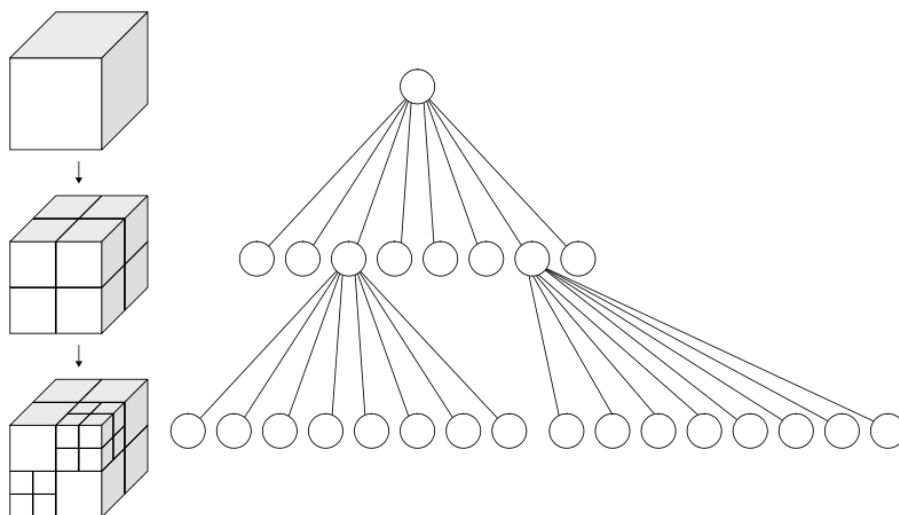


Figura 1.21. *Rappresentazione di un albero ottale (octree)*

La scomposizione dello spazio può essere raggiunta con altri approcci che possono seguire meglio la forma esterna o le caratteristiche interne, utilizzando i poliedri. In questo caso è possibile adottare elementi più grandi dove la variazione di forma o caratteristiche è bassa e viceversa.

Le superfici discrete nell'elaborazione della geometria e nella progettazione assistita da computer sono comunemente rappresentate da *mesh* triangolari. I triangoli sono popolari poiché sono gli oggetti bidimensionali lineari più semplici, banalmente planari e facilmente adatti a formulazioni agli elementi finiti. Tuttavia, le *mesh* triangolari sono svantaggiose nella pratica per diversi motivi: hanno molti spigoli, e di conseguenza elevate complessità di vertici, e generalmente non ammettono superfici sfalsate (è impossibile costruirle con qualsiasi “spessore”). Alla luce di ciò, c'è un crescente interesse per le *mesh* poligonali, dove ogni faccia è un poligono non triangolare.

Nella modellazione 3D, le *mesh* volumetriche sono il corrispondente volumetrico delle *mesh* poligonali, composte da poliedri quali esaedri o tetraedri (Figura 1.22).



Figura 2.6. Elementi delle *mesh* volumetriche: a) esaedro; b) tetraedro

A differenza delle *mesh* poligonali, che rappresentano solo la superficie come poligoni, le *mesh* volumetriche discretizzano anche la struttura interna dell'oggetto (Figura 1.23).

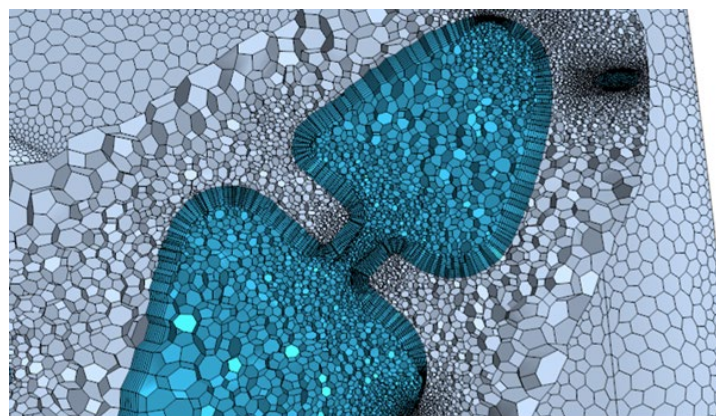


Figura 1.24. Applicazione di una *mesh* poliedrica

Un'applicazione delle *mesh* volumetriche è nell'analisi agli elementi finiti, che può utilizzare *mesh* volumetriche regolari o irregolari per calcolare le sollecitazioni e le forze interne in un oggetto per l'intero volume dell'oggetto.

1.6.3.2 Rappresentazione volumetrica basata su funzioni

Si è visto come gli approcci esistenti alla modellazione delle microstrutture includono quelli che si basano su superfici (rappresentazioni di confine o B-Rep) e voxel (rappresentazioni di volume discreti). Sebbene in una certa misura le microstrutture possano essere generate utilizzando queste rappresentazioni, molti dei problemi noti e dei limiti di entrambe le rappresentazioni sono amplificati dalla complessità geometrica delle microstrutture. Il problema si aggrava ulteriormente quando si modellano microstrutture irregolari. Si possono distinguere tra problemi quantitativi (dimensione del modello e tempo di elaborazione) e problemi qualitativi (validità del modello, precisione, parametrizzazione, operabilità e producibilità) delle rappresentazioni esistenti (Pasko et al., 2011):

- Dimensioni e tempo di lavorazione

I modelli basati sulla superficie di dimensioni moderate contenenti microstrutture reticolari di alta qualità possono includere un numero così elevato di poligoni che diventa difficile o impossibile da renderizzare per l'hardware grafico. Sebbene la dimensione dei modelli di voxel non dipenda direttamente dalla complessità dell'oggetto, il numero di voxel necessari per rappresentare superfici di alta qualità insieme a piccoli dettagli può facilmente superare le capacità disponibili della memoria del computer. Il tempo per il rendering, così come per altri tipi di elaborazione, cresce con le dimensioni del modello.

- Validità e precisione

Problemi noti derivanti da modelli e approcci CAD tradizionali basati su B-Rep (come *crack* nelle superfici, autointersezioni di poligoni, falsi poligoni aggiuntivi lasciati dalla modellazione e orientamento normale invertito), già problematici per l'industria manifatturiera, diventano sempre più difficili o impossibili da modellare e produrre su nano e micro-scala. I modelli B-Rep e voxel vengono solitamente creati all'interno di un'unità di delimitazione specificata e quindi replicati in modo iterativo nello spazio senza considerare la coerenza spaziale della struttura, che può causare ulteriori crepe nel modello ai confini di ciascuna unità. In generale, sia la B-Rep poligonale, sia la rappresentazione voxel non sono esatte e si limitano ad approssimare la geometria modellata con precisione limitata.

- Parametrizzazione e operabilità

Il supporto della generazione di modelli con parametri variabili è fondamentale per la modellazione di microstrutture, perché la loro geometria può dipendere da una serie di fattori come la distanza dalla superficie dell'oggetto, condizioni di resistenza predefinite, variazioni di

densità e altri. Le rappresentazioni esistenti hanno un supporto limitato o nullo per la parametrizzazione dei modelli di microstruttura. Quando i parametri vengono modificati, i modelli B-Rep e voxel devono essere rigenerati utilizzando una procedura o un metodo separato di alto livello. Un utente potrebbe dover applicare ulteriori operazioni specifiche su microstrutture come offset, blend e deformazioni di forma, che hanno un supporto limitato o nullo all'interno delle rappresentazioni attuali.

- Producibilità

Le microstrutture complesse controllate non sono facili da produrre utilizzando la maggior parte delle tecnologie esistenti. Attualmente la migliore soluzione può essere trovata nella prototipazione rapida o nelle tecnologie di fabbricazione digitale come la sinterizzazione laser, la stereolitografia e altri processi additivi, che producono un oggetto fisico strato per strato. Quando l'oggetto diventa grande o sono necessarie microstrutture finemente impacchettate, i modelli B-Rep aumentano notevolmente di dimensioni e diventano molto difficili o impossibili da visualizzare per gli attuali sistemi hardware e ancora più difficili da sezionare come richiesto da molti sistemi e processi di fabbricazione digitale. Sebbene gli attuali sistemi di fabbricazione digitale abbiano risoluzioni limitate, negli ultimi anni hanno raggiunto una precisione sempre maggiore e si prevede che questa tendenza continuerà. Anche così, attualmente non è raro vedere difetti o sezioni mancanti di un oggetto, in gran parte a causa della complessità implicata nella creazione di sezioni trasversali appropriate dal formato STL standard. Diversi sistemi di fabbricazione digitale accettano modelli di voxel come input; tuttavia, i voxel hanno problemi di *aliasing* noti a meno che non vengano forniti a risoluzioni molto elevate che richiedono grandi quantità di memoria. Per i sistemi che non funzionano con i voxel, è necessaria la poligonizzazione dei dati dei voxel.

Nonostante B-Rep e voxel rimangano i principali modelli utilizzati dai ricercatori per rappresentare le microstrutture, in certe situazioni emergono inconvenienti che possono essere affrontati utilizzando modelli procedurali basati su funzioni sotto forma di *Function Representation* (F-Rep).

La modellazione volumetrica, quindi, comprende anche la descrizione delle forme come risultato di una funzione che converte qualsiasi punto in valori reali, $\mathbb{R}^3 \rightarrow \mathbb{R}$. L'idea principale della modellazione basata su funzioni (o modellazione di "superficie implicita") è quella di definire insiemi di punti tramite campi scalari trivariati o funzioni reali continue di coordinate puntiformi $f(x, y, z)$.

Il dominio geometrico della rappresentazione F-Rep nello spazio 3D include solidi con modelli *non-manifold* ed entità di dimensione inferiore (superfici, curve, punti) definite dal valore zero della funzione. Una primitiva può essere definita da un'equazione o da una procedura a "scatola nera" che converte le coordinate del punto nel valore della funzione. Gli oggetti voxel possono essere usati come primitive.

Si noti che lo spazio di progettazione non è limitato dall'esistenza dello spazio euclideo E^3 , dal momento che la stampa di geometrie non euclidee è un argomento di interesse nell'AM oggi (Mensch et al., 2021). Anche se la F-Rep esisteva prima dell'ascesa dell'AM, l'uso di funzioni per rappresentare modelli solidi ha dimostrato vantaggi nel fornire maggiore libertà di progettazione rispetto, per esempio, ai metodi di modellazione booleani convenzionali.

I moderni metodi F-Rep consentono la modellazione di funzioni sia esplicite, sia implicite. È consentita anche la modellazione di superfici implicite che delimitano il corpo solido senza calcoli necessari per convertirle in esplicite. Pertanto, la capacità di utilizzare funzioni implicite come input è un vantaggio significativo della F-Rep, che si amplifica ulteriormente perché le strutture reticolari possono essere definite come un insieme di funzioni.

Tuttavia, ci sono alcuni svantaggi nell'utilizzare la modellazione F-Rep. La definizione di una struttura geometricamente complessa come una struttura reticolare eterogenea è complicata dalla necessità di definire regole in base alle quali i parametri del reticolo o la tipologia variano in tutta la struttura. La revisione della letteratura correlata ha identificato che un tale processo di progettazione può essere complicato, e che comunque finora non esiste uno strumento che lo semplificherebbe in modo sufficientemente significativo, mentre è chiaro che la produzione additiva ne trarrebbe vantaggio.

Capitolo 2

Strumenti e metodi per la progettazione di Functionally Graded Lattice Structures

Le *Functionally Graded Lattice Structures* (FGLSs), rispetto alle strutture reticolari tradizionali, hanno una caratteristica chiave: il loro design graduato aiuta a raggiungere proprietà avanzate relative alla resistenza strutturale, consentendo funzionalità come la crescita ossea, che la sola struttura reticolare non sarebbe in grado di garantire (Goel & Anand, 2019).

Con la conoscenza degli strumenti che si hanno a disposizione, in questo capitolo si propone una metodologia di progettazione di strutture *lattice* con gradiente di porosità.

Una sfida chiave che si deve affrontare come progettista di una struttura reticolare per un'applicazione specifica è quella di scegliere le variabili di progettazione del reticolo appropriate. Il tipo di cella e la dimensione dei pori svolgono ruoli cruciali nel determinare, per esempio, la rigidità e la resistenza strutturali, ma le relazioni esatte tra queste proprietà sono generalmente poco conosciute. Ciò è particolarmente problematico per qualsiasi tentativo di combinare strutture reticolari con l'ottimizzazione della topologia (TO), basti pensare a un'impalcatura porosa biomimetica come sostituto osseo, che richiede di ottimizzare al meglio la porosità, per garantire che i pori consentano la diffusione dei nutrienti per facilitare la crescita dei tessuti e assomigliare al tessuto osseo naturale e che vi sia una superficie abbastanza grande per l'attaccamento cellulare. Si mostrerà come si ricava il valore delle densità relative di diversi tipi di reticolo a cui corrispondono i parametri da assegnare alle celle unitarie. La densità relativa delle strutture cellulari ha un ruolo importante anche nell'idoneità di un reticolo per la stampa, a causa del vincolo del raggio minimo stampabile e della polvere intrappolata all'interno di un'inclusione, per esempio nel caso di stampa a fusione a letto di polvere. Al superamento del limite superiore di densità stampabile, infatti, la produzione è formalmente consentita, ma il materiale intrappolato aggiungerà massa rispetto alla densità progettata. Pertanto verranno analizzati tipo, dimensione e frazione di volume della cella unitaria per raggiungere le proprietà desiderate e, in base a questi, generare gradienti volumetrici.

2.1 Strumenti: plug-in di Grasshopper per la modellazione volumetrica

In questo lavoro di tesi, le strutture reticolari a gradiente di porosità verranno progettate utilizzando Grasshopper, un linguaggio di programmazione visivo direttamente integrato nel software CAD commerciale per la modellazione 3D Rhinoceros 7 (Robert McNeel & Associates, USA).

Si è sottolineato come le recenti tecnologie di produzione digitale offrano le condizioni ideali per la creazione di strutture reticolari con elevata complessità geometrica e gradienti funzionali alle applicazioni desiderate. Tuttavia, la progettazione e lo sviluppo di queste strutture intrinsecamente complesse richiedono spesso l'utilizzo di metodi e strumenti software specializzati. La questione di come progettare un oggetto come parte digitale di un progetto concettuale è ancora aperta. Gli approcci CAD esistenti mostrano problemi significativi quando si tratta di forme geometricamente complesse, come reticoli eterogenei e strutture bio-ispirate (Letov & Zhao, 2022). L'industria AM si trova attualmente in una situazione in cui la libertà di produzione supera la libertà di progettazione. È per questo che la scelta del software da utilizzare in questo elaborato è ricaduta su Grasshopper, software di modellazione algoritmica per la generazione e il controllo di forme semplici e complesse, che consente di superare, attraverso l'utilizzo di specifici *plug-in* per la modellazione volumetrica, quelli che sono i limiti derivati dalla sola modellazione di superfici.

Grasshopper (Robert McNeel & Associates, USA) per Rhino consente di generare forme 3D mediante la definizione di un diagramma a nodi in grado di descrivere le relazioni matematiche e geometriche presenti all'interno di un modello. I modelli tridimensionali sviluppati con Grasshopper sono sistemi dinamici e modificabili in tempo reale mediante la variazione dei parametri definiti durante la costruzione del diagramma, con vantaggi immediati sul controllo e sulla razionalizzazione della forma.

Si possono distinguere due grandi classi: i parametri, che contengono un'informazione (numeri, vettori, geometrie, ...), e i componenti, che eseguono un'operazione (traslazione, copia, suddivisione, scala, ...). I componenti, detti *building blocks*, sono raccolti per gruppi di funzioni e vengono organizzati per schede in base alla tipo di funzione che svolgono. Questi vengono trascinati e collegati all'interno dello spazio di progetto, chiamato *canvas*, visualizzando il risultato in una finestra grafica. Ogni componente riceve uno o più input ed elabora uno o più output: il processo di collegamento dei vari componenti forma l'algoritmo di modellazione.

Questa piattaforma offre anche un'elevata connettività ad altri linguaggi di programmazione, come Python o MATLAB.

Come si visto, con modellazione volumetrica si intende un termine generico per metodi come rappresentazione di funzioni implicite (F-Rep) o diverse strategie di discretizzazione (voxel densi o *octree* sparsi). La modellazione volumetrica, come codifica della geometria, è molto robusta nella gestione delle modifiche alla topologia. Offre un'alternativa alla più comune rappresentazione dei confini (B-Rep) utilizzata dalla maggior parte delle applicazioni di progettazione assistita da computer. Per tutte le ragioni descritte, le rappresentazioni volumetriche sembrano essere l'approccio più adatto per modellare strutture cellulari a gradiente funzionale. Lo spazio è descritto in termini puramente matematici, le operazioni booleane (addizione, sottrazione e intersezione), così come il *blending* graduale tra gli oggetti, vengono eseguiti senza la necessità di calcolare i punti di intersezione e ricostruire la topologia di come i vertici sono collegati. Non c'è geometria - vertici o linee o triangoli - finché non deve essere renderizzata sullo schermo (nemmeno allora necessariamente) o esportata in qualche altro pacchetto CAD o stampante 3D. Qualsiasi numero di primitive può essere facilmente combinato, l'isosuperficie di livello zero creerà sempre una *mesh* a tenuta stagna. Oggetti di qualsiasi complessità possono essere facilmente trasformati in un guscio cavo o riempiti con intricate strutture reticolari, difficili da modellare con il posizionamento esplicito di vertici e superfici.

Poiché in questo scenario sono disponibili solo poche implementazioni di software CAD, si scelgono e si approfondiscono i *plug-in* per la modellazione volumetrica di Grasshopper: *Dendro*, *Axolotl* e *Monolith*.

2.1.1 Dendro

Dendro (ECR Labs, Los Angeles, USA) è un *plug-in* di modellazione volumetrica per Grasshopper basato sulla libreria OpenVDB (Museth, 2013). Rende il lavoro con i volumi non diverso dalla gestione di qualsiasi altra geometria in Grasshopper (Figura 2.1). Funziona con molti componenti nativi, evitando così il "blocco" che si trova in altri *plug-in*.

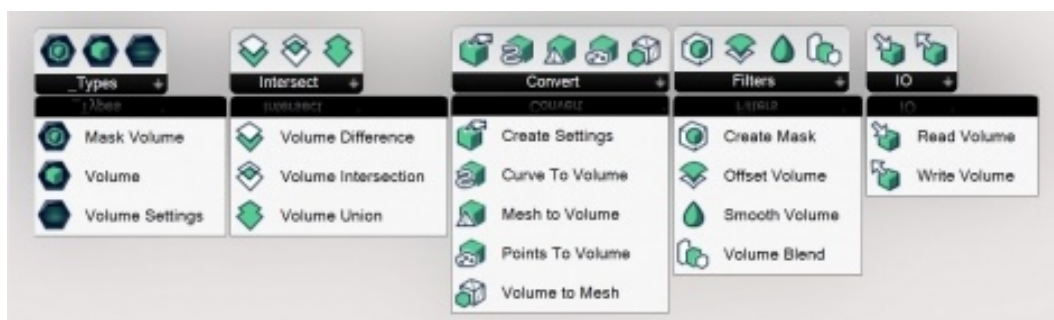


Figura 2.1: *Componenti di Dendro.*

Consente di creare oggetti di volume a partire da *mesh*, curve o punti, e rende possibile trasformare un reticolo curvo precedentemente progettato in un reticolo volumetrico, con controllo dello spessore, *smooth* delle superfici e utilizzando operazioni booleane, consentendo quindi di eseguire varie operazioni sui volumi. Quando si lavora con *mesh* o B-Rep, questo tipo di operazioni sono spesso pesanti dal punto di vista computazionale, soggette a errori o non possono gestire geometrie complesse. Le strutture dei dati del volume di OpenVDB consentono un calcolo più rapido con una maggiore ripetibilità, consentendo di sfruttare operazioni più complesse all'interno di Grasshopper:

- *Wrapping* di *mesh*, curve o punti
- Operazioni booleane veloci
- Offset di geometrie complesse
- Operazioni di *smoothing*
- Output di *mesh* a tenuta stagna
- Alta ripetibilità

Per un approfondimento sul funzionamento e sui vantaggi della libreria OpenVDB si consulti l'appendice A.

Il flusso di lavoro tipico per il *plug-in Dendro* consiste nel generare un volume, eseguire le operazioni su di esso e quindi generare il volume come *mesh*. Gli oggetti volume possono essere creati da *mesh*, curve o punti. Si illustrano dapprima i comandi principali che caratterizzano il *plug-in*.

Fondamentale è il *building block* "Create Settings" (Figura 2.2). Il componente delle impostazioni ha quattro ingressi:

1. Voxel size (S): è la dimensione x, y, z dei singoli voxel che riempiono il volume. È consigliabile tenere questo valore il più grande possibile mentre si lavora, e ridurlo nello step finale. Si può pensare alla *voxel size* come alla risoluzione del volume;
2. Bandwidth (B): la larghezza di banda estende il campo voxel disponibile attorno al volume. I voxel all'interno di questa banda sono impostati attivi, tutto il resto è inattivo. Questo controlla il conteggio dei voxel attivi, quindi è preferibile mantenere il suo valore il più piccolo possibile per ridurre al minimo il tempo di calcolo;

3. Isovalue (I): è l'accuratezza della *mesh* risultante rispetto al valore originale. Può essere astrattamente pensata come un offset positivo o negativo. In genere si desidera tenerlo a zero per mantenere la precisione al volume effettivo;
4. Adaptivity (A): l'adattività imposta l'uniformità delle facce della *mesh*. I valori possono variare da 0 a 1, con un valore 0 più equalizzato e denso. Una maggiore adattabilità consentirà una maggiore variazione nella dimensione del poligono, con conseguente minor numero di poligoni e calcoli più rapidi.

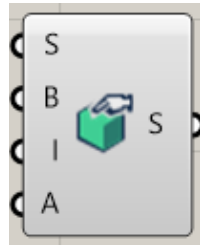


Figura 2.2: *Componente Create Settings*

Con una comprensione delle basi delle impostazioni del volume, si possono utilizzare i componenti di conversione per passare da e verso i dati del volume. Per convertire la geometria in volumi, è possibile utilizzare i componenti “Mesh To Volume”, “Curve To Volume” e “Points To Volume”. Si usa “Volume To Mesh” per riportare i dati volumetrici in un formato nativo di Grasshopper, quindi in una *mesh* chiusa.

Il componente “Points To Volume” genera volumi dai dati di punti. Simile a “Curve to Volume”, questo componente prende un raggio e crea una sfera di volume in ogni punto. Può essere utilizzato per replicare le caratteristiche della “Curve to Volume”, ma con un controllo maggiore.

“Curve to Volume” prende qualsiasi curva come input e genera da essa un volume semplice. La curva è avvolta da un profilo sferico di un determinato raggio. L'input del raggio definisce la dimensione di queste sfere in base alle unità globali di Rhino.

I filtri sono un gruppo di componenti *Dendro* che modificano gli oggetti volume. I filtri offrono una straordinaria flessibilità nella modifica dei dati di input volumetrici. Utilizzando altri volumi come maschere, si possono perfino isolare aree specifiche a cui applicare il filtro.

“Smooth Volume” esegue un'operazione di *smoothing* su un volume di input, che consiste nell'applicazione di una funzione di filtro il cui scopo è evidenziare i *pattern* significativi, attenuando il rumore (Figura 2.3). Il componente ha il seguenti input:

1. Volume (V): è il volume di input su cui verrà eseguita l'operazione regolare. L'input può essere uno o più volumi;

2. Type (T): esistono quattro tipi di levigatura disponibili, ovvero Gaussiana (0), Laplaciana (1), Media (2) e Mediana (3).
3. Width (W): la larghezza può essere considerata come la scala dell'effetto levigante sul volume in ingresso. L'input deve essere un numero intero positivo. L'input della larghezza non ha effetto sull'attenuazione del tipo laplaciano;
4. Iterations (I): è il numero di volte in cui l'operazione di *smoothing* viene eseguita sul corpo di input.

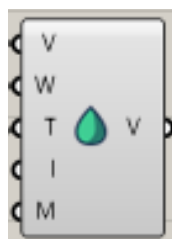


Figura 2.3. *Componente Smooth Volume*

Il componente “Offset” esegue l'offset del confine esterno di un volume. L'input Distanza (D) specifica la quantità di offset.

Il componente “Blend” accetta due input di volume qualsiasi e li trasforma tra di loro. La posizione in cui valutare il *blend* è controllata dall'input del parametro (t) (intervallo 0-1). Il numero End Time (E) stabilisce il limite superiore del passo temporale ed è legato alla dimensione del voxel.

I componenti del filtro di *Dendro* (*Smooth*, *Offset* e *Morph*) hanno tutti un ingresso "Mask (M)", che consente di eseguire l'operazione di filtraggio in aree specifiche. I dati della maschera vengono creati dal componente “Create Mask”. La maschera è definita da un volume che rappresenta i limiti della maschera. Le maschere funzionano in modo simile alla funzione maschera che si trova in un editor di immagini 2D, consentendo all'utente di definire o limitare un'area che sarà interessata da un filtro.

Le operazioni booleane costituiscono una caratteristica molto interessante di *Dendro*. Laddove le *mesh* booleane (e persino NURBS) possono essere soggetti a problemi, le booleane di volume sono incredibilmente stabili e possono essere computazionalmente più leggere. Le operazioni booleane standard disponibili sono Unione, Differenza e Intersezione.

L'Unione combina tutti i volumi di input sovrapposti in un unico corpo. Il componente deve avere più volumi di input sullo stesso ramo di un albero per funzionare correttamente.

L'Intersezione emette qualsiasi area volumetrica sovrapposta da entrambi gli ingressi A e B. Gli input possono essere volumi singoli o liste.

La Sottrazione booleana sottrae l'input B dall'input A. Gli input possono essere volumi singoli o liste.

2.2.2 Axolotl

Axolotl (Mathias Bernhard, Zurich, Volumetric Modelling - Dbt, n.d.) è un *plug-in* che fornisce un insieme di componenti per la modellazione volumetrica in Grasshopper. Funzionano in base al principio delle funzioni di distanza con segno (SDF). La modellazione volumetrica, come precedentemente analizzato, si basa anche sulla rappresentazione di funzioni (F-Rep). Lo spazio è considerato un campo di valori pieno e continuo. Questi valori possono essere descritti da una funzione in modo che per qualsiasi coordinata in un dato spazio 3D, vi sia un rispettivo valore che può essere mappato a quella coordinata. Invece di definire un oggetto 3D come una raccolta di poligoni o superfici e curve, viene descritto in termini puramente matematici come un campo di valori associato a coordinate (Stieler et al., 2022).

Molte primitive geometriche (sfera, cubo, cono, cilindro, piano, solidi platonici, ...) possono infatti essere descritte non esplicitamente posizionando i vertici nello spazio e collegandoli con spigoli e facce, ma tramite una funzione $v = f(x, y, z)$. In questa formula, v è la distanza del punto (x, y, z) dalla superficie della forma. L'appartenenza al punto dipende dal segno della funzione di definizione valutata in quel punto. Tutte le posizioni dei punti in cui questa funzione restituisce $v = 0$ formano la *skin* dell'oggetto, detta isosuperficie, che divide inequivocabilmente l'interno dall'esterno. Tutti quelli che danno un valore negativo si trovano all'interno dell'oggetto, tutti quelli con valori positivi si trovano al di fuori di esso. Tale funzione è chiamata *distance function* e lo spazio è definito come campo di distanza con segno (SDF) (Figura 2.4).

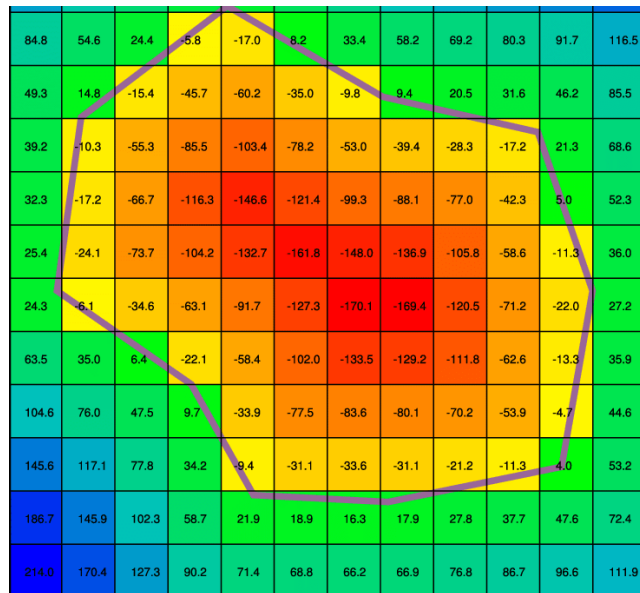


Figura 2.4. Rappresentazione del campo della distanza con segno di un ostacolo poligonale 2D. Lo spazio è discretizzato in celle. Le celle che memorizzano una distanza negativa si trovano all'interno dell'ostacolo; le celle con una distanza positiva sono all'esterno

La combinazione di oggetti ora non è un problema geometrico di calcoli difficili di intersezioni, ma puramente aritmetica. Se l'oggetto A è definito da $a = f(p)$ e l'oggetto B con $b = g(p)$, con $p \in \mathbb{R}^3$, allora l'unione booleana $A \cup B$ è definita come $\min(a, b)$, l'intersezione $A \cap B$ come $\max(a, b)$ e la sottrazione $A - B$ come $\max(a, -b)$. Sono possibili molte più combinazioni con a e b, per esempio per produrre raccordi lisci, angoli di smusso, transizioni a gradini o scanalature.

Molte operazioni come teoria degli insiemi, *blending*, offset, proiezioni, deformazioni non lineari, *sweeping* e altre sono state formulate per questa rappresentazione in modo tale da produrre funzioni continue a valori reali come output, quindi garantire la proprietà di chiusura di questa rappresentazione. Le funzioni R originariamente introdotte forniscono una continuità C^k per le funzioni che definiscono esattamente le operazioni di teoria degli insiemi. A causa di questa proprietà, il risultato di qualsiasi operazione supportata può essere considerato come input per un'operazione successiva; quindi modelli molto complessi possono essere creati in questo modo da un'unica espressione funzionale.

Nelle rappresentazioni di confine (B-Rep), gli oggetti sono definiti da vertici collegati da spigoli e facce. Quando si lavora con *mesh* o B-Rep, l'ispessimento delle linee dello scheletro di una struttura avviene tipicamente costruendo due anelli di punti alle due estremità di ciascuna linea con un offset predefinito dai suoi punti finali e collega gli anelli per formare cilindri. Per ogni nodo, vengono collezionati tutti i punti dell'anello, viene calcolato un involuppo convesso del poliedro e vengono aggiunte le facce dei cilindri alla *mesh* finale. Funziona bene in molti casi,

ma dà errore quando vi sono angoli molto acuti tra due linee collegate o piccole aree di superficie all'interno di anelli di linee, che risultano in *mesh* autointersecanti (Hesselgren et al., 2018). La costruzione basata su F-Rep tratta l'ispessimento come l'unione di più funzioni cilindriche. Ogni volta che le aree internodali diventano troppo piccole o i cilindri troppo spessi per un'apertura, la topologia viene semplicemente modificata senza causare errori (Figura 2.5). Le linee non devono essere collegate end-to-end, ma possono anche formare giunti a T. La microstruttura risultante definisce un *signed distance field* (SDF).

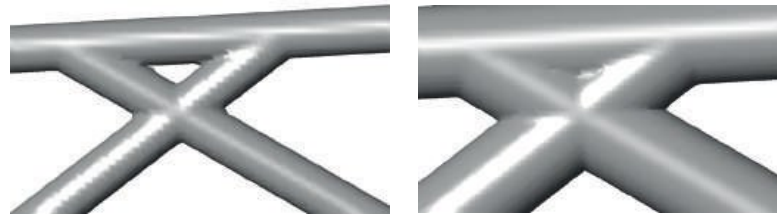


Figura 2.5: *Ispessimento delle linee con cilindri F-Rep; il foro diventa un nodo solido all'aumentare del diametro, senza causare problemi topologici*

Un tipico flusso di lavoro con *Axolotl* consiste nello scegliere oggetti da *primitives*, combinarli usando ad esempio operazioni booleane da *combinations*, apportare ulteriori *modifications*, se necessario, e infine visualizzare l'oggetto distanza utilizzando un componente dalla scheda *meshing*. Inoltre, ci sono diversi tipi di *lattice*, alcune funzioni *math*, componenti per convertire la *geometry* di Rhino in un SDF e alcune utilità di *analysis* dell'SDF (Figura 2.6).

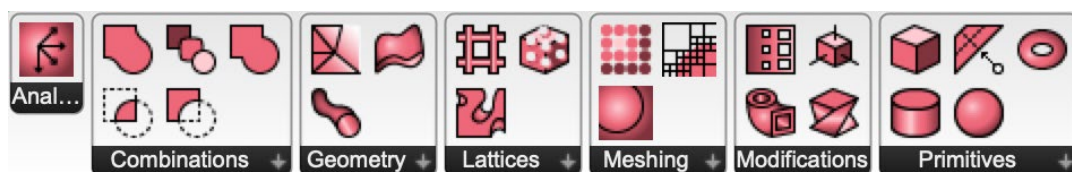


Figura 2.6. *Palette del plug-in Axolotl*

Gli oggetti tridimensionali, per applicazioni come il rendering o la produzione additiva, sono definiti dalla loro *skin* che separa l'interno dall'esterno. Per generare questa superficie limite utilizzando un algoritmo *marching cubes*, solo i valori di distanza per i voxel che contengono un frammento della superficie devono essere calcolati alla massima risoluzione. Il calcolo e la memorizzazione dei valori di distanza per griglie di voxel densi sono computazionalmente molto costosi e vengono moltiplicati di un fattore otto tagliando a metà la lunghezza del lato del voxel. A seconda dell'applicazione, per vaste regioni del volume non è richiesto il calcolo di un valore di distanza preciso e sarà sufficiente solo un'approssimazione grossolana del valore. La complessità computazionale dei modelli voxel, infatti, è $O(n^3)$ (Adalsteinsson & Sethian, 1995). Una soluzione a questa sfida per mitigare la complessità e ridurla a un $O(n^2)$ è

utilizzare il metodo degli *sparse voxel octree* (SVO). Il metodo SVO tenta anche di semplificare il processo di rendering dei voxel combinando grandi porzioni di voxel, riducendo così efficacemente la loro quantità. Il metodo SVO funziona come segue:

1. La cella radice viene creata per comprendere la regione interessata (ad esempio il cubo di delimitazione minimo di tutti gli oggetti).
2. Ogni cella memorizza il suo punto centrale, la lunghezza del lato e_1 , una lista di celle figlie (vuota all'inizio) e il suo livello di suddivisione n . La cella radice ha livello $n=0$.
3. La posizione centrale della cella viene inserita come *query point* nelle funzioni di distanza cumulata dell'albero CSG.
4. Se la distanza restituita è inferiore a $\sqrt{3}/2 * e$ (metà della diagonale del cubo), vengono create otto celle figlie dividendo la cella originale in due lungo tutti gli assi X, Y e Z. Il livello dei nodi figli è $n+1$ e la loro lunghezza del lato è $e/2$.
5. Il processo dallo step 3 viene ripetuto ricorsivamente per tutti gli otto nodi figli, fino a raggiungere un livello di suddivisione massimo specificato.

La Figura 2.7 mostra il risultato di questa procedura come equivalente 2D, una suddivisione *quadtree*. Il colore è dato dalla distanza del centro della cella dal bordo della forma, il numero nei quadrati indica il livello di suddivisione n .

Axolotl crea le isosuperfici attraverso l'algoritmo *marching cubes* con differenti strategie di discretizzazione: da una griglia densa di voxel, ma anche da un *octree*.

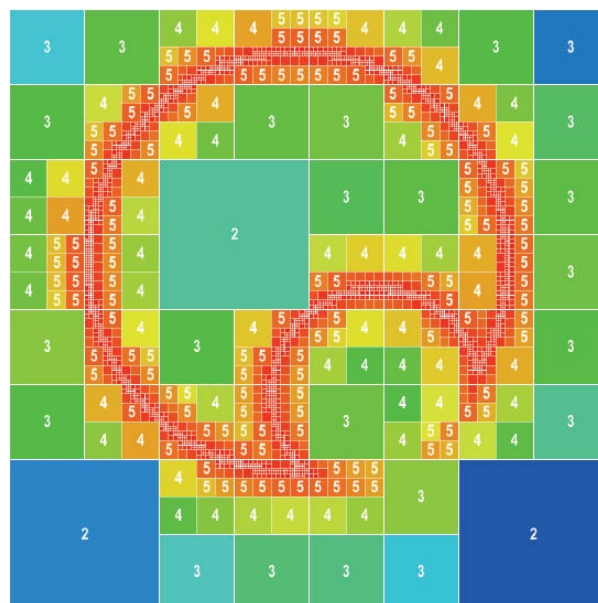


Figura 2.7. Visualizzazione *quadtree*

A seguire, una breve panoramica dei comandi più importanti utilizzati negli algoritmi di modellazione con *Axolotl*.

I comandi “Octree” e “MCubeOctree” creano, rispettivamente, una suddivisione SVO e un’isosuperficie *marching cubes* da un *octree*; “MCubeGrid” e “DenseGrid” creano, rispettivamente, un’isosuperficie *marching cubes* da una griglia densa e campionano un oggetto SDF in una griglia densa. Si può quindi osservare che, per poter creare la *mesh*, vi sono vari componenti, a seconda che si voglia creare un’isosuperficie da una suddivisione SVO oppure da una griglia densa di voxel.

All’interno del componente “Lattice” è presente un database per creare vari tipi di celle. In input chiede il tipo di *lattice* che si vuole generare, la dimensione della cella unitaria e lo spessore delle aste, per fornire in output l’oggetto basato su SDF. Similmente è possibile creare anche TPMS.

Come per *Dendro*, su *Axolotl* si possono utilizzare le operazioni booleane, che, come output, restituiscono un oggetto basato su SDF.

2.2.3 *Monolith*

Monolith (Michalatos & Payne, 2016) è un editor di modelli basato su voxel. La modellazione basata su voxel sta guadagnando notevole popolarità per le sue applicazioni alla modellazione di strutture reticolari eterogenee.

È un progetto di ricerca incentrato sullo sviluppo di nuove rappresentazioni digitali e flussi di lavoro di progettazione che abitano lo spazio tra la tradizionale progettazione assistita da computer (CAD) e l’elaborazione di immagini tridimensionali basata su voxel, con l’obiettivo di consentire un livello di controllo molto fine sulla distribuzione volumetrica del materiale, soprattutto alla luce delle nuove tecnologie di stampa 3D multimateriale. Per la produzione convenzionale, la scelta di una corretta dimensione dei voxel può essere ambigua, specialmente con modelli solidi con un alto grado di curvatura. Tuttavia, per la produzione additiva, non è necessario che la dimensione dei voxel superi la tolleranza di una stampante 3D, poiché la risoluzione dei voxel più elevata non sarebbe realizzabile (Y. F. Zhao et al., 2021). I modelli basati su voxel offrono flessibilità computazionale, consentendo operazioni con matrici 3D. *Monolith* può essere utilizzato come programma *stand alone* o come estensione di Rhino. Questo plug-in offre una suite completa di strumenti che danno la possibilità di costruire rapidamente modelli dettagliati basati su voxel e visualizzarli in modi unici utilizzando la pipeline di rendering OpenGL personalizzata (Figura 2.8).



Figura 2.8: *Componenti di Monolith.*

Per raggiungere questo obiettivo sono stati affrontati tre problemi dai progettisti di questo *plugin*. Il primo corrisponde alle strutture dati per descrivere i campi materiali, il secondo è l'interazione dell'utente e i flussi di lavoro di progettazione e il terzo consiste nei tipi di output e di elaborazione che potrebbero collegare i modelli digitali ai processi di fabbricazione additivi.

Per la struttura dati, si utilizzano modelli voxel multicanale. Similmente a un'immagine che contiene canali RGB, la struttura dati di *Monolith* contiene una serie di canali che definiscono diverse proprietà geometriche e di materiale. Ciascun canale rappresenta un tipo di campo che descrive un valore scalare che cambia da punto a punto nello spazio.

Il canale più elementare è il canale della forma, che determina il confine solido/vuoto. Da questo campo si estraggono due isosuperfici nidificate che determineranno i confini esterni e interni del volume solido. Un secondo canale determina il rapporto di miscelazione del materiale e sarà interpretato utilizzando tecniche tipografiche tridimensionali in un micro-pattern con porosità e anisotropia desiderate. In sostanza, questo canale descrive il rapporto di miscelazione dei materiali in ogni punto dello spazio. Infine, tre canali aggiuntivi descrivono i campi per le coordinate volumetriche della texture (U, V, W) che consentono il flusso arbitrario di micro-modelli all'interno del volume. Oltre a ciò, la struttura dati è sufficientemente estendibile per facilitare lo scambio di informazioni con i moduli di simulazione. Ad esempio, è possibile aggiungere canali supplementari a un modello voxel che contiene le informazioni geometriche di base per memorizzare i risultati di una simulazione di analisi strutturale.

Ciascun voxel può essere utilizzato come contesto di disegno. Sono supportate operazioni come il disegno anti-aliasing di linee, punti, piani, modelli di voxel trasformati e *mesh*. Queste operazioni utilizzano il super-campionamento tridimensionale per evitare effetti di pixellazione e aliasing lungo i confini.

L'approccio di *Monolith* si concentra sulla fluidità dell'esperienza dell'utente cercando di fornire un'interfaccia completa per la modellazione volumetrica. L'interazione col modello è mediata attraverso un paradigma di compositing a strati non dissimile dalle tradizionali applicazioni di elaborazione delle immagini. I livelli stessi possono ospitare interfacce utente

estese con flussi di lavoro autonomi e sono organizzati in una pila che viene composta per creare il modello finale. Ogni livello rappresenta un'operazione di modellazione (per esempio la voxelizzazione di una *mesh*) o un'operazione di filtraggio (per esempio la mappa di distanza con segno). La composizione dei livelli è mediata da una funzione di composizione che determina il tipo di fusione tra un livello e quello al di sopra nella pila. Queste funzioni di composizione semplicemente corrispondono a differenti operazioni aritmetiche ponderate tra i corrispondenti valori dei voxel. Geometricamente corrispondono alle operazioni booleane. L'addizione corrisponde all'unione, la differenza alla sottrazione e la moltiplicazione all'intersezione dei solidi. Tuttavia, poiché i modelli voxel sono sfocati, sono caratterizzati da gradienti piuttosto che da semplici condizioni di solido/vuoto. Operazioni booleane e di morphing molto complesse che metterebbero alla prova la maggior parte dei software basati su B-Rep risultano banali quando si utilizza una rappresentazione voxel.

L'utente può aggiungere vari livelli di generatori di campi voxel che consentono di definire in modo interattivo gradienti tridimensionali. I più notevoli corrispondono al livello basato sulla formula, che genera campi voxel da una formula fornita dall'utente, che viene valutata in ogni posizione x, y, z attraverso il campo voxel. I metodi di modellazione basati su immagini si basano su immagini in scala di grigi fornite dall'utente che vengono utilizzate come sezioni trasversali nelle operazioni di loft o estrusione. Il *layer* di campo della distanza geometrica consente l'uso di primitive geometriche convenzionali (linee, punti, cerchi, riquadri) come geometrie di base per la generazione di un campo voxelizzato attorno a esse. Il livello di modifica della griglia consente un modo più controllato e strutturato di definire gradienti arbitrari nello spazio. Il livello di *free paint* consente all'utente di modificare il campo applicando operazioni locali in modo interattivo. I campi voxel creati dai livelli del generatore possono poi essere manipolati utilizzando vari filtri ed effetti.

Per rendere possibile la progettazione gerarchica dei materiali, l'utente deve essere in grado di controllare la forma della microstruttura (cioè la sua porosità variabile) da un punto all'altro dello spazio e la sua direzionalità. Il controllo di queste proprietà consente la realizzazione di materiali con interessanti caratteristiche ottiche ed elastiche anisotrope. Ci sono tre elementi che controllano il *micro-pattern* all'interno del volume. Il primo è il canale del rapporto materiale che determina il gradiente macroscopico che si vuole ottenere. Questo gradiente viene reinterpretato come porosità variabile o rapporto di miscelazione per più materiali. Il secondo elemento è il *pattern* da ripetere in tutto il volume per ottenere l'effetto *half-toning* tridimensionale. Questo motivo viene ripetuto e moltiplicato con il campo del rapporto materiale. Il risultato di questa operazione dopo la binarizzazione è un *micro-pattern* la cui

porosità dipende del campo del rapporto materiale sottostante. Infine, per determinare come il *micro-pattern* scorre attraverso il volume, è possibile utilizzare un insieme di tre canali che corrisponderebbero alle coordinate UVW (Figura 2.9).

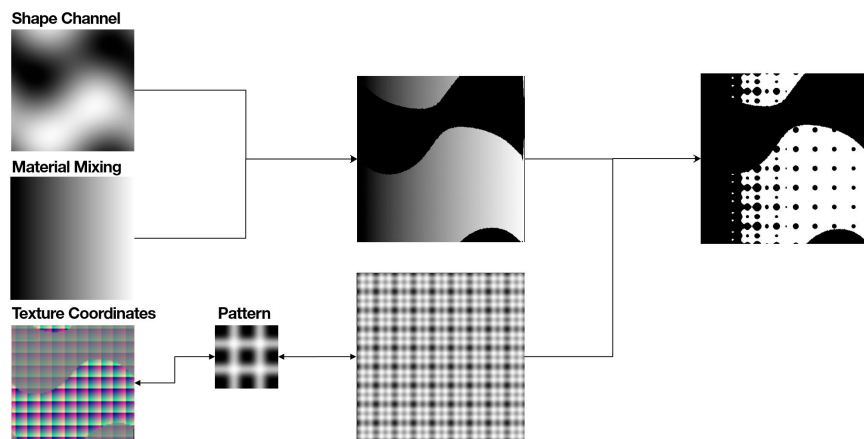


Figura 2.9. Composizione di una mappa di micro-pattern e half-toning.

L'utente può controllare il flusso delle coordinate UVW utilizzando varie estensioni volumetriche dei metodi convenzionali di generazione delle coordinate della *texture*. Inoltre, *Monolith* consente di importare una *mesh* e quindi propaga le coordinate UV della *mesh* all'intero campo voxel lungo la funzione di distanza con segno, che diventa la coordinata W. L'utente può definire i *micro-pattern* sia attraverso semplici immagini bidimensionali in scala di grigi (es. bitmap), sia attraverso immagini volumetriche che ha creato all'interno di *Monolith* stesso.

Per consentire la post-elaborazione o la fabbricazione dei modelli, è necessario tradurli per adattarli ad output compatibili con i software B-Rep esistenti e i driver di stampa. L'output più semplice è una *mesh* del livello impostato di superficie. Tuttavia, per descrivere i gradienti di materiali sono state aggiunte due opzioni. La prima consiste nell'esportare una cascata di *mesh* chiuse che rappresentano diverse regioni di gradiente lungo il canale di miscelazione del materiale. Per ottenere ciò si creano le isosuperfici del canale di rapporto di materiale che vengono poi tagliate col canale della forma. Questo metodo consente l'uso del modello con stampanti multimateriale che supportano i gradienti indirettamente attraverso l'assegnazione di diversi materiali a *mesh* adiacenti. Un secondo metodo per tradurre i modelli voxel continui in qualcosa che assomigli a un gradiente consiste nell'utilizzare una *mesh* iperfina che rappresenta il modello di porosità variabile di un materiale che si mescola in un altro.

I modelli voxel sono, in un certo senso, più facili e diretti da simulare in quanto non richiedono una fase di discretizzazione che complica la maggior parte delle simulazioni agli elementi finiti

dei modelli basati su B-Rep. Grazie al passo costante nel campo voxel, alcune operazioni come la costruzione di matrici di rigidità sono molto semplificate, con un significativo guadagno di prestazione.

Tra i componenti principali di *Monolith* si presenta “Construct Base Voxel”, che costruisce una classe voxel generica basata sul riquadro di delimitazione, sulla risoluzione, cioè l’impostazione della dimensione del più piccolo voxel in cui viene discretizzato il modello, e sulla designazione del canale.

Con il comando “Construct Geometric Source” si definisce una sorgente geometrica per la generazione del campo.

Per assegnare ai voxel le informazioni desiderate, si utilizza il comando “Add Source”: in input vi sono i voxel a cui assegnare le proprietà, il canale della *shape* e, facoltativo, il canale del *material ratio*, che identifica la densità del materiale o i canali Red, Green, Blue. A ogni voxel vengono infatti associate due informazioni: la prima identifica se il *voxel* è “pieno” o “vuoto”, la seconda si riferisce ai materiali presenti.

Una volta ottenuti i voxel, si può generare la *mesh* col comando “Get Iso Mesh”. Si possono utilizzare i valori provenienti da un determinato *channel*, quindi utilizzando il blocchetto “Get Channel Names”. Infine si deve inserire l’intervallo degli isovalori.

2.2 Metodo di progettazione delle FGLSs

Noti gli strumenti di modellazione volumetrica disponibili, in questa sezione si illustra il metodo utilizzato per la progettazione di strutture reticolari con gradiente di porosità. Ecco la schematizzazione del metodo proposta (Tabella 2.1):

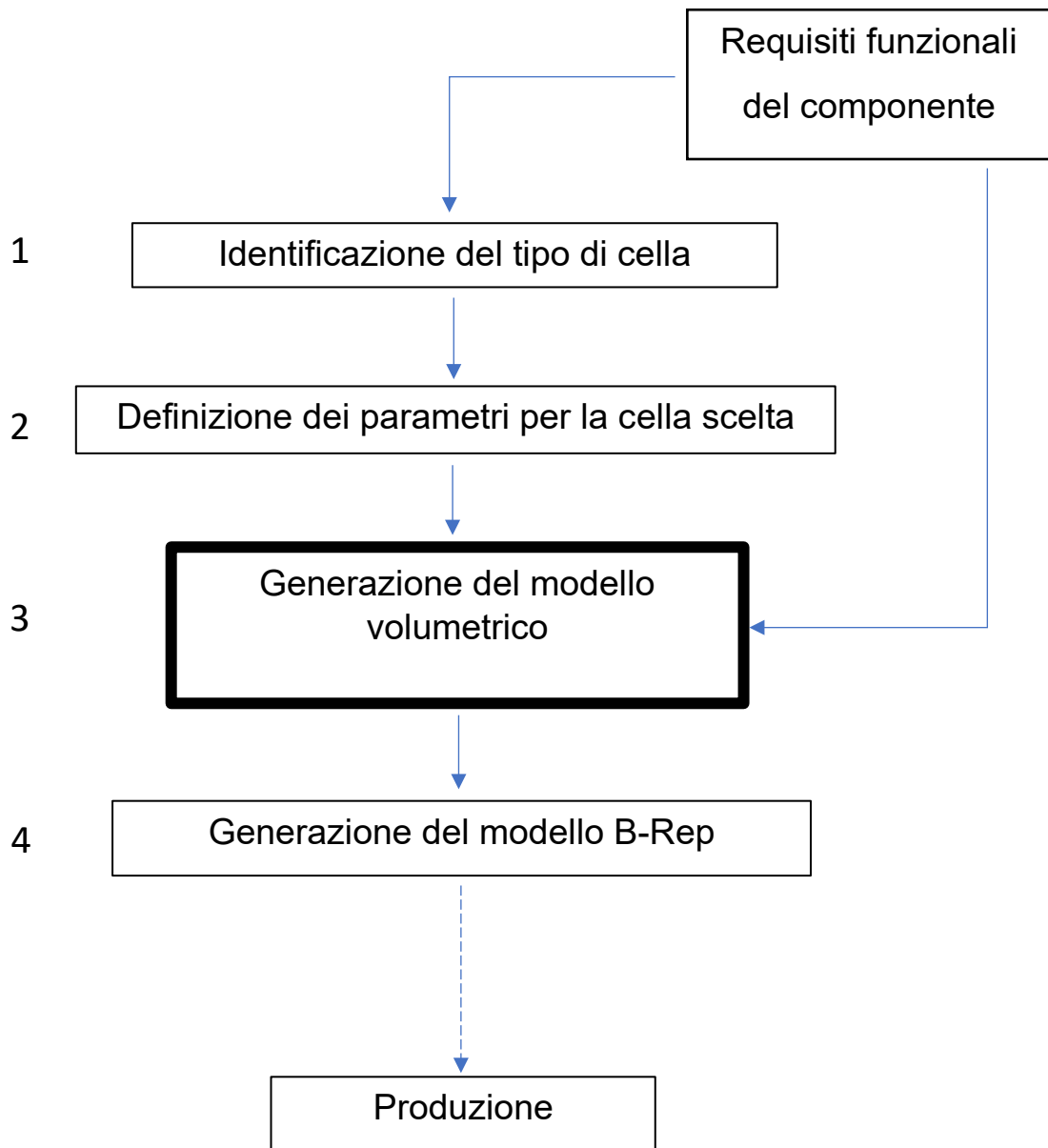


Tabella 2.1. Schematizzazione del metodo proposto per la creazione delle strutture reticolari graduate

Di seguito si illustrano le varie fasi.

1. Identificazione del tipo di cella

Nella prima fase del metodo presentato, si identifica il tipo di cella utilizzato, che si può scegliere in base ai requisiti funzionali che la struttura deve soddisfare.

Per quanto riguarda le strutture *truss-based*, si analizzano le seguenti celle (Figura 2.10):

- *Octet-truss*

Struttura *stretching dominated*, cubica a facce centrate ortotropa, non esplorata a fondo in passato. Mostra un'elevata rigidezza e resistenza specifiche con una bassa densità relativa. Tuttavia, la struttura dell'ottetto con bassa densità relativa non è stabile e si traduce in una significativa diminuzione dello stress dopo il primo picco, il che la rende inappropriata per l'assorbimento di energia, ma più adatta per applicazioni strutturali leggere (Xiao et al., 2020).

- Cubica a corpo centrato

Tipico esempio di struttura *bending dominated*, tra le più utilizzate e studiate, con studi che evidenziano le sue speciali proprietà meccaniche di alta resistenza, stabilità e assorbimento di energia (M. Zhao et al., 2022);

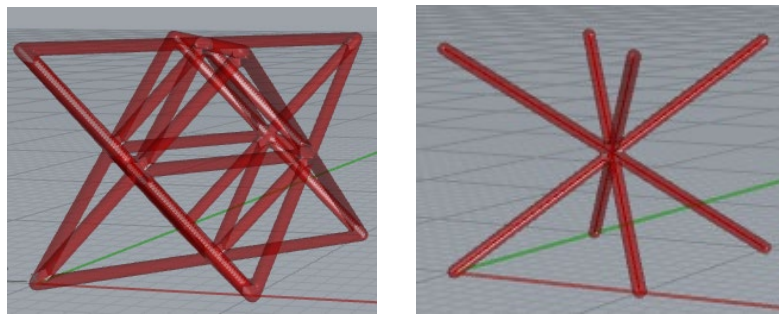


Figura 2.10. Schematizzazione di: a) octet lattice; b) BCC lattice

Una struttura *bending-dominated* è dominata dalla flessione, una *stretch-dominated* è dominata dallo stiramento. La selezione di una di queste topologie dipende, tra gli altri fattori, dalle prestazioni desiderate per il reticolo e dalle proprietà del materiale selezionato per la produzione (Figura 2.11).

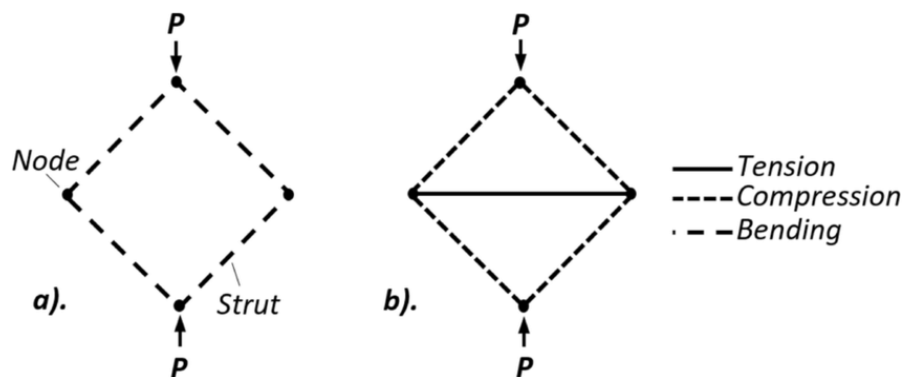


Figura 2.11. Struttura: a) *bending-dominated*, b) *stretch-dominated* (Alghamdi et al., 2020)

Le topologie *bending-dominated* hanno un modulo e una resistenza allo snervamento inferiori rispetto alle tipologie *stretch-dominated* con la stessa densità relativa, rendendo le

celle dominate dallo stiramento più adatte per strutture leggere e le celle dominate dalla flessione più adatte per assorbire energia.

Per quanto riguarda le strutture *shell-based*, si scelgono di analizzare le TPMS:

- Schwarz P
- Giroide
- Schwarz D

La Schwarz-P si può utilizzare in applicazioni che richiedono rigidità e resistenza elevate in una direzione di carico ben definita. Se, invece, l'applicazione richiede che la parte reticolare subisca un'elevata deformazione prima della rottura, sarebbe preferibile un reticolo giroide o Schwarz Diamond (Maskery et al., 2018). La primitiva di Schwarz, inoltre, è tra le celle unitarie che mostrano maggiore permeabilità ai fluidi, rendendola ideale in termini di osteointegrazione (Jung & Torquato, 2005).

2. Definizione dei parametri per la cella scelta

L'osso è un biomateriale poroso complesso con proprietà anisotropiche non omogenee che supporta organi, muscoli e altri tessuti del corpo. La sua struttura – e quindi le sue proprietà meccaniche – sono il risultato della sua funzionalità richiesta: la porosità e la direzione delle fibre del tessuto osseo sono controllate da processi biologici che adattano la topologia del tessuto locale ai requisiti meccanici. Ciò porta a un'ampia gamma di porosità, proprietà meccaniche e anisotropia all'interno dello stesso osso, tutte ottimizzate per la sua funzionalità macroscopica ottimale. È noto che questa struttura macroscopica è in equilibrio mecano-biologico locale: quando il tessuto osseo locale non è sottoposto a stress, il flusso locale di nutrienti viene interrotto. Questo porta a una riduzione del tessuto osseo. Lo stimolo meccanico locale è quindi necessario per mantenere vivo e sano il tessuto osseo. Ciò può essere ottenuto solo quando il sistema vascolare ospitato dalla struttura porosa dell'osso trasporta un flusso continuo di nutrienti. Pertanto questi due aspetti (stimolo meccanico e supporto nutritivo per l'osteoiduzione) diventano fondamentali per una progettazione ottimale dell'impianto osseo (Barba et al., 2019).

Molti studi hanno collegato le proprietà meccaniche del reticolo alla geometria e porosità e diversi autori hanno studiato l'effetto della geometria (dimensione dei pori e cella reticolare) nelle prestazioni di osteointegrazione. Warnke et al. hanno studiato l'effetto della dimensione dei pori (450-1200 μm) sulla crescita degli osteoblasti: hanno scoperto che i pori inferiori a 600 μm sono adatti per la completa crescita del tessuto (Warnke et al., 2008). Van Bael et al. hanno

studiato l'effetto di diverse geometrie dei pori (triangolari, quasi circolari e quadrati) e dimensioni dei pori (500 μm e 1000 μm) su popolazioni di cellule viventi e sui loro tassi di differenziazione. Hanno scoperto che i pori circolari più piccoli - 500 μm - migliorano l'attaccamento cellulare iniziale, ma è più probabile che i pori più grandi - 1000 μm - ospitino cellule viventi dopo 14 giorni (van Bael et al., 2011).

All'aumentare dello spessore delle strutture reticolari che si andranno a progettare corrisponde una riduzione della dimensione nominale media dei pori, che è stata sperimentalmente correlata alla densità relativa della struttura, come illustrato nella Figura 2.12: la densità relativa aumenta (logaritmicamente) con la dimensione dei pori decrescente.

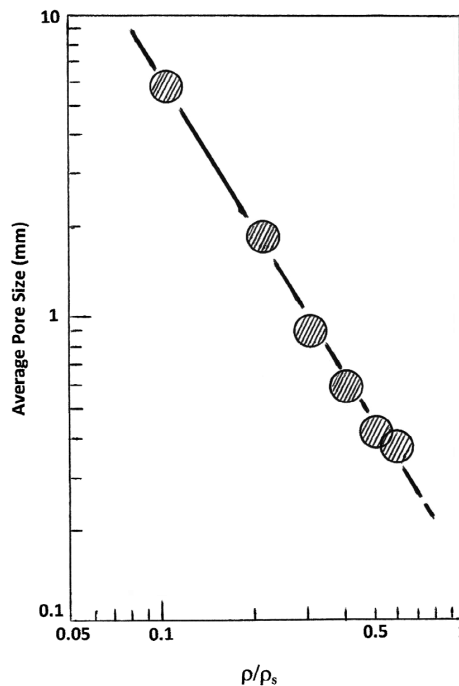


Figura 2.12. Dimensione media dei pori a cellule aperte rispetto alla densità relativa per prodotti fabbricati Ti-6Al-4V (Murr, 2018)

È ormai ben stabilito che la porosità in generale, essenzialmente per qualsiasi materiale poroso a struttura cellulare aperta, è correlata alla rigidità o modulo di Young, E , per il materiale scelto, come rappresentato dall'equazione (Murr, 2018):

$$\frac{E}{E_S} = \left(\frac{\rho^*}{\rho_s}\right)^2$$

dove E_S è il modulo elastico (di Young) per il materiale completamente denso (solido) e $\left(\frac{\rho^*}{\rho_s}\right)$ è la densità relativa. La Figura 2.20 riproduce questi dati per *mesh* e schiume a base di Ti-6Al-4V e Co. Come esempio si riportano i moduli di Young (E) e le densità relative per i prodotti in *mesh* o schiuma Ti-6Al-4V per $E = 1$ GPa ed $E = 10$ GPa; corrispondenti ai regimi ossei

femorali prossimi al trabecolare e corticale. Le rispettive porosità sono mostrate come 83% per il modulo di Young di 1 GPa e 56% per il modulo di Young di 10 GPa (Figura 2.13).

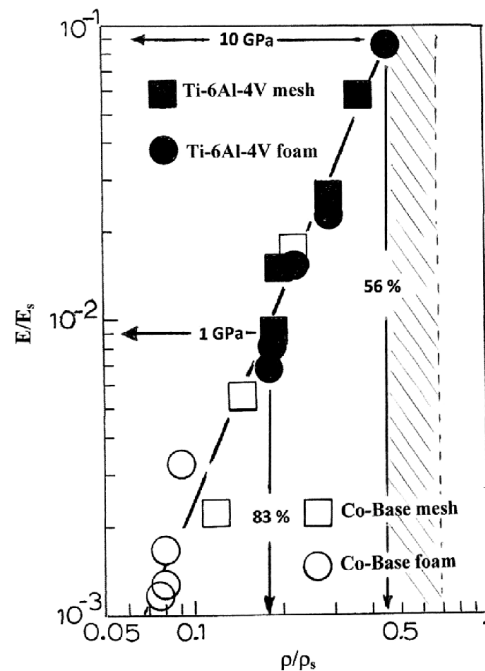


Figura 2.13. Rigidità relativa (E/E_s) rispetto a densità relativa (ρ/ρ_s) per le leghe Ti-6Al-4V e Co-26-Cr-6Mo-0.22C 4V (Murr, 2018)

Pertanto i parametri della cella sono legati alla dimensione dei pori, che influenza vascolarizzazione e accrescimento osseo, al tipo di materiale e alla sua densità, che determinano modulo elastico e caratteristiche meccaniche.

3. Generazione del gradiente

Per generare un gradiente, si esplorano due metodi di modellazione:

- Basati su *distance field*

La variazione spaziale di parametri come la lunghezza e il diametro dell'asta, la dimensione della cella unitaria o lo spessore della *shell* (nel caso di reticoli basati su superfici) può essere definita sotto forma di una funzione matematica per ottenere un gradiente funzionale in una struttura reticolare (*equation driven*).

- Basati sull'isovalore

Quando si modellano le strutture TPMS, si può scegliere di ottenere un gradiente di porosità facendo variare l'isovalore, attraverso la definizione di un'equazione.

In modo alternativo la porosità può essere ottenuta a partire da immagini (*image driven*).

Le immagini possono essere utilizzate come input per definire il gradiente di una struttura reticolare. In alcuni casi, queste immagini sono generate da processi di

ottimizzazione topologica (solitamente in scala di grigi) e vengono utilizzate come mappe di densità per definire i gradienti nei reticoli (Panesar et al., 2018). Le immagini mediche vengono utilizzate come input per l'ingegneria del tessuto osseo e la produzione di impianti ortopedici (Mahmoud & Elbestawi, 2017).

In base ai requisiti funzionali viene stabilita la porosità. Per poterla determinare quantitativamente, si ricavano le equazioni che correlano la densità relativa in funzione dello spessore e della dimensione della cella, o dell'isovalore. La porosità, ovvero il grado di vuoto all'interno di un materiale, può infatti essere espressa come il complemento a uno della densità relativa, cioè $1 - \rho^*/\rho_s$. Partendo dal presupposto che il volume di una struttura reticolare è dato dal volume della parte solida sommato al volume della parte porosa, si può dimostrare che:

$$\rho_{rel} = \frac{\rho^*}{\rho_s} = \frac{\frac{m}{V_{str}}}{\frac{m}{V_s}} = \frac{V_s}{V_{str}} = \frac{V_{str} - V_p}{V_{str}} = 1 - P$$

dove ρ^* è la densità del materiale cellulare, ρ_s è la densità del solido, V_{str} è il volume totale, m la sua massa, V_s è il volume del solido, V_p il volume dei pori, P è la porosità, ovvero la frazione volumetrica di poro rispetto al volume totale.

Nello specifico, il primo obiettivo che ci si pone è trovare l'andamento della densità relativa della struttura reticolare *truss-based* rispetto al rapporto tra raggio dell'asta e lunghezza della cella. Per far ciò, si inizia definendo le dimensioni delle celle unitarie di base e la dimensione del raggio delle aste, che verrà fatto variare.

Nella tabella sotto riportata si definiscono i valori utilizzati (in mm) nella realizzazione delle celle delle due strutture *truss-based* di riferimento (Tabella 2.2). Con L si intende la lunghezza della cella, con r_{min} la lunghezza del raggio minimo delle aste e con *voxel size* la dimensione del voxel con cui si discretizza il volume della struttura.

	L	r_{min}	<i>voxel size</i>
Octet-truss	10	0.2	0.1
BCC	4	0.2	0.1

Tabella 2.2. Parametri scelti per le celle octet-truss

Per creare i modelli CAD dei reticoli, si implementa l'algoritmo tramite Grasshopper, Rhinoceros e i dati ottenuti vengono analizzati in MATLAB (Cleve Moler, USA), ambiente per il calcolo numerico e l'analisi statistica.

Le celle vengono ripetute due volte lungo ciascuna direzione tramite *ArrayBox*, e tutte le curve risultanti vengono unite col comando “Join”, prima di essere importate in Grasshopper per realizzare lo spessore tramite il componente del *plug-in Dendro* “Curve To Volume” (Figura 2.14). Lo spessore, ovvero il raggio delle aste, viene fatto variare da un minimo di 0.2 mm fino a quando la struttura raggiunge l’ingombro massimo e si scollega automaticamente.

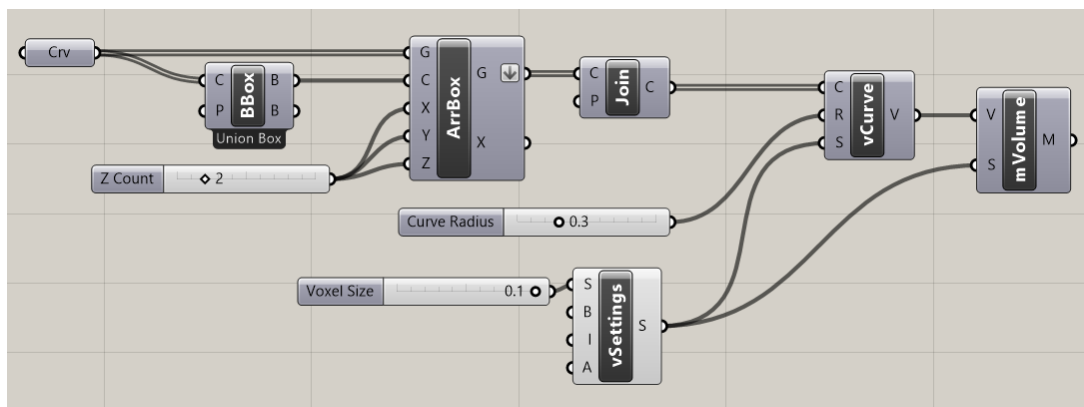


Figura 2.14. Algoritmo per la modellazione delle celle unitarie truss-based

Si può ora effettuare un *bake* della geometria e lavorare su Rhinoceros: la *mesh*, ottenuta col comando “Volume To Mesh” di *Dendro*, mantenendo gli stessi *settings*, viene intersecata con il cubo che delimita la cella tramite il comando “MeshBooleanIntersection”, di cui è necessario conoscere il volume (L^3). La cella presa in considerazione è quella centrale a ciascuna struttura ottenuta.

Per la cella *octet-truss* (Figura 2.15; 2.16):

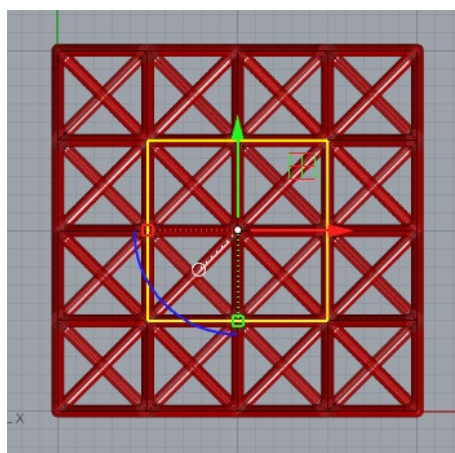


Figura 2.15. Vista superiore del taglio della cella octet-truss

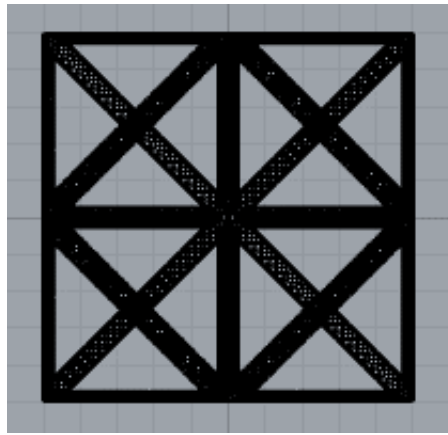


Figura 2.16. *Vista superiore della cella octet-truss tagliata*

Per la cella cubica a corpo centrato (Figura 2.17; 2.18):

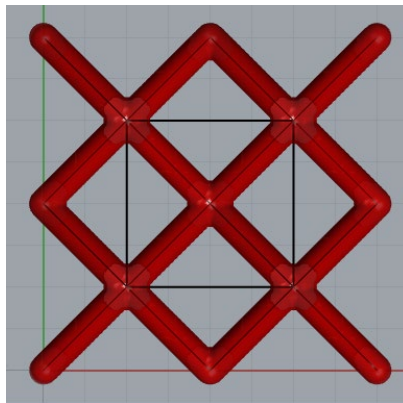


Figura 2.17. *Vista superiore del taglio della cella cubica a corpo centrato*

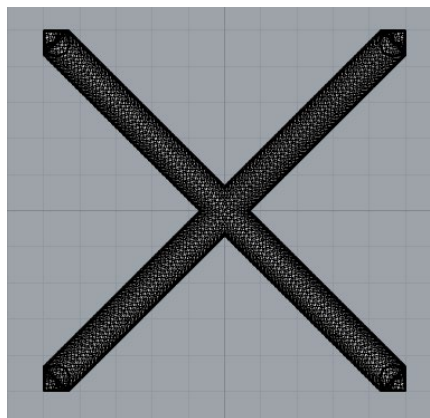


Figura 2.18. *Vista superiore della cella cubica a corpo centrato tagliata*

Selezionando *Analize*→*Mass Properties*→*Volume*, è possibile calcolare il valore del volume della struttura delimitata dal cubo.

Il dominio in cui si itera viene suddiviso in più valori discreti e si calcola la densità relativa corrispondente a ciascun valore del raggio selezionato. Per avere dati confrontabili per qualsiasi

lunghezza della cella unitaria, è preferibile calcolare tale valore rispetto al rapporto adimensionale tra raggio e lunghezza L della cella.

I risultati preliminari dell'andamento della funzione densità relativa, necessari per avere il controllo del gradiente nel metodo che si vuole implementare, vengono di seguito riportati, sia per la cella *octet-truss*, sia per la cella cubica a corpo centrato (Figura 2.19; Figura 2.20):

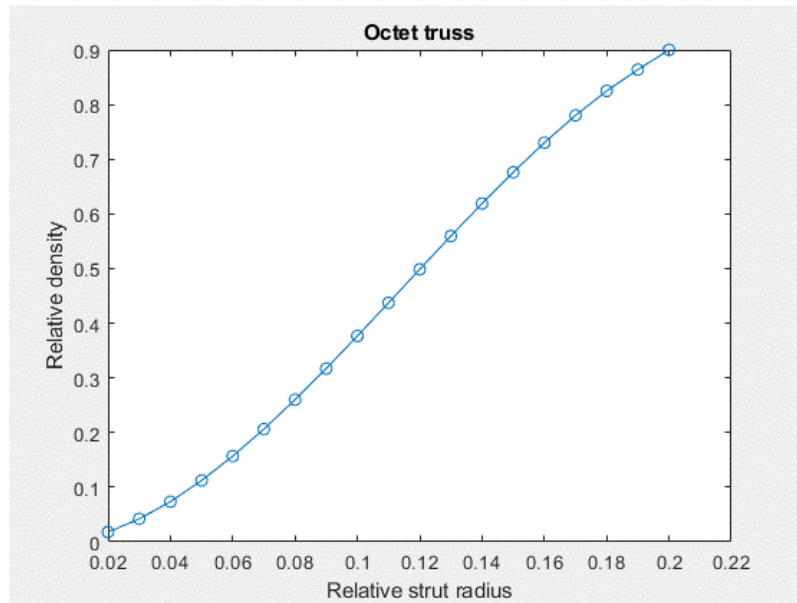


Figura 2.19. Andamento della funzione densità relativa di una cella *octet-truss*

L'equazione interpolante della densità di una cella *octet-truss* rispetto al rapporto tra raggio e lunghezza della cella si può esprimere tramite l'equazione:

$$\rho_{rel}^{Oct} = -149.85 \left(\frac{r}{L}\right)^3 + 52.037 \left(\frac{r}{L}\right)^2 + 0.098 \left(\frac{r}{L}\right) - 0.004$$

Si riportano i risultati ottenuti per diverse densità relative di celle *octet-truss* (Figura 2.21):

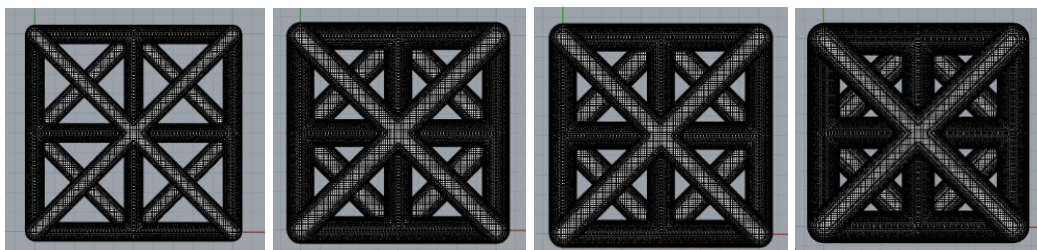


Figura 2.21. Densità cella *octet-truss*: a) 10%; b) 20%; c) 30%; d) 40% (vista frontale)

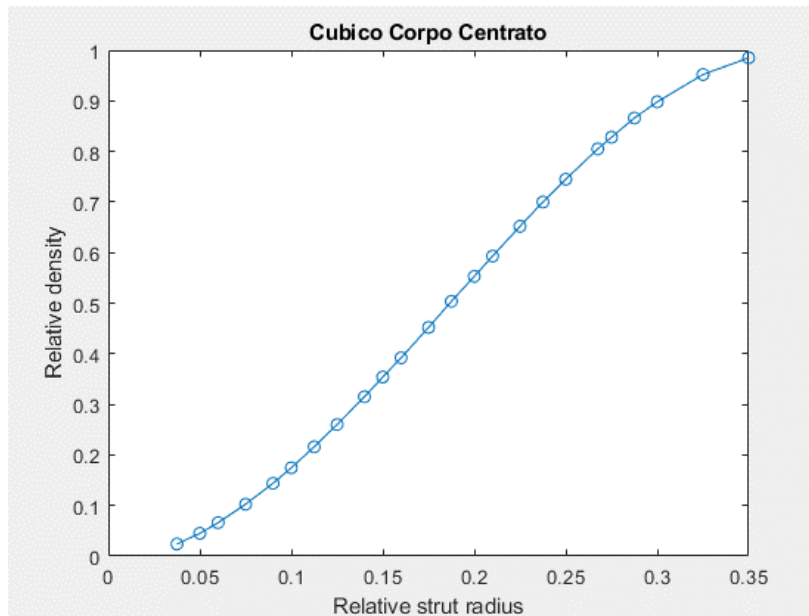


Figura 2.20. Andamento della funzione densità relativa di una cella cubica a corpo centrato

L'equazione interpolante della densità di una cella cubica a corpo centrato rispetto al rapporto tra raggio e lunghezza della cella si può esprimere tramite l'equazione:

$$\rho_{rel}^{BCC} = -38.93 \left(\frac{r}{L}\right)^3 + 21,67 \left(\frac{r}{L}\right)^2 + 0,012 \left(\frac{r}{L}\right) - 0,005$$

Si riportano i risultati ottenuti per alcune celle cubiche a corpo centrato (Figura 2.22):

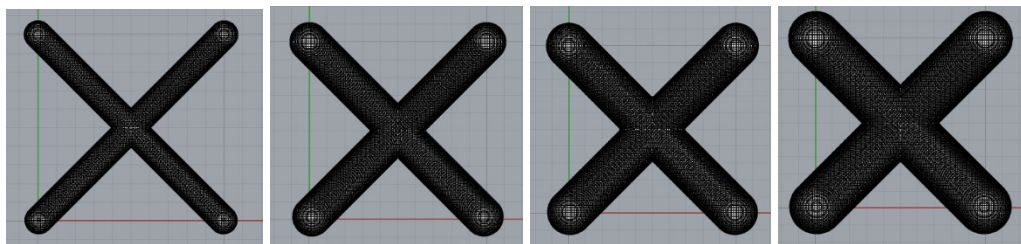


Figura 2.22. Densità cella CCC: a)10%; b) 20%; c) 30%; d) 40% (vista frontale)

Nella figura seguente vengono rappresentate le due funzioni sullo stesso grafico (Figura 2.23):

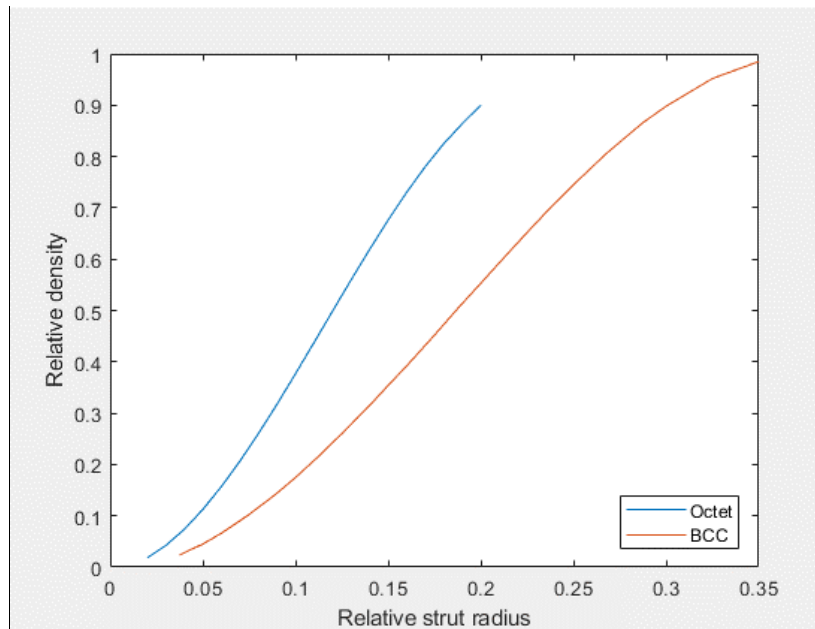


Figura 2.23. Confronto tra le densità relative di celle diverse

È possibile osservare che, per determinati intervalli, entrambe le strutture presentano un andamento lineare e che, oltre a un certo valore del raggio, non è più possibile aumentare la densità della struttura.

Ciò che si può notare è che la cella *octet-truss* ha un limite di stampabilità più basso rispetto alla cella cubica a corpo centrato. Poiché nella prima struttura vi sono più aste interconnesse, all'aumentare del raggio delle aste, la disconnessione avviene più facilmente e si raggiunge prima il limite di densità rispetto alla CCC che, avendo le aste derivanti da un unico centro, consente di far aumentare il raggio fino alla quasi scomparsa della cavità. Nello specifico, la densità massima raggiunta dalla cella *octet-truss* si attesta allo 0.90, mentre per la CCC allo 0.99.

Per quanto concerne le strutture TPMS, per calcolare la loro densità relativa si generano le isosuperfici della funzione F corrispondenti all'isovalore k , che è il parametro che viene fatto variare, in modo da trattare queste superfici come il confine tra materiale solido e vuoto. Lavorando con TPMS in fase *network*, è necessario chiudere l'isosuperficie.

L'obiettivo che ci si pone ora è esprimere la densità relativa ρ_{rel} in funzione dell'isovalore k . Per poter determinare la relazione tra ρ_{rel} e k , l'intervallo può essere discretizzato, e la densità relativa viene computata per ciascuna cella in corrispondenza di ogni valore di k .

Le TPMS scelte hanno dimensioni 20x20x20 mm, e la stessa risoluzione di 20 viene mantenuta su tutti e tre gli assi.

	L	X_{res}	Y_{res}	Z_{res}
Schwarz P	20	20	20	20
Schwarz D	20	20	20	20
Giroide	20	20	20	20

Tabella 2.3. Parametri scelti per le celle TPMS

L'algoritmo messo a punto per questo scopo è stato implementato in Grasshopper sfruttando il plug-in *Millipede* con i dati forniti nella Tabella 2.3 (Figura 2.2): consiste nella generazione di array di punti a cui associare il valore dell'espressione scritta nel componente *Math*. Il *building block* "Iso surface" del plug-in *Millipede*, utilizza l'algoritmo *marching cubes* per estrarre isosuperfici da un campo di valori di densità (Figura 2.25). Questi valori devono essere definiti su una griglia tridimensionale e possono essere generati tramite la semplice funzione $F(x, y, z)$.

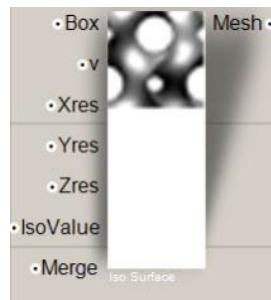


Figura 2.25. componente del plug-in *Millipede* per creare isosuperfici *Marching Cubes*

In ingresso richiede:

- *Box* [Bounding Box]: la bounding box della geometria;
- *v* [Lista di numeri]: questo elenco deve contenere i numeri X_{res} x Y_{res} x Z_{res} che designano i valori del campo in ciascun punto della griglia all'interno di *Box*. I valori dovrebbero essere ordinati in modo tale che la direzione *x* venga attraversata prima, poi *y* e poi gli strati *z*;
- X_{res} , Y_{res} , Z_{res} [Intero]: la risoluzione della griglia tridimensionale;
- *IsoValue* [Numero]: la soglia alla quale tagliare un'isosuperficie attraverso il campo di densità;
- *Merge* [Boolean]: se vero, la *mesh* risultante avrà i suoi vertici coincidenti fusi e apparirà più uniforme (non altera le posizioni dei vertici, fonde solo i vertici coincidenti).

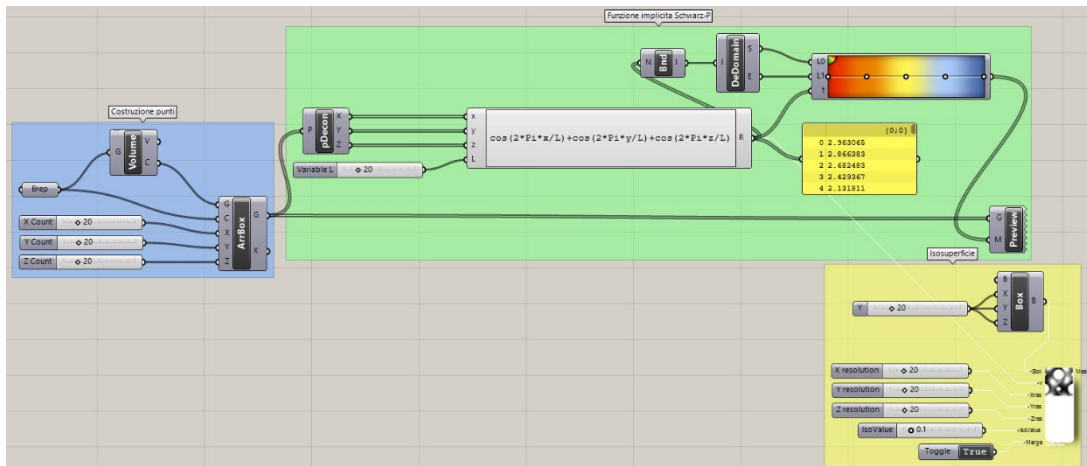


Figura 2.24. Algoritmo per la creazione delle isosuperfici delle TPMS

Per una cella Schwarz P, lo spazio chiuso considerato è rappresentato dalla disequazione

$$F(x, y, z) \leq k, \quad -3 \leq k \leq 3$$

con l'isovalore k compreso nell'intervallo $[-3, 3]$.

Nel caso delle celle in fase matrice, è necessario generare due isosuperfici corrispondenti agli isovalori k e $-k$. Lo spazio racchiuso da queste due isosuperfici è rappresentato dalla disequazione:

$$-k \leq F(x, y, z) \leq k, \quad 0 \leq k \leq 3$$

Con questa tipologia si considerano solo gli isovalori k non negativi, cioè $k \in [0, 3]$. A seguito di varie osservazioni, si può notare che una cella primitiva di Schwarz in fase matrice può essere costruita come differenza booleana tra due celle in fase solida.

Nella Figura 2.26. si mostrano alcuni dei risultati ottenuti facendo variare l'isovalore (Figura 1.1):

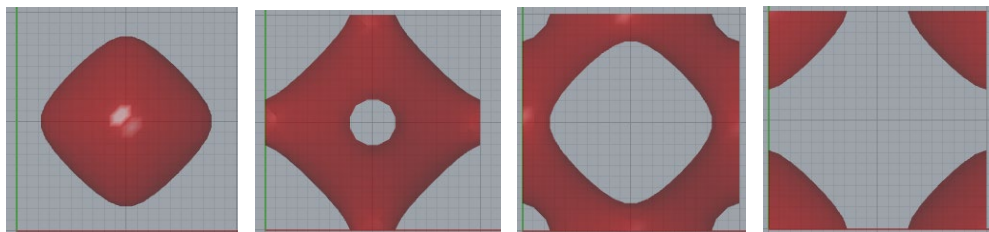


Figura 2.26. Vista superiore di celle P-Schwarz ottenute con diversi isovalori: a) -1.3; b) -0.8; c) 0.6; d) 1.3

Si può osservare che, per ogni $k > 0$, esiste una regione della cella che consente la connettività con le celle vicine, e che quando $k > 1$ nella cella appare una cavità interna. Da notare, inoltre, che il parametro k è indipendente dalla lunghezza della cella unitaria.

Consideriamo la cella in fase *network*. Per poter calcolare il volume della struttura appena ottenuta, si effettua un'operazione di *bake* della geometria in Rhinoceros, per poter creare la fase solida attraverso la creazione di una *Mesh Box*. La *mesh* creata è poligonale, con opzione *Smooth and Slower*, che offre una migliore risoluzione a scapito di tempi di *meshing* più lunghi. Ogniqualevolta la *mesh* si presenti aperta, si mettono a punto diverse strategie per chiudere eventuali facce mancanti o *naked-edges*. Nella finestra pop-up del controllo della *mesh* si seleziona *Naked Edges*: qualsiasi bordo di superficie che non sia unito ad un altro verrà contrassegnato. Questo significa che il modello non è una polisuperficie chiusa solida e di conseguenza non può essere esportato e utilizzato nelle macchine di prototipazione rapida.

Per correggere i *naked edges* (Figura 2.27):

- Si seleziona il modello e lo si uniscono tutti gli *edges* tramite il comando *Join*;
- Col comando *MatchMeshEdges* si spostano i bordi della faccia di una *mesh* aperta sino a farli combaciare con i bordi della faccia adiacente. In primo luogo, il comando cerca di far corrispondere i vertici, quindi prova a suddividere i bordi per farli combaciare. Nessuna parte della *mesh* viene spostata oltre la tolleranza specificata. È possibile usare questo comando sia su tutta la *mesh* che solo sui bordi selezionati;
- Col comando *DeleteMeshFaces* è possibile rimuovere facce problematiche della *mesh* e ricostruirle con *FillMeshHole*;
- Si ripetono questi passaggi fino a quando tutti i *naked edges* non sono stati riparati.

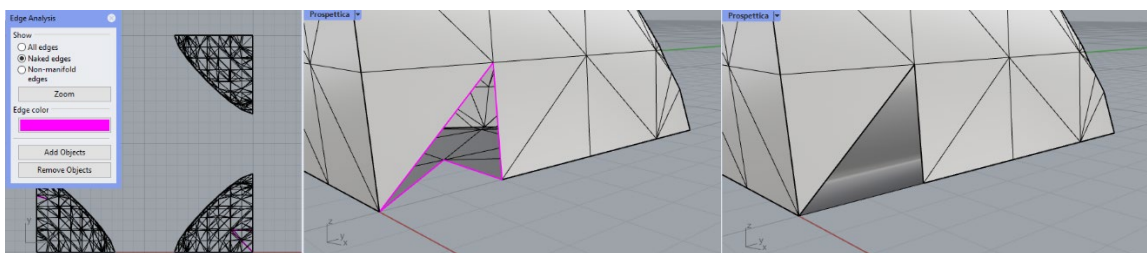


Figura 2.27. Riparazione della *mesh*: prima della riparazione, i bordi magenta sono aperti

Si controllano le normali (Figura 2.28):

- Si clicca sulla scheda *Visualizza*;
- Sotto le impostazioni dell'oggetto, si clicca sulla casella di controllo *Color Backfaces*;
- Qualsiasi parte del modello che ha il colore cambiato è al rovescio, ovvero le normali della superficie puntano verso l'interno del modello. Per stampare correttamente il modello le normali devono indicare l'esterno del modello;

- Per cambiare la direzione di qualsiasi superficie che ha normali che puntano verso l'interno del modello le si seleziona e, col comando *Flip*, si inverte il verso (Figura 1.1).

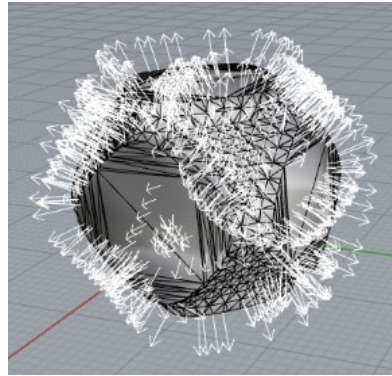


Figura 2.28. Direzione corretta delle normali: puntano dall'interno verso l'esterno

Il volume si calcola come nel caso precedente dal menu *Analyze*→*Mass Properties*→*Volume*. Si misura la densità relativa ρ_{rel} di ciascuna cella corrispondente agli isovalori selezionati.

I risultati sono mostrati nella Figura 2.29, con l'aggiunta di $\rho_{rel} = 0, k = -3$ e $\rho_{rel} = 1, k = 3$:

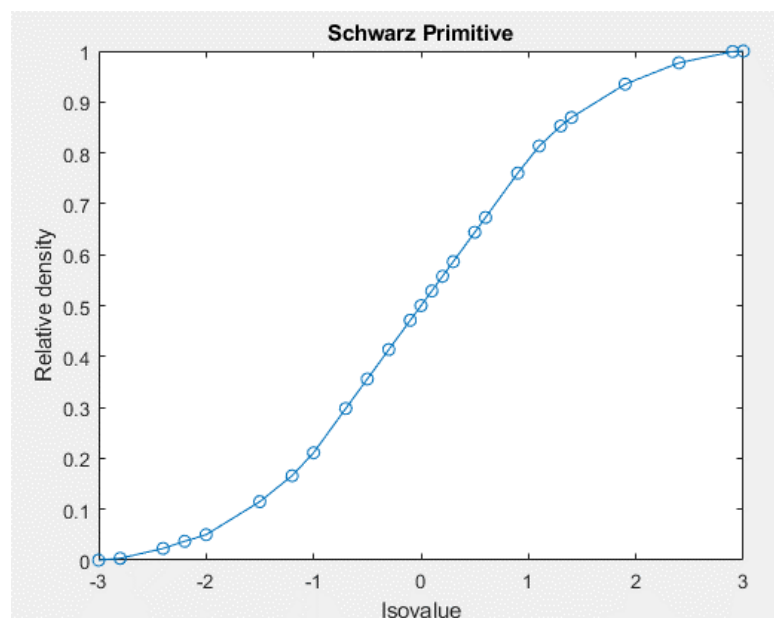


Figura 2.29. Fitting curve di densità relativa rispetto all'isovalore per la Schwarz Primitive in fase network

Analizzando il risultato ottenuto, si può notare che maggiore è il valore di k , maggiore è la densità relativa. Si può inoltre osservare che:

1. Esiste un comportamento lineare per $k \in [-1, 1]$ e un comportamento non lineare al di fuori di questo intervallo,
2. Per $\rho_{rel} = 0.5, k = 0$;
3. Il grafico è simmetrico rispetto al punto $(\rho_{rel} = 0.5, k = 0)$.

L'equazione interpolante è $\rho_{rel}^{SchwarzP} = -0.014k^3 + 0.287k + 0.501$, mentre l'intervallo lineare si può approssimare con $\rho_{rel}^{SchwarzP} = 0.288k + 0.5$.

Alcuni risultati ottenuti al variare di k (Figura 2.30):

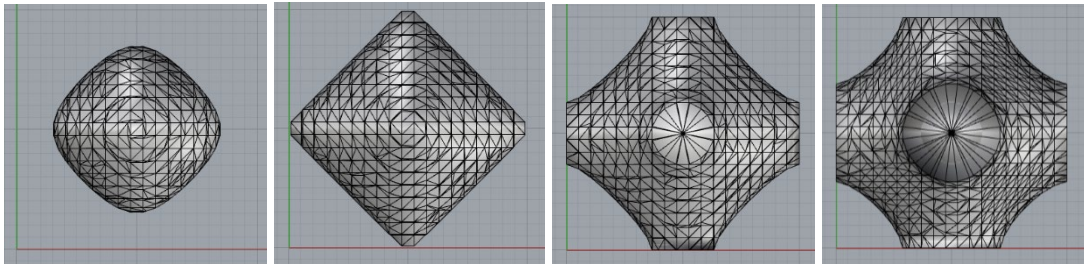


Figura 2.30. Densità cella Schwarz P: a) 10%; b) 20%; c) 30%; d) 40% (vista frontale)

Ora si analizzano le celle Schwarz Diamond. L'isovalore k è compreso nell'intervallo $[-1, 1]$.

$$F(x, y, z) \leq k, -1 \leq k \leq 1$$

I risultati sono mostrati nella Figura 2.31: (Figura 1.1):

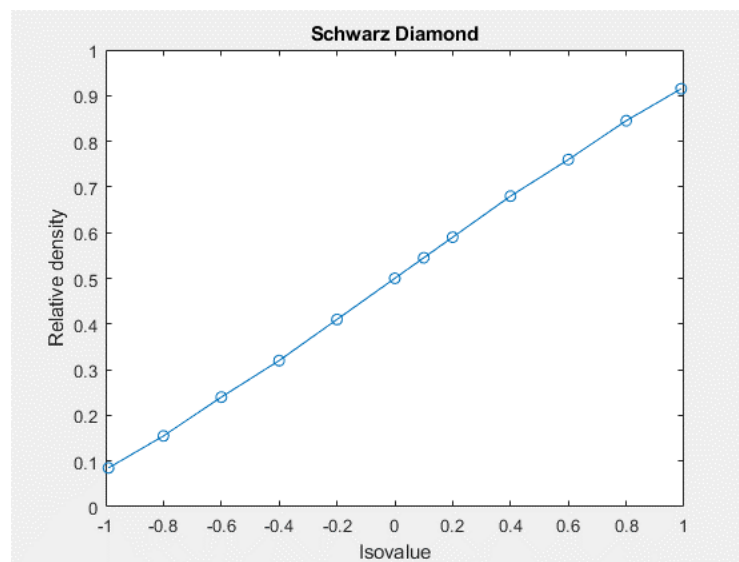


Figura 2.31. Fitting curve di densità relativa rispetto all'isovalore per la Schwarz Diamond in fase network

Anche per questa struttura si può osservare che:

1. Esiste un comportamento lineare per $k \in [-0.8, 0.8]$ e un comportamento non lineare al di fuori di questo intervallo,
2. Per $\rho_{rel} = 0.5, k = 0$;
3. Il grafico è simmetrico rispetto al punto $(\rho_{rel} = 0.5, k = 0)$.

La funzione interpolante si può esprimere tramite l'espressione

$$\rho_{rel}^{SchwarzD} = -0.0005k^2 + 0.4279k + 0.5004.$$

Alcuni risultati ottenuti al variare di k (Figura 2.32):

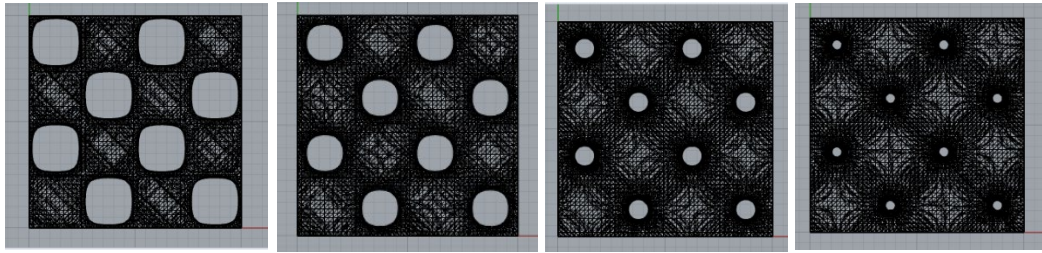


Figura 2.32. Densità cella Schwarz D : a) 10%; b) 20%; c) 30%; d) 40% (vista frontale)

L'ultima TPMS rimasta da analizzare è il giroide. Si procede come nei casi precedenti.

Si plottano i risultati ottenuti, approssimati dall'equazione $\rho_{rel}^{Giroide} = 0,337k + 0,5$ (Figura 2.33):

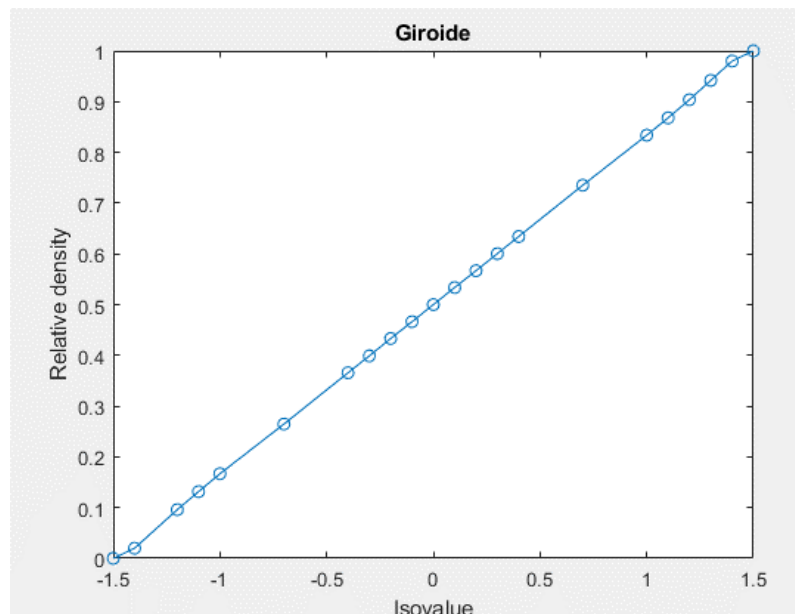


Figura 2.33. Fitting curve di densità relativa rispetto all'isovalore per il giroide in fase network

Si osserva che:

1. Per $k = 0$, la superficie minima del giroide divide lo spazio in due regioni elicoidali, ognuna delle quali rivendica una frazione di volume del 50%;
2. All'aumentare di k , anche le proporzioni delle due regioni aumentano o diminuiscono in modo monotono. Quando k è positivo, la frazione di volume aumenta nella direzione positiva. Quando k è negativo, la frazione di volume diminuisce nella direzione negativa;
3. Per $|k| = 1.41$, viene generata la cosiddetta superficie "pinch-off" e la superficie sarà disconnessa. Per una densità più bassa, una tale struttura perde le sue proprietà

meccaniche e per una densità più alta, l'area chiusa risultante impedisce la rimozione di polvere o liquido per la tecnica AM;

4. Per $1.41 < |k| < 1.5$, la superficie giroide non è più connessa;
5. Quando $k > 1.5$, la superficie scompare.

Si riportano alcuni risultati al variare dell'isovalore (Figura 2.34)

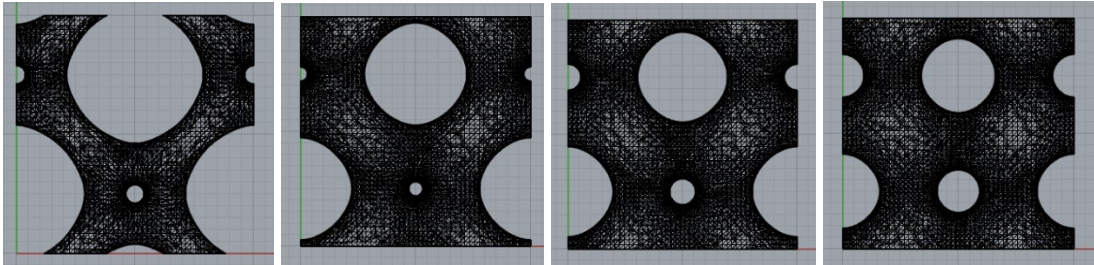


Figura 2.34. Densità cella giroide: a) 10%; b) 20%; c) 30%; d) 40% (vista frontale)

Le densità relative delle tre TPMS analizzate vengono riportate in un unico grafico, utile per un confronto rapido (Figura 2.35):

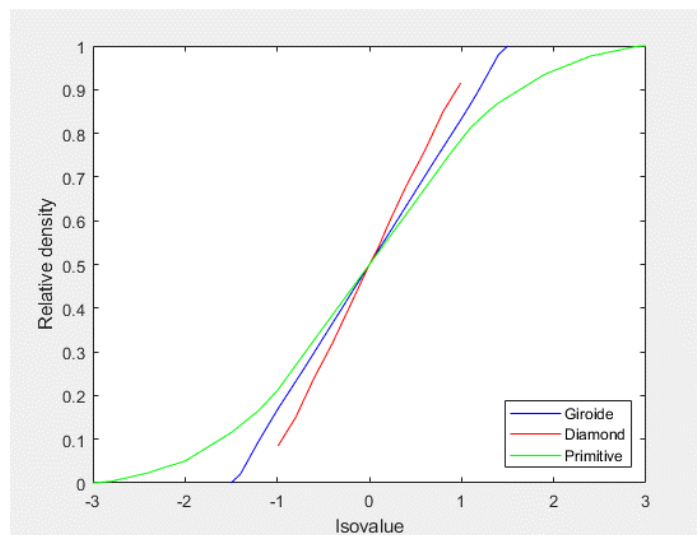


Figura 2.35. Confronto tra densità relative delle diverse TPMS

Dal momento che *Dendro* sarà uno dei plug-in più utilizzati per creare volumi negli algoritmi di modellazione che seguono, si riporta il metodo utilizzato per calcolare la densità relativa delle TPMS a partire dalla decostruzione dell'isosuperficie, dalla selezione dei punti che la costituiscono e dalla creazione del volume dall'assegnazione del loro raggio.

I due metodi non coincidono: se con il metodo appena presentato si lavora con funzioni e si modifica l'isovalore, con *Dendro* si lavora con dati volumetrici modificabili attraverso il raggio.

Nella figura sottostante si può notare la differenza esistente tra la *mesh* creata a partire dalla definizione della funzione Schwarz-P, con la generazione di due superfici di isovalore k e $-k$, in questo esempio pari a 0.3, e la *mesh* ottenuta a partire dall'isosuperficie di isovalore zero ottenuta da funzioni di distanza, dalla sua decostruzione in punti e dall'assegnazione di un volume con Dendro di raggio pari a 0.3 (Figura 2.36).

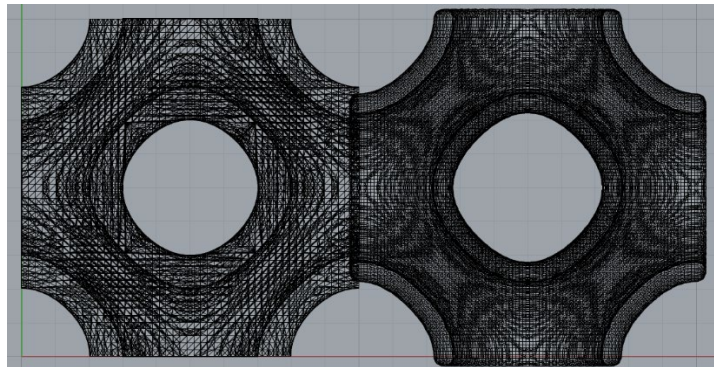


Figura 2.36. Mesh a confronto ottenute da algoritmi che lavorano con parametri differenti: a) isovalore; b) raggio delle sfere

Nella tabella 2.4 vengono riportati i parametri utilizzati nel nuovo algoritmo (Figura 2.37). I valori sono da considerarsi tutti in mm. Si generano le TPMS attraverso il comando dedicato di *Axolotl*, la *mesh* ottenuta si decostruisce nei suoi punti costituenti, che si trasformano in sfere di volume assegnando loro un raggio tramite il comando di *Dendro* “Points To Volume”.

	L	r_{min}	<i>voxel size</i>
Schwarz P	10	0.2	0.1
Schwarz D	10	0.2	0.1
Giroide	10	0.2	0.1

Tabella 2.4. Parametri scelti per le celle TPMS da utilizzare nell'algoritmo con *Dendro*

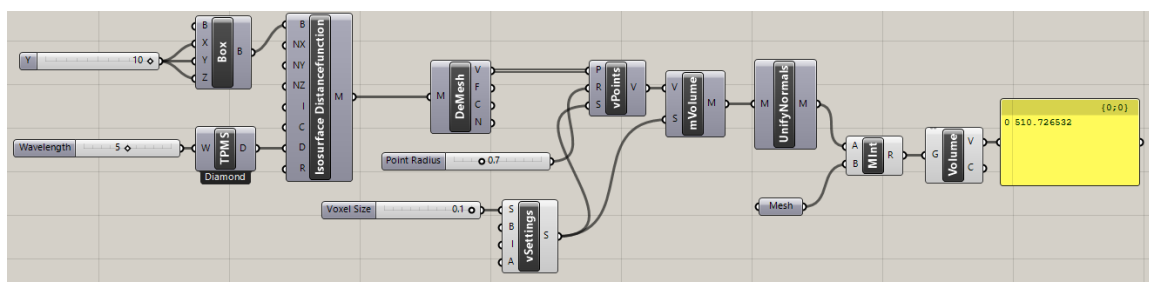


Figura 2.37. Algoritmo di modellazione di TPMS con *Dendro*

Nel calcolo del volume, si procede a tagliare la cella col cubo che delimita la struttura, di volume L^3 . Il procedimento può essere eseguito direttamente in Grasshopper attraverso il componente “Mesh Intersection” (Figura 2.38).

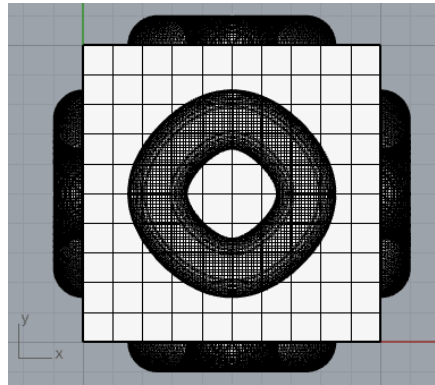


Figura 2.38. Taglio della cella Schwarz P

Le densità relative ottenute vengono mappate in un grafico, le cui ascisse sono rappresentate dalla lunghezza del raggio dei punti rispetto alla lunghezza della cella (Figura 2.39).

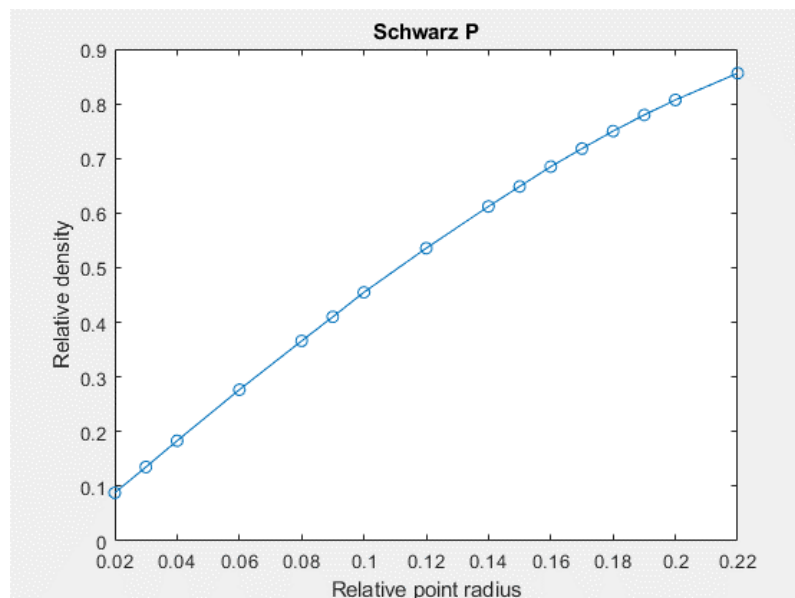


Figura 2.39. Densità relativa di una Schwarz P rispetto al rapporto tra raggio e lunghezza della cella

L’equazione che approssima l’andamento è:

$$\rho_{rel}^{SchwarzP} = -6.647 \left(\frac{r}{L}\right)^2 + 5.477 \left(\frac{r}{L}\right) - 0.0246.$$

Si riportano alcuni risultati al variare del raggio (Figura 2.40)

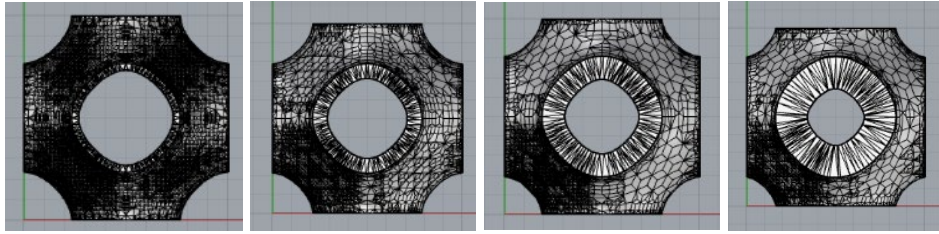


Figura 2.40. Densità cella Schwarz P: a) 10%; b) 20%; c) 30%; d) 40% (vista frontale)

Si ripete lo stesso identico procedimento per il giroide. La cella viene tagliata con lo stesso cubo (Figura 2.41):

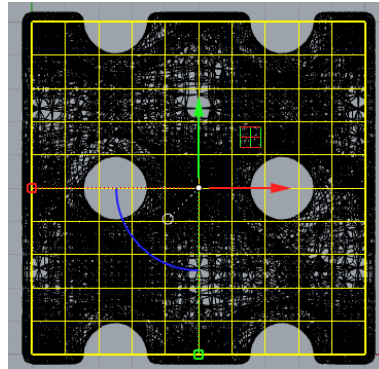


Figura 2.41. Taglio della cella giroide

Si plottano i risultati ottenuti (Figura 2.42):

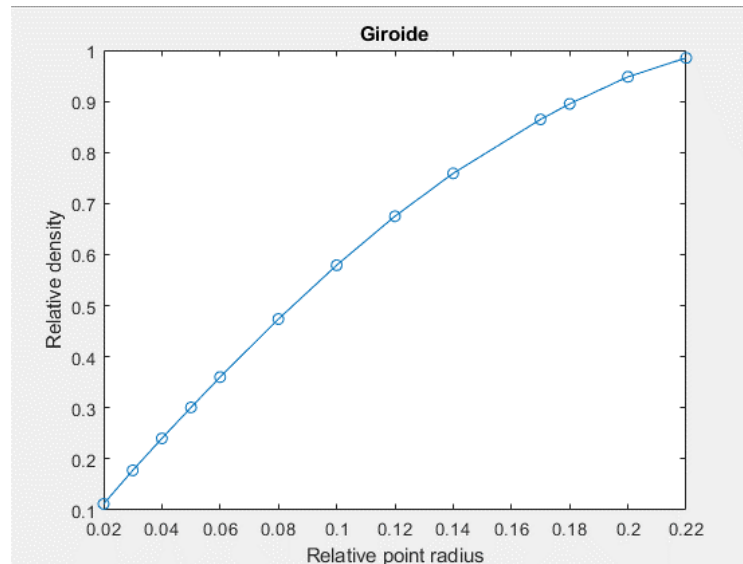


Figura 2.42. Densità relativa di un giroide rispetto al rapporto tra raggio e lunghezza della cella

L'equazione che approssima l'andamento della densità relativa del giroide è:

$$\rho_{rel}^{Giroide} = -12,472 \left(\frac{r}{L}\right)^2 + 7,4019 \left(\frac{r}{L}\right) - 0,0353.$$

Alcuni risultati al variare del raggio (Figura 2.43)

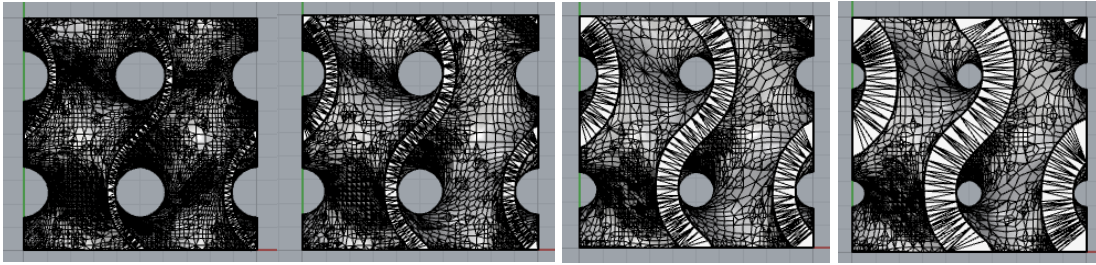


Figura 2.43. Densità cella giroide: a) 10%; b) 20%; c) 30%; d) 40% (vista frontale)

Stesso procedimento anche per la *Schwarz Diamond*. La cella viene tagliata col cubo e si calcola il volume della struttura ottenuta (Figura 2.44):

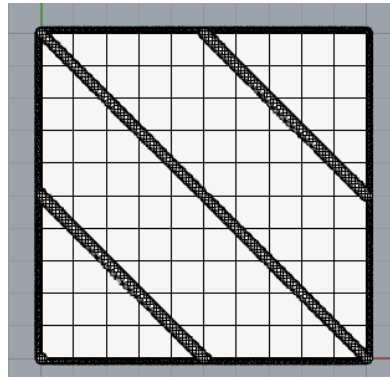


Figura 2.44. Taglio della cella Schwarz D

La densità relativa della *Schwarz Diamond* viene di seguito mostrata (Figura 2.45):

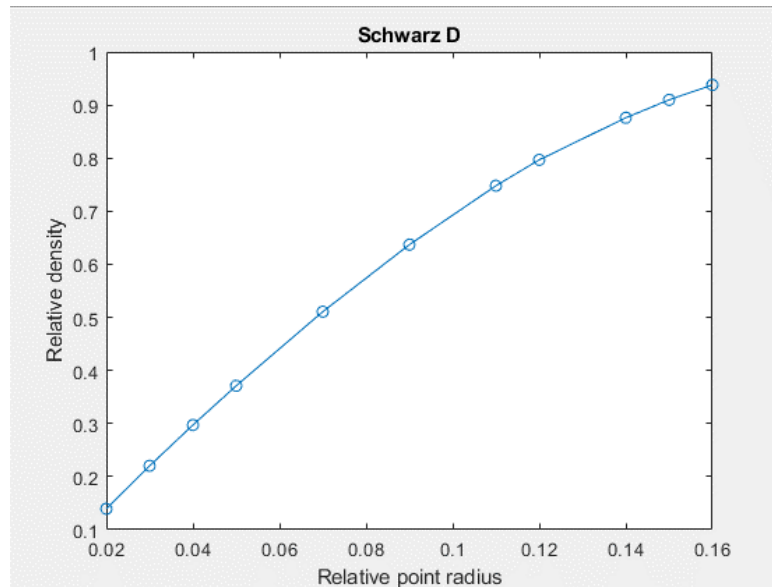


Figura 2.45. Densità relativa di una Schwarz D rispetto al rapporto tra raggio e lunghezza della cella

L'equazione risultante è $\rho_{rel}^{SchwarzD} = -20.252 \left(\frac{r}{L}\right)^2 + 9.402 \left(\frac{r}{L}\right) - 0.0445$.

Al variare del raggio si ottengono le seguenti strutture (Figura 2.46):

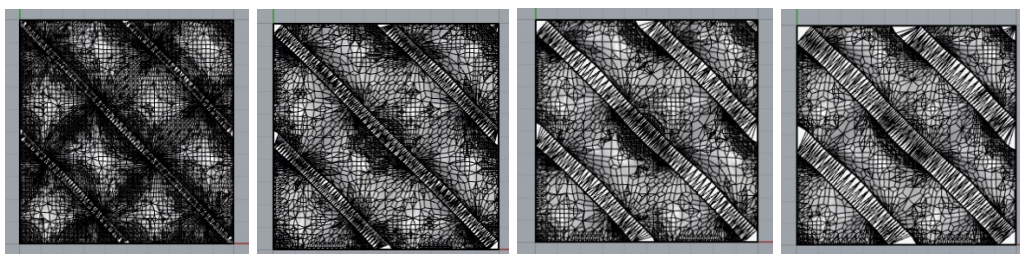


Figura 2.46. Densità cella Schwarz D: a) 10%; b) 20%; c) 30%; d) 40% (vista frontale)

Vengono plottati i risultati per tutte le TPMS analizzate (Figura 2.47):

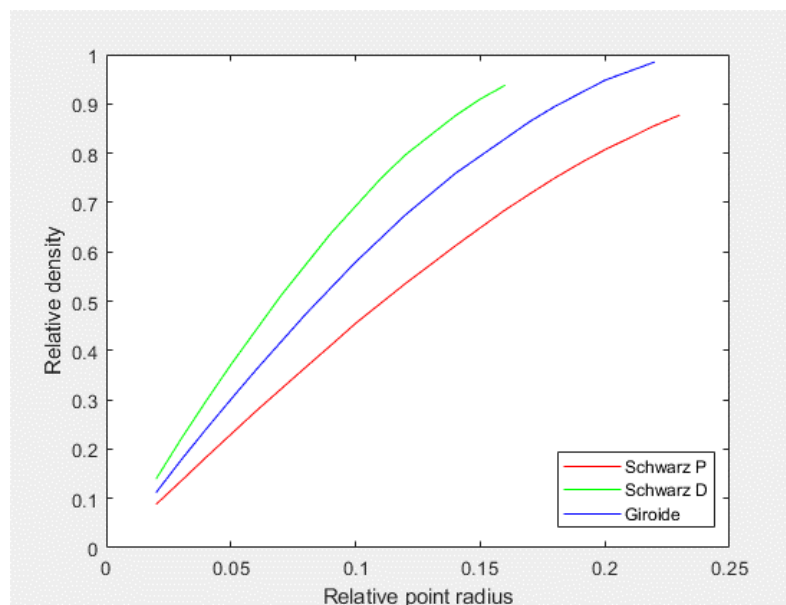


Figura 2.47. Confronto tra densità relative delle diverse TPMS rispetto al rapporto raggio su lunghezza della cella

Identificati i valori con cui determinare le densità relative, si procede mostrando diversi esempi di FGLSs create applicando i metodi per la generazione del gradiente presentati in questo capitolo, messi a punto in Grasshopper sfruttando i *plug-in* selezionati per la modellazione volumetrica.

- Strutture *truss-based* con distanza da un'entità geometrica

Dopo aver identificato la cella, che può essere costruita all'interno di Rhinoceros attraverso più linee e poi importata in Grasshopper come *Multiple Curves*, e scelto il numero di ripetizioni delle celle tramite la costruzione di un *Array*, che può essere modificato in tempo reale tramite lo *slider*, si procede a generare lo spessore della *lattice*.

Volendo generare il gradiente tramite i valori della distanza da un'entità geometrica, è necessario stabilire quali sono i punti da cui calcolare la distanza. Si possono utilizzare per esempio i punti medi delle curve contenute nell'*Array Box*, ma è possibile anche suddividere le

curve in segmenti di uguale lunghezza tramite il componente “Divide Curve”, che in output fornisce tutti i punti di divisione, aumentandoli o diminuendoli tramite il parametro *Count*. Definiti i punti, si deve scegliere l’entità geometrica, che funge da elemento attrattore.

Gli attrattori sono elementi che agiscono come magneti virtuali, attraendo o respingendo altri oggetti. In Grasshopper, qualsiasi geometria referenziata da Rhino o creata all'interno di Grasshopper può essere utilizzata come attrattore. Gli attrattori possono influenzare un numero qualsiasi di parametri degli oggetti circostanti tra cui scala, rotazione, posizione. Questi parametri vengono modificati in base alla loro relazione con la geometria dell'attrattore.

Nei modelli che verranno mostrati si scelgono come attrattori un piano e un punto (Figura 2.48; 2.49).

Le distanze ottenute vengono ricalcolate in un nuovo dominio attraverso il comando “Remap”. Questo componente prende in input i valori da rimappare in un nuovo dominio numerico, il dominio di origine, ottenibile attraverso il componente *Bounds*, e il dominio di destinazione, che bisogna costruire con *Construct Domain*, selezionando gli estremi scelti. Nella scelta degli estremi del dominio, si fa riferimento ai corrispondenti grafici sulla densità relativa, in modo da avere una porosità controllata. Le nuove distanze vengono assegnate come raggio, input del comando di *Dendro* “Points To Volume” che, come visto, converte i punti ricavati dalle curve in un volume.

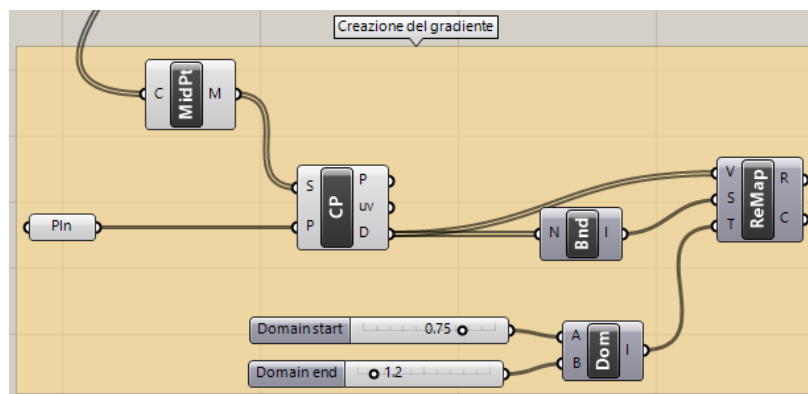


Figura 2.48. Creazione del gradiente tramite distanza da un piano

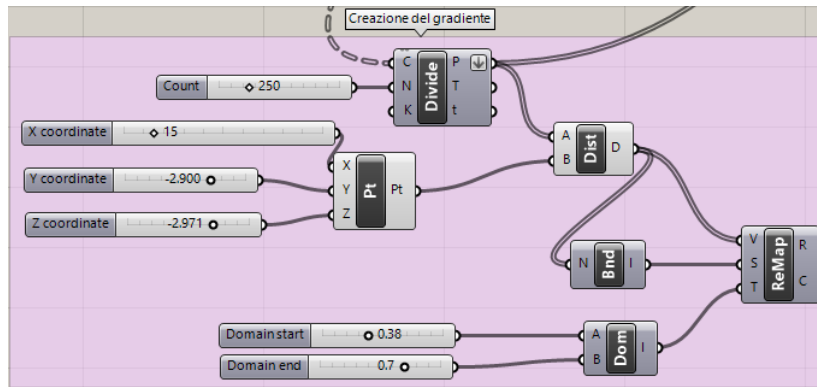


Figura 2.49. Creazione del gradiente tramite distanza da un punto attrattore

Da notare che all’output del componente “Divide Curve” si applica un *Flatten*, così da avere tutti i dati in un’unica lista.

- TPMS a gradiente con valutazione di un’espressione matematica

Un altro metodo per generare un gradiente consiste nella valutazione di un’espressione matematica.

Dopo aver creato un’isosuperficie di isovalore assegnato, ottenibile utilizzando il componente di *Axolotl* “Isosurface Distancefunction” dando in input una “Bounding Box”, costruita parametricamente, e l’oggetto distanza costituito dalla TPMS selezionabile dalle *lattice* disponibili di *Axolotl* di *wavelength* assegnata, ovvero distanza dopo cui il pattern si ripete periodicamente, si procede con la generazione del gradiente (Figura 2.50).

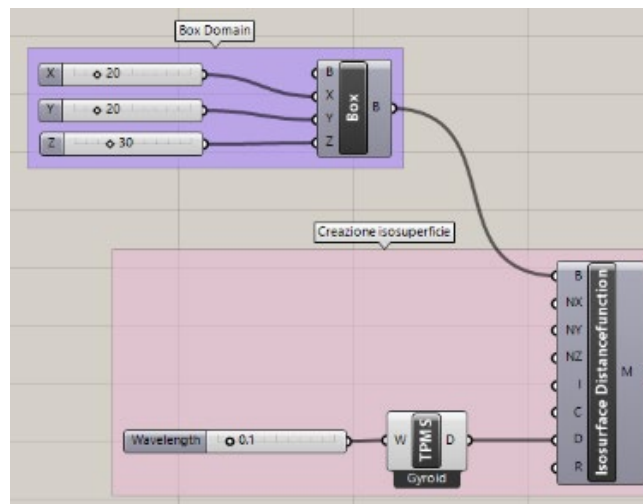


Figura 2.50. Creazione dell’isosuperficie con *Axolotl*

Ipotizzando di voler produrre un gradiente verticale lungo l’asse z, si decostruisce l’isosuperficie con “Deconstruct Mesh”, si prendono i valori corrispondenti alla terza componente e li si inseriscono in “Expression”, presente nella sezione *Script* di *Maths*, che permette di valutare un’espressione scritta nel suo pannello. In input bisogna inserire i valori

degli estremi del dominio in z e di valori dei raggi massimi e minimi, oltre a tutte le componenti z. I valori di output diventano i raggi del comando di *Dendro* “Points To Volume” (Figura 2.51).

L’equazione scelta è lineare e in input vengono assegnati i valori massimi e minimi dello spessore che si intende assegnare, per poter ottenere una densità graduata.

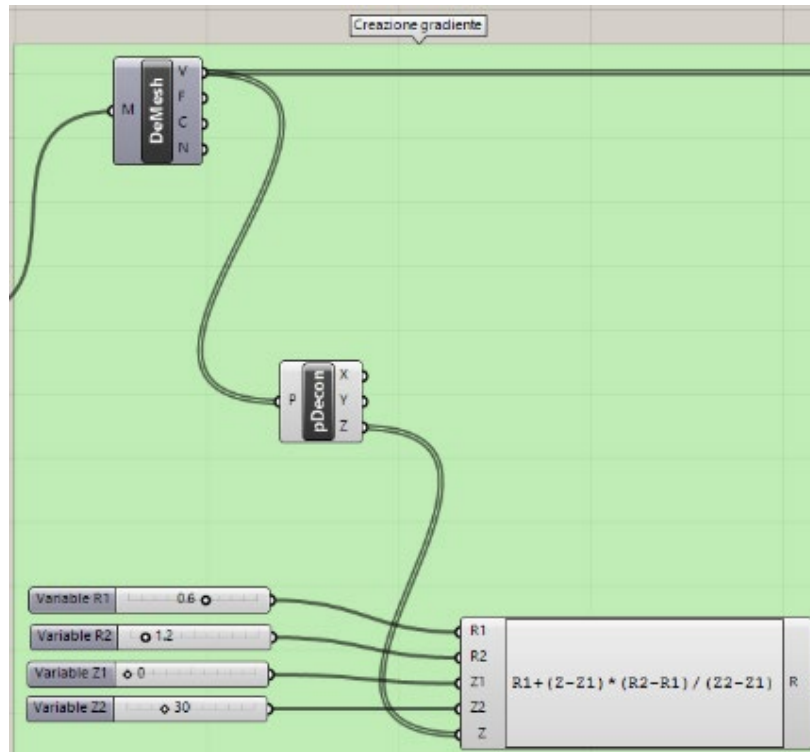


Figura 2.51. Algoritmo di modellazione del giroide a gradiente lungo l’asse z

- Strutture con gradiente dal *building block Graph Mapper*

Il componente di Grasshopper *Graph Mapper* funziona in questo modo: qualunque valore si immette, verrà letto sull’asse x (il lato inferiore del componente) e mappato utilizzando la curva sull’asse y (il lato sinistro del componente). L’intervallo rappresentato dai lati è impostabile facendo doppio clic sul componente. Anche il tipo di curva è modificabile (Figura 2.52).

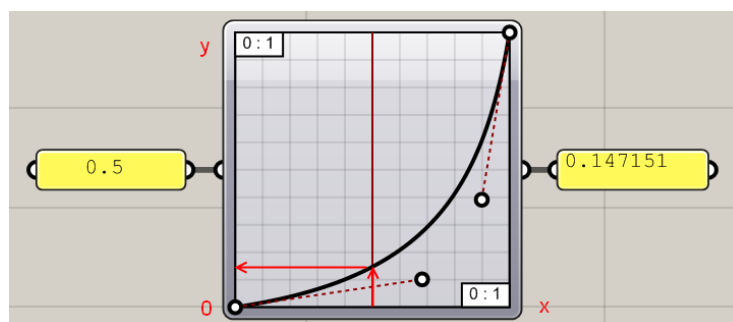


Figura 2.52. Esempio di utilizzo del componente *Graph Mapper*

Di default i valori settati all'interno sono compresi, per entrambi gli assi, tra 0 e 1. È preferibile mantenerli tali, e utilizzare un *Remap* per i valori in input e in output, in modo da poter facilmente modificare in tempo reale la curva di partenza senza dover impostare i parametri all'interno del componente. Per avere controllo sui valori della porosità, basta impostare gli estremi del dominio output di *Graph Mapper* tenendo presente i grafici della densità relativa (Figura 2.53).

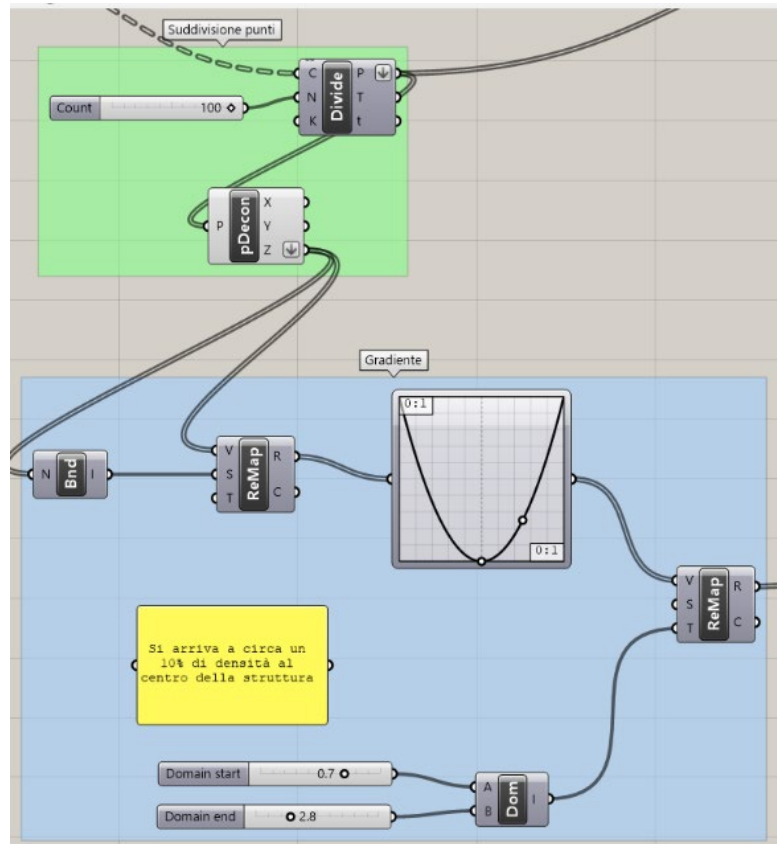


Figura 2.53. Gradiente decrescente e crescente ottenuto con Graph Mapper

- Strutture con densità data da immagine bitmap

Le bitmap sono "immagine digitali" costituite da un insieme di dati che specificano il colore dei singoli punti dello schermo. A ciascun pixel corrispondono, di fatto, un valore cromatico e due coordinate spaziali. Le sorgenti bitmap sono il modo più primitivo per definire un modello di immagine per modificare il campo voxel. Sono caratterizzate da due proprietà:

1. Risoluzione, determinata dal numero di pixel contenuti nell'unità di misura considerata;
2. Profondità, definita dalla memoria che si dedica ad ogni pixel, ovvero dal numero di bit dedicati ad ogni pixel per descrivere il colore. Maggiore è il numero di bit, maggiore è il numero di colori che è possibile descrivere

In sostanza, *Bitmap Source*, componente di *Monolith*, definisce semplicemente il percorso dell'immagine e il canale da campionare. Il componente *Add* o *Multiply Sources* itererà attraverso il campo voxel e campionerà l'immagine per il canale specificato e la aggiungerà/moltiplicherà con il valore di base. Il modo in cui una sorgente bitmap è orientata/mappata in tre dimensioni può essere modificato utilizzando uno degli altri componenti basati su immagine (ad esempio *Image Loft* o *Image Blend*). Il componente *Image Loft* crea un'interpolazione tra le immagini impilate lungo l'asse z. È possibile aggiungere tutte le origini bitmap all'input dell'immagine che le interpolerà tra di loro utilizzando i valori normalizzati dell'asse z (ovvero 0-1). Si noti che l'ordine dell'elenco in cui si aggiungono le bitmap e le posizioni dell'asse z è importante.

4. Generazione del modello superficie

La generazione della *mesh* è l'ultimo passaggio per portare a termine il metodo implementato.

Molti algoritmi utilizzati nella modellazione volumetrica, per esempio l'algoritmo di *marching cubes* (Lorenson e Cline, 1987) per la generazione di *mesh* isosuperficiali, traggono la loro origine nell'imaging medico. Con le scansioni CT e MRI, questo dominio fornisce dati come una pila di immagini 2D che insieme formano un volume di voxel. *Marching cubes* è un semplice algoritmo per creare una *mesh* triangolare da una funzione implicita. L'input dell'algoritmo è una voxelizzazione dello spazio che rappresenta un campo scalare $v = f(x, y, z)$, che può essere binario o no (Figura 2.54). Se i dati in input non sono binari, l'algoritmo richiede un parametro aggiuntivo (valore di *threshold* o *isovalue* per classificare i campioni interni o esterni dalla superficie. Se il modello in input è binario, la superficie separa semplicemente i punti interni dai punti esterni, se non è binario, l'isosuperficie dovrà unire tutti i punti che hanno lo stesso isovalore scelto.



Figura 2.54. Schematizzazione del processo di creazione della mesh con l'algoritmo *marching cubes*

Funziona iterando ("marciando") su una griglia uniforme di cubi sovrapposti a una regione della funzione (Figura 2.55).

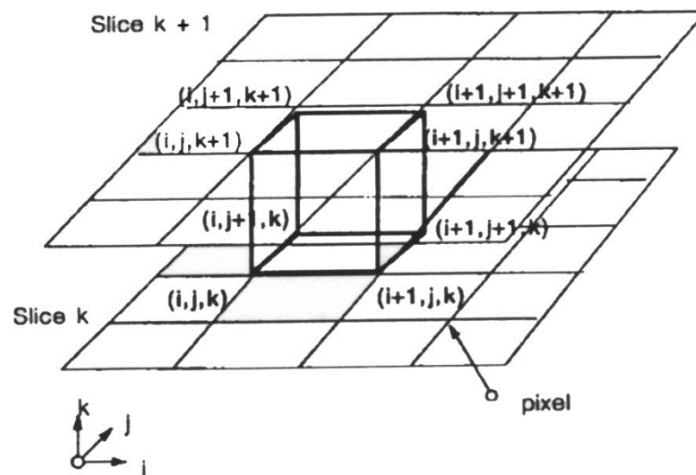


Figura 2.55. Inizializzazione dell'algoritmo marching cubes

Se tutti gli 8 vertici del cubo sono positivi, o tutti gli 8 vertici sono negativi, il cubo è interamente sopra o interamente sotto la superficie e non vengono emessi triangoli. Altrimenti, il cubo si trova a cavallo della funzione e vengono generati alcuni triangoli e vertici. Poiché ogni vertice può essere positivo o negativo, ci sono tecnicamente 2^8 possibili configurazioni, ma molte di queste sono equivalenti l'una all'altra, quindi ci sono solo 15 casi unici (Figura 2.56).

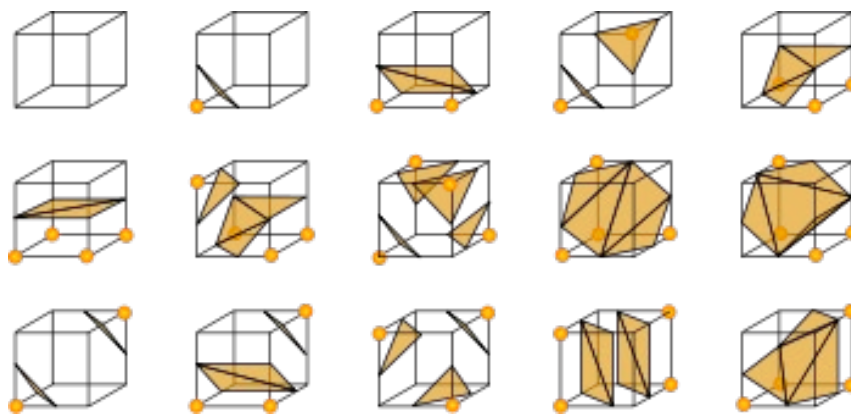


Figura 2.56. Configurazioni possibili dell'intersezione tra i cubi e la superficie

Per ciascun cubo che viene processato, bisogna generare la geometria, ovvero i vertici dell'isosuperficie, e la topologia, quindi i triangoli che connettono questi vertici. Assumendo che il campo sia continuo, i bordi che hanno mostrato un cambiamento di segno devono essere intersecati dalla superficie. L'algoritmo crea un vertice per ciascun bordo della griglia che mostra un cambiamento del segno e l'esatta posizione del vertice lungo il bordo è computata attraverso l'interpolazione lineare.

Si segue l'iterazione su tutti i cubi, aggiungendo triangoli a una lista e la *mesh* finale è l'unione di tutti questi triangoli. Più piccoli sono i cubi, più piccoli saranno i triangoli della *mesh*,

rendendo l'approssimazione più simile alla funzione target. La superficie di output dovrà sempre adempiere a queste condizioni: deve superare i punti interni da quelli esterni, deve avere un orientamento ed essere chiusa.

Lavorando coi volumi di *Dendro*, "Volume To Mesh" è il singolo componente di output per riportare i dati volumetrici in un formato nativo Grasshopper. Il componente "Volume To Mesh" prenderà input Volume (V) singoli o multipli e li convertirà nell'output della *mesh*. Le impostazioni che guidano la generazione della *mesh* sono *Isovalue* e *Adaptivity* che escono dal componente "Create Settings".

In alcuni casi, la conversione di un volume in una *mesh* può risultare in una "Mesh non valida". La soluzione più semplice in questi casi è apportare una modifica all'*Isovalue* e portarlo da 0 a un numero piccolo come 0,001 o 0,002, che di solito si tradurrà in una *mesh* output valida. Se il problema persiste, si può provare a modificare le impostazioni di *Adaptivity*.

Affinché il componente funzioni, la *mesh* dev'essere "chiusa". Si possono generare volumi validi da *mesh* "aperte" se la *voxel size* è maggiore di qualsiasi spazio vuoto nella *mesh*. Questa funzione può essere utile per la riparazione della *mesh* o per creare una *mesh* a tenuta stagna per la stampa 3D.

Prima di esportare la *mesh* si unificano le direzioni delle normali delle facce della *mesh* in modo che siano coerenti, tramite il comando "Mesh Unify Normals".

A seguire la progettazione del modello, vi è la fase di produzione, che esula dall'obiettivo di questo elaborato. Per preparare un modello 3D che può essere stampato con successo, questo deve essere a tenuta stagna, e per ottenerlo è necessaria una polisuperficie chiusa. Il motivo più comune per cui i modelli diventano non stampabili è dovuto a *bad objects*, ovvero oggetti che presentano problemi strutturali, e *naked edges*, già discussi in precedenza. Dopo aver corretto le eventuali problematiche che emergono, il modello deve essere esportato come modello *mesh* per la stampa 3D, in formato .STL.

Capitolo 3

Risultati e discussione

Nel metodo proposto nel capitolo precedente sono state illustrate diverse strategie per la generazione di gradienti di porosità. Ora si mostrano e si discutono i risultati ottenuti, e si riportano infine alcune problematiche riscontrate durante l'implementazione delle tecniche di modellazione volumetrica di queste strutture.

Alla maggior parte delle strutture realizzate in Grasshopper, a seguito della creazione del gradiente tramite modelli volumetrici e prima di generare la *mesh*, è stato applicato un filtro al volume ottenuto, tramite il componente "Smooth Volume" di *Dendro* (Figura 3.1):

Di default il valore è settato a 2, quindi sul filtro di tipo Media. Spesso si è scelto quello gaussiano (*type 0*). Il funzionamento di questo filtro è simile al filtro mediano, ma si differenzia per i pesi della maschera. I pesi sono distribuiti in maniera significativa verso il centro e in maniera meno significativa verso la periferia.

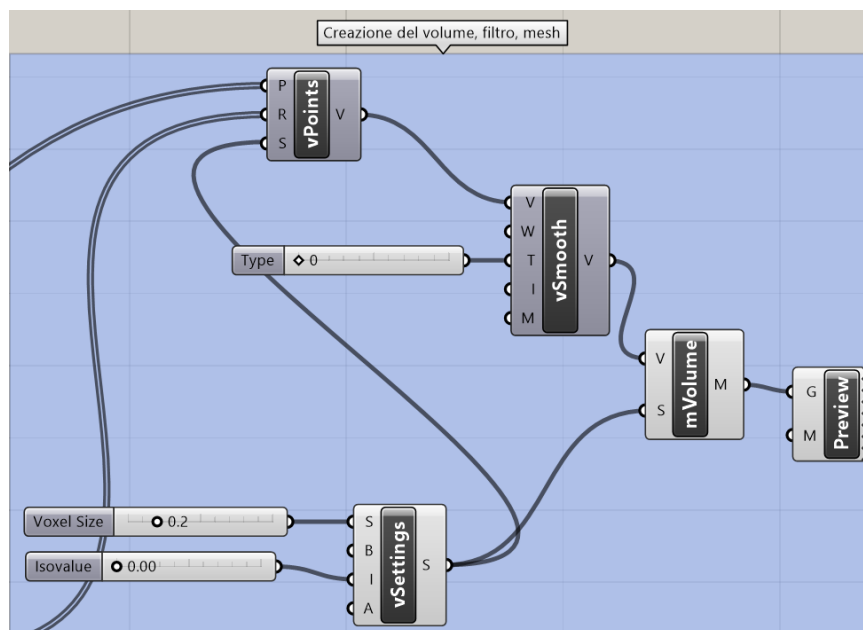


Figura 3.1. Applicazione di smooth del volume

3.1 FGLSs ottenute

- Risultati dei modelli con gradiente ottenuto da distanza da entità geometrica

Nel primo modello presentato, il gradiente è stato generato dalla distanza da un'entità geometrica.

Con la cella *octet-truss*, di lato $L = 10\text{ mm}$, scegliendo opportunamente i parametri che vengono fatti variare, si può ottenere, per esempio, una struttura con densità relativa che varia dal 70% al 20%, e di conseguenza porosità dal 30% all'80% (Figura 3.2):

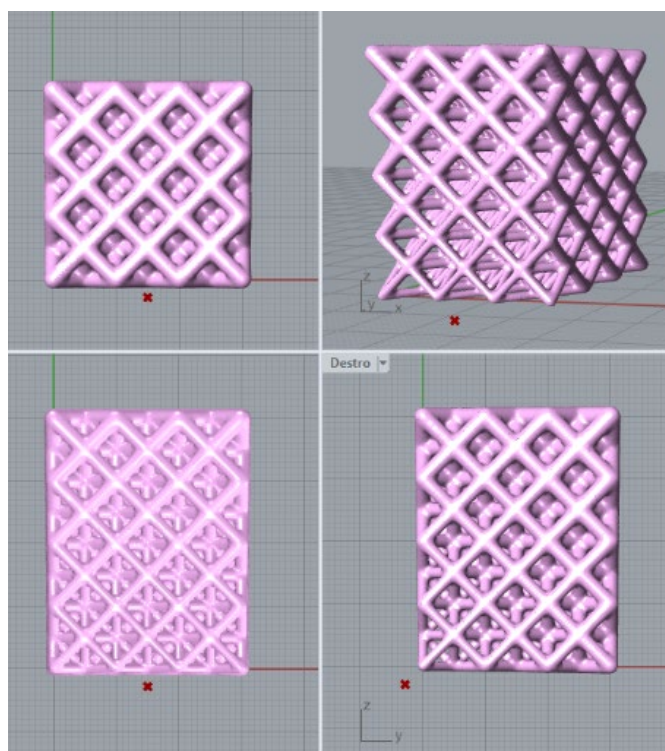


Figura 3.2. Viste della struttura reticolare octet-truss con gradiente tramite distanza da punto attrattore

La stessa struttura è stata ottenuta anche a partire da una cella cubica a corpo centrato, con le stesse identiche dimensioni, e con la stessa densità relativa da raggiungere (Figura 3.3):

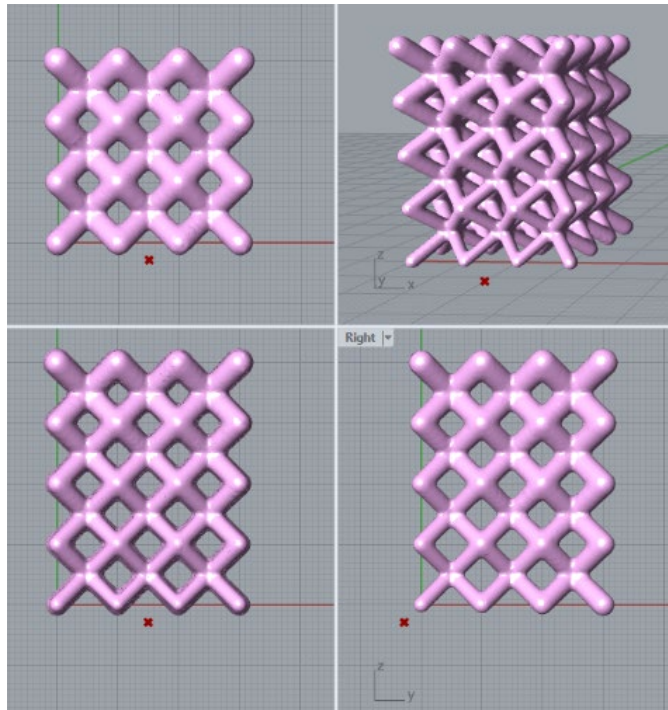


Figura 3.3. *Viste della struttura reticolare cubica a corpo centrato con gradiente tramite distanza da un punto attrattore*

Quello che si può notare è che le due configurazioni presentano una significativa differenza nello spessore delle aste a parità di porosità: nel caso delle celle cubiche a corpo centrato è necessario aumentare significativamente il diametro.

Si mostra anche il risultato ottenuto per una struttura cubico a corpo centrato con distanza da un piano, posto alla base inferiore nel piano xy (Figura 3.4)

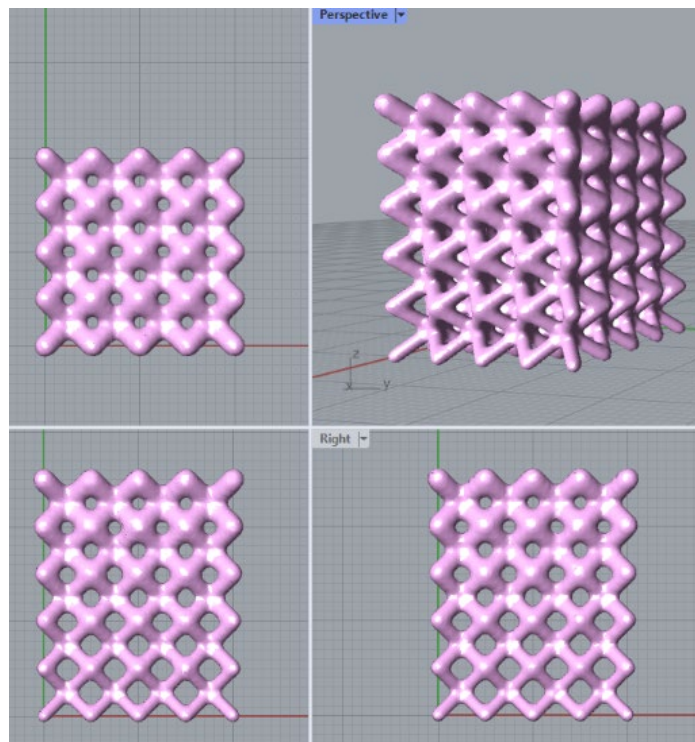


Figura 3.4. *Viste della struttura reticolare cubica a corpo centrato con gradiente tramite distanza da un piano attrattore*

Qui si può osservare che, rispetto al punto attrattore, il piano consente di avere un gradiente verticale uniforme sia frontalmente, sia lateralmente.

- Risultati dei modelli con gradiente ottenuto da espressione matematica

Nella generazione del gradiente per le strutture TPMS con valutazione di un'espressione, vi è la libertà di poter scrivere l'equazione che meglio descrive il gradiente che si vuole ottenere. In questo caso l'espressione che si valuta è di tipo lineare, lungo z.

Di seguito si confrontano tutte e tre le TPMS analizzate (Figura 3.5;3.6;3.7). Si è scelto di mantenere la stessa variazione di densità relativa, dal 75% al 25%.

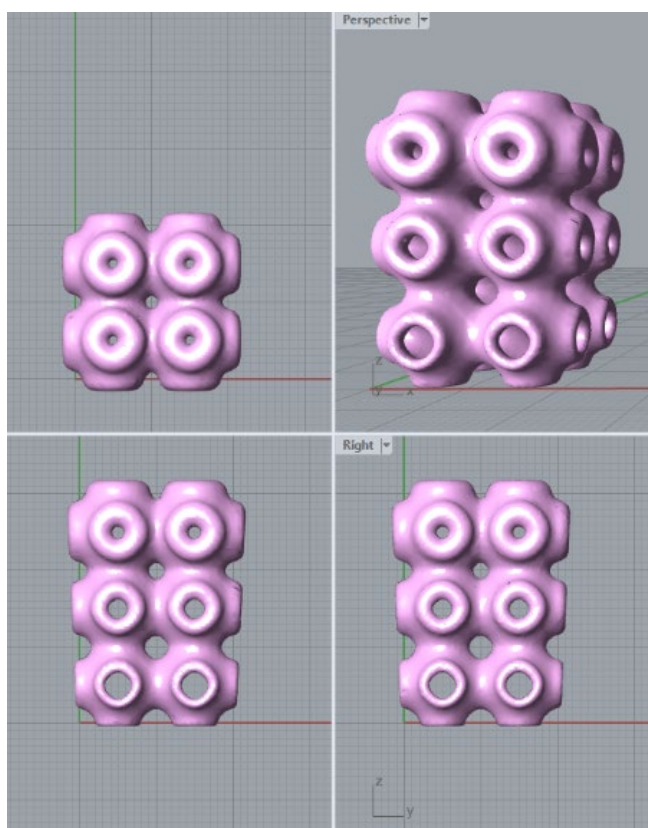


Figura 3.5. *Viste della struttura reticolare Schwarz Primitive con gradiente lungo l'asse z*

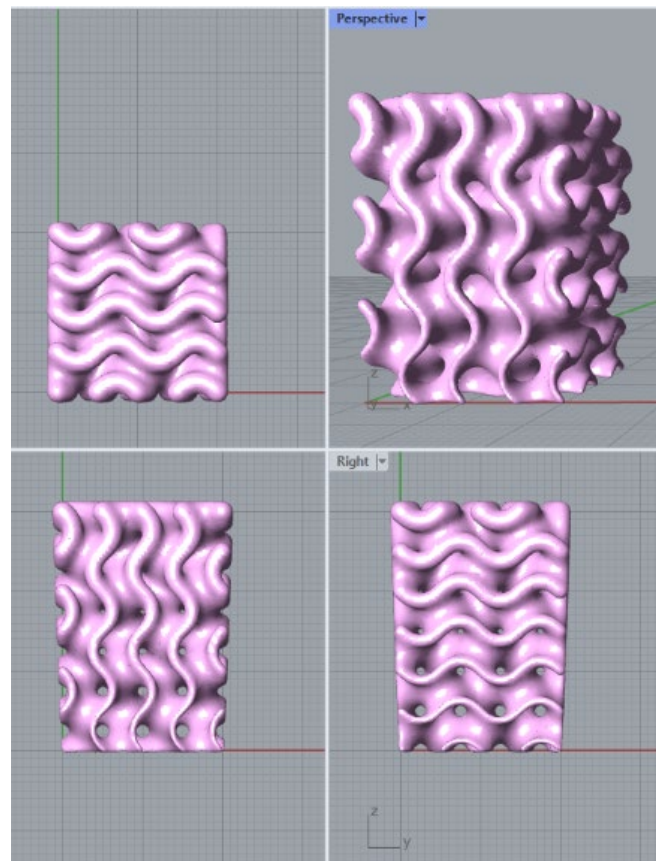


Figura 3.6. *Viste della struttura reticolare a giroide con gradiente lungo l'asse z*

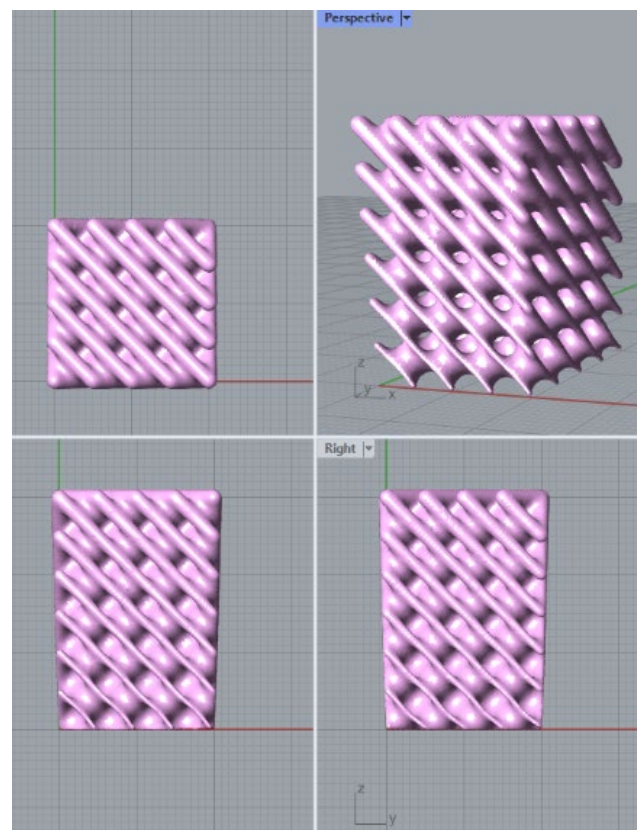


Figura 3.7. Viste della struttura reticolare Schwarz Diamond con gradiente lungo l'asse z

Quello che si può notare è che, a parità di densità relativa, la struttura reticolare costruita dalla *Schwarz Diamond* è quella che richiede un diametro delle sfere più piccolo per raggiungere la stessa densità delle altre TPMS. Dimensioni poco maggiori sono attribuite al giroide. Valori di raggio più elevati sono richiesti dalla *Schwarz Primitive*, rendendola adatta per applicazioni che richiedono rigidità e resistenza elevate in una direzione di carico definita. Si può osservare, inoltre, che questa struttura è tra le più permeabili.

Si riporta l'algoritmo completo per la modellazione volumetrica di queste strutture (Figura 3.8):

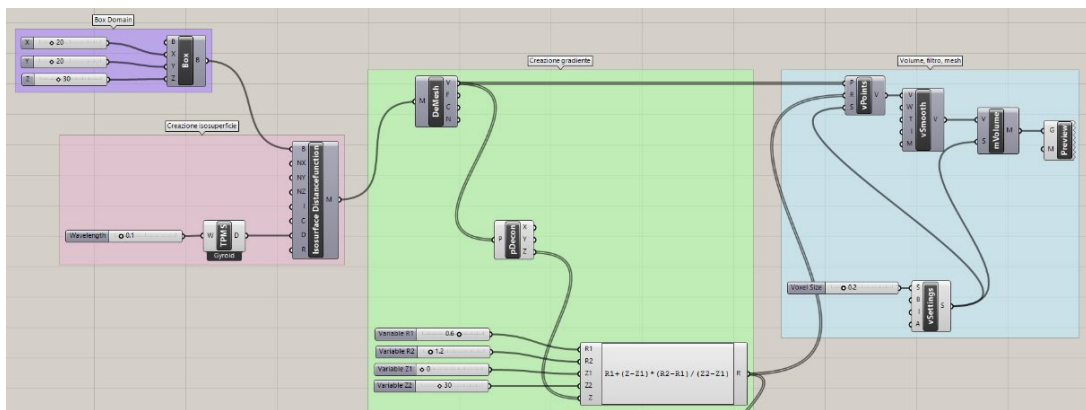


Figura 3.8. Algoritmo di modellazione del giroide a gradiente lungo l'asse z

Le funzioni si possono utilizzare anche per definire le TPMS, come visto in fase di definizione della densità relativa. Aggiungendo all'espressione scritta nel componente *Math* un'altra funzione, per esempio $g * z$, dove g è un parametro, e z la componente verticale, si può creare un altro gradiente verticale.

Si è creata una struttura 30x30x30 mm; con 30 come risoluzione di x, y e z e si è generata una Schwarz P in fase solida, assegnando l'isovalore, in questo caso $k=0.3$, cui corrisponde una densità relativa di circa il 55% che decresce lungo z.

A seguito del *bake* della *mesh* in Rhinoceros, si chiude con il comando *Cap*, ottenendo così la struttura cercata (Figura 3.9).

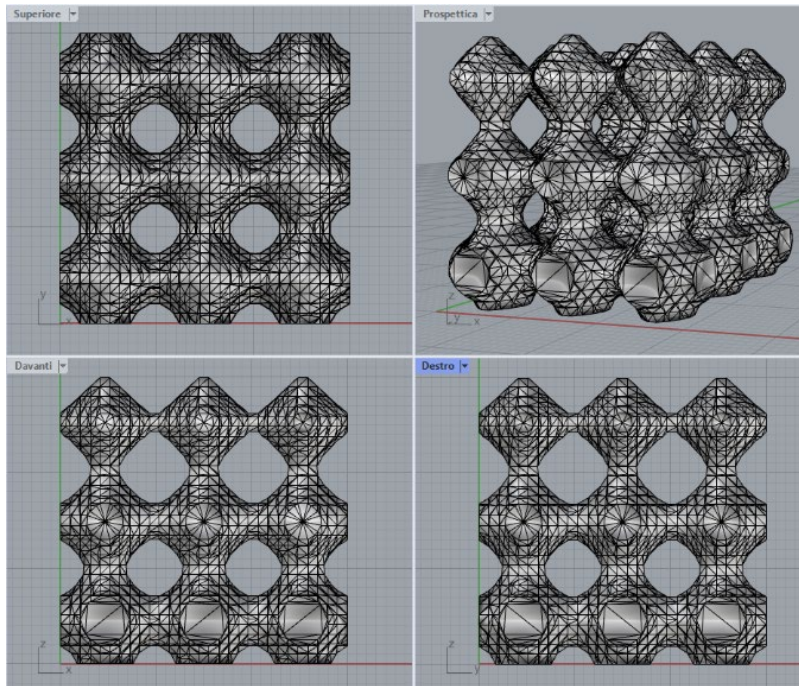


Figura 3.9. Viste di una struttura Schwarz P in fase solida con gradiente

Si riporta anche l’algoritmo di modellazione messo a punto (Figura 3.10)

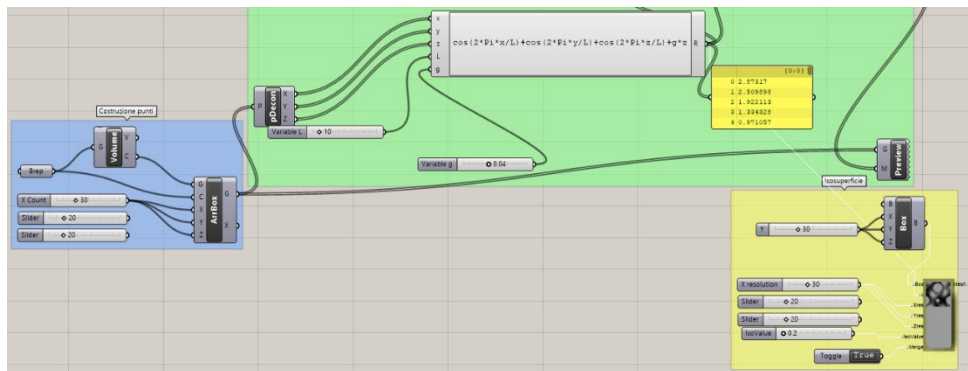


Figura 3.10 Algoritmo di modellazione di una TPMS in fase solida

- Risultati dei modelli con gradiente da *Graph Mapper*

Nella realizzazione del gradiente tramite *Graph Mapper*, si è selezionata una funzione, tra quelle proposte nel componente di Grasshopper, che permettesse di ottenere una struttura con porosità massima al centro, e minima nella parete superiore e inferiore. In questo caso si è scelta una parabola.

La struttura ottenuta parte da una cella cubica a corpo centrato, altamente porosa al centro: la densità relativa, dal 65%, decresce fino a giungere al circa il 10% al centro.

I risultati vengono mostrati nella Figura 3.11.

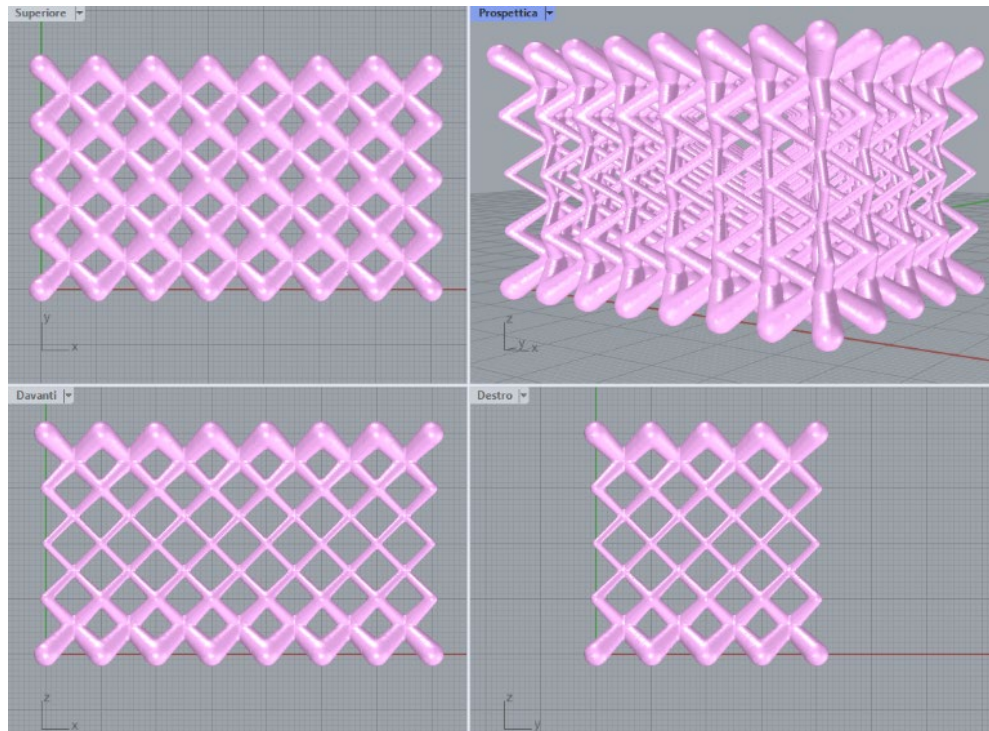


Figura 3.11. *Viste della struttura reticolare graduata a porosità controllata massima al centro*

Strutture di questo tipo vengono richieste per avere un'alta permeabilità nella zona centrale, ma sostenuta da pareti più rigide. Se si fa riferimento alla biodegradabilità di un sistema, questo profilo consente di avere due differenti tipi di degradazione.

- Risultati del modello con gradiente da immagine bitmap

Come fonte bitmap si può scegliere, ad esempio, una sezione trasversale di diafisi femorale e, tramite l'utilizzo del plug-in *Monolith*, creare una struttura a giroide la cui densità viene definita sulla base di quest'immagine (Figura 3.12).

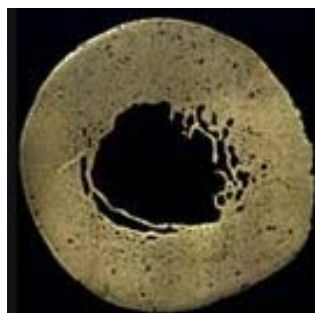


Figura 3.12. *Esempio di sezione trasversale di diafisi femorale*

Per poter realizzare l'algoritmo, è innanzitutto necessario costruire un campo voxel che definisce il confine, la risoluzione voxel, ovvero il numero di voxel in cui si vuole discretizzare il modello e la configurazione del canale con il comando "Create Voxel". Attraverso "Function Source" si può descrivere la funzione matematica sorgente per la generazione del campo. In questo caso si seleziona un giroide da quelle disponibili nel preset. È necessario popolare i voxel creati, inizialmente privi di informazioni, con "Add source", definendo i canali della forma e della distribuzione di materiale. Per creare le isosuperfici si utilizza il comando "Get Iso Mesh", dando in input anche l'intervallo di isovalori (Figura 3.13).

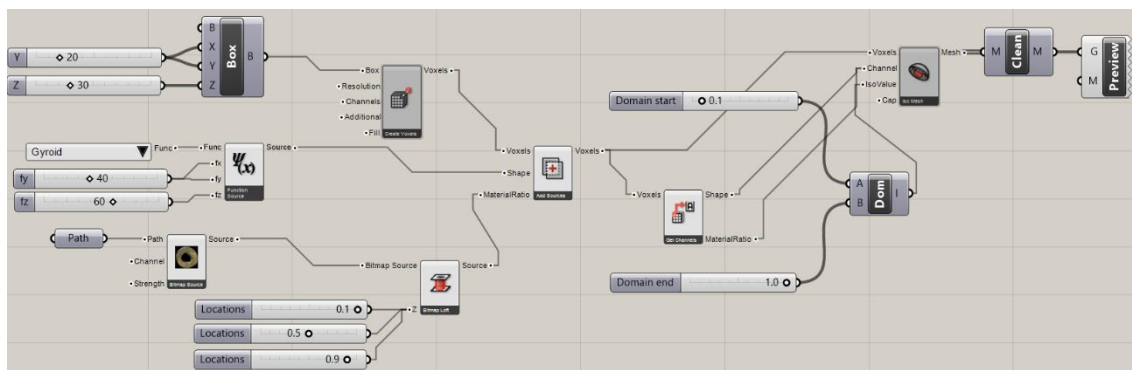


Figura 3.13. Algoritmo di modellazione di una struttura reticolare con densità definita da bitmap

Ecco la struttura che si è ottenuta (Figura 3.14):

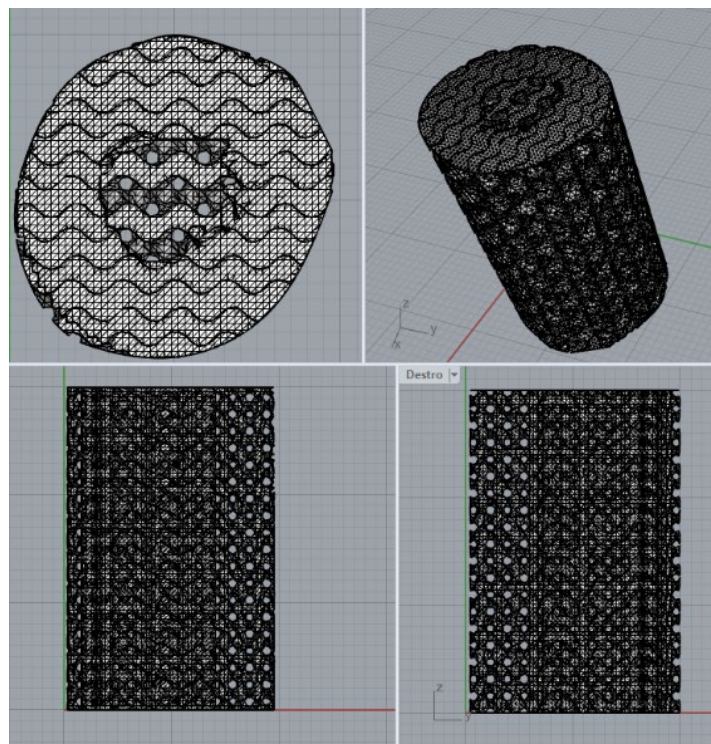


Figura 3.14. Viste della struttura reticolare creata da bitmap con Monolith

Nell'appendice B si potranno trovare altri esempi di strutture volumetriche, tra cui la modellazione di algoritmi che consentono di realizzare Voronoi 3D con diverse strategie e l'applicazione di gradienti radiali.

3.2. Limiti riscontrati

Tra i limiti riscontrati in fase di elaborazione degli algoritmi di modellazione volumetrica proposti, è innanzitutto opportuno sollevare la questione legata alla voxelizzazione dei modelli.

All'aumentare della dimensione dei voxel, vi è un notevole risparmio legato ai tempi di elaborazione, a scapito di una visualizzazione pixellata dovuta all'*aliasing*. Per ridurre quest'effetto, è necessaria un'elevata risoluzione: quest'aspetto rischia di diventare problematico nel momento in cui si vogliono sfruttare i vantaggi legati all'utilizzo di questa tecnica di modellazione volumetrica discreta, per applicazioni che richiedono controllo delle proprietà all'interno dell'oggetto.

A titolo di esempio, si mostrano i tempi di elaborazione richiesti dal software per ciascuna *voxel size* del modello volumetrico presentato di struttura reticolare TPMS con gradiente determinato da un'equazione (Tabella 3.1).

Voxel size (mm)	Tempo (s)
0.05	40.1
0.08	16.6
0.1	11.9
0.15	6.3
0.2	4.5
0.23	4

Tabella 3.1. *Voxel size/Tempo di elaborazione*

I risultati si possono così graficare (Figura 3.15):

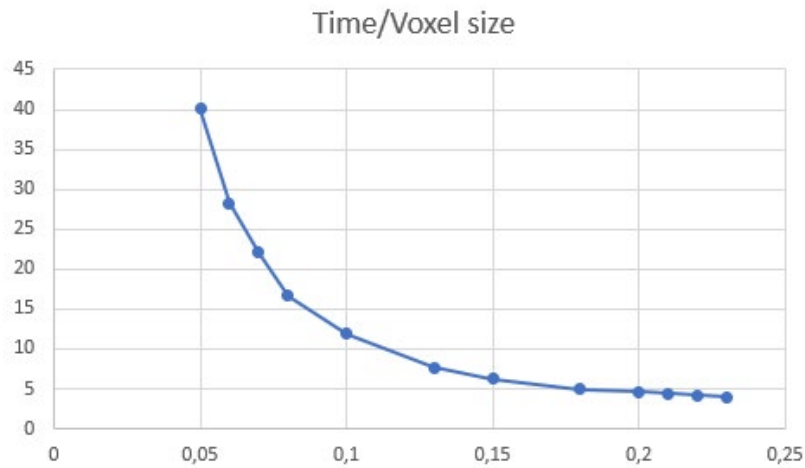


Figura 3.15. Andamento dei tempi di elaborazione in funzione della dimensione dei voxel

La situazione si complica anche all'aumento del numero di celle di cui è composta la struttura.

All'aumentare delle celle che compongono il modello, aumentano i tempi di elaborazione.

Si riportano i dati misurati relativi al modello *octet-truss* con gradiente basato sulla distanza da un punto attrattore, modificando lo slider dell'*Array*, tenuto uguale per x,y e z e pari a 2,3,5,7 e 10, con voxel size sempre pari a 0.2 mm (Tabella 3.2):

Numero celle per asse	Tempo (s)
2	2.2
3	6.1
5	23.7
7	66
10	186

Tabella 3.2. Numero di celle per asse/Tempo di elaborazione

I risultati corrispondono al seguente grafico (Figura 3.16)

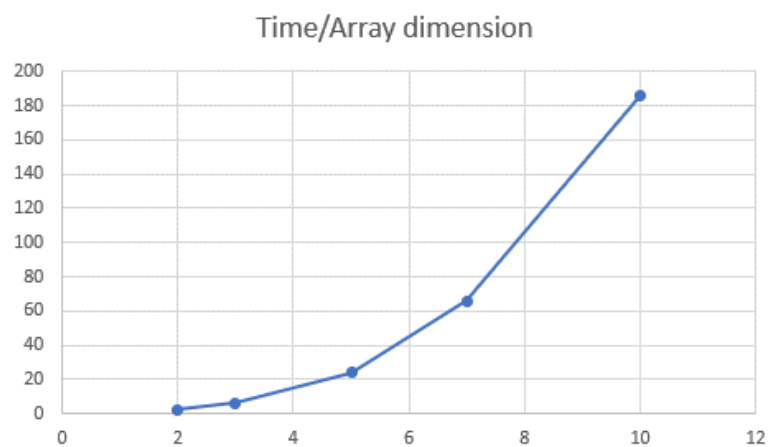


Figura 3.16. Andamento dei tempi di elaborazione in funzione del numero di celle per asse

Infine, per lo stesso modello, si computa anche il tempo per elaborare diverse suddivisioni dei punti della struttura, operazione necessaria per generare il gradiente assegnando diversi valori di raggio alle sfere di volume ottenute (Tabella 3.3):

Numero di punti	Tempo (s)
100	3.7
150	3.9
200	4.3
250	4.5
300	4.8
500	6.1
1000	9.3
2000	14.8

Tabella 3.3. Numero di celle per asse/Tempo di elaborazione

Ecco il grafico ottenuto (Figura 3.17):

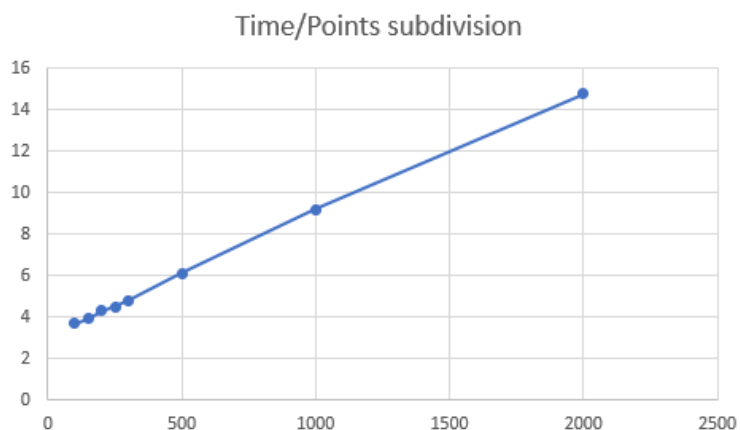


Figura 3.17. *Andamento dei tempi di elaborazione in funzione del numero di punti di suddivisione della struttura*

Dall'andamento delle funzioni sopra riportate si possono cogliere interessanti aspetti sull'utilizzo della discretizzazione in voxel dei modelli volumetrici. Per quanto riguarda la dimensione dei voxel, si nota un aumento esponenziale del tempo di elaborazione al diminuire della dimensione del voxel. Al di sotto di una certa soglia, il tempo richiesto quadruplica dimezzando la *voxel size*: se la dimensione del cubetto si dimezza, per riempire lo stesso volume è necessario quadruplicare il numero di voxel, e di conseguenza il tempo per crearli. Anche il numero di celle per asse causa un andamento simile, a parità di *voxel size*. Il tempo in funzione dei punti di suddivisione assume invece un andamento lineare.

Conclusioni e direzioni future

Grazie ai grandi vantaggi esibiti, i FGMs saranno tra i materiali con un ruolo più significativo nel prossimo futuro. Pertanto, sarà necessario uno sforzo maggiore per migliorare e sviluppare i metodi di progettazione, in modo da adattarsi alla quarta rivoluzione industriale.

L'impiego di FGLSs nella progettazione e produzione di impianti ortopedici dovrebbe avere un futuro promettente (Mahmoud & Elbestawi, 2017). Le strutture reticolari a gradiente di porosità si distinguono per essere in grado di ridurre la protezione dallo stress e al contempo migliorare l'osteointegrazione e la stabilità dell'impianto.

Per aumentare sostanzialmente l'uso di queste tecnologie nel settore degli impianti biomedicali, è necessario affrontare diverse sfide:

- La caratterizzazione complessiva delle strutture reticolari deve essere migliorata. Dovrebbe essere sviluppato un protocollo standard per la valutazione delle dimensioni, della microstruttura e delle prestazioni meccaniche.
- Dovrebbe essere stabilita una libreria per le diverse celle unitarie utilizzate negli impianti ortopedici per consentire la valutazione delle prestazioni delle celle unitarie in diverse applicazioni. Inoltre, il database dovrebbe contenere informazioni sia meccaniche, sia biologiche sulle varie celle unitarie.
- Lo sviluppo di nuovi FGMs progettati a partire da biomateriali che utilizzano tecnologie AM dovrebbe essere studiato in modo più approfondito. La combinazione di materiali metallici, ceramici e polimerici potrebbe portare a un impianto con elevata funzionalità. La presenza di un database condiviso che integri le informazioni sui materiali può essere utile per evitare risultati indesiderati.
- Sono necessari ulteriori studi sperimentali per convalidare questi modelli e migliorare ulteriormente la loro applicabilità negli impianti ortopedici, valutando anche le prestazioni a lungo termine.

In vista del raggiungimento degli obiettivi futuri sperati, grande attenzione dovrebbe essere posta nella modellazione di queste strutture.

Il contributo di questo lavoro di tesi è dato dalla messa a punto di una metodologia di progettazione basata sulla modellazione volumetrica in grado di progettare efficacemente strutture reticolari che presentano un gradiente di porosità funzionale alle applicazioni biomediche, superando le rappresentazioni inadeguate basate sui tradizionali sistemi di progettazione dei confini, che non tengono in considerazione la struttura interna. Le FGLSs presentano strutture interne complesse e richiedono una precisa distribuzione del materiale a livello microstrutturale. A tal fine si è mostrata la potenzialità delle tecniche basate su voxel o su rappresentazione di funzioni di distanza offerta dai *plug-in* per la modellazione volumetrica di Grasshopper, innovativi in uno scenario di scarsa disponibilità di software dedicati a questo scopo.

In definitiva, con questi primi passi nel mondo della modellazione volumetrica, si sono poste le basi su cui si può pensare di costruire, nel prossimo futuro, strutture sempre più efficienti e applicabili, in modo da sfruttare al meglio tutti i vantaggi che offrono.

Appendice A

In questa appendice viene riportato un approfondimento sul funzionamento e i vantaggi dell'utilizzo della libreria OpenVDB, che costituisce un elemento fondamentale del *plug-in Dendro*.

OpenVDB è una libreria C++ *open source* che consiste in una nuova struttura di dati gerarchici per la rappresentazione efficiente di dati volumetrici sparsi e variabili nel tempo, discretizzati su una griglia 3D. "VDB", così chiamata perché è una griglia volumetrica e dinamica che condivide diverse caratteristiche con gli alberi B+, sfrutta la coerenza spaziale dei dati variabili nel tempo per codificare separatamente e in modo compatto i valori dei dati e la topologia della griglia. VDB modella uno spazio di indice 3D virtualmente infinito che consente un accesso rapido e coerente ai dati in volumi sparsi ad alta risoluzione. Non impone restrizioni di topologia alla scarsità dei dati volumetrici e supporta modelli di accesso casuale rapidi (media $O(1)$) quando i dati vengono inseriti, recuperati o eliminati. Ciò è in contrasto con la maggior parte delle strutture dati volumetriche sparse esistenti, che presuppongono una topologia statica o multiforme e richiedono modelli di accesso ai dati specifici per compensare un accesso casuale lento. Poiché la struttura dei dati VDB è fondamentalmente gerarchica, facilita anche il campionamento adattivo della griglia e la struttura di accelerazione intrinseca porta ad algoritmi veloci che sono adatti per le simulazioni.

Nella maggior parte dei casi i dati volumetrici sono rappresentati su griglie 3D regolari e spazialmente uniformi, in parte perché tali rappresentazioni sono semplici e convenienti. Inoltre, la maggior parte degli algoritmi volumetrici e degli schemi numerici hanno una forte preferenza per il campionamento uniforme. La discretizzazione di operatori differenziali, l'interpolazione, i kernel di convoluzione e altre tecniche simili possono essere generalizzate solo con difficoltà a griglie non uniformi. Tuttavia, alcune applicazioni volumetriche, come la modellazione volumetrica e il ray marching, traggono vantaggio dal campionamento a risoluzione variabile, quindi è desiderabile impiegare una struttura dati che supporti sia il campionamento uniforme, sia gerarchico. Inoltre, molti algoritmi volumetrici come la geometria solida computazionale (CSG) traggono vantaggio in modo significativo da una rappresentazione gerarchica dei dati. Sebbene le griglie regolari dense siano convenienti per

diversi motivi, soffrono almeno di un grave difetto: la loro impronta di memoria cresce proporzionalmente al volume dello spazio di inclusione. Poiché la maggior parte delle applicazioni volumetriche utilizzate nella produzione di effetti visivi non richiedono che i dati vengano campionati in modo uniforme ovunque in una griglia densa, la soluzione è chiaramente quella di impiegare una struttura di dati volumetrica sparsa. Un volume così sparso dovrebbe idealmente avere il vantaggio che l'impronta di memoria si ridimensiona solo con il numero di voxel che contengono valori campionari significativi e non con il volume dello spazio di inclusione denso. Sebbene siano state proposte numerose strutture di dati 3D sparse, la maggior parte è progettata esplicitamente per il campionamento adattivo, ha un accesso ai dati lento o restrittivo, non scala a una risoluzione estrema o non può gestire facilmente simulazioni numeriche con topologia dinamica. La struttura dei dati VDB è efficiente in termini di memoria, supporta la simulazione di dati variabili nel tempo, è in grado di codificare una topologia arbitraria e facilita il campionamento sia uniforme, sia adattivo, consentendo al contempo un accesso rapido e illimitato ai dati.

Riassumendo, VDB si caratterizza per:

- Dinamicità

A differenza della maggior parte delle strutture dati volumetriche sparse, VDB è sviluppata sia per la topologia dinamica, sia per i valori dinamici tipici delle simulazioni numeriche dipendenti dal tempo e dei volumi animati. Ciò richiede un'implementazione efficiente di iteratori alle differenze finite sparsi, morfologia topologica e algoritmi di ricostruzione, nonché buffer di valori temporali per schemi di integrazione numerica coerenti con la cache;

- Memoria efficiente

L'allocazione dinamica e gerarchica di nodi compatti porta a una struttura di dati sparsi efficiente in termini di memoria che consente un'estrema risoluzione della griglia. Per ridurre ulteriormente l'ingombro su disco, si utilizza un'efficiente tecnica di compressione basata sulla topologia che può essere combinata con la quantizzazione dei bit e gli schemi di compressione standard;

- Topologia generale

A differenza della maggior parte delle strutture di dati dinamici esistenti per set di livelli a banda stretta, VDB può rappresentare efficacemente dati di volume sparsi con una topologia dinamica arbitraria. Ciò implica che nel contesto di griglie strutturate regolari VDB può

essere utilizzato come una struttura di dati volumetrica generica, invece di supportare semplicemente applicazioni dinamiche di set di livelli.

- Accesso rapido ai dati casuale e sequenziale

VDB supporta la ricerca, l'inserimento e l'eliminazione di dati casuali a tempo costante veloce. Offre inoltre iteratori di accesso a stencil sequenziali veloci (a tempo costante), essenziali per simulazioni efficienti che utilizzano schemi alle differenze finite. I modelli di accesso spazialmente coerenti hanno anche una complessità computazionale ammortizzata che è indipendente dalla profondità dell'albero B+ sottostante.

- Spazio praticamente infinito

VDB modella una griglia illimitata nel senso che lo spazio delle coordinate accessibile è limitato solo dalla precisione in bit delle coordinate segnate.

- Algoritmi gerarchici efficienti

La struttura B+ *tree* offre i vantaggi della coerenza della cache, dell'accelerazione intrinseca del volume di delimitazione e delle operazioni rapide per ramo (rispetto per voxel). Inoltre si presta bene a tecniche di ottimizzazione standard come il blocco, la vettorizzazione SSE e il multithreading.

- Risoluzione adattiva

A differenza della maggior parte delle strutture dati a set di livelli a banda stretta esistenti, VDB è gerarchica e può memorizzare valori in qualsiasi livello della struttura ad albero sottostante. Tuttavia, è importante notare che questi valori multilivello in base alla progettazione non si sovrappongono nello spazio dell'indice. Pertanto, VDB è diverso dalle strutture dati multirisoluzione come brickmap o mipmap che rappresentano concettualmente gli stessi valori nello spazio dell'indice a più livelli di dettaglio.

- Semplicità

Rispetto ad alcune strutture di dati sparse esistenti VDB è relativamente semplice sia da implementare, sia da applicare. Si basa su concetti ben noti come blocchi e alberi e supporta l'accesso casuale attraverso un'interfaccia simile ai normali volumi densi.

- Configurabilità

In base alla progettazione, VDB è altamente configurabile in termini di profondità dell'albero, fattori di ramificazione e dimensioni dei nodi. Ciò consente di adattare la griglia

alle applicazioni specifiche al fine di ottimizzare fattori come l'ingombro della memoria, l'utilizzo della cache e i modelli di accesso previsti.

- Algoritmo di memoria esterno

VDB supporta il semplice streaming out-of-core. Più specificamente, si può ridurre l'ingombro in memoria memorizzando i valori della griglia fuori dal core e mantenendo in memoria solo la topologia della griglia. I valori vengono quindi caricati su richiesta, ad esempio durante l'accesso ai blocchi durante il ray-tracing.

Sebbene molte strutture dati esistenti possiedano certamente sottoinsiemi di queste caratteristiche, VDB è la prima a incarnarle tutte.

Appendice B

Nella seguente appendice si riportano ulteriori algoritmi di modellazione volumetrica, che seguono il metodo messo a punto in questo elaborato. Si tratta di più versioni delle strutture Voronoi e una struttura che presenta due gradienti radiali diversi al suo interno.

- Strutture Voronoi

In questi modelli sono state progettate FGLS porose per imitare da vicino la struttura naturale dell'osso trabecolare utilizzando un diagramma Voronoi.

Lo scheletro della struttura si realizza popolando una geometria attraverso il comando “Populate Geometry”, che crea dei punti in maniera randomica all'interno della *Box* individuata. Si può modificare il numero di punti o il *seed*, numero intero che definisce la sequenza esatta di numeri pseudo-casuali per avere una struttura più o meno densa, ma in questo caso si lasceranno i valori predefiniti. Inserendo i punti ottenuti e la *Box* popolata nel componente “Voronoi 3D”, si ottiene il diagramma volumetrico Voronoi. Le celle create vengono rappresentate come Brep. Così vengono visualizzate nella schermata di Rhinoceros (Figura 1).

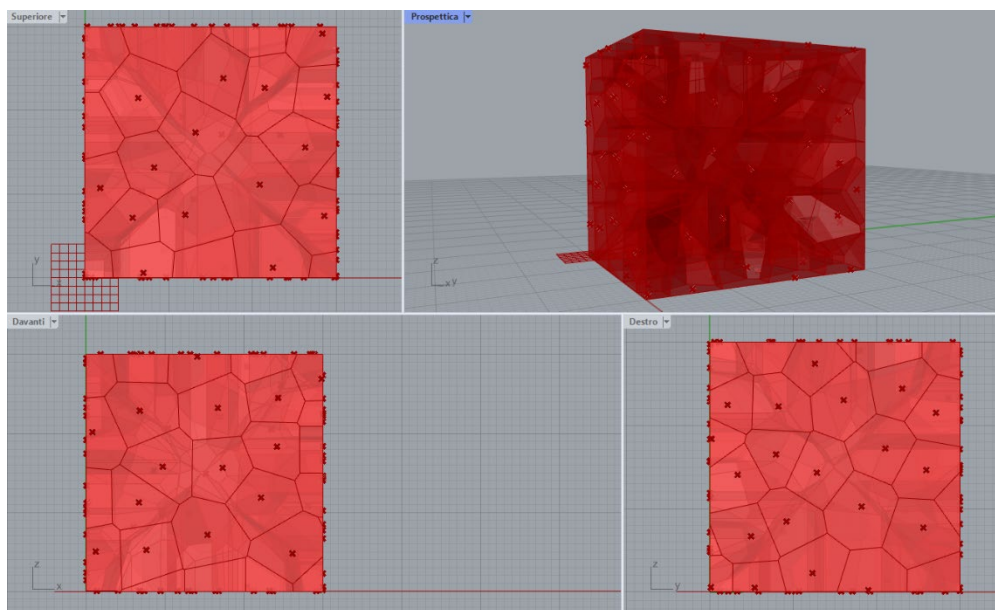


Figura 1. Visualizzazione del Voronoi 3D su Rhinoceros

A questo punto è possibile decostruire le Brep ottenute nelle loro parti costituenti, ovvero le facce, elementi di tipo *surface*, i bordi, di tipo *curve*, e i vertici, di tipo *points*. La strategia che si vuole implementare in questo modello consiste nell'estrarre gli *edges* e applicar loro uno

spessore attraverso il calcolo di un raggio risultate da una funzione matematica. Si vuole creare un gradiente lineare lungo l'asse z.

Dopo aver flattato gli *Edges*, si procede col comando “Remove Duplicate Lines”, presente nel *plug-in* implementato in Grasshopper *Kangaroo 2*, che pulisce la lista di linee, rimuovendo quelle simili tra loro, ovvero combinando quelle i cui punti di inizio/fine più vicini a una tolleranza scelta.

Ora si suddivide la curva in punti e col componente “Deconstruct Point” si decostruiscono i punti, in modo da selezionare la componente z e relazionarla linearmente con i valori di raggi scelti (Figura 2):

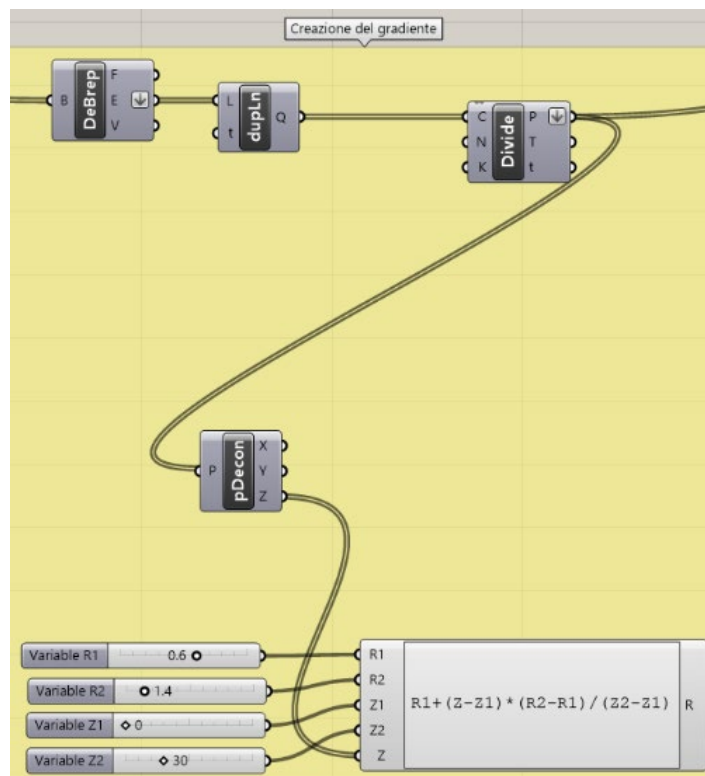


Figura 2. Creazione del gradiente tramite valutazione di un'espressione

I risultati ottenuti dall'espressione vengono inseriti come input del raggio del componente di *Dendro* “Points To Volume”. Il volume ottenuto è stato poi filtrato, preferendo un filtro gaussiano, ed è stata generata la *mesh*. Ecco il risultato ottenuto (Figura 3):

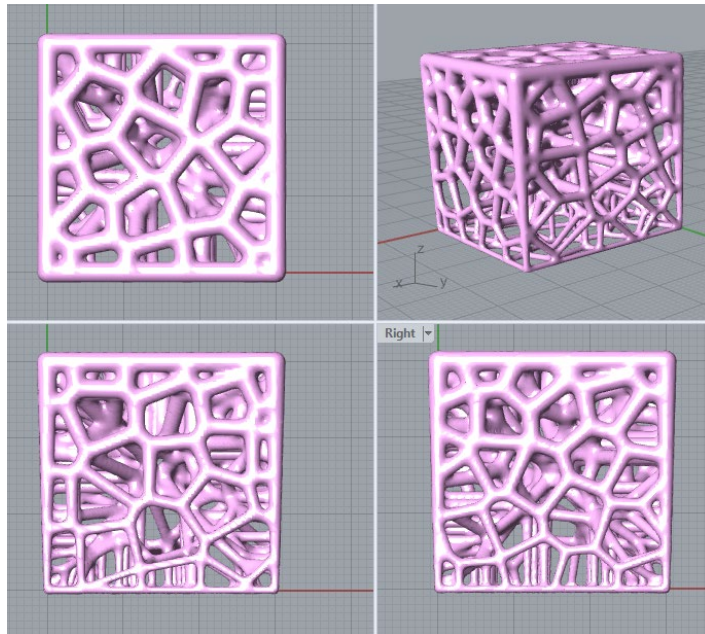


Figura 3. Viste della struttura Voronoi realizzata tramite valutazione di un'espressione

In quest'altro modello, la struttura Voronoi è stata creata su una forma cilindrica. Per fare ciò è stato necessario intersecare le celle con una Brep, il cilindro, attraverso il comando "Solid Intersection": esegue un'intersezione solida su due set Brep.

Il gradiente che si vuole creare stavolta è radiale: si considerano i valori delle coordinate x e y dei punti medi e si valuta l'espressione con la stessa modalità del modello precedente.

Si esegue poi un *Remap* dei dati per fare in modo che rientrino all'interno di un dominio scelto (Figura 4).

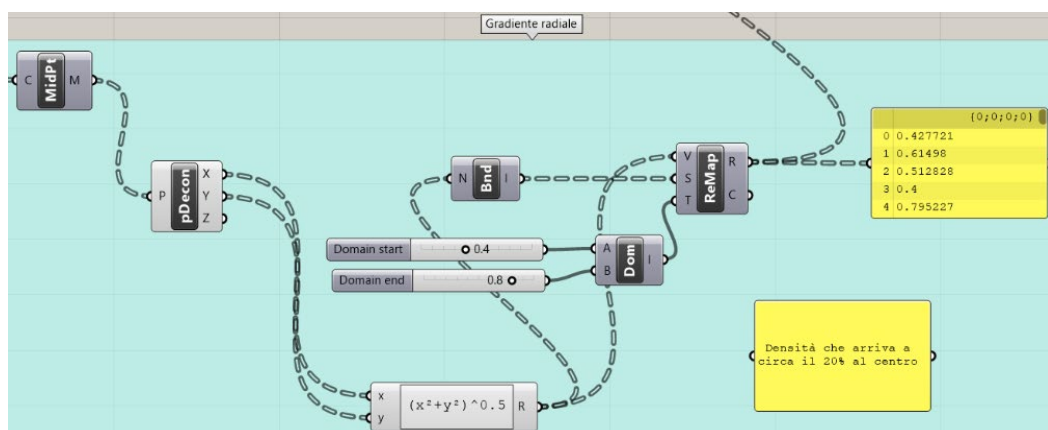


Figura 4. Creazione del gradiente radiale

Emerge però un problema: vi sono valori del raggio che non sono accettabili come input del componente di *Dendro* "Curve To Volume" in quanto più grandi rispetto alla lunghezza di alcune curve. Dopo averle ispezionate, col componente "Dispatch" si inviano gli elementi da una lista in due liste di destinazione, di cui si sceglie quella che contiene le curve più lunghe

rispetto a una lunghezza assegnata. Per poter far funzionare l'algoritmo creato, è necessario aggiungere questa parte.

Si ottiene così un reticolo con diametro degli *edge* delle celle Voronoi che si riduce verso il centro della circonferenza (Figura 5).

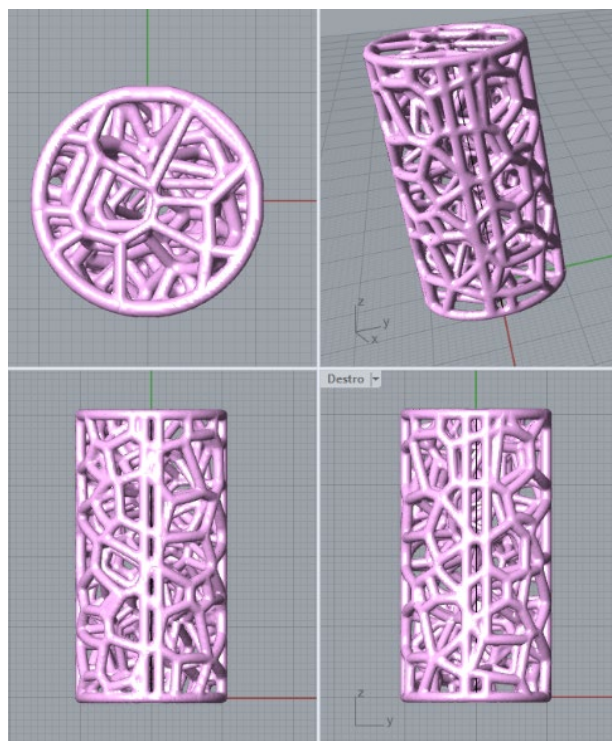


Figura 5. *Viste del cilindro Voronoi realizzata tramite valutazione di un'espressione*

Per ovviare al problema della lunghezza delle curve e avere una variazione del gradiente radiale anche lungo la singola asta, si può modificare l'algoritmo in modo da lavorare con punti, e non con curve. Si osserva il gradiente radiale risultante (Figura 6):

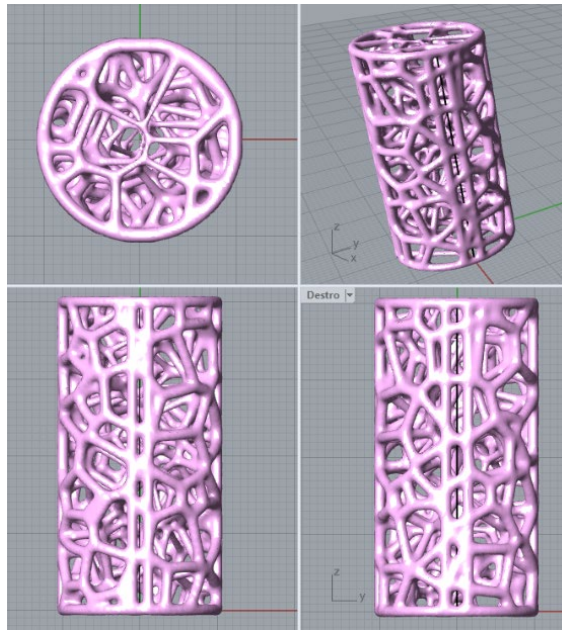


Figura 6. Vista superiore della struttura Voronoi ottenuta

- Struttura a due gradienti

Nella stessa struttura è anche possibile selezionare dei punti in modo tale da creare due gradienti diversi. L'idea alla base di questo algoritmo di modellazione è di modellare una struttura reticolare con cella CCC con una parte esterna molto più densa rispetto alla parte interna.

Per fare ciò, dopo aver suddiviso il reticolo in punti, si sfrutta il comando "Point in Brep", che testa se un punto si trova all'interno di una Brep chiusa. È quindi necessario creare una Brep da inserire al centro della struttura per suddividere i punti. Si sceglie un cilindro, in quanto questa sarà la forma finale che verrà assegnata alla FGLS creata. Si noti che è necessario chiudere le basi del cilindro. L'output di questo componente corrisponde a una lista di valori booleani che indicano se i punti appartengono o meno al cilindro. Attraverso un "Dispatch" si inviano gli elementi della lista in due liste di destinazione: quella dei punti interni (valori True) e quella dei punti esterni (valori False) (Figura 7).

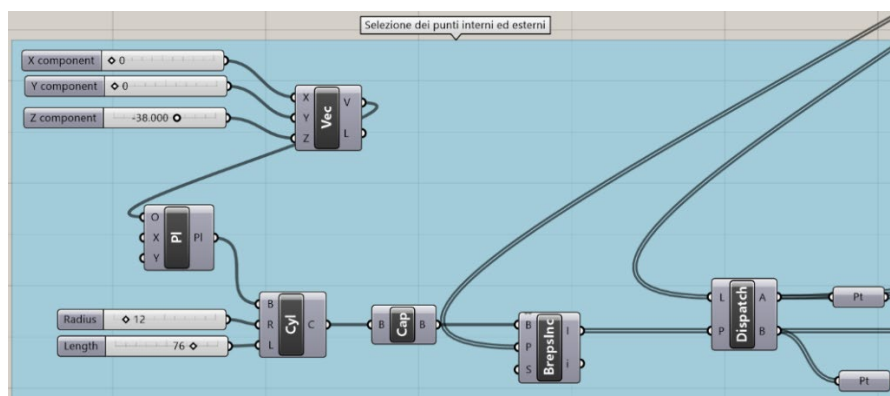


Figura 7. Selezione dei punti esterni ed interni alla Brep che attraversa la struttura reticolare

Si vogliono creare dei gradienti radiali attraverso il metodo dell'espressione già presentato (Figura 8).

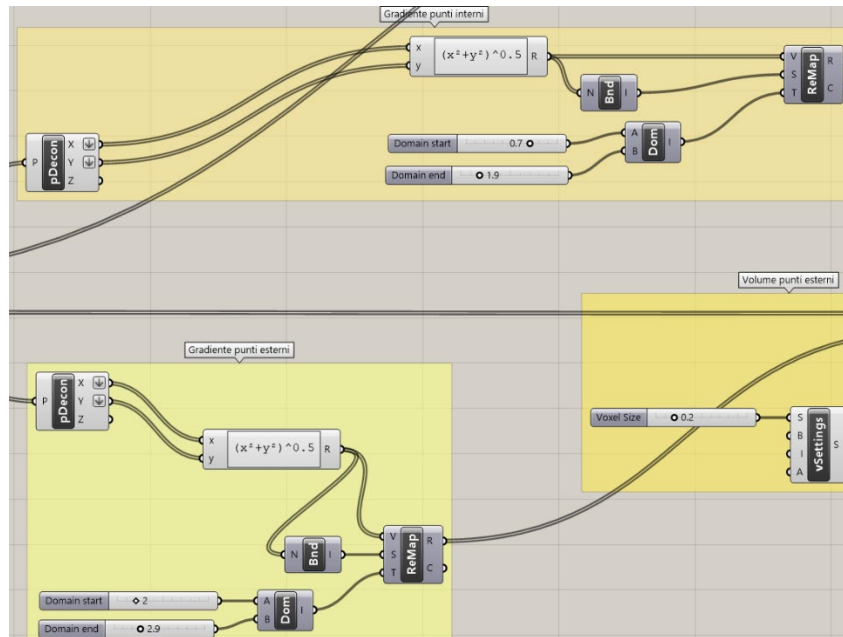


Figura 8. Assegnazione dei gradienti alle due porzioni di struttura

Al termine delle operazioni, col comando di *Dendro* “Volume union” si esegue l’unione sull’insieme di volumi.

Questa è la struttura reticolare graduata ottenuta (Figura 9). Si noti che la struttura cilindrica è stata realizzata in Rhinoceros, dopo aver effettuato un *bake* della struttura realizzata in Grasshopper, effettuando un’operazione di *MeshBooleanIntersection* tra la *mesh* e un cilindro.

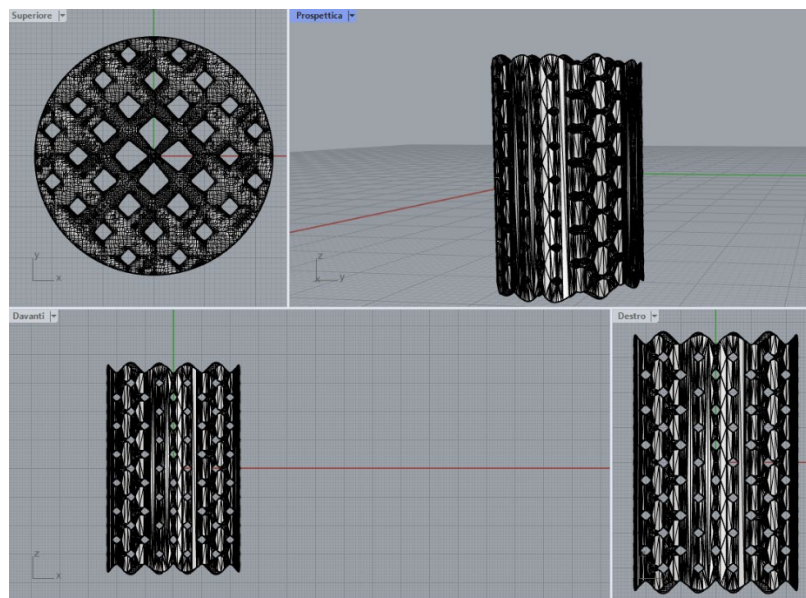


Figura 9. Viste della struttura reticolare a due gradienti radiali

L'algoritmo di modellazione implementato viene di seguito riportato (Figura 10):

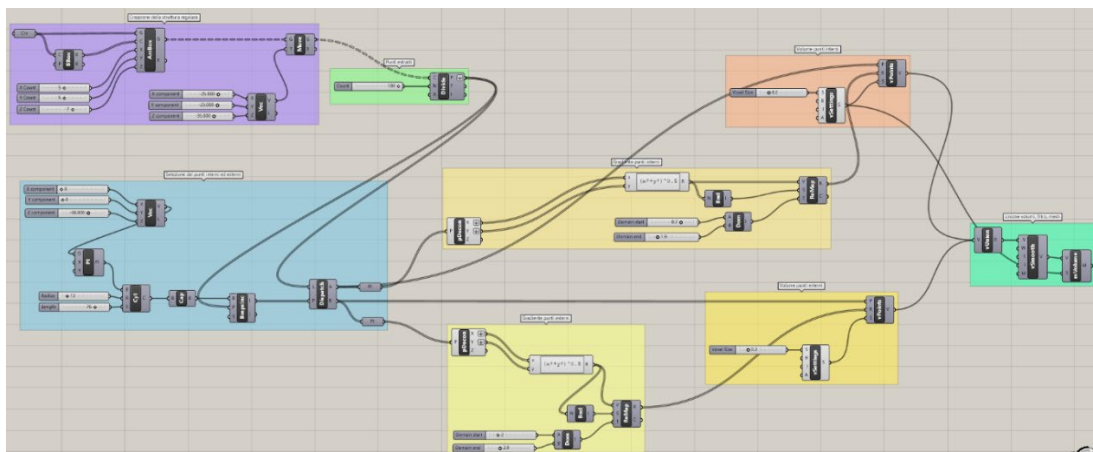


Figura 10. Risultato dell'algoritmo di modellazione a due gradienti

Bibliografia

- Adalsteinsson, D., & Sethian, J. A. (1995). A Fast Level Set Method for Propagating Interfaces. *Journal of Computational Physics*, 118(2), 269–277. <https://doi.org/10.1006/JCPH.1995.1098>
- Alghamdi, A., Maconachie, T., Downing, D., Brandt, M., Qian, M., & Leary, M. (2020). Effect of additive manufactured lattice defects on mechanical properties: an automated method for the enhancement of lattice geometry. *International Journal of Advanced Manufacturing Technology*, 108(3), 957–971. <https://doi.org/10.1007/S00170-020-05394-8>
- Arifin, A., Sulong, A. B., Muhamad, N., Syarif, J., & Ramli, M. I. (2014). Material processing of hydroxyapatite and titanium alloy (HA/Ti) composite as implant materials using powder metallurgy: A review. *Materials and Design*, 55, 165–175. <https://doi.org/10.1016/J.MATDES.2013.09.045>
- Barba, D., Alabort, E., & Reed, R. C. (2019). Synthetic bone: Design by additive manufacturing. *Acta Biomaterialia*, 97, 637–656. <https://doi.org/10.1016/J.ACTBIO.2019.07.049>
- Bharti, I., Gupta, N., & Gupta, K. M. (2013). Novel Applications of Functionally Graded Nano, Optoelectronic and Thermoelectric Materials. *International Journal of Materials, Mechanics and Manufacturing*, 221–224. <https://doi.org/10.7763/IJMMM.2013.V1.47>
- Bhate, D., Penick, C. A., Ferry, L. A., & Lee, C. (2019a). Classification and Selection of Cellular Materials in Mechanical Design: Engineering and Biomimetic Approaches. *Designs 2019, Vol. 3, Page 19*, 3(1), 19. <https://doi.org/10.3390/DESIGNS3010019>
- Bhate, D., Penick, C. A., Ferry, L. A., & Lee, C. (2019b). Classification and Selection of Cellular Materials in Mechanical Design: Engineering and Biomimetic Approaches. *Designs 2019, Vol. 3, Page 19*, 3(1), 19. <https://doi.org/10.3390/DESIGNS3010019>
- Chahal, S., Kumar, A., & Hussian, F. S. J. (2019). Development of biomimetic electrospun polymeric biomaterials for bone tissue engineering. A review. *Journal of Biomaterials Science, Polymer Edition*, 30(14), 1308–1355. <https://doi.org/10.1080/09205063.2019.1630699>
- dos Reis, M. Q., Marques, E. A. S., Carbas, R. J. C., & da Silva, L. F. M. (2020). Functionally graded adherends in adhesive joints: An overview. *Journal of Advanced Joining Processes*, 2. <https://doi.org/10.1016/J.JAJP.2020.100033>
- El-Galy, I. M., Saleh, B. I., & Ahmed, M. H. (2019). Functionally graded materials classifications and development trends from industrial point of view. *SN Applied Sciences*, 1(11). <https://doi.org/10.1007/S42452-019-1413-4>
- Glassner, A. S. (1989). *An Introduction to ray tracing*. 327.

- Goel, A., & Anand, S. (2019). Design of Functionally Graded Lattice Structures using B-splines for Additive Manufacturing. *Procedia Manufacturing*, 34, 655–665. <https://doi.org/10.1016/J.PROMFG.2019.06.193>
- Grigolato1, L., 2*, Rosso1, S., 3, Meneghello1, R., 3, Concheri1, G., Savio1, 2 and Gianpaolo, & 2. (2019). Heterogeneous objects representation for Additive Manufacturing: a review. *Instant Journal of Mechanical Engineering*, 1(1), 14–23. <https://doi.org/10.36811/IJME.2019.110002>
- Haïat, G., Naili, S., Grimal, Q., Talmant, M., Desceliers, C., & Soize, C. (2009). Influence of a gradient of material properties on ultrasonic wave propagation in cortical bone: Application to axial transmission. *The Journal of the Acoustical Society of America*, 125(6), 4043–4052. <https://doi.org/10.1121/1.3117445>
- Hesselgren, L., Kilian, A., Hornung, O. S., Malek, S., Olsson, K.-G., & Williams, C. J. K. (2018). Advances in Architectural Geometry 2018. *AAG - Advances in Architectural Geometry*, 112–139. <https://research.chalmers.se/en/publication/504188>
- Hughes, J. F., Dam, A. van, McGuire, M., Sklar, D. F., Foley, J. D., Feiner, S. K., & Kurt Akeley. (1996). *Computer Graphics: Principles and Practice* (J. D. Foley, Ed.). Addison-Wesley Professional. <https://books.google.com/books?id=-4ngT05gmAQC&pg=PA557>
- Jones, A., Leary, M., Bateman, S., & Easton, M. (2022). Parametric design and evaluation of TPMS-like cellular solids. *Materials & Design*, 221, 110908. <https://doi.org/10.1016/J.MATDES.2022.110908>
- Jung, Y., & Torquato, S. (2005). Fluid permeabilities of triply periodic minimal surfaces. *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, 72(5 Pt 2). <https://doi.org/10.1103/PHYSREVE.72.056319>
- Karageorgiou, V., & Kaplan, D. (2005). Porosity of 3D biomaterial scaffolds and osteogenesis. *Biomaterials*, 26(27), 5474–5491. <https://doi.org/10.1016/J.BIOMATERIALS.2005.02.002>
- Knoppers, G. E., Gunnink, J. W., van den Hout, J., & van Vliet, W. P. (2004). *The Reality of Functionally Graded Material Products*. <https://doi.org/10.26153/TSW/5706>
- Kou, X. Y., & Tan, S. T. (2007). Heterogeneous object modeling: A review. *Computer-Aided Design*, 39(4), 284–301. <https://doi.org/10.1016/J.CAD.2006.12.007>
- Letov, N., & Zhao, Y. F. (2022). A geometric modelling framework to support the design of heterogeneous lattice structures with non-linearly varying geometry. *Journal of Computational Design and Engineering*, 9(5), 1565–1584. <https://doi.org/10.1093/JCDE/QWAC076>
- Luca Grigolato, Stefano Rosso, Roberto Meneghello, Gianmaria Concheri, & Gianpaolo Savio. (2019). Heterogeneous objects representation for Additive Manufacturing: a review. *Instant Journal of Mechanical Engineering*, 14–23. <https://doi.org/10.36811/ijme.2019.110002>
- Mahamood, R., Akinlabi, E., Shukla, M., & Pityana, S. L. (2012). *Functionally Graded Material: An overview*. 2065. <https://doi.org/10.1016/j.compstruct.2012.02.012>

- Mahamood, R. M., & Akinlabi, E. T. (2015). Laser metal deposition of functionally graded Ti6Al4V/TiC. *Materials & Design, Complete*(84), 402–410. <https://doi.org/10.1016/J.MATDES.2015.06.135>
- Mahmoud, D., & Elbestawi, M. A. (2017a). Lattice Structures and Functionally Graded Materials Applications in Additive Manufacturing of Orthopedic Implants: A Review. *Journal of Manufacturing and Materials Processing 2017, Vol. 1, Page 13, 1*(2), 13. <https://doi.org/10.3390/JMMP1020013>
- Mahmoud, D., & Elbestawi, M. A. (2017b). Lattice Structures and Functionally Graded Materials Applications in Additive Manufacturing of Orthopedic Implants: A Review. *Journal of Manufacturing and Materials Processing 2017, Vol. 1, Page 13, 1*(2), 13. <https://doi.org/10.3390/JMMP1020013>
- Maskery, I., Sturm, L., Aremu, A. O., Panesar, A., Williams, C. B., Tuck, C. J., Wildman, R. D., Ashcroft, I. A., & Hague, R. J. M. (2018). Insights into the mechanical properties of several triply periodic minimal surface lattice structures made by polymer additive manufacturing. *Polymer, 152*, 62–71. <https://doi.org/10.1016/J.POLYMER.2017.11.049>
- Mensch, T. E., Delesky, E. A., Learsch, R. W., Foster, K. E. O., Yeturu, S. K., Srubar, W. v., & Miyake, G. (2021). Mechanical evaluation of 3D printed biomimetic non-Euclidean saddle geometries mimicking the mantis shrimp. *Bioinspiration & Biomimetics, 16*(5), 056002. <https://doi.org/10.1088/1748-3190/AC0A33>
- Miao, X., & Sun, D. (2009). Graded/Gradient Porous Biomaterials. *Materials 2010, Vol. 3, Pages 26-47, 3*(1), 26–47. <https://doi.org/10.3390/MA3010026>
- Murr, L. E. (2018). Additive manufacturing of biomedical devices: an overview. *Materials Technology, 33*(1), 57–70. <https://doi.org/10.1080/10667857.2017.1389052>
- Museth, K. (2013). VDB: High-resolution sparse volumes with dynamic topology. *ACM Transactions on Graphics, 32*(3). <https://doi.org/10.1145/2487228.2487235>
- Novelline, R. A., & Squire, L. F. (1997). *Squire's fundamentals of radiology*. 621.
- Panesar, A., Abdi, M., Hickman, D., & Ashcroft, I. (2018). Strategies for functionally graded lattice structures derived using topology optimisation for Additive Manufacturing. *Additive Manufacturing, 19*, 81–94. <https://doi.org/10.1016/j.addma.2017.11.008>
- Pasko, A., Fryazinov, O., Vilbrandt, T., Fayolle, P. A., & Adzhiev, V. (2011). Procedural function-based modelling of volumetric microstructures. *Graphical Models, 73*(5), 165–181. <https://doi.org/10.1016/J.GMOD.2011.03.001>
- Pattappa, G., Li, Z., Peroglio, M., Wismer, N., Alini, M., & Grad, S. (2012). Diversity of intervertebral disc cells: phenotype and function. *Journal of Anatomy, 221*(6), 480–496. <https://doi.org/10.1111/J.1469-7580.2012.01521.X>
- Rossen, S. van, & Baranowski, M. (2011). Game Development Tools. In *Game Development Tools*. CRC Press. <https://doi.org/10.5860/choice.49-1514>
- Savio, G., Meneghello, R., Rosso, S., & Concheri, G. (2019). 3D model representation and data exchange for additive manufacturing. *Lecture Notes in Mechanical Engineering, 412–421*. https://doi.org/10.1007/978-3-030-12346-8_40/COVER

- Savio, G., Rosso, S., Meneghello, R., & Concheri, G. (2018). Geometric modeling of cellular materials for additive manufacturing in biomedical field: A review. *Applied Bionics and Biomechanics*, 2018. <https://doi.org/10.1155/2018/1654782>
- Shapiro, V. (2002). Solid Modeling. *Handbook of Computer Aided Geometric Design*, 473–518. <https://doi.org/10.1016/B978-044451104-1/50021-6>
- Sharma, G. K., & Gurumoorthy, B. (2019). Modelling multiply connected heterogeneous objects using mixed-dimensional material reference features. *Journal of Computational Design and Engineering*, 6(3), 337–347. <https://doi.org/10.1016/J.JCDE.2018.11.003>
- Shin, K. H., & Dutta, D. (2001). Constructive Representation of Heterogeneous Objects. *Journal of Computing and Information Science in Engineering*, 1(3), 205–217. <https://doi.org/10.1115/1.1403448>
- Simske, S. J., Ayers, R. A., & Bateman, T. A. (1997). Porous Materials for Bone Engineering. *Materials Science Forum*, 250, 151–182. <https://doi.org/10.4028/WWW.SCIENTIFIC.NET/MSF.250.151>
- Stieler, D., Schwinn, T., & Menges, A. (2022). Volumetric intersections: Modularization approaches for freeform prefab concrete construction. *Civil Engineering Design*, 4(1–3), 3–13. <https://doi.org/10.1002/CEND.202100047>
- Tamburrino, F., Graziosi, S., & Bordegoni, M. (2018). The design process of additively manufactured mesoscale lattice structures: A review. *Journal of Computing and Information Science in Engineering*, 18(4). <https://doi.org/10.1115/1.4040131/369017>
- Thieme, M., Wieters, K. P., Bergner, F., Scharnweber, D., Worch, H., Ndop, J., Kim, T. J., & Grill, W. (2001). Titanium powder sintering for preparation of a porous functionally graded material destined for orthopaedic implants. *Journal of Materials Science. Materials in Medicine*, 12(3), 225–231. <https://doi.org/10.1023/A:1008958914818>
- van Bael, S., Kerckhofs, G., Moesen, M., Pyka, G., Schrooten, J., & Kruth, J. P. (2011). Micro-CT-based improvement of geometrical and mechanical controllability of selective laser melted Ti6Al4V porous structures. *Materials Science and Engineering: A*, 528(24), 7423–7431. <https://doi.org/10.1016/J.MSEA.2011.06.045>
- Volume Graphics. (2000). *Volume Graphics*. <https://doi.org/10.1007/978-1-4471-0737-8>
- Volumetric Modelling - dbt*. (n.d.). Retrieved November 30, 2022, from <https://dbt.arch.ethz.ch/research-stream/volumetric-modelling/>
- Warnke, P. H., Douglas, T., Wollny, P., Sherry, E., Steiner, M., Galonska, S., Becker, S. T., Springer, I. N., Wiltfang, J., & Sivananthan, S. (2008). Rapid Prototyping: Porous Titanium Alloy Scaffolds Produced by Selective Laser Melting for Bone Tissue Engineering. *https://Home.Liebertpub.Com/Tec*, 15(2), 115–124. <https://doi.org/10.1089/TEN.TEC.2008.0288>
- Xiao, L., Xu, X., Song, W., & Hu, M. (2020). A Multi-Cell Hybrid Approach to Elevate the Energy Absorption of Micro-Lattice Materials. *Materials 2020, Vol. 13, Page 4083*, 13(18), 4083. <https://doi.org/10.3390/MA13184083>
- Xiong, Y. Z., Gao, R. N., Zhang, H., Dong, L. L., Li, J. T., & Li, X. (2020). Rationally designed functionally graded porous Ti6Al4V scaffolds with high strength and toughness built via

selective laser melting for load-bearing orthopedic applications. *Journal of the Mechanical Behavior of Biomedical Materials*, 104, 103673. <https://doi.org/10.1016/J.JMBBM.2020.103673>

Zhao, M., Zhang, D. Z., Li, Z., Zhang, T., Zhou, H., & Ren, Z. (2022). Design, mechanical properties, and optimization of BCC lattice structures with taper struts. *Composite Structures*, 295, 115830. <https://doi.org/10.1016/J.COMPSTRUCT.2022.115830>

Zhao, R., Yang, R., Cooper, P. R., Khurshid, Z., Shavandi, A., & Ratnayake, J. (2021). Bone grafts and substitutes in dentistry: A review of current trends and developments. *Molecules*, 26(10). <https://doi.org/10.3390/MOLECULES26103007>

Zhao, Y. F., Sun, S., Velivela, P. T., & Letov, N. (2021). Challenges and Opportunities in Geometric Modeling of Complex Bio-Inspired Three-Dimensional Objects Designed for Additive Manufacturing. *Journal of Mechanical Design, Transactions of the ASME*, 143(12). <https://doi.org/10.1115/1.4051720/1114474>

