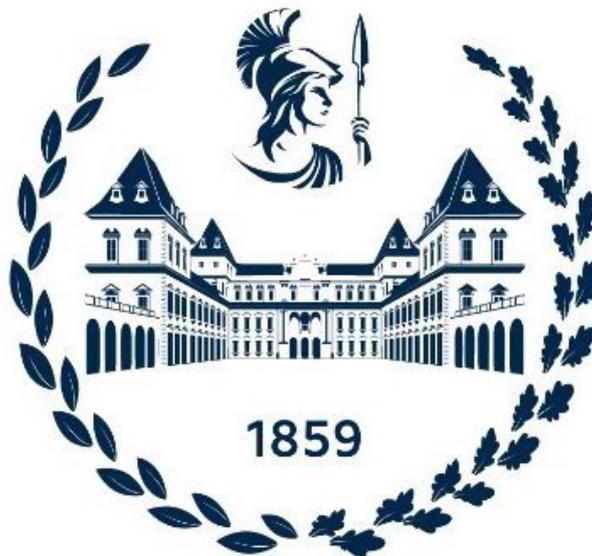


POLITECNICO DI TORINO

Laurea Magistrale in Ingegneria Meccanica

Progettazione Meccanica



Tesi di Laurea Magistrale

Analisi e Ottimizzazione di Strutture in Metallo per Additive Manufacturing

Supervisor:

Prof.ssa Francesca Maria Curà

Prof. Enrico D'Amato

Ing. Mattia Vichi

Candidato:

Antonio Gallo

306025

Anno Accademico 2023/2024

*A Nonno.
A te che oggi,
in un modo o nell'altro,
sei comunque qui con me.*

Abstract

La crescente esigenza di migliorare le prestazioni in settori come il motorsport o l'aerospazio, ha spinto la progettazione strutturale verso soluzioni sempre più estreme. Frequentemente, infatti, l'obiettivo di progetto consiste nel minimizzare il peso al fine di diminuire i carichi, aumentare le velocità o ottenere un risparmio di carburante, a discapito del costo finale del prodotto. Questo trend ha portato all'impiego di materiali sempre più leggeri e performanti, oltre che alla ricerca di soluzioni di design ottimali in cui l'impiego del materiale sia ridotto al minimo necessario. Di conseguenza, sono state sviluppate tecniche computazionali avanzate che permettono di ottenere strutture non convenzionali, a valle del processo di ottimizzazione, in cui la distribuzione del materiale segue il principio di "impiegare materiale solo dove è necessario" [1]. Per sfruttare il potenziale di tali tecniche di ottimizzazione, è indispensabile l'uso di tecnologie di fabbricazione additiva, le quali consentono di realizzare componenti con geometrie complesse che sarebbero estremamente costosi, difficili o addirittura impossibili da produrre con metodi di manifattura tradizionali.

L'obiettivo di questa tesi di Laurea Magistrale, dal titolo *Analisi e Ottimizzazione di Strutture in Metallo per Additive Manufacturing* consiste nella gestione di modelli di grandi dimensioni attraverso l'uso di algoritmi, oltre che nell'ottimizzazione di strutture attraverso particolari tecniche. Lo studio è quindi articolato in tre fasi principali, di cui le prime due riguardano l'ottimizzazione delle componenti metalliche di un telaio ibrido metallo-composito per autoveicoli.

Nella prima fase viene eseguita un'ottimizzazione topologica classica che mira a individuare la distribuzione ottimale del materiale all'interno della struttura, mantenendo solo gli elementi necessari per resistere ai carichi applicati, eliminando il materiale superfluo. Da essa si ricava un modello di riferimento, rispetto al quale si valutano i risultati delle fasi successive.

Nella seconda fase, invece, si esegue un'ottimizzazione topologica lattice. Questa tecnica consente di migliorare ulteriormente le prestazioni della struttura, senza comprometterne la funzionalità, sfruttando al meglio sia i vantaggi offerti dalla tecnica di ottimizzazione

topologica, sia quelli legati alle tecniche di produzione additiva.

La terza fase, sviluppata parallelamente all'ottimizzazione lattice, prevede la creazione di un algoritmo per la gestione del modello e dei suoi elementi, riducendo al minimo l'interazione con l'interfaccia grafica del software. Questo ulteriore sviluppo si è reso necessario poiché, per modelli di grandi dimensioni, la richiesta di risorse computazionali da parte dei software aumenta significativamente, causando un incremento dei tempi di elaborazione del file e di esecuzione delle operazioni. Ultimate le tre fasi principali dello studio, si confrontano i risultati ottenuti dall'ottimizzazione topologica classica e dall'ottimizzazione lattice, al fine di evidenziare vantaggi e svantaggi di entrambe le tecniche.

Indice

1	Stato dell'arte	1
1.1	Il Ruolo della Progettazione nel Settore Ingegneristico	1
1.1.1	Progettazione Assistita	3
1.2	Metodo degli Elementi Finiti	4
1.2.1	Modellazione / Pre-Processing	5
1.2.2	Soluzione / Analisi	8
1.2.3	Visualizzazione dei Risultati / Post-Processing	9
1.3	Processo di Ottimizzazione	10
1.3.1	Tipologie di Ottimizzazione Strutturale	11
1.3.2	L'Algoritmo di Ottimizzazione	13
1.4	Fabbricazione Additiva	17
1.4.1	Principi e Caratteristiche	17
1.4.2	Fasi della Fabbricazione Additiva	18
1.4.3	Vantaggi e Svantaggi	20
2	Analisi FEM preliminare	24
2.1	Preparazione del Modello	24
2.1.1	Descrizione del Modello Importato	24
2.1.2	Creazione della Mesh	26
2.1.3	Definizione dei Collegamenti	29
2.1.4	Definizione dei Materiali e delle Proprietà	31
2.1.5	Condizioni al Contorno	34
2.2	Analisi lineare statica	37
2.2.1	Definizione degli Obiettivi	42
3	Ottimizzazione Topologica	44
3.1	Definizione di Design e Non-Design Space	44
3.2	Definizione dei Target e dei Controlli dell'Ottimizzazione	46

3.2.1	Response, Obiettivo e Vincoli	47
3.2.2	Variabili di Design	50
3.2.3	Controlli di Ottimizzazione	51
3.3	Ottimizzazione Parametrica della Struttura	53
3.3.1	Fasi di Elaborazione dei Risultati	53
3.3.2	Risultati	55
3.4	Risultati dell'Ottimizzazione Topologica	59
4	Ottimizzazione di Strutture Lattice	61
4.1	Basi per l'Ottimizzazione Lattice	61
4.1.1	Phase I	62
4.1.2	Elaborazione del Modello tra Phase I e Phase II	67
4.1.3	Phase II	70
4.1.4	Gestione dei File	71
5	Gestione del Modello in Altair Compose	74
5.1	Basi per la Creazione dello Script	74
5.1.1	Introduzione al Software	74
5.1.2	Regole di Formattazione del Testo	74
5.2	Processo di Modifica della Struttura	76
5.2.1	Fasi del Processo	76
5.2.2	Creazione degli Algoritmi	76
5.2.3	Effetto dei Codici sulla Struttura	84
6	Risultati dell'Ottimizzazione Lattice	87
6.1	Considerazioni sui Parametri di Controllo	87
6.2	Risultati Phase I	88
6.3	Risultati Phase II	91
7	Conclusioni	96

Elenco delle figure

1.1	Fasi di sviluppo di un nuovo prodotto [3].	1
1.2	Ciclo iterativo del processo di ottimizzazione.	2
1.3	Fasi dell'analisi FEM.	5
1.4	Comandi da interfaccia per la pulizia della geometria.	6
1.5	Effetto degli errori di importazione della geometria sulla qualità finale della mesh [8].	6
1.6	Elementi trave 1D, guscio 2D e solido 3D.	7
1.7	Curve tensione-deformazione per materiale con comportamento lineare e non lineare.	9
1.8	Esempio di ottimizzazione topologica.	11
1.9	Esempio di ottimizzazione topografica.	11
1.10	Esempio di ottimizzazione dimensionale	12
1.11	Esempio di ottimizzazione di forma con variazioni alternative predefinite.	12
1.12	Esempio di ottimizzazione lattice.	13
1.13	Funzione con diversi minimi locali.	14
1.14	Rappresentazione del metodo del gradiente di Newton.	15
1.15	Metodo del Gradiente.	16
1.16	Confronto tra due diverse tecniche di fabbricazione applicate al medesimo componente.	17
1.17	Fasi della fabbricazione digitale del componente e del processo produttivo.	19
1.18	Flusso di lavoro dell'intero processo di produzione tramite fabbricazione additiva.	19
1.19	Costi e guadagni di un componente ottenuto tramite AM, rispetto ad uno tradizionale.	22
1.20	Confronto dei costi di produzione.	22
2.1	Modello di riferimento.	25
2.2	Elemento solido isoparametrico a otto nodi	26
2.3	Modalità di creazione della mesh voxel	27

2.4	Effetto delle entità geometriche sul risultato finale della mesh voxel.	27
2.5	Vista anteriore del front frame prima e dopo la creazione della mesh.	28
2.6	Vista inclinata del front frame prima e dopo la creazione della mesh.	28
2.7	Vista anteriore del rear frame prima e dopo la creazione della mesh.	29
2.8	Vista inclinata del rear frame prima e dopo la creazione della mesh.	29
2.9	Dettaglio definizione RBE2.	30
2.10	Card di definizione dell'elemento RBE2.	30
2.11	Particolare degli RBE3, con la card di definizione del collegamento.	31
2.12	Trave a sezione circolare.	32
2.13	Grandezze caratteristiche del materiale composito.	33
2.14	Linee di riferimento per l'elemento guscio.	33
2.15	Finestra dedicata alla definizione di un carico statico.	35
2.16	Spostamento in direzione X.	39
2.17	Spostamento in direzione Y.	40
2.18	Spostamento in direzione Z.	41
3.1	Vista laterale di front e rear frame.	45
3.2	Elementi appartenenti al design space.	46
3.3	Finestre per la definizione delle prime due responses.	47
3.4	Schermata per la definizione dell'obiettivo di ottimizzazione.	48
3.5	Schermata per la definizione dei vincoli di ottimizzazione.	49
3.6	Esempio dell'effetto del parametro DISCRETE sulle caratteristiche della struttura.	52
3.7	Effetto del parametro MINDIM sulle caratteristiche della struttura.	52
3.8	Caption	54
3.9	Andamento della massa in funzione della percentuale di filtraggio degli elementi.	54
3.10	Dettaglio delle strutture ottenute dalla conversione delle ottimizzazioni topologiche per la seconda (a sinistra) e la terza simulazione (a destra). . .	56
3.11	Dettaglio del front frame ottenuto dalla conversione delle ottimizzazioni topologiche per diversi valori di MINDIM.	58
3.12	Dettaglio del rear frame ottenuto dalla conversione delle ottimizzazioni topologiche per diversi valori di MINDIM. MINDIM=30 a sinistra; MIN- DIM=40 a destra.	59
4.1	Differenze tra la fase 1 dell'ottimizzazioneelattice e l'ottimizzazione topologica. .	61
4.2	Esempio di card per la definizione della proprietà di una beam.	62

4.3	Esempio di card per la definizione della variabile di design in un nodo. . . .	63
4.4	Esempio di card per la definizione delle DVPREL.	64
4.5	Riferimenti di una specifica variabile di design.	64
4.6	Tipologie di celle lattice applicabili all'elemento esaedrico.	65
4.7	LT per elementi diversi dalla cella esagonale.	65
4.8	Confronto tra le diverse prestazioni di rigidezza riscontrate in strutture con porosità LOW, MED e HIGH.	67
4.9	Strumenti per il cleanup degli elementi vuoti o inutilizzati del modello. . .	68
4.10	Duplicazione del MAT1.	69
4.11	Riassegnazione del materiale agli elementi solidi.	69
4.12	Funzioni di penalizzazione implementate nella fase di Lattice Sizing+. . . .	71
4.13	Esempio di file di input per OptiStruct [11].	73
5.1	Esempio di riga a formato fisso.	75
5.2	Definizione delle dimensioni dei vettori e creazione delle struct.	77
5.3	Funzione helper per la lettura delle coordinate dei nodi.	78
5.4	Ciclo di lettura e memorizzazione delle variabili di interesse.	79
5.5	Funzione helper per la scrittura delle righe relative a CBEAM, seguita dal ciclo per l'individuazione delle beams da specchiare.	80
5.6	Particolare degli RBE3 del lato sinistro anteriore, prima e dopo la cancel- lazione degli RBE3 superflui.	82
5.7	Processo di selezione delle entità da mantenere nel modello.	84
5.8	Processo di specchiatura degli elementi.	85
5.9	Effetto dell'algoritmo di specchiatura sulle proprietà delle beams.	86
6.1	Dettaglio struttura lattice del front frame, al termine della Phase I.	89
6.2	Dettaglio struttura lattice del rear frame, al termine della Phase I.	90
6.3	Particolare delle beams per il front frame.	94
6.4	Particolare delle beams per il rear frame.	95
7.1	Discreteness tables.	98

Elenco delle tabelle

2.1	Direzione e punto di applicazione della forza associati al relativo subcase. . .	36
2.2	Spostamento misurato, ammissibile e relative rigidezze in relazione al Subcase.	43
3.1	Effetto del parametro DOPTPRM,DISCRETE sulle caratteristiche del componente.	55
3.2	Effetto del parametro DOPTPRM,MINDIM sulle caratteristiche del componente.	57
3.3	Response ottenute dall'ottimizzazione topologica.	60
5.1	Riga per la definizione dei dati relativi ad un elemento CBEAM (ID=589967). . .	78
5.2	Riga per la definizioe dei dati relativi ad un elemento GRID (ID=103302). . .	78
6.1	Effetto del parametro POROSITY sulla massa ottenuta dalla topologica. . .	87
6.2	Risultati ottenuti dalle varie fasi dell'ottimizzazione lattice.	91
6.3	Response ottenute dall'analisi relativa alla struttura convertita a valle della Phase I.	92
6.4	Response ottenute dall'ottimizzazione lattice.	93
7.1	Tabella riassuntiva dei risultati ottenuti.	96
7.2	Confronto del valore delle response e dei vincoli attivi per le due diverse tecniche di ottimizzazione.	97

Acronimi

AM:Additive Manufacturing
ASCII:American Standard Code for Information Interchange
CAD:Computer-Aided Design
CAE:Computer-Aided Engineering
CFD:Computational Fluid Dynamics
DED:Direct Energy Deposition
EBAM:Electron Beam Additive Manufacturing
EBM:Electron Beam Melting
FA:Fabbricazione Additiva
FEM:Finite Elements Method
L-PBF:Laser-Powder Bed Fusion
LB:Lower Bound
MOO:Multi-Objective Optimization
NVH:Noise, Vibration and Harshness
PA:Progettazione Assistita
PDE:Partial Derivative Equations
SIMP:Solid Isotropic Material with Penalization
STL:Standard Triangle Language
UB:Upper Bound

1. Stato dell'arte

1.1 Il Ruolo della Progettazione nel Settore Ingegneristico

Con il termine “Progettazione” si intende quell’insieme di processi mirati a trasformare un’idea in un prodotto finito, soddisfacendo al contempo determinati requisiti tecnici, funzionali ed economici di un determinato prodotto [2]. Le fasi che costituiscono lo sviluppo di un nuovo prodotto possono essere raccolte in alcune categorie principali:



Figura 1.1: Fasi di sviluppo di un nuovo prodotto [3].

- **Idea** Lo sviluppo di un’idea può basarsi sia sul miglioramento di una tecnologia esistente sia sull’introduzione di una nuova. Nelle grandi aziende, spesso esistono interi reparti dedicati esclusivamente allo sviluppo di idee innovative.
- **Ricerca e Sviluppo** Ricerca di mercato e di prodotto; fondamentale per verificare la fattibilità dell’investimento in relazione alle prospettive di guadagno. Coinvolge l’analisi delle esigenze del mercato e delle potenzialità del prodotto. Una volta individuati i fabbisogni del mercato nella fase di ricerca, il prodotto viene sviluppato per soddisfare tali esigenze. Questo include la progettazione dettagliata, la prototipazione, l’ottimizzazione e la pianificazione della produzione. In questo studio si fa riferimento proprio alla fase di sviluppo del prodotto, con particolare attenzione all’ottimizzazione dello stesso.

- **Testing** Nella fase di testing , il prototipo viene testato per verificare e ottimizzare le sue caratteristiche finali. Il testing include prove di funzionalità, qualità e sicurezza per garantire che il prodotto soddisfi gli standard richiesti.
- **Introduzione e Analisi** Dopo l'introduzione del prodotto, viene effettuata un'analisi della risposta dei consumatori e della performance del prodotto sul mercato. Queste informazioni sono utilizzate per apportare eventuali miglioramenti al prodotto stesso.

Nella fase di ricerca e sviluppo è inclusa l'ottimizzazione, la quale costituisce l'evoluzione del prodotto verso la sua forma finale, e può essere riassunta nei seguenti passaggi iterativi:

1. Creazione del design.
2. Analisi del modello.
3. Valutazione dei risultati.
4. Se il design ottimale è raggiunto interrompo il ciclo di ottimizzazione, altrimenti effettuo delle modifiche e ripeto il ciclo dal punto 2.

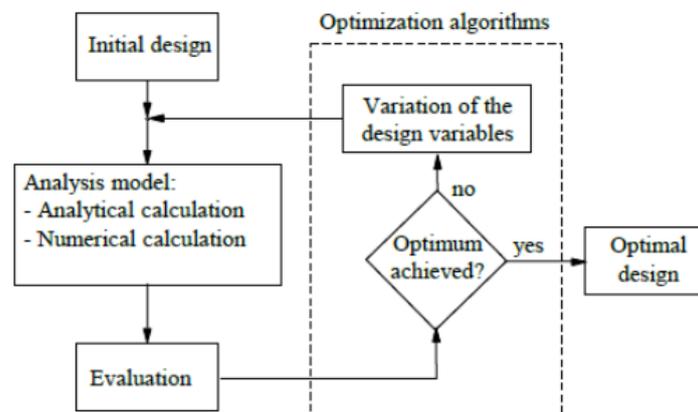


Figura 1.2: Ciclo iterativo del processo di ottimizzazione.

Dunque, l'ottimizzazione di un determinato design è un processo di tipo iterativo, dove in ciascuna iterazione viene modificato uno o più parametri di controllo al fine di soddisfare le specifiche del progetto. Tuttavia, la definizione di design soddisfacente risulta spesso complicata, poiché dipende dal settore di riferimento [4]. Ad esempio, nel settore automobilistico pubblico, l'enfasi è posta principalmente sulla sicurezza, sul comfort e sull'economicità, mentre nel settore automobilistico sportivo, la priorità è data alla prestazione, a scapito dei costi. Ancora, nel caso dei veicoli di lusso, un design soddisfacente è rappresentato da un compromesso tra le esigenze dei settori pubblico e sportivo.

1.1.1 Progettazione Assistita

A partire dagli anni '60, con l'introduzione dei primi sistemi Computer-Aided Design (CAD), e dagli anni '70, con l'avvento dei sistemi Computer-Aided Engineering (CAE), è stato introdotto il concetto di progettazione assistita. Questo approccio fa uso di codici di calcolo come supporto alla progettazione, sfruttando le capacità computazionali degli elaboratori per eseguire operazioni matematiche complesse e numerose in tempi ridotti. Il codice di calcolo richiede lo sviluppo di un modello matematico del sistema da esaminare, in grado di simularne il comportamento in determinate condizioni. Tale modello varia in base al fenomeno in esame, che può essere di natura meccanica, termica, elettrica, ecc. Solitamente, il codice di calcolo permette la scrittura del modello matematico in modo "trasparente", così che l'utente debba semplicemente realizzare il modello attraverso l'interfaccia del software.

Uno dei grandi vantaggi della PA risiede nella semplificazione del lavoro del progettista, poiché quest'ultimo non deve gestire manualmente numerose equazioni matematiche per la risoluzione del problema. Un ulteriore fattore che ha contribuito alla diffusione massiva della PA è l'importante risparmio sui costi di progettazione. Essa, infatti, consente di simulare il comportamento del prodotto prima della sua produzione, permettendo così di identificare e risolvere eventuali problemi nelle fasi iniziali del progetto, riducendo significativamente i costi associati alla prototipazione fisica. Tuttavia, esiste il rischio che l'utente perda di vista la fisica del problema, non riuscendo a giudicare criticamente la validità dei risultati ottenuti. Infatti, il fatto che un codice di calcolo fornisca un risultato non garantisce che esso sia corretto; è compito dell'ingegnere individuare gli errori che portano a incongruenze tra il modello virtuale e quello fisico.

Come accennato sopra, l'analisi numerica di un modello è intrinsecamente soggetta a errori numerici dovuti a vari fattori:

- Limitazioni nel modello o nel codice.
- Errori nei dati di partenza.
- Errori numerici derivanti dai limiti nell'approssimazione dei risultati.
- Errori commessi dall'utente a causa di cattiva interpretazione dei dati, errori di modellazione o semplici sviste.

Escludendo quelli legati all'utente, in ogni modello si riscontra sempre la presenza di tutti gli errori appena elencati. Tuttavia, gli errori di approssimazione possono essere limitati, anche se non completamente eliminati, usando semplici accorgimenti. Infatti, per poter migliorare l'approssimazione della risposta, facendola così convergere al risultato "esatto",

si può aumentare il grado del polinomio delle funzioni di forma oppure, in alternativa, si aumenta il numero dei nodi utilizzati per la discretizzazione del continuo [5].

Nella pratica, il grado delle funzioni di forma si incrementa scegliendo degli elementi che a parità del numero di nodi presentano delle funzioni di forma di grado più alto come, ad esempio, nel caso si implementino gli elementi con modi incompatibili. Questi ultimi consistono in una correzione delle funzioni di forma considerando dei termini mancanti, di grado superiore a quelli presenti nella funzione di forma dell'elemento normale [5]. Per quanto riguarda, invece, la regolazione sul numero di nodi presenti nel modello, esistono due alternative: la prima consiste nel scegliere degli elementi analoghi ma con un numero di nodi superiore, come ad esempio l'elemento solido esaedrico a 16 nodi, piuttosto che quello a 8 nodi; il secondo consiste nel modificare la dimensione caratteristica della mesh al fine di introdurre un maggior numero di elementi (e quindi di nodi) all'interno del volume di creazione della mesh.

Poiché vi è la possibilità che nel modello vi sia un elevato numero di errori, per garantire l'accuratezza dei risultati forniti è necessario effettuare la validazione sperimentale del modello, sottoponendo il componente reale a condizioni di carico specifiche. Queste condizioni di carico sono limitate ad alcuni aspetti di funzionamento o a specifici parametri di riferimento scelti in modo tale da essere sufficienti a confermare i risultati ottenuti dal codice. Una volta validato, il modello può essere impiegato per simulare ogni altra condizione di interesse.

1.2 Metodo degli Elementi Finiti

Il Finite Elements Method (FEM) è una tecnica di risoluzione numerica utilizzata per determinare soluzioni approssimate per le equazioni differenziali alle derivate parziali (PDE) su un dominio finito [6]. Questo metodo analizza delle funzioni che rappresentano l'approssimazione del campo di spostamenti all'interno dell'elemento. Tali funzioni possono essere ricavate sia con metodi energetici (ad es. con il principio dei lavori virtuali o con l'equazione di Lagrange) che con metodi variazionali (ad es. con il metodo di Galerkin o con il metodo di Rayleigh-Ritz).

Le equazioni di approssimazione, tuttavia, non possono essere applicate al corpo continuo poiché l'errore della soluzione sarebbe elevato. È dunque necessario effettuare una discretizzazione del continuo facendo uso di elementi delimitati da nodi, segmenti o facce, a seconda che essi siano rispettivamente mono, bi o tri-dimensionali. A questo punto si ricavano i valori degli spostamenti ai nodi, per poi interpolare i risultati sull'elemento.

In base a quanto appena affermato, si evince la necessità di una fase preliminare di preparazione del modello. A tal proposito, si introducono gli step principali di un'analisi FEM, i quali sono divisi in tre fasi principali come illustrato in figura 1.3: pre-processing, analisi e post-processing [7]. Per ciascuna fase del processo sono riportati alcuni tra i software della suite Altair HyperWorks 2023.1 utilizzati in questo studio. Inoltre, sulla destra sono elencate le operazioni fondamentali per la modellazione e la visualizzazione dei risultati.

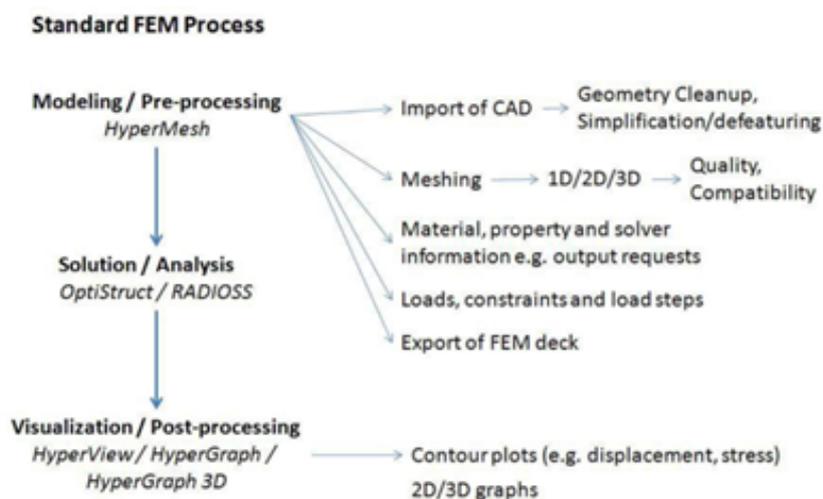


Figura 1.3: Fasi dell'analisi FEM.

1.2.1 Modellazione / Pre-Processing

Dalla fase di progettazione iniziale, non trattata in questa tesi, si importa il file CAD contenente le informazioni sulla geometria del componente. Come osservabile in figura 1.3, in questa fase iniziale si svolgono alcune operazioni finalizzate alla definizione dei parametri caratteristici del modello. La fase di modellazione è la più complessa e critica delle tre, poiché è proprio in questa fase che possono verificarsi gli errori di modellazione e quelli legati all'immissione dei dati di partenza.

Importazione del file CAD Durante l'importazione del CAD, possono verificarsi errori che portano a superfici non rifinite, mancanti o duplicate, bordi liberi o superfici la cui forma non permette la creazione di una mesh corretta nelle fasi successive. È quindi necessario eseguire alcune operazioni che permettano, ove possibile, di evitare l'insorgere di tali problemi. Una di queste operazioni è la pulizia della geometria. A tal fine, HyperMesh mette a disposizione lo strumento Surface Repair, che include le funzionalità illustrate in figura 1.4.

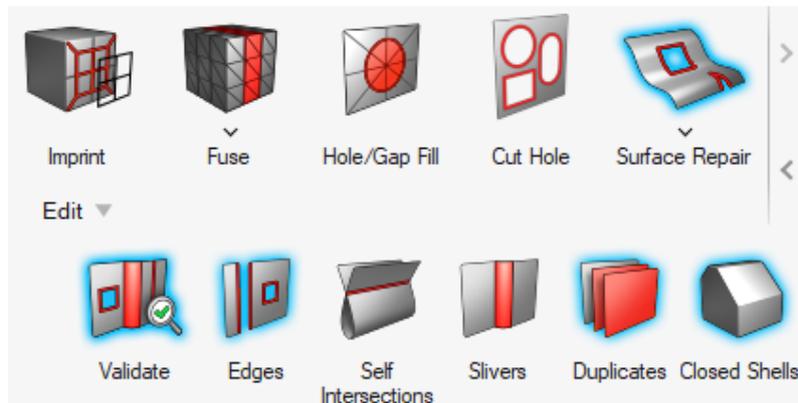


Figura 1.4: Comandi da interfaccia per la pulizia della geometria.

Ognuno di questi strumenti è associato a metodi automatici di riparazione dell'errore, proposti dal software in funzione della problematica riscontrata. Gli stessi metodi possono essere applicati tramite gli strumenti di editing manuale presenti nel pannello "Topology", nel caso in cui la riparazione automatica non sia soddisfacente.

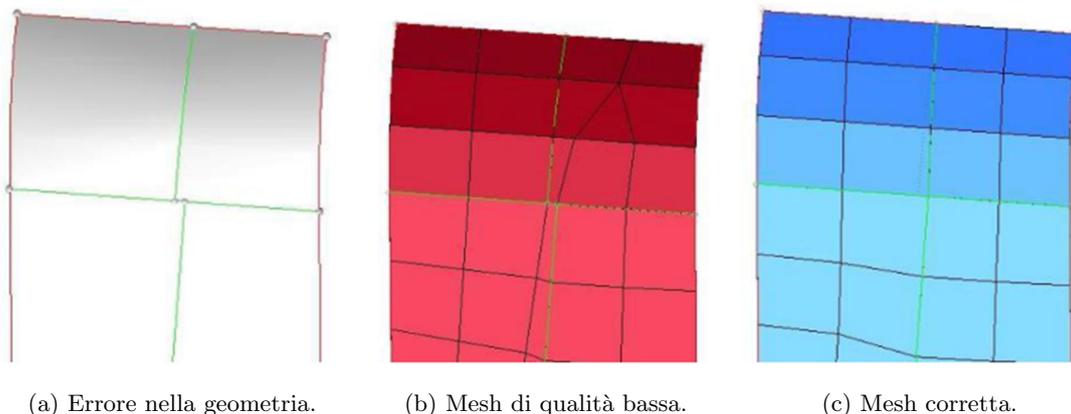


Figura 1.5: Effetto degli errori di importazione della geometria sulla qualità finale della mesh [8].

Un esempio di errore di geometria è riportato in figura 1.5a, dove è rappresentata la geometria importata; in essa si può notare un disallineamento laterale dei bordi, che si ripercuote sulla qualità dei passaggi successivi. Infatti, ciò comporta la generazione di una mesh di scarsa qualità come quella riportata in figura 1.5b. A destra, invece, è visibile il risultato ottenuto a valle della pulizia della geometria. Unendo i due nodi centrali, infatti, si osserva una mesh priva di elementi distorti e, dunque, qualitativamente migliore.

Un ulteriore passo riguarda l'eliminazione di tutte quelle informazioni che complicano il modello senza influire significativamente sul risultato dell'analisi. Esempi di tali informazioni sono flettature, raccordi, piccoli fori e loghi aziendali presenti nel CAD. Per effettuare queste operazioni viene utilizzato un apposito strumento presente nell'interfaccia

grafica di HyperMesh: (*Defeature* \rightarrow *Topology*).

Mesh Una volta sistemata la geometria, si procede alla realizzazione della mesh. Questa può essere costituita da elementi beam (1D), shell (2D), solidi (3D) o da un mix di questi, a seconda del tipo di struttura che si intende realizzare. In questa fase è fondamentale prestare particolare attenzione a qualità, connettività e alle normali degli elementi, tutti indicatori della corretta creazione della mesh; la presenza all'interno della mesh di elementi eccessivamente distorti, infatti, può costituire fonte di errore nel calcolo della soluzione della PDE sull'elemento stesso. Complessivamente, quindi, minore sarà la qualità della mesh, maggiore sarà l'errore nel calcolo del risultato finale. È inoltre necessario scegliere un numero di elementi sufficiente a ridurre l'errore di approssimazione, come accennato nel paragrafo 1.1.1, ma non così elevato da causare tempi di calcolo eccessivi. Questa regolazione può essere effettuata modificando la dimensione caratteristica dell'elemento stesso: aumentando questo valore, a parità di volume di controllo della mesh, si riduce il numero complessivo di elementi generati.

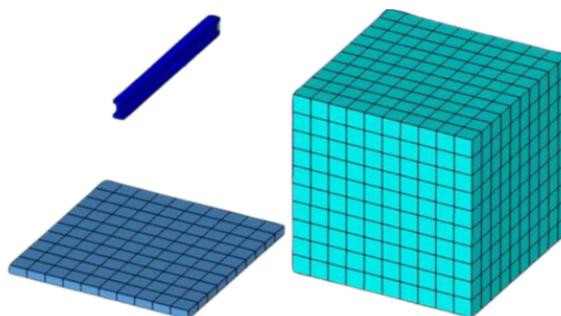


Figura 1.6: Elementi trave 1D, guscio 2D e solido 3D.

Materiali, Proprietà, Condizioni al Contorno e Informazioni sugli Output Ultimata la mesh si assegnano le proprietà e il materiale ad ogni elemento. Il materiale definisce quantitativamente le caratteristiche fisiche, mentre la proprietà definisce le caratteristiche matematiche dell'elemento.

Successivamente si applicano carichi e vincoli che simulano, quanto più possibile, le condizioni di applicazione reali del componente. La definizione delle condizioni di carico rilevanti è affidata alle normative in vigore o, talvolta, all'esperienza dell'ingegnere. Inoltre, è possibile analizzare e ottimizzare il componente in modo che esso possa essere sottoposto a diverse condizioni di carico in contemporanea.

Infine, in questa fase vengono definite informazioni aggiuntive come il tipo di analisi da eseguire, quale solutore utilizzare, quali risultati esportare o quali file generare come out-

put. Tutte le informazioni appena definite vengono esportate da HyperMesh in un file ".fem". Si tratta di un file di tipo ASCII basato sulla sintassi utilizzata dallo specifico solutore, utilizzato da quest'ultimo in fase di analisi.

1.2.2 Soluzione / Analisi

A valle della definizione dei parametri del modello e delle opzioni associate al solutore si svolge la fase di analisi. Quest'ultima consiste in un processo completamente automatizzato, la cui interfaccia con l'utente permette di monitorarne lo stato di avanzamento. In caso di errore nella definizione del modello, ove previsto, il software effettua delle scelte di default; in alternativa segnala l'interruzione del processo di analisi riportando le informazioni sull'errore riscontrato.

Esistono diverse tipologie di analisi realizzabili tramite i software forniti da Altair:

- Analisi lineare statica
- Analisi statica non-lineare
- Analisi dinamica
- Analisi del buckling
- Analisi termica
- Analisi di fatica
- Ottimizzazione
- Analisi CFD
- Analisi dell'impatto
- Analisi NVH

Analisi Lineare Statica

Con il termine lineare si fa riferimento a tutte quelle analisi in cui tensione e deformazione sono linearmente dipendenti (figura 1.7). La relazione che lega queste due grandezze è del tipo:

$$\sigma = E\epsilon \quad (1.1)$$

Dove σ è il tensore delle tensioni, E è, almeno per quanto riguarda l'analisi statica lineare, una costante del materiale conosciuta come modulo di Young ed ϵ è il tensore delle deformazioni. Nella realtà, questa assunzione è valida solo per bassi valori di deformazione, infatti, con l'aumentare di quest'ultima e con il conseguente insorgere di fenomeni plastici,

la risposta del materiale diverge dal comportamento lineare. Un esempio di curva sperimentale sul piano tensione-deformazione è riportata in figura 1.7; in essa si osserva che il materiale reale va incontro a fenomeni di plasticizzazione se supera determinati valori di deformazione.

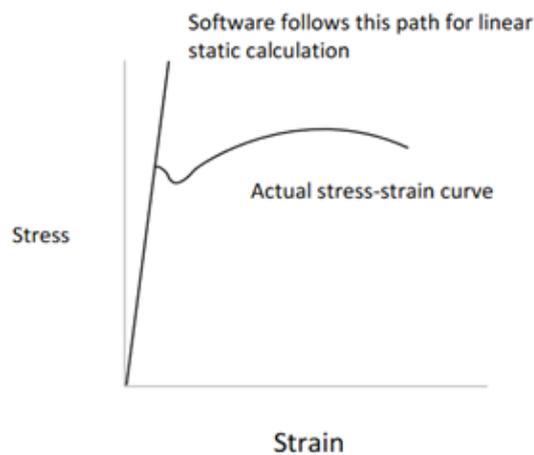


Figura 1.7: Curve tensione-deformazione per materiale con comportamento lineare e non lineare.

Il termine legato alla statica, invece, implica che il carico rimanga costante nel tempo. Inoltre, il modello deve soddisfare le condizioni di equilibrio delle forze e dei momenti in ogni singolo nodo e in ogni istante di tempo, in modo tale da renderne la sommatoria complessiva pari a zero. L'equazione di base per la soluzione di strutture sotto carico statico è:

$$Ku = P \quad (1.2)$$

Dove K è la matrice di rigidità della struttura, il vettore u rappresenta il vettore degli spostamenti e P è il vettore dei carichi. Si tratta di una equazione di equilibrio la cui soluzione viene calcolata in maniera diretta per problemi di tipo lineare e in maniera iterativa per problemi non lineari. Con essa si calcolano gli spostamenti nodali degli elementi. Ricavati gli spostamenti nodali, il software procede calcolando gli sforzi usando le equazioni costitutive del materiale (equazione 1.1).

1.2.3 Visualizzazione dei Risultati / Post-Processing

Una volta ottenuta la soluzione dell'analisi, i risultati possono essere trattati utilizzando software dedicati come HyperView, che consente di osservare gli andamenti delle variabili ingegneristiche nel modello, o HyperGraph, che permette la creazione di grafici 2D e 3D dei risultati. Questa fase permette di analizzare i risultati definiti durante la modellazione,

come grandezze ingegneristiche quali spostamenti, tensioni o altre variabili di interesse, e di manipolarli per calcolare misure derivate o esportarli per un'eventuale elaborazione tramite altri strumenti software. In base alle osservazioni effettuate durante la fase di post-processing, è possibile identificare le modifiche necessarie, da apportare al modello per migliorarne le prestazioni.

1.3 Processo di Ottimizzazione

L'ottimizzazione si definisce come “il raggiungimento del massimo risultato possibile con i termini dati o in relazione a un determinato fine” [2]. Inoltre, poiché i parametri di progetto sono definiti in maniera tale da garantire la funzionalità del componente, la direzione di sviluppo per il conseguimento di un design ottimale è funzione di un obiettivo specifico, il quale viene definito a priori [9].

Nei processi di ottimizzazione si possono avere più obiettivi: potrebbe essere necessario, ad esempio, cercare un design che migliori contemporaneamente il peso e le prestazioni della struttura senza tenere conto dei costi, oppure, al contrario, si potrebbe mirare a minimizzare i costi di produzione massimizzando al contempo la sicurezza, trascurando però le prestazioni. Questi approcci, definiti come ottimizzazioni multi-obiettivo (MOO), prendono in considerazione simultaneamente più obiettivi, spesso in conflitto tra loro. Di conseguenza, può essere necessario scegliere una soluzione finale che rappresenti un compromesso tale da soddisfare al meglio tutti i requisiti progettuali.

Per un corretto processo di ottimizzazione è necessario definire:

- **Spazio di progettazione:** è il volume entro il quale può svilupparsi il componente. È definito in genere in funzione degli ingombri nominali che esso può avere all'interno di un assieme o dell'ambiente di lavoro.
- **Variabili di design:** si tratta di tutti quei parametri costruttivi del modello il cui valore deve essere ottimizzato. Contemporaneamente, esistono dei parametri che non vanno modificati.
- **Vincoli:** le variabili di design possono essere modificate a piacimento purché non vengano violati i vincoli di design. Il rispetto di tutti i vincoli porta ad un design “fattibile” (“Feasible Design” in OptiStruct).
- **Obiettivi:** in generale, l'obiettivo consiste nel migliorare il comportamento della struttura. L'aspetto da migliorare viene in genere definito dall'utente, come ad

esempio la minimizzazione della massa o la massimizzazione della prima frequenza naturale.

In questo studio si confrontano le caratteristiche di una struttura ottenuta mediante ottimizzazione topologica classica, con quelle di una struttura ottenuta tramite ottimizzazione di strutture lattice. Quest'ultima metodologia prevede l'uso combinato dell'ottimizzazione topologica nella prima fase, seguita dall'ottimizzazione dimensionale nella seconda fase.

1.3.1 Tipologie di Ottimizzazione Strutturale

HyperWorks mette a disposizione diverse tipologie di ottimizzazione, che possono essere suddivise in due categorie principali [8]. Da un lato ci sono le ottimizzazioni di concetto, come l'ottimizzazione topologica, topografica o free-size, il cui scopo è ottenere la configurazione ottimale di materiale. Dall'altro, ci sono le ottimizzazioni destinate al fine-tuning della struttura, come l'ottimizzazione di forma o dimensionale, che mirano a migliorare i dettagli delle singole parti senza alterare la configurazione generale.

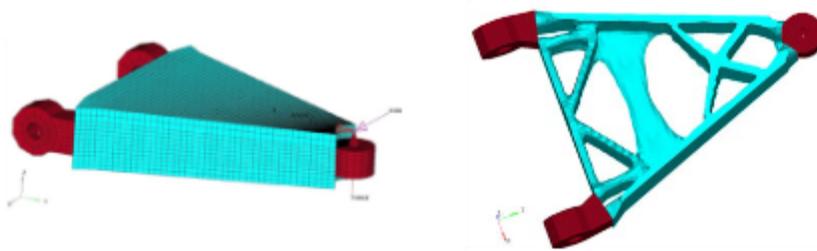


Figura 1.8: Esempio di ottimizzazione topologica.

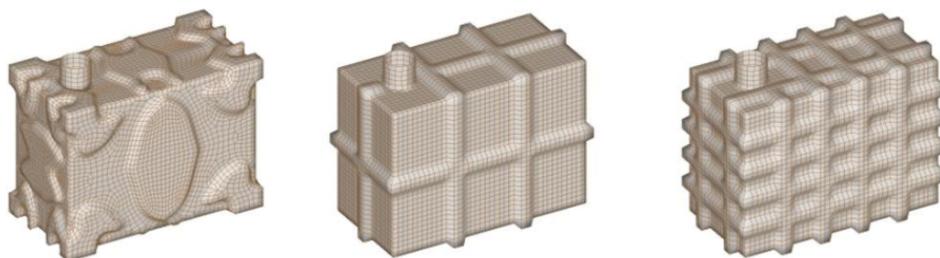


Figura 1.9: Esempio di ottimizzazione topografica.

- **Ottimizzazione Topologica:** agisce sulla variabile di design degli elementi per ottenere valori pari a 0 o 1. Gli elementi che fanno parte del design ottimale avranno una variabile di design tendente a 1, mentre gli altri a 0. Questo tipo di struttura consegue la definizione di una soglia di transizione: gli elementi con valori superiori a tale soglia saranno considerati solidi, mentre quelli inferiori verranno eliminati.

- **Ottimizzazione Topografica:** una variante avanzata dell'ottimizzazione di forma, dove il software genera una serie di rinforzi che seguono pattern specifici. Generalmente applicata a gusci e piastre, può anche essere utilizzata su strutture solide.



Figura 1.10: Esempio di ottimizzazione dimensionale

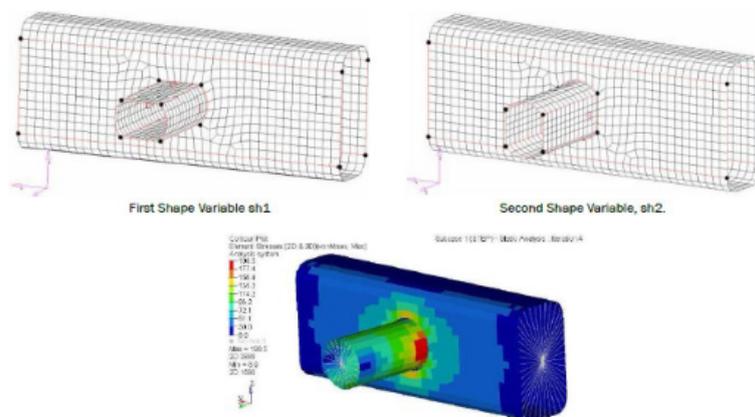


Figura 1.11: Esempio di ottimizzazione di forma con variazioni alternative predefinite.

- **Ottimizzazione Dimensionale:** maggiormente nota come "Size Optimization", riportata in figura 1.10, va a modificare i parametri definiti all'interno delle proprietà dell'elemento. Ad esempio, effettuando una size optimization su un guscio, questa andrà a modificare lo spessore al fine di trovarne uno ottimale. Tuttavia, tutti gli elementi a cui è stata assegnata una determinata proprietà avranno spessore uniforme.
- **Ottimizzazione di Forma:** nota anche come "Free-Shape Optimization" (figura 1.11), questa tecnica modifica la forma del componente variando le coordinate nodali

degli elementi. Un problema legato a questo approccio è la necessità di predeterminare le variazioni di forma; tuttavia, per risolvere tale limite, il software definisce le distanze entro cui avviene la variazione.

- **Ottimizzazione Lattice:** l'obiettivo di questa ottimizzazione (figura 1.12) è mantenere le densità intermedie risultanti da una precedente ottimizzazione topologica. In questo modo, la distribuzione del materiale diventa più efficiente, riducendo ulteriormente il peso complessivo.

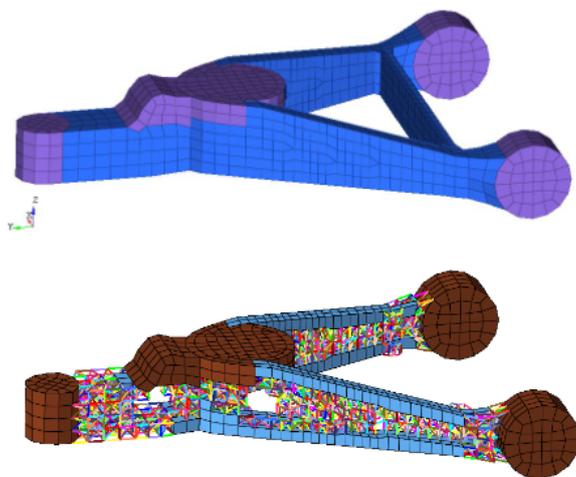


Figura 1.12: Esempio di ottimizzazione lattice.

1.3.2 L'Algoritmo di Ottimizzazione

Come accennato in precedenza, l'obiettivo dell'algoritmo di ottimizzazione è migliorare le prestazioni della struttura attraverso un processo iterativo. Questo algoritmo minimizza una specifica funzione obiettivo espressa in funzione del vettore delle variabili di design. Pur essendo progettato esclusivamente per minimizzare, la massimizzazione può essere eseguita se la si interpreta come la minimizzazione del reciproco della funzione obiettivo [4].

La capacità dell'algoritmo di raggiungere la soluzione ottimale dipende in gran parte dalla definizione delle funzioni obiettivo e dei vincoli. Se le funzioni sono altamente non lineari, infatti, l'algoritmo potrebbe richiedere più tempo per trovare una soluzione. Inoltre, il tipo di problema varia in base ai target definiti: OptiStruct, ad esempio, seleziona l'algoritmo più adeguato in base alla presenza, nel modello, di un numero di vincoli superiore rispetto alle variabili di design e viceversa.

La ricerca del minimo della funzione obiettivo è ulteriormente complicata dal fatto che essa raramente è convessa. Le funzioni convesse presentano tipicamente un solo minimo

(o massimo) assoluto, ma nelle funzioni di ordine superiore possono esistere più minimi locali, con un solo vero minimo globale, come mostrato in figura 1.13. L'algoritmo ha quindi il compito di individuare questo minimo assoluto, assicurando al contempo che il design finale rispetti tutti i vincoli definiti.

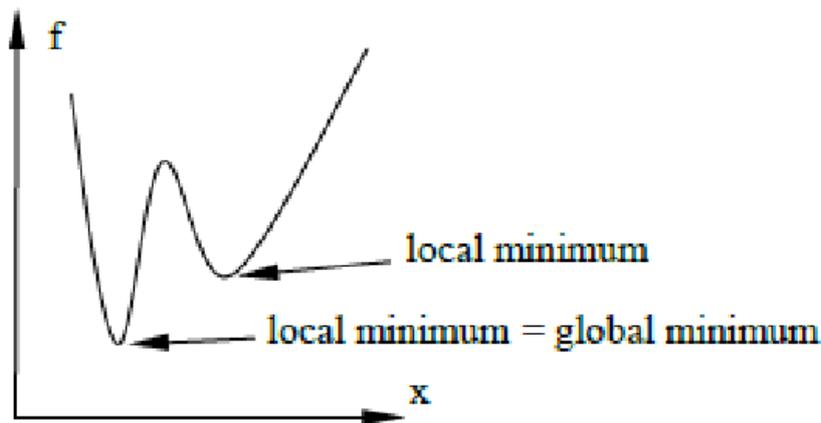


Figura 1.13: Funzione con diversi minimi locali.

In caso di problemi non convessi, risolti tramite metodi basati sul gradiente, il punto di partenza dell'ottimizzazione influisce direttamente sul risultato. Questo rende l'algoritmo particolarmente sensibile a variazioni e potrebbe portare a risultati diversi se si parte da differenti configurazioni iniziali. In generale, scegliendo n punti di partenza diversi, si otterranno n soluzioni differenti per l'ottimizzazione. Pertanto, non è garantito che il minimo individuato dall'algoritmo corrisponda al minimo globale.

Dal punto di vista del progettista, è importante capire che la ricerca del minimo è un processo iterativo, gestibile tramite vari parametri. È possibile, ad esempio, definire una tolleranza di convergenza sotto la quale un design si definisce accettabile, che si basa sulla differenza tra due iterazioni successive. Allo stesso modo è possibile gestire la velocità di convergenza alla soluzione, l'algoritmo di ottimizzazione e numerose altre opzioni.

Facendo riferimento alla convergenza, il software ne distingue due tipologie:

- **Regular Convergence:** si verifica quando la differenza percentuale della funzione obiettivo tra due iterazioni successive è inferiore alla tolleranza definita dall'utente.
- **Soft Convergence:** si verifica quando la variazione della funzione obiettivo tra due iterazioni è minima o nulla, dunque le response non vengono ricalcolate nell'ultima iterazione poiché la struttura rimane sostanzialmente invariata.

Le ottimizzazioni svolte in questo studio si basano sulla convergenza regolare, la cui soglia è fissata di default dal software, ma è modificabile dall'utente.

Metodo del Gradiente Il metodo più comunemente utilizzato per individuare il minimo locale della funzione obiettivo è il metodo del gradiente. Questo approccio sfrutta l'inclinazione della funzione per determinare la direzione esatta in cui correggere la previsione iniziale sul comportamento della struttura a seguito delle modifiche apportate. Partendo da una stima iniziale del design ottimale, il metodo del gradiente calcola la direzione in cui la funzione obiettivo decresce più rapidamente, basandosi sul gradiente della funzione stessa. Successivamente, si muove nella direzione calcolata fino a un certo punto, ripetendo il processo fino a convergere verso il minimo. La convergenza viene raggiunta quando il gradiente della funzione obiettivo diventa nullo.

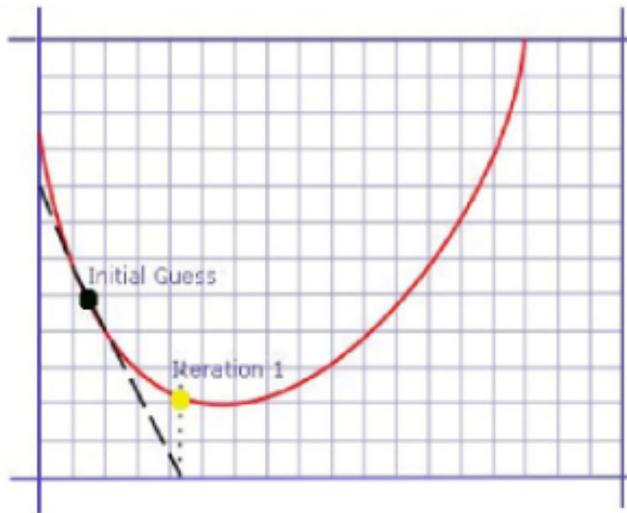


Figura 1.14: Rappresentazione del metodo del gradiente di Newton.

Volendo esplicitare i passaggi del metodo del gradiente:

1. Si parte dal punto x_0 .
2. Si valuta la funzione $F(x_i)$ e il gradiente della funzione $\tilde{N}F(x_i)$ in x_i .
3. Si determina il punto successivo usando la direzione negativa data dal gradiente della funzione secondo la formula: $x_{i+1} = x_i - g\tilde{N}F(x_i)$.
4. Ripetendo gli step 2 e 3, la funzione converge al minimo.

La descrizione del metodo qui riportata risulta particolarmente semplificata e poco realistica rispetto al funzionamento reale del metodo del gradiente. Tuttavia, tale descrizione risulta più che sufficiente per comprendere il funzionamento generale del metodo, poiché le metodologie di ricerca del minimo negli algoritmi di ottimizzazione non sono oggetto di studio in questa sede.

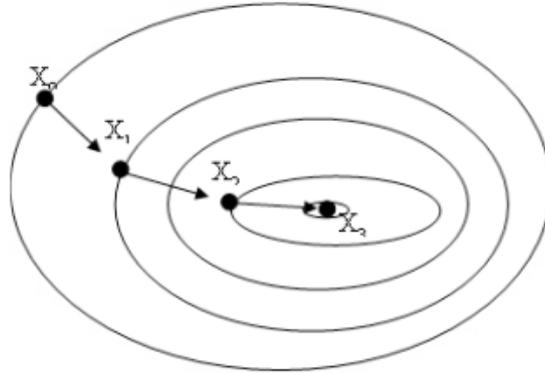


Figura 1.15: Metodo del Gradiente.

Il metodo del gradiente si basa sulla sensibilità delle risposte del sistema alle variazioni delle variabili di design. L'algoritmo deve essere in grado di valutare l'impatto delle modifiche al design sulle prestazioni della struttura per operare correttamente. Nel contesto di codici di analisi strutturale lineare, si possono utilizzare metodi come le differenze finite o approcci analitici basati sulle derivate delle response. Queste formulazioni descrivono il comportamento delle response con equazioni algebriche, garantendo così la continuità delle variabili di design tra un'iterazione e l'altra.

Infine, OptiStruct implementa ulteriori tecniche utili ad accelerare il processo di ottimizzazione. Queste ultime sono implementate dal software in maniera "trasparente" rispetto all'utente.

- **Constraint Screening:** individua i vincoli più critici per l'iterazione corrente.
- **Constraint Linking:** riduce il numero di vincoli da considerare nel caso in cui si definisca una simmetria.
- **Constraint Deletion:** elimina i vincoli non fondamentali per l'iterazione, qualora non siano presenti violazioni.

1.4 Fabbricazione Additiva

1.4.1 Principi e Caratteristiche

Il termine *Fabbricazione Additiva* si riferisce a un insieme di processi di fabbricazione in cui il componente viene realizzato mediante l'aggiunta ripetuta di materiale, contrariamente ai processi di produzione tradizionali che ne prevedono la rimozione. Questo processo consiste nell'accumulare materiale strato dopo strato fino al completamento l'oggetto finale.

Dal punto di vista del progettista, le tecnologie di Additive Manufacturing offrono una maggiore libertà progettuale. Infatti, grazie alle caratteristiche del processo di stampa è possibile ottenere forme non convenzionali, avvicinandosi così alla condizione ottimale per quanto riguarda la disposizione del materiale nello spazio, quasi senza vincoli sulla complessità geometrica. Queste caratteristiche consentono inoltre di ripensare l'intero assieme e, quando possibile, ridurre il numero di componenti integrando le funzioni di più parti in un unico elemento.

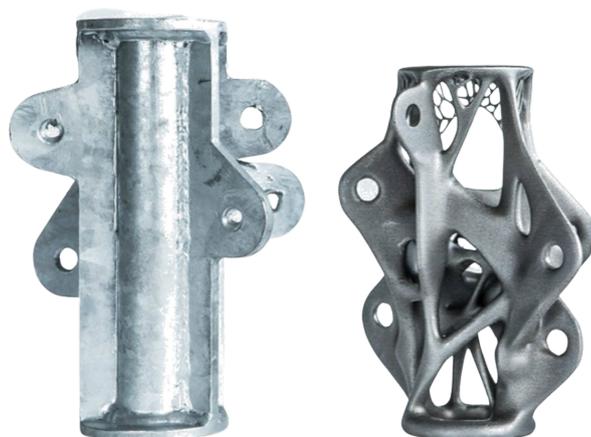


Figura 1.16: Confronto tra due diverse tecniche di fabbricazione applicate al medesimo componente.

Per sfruttare appieno il potenziale legato alla libertà di progettazione, è fondamentale utilizzare in modo efficace l'ottimizzazione topologica. Questa metodologia, una volta definite le condizioni operative del componente e le prestazioni richieste, genera un design significativamente più leggero rispetto al componente ottenuto tramite tecniche tradizionali, come illustrato in figura 1.16.

Tuttavia, il design derivato dall'ottimizzazione topologica deve essere rivisitato tenendo conto delle regole progettuali specifiche per le tecnologie utilizzate, poiché anche nell'ambito dell'Additive Manufacturing esistono delle limitazioni.

Un esempio di tali limitazioni è rappresentato dalla necessità di creare supporti per le

superfici inclinate con un angolo inferiore ai 45° rispetto al piano, in quanto il materiale presente nel volume di stampa non fornisce un adeguato supporto per il materiale stampato (almeno per quanto riguarda la gran parte delle tecnologie presenti in commercio). Inoltre, ciascuna tecnologia di stampa presenta dimensioni minime per le caratteristiche che possono essere realizzate; ad esempio, nella tecnologia L-PBF, la dimensione minima dei fori è di 0,45 mm e lo spessore delle pareti che costituiscono il componente non deve essere inferiore a 1 mm. Si deduce che oltre agli effettivi parametri di stampa, quali temperature, velocità o simili, anche l'orientazione della parte rispetto al piano di stampa costituisce una variabile progettuale di notevole importanza.

1.4.2 Fasi della Fabbricazione Additiva

Il processo di fabbricazione additiva si divide in due fasi principali, la prima relativa alla fabbricazione digitale e la seconda al processo fisico [10].

Fabbricazione Digitale Consiste in progettazione, impostazione e simulazione del processo di stampaggio, ed è interamente svolta attraverso software dedicati. Le fasi che caratterizzano la fabbricazione digitale sono riportate in figura 1.17.

1. **Progettazione e ottimizzazione** del componente.
2. Ancora nell'ambiente di lavoro CAD, si esegue la **poligonalizzazione** della superficie esterna del componente, ovvero la si approssima con superfici triangolari. Si esporta quindi il file STL, che contiene le informazioni sulla posizione dei vertici di ogni triangolo e della normale alla superficie il cui verso indica l'esterno del componente.
3. Il file STL viene poi importato in un software specifico, solitamente fornito dal produttore della macchina di stampa, che consente la **definizione dei dettagli di processo**, come l'orientamento delle parti, la generazione dei supporti e i parametri di processo e di **slicing**. Questi software spesso includono una funzione di simulazione del processo di stampa additiva, utile per identificare eventuali problematiche prima della produzione. Una volta completata questa fase, il software genera un codice che comunica alla macchina le operazioni da eseguire in sequenza.
4. Il file di istruzioni generato viene caricato in macchina, per poi avviare la produzione.
5. Al termine del processo si effettuano tutte le operazioni di rifinitura necessarie per completare il pezzo.

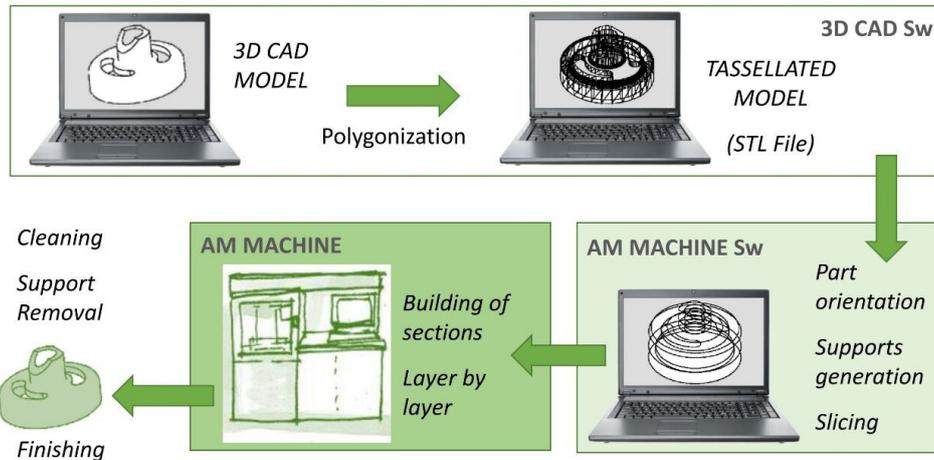


Figura 1.17: Fasi della fabbricazione digitale del componente e del processo produttivo.

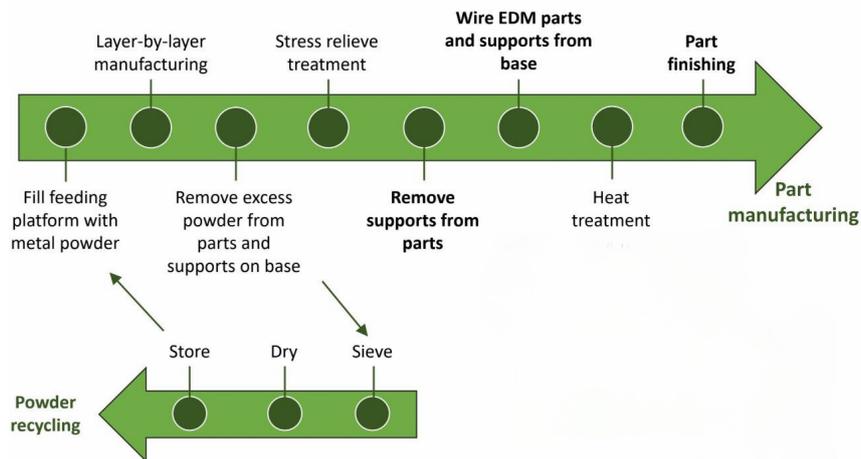


Figura 1.18: Flusso di lavoro dell'intero processo di produzione tramite fabbricazione additiva.

Flusso di Lavoro Le ultime due fasi descritte nel paragrafo precedente possono essere ulteriormente scomposte in alcune sotto-fasi.

La sequenza di operazioni che parte dal materiale grezzo e termina con il prodotto finito, riportata in figura 1.18, è descritta di seguito:

1. **Alimentazione della piattaforma:** si alimenta la piattaforma con del materiale grezzo, distribuendolo attraverso un apposito strumento integrato in macchina, detto recoater. Questo distribuisce uno strato uniforme di materiale con spessore predefinito. Nella maggior parte dei processi che coinvolgono materiali metallici, questi ultimi sono alimentati in macchina sotto forma di polvere metallica.
2. **Fusione del materiale:** il materiale appena depositato viene fuso localmente nei punti che fanno parte del componente, utilizzando una fonte di energia concentrata

di tipo laser ottico o fascio di elettroni.

3. **Rimozione e riciclo della polvere metallica:** dopo la stampa, la polvere residua viene recuperata attraverso un processo di filtraggio, asciugatura e stoccaggio, per permetterne il riutilizzo se ancora conforme agli standard qualitativi.
4. **Trattamento termico di distensione:** a causa degli elevati gradienti termici a cui è sottoposto durante la fusione, il materiale accumula stress residui significativi. Al fine di eliminare questi ultimi, si rende necessario un trattamento di ricottura di distensione al termine della stampa.
5. **Rimozione dei supporti e separazione dalla base di stampa:** è possibile rimuovere i supporti dalla parte e, successivamente rimuovere la parte stessa dalla base di stampa. Questo processo si svolge a valle della ricottura poiché visti gli stress residui, il componente potrebbe deformarsi se nuovamente riscaldato in assenza di un supporto che lo tenga in posizione.
6. **Trattamenti termici:** ulteriori trattamenti termici vengono eseguiti per conferire al materiale le proprietà desiderate, poiché a valle della ricottura di distensione, il metallo risulta eccessivamente duttile.
7. **Finitura:** il processo si conclude con la finitura della parte.

Esistono però altre tecnologie additive che utilizzano tecniche di deposizione del materiale e processi differenti. Ad esempio, nella tecnologia DED, il materiale viene depositato, sotto forma di polvere, direttamente nella zona fusa dal laser attraverso appositi ugelli. In altre tecnologie, come nella EBAM, si utilizzano bobine di metallo, che alimentano la zona fusa attraverso un filo, il quale viene fuso dalla fonte di energia al momento della deposizione.

Infine, esistono tecniche che prevedono la deposizione di un materiale polimerico, denominato "binder", sopra uno strato di polvere metallica. Questa resina viene distribuita attraverso un sistema simile a quello delle tradizionali tecnologie di stampa 2D. Successivamente, il polimero viene curato e, nelle fasi seguenti, rimosso. In ultimo, il pezzo viene sinterizzato, così da ottenere il componente finito.

1.4.3 Vantaggi e Svantaggi

Si riportano di seguito i principali vantaggi e svantaggi associati alle tecniche di FA. Questi sono legati sia ai prodotti ottenuti dalle suddette tecniche, che ai processi impiegati per realizzarli.

Vantaggi di prodotto

- Macchina unica per forme pressoché illimitate.
- Nessuno stampo o strumento necessario.
- Sono ammessi sottosquadri.
- Pochi e invariati step di manifattura.
- Ridotto intervento da parte degli operatori.
- Il tempo e i costi sono dipendenti dalla sola dimensione, non più dalla complessità.

Svantaggi di prodotto

- I sottosquadri richiedono la presenza di supporti.
- La finitura superficiale a valle del processo è molto grezza.
- Sono disponibili sul mercato un numero limitato di materiali adatti ai processi della FA, la maggior parte sono polimeri.
- Materiali e macchinari particolarmente costosi.

Vantaggi di processo

- Libertà di design.
- Strutture ultraleggere.
- Design integrato di assiemi.
- Customizzazione di massa.

Svantaggi di processo

- Velocità di produzione limitate.
- Dimensioni del volume di stampa limitate.
- La macchina può gestire pochi materiali o colori in contemporanea.
- La produttività risulta relativamente bassa.

Date le caratteristiche, la fabbricazione additiva rappresenta uno strumento particolarmente utile, ma il suo utilizzo dipende dal contesto specifico di applicazione. In generale, le caratteristiche che influenzano l'aumento o la diminuzione dei costi rispetto alle tecniche di fabbricazione tradizionali sono riportate in figura 1.19. La convenienza dell'investimento, dunque, dipende dal guadagno che si ha sulla riduzione del peso, sul risparmio per i costi di assemblaggio e sul guadagno in merito al ciclo vita del prodotto, che teoricamente risulta meglio ottimizzato per determinate condizioni di lavoro, rispetto allo stesso com-

ponente progettato per tecniche di manifattura tradizionali.

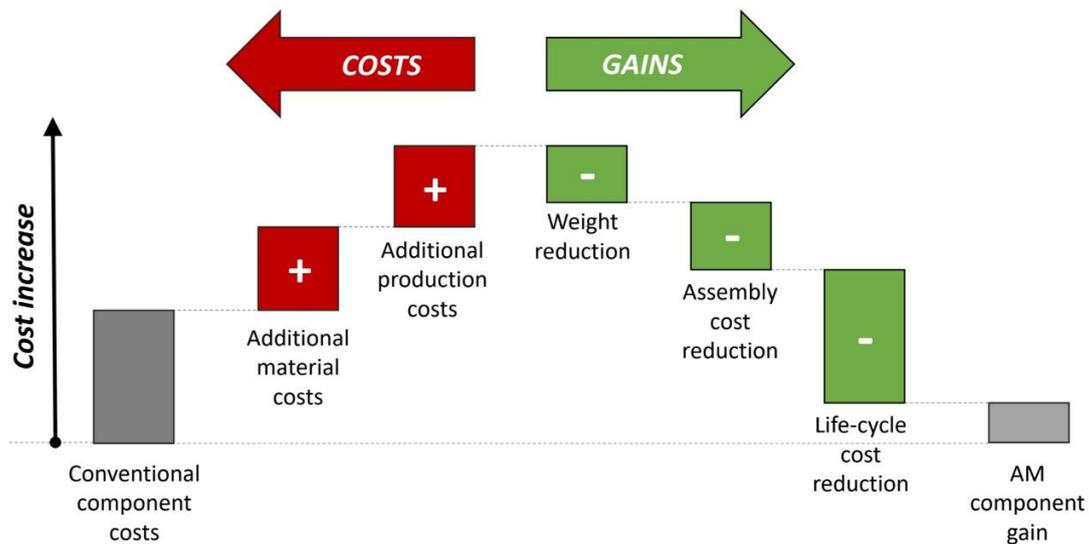


Figura 1.19: Costi e guadagni di un componente ottenuto tramite AM, rispetto ad uno tradizionale.

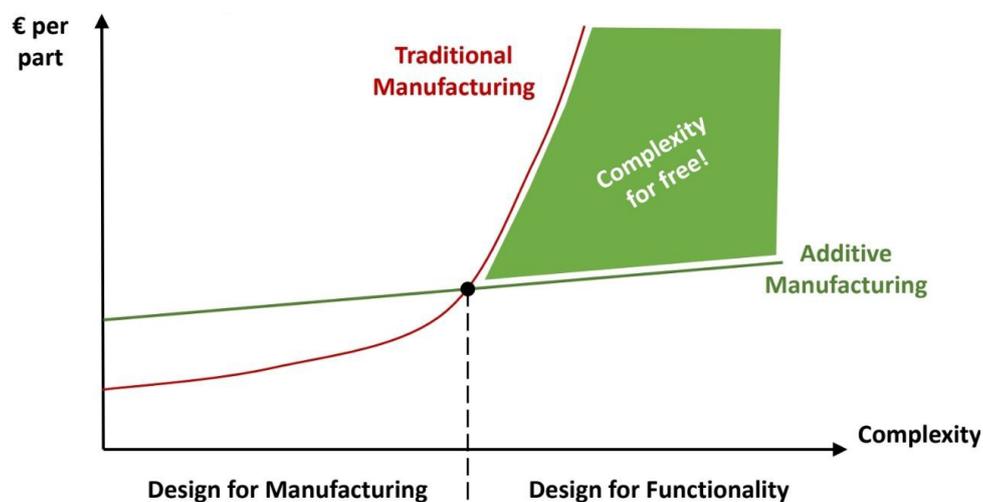


Figura 1.20: Confronto dei costi di produzione.

Complessivamente, si può affermare che le tecniche di FA risultano convenienti in termini economici solo quando i casi di applicazione richiedono design particolarmente complicati o prestazionali. Questo trend è osservabile in figura 1.20: per design relativamente semplici, ovvero per quelli che includono principalmente forme geometriche regolari e che non presentano eccessivi sottosquadri, le tecniche di manifattura tradizionali rimangono le più convenienti; al contrario, in presenza di complicazioni geometriche, le tecniche tradizionali diventano eccessivamente costose sia in termini tecnici che economici. Inoltre, nel

grafico riportato si osservano due aree fondamentali, separate dal turning point relativo alla convenienza delle diverse tecniche di manifattura. Queste si distinguono in funzione dei criteri da tenere in considerazione durante la progettazione: con le tecnologie di FA, la progettazione può tenere conto quasi esclusivamente della funzionalità del componente, a differenza del caso analogo, nel quale bisogna tenere conto, in fase di progettazione, della producibilità del componente.

2. Analisi FEM preliminare

Nella prima parte dello studio si effettua una analisi strutturale al fine di verificare la resistenza della struttura. Inoltre, in funzione dei risultati ottenuti, si definiscono gli obiettivi da raggiungere in fase di ottimizzazione.

La fase di pre-processing si svolge nell'ambiente di lavoro di HyperMesh 2023.1, mentre per la fase di post-processing viene utilizzato HyperView 2023.1.

2.1 Preparazione del Modello

2.1.1 Descrizione del Modello Importato

Il modello di riferimento analizzato è un telaio associato a un veicolo stradale di tipo didattico, quindi non rappresentativo, in quanto a dimensioni, di un veicolo reale. Le dimensioni del modello, infatti, sono pari a $550 \times 1100 \times 2344 \text{ mm}$, corrispondenti rispettivamente ad altezza, larghezza e lunghezza del veicolo.

Il modello è costituito da quattro componenti principali:

- **TUB**: è realizzato in laminati di fibra di carbonio e costituisce il corpo centrale del veicolo. Poiché l'analisi si concentra sulla parte metallica, questo componente verrà trattato superficialmente e considerato come una "black box" durante le fasi di analisi e ottimizzazione. Le proprietà associate al TUB sono descritte in PCOMP, mentre la definizione del materiale avviene attraverso la card MAT8
- **FRAME INTERFACE**: si tratta di una piastra metallica che facilita il collegamento tra la struttura in composito e le altre parti del modello (elementi verdi in figura 2.1). Anche questo componente non sarà oggetto di studio approfondito. In questo caso, la proprietà che descrive il comportamento degli elementi è PSHELL, mentre il materiale è alluminio, dunque MAT1.

- **FRONT FRAME:** rappresenta la parte frontale del modello (in grigio, in basso a sinistra, in figura 2.1), che funge da punto di attacco per le sospensioni anteriori. In figura è mostrata solo metà del componente reale, poiché deve essere specchiato rispetto al piano x-z per ottenere la geometria completa. In questo caso, come nel successivo, il tipo di proprietà in uso è PSOLID, mentre il materiale, l'alluminio, è definito dalla card MAT1.
- **REAR FRAME:** rappresenta la parte posteriore del modello (in grigio, in alto a destra, in figura 2.1), che serve da punto di attacco per le sospensioni posteriori. Le considerazioni fatte per il *front frame* valgono anche per questo componente.

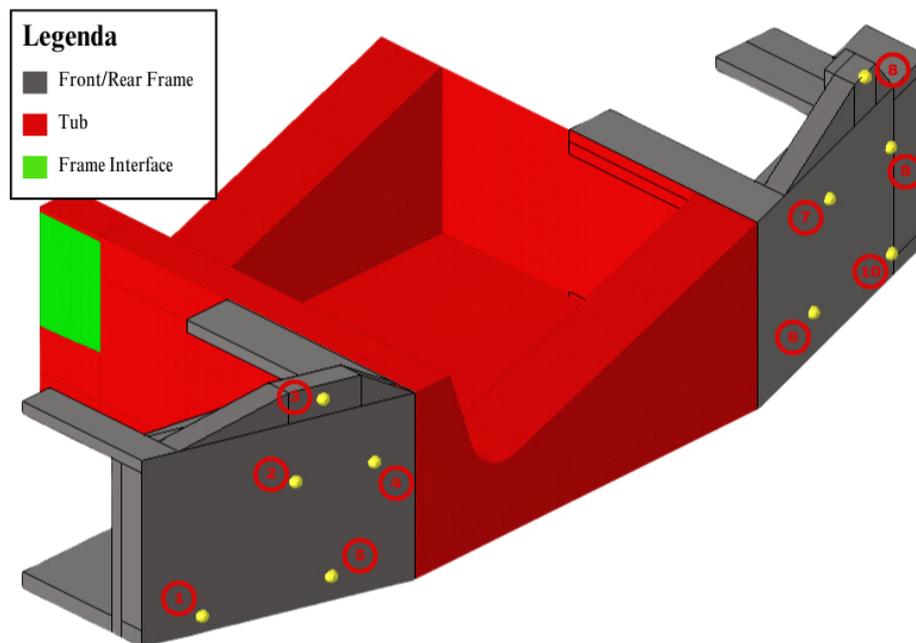


Figura 2.1: Modello di riferimento.

Infine, i punti di applicazione dei carichi sono mostrati in giallo in figura 2.1. Accanto a ciascun punto è indicata la numerazione del nodo corrispondente, per facilitarne l'identificazione durante l'analisi. Lo studio si concentrerà su *front* e *rear frame*, la cui massa totale misurata è pari a 450kg . Tuttavia, tale valore si riferisce all'intero spazio di design e, pertanto, non è direttamente confrontabile con le masse risultanti dalle diverse fasi di ottimizzazione. I design ottenuti attraverso l'ottimizzazione presenteranno infatti prestazioni significativamente inferiori rispetto alla configurazione originale.

2.1.2 Creazione della Mesh

La tecnica di *voxel meshing* riempie un determinato volume con elementi cubici di dimensione predefinita. Questa tecnica è utilizzata principalmente per le ottimizzazioni topologiche; tuttavia, la struttura ottenuta mediante questo metodo non fornisce risultati significativi quando si analizzano gli stress a cui essa è sottoposta [4].

Gli elementi usati per riempire il volume sono denominati CHEXA: si tratta di elementi solidi isoparametrici a otto nodi. Per questi elementi la formulazione del metodo degli elementi finiti prevede funzioni di approssimazione del primo ordine, espresse in funzione delle coordinate naturali dell'elemento. La trasformazione dalle coordinate assolute a quelle naturali dell'elemento è illustrata in figura 2.2.

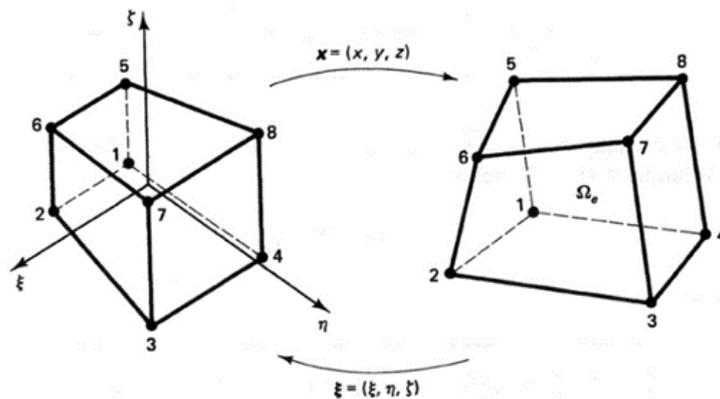


Figura 2.2: Elemento solido isoparametrico a otto nodi

Nel pannello dedicato di HyperMesh (*DesignSpace* \rightarrow *Global*) è possibile configurare le opzioni per la creazione della mesh voxel. In questa sezione si può impostare la dimensione degli elementi e, nel campo "Create voxels at.", definire i limiti geometrici che gli elementi devono rispettare. Sono disponibili tre opzioni:

- Intersezione e vuoti.
- Solo vuoti.
- Solo intersezioni.

Come mostrato in figura 2.3, nel primo caso la mesh riempie completamente il volume delimitato dalla superficie chiusa, anche se alcuni elementi potrebbero estendersi oltre il limite del volume definito. Nella seconda opzione, illustrata nell'immagine centrale, la mesh è generata esclusivamente all'interno del volume, assicurando che nessun elemento o parte di esso oltrepassi la superficie nominale. Infine, nel terzo caso, gli elementi sono

creati solo in corrispondenza della superficie nominale, lasciando un vuoto al suo interno.

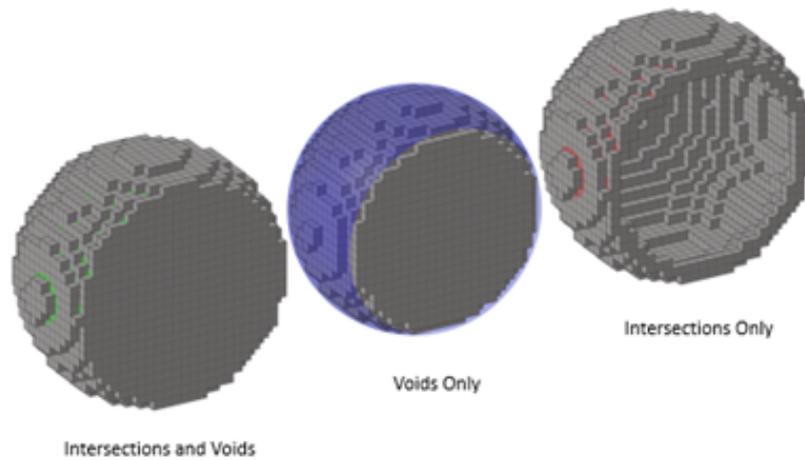


Figura 2.3: Modalità di creazione della mesh voxel

Nel caso in esame, vista la necessità di rispettare gli ingombri nominali definiti dalla geometria, si è selezionata la seconda opzione. In tal modo si evita che le porzioni di elementi fuoriuscenti dalla geometria possano essere fonti di interferenze in fase di montaggio della struttura.

Questa tipologia di mesh, inoltre, non è affetta dalla presenza di entità geometriche quali punti o linee sulla superficie, poiché tiene conto esclusivamente del volume racchiuso da esse. Infatti, in figura 2.4 si può osservare come entrambe le geometrie (grezza nell'immagine a sinistra e rifinita nell'immagine al centro) portano alla stessa distribuzione spaziale degli elementi. Tramite l'uso dello strumento "Surface repair" (paragrafo 1.2.1, pagina 5) si è confermato, inoltre, che non vi sono errori nella costruzione della geometria.

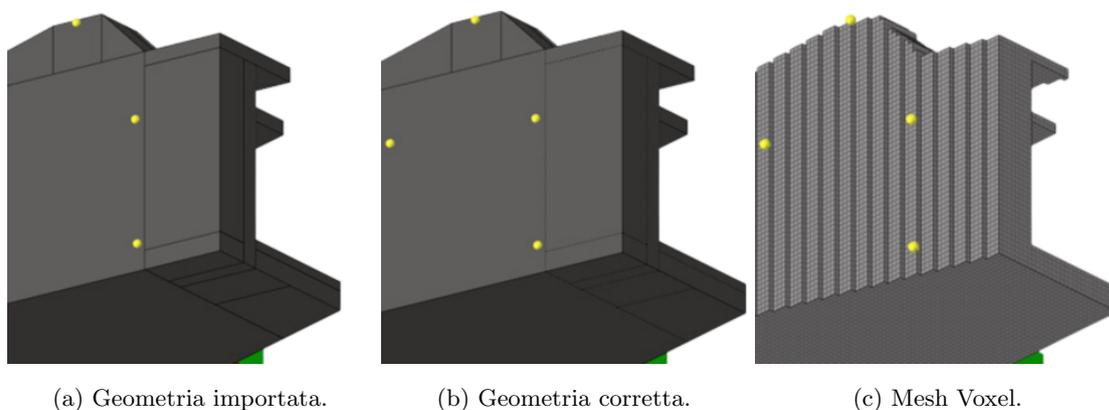


Figura 2.4: Effetto delle entità geometriche sul risultato finale della mesh voxel.

Sulla base di quanto appena affermato, è possibile impostare la mesh voxel di entrambe le geometrie relative al *front* e *rear frame*, per poi specchiare gli elementi generati.

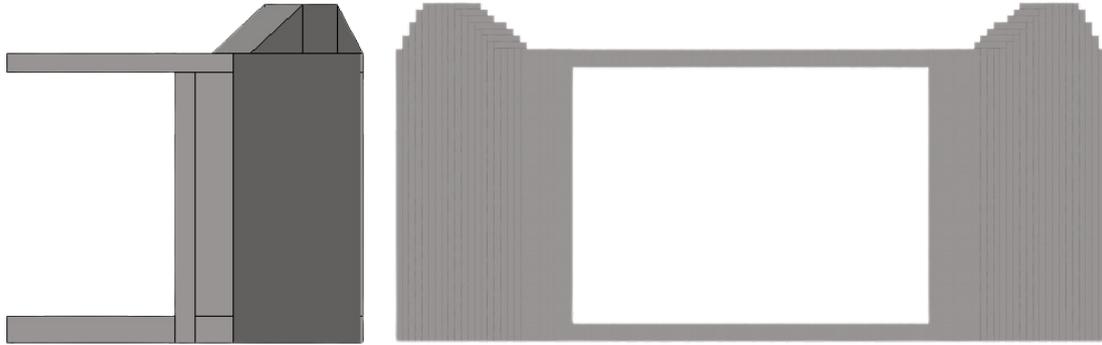


Figura 2.5: Vista anteriore del front frame prima e dopo la creazione della mesh.

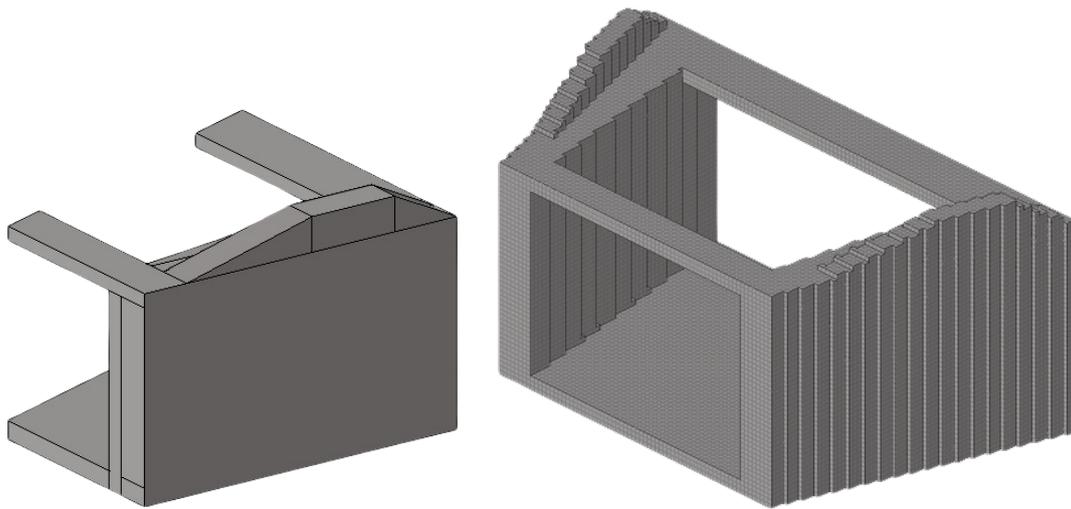


Figura 2.6: Vista inclinata del front frame prima e dopo la creazione della mesh.

Nelle figure 2.5 e 2.6, si riportano a sinistra le rappresentazioni della geometria del *front frame*, pre-definita nel modello. Questa rappresenta il volume nominale della struttura. Sulla destra, invece, si riportano due viste raffiguranti i risultati ottenuti dall'operazione di meshing, con conseguente specchiatura rispetto al piano $y = 0$.

La stessa osservazione vale per le figure 2.7 e 2.8, le quali raffigurano, a loro volta, le rappresentazioni analoghe al caso precedente, questa volta però per il *rear frame*.

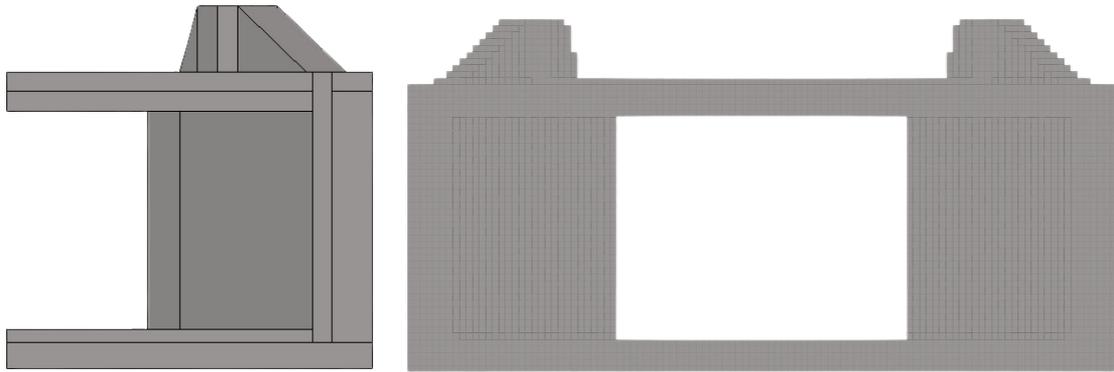


Figura 2.7: Vista anteriore del rear frame prima e dopo la creazione della mesh.

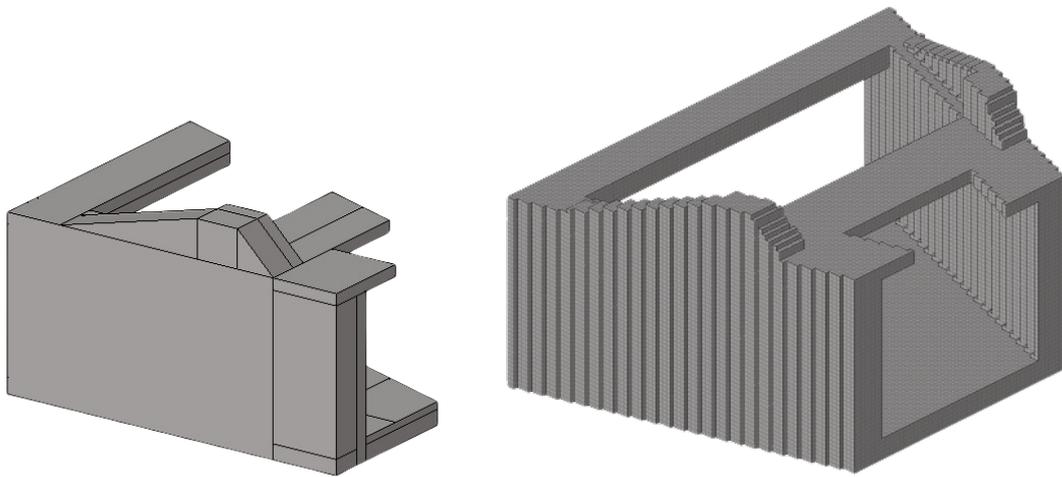


Figura 2.8: Vista inclinata del rear frame prima e dopo la creazione della mesh.

2.1.3 Definizione dei Collegamenti

Si procede ora a definire i collegamenti tra le diverse parti del modello. Gli elementi della mesh voxel del *front* e *rear frame* risultano infatti scollegati sia dalla struttura in materiale composito che dagli elementi shell. Analogamente, non esiste alcun collegamento tra la mesh voxel e i punti di applicazione del carico. Per risolvere questo problema vengono impostati degli elementi di collegamento tra i vari componenti.

Al fine di collegare la mesh voxel ai punti di applicazione del carico, la cui posizione è già definita nel modello, si introducono gli elementi rigidi RBE2. Questi elementi permettono di definire un moto rigido, i cui gradi di libertà indipendenti sono specificati in corrispondenza di un singolo nodo, mentre può essere specificato un numero indefinito di nodi dipendenti.

In figura 2.10, infatti, si osserva la presenza di un solo nodo indipendente che presenta tutti i gradi di libertà attivi, mentre i nodi dipendenti sono superiori alle 100 unità. Que-

sti ultimi sono stati definiti sugli elementi solidi nell'intorno del nodo indipendente, con un raggio di $10mm$.

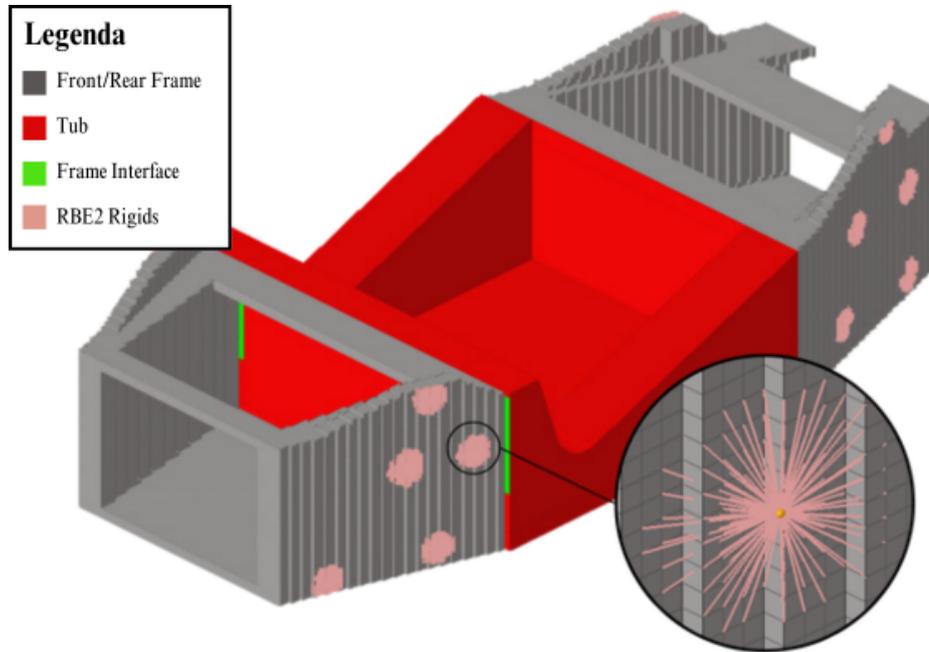


Figura 2.9: Dettaglio definizione RBE2.

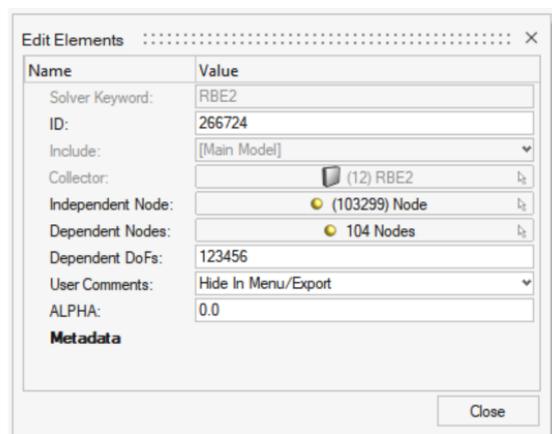


Figura 2.10: Card di definizione dell'elemento RBE2.

Per quanto riguarda il collegamento tra gli elementi esaedrici ed il corpo centrale della struttura, si definiscono dei collegamenti di tipo RBE3. Questo tipo di collegamento è comunemente usato per connettere parti diverse. A differenza degli RBE2, gli RBE3 non costituiscono un collegamento rigido, ma una connessione flessibile tra i nodi. In particolare, lo spostamento del nodo centrale (dipendente) è dato dalla somma pesata degli spostamenti dei nodi indipendenti. In figura 2.11 è riportato il particolare di un RBE3,

con la relativa card di definizione. Al suo interno si evidenzia la presenza di otto nodi indipendenti ed uno dipendente, per il quale sono attivi tutti i gradi di libertà. Quest'ultima affermazione implica che, in corrispondenza del nodo dipendente sono trasmesse sia le forze che i momenti.

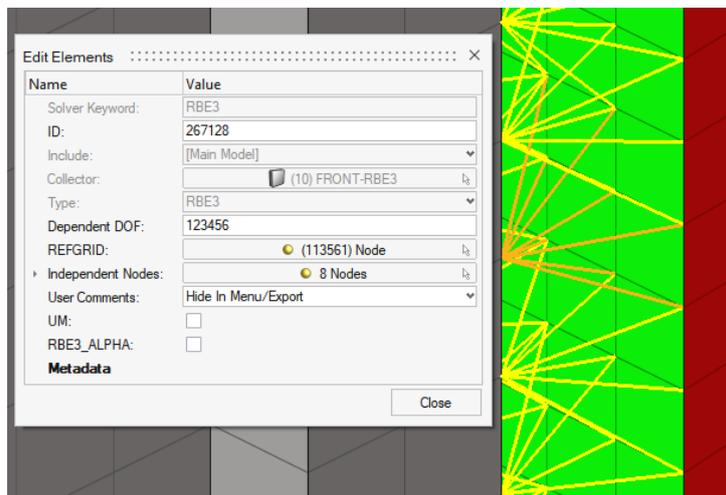


Figura 2.11: Particolare degli RBE3, con la card di definizione del collegamento.

2.1.4 Definizione dei Materiali e delle Proprietà

Nella libreria di HyperMesh esistono numerose tipologie di materiali che permettono di rappresentare in maniera precisa svariate caratteristiche, come ad esempio isotropia, anisotropia e ortotropia. Per la definizione del modello sono sufficienti due tipologie di materiali: uno che descriva le caratteristiche del materiale composito ed uno per quelle dell'alluminio.

La definizione delle proprietà è utile, invece, a caratterizzare il comportamento di un elemento in funzione della sua forma e dimensione.

Design Space MAT1 è la card che definisce le grandezze caratteristiche per un materiale lineare isotropo, il cui comportamento è indipendente dalla temperatura. Ogni materiale richiede l'inserimento dei valori numerici delle grandezze caratteristiche del materiale; per quelle fondamentali sono previsti dei valori di default in caso di mancato inserimento del dato. Per l'alluminio si sono indicati:

- Modulo di elasticità (E): 70 GPa
- Modulo di Poisson (NU): 0,33
- Densità (RHO): 2700 kg/m^3

PSOLID definisce le proprietà degli elementi solidi quali, ad esempio, CHEXA. Al suo interno si definisce il materiale al quale la proprietà fa riferimento, in questo caso alluminio (ALU).

PBEAML è una proprietà che si presenta in una fase più avanzata dello studio, ovvero risulta dalla conversione degli elementi solidi in struttura lattice (vedi sezione 4.1.1). Definisce le proprietà di un elemento trave in funzione delle sue dimensioni agli estremi. È possibile rappresentare svariate tipologie di sezione, in funzione dei campi di definizione delle caratteristiche dell'elemento che si vanno a riempire nella fase di creazione della proprietà. Per l'ottimizzazione topologica di tipo lattice, questa definizione è svolta in automatico dal software. Tuttavia, è possibile affermare che nel caso in esame vengono definite travi a sezione circolare di forma tronco conica (figura 2.12). Per ogni elemento il software definisce in automatico i seguenti parametri:

- MID: ID del materiale di cui è composta la trave. In questo caso, alluminio.
- TYPE: in questo caso definita come ROD, ossia una trave avente forma tronco-conica.
- DIM1(A): raggio della trave nel nodo A.
- DIM1(B): raggio della trave nel nodo B.

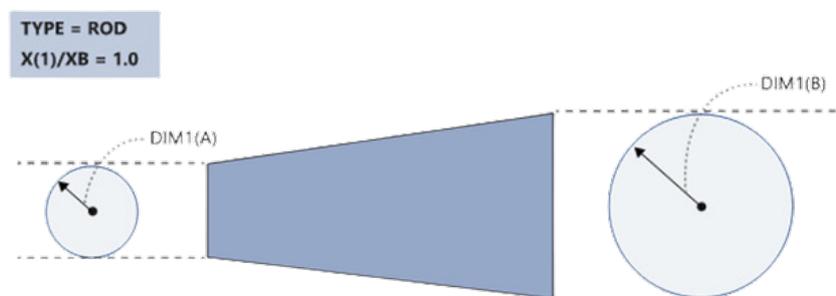


Figura 2.12: Trave a sezione circolare.

Struttura in composito MAT8 è il materiale che definisce le caratteristiche per un materiale lineare ortotropo bidimensionale; come nel caso precedente le caratteristiche definite sono indipendenti dalla temperatura. Le grandezze definite per le lamine di fibra di carbonio sono:

- E1,E2: moduli di elasticità nelle due direzioni principali, definite come longitudinale e trasversale. Entrambi pari a 65000MPa .
- NU12: modulo di Poisson per il carico uniassiale in direzione longitudinale. Questo è sufficiente poiché il modulo di Poisson in direzione trasversale è ricavato automati-

camente dal software usando la relazione $NU_{12} \cdot E_2 = NU_{21} \cdot E_1$. Il valore del modulo di Poisson è fissato pari a 0,05.

- G12,G1Z, G2Z: moduli di taglio, pari a $4000MPa$
- RHO: densità $1600kg/m^3$
- Xt, Xc, Yt, Yc, S: tensioni o deformazioni ammissibili nelle direzioni longitudinale e trasversale. Sono usate per verificare la resistenza a frattura. Vengono definite come in figura 2.13.

Xt:	800.0
Xc:	400.0
Yt:	800.0
Yc:	400.0
S:	60.0

Figura 2.13: Grandezze caratteristiche del materiale composito.

PCOMPP costituisce l'insieme delle proprietà di un laminato composito definito a partire dalla singola lamina. Nel caso in esame si è imposto:

- Z0: rappresenta la distanza del piano di riferimento dell'elemento dalla superficie inferiore del guscio stesso. Il valore assegnato è zero; ciò indica che i due piani appena definiti sono coincidenti (figura 2.14).
- SB: è lo stress di taglio inter-laminare ammissibile. Fissato pari a 50 Mpa
- FT: richiede l'inserimento del codice relativo alla teoria di cedimento da utilizzare. Nel caso in esame si è inserito il codice STRN, che corrisponde alla teoria della massima deformazione.

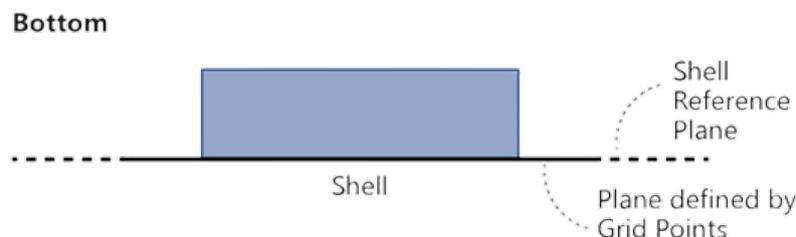


Figura 2.14: Linee di riferimento per l'elemento guscio.

PSHELL definisce le proprietà del guscio, distinguendo tra resistenza membranale, flessionale o trasversale. Permette inoltre di prendere in considerazione l'accoppiamento degli

sforzi membranali e flessionali. Per quanto riguarda il modello in esame, si definiscono i seguenti campi:

- MID1, MID2 e MID3: identificano, in ordine, il materiale coinvolto nella resistenza agli sforzi di tipo membranale, flessionale e trasversale. In tutti e tre i casi si indica l'ID relativo all'alluminio. Avendo definito MID3 si rientra nella teoria dei gusci spessi. MID4 non è indicato, disaccoppiando i comportamenti membranale e flessionale.
- T: valore di default per lo spessore della membrana, impostato pari a 10mm.
- $12I/T^3$: parametro relativo alla rigidezza a flessione. Può essere definito come il rapporto tra il momento di inerzia flessionale del guscio (I) ed il momento di inerzia flessionale di un guscio omogeneo ($T^3/12$).
- ZOFFS: definisce l'offset tra il piano definito dai punti che costituiscono l'elemento ed il piano di riferimento del guscio. In questo caso si imposta il formato su Surface e si imposta che il piano definito dai nodi degli elementi rappresenti la parte inferiore del guscio.

2.1.5 Condizioni al Contorno

Per condizioni al contorno si intende l'insieme delle informazioni che permettono di definire la fisica del problema. Qualunque tipo di calcolo strutturale necessita di numerose informazioni, sia sulla struttura stessa, sia sulle forzanti esterne che agiscono su di essa e sui vincoli che la fissano nello spazio.

Carichi Si applica un carico statico per ogni direzione principale, per ogni nodo. Complessivamente, ciò comporta la definizione di trenta casi di carico distinti, ognuno con una forza applicata in un determinato punto. Poiché il modello è di natura didattica, i valori reali delle forze a cui il modello è soggetto non sono noti; pertanto, queste ultime sono fissate pari a $1000N$.

Attraverso l'apposito comando (*Analyze* → *Structural* → *Loads*) è possibile dare al software l'istruzione "FORCE" (figura 2.15), la quale crea un carico statico applicato in un punto. Le sue componenti possono essere definite sia in funzione del sistema di riferimento locale, sia in funzione di quello globale. Poiché, in questo caso, le forze sono orientate parallelamente ai tre assi del sistema di riferimento globale, si utilizza proprio quest'ultimo per specificarne la direzione. Infatti, per definire la direzione di applicazione di un carico, si usa un sistema di versori unitari (N_1, N_2, N_3). In caso di carico allineato parallelamente all'asse x , la terna di versori sarà $(1,0,0)$. La tabella 2.1 riporta, per ogni

subcase, il punto di applicazione del carico e la direzione dello stesso.

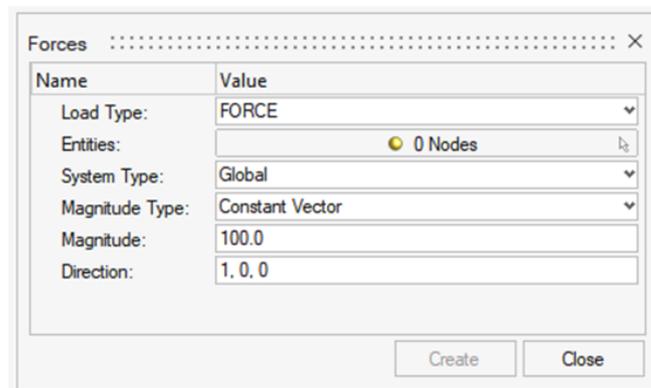


Figura 2.15: Finestra dedicata alla definizione di un carico statico.

Ciascuna di queste forze deve poi essere assegnata a un determinato "Load Collector", ovvero un raccogliitore dei carichi e dei vincoli da utilizzare contemporaneamente nella stessa condizione di carico. Nel modello in esame, considerando l'uso dell'analisi di tipo Inertia Relief (paragrafo 2.1.5), i raccoglitori conterranno esclusivamente delle forze.

Vincoli Nell'ambito della scienza delle costruzioni, si definisce isostatica una struttura che presenta un numero di vincoli pari al numero dei suoi gradi di libertà interni ed esterni. Se i vincoli sono superiori al numero di gradi di libertà, allora la struttura si dice iperstatica; in caso contrario si dice labile o isostatica. Inoltre, in genere, in corrispondenza del vincolo si ha che la forzante è diversa da zero.

In HyperMesh si definiscono i vincoli attraverso le istruzioni SPC e SPCD, che specificano l'entità da vincolare ed il valore di spostamento forzato in corrispondenza del vincolo stesso. Le forze risultanti dalla presenza del vincolo vengono calcolate in maniera automatica dal software e aggiunte alle altre forzanti.

Tuttavia, per simulare al meglio strutture non vincolate è necessario effettuare un'analisi di tipo Inertia Relief (INREL). È il caso, ad esempio, di aerostutture in volo, veicoli in movimento o corpi nello spazio. In questo tipo di analisi non sono presenti vincoli esterni, ma le forze vengono bilanciate applicando al corpo accelerazioni traslazionali e rotazionali. Applicando queste accelerazioni ai singoli elementi si fa in modo che la somma totale delle forze applicate al corpo sia zero.

Subcase	Direzione di Carico	Punto di Applicazione
1	X	1
2	Y	1
3	Z	1
4	X	2
5	Y	2
6	Z	2
7	X	3
8	Y	3
9	Z	3
10	X	4
11	Y	4
12	Z	4
13	X	5
14	Y	5
15	Z	5
16	X	6
17	Y	6
18	Z	6
19	X	7
20	Y	7
21	Z	7
22	X	8
23	Y	8
24	Z	8
25	X	9
26	Y	9
27	Z	9
28	X	10
29	Y	10
30	Z	10

Tabella 2.1: Direzione e punto di applicazione della forza associati al relativo subcase.

2.2 Analisi lineare statica

Una volta definite le informazioni del modello che descrivono le sollecitazioni a cui il componente è sottoposto durante il suo funzionamento, è possibile effettuare una prima analisi statica. Come già accennato in precedenza, lo scopo di questa analisi statica è quello di osservare la risposta della struttura e definire, di conseguenza, gli obiettivi dell'ottimizzazione.

Poiché la mesh Voxel non offre risultati soddisfacenti in termini di distribuzione delle tensioni all'interno dell'elemento, si analizzano solo gli spostamenti nodali nel punto di applicazione della forza. Ad esempio, nel primo Subcase, in cui la forza viene applicata al primo nodo con direzione parallela all'asse x, lo spostamento verrà misurato lungo la direzione x, sempre in corrispondenza del primo nodo.

La misurazione appena descritta può essere effettuata in HyperView e il processo di misura degli spostamenti può essere riassunto in tre passaggi principali:

- Selezione del subcase da analizzare attraverso un apposito menù a tendina.
- All'interno dello strumento “*Plot* → *Contour*” si selezionano i componenti associati a *front* e *rear frame*, per poi scegliere il tipo di risultato e le specifiche dello stesso, che nel caso in esame sarà lo spostamento lungo la direzione di interesse per lo specifico Subcase.
- Si passa poi allo strumento “*Query* → *Table*”, che tra le varie opzioni permette di riportare l'ID del nodo, il numero del Subcase selezionato ed il valore numerico dello spostamento misurato nel punto stesso, nella direzione indicata al passaggio precedente.

Reiterando i passaggi appena descritti per ogni Subcase si ottengono i valori riportati in tabella 2.2 dove sono riassunti i valori degli spostamenti nodali misurati e delle rigidità ricavate. Inoltre, dalle prime tre colonne è possibile osservare l'associazione tra Subcase, direzione di carico e punto di applicazione. A partire dal valore misurato dello spostamento si ricava la rigidità corrispondente, ipotizzando una relazione lineare elastica tra la forza applicata e lo spostamento:

$$k = \frac{F}{x} \quad (2.1)$$

Per quanto riguarda i casi di applicazione delle forze lungo l'asse x, le figure 2.16 mostrano le distribuzioni di deformazione sui *front* e *rear frame*.

Si nota che i valori massimi di deformazione si registrano quando il carico è applicato ai punti 3 e 8, raggiungendo rispettivamente 23,66 mm e 18,62 mm. Al contrario, i valori di deformazione più bassi si osservano nei punti 1, 5, 6 e 10, con spostamenti rispettivi di 2,50 mm, 2,56 mm, 3,19 mm e 2,59 mm. Questi risultati evidenziano una maggiore rigidità nella parte inferiore della struttura, rispetto a una parte superiore meno rigida, coerentemente con la maggiore concentrazione di materiale alla base sia del *front* che del *rear frame*.

Nelle figure 2.17 sono riportate le distribuzioni dei valori di spostamento lungo l'asse *y*. Analogamente al caso precedente, si riscontrano spostamenti maggiori nella parte superiore della struttura e valori inferiori alla base, mentre la parte centrale presenta spostamenti intermedi tra questi due estremi. Complessivamente, dato che i valori di spostamento sono più contenuti, si può affermare che la struttura è meno sensibile ai carichi trasversali rispetto a quelli longitudinali.

Infine, nelle figure 2.18 si riportano le distribuzioni dei valori di spostamento lungo l'asse *z*. Si tratta, in generale, della direzione di carico che genera le maggiori sollecitazioni sulla struttura. In tutte le condizioni di carico analizzate lungo *z*, infatti, i valori di spostamento si attestano su valori di spostamento superiori ai 10 mm.

(a) X-1

(b) X-3

Figura 2.16: Spostamento in direzione X.

(a) Y-8

(b) Y-10

Figura 2.17: Spostamento in direzione Y.

(a) Z-2

(b) Z-6

Figura 2.18: Spostamento in direzione Z.

2.2.1 Definizione degli Obiettivi

La rigidezza ricavata dall'equazione 2.1 rappresenta la resistenza opposta dall'intera struttura ad una determinata sollecitazione. Tuttavia, questo valore è riferito a una struttura piena, il che rende impraticabile la riduzione della densità degli elementi in un'eventuale ottimizzazione. Se i vincoli (spostamento massimo consentito al nodo) coincidessero con gli spostamenti misurati durante l'analisi preliminare, la riduzione di materiale comporterebbe la violazione di almeno uno di essi. Infatti, riducendo la quantità di materiale presente nella struttura la si rende meno rigida, portando ad un aumento delle deformazioni a parità di condizioni di carico. È quindi necessario imporre una riduzione della rigidezza richiesta alla struttura e, di conseguenza, consentire un aumento degli spostamenti nodali ammessi.

Ad ogni condizione di carico, quindi, va associato il corrispettivo limite sullo spostamento massimo del punto di applicazione della forza. Tale valore è rappresentato proprio dallo spostamento obiettivo: si consente al software di modulare la densità degli elementi, purché la struttura sia comunque in grado di soddisfare i requisiti minimi di rigidezza imposti per ogni condizione di carico.

Nel caso in esame la rigidezza obiettivo (k_{obj}) dello studio è stata definita pari al 40% di quella originale. A partire dalla rigidezza obiettivo così determinata, sono stati calcolati gli spostamenti massimi ammessi (x_{obj}) per ciascun nodo utilizzando l'equazione 2.2.

$$x_{obj} = \frac{F}{k_{obj}} \quad (2.2)$$

Questi valori sono riportati nella colonna "Spostamento ammissibile" in tabella 2.2, mentre il punto e la direzione di misura dello spostamento associato al singolo subcase è osservabile in tabella 2.1, in maniera analoga a quanto descritto per le forzanti.

Subcase	Spostamento [m]	Rigidezza [kN/mm]	Rigidezza Ammissibile [kN/mm]	Spostamento Ammissibile [m]
1	2,500E-03	400	160	6,250E-03
2	2,407E-03	415	166	6,018E-03
3	2,327E-02	43	17	5,818E-02
4	8,235E-03	121	49	2,059E-02
5	1,982E-03	505	202	4,955E-03
6	1,188E-02	84	34	2,970E-02
7	2,366E-02	42	17	5,915E-02
8	8,538E-03	117	47	2,135E-02
9	1,128E-02	89	35	2,820E-02
10	8,788E-03	114	46	2,197E-02
11	2,568E-03	389	156	6,420E-03
12	1,189E-02	84	34	2,973E-02
13	2,559E-03	391	156	6,398E-03
14	1,411E-03	709	283	3,528E-03
15	1,102E-02	91	36	2,755E-02
16	3,193E-03	313	125	7,983E-03
17	1,091E-03	917	367	2,728E-03
18	1,429E-02	70	28	3,573E-02
19	7,305E-03	137	55	1,826E-02
20	5,000E-03	200	80	1,250E-02
21	1,405E-02	71	28	3,513E-02
22	1,862E-02	54	21	4,655E-02
23	2,126E-02	47	19	5,315E-02
24	1,625E-02	62	25	4,063E-02
25	6,360E-03	157	63	1,590E-02
26	3,423E-03	292	117	8,558E-03
27	1,776E-02	56	23	4,440E-02
28	2,488E-03	402	161	6,220E-03
29	2,235E-03	447	179	5,588E-03
30	1,871E-02	53	21	4,678E-02

Tabella 2.2: Spostamento misurato, ammissibile e relative rigidezze in relazione al Subcase.

3. Ottimizzazione Topologica

Nella seconda fase dello studio viene condotta una ottimizzazione topologica classica sulla struttura, con l'obiettivo di creare un modello di riferimento da confrontare con la struttura lattice.

A tal fine, è stato utilizzato *HyperMesh 2023.1*, con *OptiStruct* come solutore. Per individuare l'iterazione di interesse si è impiegato il software *HyperGraph*, mentre l'analisi dei risultati è stata svolta utilizzando *HyperView*. Tuttavia, non è detto che essa debba andare a modificare tutti gli elementi presenti nel modello. Potrebbe essere necessario, ad esempio, escludere alcune parti della struttura utili ad interfacciare la stessa con altre componenti dell'insieme.

3.1 Definizione di Design e Non-Design Space

Come già accennato in precedenza, l'obiettivo dell'ottimizzazione topologica è quello di produrre una distribuzione di materiale ottimale per una struttura, dato un determinato volume di controllo. Tuttavia, non è garantito che l'ottimizzazione debba coinvolgere tutti gli elementi del modello: potrebbe essere necessario, ad esempio, escludere alcune parti della struttura che si interfacciano con altre componenti dell'insieme. Se tali parti non fossero definite come non-design space, l'ottimizzazione potrebbe modificarne le proprietà o la disposizione nello spazio, causando potenziali problematiche durante la fase di assemblaggio.

Si definisce quindi il design space, ovvero la porzione di volume sulla quale l'ottimizzazione può intervenire liberamente senza compromettere la funzionalità del componente. In questo modo, si delimita l'area di interesse per lo studio, separandola dal resto del modello, che deve rimanere inalterato.

Non-Design Space A questo punto è necessario identificare gli elementi del modello che non devono essere modificati dall'ottimizzazione topologica. Generalmente, infatti, si tende a non applicare carichi o vincoli direttamente al design space, poiché durante l'ottimizzazione potrebbe essere alterato al punto da non garantire più l'accoppiamento con le altre parti dell'assieme. La distinzione tra design e non-design space è implementata nel modello creando una seconda proprietà, caratterizzata dagli stessi attributi della proprietà associata al design space, ma con un nome e un ID differenti. In questo modo, l'ottimizzazione avrà come target esclusivamente la proprietà del design space.

Come accennato, gli elementi inclusi nel non-design space sono tipicamente quelli che costituiscono l'interfaccia tra diverse entità del modello. Nel caso specifico, però, gli elementi dei *front* e *rear frame* che interagiscono con il *tub* sono stati inclusi nel design space. Questa scelta progettuale deriva dall'esigenza di analizzare i percorsi di carico più efficienti durante l'ottimizzazione. Di conseguenza, il non-design space sarà limitato esclusivamente agli elementi che fungono da collegamento tra la struttura e i carichi ad essa applicati.

I punti di applicazione delle forzanti rappresentano l'interfaccia tra il modello e l'ambiente esterno. In questi punti, la trasmissione delle forze e dei momenti al corpo avviene attraverso gli elementi RBE2, i quali presentano da un lato la forzante, dall'altro un nodo appartenente ad un elemento CHEXA. In base a quanto detto prima si evince che gli elementi i cui nodi sono collegati agli RBE2 dovranno essere esclusi dal volume di interesse per l'ottimizzazione, poiché da essi dipende la trasmissione delle sollecitazioni alla struttura in alluminio. Gli elementi CHEXA selezionati corrispondono a quelli in connessione diretta con un RBE2. In figura 3.1, tali elementi sono riportati in giallo, mentre gli RBE2 sono contrassegnati in azzurro.

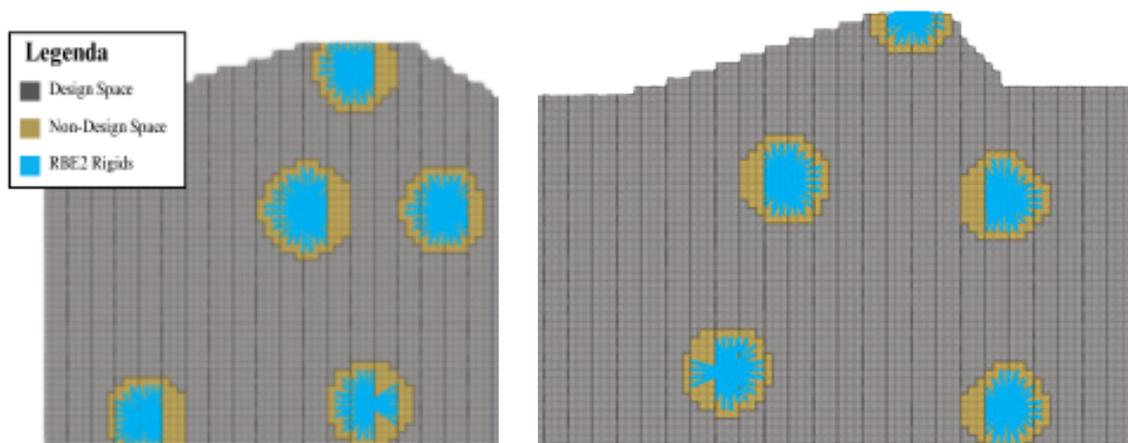


Figura 3.1: Vista laterale di front e rear frame.

Inoltre, data la simmetria del modello rispetto al piano x-z, anche i corrispettivi elementi sul lato destro vengono inclusi nel non-design space, nonostante i carichi siano applicati in modo asimmetrico solo sul lato sinistro.

Design Space Escludendo tutti gli elementi appartenenti al non-design space, i restanti elementi solidi CHEXA che compongono il *front* e il *rear frame* sono soggetti a modifiche durante l'ottimizzazione e, pertanto, inclusi nel design space. Come si può osservare in figura 3.2, la maggior parte degli elementi ottenuti tramite Voxel meshing fa parte del design space.

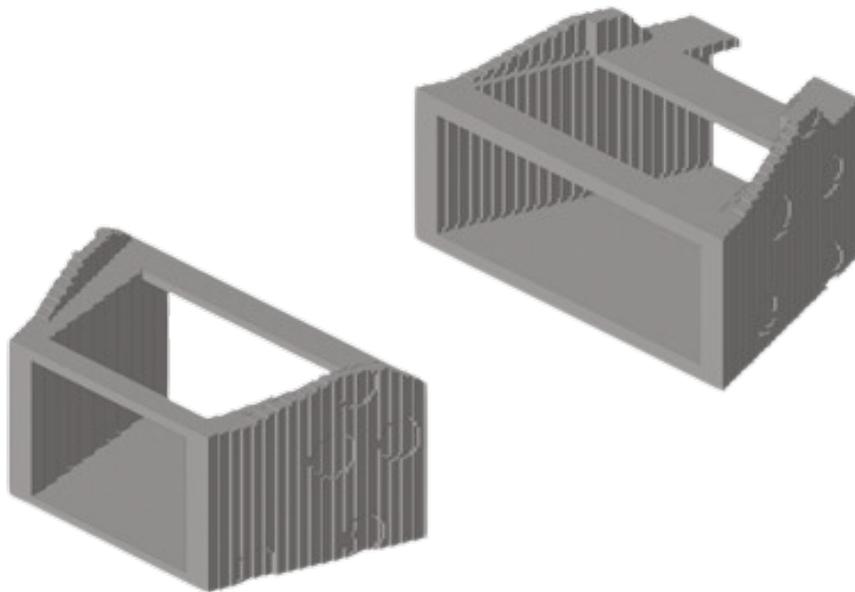


Figura 3.2: Elementi appartenenti al design space.

La massa totale del design space, che fa da riferimento per le ottimizzazioni eseguite nei capitoli successivi, è pari a $444kg$. Come affermato in precedenza, si tratta ancora una volta di una massa complessiva dello spazio di design, non confrontabile con i risultati delle ottimizzazioni data la riduzione di rigidità richiesta per queste ultime.

3.2 Definizione dei Target e dei Controlli dell'Ottimizzazione

Una volta definiti gli elementi soggetti all'ottimizzazione topologica, è possibile creare le entità necessarie per garantire una corretta esecuzione del processo. Tra queste, le più rilevanti sono: le variabili di progetto, le "Responses", l'obiettivo e i vincoli dell'ottimiz-

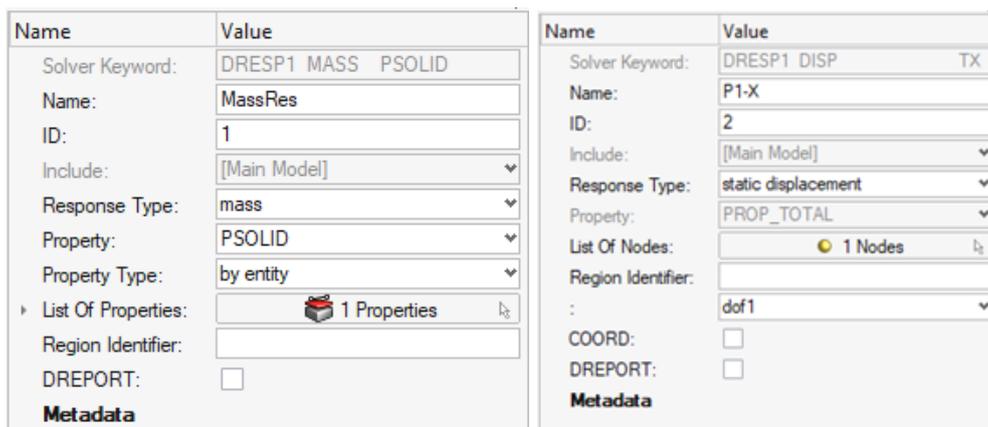
zazione. Inoltre, per un controllo più approfondito sull'analisi, vengono introdotti anche i controlli di ottimizzazione.

3.2.1 Response, Obiettivo e Vincoli

Come accennato in precedenza, l'ottimizzazione comporta una riduzione della rigidità complessiva della struttura, con conseguente aumento degli spostamenti nodali rispetto a quelli rilevati nell'analisi statica. Tale incremento deve quindi essere limitato al valore nominale ricavato nel paragrafo 2.2, tabella 2.2. Tuttavia, per imporre un valore massimo allo spostamento è necessario dapprima definire le "response".

Le "response" rappresentano i risultati ottenuti dall'analisi del modello dopo ogni iterazione nel processo di ottimizzazione. Queste possono essere utilizzate sia come obiettivi che come vincoli, a seconda delle esigenze dell'utente. Nel caso specifico, tra le diverse responses fornite da *OptiStruct*, verranno impiegate la massa, come obiettivo, e lo spostamento statico in un determinato nodo, come vincolo. La definizione delle responses è effettuata con il comando "*Optimize* → *Targets* → *Responses*", la cui finestra permette la definizione della tipologia (RTYPE) e degli attributi.

Nello specifico, la response relativa alla massa in figura 3.3a è associata alle proprietà PSOLID, il che significa che il valore riportato dalla response corrisponderà alla massa complessiva di tutti gli elementi a cui queste proprietà sono applicate. Per quanto riguarda le response relative agli spostamenti in figura 3.3b, invece, gli attributi da definire includono gli ID dei nodi di interesse e il grado di libertà lungo il quale deve essere misurato lo spostamento.



(a) Massa degli elementi solidi.

(b) Spostamento del nodo 1 in direzione x.

Figura 3.3: Finestre per la definizione delle prime due responses.

L'avanzamento di un'ottimizzazione dipende in maniera diretta dalla funzione obietti-

vo che le si impone: tra le opzioni offerte dal software, è possibile selezionare vari obiettivi come, ad esempio, la massimizzazione o minimizzazione di una specifica response. Questo tipo di informazione viene definita e memorizzata all'interno dell'entità DESOBJ. L'operazione si esegue attraverso un comando dedicato nell'interfaccia (*Optimize* \rightarrow *Targets* \rightarrow *Objectives*), che apre la schermata illustrata in figura 3.4. Da qui, si può scegliere una singola response come funzione obiettivo oppure selezionare un insieme di response, nel caso in cui l'obiettivo sia la somma dei minimi quadrati delle stesse.

Name	Value
Solver Keyword:	DESOBJ(MIN)
Name:	objective
ID:	1
Include:	[Main Model]
Objective Type:	Minimize
Response Id:	 (1) MassRes
Metadata	

Figura 3.4: Schermata per la definizione dell'obiettivo di ottimizzazione.

Nel caso specifico, la funzione obiettivo selezionata è la minimizzazione della response, che in questo studio corrisponde alla massa degli elementi solidi. Pertanto, l'obiettivo dell'ottimizzazione è di ridurre al minimo la massa degli elementi solidi. Dunque, *OptiStruct* risolve un problema di minimizzazione della funzione obiettivo del tipo:

$$\min(f(x) = f(x_1, x_2, \dots, x_n)) \quad (3.1)$$

Dove $f(x)$ è la funzione obiettivo e x_i con $i = 1, 2, \dots, n$ sono le densità dei singoli elementi, ovvero le variabili di design.

I vincoli, invece, definiscono i limiti superiori o inferiori di una determinata response. Sono definiti all'interno dell'entità DCONST e la loro disponibilità cambia a seconda del tipo di ottimizzazione e dell'obiettivo fissato. La definizione dei vincoli, analogamente a quanto avviene per le response e gli obiettivi, si effettua tramite l'utilizzo di un comando specifico dell'interfaccia grafica (*Optimize* \rightarrow *Targets* \rightarrow *Constraints*), il quale apre la finestra riportata in figura 3.5. In questa finestra è possibile selezionare la response da vincolare, impostare i limiti, sia inferiori che superiori, entro cui tale response deve rimanere e specificare le condizioni di carico in cui applicare il vincolo.

Nel caso illustrato è stato imposto un limite massimo di spostamento di 6,25mm lungo la direzione x per il punto 1, quando la struttura è soggetta alla prima condizione di carico.

Name	Value
Solver Keyword:	DCONSTR
Name:	C1-X
ID:	1
Include:	[Main Model]
▶ Response:	(2) P1-X
▶ List of Loadsteps:	1 Loadsteps
Lower Options:	<OFF>
▼ Upper Options	
Upper Options:	Upper bound
Upper Bound:	0.00625
PROB:	
Metadata	

Figura 3.5: Schermata per la definizione dei vincoli di ottimizzazione.

La definizione dei vincoli di progetto viene poi ripetuta per ogni nodo e per ciascuna condizione di carico corrispondente.

In generale, l'ottimizzazione topologica procede fino a quando la variazione della struttura tra un'iterazione e l'altra è trascurabile. Infatti, quando la variazione percentuale della response obiettivo scende al di sotto di un determinato limite, l'ottimizzazione viene considerata convergente. Questo limite è regolato dal parametro di controllo "DOPT-PRM,OBJTOL" (paragrafo 3.2.3), il cui valore di default varia a seconda del tipo di ottimizzazione usato. Tuttavia, è importante notare che la convergenza dell'ottimizzazione non garantisce automaticamente il rispetto di tutti i vincoli. Infatti, è possibile che il processo raggiunga la convergenza anche se i vincoli definiti non sono stati pienamente soddisfatti.

Un altro aspetto fondamentale dell'ottimizzazione strutturale è il fatto che al termine dell'ottimizzazione vi sia uno o più vincoli attivi. Un vincolo si definisce attivo quando il valore della response associata si avvicina, con un margine di $\pm 1\%$, al limite massimo imposto dal vincolo stesso. La struttura è tanto più efficiente quanto maggiore è il numero di vincoli attivi. Tuttavia, la condizione in cui risultano tutti attivi è difficilmente raggiungibile, soprattutto quando sono presenti molti vincoli da rispettare. Se si impone un solo vincolo, infatti, l'ottimizzazione tenderà a distribuire il materiale in modo tale da far sì che la response si avvicini al limite, attivando il vincolo. In presenza di più vincoli, invece, è possibile che almeno uno di questi risulti associato a una condizione di carico più favorevole per la struttura, e che l'aggiunta di materiale per soddisfare un'altra condizione più sfavorevole non consenta l'attivazione di tale vincolo. Questo fenomeno diventa più frequente con l'aumento del numero di vincoli imposti.

3.2.2 Variabili di Design

Nell'ottimizzazione topologica la densità del materiale varia tra 0 e 1, dove i valori 0 e 1 rappresentano rispettivamente un elemento vuoto o pieno. Tuttavia, poiché il calcolo con variabili discrete risulterebbe particolarmente complesso e costoso in termini computazionali, il metodo delle densità utilizza quest'ultima come variabile di design continua, che può assumere qualsiasi valore compreso tra 0 e 1. Questi valori intermedi rappresentano densità parziali che, in base alla soglia di conversione (vedi paragrafo 3.3.1) definita dall'utente, possono essere interpretati come solidi o vuoti.

Il metodo delle densità si basa sull'assunzione che la rigidità di un elemento sia direttamente proporzionale alla sua densità, in linea con le conoscenze dei materiali: ad esempio, l'acciaio, con una densità maggiore rispetto all'alluminio, possiede anche una maggiore rigidità. Dunque, le caratteristiche del materiale vengono ridotte in funzione della variazione di densità.

Nell'ottimizzazione topologica l'uso di variabili di design continue comporta la presenza di numerosi elementi con densità intermedia, valori che non hanno un reale significato fisico in un contesto di ottimizzazione topologica, poiché nella realtà il risultato finale consiste in una distribuzione di materiale solido all'interno del design space. Per ridurre questo fenomeno, *OptiStruct* applica tecniche di penalizzazione che scoraggiano l'assegnazione di densità intermedie, favorendo invece valori vicini agli estremi, ovvero 0 (vuoto) e 1 (pieno).

Queste tecniche sono SIMP, RAMP e Polynomial. Quella usata maggiormente è la prima e basa la riduzione di rigidità in funzione dell'equazione 3.2).

$$\tilde{K}(\rho) = \rho^p K \quad (3.2)$$

Dove, \tilde{K} e K rappresentano rispettivamente la matrice di rigidità penalizzata e quella reale dell'elemento, ρ è la densità e p è il fattore di penalizzazione. Quest'ultimo è sempre maggiore di 1 e, all'aumentare dello stesso, aumenta la discretizzazione della struttura verso densità più vicine agli estremi.

In *OptiStruct*, la quantità relativa di elementi con densità intermedia può essere stimata tramite l'indice di discretezza. Questo parametro viene ricalcolato dal software a ogni iterazione, rapportando la massa degli elementi con densità compresa tra 0,9 e 1 alla massa complessiva della struttura ottimizzata come riportato nella equazione 3.3.

$$D = \frac{\sum_{i=1}^n (\rho_i V_i)_{per\ 0.9 \leq \rho_i \leq 1}}{\sum_{i=1}^n (\rho_i V_i)} \quad (3.3)$$

Dove ρ_i e V_i rappresentano rispettivamente la densità e il volume dell'elemento i -esimo. Ripetendo questo calcolo per tutti gli intervalli di densità si ottiene la tabella degli indici di discretezza, poi riportata nel file *.out*. L'obiettivo del progettista è quindi quello di massimizzare l'indice di discretezza. A tal fine, si utilizzano specifici parametri di controllo dell'ottimizzazione o si procede con un affinamento della mesh per ottenere una maggiore precisione nei risultati.

Il valore di default del fattore di penalizzazione e la tecnica adottata dipendono dal tipo di elementi coinvolti nell'ottimizzazione. Nel caso specifico, poiché l'ottimizzazione riguarda esclusivamente elementi solidi, viene utilizzata la tecnica di penalizzazione basata sul metodo SIMP (Solid Isotropic Material with Penalization), con un fattore di penalizzazione predefinito pari a 2.

L'effettiva definizione della variabile di design all'interno del modello avviene generalmente nell'entità DESVAR, tuttavia esiste un'altra entità denominata DTPL (*Optimize* \rightarrow *DesignVariables* \rightarrow *Topology*), dedicata alla definizione di variabili di design specifiche per ottimizzazione topologica. In questa entità si seleziona il tipo di proprietà (PSOLID), la proprietà specifica da ottimizzare tra quelle di tale tipologia, e si imposta un vincolo di simmetria rispetto al piano x-z. Infatti, vista l'asimmetria delle condizioni al contorno, se si volesse ottenere un design simmetrico bisognerebbe imporre tale condizione come vincolo costruttivo. Si specifica però che la gestione della simmetria nell'ottimizzazione topologica non è completamente perfetta, poiché il solutore gestisce separatamente tutte le variabili di design.

3.2.3 Controlli di Ottimizzazione

In *HyperMesh* i controlli di ottimizzazione permettono di affinare il processo di ottimizzazione, consentendo di impostare o modificare parametri e criteri specifici che guidano il solutore verso soluzioni più efficienti. Uno degli strumenti a disposizione per la gestione di questi parametri è la card DOPTPRM, che offre diverse opzioni per controllare sia parametri generali, sia parametri specifici legati a particolari tipi di ottimizzazione.

In questa fase l'attenzione è rivolta solo ad alcune opzioni, ovvero quelle che hanno un impatto più significativo sul risultato finale, poiché la maggior parte di quelle messe a disposizione dal solutore riguardano aspetti di dettaglio non trattati in questa tesi.

- CHECKER: verifica l'eventuale presenza di una distribuzione a scacchiera degli elementi, fenomeno che può generare vuoti all'interno di un componente solido durante il filtraggio. In additive manufacturing, tali vuoti verrebbero riempiti dalle polveri metalliche durante la deposizione dei layer, aumentando il peso della struttura senza apportare benefici alla sua funzione strutturale.

- **MMCHECK**: assicura che non si verifichi una distribuzione a scacchiera delle densità degli elementi. Tuttavia, l'uso di questo parametro può comportare un effetto indesiderato, ovvero un aumento del numero di elementi con densità intermedie nelle zone di transizione tra solido e vuoto.
- **DISCRETE**: permette di controllare manualmente il fattore di penalizzazione e , di conseguenza, la distribuzione finale del materiale (figure 3.6a e 3.6b). Maggiore il valore di questo parametro, minore il numero di elementi la cui densità è compresa tra 0 e 1.

Density	%	Density	%
0.0-0.1	51.5	0.0-0.1	51.8
0.1-0.2	3.5	0.1-0.2	2.7
0.2-0.3	3.9	0.2-0.3	3.3
0.3-0.4	2.5	0.3-0.4	1.6
0.4-0.5	2.9	0.4-0.5	2.1
0.5-0.6	7.4	0.5-0.6	7.3
0.6-0.7	2.5	0.6-0.7	1.7
0.7-0.8	4.2	0.7-0.8	3.3
0.8-0.9	4.3	0.8-0.9	3.2
0.9-1.0	17.4	0.9-1.0	23.0

(a) DISCRETE=1

(b) DISCRETE=4

Figura 3.6: Esempio dell'effetto del parametro DISCRETE sulle caratteristiche della struttura.



(a) Senza MINDIM.

(b) MINDIM=60.

(c) MINDIM=90.

Figura 3.7: Effetto del parametro MINDIM sulle caratteristiche della struttura.

- **MINDIM**: questo comando penalizza la formazione di elementi strutturali con dimensioni inferiori al valore specificato nella card di controllo. Tuttavia, nonostante la presenza di questo parametro, possono comunque emergere componenti di dimensioni ridotte se risultano fondamentali per la trasmissione dei carichi. Pertanto, MINDIM funge principalmente da controllo qualitativo piuttosto che quantitativo. Inoltre, il parametro è sempre attivo in presenza di vincoli di manifattura, come il vincolo di simmetria planare applicato in questo caso. Il valore di default definito in automatico dal software è pari a tre volte la dimensione caratteristica degli elementi della mesh.

3.3 Ottimizzazione Parametrica della Struttura

In questa sezione viene descritto il processo iterativo con cui i parametri di controllo definiti in precedenza vengono progressivamente modificati al fine di individuare quelli che permettono di ottenere il peso ottimale della struttura. Partendo dai valori predefiniti dal software, ogni parametro viene modificato singolarmente fino a trovare la combinazione più efficiente. Vengono inoltre illustrate le fasi di post-processing necessarie per ottenere la struttura finale pronta per la stampa 3D.

Si utilizza l'interfaccia grafica di *HyperMesh* per la modifica dei parametri e il filtraggio dei risultati, il solutore *OptiStruct*, e le suite *HyperGraph* e *HyperView* per l'analisi e la visualizzazione dei risultati.

3.3.1 Fasi di Elaborazione dei Risultati

La grandezza principale per valutare la qualità di un'ottimizzazione rispetto a un'altra è la massa. Tuttavia, la massa risultante dall'ottimizzazione non corrisponde a quella effettiva del componente finale destinato alla stampa, a causa della presenza di densità intermedie. In effetti, la massa ottenuta dall'ultima iterazione di un'ottimizzazione topologica è la somma delle masse di tutti gli elementi del design space, ciascuno dei quali ha una densità intermedia. Inoltre, non è garantito che l'ultima iterazione sia quella con la massa più bassa. Per questo motivo è necessario identificare l'iterazione che presenta la massa minima, a condizione che in essa la percentuale di violazione dei vincoli sia pari a zero.

Questa operazione si svolge utilizzando *HyperGraph*, che consente di visualizzare graficamente l'andamento della massa del design space e della percentuale di violazione dei vincoli in funzione delle iterazioni. Dal grafico, è possibile individuare il minimo della funzione obiettivo, purché non presenti alcuna violazione dei vincoli. Se il minimo assoluto presenta violazioni, si procede al successivo minimo locale, ripetendo il processo fino a trovare un minimo valido. Nel caso riportato in figura 3.8, il minimo globale presenta una violazione del 52%, il secondo minimo locale ha una violazione del 18%, quindi la condizione viene soddisfatta al terzo minimo locale, corrispondente all'ultima iterazione. A questo punto è possibile selezionare l'iterazione più vantaggiosa in termini di peso e determinare il valore di densità alla quale avviene la transizione tra solido e vuoto. Questo processo si svolge iterativamente attraverso tre passaggi principali:

1. Creazione di un set di elementi che includa tutti quelli caratterizzati da densità inferiori a un determinato valore, il quale viene poi esportato nel modello iniziale.

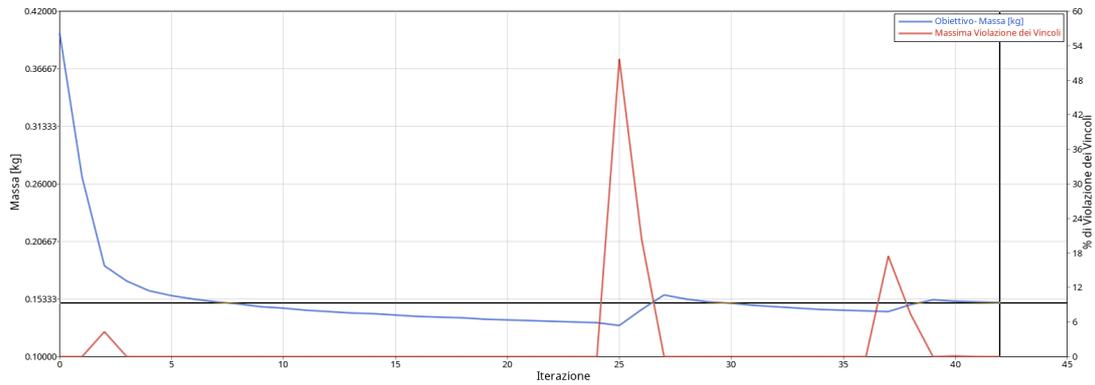


Figura 3.8: Caption

2. Nel modello, si eliminano tutti gli elementi appartenenti al set appena creato. Poiché un elevato numero di elementi viene rimosso, è fondamentale verificare che non ci siano elementi isolati o collegati alla struttura principale solo tramite uno degli spigoli. In tal caso, al fine di evitare la labilità della struttura e quindi errori nel calcolo dei risultati, bisogna eliminare anch'essi.
3. Si esegue un'analisi statica della struttura filtrata per accertarsi che non vi siano violazioni dei vincoli. Se si riscontrano violazioni, il valore della soglia di transizione definito al primo punto viene ridotto; al contrario, se non ci sono vincoli attivi, la soglia viene aumentata.

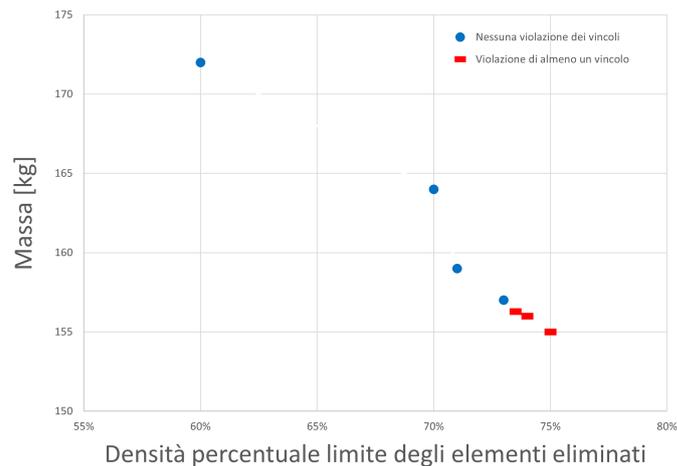


Figura 3.9: Andamento della massa in funzione della percentuale di filtraggio degli elementi.

Al termine del processo iterativo si ottiene una struttura a densità discreta, con una massa minimizzata in base ai parametri di ottimizzazione definiti. Questo processo viene ripetuto per ogni risultato ottenuto dalle ottimizzazioni, poiché non vi è garanzia che la

struttura con la massa minima risultante dalle ottimizzazioni, conduca necessariamente alla struttura più leggera una volta completata la discretizzazione degli elementi.

In figura 3.9 è riportato, a titolo esemplificativo, l'andamento della massa successivo alla conversione della struttura. Si osserva una progressiva riduzione della massa con l'aumentare della percentuale di filtraggio, dovuta alla diminuzione del numero di elementi con densità intermedia convertiti in materiale solido. In questo caso, si considera la massa corrispondente all'ultima percentuale utile prima del verificarsi di violazioni dei vincoli, che nella figura risulta essere pari al 74%.

3.3.2 Risultati

A valle dell'elaborazione si enunciano i risultati, osservando separatamente l'effetto dei vari parametri di controllo.

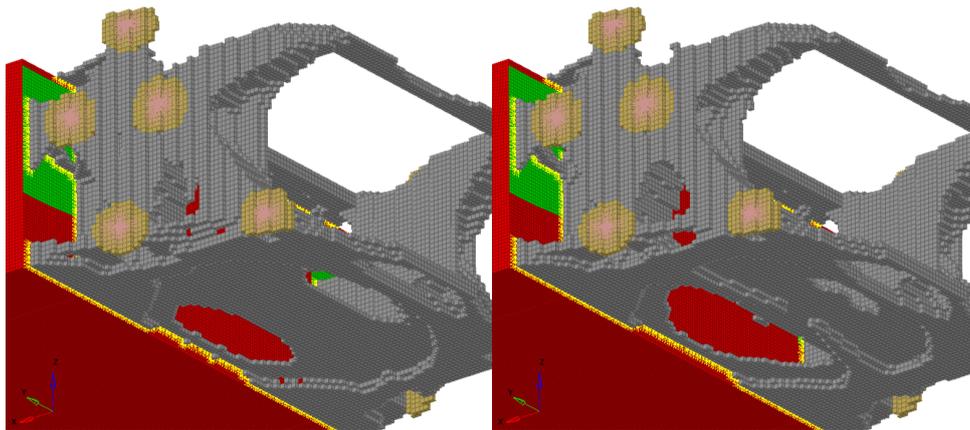
DOPTPRM,DISCRETE Il parametro di controllo DISCRETE consente, come precedentemente accennato, di regolare manualmente il fattore di penalizzazione applicato alla densità. Il suo effetto più evidente si osserva sulla massa: teoricamente, la struttura più efficiente è quella in cui ogni elemento risponde proporzionalmente allo sforzo a cui è sottoposto. Questa condizione si ottiene quando si mantengono numerose densità intermedie, minimizzando il peso complessivo. Tuttavia, penalizzando queste densità intermedie, la struttura diventa meno efficiente e, a parità di prestazioni, risulta più pesante. Infatti, in tabella 3.1, si osserva che la massa risultante a valle dell'ottimizzazione aumenta al crescere del parametro in esame.

Modificando questo parametro, però, non si influisce solo sulla distribuzione della variabile di design nel modello, ma anche sulla velocità di convergenza dell'ottimizzazione: in tabella, si nota che un incremento del valore di DISCRETE riduce il numero di iterazioni necessarie per raggiungere la convergenza. Questo comportamento in fase di ottimizzazione è dovuto al fatto che una penalizzazione più estrema porta ad una riduzione più veloce della densità. Ciò si traduce nella necessità di un minor numero di iterazioni per arrivare a target.

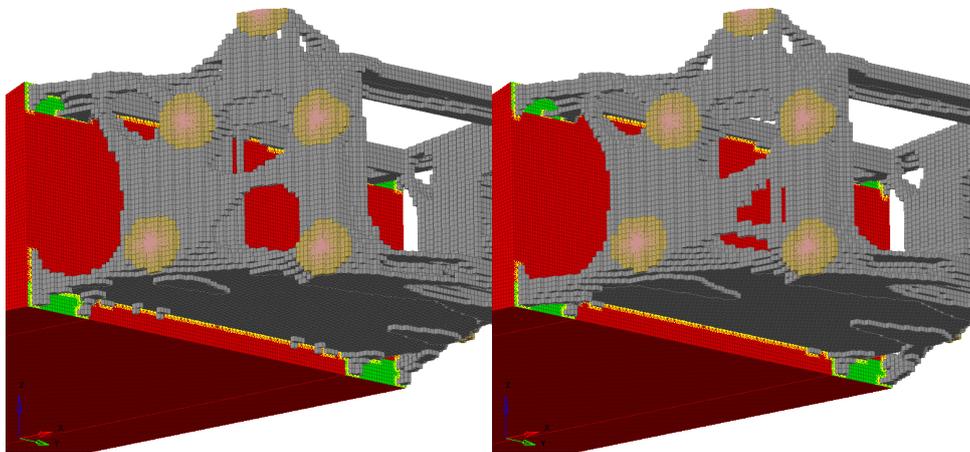
RUN	DISCRETE	N. iterazioni	Massa [kg]	Massa convertita [kg]	% di conversione
1	1	42	150	156	65%
2	2	34	155	155	74%
3	3	30	159	160	74%
4	4	29	161	157	79%

Tabella 3.1: Effetto del parametro DOPTPRM,DISCRETE sulle caratteristiche del componente.

Facendo ancora riferimento alla tabella 3.1, tra la prima e la seconda iterazione si osserva un significativo aumento sia della percentuale di conversione, sia della discretezza. Queste variazioni, nel complesso, riducono il numero di elementi trattenuti, comportando così una diminuzione della massa totale. Infatti, in una struttura con un maggior numero di elementi a densità intermedia, è spesso necessario ridurre la percentuale di filtraggio per mantenere un numero sufficiente di elementi in grado di soddisfare i vincoli di ottimizzazione.



(a) Front frame.



(b) Rear frame.

Figura 3.10: Dettaglio delle strutture ottenute dalla conversione delle ottimizzazioni topologiche per la seconda (a sinistra) e la terza simulazione (a destra).

Al contrario, tra la seconda e la terza iterazione si osserva un aumento della discretezza senza variazioni nella percentuale di filtraggio. Di conseguenza, a parità di filtraggio, aumenta il numero di elementi con densità sopra la soglia. In tali condizioni si mantengono più elementi nella conversione e si incrementa la massa.

Con un ulteriore aumento del valore di DISCRETE, la discretezza e la massa finale dell'ottimizzazione aumentano ancora, ma poiché la percentuale di conversione cresce anch'essa, l'effetto complessivo sulla struttura finale si traduce in una massa convertita inferiore rispetto al caso precedente. Tuttavia, in questa quarta iterazione non si ottiene una massa inferiore rispetto alla configurazione con DISCRETE pari a 2.

In conclusione, dai valori riportati in tabella emerge che, sebbene la massa minima si riscontri con il fattore di penalizzazione più basso, la configurazione ottimale per la struttura finale corrisponde alla seconda iterazione. Pertanto, per le fasi successive, si procede mantenendo il valore di DOPTPRM,DISCRETE pari a 2.

Il confronto tra due strutture risultanti da due simulazioni diverse è riportato nelle figure 3.10. Si osserva che, complessivamente, gli elementi che costituiscono gran parte del modello variano in maniera impercettibile in dimensioni e forma. Al contrario, le feature di dimensioni ridotte tendono a subire modifiche più evidenti di dimensione o di forma, fino a essere in alcuni casi eliminate.

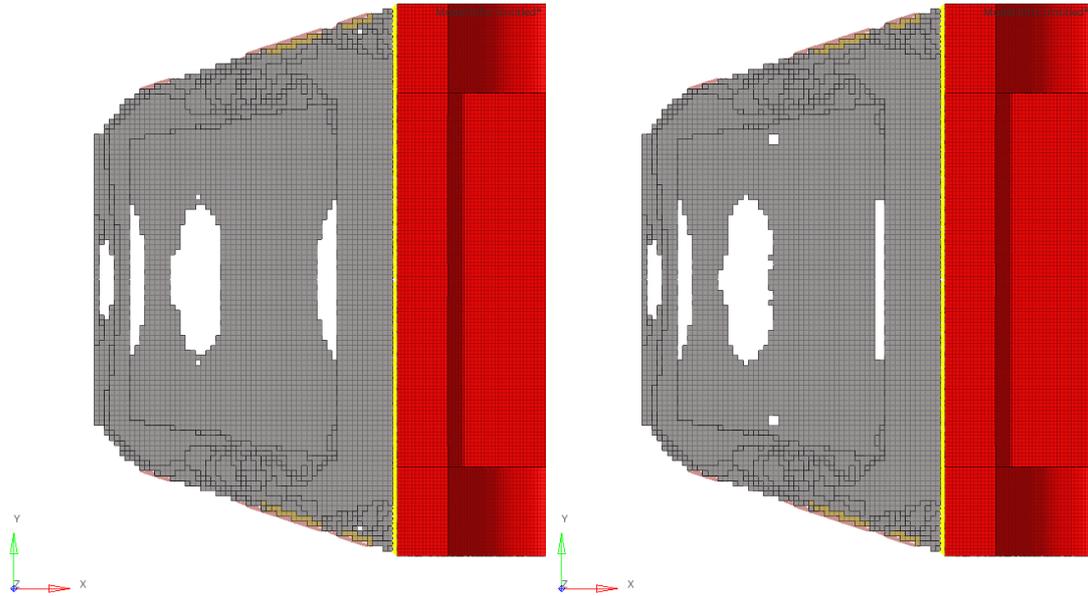
DOPTPRM,MINDIM Le tecnologie attualmente disponibili nell'ambito del Additive Manufacturing consentono la creazione di parti con un diametro nettamente inferiore alla dimensione caratteristica dell'elemento. Ne consegue che l'analisi eseguita con valori crescenti del parametro MINDIM ha il solo fine di verificarne l'effetto sulla massa finale del componente.

Tabella 3.2: Effetto del parametro DOPTPRM,MINDIM sulle caratteristiche del componente.

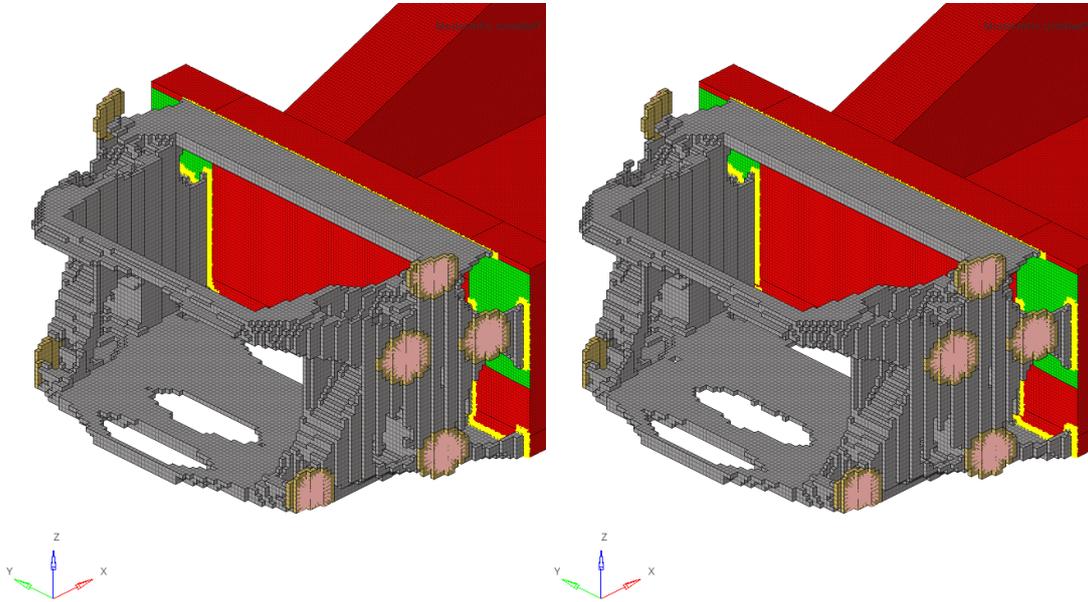
MINDIM	N. iterazioni	Massa [kg]	Massa convertita [kg]	% di conversione
30	34	155	155	74
40	33	170	156	73

In seguito alla conversione del modello utilizzando un valore di MINDIM superiore a quello di default, si è osservato un incremento della massa con l'aumentare di tale parametro. Inoltre, nelle figure 3.11 e 3.12 si può notare che, ad eccezione di un dettaglio minimo visibile nella vista dall'alto, le distribuzioni di materiale risultano pressoché identiche in entrambi i casi. Questa somiglianza è confermata anche dai dati relativi alla massa convertita, che risulta praticamente invariata nelle due configurazioni.

Pertanto, al fine di perseguire l'obiettivo principale di minimizzazione della massa, è stato scelto di adottare il valore di default di MINDIM, ovvero $MINDIM = 30$, in quanto garantisce la massa più contenuta.



(a) Vista dall'alto front frame. MINDIM=30 a sinistra; MINDIM=40 a destra.



(b) Vista inclinata front frame. MINDIM=30 a sinistra; MINDIM=40 a destra.

Figura 3.11: Dettaglio del front frame ottenuto dalla conversione delle ottimizzazioni topologiche per diversi valori di MINDIM.

DOPTPRM,CHECK/MMCHECK L'attivazione dei parametri di rilevazione e correzione per gli elementi distribuiti a scacchiera è ridondante quando il valore del MINDIM è inferiore rispetto alla dimensione caratteristica della mesh. Nel caso analizzato, MINDIM è stato definito con un valore superiore a questa soglia, tuttavia, conseguentemente all'imposizione del vincolo di simmetria, questa opzione rimane attiva di default. Dunque, è già presente nei risultati riportati finora.

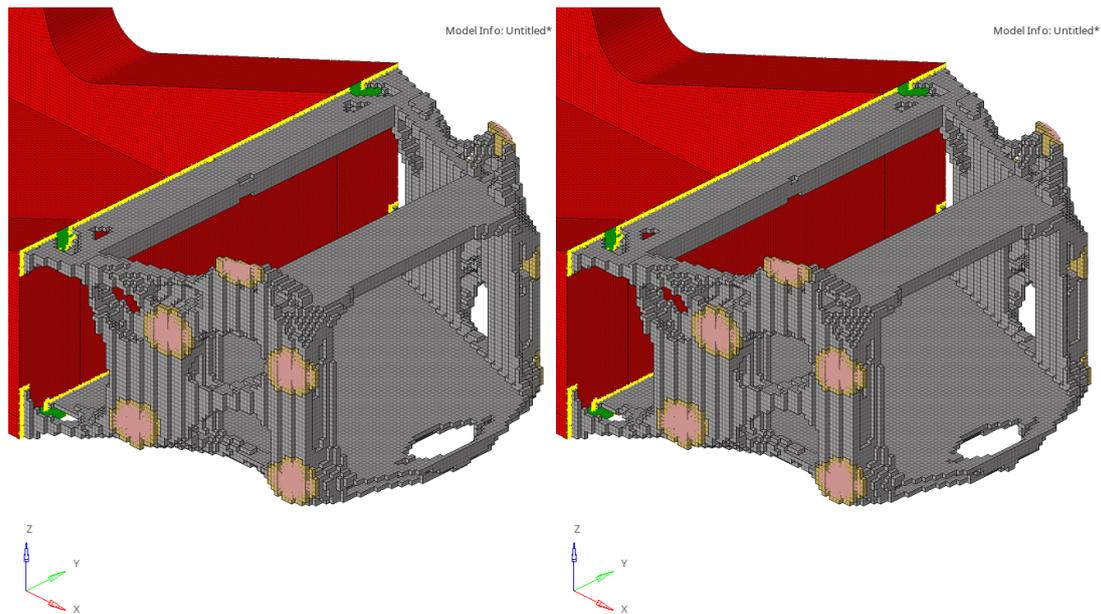


Figura 3.12: Dettaglio del rear frame ottenuto dalla conversione delle ottimizzazioni topologiche per diversi valori di MINDIM. MINDIM=30 a sinistra; MINDIM=40 a destra.

3.4 Risultati dell'Ottimizzazione Topologica

Osservando la struttura risultante dall'ottimizzazione topologica si nota immediatamente la particolare distribuzione di materiale. Quest'ultimo, infatti, risulta disposto secondo i percorsi di carico principali, ovvero su quei percorsi lungo i quali le tensioni si sviluppano in maniera più evidente. Tale configurazione ha portato ad una massa finale dell'intera struttura in metallo pari a $155kg$, pur rispettando tutti i vincoli progettuali imposti al modello.

Come osservabile in tabella 3.3, questa tecnica di ottimizzazione ha portato all'attivazione di un solo vincolo. In base a quanto detto precedentemente, nonostante l'ottimizzazione topologica fornisca risultati particolarmente più performanti rispetto a quelli ottenuti tramite manifattura tradizionale, la struttura risulta leggermente sovradimensionata per

tutte le condizioni di carico ad eccezione di quella che presenta il vincolo attivo. L'obiettivo delle fasi successive, dunque, sarà proprio la ricerca di un design in grado di spingere ancor più al limite il design della struttura.

Tabella 3.3: Response ottenute dall'ottimizzazione topologica.

Response	Response Label	Valore della Response		Vincolo di riferimento	Violazione percentuale	Attivazione del vincolo
2	P1-X	5,66E+00	<	6,25E+00	0.0	
3	P1-Y	6,00E+00	<	6,02E+00	0.0	A
4	P1-Z	5,55E+01	<	5,82E+01	0.0	
5	P2-X	1,30E+01	<	2,06E+01	0.0	
6	P2-Y	4,45E+00	<	4,96E+00	0.0	
7	P2-Z	2,87E+01	<	2,97E+01	0.0	
8	P3-X	4,15E+01	<	5,92E+01	0.0	
9	P3-Y	1,95E+01	<	2,14E+01	0.0	
10	P3-Z	2,69E+01	<	2,82E+01	0.0	
11	P4-X	1,42E+01	<	2,20E+01	0.0	
12	P4-Y	5,99E+00	<	6,42E+00	0.0	
13	P4-Z	2,33E+01	<	2,97E+01	0.0	
14	P5-X	4,92E+00	<	6,40E+00	0.0	
15	P5-Y	3,33E+00	<	3,53E+00	0.0	
16	P5-Z	2,57E+01	<	2,76E+01	0.0	
17	P6-X	5,39E+00	<	7,98E+00	0.0	
18	P6-Y	2,70E+00	<	2,73E+00	0.0	
19	P6-Z	2,85E+01	<	3,57E+01	0.0	
20	P7-X	1,37E+01	<	1,83E+01	0.0	
21	P7-Y	1,12E+01	<	1,25E+01	0.0	
22	P7-Z	2,97E+01	<	3,51E+01	0.0	
23	P8-X	4,17E+01	<	4,66E+01	0.0	
24	P8-Y	4,65E+01	<	5,31E+01	0.0	
25	P8-Z	3,71E+01	<	4,06E+01	0.0	
26	P9-X	1,35E+01	<	1,59E+01	0.0	
27	P9-Y	7,69E+00	<	8,56E+00	0.0	
28	P9-Z	4,05E+01	<	4,44E+01	0.0	
29	P10-X	6,08E+00	<	6,22E+00	0.0	
30	P10-Y	5,38E+00	<	5,59E+00	0.0	
31	P10-Z	4,45E+01	<	4,68E+01	0.0	

4. Ottimizzazione di Strutture Lattice

4.1 Basi per l'Ottimizzazione Lattice

L'idea che sta alla base dell'ottimizzazione lattice è quella di preservare le densità intermedie che altrimenti verrebbero cancellate, attraverso l'uso di celle predefinite. Definiti i valori corrispondenti al limite inferiore e al limite superiore, infatti, tutti gli elementi che appartengono alla fascia di densità comprese tra questi due valori ($LB < x < UB$), vengono sostituiti con le celle lattice. Questa tipologia di ottimizzazione è dedicata alla produzione tramite Additive Manufacturing, visto l'elevato numero di elementi che per dimensioni, densità e posizione risulterebbero impossibili da ottenere tramite metodi di manifattura classici [8].

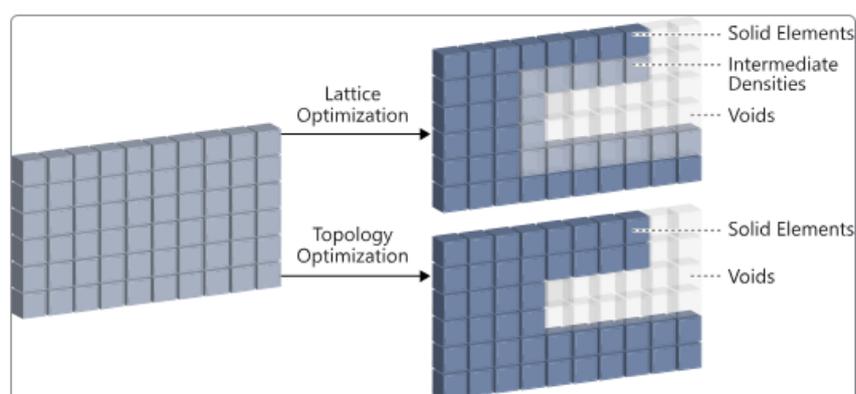


Figura 4.1: Differenze tra la fase 1 dell'ottimizzazione lattice e l'ottimizzazione topologica.

Il fine ultimo è ottenere una struttura in cui il materiale è distribuito nella maniera più efficiente possibile. Infatti, nell'ottimizzazione topologica gli elementi vengono portati a densità 0 o 1, forzando la decisione tra i due estremi pieno o vuoto. Di conseguenza si

potrebbe riscontrare una variazione significativa rispetto alla soluzione proposta dal solutore. Al contrario, per l'ottimizzazione lattice vi è una fase di transizione tra il pieno e il vuoto rappresentata proprio dagli elementi solidi convertiti in barre. La soluzione dell'ottimizzazione lattice si ottiene attraverso due fasi principali: *Phase I* e *Phase II*.

4.1.1 Phase I

La prima fase del processo prevede un'ottimizzazione topologica, in cui la principale differenza rispetto all'approccio classico consiste nell'adozione di un basso fattore di penalizzazione. Questo favorisce la presenza di elementi con densità intermedie, la cui presenza risulta particolarmente favorevole in questo tipo di ottimizzazione. Al termine della *Phase I* si procede alla conversione del modello. Questa operazione comporta l'eliminazione degli elementi la cui variabile di design risulta inferiore al valore di soglia LB, la generazione di celle lattice per gli elementi con densità compresa tra LB e UB, e la conversione in elementi solidi di quelli la cui densità supera il valore UB.

A seguito di questa conversione, *OptiStruct* crea una serie di entità utili alla definizione delle beams che compongono la struttura lattice.

Trattandosi di un nuovo tipo di elemento, con comportamento e formulazione diversi rispetto ai solidi precedenti, è necessario definire delle proprietà specifiche. Delle proprietà PBEAML vengono generate automaticamente dal software per ogni elemento trave creato. Le caratteristiche principali di tali elementi, illustrate in figura 4.2, sono descritte nella sezione 1.2.1. La generazione automatica di una proprietà per ogni beam è cruciale, poiché nella seconda fase l'ottimizzazione dimensionale gestisce il modello in modo asimmetrico, nonostante quest'ultimo sia stato ottimizzato con vincolo di simmetria in fase di ottimizzazione topologica.

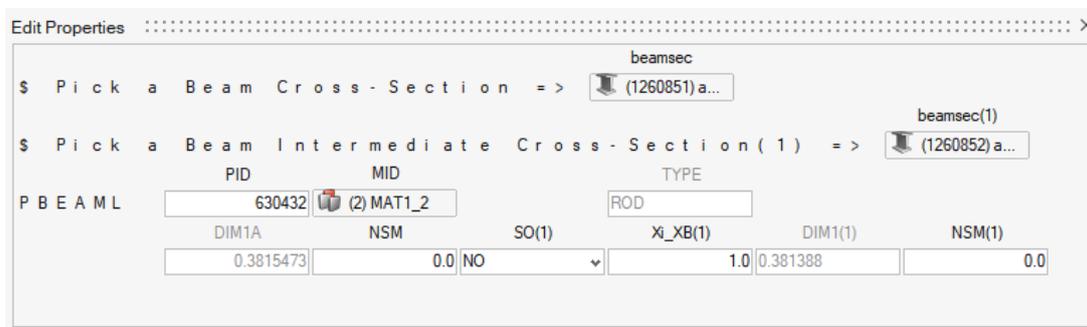


Figura 4.2: Esempio di card per la definizione della proprietà di una beam.

Definito il comportamento dei nuovi elementi e fissate le proprietà associate, *OptiStruct* procede a impostare le variabili di design per la seconda fase dell'ottimizzazione lattice. Questa fase è incentrata sull'ottimizzazione dimensionale delle travi, per cui le sezioni ai

nodi fungeranno da variabili di design. In particolare, il software assegna una variabile di design a ciascuna sezione situata all'estremo di una trave.

La card di definizione delle variabili di design presenta al suo interno, a parte l'ID e il nome della variabile stessa, tre campi fondamentali:

- **INIT** rappresenta il valore iniziale della variabile, ovvero quello ottenuto dalla conversione dell'elemento, in funzione della sua densità, in una cella lattice. I raggi che definiscono le sezioni agli estremi delle beams sono direttamente proporzionali al valore di densità assegnato all'elemento durante la *Phase I*.
- **LB** rappresenta il limite minimo che la variabile di design può raggiungere. Di default è uguale a $UB \cdot 10^{-6}$.
- **UB**, al contrario, rappresenta il limite superiore raggiungibile dalla variabile di design, anch'esso impostato di default dal software.

	ID	NAME	INIT	LB	UB
D E S I G N V A R	78991	DV.78990	0.3610045	2.5942e-06	2.594242

Figura 4.3: Esempio di card per la definizione della variabile di design in un nodo.

La terza entità generata da *OptiStruct* in questa fase definisce una relazione lineare tra una variabile di design e la relativa proprietà, chiamata DVPREL (Design Variable-Property Relation). Questa relazione lineare è espressa dall'equazione 4.1.

$$P_i = C_0 + \sum_i (COEF_i * DDVI_i) \quad (4.1)$$

Dove C_0 è una costante, generalmente pari a 0, $COEF_i$ è il coefficiente di linearità, solitamente fissato a 1, e $DDVI_i$ è il valore iniziale della variabile di design. La card di definizione di questa entità è riportata in figura 4.4, e in essa si possono osservare i valori definiti per C_0 , $COEF_i$ ed il riferimento all'ID della variabile di design $DVID_i$ che permette di identificarne il valore iniziale, per poi calcolarne la relazione lineare al fine di ottenere il valore della proprietà selezionata in "PID". Infine, nella card di definizione della DVPREL, in corrispondenza del campo "PNAME" è riportata l'informazione che comunica al software se la variabile di design deve essere assegnata in corrispondenza del nodo A o del nodo B (per l'identificazione dei nodi di un elemento trave si fa riferimento alla figura 2.12).

In sintesi, a ciascuna beam è associata una singola proprietà. A questa proprietà sono poi

collegate due variabili di design distinte, una per il nodo A e una per il nodo B. Queste a loro volta determinano il raggio della sezione circolare della barra in corrispondenza di ciascun nodo. L'associazione tra variabili di design e proprietà avviene tramite una terza entità che gestisce la relazione tra queste. In questo caso, la relazione è lineare con coefficiente 1, il che implica che il valore del raggio utilizzato nella proprietà corrisponde esattamente al valore della variabile di design.

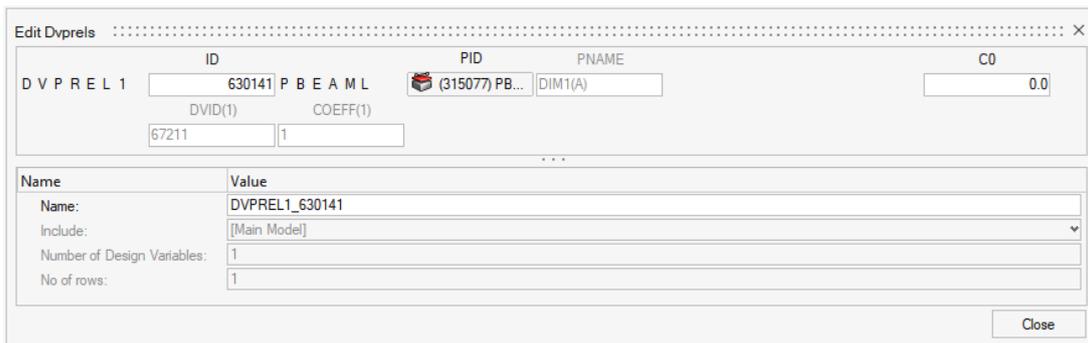


Figura 4.4: Esempio di card per la definizione delle DVPREL.

È importante notare che la stessa variabile di design può essere richiamata da più di una DVPREL, poiché in un nodo possono convergere due o più beams. Di conseguenza, la sezione degli elementi che si incontrano in un nodo sarà uguale per tutti, poiché controllata dalla medesima variabile di design. Ad esempio, in figura 4.5 sono mostrati i riferimenti relativi a una specifica variabile di design; si può notare come essa sia associata a ben otto DVPREL, ognuna delle quali è collegata ad una proprietà diversa.

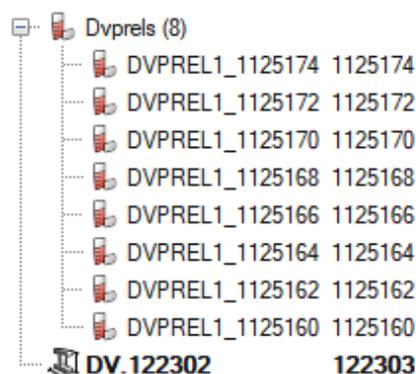


Figura 4.5: Riferimenti di una specifica variabile di design.

È possibile definire più variabili di design in un singolo nodo, implicando l'intersezione di barre con sezioni differenti in quello stesso nodo. Tuttavia, a prescindere dalla fattibilità di questa configurazione, ciò aumenterebbe significativamente il costo computazionale, rendendo il calcolo molto più oneroso, se non proibitivo, in termini di tempo, poiché

comporterebbe la gestione simultanea di un numero notevolmente maggiore di variabili di design.

Parametri di Controllo Analogamente a quanto avviene nell’ottimizzazione topologica classica, si riportano i controlli di ottimizzazione specifici per l’ottimizzazione lattice, i quali sono gestiti attraverso la card di controllo LATPRM.

In genere, la frazione di volume occupata dagli elementi beam all’inizio della seconda fase corrisponde alla densità ottenuta dagli elementi al termine della prima fase di ottimizzazione topologica. La configurazione di questi beam all’interno del volume precedentemente occupato dalla cella lattice viene determinata dall’ingegnere, in funzione di alcune forme predefinite messe a disposizione dal software.

Attraverso l’opzione **Lattice Type** (LT) è possibile scegliere tra diverse tipologie di cella lattice se si trattano elementi esaedrici. Nelle figure 4.6 è possibile osservare tutte le configurazioni disponibili. Tra queste, si prende in considerazione quella corrispondente a LT=1 (figura 4.6a), poiché, è quella che porta alla creazione del minor numero di beams a parità di volume occupato. Questo è un aspetto particolarmente importante per una migliore gestione del modello, soprattutto quando il numero di elementi è elevato. Per tutti gli altri elementi tridimensionali esistono invece celle lattice specifiche (figure 4.7), che non possono essere definite dall’utente.

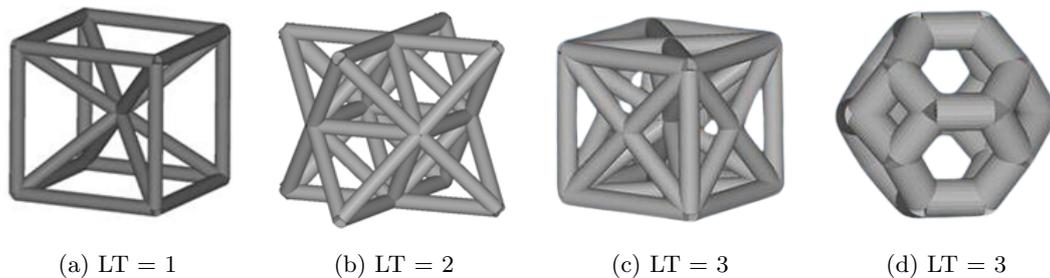


Figura 4.6: Tipologie di celle lattice applicabili all’elemento esaedrico.

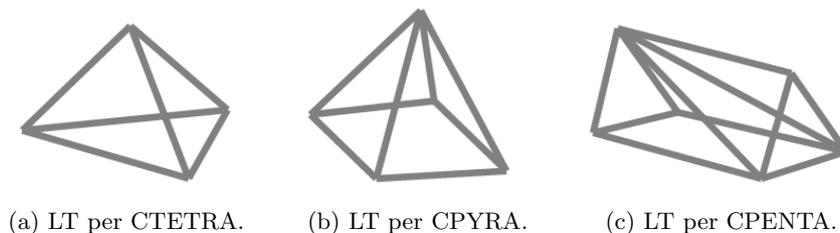


Figura 4.7: LT per elementi diversi dalla cella esagonale.

Oltre alla scelta del tipo di cella lattice, è possibile controllare ulteriori parametri, tra cui:

- **LB**: opzione che regola la soglia di transizione tra densità intermedia e vuoto.
- **UB**: permette di controllare la soglia di passaggio tra la densità intermedia e il solido.
- **CLEAN**: di default su "YES", attiva la pulizia delle barre secondo la funzione di penalizzazione "Heaviside" (si veda la sezione "Lattice Sizing +" nel paragrafo successivo). È possibile, con le altre opzioni, disattivare questa funzione o utilizzare una diversa penalizzazione.
- **POROSITY**: sono disponibili tre opzioni, che controllano direttamente il fattore di penalizzazione. Queste sono HIGH, MED e LOW.

Porosità e Rigidezza In base a quanto già affermato in precedenza, il numero di elementi con densità intermedie dipende dal fattore di penalizzazione applicato, in linea con quanto definito per l'ottimizzazione topologica classica. Questo aspetto è regolato dal parametro di controllo LATPRM, POROSITY, che consente di generare soluzioni con tre diversi livelli di penalizzazione:

- **HIGH**: genera un numero elevato di elementi con densità intermedia. In questa configurazione non viene applicato alcun fattore di penalizzazione alla relazione tra il modulo di Young e la densità.
- **MED**: produce un numero medio di elementi con densità intermedia, applicando un fattore di penalizzazione pari a 1,25. Questa opzione di porosità solitamente genera un numero inferiore di celle lattice rispetto a HIGH.
- **LOW**: comporta un numero relativamente ridotto di elementi a densità intermedia, con un fattore di penalizzazione pari a 1,8, che si avvicina maggiormente al valore di default utilizzato per l'ottimizzazione topologica classica.

Questi parametri influiscono in maniera diretta sul valore calcolato della rigidezza secondo l'equazione 4.2.

$$E = E_0 \rho^p \quad (4.2)$$

Dove E è la rigidezza ottimale dell'elemento ottenuto dalla topologica ad una determinata densità, E_0 è la rigidezza iniziale del materiale, ρ è la densità assegnata all'elemento e p è il fattore di penalizzazione.

Sebbene nella realtà non esista il concetto di rigidezza intermedia, è possibile approssimare questo comportamento sfruttando la variazione delle frazioni di volume occupate dalle beams. Tuttavia, risultati empirici hanno dimostrato che la creazione di zone a densità intermedia riproduce in modo accurato la fisica del problema solo quando il fattore di

penalizzazione è pari a 1,8, ovvero quando il parametro POROSITY è impostato su LOW. In tutti gli altri casi si riscontra una sovrastima della rigidità della struttura lattice. A tal proposito, porosità più elevate dovrebbero generalmente essere prese in considerazione solo nel caso in cui si volessero valutare altri aspetti di performance della struttura, come ad esempio analisi a cedimento per buckling.

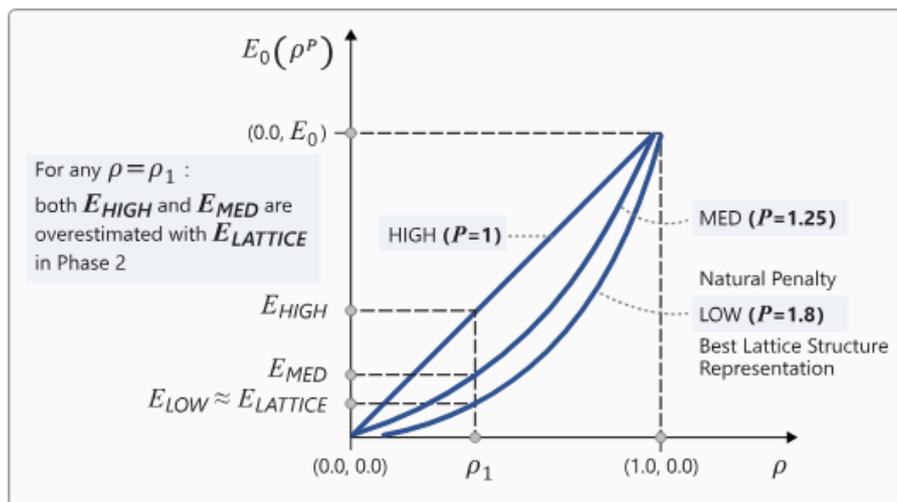


Figura 4.8: Confronto tra le diverse prestazioni di rigidità riscontrate in strutture con porosità LOW, MED e HIGH.

Infine, in base a quanto è possibile osservare in figura 4.8, la sovrastima della rigidità è tanto maggiore quanto minore è il fattore di penalizzazione. Questo fenomeno è tanto più accentuato quanto più la densità dell'elemento appartiene alla fascia intermedia.

4.1.2 Elaborazione del Modello tra Phase I e Phase II

Nel modello risultante dalla *Phase I*, è stata definita una variabile di design per ciascuno dei nodi collegati agli elementi beam. Tuttavia, nonostante l'applicazione di un vincolo di simmetria rispetto al piano x-z durante la *Phase I*, le variabili di design rimangono associate a un singolo punto, non garantendo quindi che i valori delle variabili di design associate a nodi simmetrici siano uguali. Inoltre, l'ottimizzazione dimensionale non consente di imporre vincoli di simmetria. Questo comporta che, durante la *Phase II*, in presenza di carichi o vincoli asimmetrici, la dimensione delle barre potrebbe variare in modo asimmetrico.

Risulta dunque indispensabile imporre manualmente una condizione che implichi uno sviluppo simmetrico durante l'ottimizzazione. Normalmente, questo problema può essere risolto facilmente tramite l'interfaccia grafica, eliminando metà delle barre del modello e successivamente specchiandole rispetto al piano di simmetria. In questo modo, gli ele-

menti specchiati presenterebbero le stesse proprietà della metà originaria del modello e sarebbero collegati alle stesse variabili di design, assicurando così che le due metà mantengano le stesse caratteristiche geometriche e meccaniche.

In aggiunta, questo processo permette di eliminare quasi la metà delle proprietà inutilizzate nel modello. Seguendo questa logica, anche le entità di collegamento tra le proprietà eliminate e le variabili di design ad esse associate diventano superflue, consentendo l'eliminazione di queste ultime e delle variabili di design della metà del modello rimossa. Questi passaggi, eseguiti tramite i comandi riportati in figura 4.9 "Unused" ed "Empty" nell'interfaccia grafica, semplificano e alleggeriscono il modello ottimizzato.

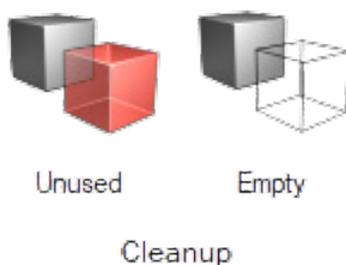


Figura 4.9: Strumenti per il cleanup degli elementi vuoti o inutilizzati del modello.

La problematica fondamentale legata al processo appena descritto è che lavorare tramite interfaccia grafica è fattibile fin quando il modello presenta un numero limitato di entità. Nel modello in esame, al contrario, si ottengono centinaia di migliaia di elementi beam, ai quali sono associate altrettante proprietà e DVPREL; ad appesantire ulteriormente il modello vi è la presenza di un numero di variabili di design particolarmente elevato. Queste caratteristiche si traducono praticamente in un significativo rallentamento di tutti i processi legati alla gestione del modello.

Ad esempio, la semplice cancellazione degli elementi beam appartenenti a una delle due metà del modello richiede un tempo quantificabile qualitativamente tra le due e le quattro ore. A ciò si aggiungono ulteriori operazioni particolarmente lente come la specchiatura della metà rimanente, l'eliminazione delle proprietà, delle DVPREL e delle DESVAR. Questi fattori rendono il processo computazionalmente impraticabile, evidenziando la necessità di una soluzione alternativa per affrontare il problema.

La soluzione a questo problema è stata individuata nello svolgimento delle operazioni di cancellazione e specchiatura attraverso il software di calcolo *Altair Compose*. Il processo attuato verrà descritto nel capitolo 5.

Alcune operazioni possono comunque essere eseguite tramite l'interfaccia grafica, poiché riguardano un numero limitato di entità e non comportano un elevato carico computa-

zionale. Tra queste rientra la ridefinizione della response associata all'obiettivo. Infatti, durante la *Phase I*, la response consiste nella massa di tutti gli elementi con proprietà PSOLID, escludendo quelli appartenenti al Non-Design Space. Tuttavia, per l'ottimizzazione dimensionale è necessario riferirsi esclusivamente alle beam.

Per soddisfare questa condizione, si procede in tre fasi eseguite tramite l'interfaccia grafica di *HyperMesh*:

- Si duplica la card del materiale MAT1 associato all'alluminio.
- Alle proprietà degli elementi solidi e degli elementi shell si riassegna il materiale appena creato.
- Nella response che controlla la massa si riassegna l'entità da controllare, cambiandola da *Property* \rightarrow *PSOLID* a *Materials* \rightarrow *MAT1_2*.

In questo modo si impone al solutore di controllare la massa relativa al materiale indicato, al quale sono assegnate tutte le barre presenti nel modello. Nelle figure 4.10 e 4.11 viene fornita una rappresentazione grafica che illustra la sequenza di operazioni e i loro effetti sul modello. In esse è riportata la rappresentazione grafica dei materiali: nell'immagine a sinistra si nota la presenza di un solo tipo di materiale, mentre in quella a sinistra ve ne sono due, uno per i solidi ed uno per le beam.

Name	ID		I.	D	Type	Card Image
MAT1_2	2		0		ISOTROPIC	MAT1
MAT8_1	1		0		ORTHOTROPIC	MAT8

Name	ID		I.	D	Type	Card Image
MAT1_2	2		0		ISOTROPIC	MAT1
MAT8_1	1		0		ORTHOTROPIC	MAT8
MAT1_21	3		0		ISOTROPIC	MAT1

Figura 4.10: Duplicazione del MAT1.

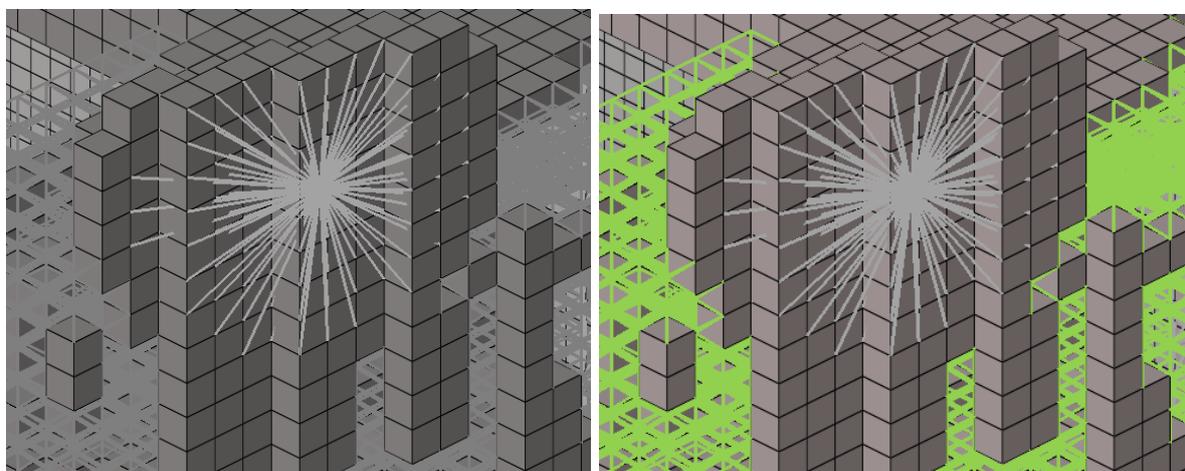


Figura 4.11: Riassegnazione del materiale agli elementi solidi.

Un ulteriore passaggio utile per semplificare il modello consiste nell'eliminare gli elementi RBE3 di collegamento tra la struttura in composito e quella in alluminio che risultano superflui dopo la *Phase I*. Vista l'eliminazione di un elevato numero di elementi di interfaccia con il *TUB*, infatti, i RBE3 che generavano il contatto rimangono collegati solo alla parte in materiale composito e, non trasmettendo alcuna forza, risultano inutili ai fini dell'analisi.

Si procede poi eliminando tutti gli elementi che, a seguito della conversione, non risultano più collegati alla struttura principale, poiché potrebbero introdurre labilità nel modello durante le fasi di analisi.

Infine, si esegue un controllo completo sul modello al fine di identificare eventuali errori che potrebbero presentarsi in fase di ottimizzazione. Il comando che permette di effettuare questa operazione è *Validate* \rightarrow *Check* \rightarrow *Model*.

4.1.3 Phase II

La struttura lattice è generata solo alla fine della *Phase I*, dunque non è possibile prevedere la risposta reale della struttura lattice perché in questa fase le reponse sono calcolate solo al termine delle iterazioni dell'ottimizzazione topologica, dunque tenendo conto delle densità degli elementi ma non della struttura convertita in lattice. In seguito alla conversione, infatti, si osservano due principali motivi di perdita di rigidità per la struttura:

- La prima è dovuta alla cancellazione degli elementi al di sotto del limite inferiore, che porta ad un aumento degli spostamenti e delle tensioni presenti nella struttura.
- La seconda è dovuta al fatto che, usando i valori di default del fattore di penalizzazione, la rigidità della struttura lattice è sovrastimata.

La seconda fase, dunque, consiste in una ottimizzazione dimensionale che ha come obiettivo il dimensionamento delle sezioni agli estremi degli elementi CBEAM. Durante questa fase si ha un controllo diretto sul modello ottimizzato poiché si sta analizzando effettivamente la struttura lattice. A differenza della struttura costituita da elementi solidi esaedrici, la lattice è in grado di gestire le anisotropie all'interno della sua struttura, rendendola teoricamente più efficiente.

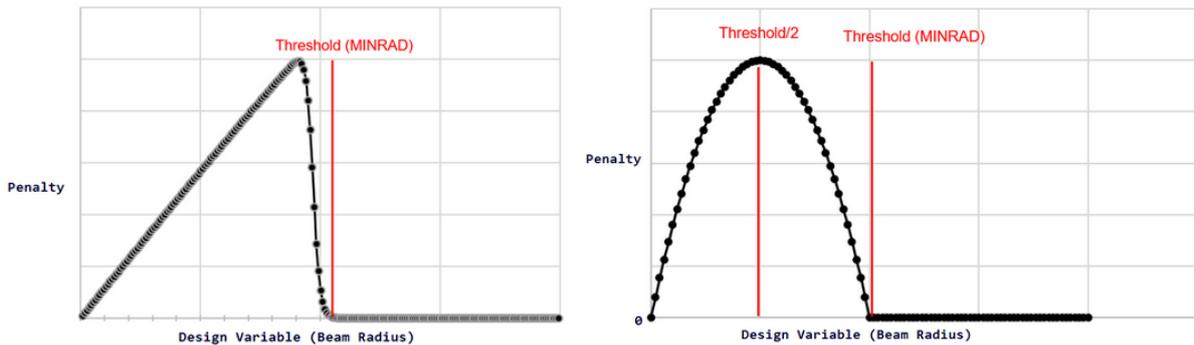
Lattice Sizing+ L'ottimizzazione *Lattice Sizing+* consiste in una fase aggiuntiva che viene eseguita al termine dell'ottimizzazione dimensionale delle beam, con l'obiettivo di eliminare quelle il cui raggio è inferiore al valore minimo definito dal parametro di controllo LATPRM,MINRAD.

Questa fase è costituita da due ulteriori step al termine dell'ottimizzazione lattice. Il primo prevede l'introduzione di un fattore di penalizzazione aggiuntivo, come riportato nell'equazione 4.3, che si somma a quello già definito in precedenza:

$$pen = \left(\sum_{i=1}^{n_{DV}} pen_i \right) \frac{obj_c}{pen_c} \quad (4.3)$$

Dove pen è il termine aggiuntivo del fattore di penalizzazione, pen_i rappresenta il fattore di penalizzazione associato a ciascuna variabile di design, obj_c è il valore dell'obiettivo a convergenza ottenuto nella precedente ottimizzazione di dimensionamento, e pen_c è il valore totale di penalizzazione alla fine di tale ottimizzazione.

Il fattore di penalizzazione associato a ciascuna variabile di design dipende dal metodo selezionato. Se si sceglie l'opzione LATPRM,CLEAN,YES, viene applicata la funzione di penalizzazione "Heaviside"; in alternativa, selezionando LATPRM,CLEAN,LESS, si utilizza una funzione di penalizzazione di tipo quadratico.



(a) Heaviside (LATPRM,CLEAN,YES).

(b) Quadratica (LATPRM,CLEAN,LESS).

Figura 4.12: Funzioni di penalizzazione implementate nella fase di Lattice Sizing+.

4.1.4 Gestione dei File

Per eseguire il modello creato in *HyperMesh* tramite il solutore *OptiStruct*, è necessario esportarlo nel formato *.fem*. Quest'ultimo consiste in un file di testo strutturato secondo specifiche regole di formattazione, suddiviso in tre sezioni principali illustrate a titolo di esempio in figura 4.13:

1. **Opzioni I/O:** la prima sezione definita nel file di input controlla la gestione generale dell'analisi o dell'ottimizzazione. Qui vengono specificati il tipo, il formato e la frequenza degli output, il tipo di simulazione da eseguire (analisi, controllo o ottimizzazione), oltre alla posizione e ai nomi dei file di input, output e temporanei.
2. **Definizione dei Subcase:** in essa si elencano i carichi e i vincoli associati ad ogni singolo subcase. Inoltre, in questa sezione è possibile controllare il tipo di output

e la frequenza e, nel caso si stia definendo un'ottimizzazione, si riportano anche le informazioni riguardanti gli obiettivi e i vincoli.

3. **Bulk Data:** questa sezione include tutti i dati relativi agli elementi finiti, come la posizione dei nodi, gli elementi, le proprietà, i materiali, i carichi, i vincoli, le condizioni al contorno e i sistemi di coordinate. Nel caso di ottimizzazioni, contiene anche la definizione delle variabili di design, delle response e dei vincoli di ottimizzazione. La sezione è sempre introdotta dal comando "BEGIN BULK".

Nel file di input, le entità sono identificate tramite un valore intero specificato nel campo di definizione dell'ID di ciascuna entità. Questo ID può essere richiamato all'interno di altre entità per stabilire collegamenti tra di esse. Ad esempio, se si definisce un materiale con un determinato ID e successivamente si richiama tale ID nel campo MID di una proprietà, quella proprietà sarà associata al materiale indicato. Allo stesso modo, se un nodo del modello viene assegnato a un ID specifico, è possibile richiamarlo nel campo GID_i di un elemento per indicare che quel nodo fa parte dell'elemento appena definito.

```

$ OPTISTRUCT 5.0 Input Deck Generated by HyperMesh Version5.1
$
$   Template:  optistruct
$
$
$
TITLE = Plate Model
ANALYSIS
FORMAT H3D
FORMAT HM
TMPDIR = /myhome/scratch
SCREEN NONE
-----$
$                               Case Control Cards   $
$ -----$
$ HMNAME LOADSTEPS           1Load
$
SUBCASE   1
  SPC =   1
  LOAD =   1
$
$ BEGIN BULK
$
$   GRID Data
$
GRID      11      33.3333330      0.0
GRID      12      31.3296290      0.0
GRID      13      29.3259250      0.0
...
...
$
$   CQUAD4 Elements
$
CQUAD4      1      2      74      11      12      75
CQUAD4      2      2      40      41      42      76
CQUAD4      3      2      76      42      43      77

```

Figura 4.13: Esempio di file di input per OptiStruct [11].

5. Gestione del Modello in Altair Compose

5.1 Basi per la Creazione dello Script

5.1.1 Introduzione al Software

All'interno della suite software fornita da Altair è disponibile la piattaforma *Compose*, un software di programmazione e calcolo numerico utilizzato per l'analisi di dati, lo sviluppo di algoritmi e la creazione di modelli. Inoltre, integra funzioni specifiche per la visualizzazione e l'analisi di varie tipologie di file, tra cui i file CAE ricavati dagli altri software presenti nella suite.

Tra i diversi linguaggi disponibili, Compose supporta il linguaggio OML, un linguaggio ad alto livello (molto astratto rispetto al linguaggio di base del computer), basato su computazioni numeriche matriciali, simile a quello utilizzato in Matlab. Il software presenta un ambiente di programmazione unificato e interattivo, dedicato al supporto alla programmazione in linguaggio OML [12].

5.1.2 Regole di Formattazione del Testo

Nella sezione 4.1.4 si è constatato che il file `.fem` contiene tutte le informazioni utili alla definizione del modello e alla determinazione dei parametri essenziali al corretto svolgimento dell'analisi o dell'ottimizzazione. Per poter manipolare questo file tramite script, è quindi fondamentale comprendere in dettaglio il suo formato e la struttura delle informazioni in esso contenute.

Di seguito vengono riportate le regole fondamentali per la corretta formattazione del testo all'interno del file, sulla base delle informazioni ricavate dalle guide ufficiali del software:

- Il file è costituito da una sequenza di righe, ognuna delle quali può contenere fino ad un massimo di 80 caratteri. Ognuna di queste linee contiene fino a nove campi a formato fisso, ognuno costituito da otto caratteri. L'uso del formato fisso limita entro un determinato range l'uso di numeri interi, poiché non è possibile definire numeri che occupino più di 8 caratteri. Ciò può generare errori sull'accuratezza dei numeri a virgola mobile, tuttavia, questo non influisce sulla quantità di memoria interna occupata dal file.
- Nonostante ogni riga possa contenere un massimo di 80 caratteri ed ogni campo ne possa contenere otto, il numero massimo di campi in una riga è limitato a 9. Infatti, sia il contenuto dell'ultimo campo della prima riga, sia quello del primo campo di eventuali righe in continuità con essa, sono ignorate dal software quando è in uso il formato fisso.



Figura 5.1: Esempio di riga a formato fisso.

- La riga può iniziare con dei simboli. Quando questi simboli sono il dollaro "\$", il doppio slash "//" o il simbolo "#", la riga viene letta come un commento; al contrario, quando questa comincia con un campo vuoto, con un "+" o con un "*", la riga viene trattata come una continuazione di quella precedente. Quando, invece, la riga è dedicata alla definizione di un'entità, all'interno del primo campo viene indicato il nome dell'entità stessa. Ad esempio, per la definizione di un nodo, nel primo campo è riportata la stringa "GRID".
- All'interno dei campi destinati ad un valore intero non sono ammessi simboli. Al contrario, nei campi in cui deve essere indicato un valore di tipo float, nel caso in cui venga indicato un intero, questo viene convertito automaticamente in un numero a virgola mobile. I numeri possono essere riportati in notazione esponenziale, anche se l'uso della lettera "E" è facoltativo. Nei file esportati da HyperMesh, ad esempio, sono riportati senza l'uso della lettera "E", infatti il numero $9 \cdot 10^6$ è indicato come $9 + 06$.

In generale, le regole appena definite valgono per l'intera sezione del file dedicata alla definizione dei dati, ovvero la sezione *Bulk Data*, riportata in figura 4.13.

5.2 Processo di Modifica della Struttura

5.2.1 Fasi del Processo

L'uso di Altair Compose per la fase di modifica della struttura, intermedia alle due fasi dell'ottimizzazione lattice, è giustificata dall'elevato carico computazionale dei processi di cancellazione delle beams e delle entità a queste legate. Questo processo richiede però la generazione di un algoritmo che effettui tali modifiche, poiché non esistono funzioni dedicate integrate nel software.

La cancellazione tramite script avviene attraverso l'uso di vettori contenenti tutte le informazioni necessarie per la manipolazione e riscrittura delle entità da modificare. Questa soluzione offre significativi vantaggi in termini di riduzione dei tempi di elaborazione, riducendo il carico sulla macchina e minimizzando il rischio di perdita dei progressi a causa di crash del software, un problema riscontrato frequentemente durante la gestione del modello tramite interfaccia grafica.

Il processo di modifica avviene in tre fasi diverse, che integrano l'uso di script e la manipolazione della struttura tramite interfaccia grafica:

1. **Specchiatura degli elementi beam:** in questa fase si procede alla specchiatura degli elementi beam controllando l'assegnazione delle proprietà. Per garantire un corretto collegamento delle barre è necessario generare un nuovo elemento GRID per ogni nodo associato alle barre stesse. Nei punti di giunzione delle travi, quindi, si vanno a generare dei cluster di nodi, uno per ogni beam che converge in un determinata coordinata.
2. **Equivalenza dei nodi:** tramite l'interfaccia grafica si effettua l'equivalenza dei cluster di nodi creati nella fase precedente. Questa operazione, pur se effettuata in *HyperMesh*, richiede un tempo relativamente breve rispetto alla manipolazione di altre entità, riducendo ulteriormente i tempi di elaborazione.
3. **Cancellazione delle entità inutilizzate:** l'ultima fase consiste nell'eliminazione di tutte quelle entità che risultano inutilizzate dal modello, ancora una volta attraverso l'uso di un secondo script.

5.2.2 Creazione degli Algoritmi

Facendo riferimento a quelle definite nel paragrafo precedente, si procede a illustrare nel dettaglio le caratteristiche di ciascuna fase.

In generale, la struttura dell'algoritmo si divide in tre passaggi fondamentali: lettura file

FEM e memorizzazione dei dati relativi alle entità di interesse, manipolazione dei dati memorizzati e scrittura in un file secondario che contenga tutte le entità modificate da integrare nei file *fem* completi. Di seguito si analizzano in dettaglio i passaggi appena definiti per quanto riguarda le fasi di specchiatura delle *beam* e cancellazione delle entità inutilizzate.

Specchiatura degli elementi BEAM Una volta aperto il file FEM in modalità lettura, la prima operazione consiste nell’ottenere il numero di elementi e il numero di nodi definiti nel modello, al fine di creare dei vettori le cui dimensioni permettano di memorizzare tutti i dati di interesse. Questo passaggio viene eseguito utilizzando delle funzioni integrate nel software come $elemnum = getnumelems3d(file)$ e $nodenum = getnumnodes3d(file)$, dove "file" rappresenta il percorso del file da analizzare all’interno della macchina. Tali comandi consentono di estrarre informazioni dai file CAE ottenuti al termine della *Phase I*. Successivamente, viene definita la variabile *elemstartid*, utilizzata per gestire il fatto che il numero totale di elementi restituito dalla funzione include tutti gli elementi del modello, non solo quelli di tipo *beam*, di interesse in questa fase. Poiché non è necessario considerare l’intera lunghezza del vettore degli elementi, *elemstartid* permette di avviare i controlli a partire dal primo elemento che corrisponde a una trave.

```

13 % Apro il file
14 file = 'C:\Users\gallo\Desktop\Tesi Magistrale\File tesi\RF_ONLY\lattice\rf_lat_rst028_lattice.fem';
15 fid = fopen(file, 'rt');
16
17 % Contatori e pre-allocazioni
18 elemnum = getnumelems3d(file)*1;
19 nodenum = getnumnodes3d(file)*1;
20 elemstartid = 0;
21
22 % Pre-allocazione dei vettori
23 cbeam = struct('eid', zeros(elemnum, 1), 'pid', zeros(elemnum, 1),...
24 'ga', zeros(elemnum, 1), 'gb', zeros(elemnum, 1), 'x', zeros(elemnum, 1),...
25 'y', zeros(elemnum, 1), 'z', zeros(elemnum, 1));
26 grid = struct('id', zeros(nodenum, 1), 'x', zeros(nodenum, 1),...
27 'y', zeros(nodenum, 1), 'z', zeros(nodenum, 1), 'cd', zeros(nodenum, 1));

```

Figura 5.2: Definizione delle dimensioni dei vettori e creazione delle struct.

Questi valori sono poi utilizzati per determinare la lunghezza dei vettori da definire. Infatti, i vettori associati agli elementi avranno lunghezza "elemnum", mentre i vettori associati ai nodi avranno lunghezza "nodenum". Poiché in questa fase si effettua la specchiatura dei soli elementi geometrici, le informazioni essenziali sono contenute nelle righe associate agli elementi e ai nodi. In particolare, si definiscono delle strutture (figura 5.2), il cui contenuto corrisponde ai campi utilizzati nella definizione dei Bulk Data di una determinata entità.

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
CBEAM	EID	PID	GA	GB	X1/G0	X2	X3	OFFT	
CBEAM	589967	314851	216057	216067	2.0	3.0	-9	BLANK	

Tabella 5.1: Riga per la definizione dei dati relativi ad un elemento CBEAM (ID=589967).

In base a quanto si osserva in tabella 5.1, le informazioni necessarie alla definizione di un elemento CBEAM sono: l'ID dell'elemento (EID), l'ID della proprietà associata a questo elemento (PID), l'ID dei nodi che costituiscono gli estremi dell'elemento (GA e GB) e le componenti del vettore v (X1, X2 e X3), parallelo allo spostamento del nodo A nel sistema di coordinate assoluto. Nel caso in esame, l'elemento considerato è il primo tra quelli definiti come CBEAM nel modello, pertanto, durante la lettura dei dati, la variabile "elemstartid" viene inizializzata a $elemstartid = 589967$. Questo consente di ignorare i primi 589966 elementi del vettore durante i controlli, velocizzando considerevolmente l'esecuzione dello script.

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
GRID	ID	CP	X1	X2	X3	CD	PS		
GRID	103302	BLANK	-270.502	-392.807	101.356	BLANK	BLANK		

Tabella 5.2: Riga per la definizione dei dati relativi ad un elemento GRID (ID=103302).

I nodi, come riportato in tabella 5.2, vengono definiti in modo analogo. Per le entità GRID, infatti, si memorizzano le variabili relative all'ID del nodo e alla posizione del punto nel sistema di coordinate globale (X1, X2 e X3); tutti gli altri campi vengono ignorati. Le strutture generate saranno dunque costituite da un numero di vettori, di lunghezza analoga, pari al numero di variabili da memorizzare per ogni entità definita.

```

34  % Funzioni helper
35  function out = process_coordinates(coord_string)
36  if contains(coord_string(5:8), '-')
37      R = sscanf(coord_string, '%f-%d');
38      out = R(1)^(-R(2));
39  elseif contains(coord_string(5:8), '+')
40      R = sscanf(coord_string, '%f+%d');
41      out = R(1)^R(2);
42  else
43      out = str2double(coord_string);
44  end
45  end

```

Figura 5.3: Funzione helper per la lettura delle coordinate dei nodi.

La procedura per la memorizzazione delle informazioni contenute all'interno dei file è funzione della formattazione delle righe descritta in precedenza. La lettura del file, infatti,

avviene riga per riga, tramite la funzione "fgetl". Questa memorizza l'intera stringa nella variabile "riga", conseguentemente interpretata in base alla sua formattazione interna.

```

47 % Ciclo per la lettura del file linea per linea
48 while ~feof(fid)
49     counter = counter + 1;
50     riga = fgetl(fid);
51     if length(riga) > 8
52         name = riga(1:8);
53
54         if strcmp(name, 'CBEAM ')
55             count = count + 1;
56             b = str2double(riga(9:16));
57             cbeam.eid(b) = round(b);
58             if cbeam.eid(b) > 0 && elemstartid == 0
59                 elemstartid = cbeam.eid(b);
60             end
61             cbeam.pid(b) = str2double(riga(17:24));
62             cbeam.ga(b) = str2double(riga(25:32));
63             cbeam.gb(b) = str2double(riga(33:40));
64             cbeam.x(b) = str2double(riga(41:48));
65             cbeam.y(b) = str2double(riga(49:56));
66             cbeam.z(b) = str2double(riga(57:64));
67
68             elseif strcmp(name, 'GRID ')
69
70                 b = str2double(riga(9:16));
71                 grid.id(b) = b;
72                 grid.x(b) = process_coordinates(riga(25:32));
73                 grid.y(b) = process_coordinates(riga(33:40));
74                 grid.z(b) = process_coordinates(riga(41:48));
75
76             end
77         end
78     end
79     fclose(fid);

```

Figura 5.4: Ciclo di lettura e memorizzazione delle variabili di interesse.

Se il primo campo, ovvero i primi otto caratteri della stringa, corrisponde alla sequenza di caratteri "CBEAM", si memorizzano le variabili relative alle travi; se, invece, contiene la sequenza di caratteri "GRID", si memorizzano quelle relative ai nodi con le modalità riportate in figura 5.4. In fase di lettura delle entità di tipo GRID è necessario introdurre una funzione ausiliare in conseguenza al fatto che è possibile, per le coordinate dei nodi, che queste siano espresse in notazione scientifica. Per questo motivo è necessario individuare se in corrispondenza degli ultimi tre caratteri del campo in esame sono presenti eventuali segni positivi o negativi che precedono il valore numerico dell'esponente della notazione scientifica. A tal fine, si definisce una funzione (riportata in figura 5.3) in grado di svolgere tale compito.

Successivamente, a valle dell'elaborazione dei dati si procede alla riscrittura delle righe

utili alla ridefinizione degli elementi specchiati. In particolare, nelle prime righe di codice in figura 5.5 è riportata la funzione dedicata alla scrittura delle righe di definizione degli elementi CBEAM, sviluppata per semplificare lo script e velocizzare le operazioni di scrittura su file. Si osserva come, per la definizione di ogni variabile, si riserva una sequenza di otto caratteri, coerentemente con quanto osservato in fase di lettura.

```

96 % Funzione helper per scrivere gli elementi CBEAM
97 function write_cbeam(fid, eid, pid, ga, gb, x, y, z)
98     fprintf(fid, 'CBEAM %8d%8d%8d%8d%-8.1f%-8.1f%-8.1f\n', eid, pid, ga, gb, x, y, z);
99 end
100
101 % Ciclo sugli elementi specchiati
102 mirror = zeros(elemnum / 2, 1);
103 neutral = zeros(elemnum / 2, 1);
104 mc = 1;
105 nc = 1;
106
107 for i = elemstartid:elemnum
108
109     if ~(cbeam.ga(i) == 0 || cbeam.gb(i) == 0)
110         ga_y = grid.y(cbeam.ga(i));
111         gb_y = grid.y(cbeam.gb(i));
112
113         if ga_y >= 0 && gb_y >= 0 && ~(ga_y == 0 && gb_y == 0)
114             mirror(mc) = cbeam.eid(i);
115             mc = mc + 1;
116         elseif (ga_y == 0 && gb_y == 0) || (ga_y + gb_y == 0)
117             neutral(nc) = cbeam.eid(i);
118             write_cbeam(fid, neutral(nc), cbeam.pid(neutral(nc)), cbeam.ga(neutral(nc)), cbeam.gb(neutral(nc)), ...
119                 cbeam.x(neutral(nc)), cbeam.y(i), cbeam.z(i));
120             nc = nc + 1;
121         end
122     end
123 end

```

Figura 5.5: Funzione helper per la scrittura delle righe relative a CBEAM, seguita dal ciclo per l'individuazione delle beams da specchiare.

Nella stessa immagine è mostrato il ciclo che identifica le travi da specchiare. La logica seguita dal ciclo è la seguente: volendo specchiare tutti gli elementi i cui nodi hanno coordinata y positiva, eliminando i loro complementari, se uno dei due nodi dell'elemento ha coordinata y positiva e l'altro nodo ha coordinata y maggiore o uguale a 0, l'ID dell'elemento viene memorizzato nel vettore "mirror"; quest'ultimo, infatti, è un vettore creato appositamente al fine di memorizzare gli indici degli elementi da mantenere (facenti parte della metà originale del modello) e specchiare (nuovi elementi generati a partire dalla metà originale, aventi le stesse proprietà degli elementi ad essi simmetrici). Se, invece, entrambi i nodi hanno coordinata y pari a 0, gli ID vengono memorizzati nel vettore "neutral", ovvero un vettore utilizzato per la memorizzazione degli indici che individuano quegli elementi appartenenti al piano di simmetria e che non necessitano dunque di specchiatura. Questi ultimi vengono scritti direttamente nel file di output senza ulteriori modifiche. Gli elementi con nodi aventi coordinata y negativa vengono ignorati in questa fase.

A questo punto è possibile aprire il file di scrittura dei risultati, poiché gli step di elabora-

zione e scrittura dei dati avvengono, in alcuni casi, nel medesimo ciclo. Definite le entità da mantenere, si procede alla specchiatura degli elementi e dei nodi. La logica seguita dall'algoritmo di specchiatura prevede un ciclo sul vettore "mirror". Se l'elemento i -esimo del vettore è diverso da 0, allora esso contiene l'ID della barra da specchiare. Come primo passo, si scrive l'elemento beam originale, quindi si verifica la coordinata y dei nodi dell'elemento. Se la coordinata y del nodo A è pari a 0, si specchia il nodo B creando una entità GRID con coordinata y opposta rispetto a quella del nodo B originale. Avendo memorizzato la coordinata y all'interno del vettore $grid.y$ e avendo definito il vettore delle coordinate specchiate come $m_grid.y$, la specchiatura del nodo consiste semplicemente nel porre $m_grid.y(B) = -grid.y(B)$, mantenendo le coordinate x e z uguali a quelle del nodo originale. Una volta creato il nodo specchiato, si definisce l'elemento CBEAM specchiato utilizzando come nodo A quello originale, come nodo B quello specchiato, e mantenendo la stessa proprietà dell'elemento originale. Quest'ultima condizione è quella che garantisce l'imposizione della simmetria nel modello. Per completare la specchiatura occorre aggiornare le coordinate del vettore v associato all'elemento, considerando la nuova inclinazione dell'elemento specchiato rispetto al sistema di riferimento.

Se, al contrario, è il nodo B ad avere coordinata y nulla e il nodo A ha coordinata y maggiore di 0, allora si effettua la specchiatura del nodo A e il nodo B rimane in comune tra le due beams.

Ancora, se sia il nodo A che il nodo B hanno coordinata y maggiore di 0, allora si specchiano entrambi, secondo la procedura appena descritta, applicata sia al nodo A che al nodo B. In questa fase è di fondamentale importanza verificare la corretta assegnazione dei nodi all'elemento, poiché la proprietà tiene conto di tale ordine. Se quest'ordine dovesse essere invertito, infatti, le dimensioni dei due estremi della trave non rispetterebbero il vincolo di simmetria se confrontate con quella originale.

L'ultima operazione eseguita dall'algoritmo di riscrittura degli elementi beam consiste nell'inserimento di una serie di commenti nel file di testo generato, il cui scopo è assegnare correttamente gli elementi al componente di riferimento, nel caso in cui questi non fossero posizionati correttamente nello script. A tal fine, il codice inizia scrivendo una riga standard con il commento "\$HMMOVE #", dove il simbolo "#" rappresenta l'ID del componente (tale ID è facilmente individuabile all'interno del file *fem* in corrispondenza delle righe di definizione degli elementi *beam*) al quale si vogliono assegnare gli elementi i cui ID sono riportati in seguito alla riga di commento. Poiché gli elementi da assegnare al componente corretto sono in numero particolarmente elevato, nelle righe successive alla prima, sempre precedute dal simbolo "\$", vengono elencati gli ID degli elementi da assegnare al componente, rispettando la convenzione secondo cui il primo campo è riservato

al simbolo "\$" che indica un commento, e l'ultimo campo viene omissso.

Equivalenza dei Cluster di Nodi La fase successiva consiste nell'equivalenza dei cluster di nodi generati nella fase precedente. Infatti, avendo generato due nodi per ognuna delle barre presenti nella metà specchiata, si ottiene che per ogni punto di giunzione in questa parte di modello si avrà un nodo per ognuna delle beam che si collegano a questa giunzione. Ciò vuol dire che le beam risulteranno tra loro scollegate. L'equivalenza, dunque, unisce tutti i nodi convergenti in un punto che rientrano entro una certa tolleranza definita dall'utente. Questa operazione si effettua tramite il comando *Validate* → *FixElements* → *Equivalence* e, fissato il valore di tolleranza (è sufficiente il valore di 1mm), i nodi vengono uniti e, di conseguenza, le travi risultano collegate. I cluster sono presenti solo nella metà specchiata del modello, poiché la metà originale viene ricostruita allo stesso modo rispetto alla struttura iniziale.

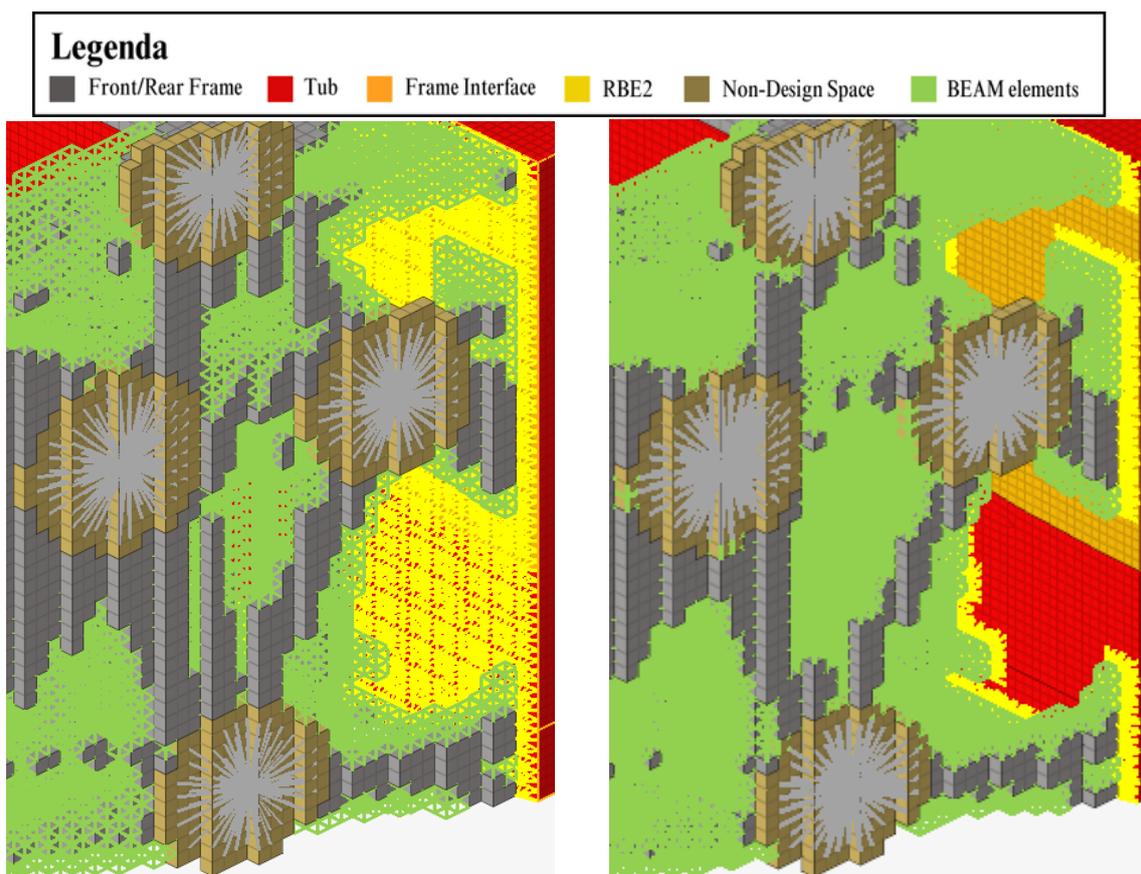


Figura 5.6: Particolare degli RBE3 del lato sinistro anteriore, prima e dopo la cancellazione degli RBE3 superflui.

In questa fase si procede anche all'eliminazione degli RBE3 inutilizzati per via della cancellazione di determinati elementi. Questa operazione snellisce ulteriormente il modello

riducendo il numero di elementi presenti al suo interno. L'effetto di quest'ultimo passaggio è visibile nelle figure 5.6.

Cancellazione delle Entità Inutilizzate Gli step che costituiscono l'algoritmo, anche in questo caso, consistono in: lettura, manipolazione e scrittura dei risultati. Tuttavia, i dati necessari allo svolgimento di questa fase sono più numerosi. In particolare, per la cancellazione delle proprietà, delle DVPREL e delle variabili di design, associate alle barre eliminate nella fase di specchiatura, bisogna memorizzare le variabili indicate di seguito:

- **DESVAR:** per le variabili di design si creano quattro vettori, in cui si memorizzano rispettivamente l'ID, il valore iniziale della variabile (XINIT) e il massimo (XUB) e il minimo (XLB) valore che può essere assegnato alla variabile.
- **DVPREL:** la definizione delle relazioni tra variabili di design e proprietà passa attraverso la memorizzazione dell'ID dell'entità, della stringa che determina se la variabile di design fa riferimento al nodo A oppure al nodo B (PNAME), dell'ID della proprietà (PID) e dell'ID della variabile di design da collegare (DVID1).
- **CBEAM:** come nel caso precedente si memorizzano ID, PID, GA e GB.
- **GRID:** ancora, come nel caso precedente, si memorizzano l'ID e le coordinate del nodo (X1, X2, X3).
- **PBEAML:** per quanto riguarda le proprietà, le variabili da salvare in memoria sono l'ID, il raggio del cono nel nodo A (DIM1(A)) ed il raggio del cono nel nodo B (DIM1(B)).

Analogamente all'algoritmo definito prima, si procede alla lettura e memorizzazione dei dati appena enunciati.

Dopodiché si procede alla manipolazione dei dati, la cui logica consiste in tre cicli effettuati in sequenza: il primo sul vettore $p.id$, poi sul vettore $dvprel.id$ e il terzo sul vettore $desvar.id$.

Nel primo ciclo si verifica se l'elemento i -esimo del vettore $p.id$ è presente nel vettore $cbeam.pid$, ovvero se la proprietà in esame viene effettivamente richiamata da almeno un elemento. In caso di esito negativo, la variabile $p.id(i)$ viene annullata, passando così all'iterazione successiva.

Nel secondo ciclo viene applicata una logica simile: si controlla se l'elemento i -esimo del vettore $dvprel.pid$ è contenuto nel vettore $p.id$, che è stato aggiornato nel ciclo precedente. Se non vi è corrispondenza, sia $dvprel.pid(i)$ che $dvprel.dvid1(i)$ vengono annullati. L'annullamento di $dvprel.dvid1(i)$ consente, al termine del ciclo, di ottenere un vettore

dvprel.dvid1 contenente esclusivamente gli ID delle variabili di design da mantenere. Infine, il terzo ciclo verifica se l'elemento *desvar.id(i)* è presente nel vettore *dvprel.dvid1*. In caso di riscontro negativo, la variabile *desvar.id(i)* viene annullata.

```

109 □ for i = 1:length(p.id)
110 □     if not (ismember(p.id(i),cbeam.pid))
111     p.id(i) = 0;
112     end
113 end
114
115 □ for i = 1:length(dvprel.id)
116 □     if not (ismember(dvprel.pid(i),p.id))
117     dvprel.id(i) = 0;
118     dvprel.dvid1(i) = 0;
119     end
120 end
121
122 □ for i = 1:length(desvar.id)
123 □     if not (ismember(desvar.id(i),dvprel.dvid1))
124     desvar.id(i) = 0;
125     end
126 end

```

Figura 5.7: Processo di selezione delle entità da mantenere nel modello.

A questo punto, è possibile scrivere su file di output tutti gli elementi i cui ID sono diversi da 0, poiché tutti gli ID di quegli elementi che non avevano un riscontro nel modello reale, sono stati impostati pari a 0. Il codice impiegato per effettuare il processo di eliminazione dei dati è riportato in figura 5.7. In essa si è riportato il codice relativo ai tre cicli enunciati sopra.

5.2.3 Effetto dei Codici sulla Struttura

I codici descritti nel paragrafo precedente generano i loro risultati in due file di testo distinti, i cui effetti sul modello vengono riportati in questo paragrafo.

L'intero processo di specchiatura degli elementi beam è schematizzato in figura 5.8. La figura illustra le tre fasi principali della procedura di modifica, coerentemente con quanto descritto nel paragrafo 5.2.1, specificando il software utilizzato in ciascuna di esse. Inoltre, sono rappresentati i file di input e output relativi a ogni fase: i file contrassegnati in verde corrispondono ai file *fem* completi, che contengono tutte le informazioni sull'intero modello, mentre quelli evidenziati in blu sono i file di testo generati dai due algoritmi, contenenti esclusivamente le entità modificate durante il processo. Infine, le linee tratteggiate indicano la necessità di integrare manualmente le entità riscritte nei file di testo all'interno dei file *fem*. Questa operazione viene eseguita manualmente utilizzando un

editor di file sorgente, nello specifico *Notepad++*.

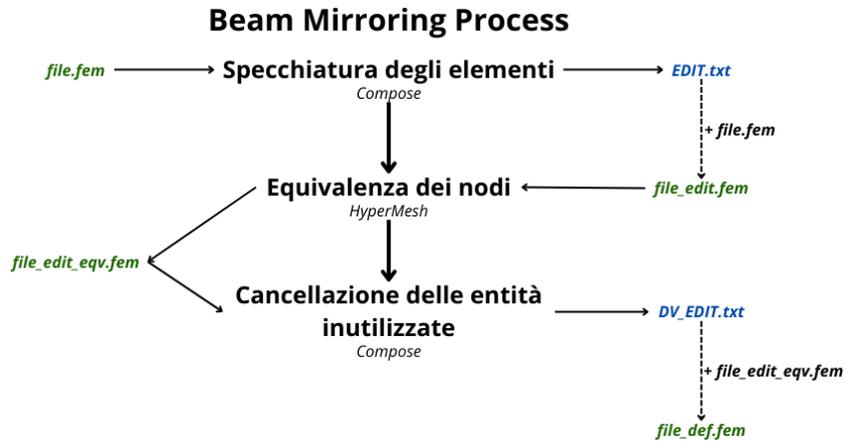
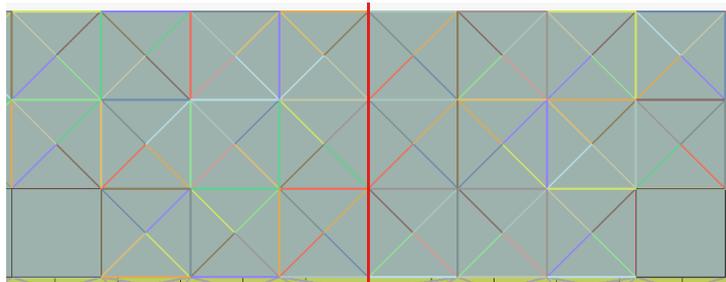


Figura 5.8: Processo di specchiatura degli elementi.

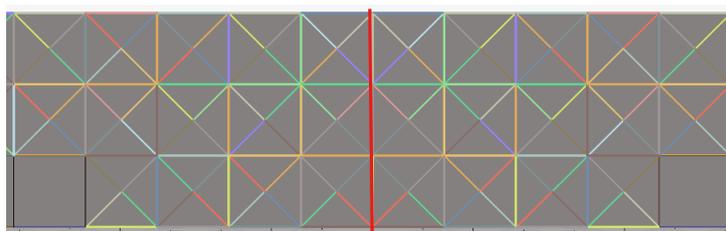
La prima fase richiede in input il file risultante dalla *PhaseI* dell’ottimizzazione lattice. Conclusa la fase di elaborazione, il software produce un file chiamato ”*EDIT.txt*” nel quale vengono elencati, in ordine, gli elementi, i nodi e i commenti necessari per assegnare gli elementi a un determinato componente. Come accennato in precedenza, i dati contenuti in questo file devono essere integrati nel file FEM al fine di implementare correttamente le modifiche nel modello. In particolare, per quanto riguarda le righe relative agli elementi, si effettua una sostituzione totale di quelli preesistenti nel file *fem*. Al loro posto si inseriscono le righe presenti nel file di testo poiché sia gli elementi originali da mantenere, sia quelli specchiati sono stati completamente riscritti. Al contrario, le informazioni relative ai nodi (GRID) e ai commenti devono essere aggiunte a quelle già presenti nel file *fem*, poiché quelle riportate nel file di testo non rappresentano tutte le entità effettivamente esistenti nel modello.

L’effetto della specchiatura sulla struttura lattice è illustrato nelle figure 5.9, all’interno delle quali il colore di ogni segmento rappresenta univocamente una proprietà. Nello specifico, la figura 5.9a mostra la condizione iniziale del modello, in cui a ciascuna *beam* è associata una proprietà distinta, senza che si manifesti alcuna simmetria rispetto al piano indicato dalla linea rossa posta in mezzeria. In questa configurazione, l’assenza di simmetria è evidente osservando la distribuzione dei colori delle proprietà. Diversamente, nella figura 5.9b, è possibile osservare la disposizione simmetrica delle proprietà delle travi rispetto al piano centrale, a dimostrazione dell’applicazione dell’algoritmo di specchiatura. Infine, il codice implementato nella terza fase richiede in input il file risultante dalla seconda. Genera un file denominato ”*DV_EDIT.txt*” che contiene, in ordine, le variabili di design, i DVPREL e le proprietà. Come nel caso precedente, per rendere attive le

modifiche nel modello, è necessario inserire le righe del file di testo all'interno del file FEM. Tuttavia, poiché per tutte e tre le entità coinvolte da questo algoritmo sono state eseguite operazioni di cancellazione e riscrittura completa, le righe del file di testo devono sostituire integralmente quelle già presenti nel modello.



(a) Proprietà delle travi asimmetriche, rispetto ai due lati del piano di simmetria indicato con la linea rossa.



(b) Proprietà delle travi simmetriche, rispetto ai due lati del piano di simmetria indicato con la linea rossa.

Figura 5.9: Effetto dell'algoritmo di specchiatura sulle proprietà delle beams.

6. Risultati dell'Ottimizzazione Lattice

6.1 Considerazioni sui Parametri di Controllo

In una prima serie di simulazioni è stato analizzato il comportamento della massa obiettivo risultante al termine dell'ottimizzazione topologica in funzione della variazione del parametro LATPRM, POROSITY. Tale massa, come già osservato nel processo descritto nel capitolo relativo all'ottimizzazione topologica, non rappresenta ancora un valore finale, in quanto necessita prima di essere convertita in una struttura lattice e successivamente ottimizzata dimensionalmente per le beams.

Tabella 6.1: Effetto del parametro POROSITY sulla massa ottenuta dalla topologica.

POROSITY	Massa Topologica (Phase I)	Massima % di Violazione dei Vincoli (Iteration 0-Phase II)
LOW	117	73%
MEDIUM	99	315%
HIGH	84	534%

Queste simulazioni hanno quindi uno scopo principalmente dimostrativo, poiché, come accennato in precedenza, l'errore nella stima della rigidità degli elementi beam può influire significativamente sulla rigidità complessiva della struttura. Questo effetto è particolarmente evidente nella tabella 6.1, dove, a parità di altri parametri, viene mostrato l'andamento di alcune variabili in funzione della variazione del parametro che regola la porosità della struttura. In particolare, nella colonna relativa alla massima percentuale di violazione dei vincoli, si può osservare di quanto si supera il valore massimo di spostamento imposto dal vincolo. L'aumento di tale violazione è causato dalla sovrastima della

rigidezza degli elementi beam, un errore che si amplifica a causa dell'elevato numero di beam presenti nella struttura. All'aumentare della porosità, ovvero della percentuale di elementi con densità intermedia, questo effetto diventa più accentuato. Nei casi di carico più critici, infatti, si registra una violazione dei vincoli superiore di oltre cinque volte rispetto al valore nominale per la configurazione con porosità HIGH e di oltre tre volte per quella con porosità MEDIUM.

Nonostante ciò, da queste ottimizzazioni è possibile osservare un trend decrescente della massa in funzione dell'aumento della porosità. Questo fenomeno conferma quanto già affermato: una distribuzione del materiale che privilegia le densità intermedie risulta più efficiente rispetto a una distribuzione discreta del materiale.

Al fine di ridurre al minimo l'errore associato alla stima delle rigidezze degli elementi, per le successive iterazioni si prende in considerazione il solo parametro di controllo LOW.

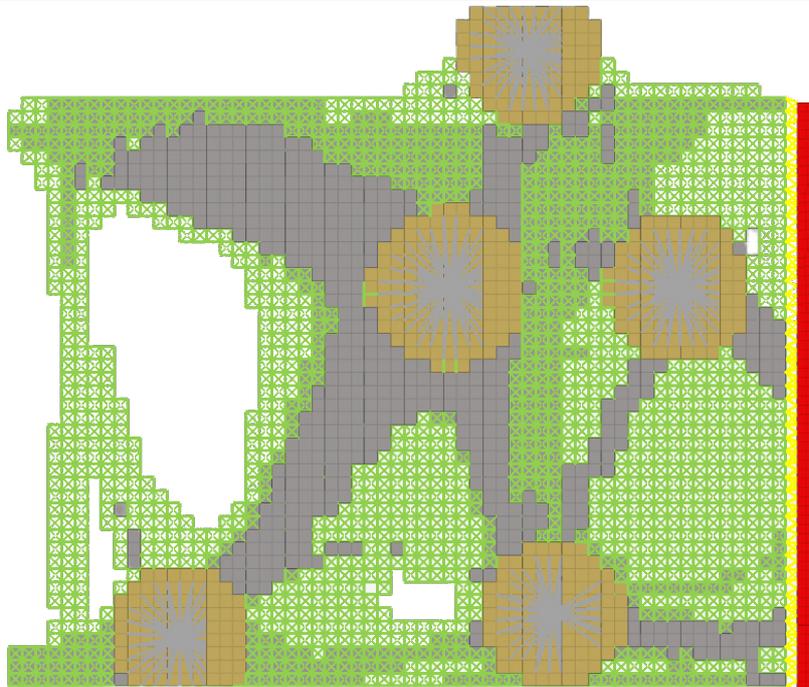
6.2 Risultati Phase I

Al termine della prima fase di ottimizzazione si ottiene una struttura già convertita in funzione dei valori di LB e UB definiti e dei risultati dell'ultima iterazione dell'ottimizzazione topologica.

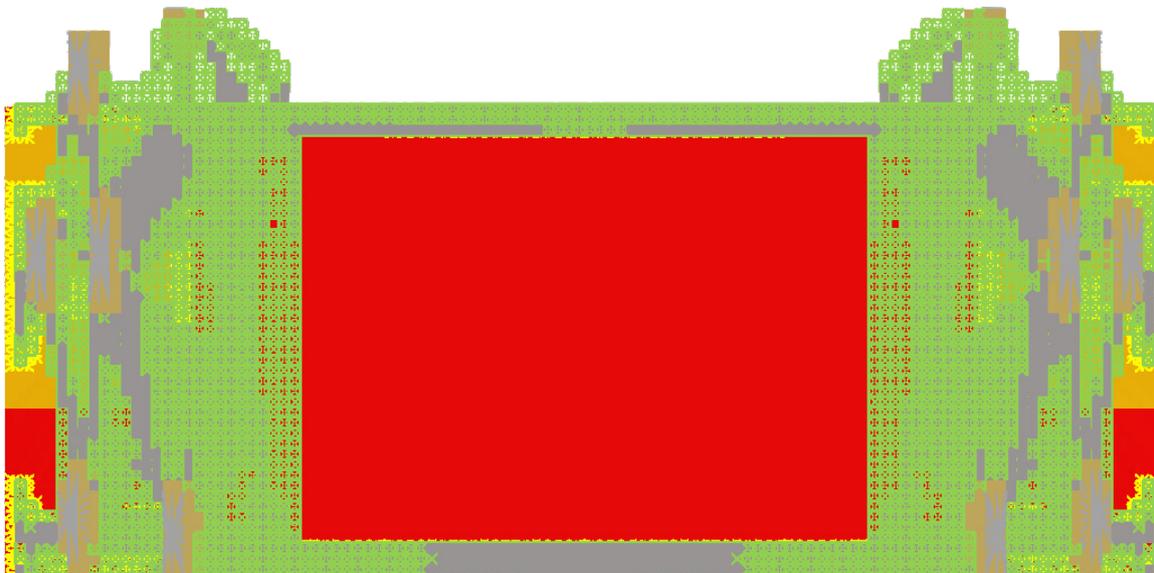
In particolare, si riportano nelle figure 6.1 e 6.2 le strutture convertite sulla base dei parametri dell'ottimizzazione definitiva: $UB = 0,6$ e $LB = 0,05$. In queste immagini si può osservare che la maggior parte degli elementi appartenenti alla fascia intermedia sono stati convertiti in celle lattice, mentre una porzione significativa è stata definitivamente eliminata. Solo una percentuale relativamente piccola degli elementi è stata effettivamente trasformata in struttura solida.

Naturalmente, il peso complessivo della struttura convertita risulta diverso rispetto a quello dell'ultima iterazione dell'ottimizzazione topologica. In particolare, si osserva un incremento del peso totale dello spazio di design di circa il 6%, con la massa che passa da $117kg$ (misurati nell'ultima iterazione dell'ottimizzazione topologica) a $127kg$ per la struttura convertita in lattice. Questo aumento di peso è coerente con il fatto che, approssimando la struttura rispetto alla configurazione ideale ricavata dall'ottimizzazione topologica, si ottiene una soluzione meno efficiente e quindi più pesante.

A questo punto si attua il processo di modifica della struttura, al fine di ottenere un modello che, in fase di ottimizzazione dimensionale, mantenga le condizioni simmetria. Al termine di questa fase è possibile procedere all'ottimizzazione dimensionale delle sezioni delle beams.

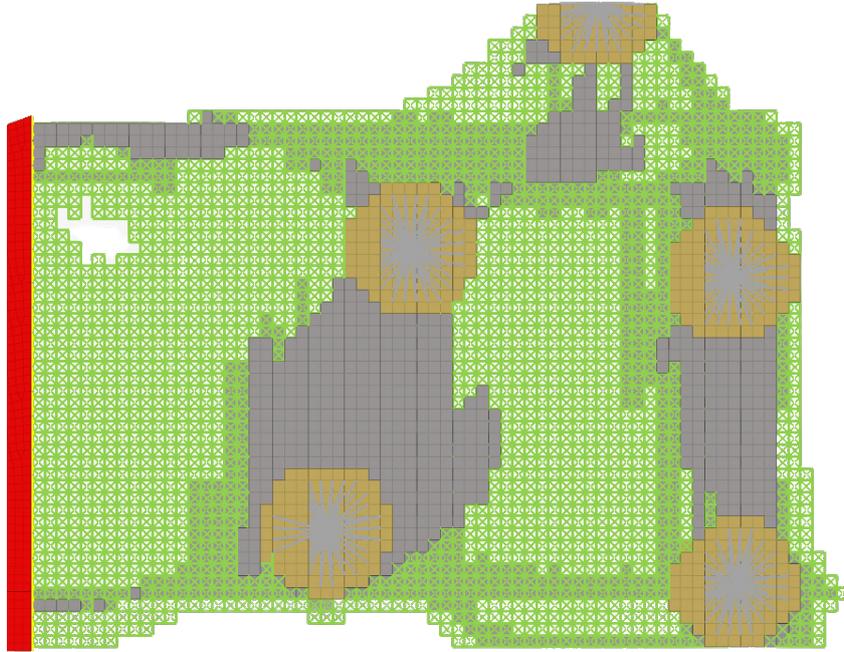


(a) Vista laterale.

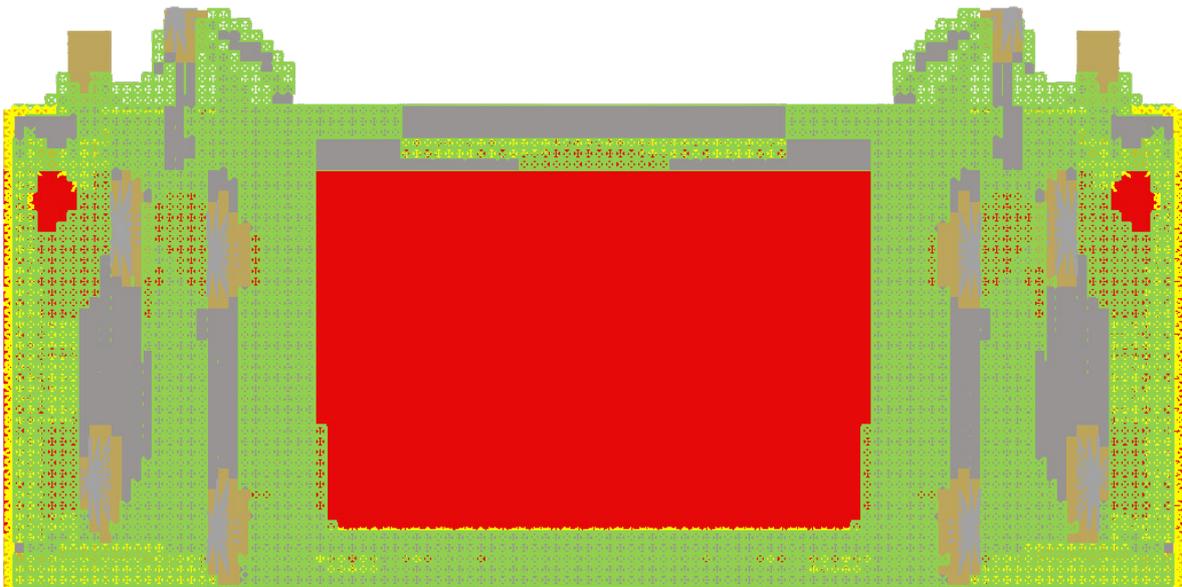


(b) Vista frontale.

Figura 6.1: Dettaglio struttura lattice del front frame, al termine della Phase I.



(a) Vista laterale.



(b) Vista frontale.

Figura 6.2: Dettaglio struttura lattice del rear frame, al termine della Phase I.

6.3 Risultati Phase II

Completata l'ottimizzazione associata alla *Phase II*, la struttura ottenuta mantiene una forma complessivamente invariata rispetto alla fase precedente. La principale differenza risiede nella dimensione delle sezioni assegnate a ciascun nodo, che risulta meglio ottimizzata rispetto alla *Phase I*.

Tabella 6.2: Risultati ottenuti dalle varie fasi dell'ottimizzazione lattice.

POROSITY	LB	UB	Massa Topologica [kg]	Massa Convertita [kg]	Massa Ottimizzata [kg]
LOW	0.05	0.6	117	127	116

Le masse relative alle diverse fasi del processo sono riportate in tabella 6.2. In questa tabella, oltre ai valori definitivi di LB e UB, vengono riportati i risultati relativi alla massa per ogni fase del processo di ottimizzazione. Nella colonna "Massa Topologica" si trova il valore della massa ottenuta al termine dell'ottimizzazione topologica eseguita nella *Phase I*, che si riferisce agli elementi solidi esagonali appartenenti allo spazio di design, la cui densità è stata ottimizzata. Nella colonna "Massa Convertita" è invece indicato il peso della struttura dopo la conversione degli elementi in celle lattice, prendendo in considerazione sia gli elementi beam sia gli elementi solidi esagonali che non appartengono allo spazio non di design. Infine, nella colonna relativa alla "Massa Ottimizzata", sono riportati i risultati ottenuti al termine dell'ottimizzazione dimensionale eseguita nella *Phase II*.

L'effetto della variazione dimensionale delle sezioni può essere osservato confrontando le tabelle 6.3 e 6.4, che si riferiscono rispettivamente alla struttura convertita e alla struttura lattice ottimizzata. Dal confronto emerge che, nonostante la sua massa sia inferiore rispetto a quella della struttura convertita, la struttura lattice ottimizzata non presenta violazione di vincoli. Al contrario, analizzando il risultato della conversione si riscontrano numerose violazioni dei vincoli, arrivando fino al 23% di violazione. Dunque, la struttura ottimizzata risponde in modo più efficiente ai carichi, con tutti i vincoli a target, portando contemporaneamente ad un alleggerimento complessivo della struttura.

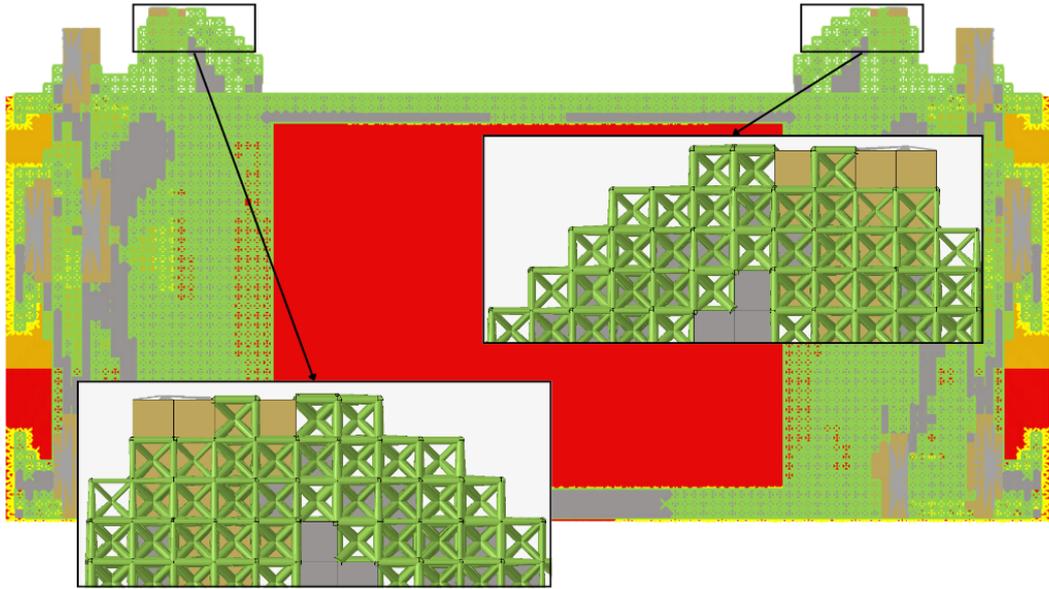
Nelle figure 6.4 e 6.3 sono mostrati i dettagli finali di alcune barre, visualizzate sia frontalmente che lateralmente. Nelle viste frontali è possibile riscontrare la simmetria del modello, rispetto al piano in mezzeria, imposta durante le fasi intermedie tra le due ottimizzazioni.

Tabella 6.3: Response ottenute dall'analisi relativa alla struttura convertita a valle della Phase I.

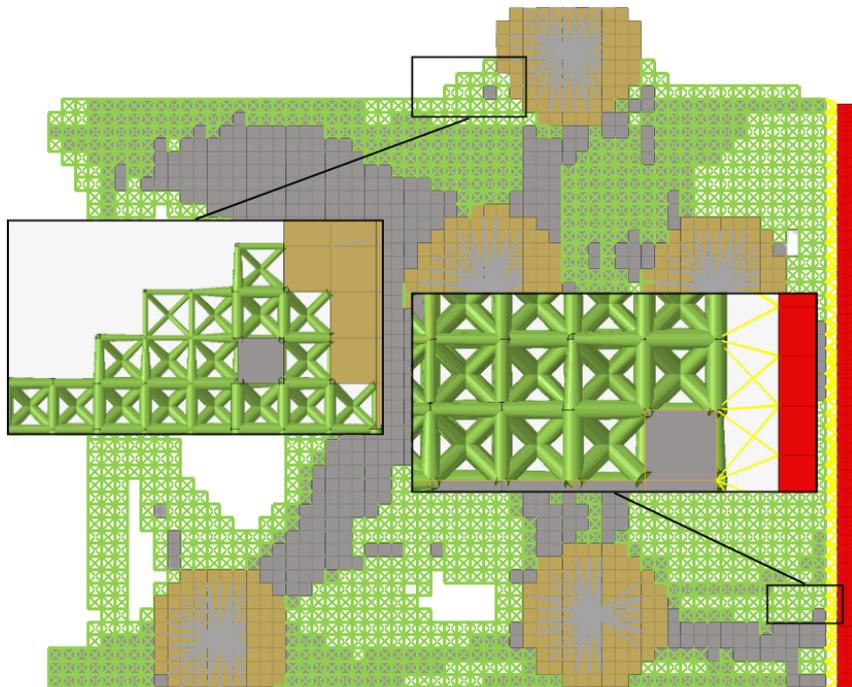
Response	Response Label	Valore della Response		Vincolo di riferimento	Violazione percentuale	Attivazione del vincolo
2	P1-X	6,98E+00	<	6,25E+00	11.7	V
3	P1-Y	6,53E+00	<	6,02E+00	8.4	V
4	P1-Z	5,82E+01	<	5,82E+01	0.1	A
5	P2-X	1,37E+01	<	2,06E+01	0.0	
6	P2-Y	5,19E+00	<	4,96E+00	4.7	V
7	P2-Z	3,07E+01	<	2,97E+01	3.3	V
8	P3-X	4,25E+01	<	5,92E+01	0.0	
9	P3-Y	2,44E+01	<	2,14E+01	14.5	V
10	P3-Z	2,96E+01	<	2,82E+01	5.0	V
11	P4-X	1,55E+01	<	2,20E+01	0.0	
12	P4-Y	7,90E+00	<	6,42E+00	23.0	V
13	P4-Z	2,58E+01	<	2,97E+01	0.0	
14	P5-X	4,98E+00	<	6,40E+00	0.0	
15	P5-Y	3,84E+00	<	3,53E+00	8.8	V
16	P5-Z	2,82E+01	<	2,76E+01	2.5	V
17	P6-X	5,65E+00	<	7,98E+00	0.0	
18	P6-Y	2,85E+00	<	2,73E+00	4.5	V
19	P6-Z	2,82E+01	<	3,57E+01	0.0	
20	P7-X	1,45E+01	<	1,83E+01	0.0	
21	P7-Y	1,36E+01	<	1,25E+01	8.9	V
22	P7-Z	2,95E+01	<	3,51E+01	0.0	
23	P8-X	4,94E+01	<	4,66E+01	6.1	V
24	P8-Y	5,75E+01	<	5,31E+01	8.3	V
25	P8-Z	4,02E+01	<	4,06E+01	0.0	A
26	P9-X	1,48E+01	<	1,59E+01	0.0	
27	P9-Y	8,65E+00	<	8,56E+00	1.1	V
28	P9-Z	4,24E+01	<	4,44E+01	0.0	
29	P10-X	7,25E+00	<	6,22E+00	16.6	V
30	P10-Y	6,30E+00	<	5,59E+00	12.6	V
31	P10-Z	4,63E+01	<	4,68E+01	0.0	

Tabella 6.4: Response ottenute dall'ottimizzazione lattice.

Response	Response Label	Valore della Response		Vincolo di riferimento	Violazione percentuale	Attivazione del vincolo
1	MassRes	5,23E+01	MIN			
2	P1-X	6,28E+00	<	6,25E+00	0.4	A
3	P1-Y	6,05E+00	<	6,02E+00	0.5	A
4	P1-Z	5,72E+01	<	5,82E+01	0.0	
5	P2-X	1,32E+01	<	2,06E+01	0.0	
6	P2-Y	4,98E+00	<	4,96E+00	0.4	A
7	P2-Z	2,98E+01	<	2,97E+01	0.4	A
8	P3-X	4,02E+01	<	5,92E+01	0.0	
9	P3-Y	2,16E+01	<	2,14E+01	0.9	A
10	P3-Z	2,83E+01	<	2,82E+01	0.4	A
11	P4-X	1,46E+01	<	2,20E+01	0.0	
12	P4-Y	6,48E+00	<	6,42E+00	0.9	A
13	P4-Z	2,46E+01	<	2,97E+01	0.0	
14	P5-X	4,73E+00	<	6,40E+00	0.0	
15	P5-Y	3,55E+00	<	3,53E+00	0.5	A
16	P5-Z	2,74E+01	<	2,76E+01	0.0	A
17	P6-X	5,20E+00	<	7,98E+00	0.0	
18	P6-Y	2,73E+00	<	2,73E+00	0.2	A
19	P6-Z	2,71E+01	<	3,57E+01	0.0	
20	P7-X	1,36E+01	<	1,83E+01	0.0	
21	P7-Y	1,26E+01	<	1,25E+01	0.6	A
22	P7-Z	2,85E+01	<	3,51E+01	0.0	
23	P8-X	4,65E+01	<	4,66E+01	0.0	A
24	P8-Y	5,34E+01	<	5,31E+01	0.5	A
25	P8-Z	3,90E+01	<	4,06E+01	0.0	
26	P9-X	1,45E+01	<	1,59E+01	0.0	
27	P9-Y	8,59E+00	<	8,56E+00	0.4	A
28	P9-Z	4,14E+01	<	4,44E+01	0.0	
29	P10-X	6,24E+00	<	6,22E+00	0.3	A
30	P10-Y	5,61E+00	<	5,59E+00	0.3	A
31	P10-Z	4,54E+01	<	4,68E+01	0.0	

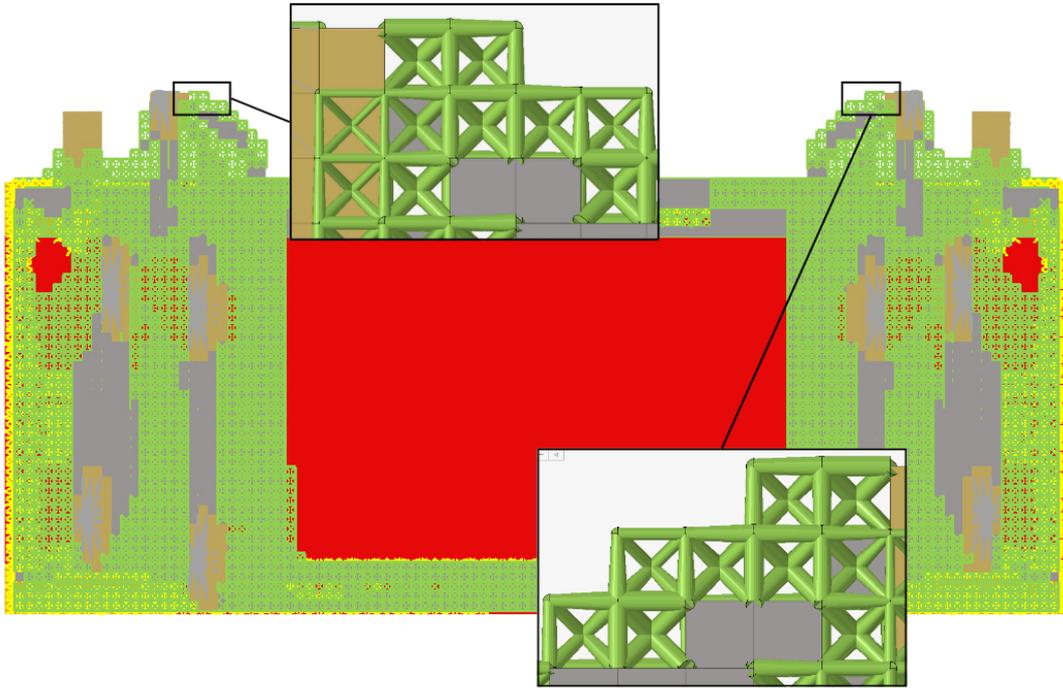


(a) Vista frontale.

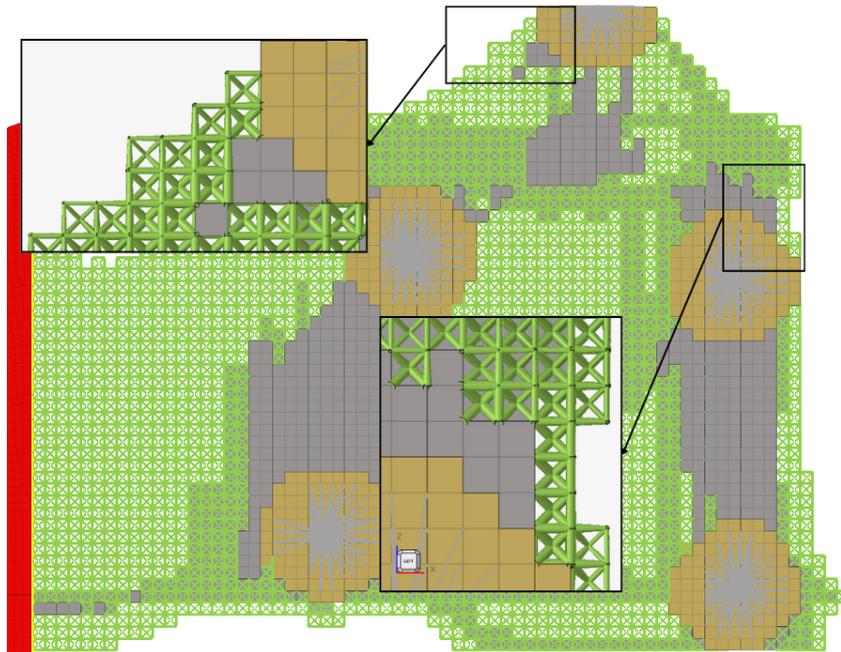


(b) Vista laterale.

Figura 6.3: Particolare delle beams per il front frame.



(a) Vista frontale.



(b) Vista laterale.

Figura 6.4: Particolare delle beams per il rear frame.

7. Conclusioni

Per avere una visione d'insieme della situazione, in tabella 7.1 sono riepilogati i risultati ottenuti dalle due diverse tecniche di ottimizzazione. In essa sono riportati i dati relativi alla massa finale del solo design space, il fattore di penalizzazione adottato dal software durante l'ottimizzazione, il numero di vincoli attivi per ciascuna struttura, la soglia di transizione solido-vuoto utilizzata nell'ottimizzazione topologica e i valori di LB e UB impiegati nell'ottimizzazione lattice.

Tabella 7.1: Tabella riassuntiva dei risultati ottenuti.

Variabile di Interesse	O. Topologica	O. Lattice
Massa [kg]	155	116
Fattore di Penalizzazione	3	1,8
Soglie di Transizione	74%	5%
		60%
Numero di Vincoli Attivi	1	16
Indice di Discretezza	20%	7%

Successivamente, in tabella 7.2, si riassumono i risultati osservati nelle tabelle 3.3 e 6.4. In essa sono riportate le response associate ai singoli subcase, per le quali si osserva il valore della response stessa e, nelle colonne successive, si osserva la lettera "A" in corrispondenza dei subcase in cui il vincolo risulta attivo.

Dall'analisi dei dati emerge che, come previsto, la massa risultante dall'ottimizzazione lattice è inferiore rispetto a quella ottenuta con l'ottimizzazione topologica classica, con una riduzione percentuale complessiva della massa del 25%. Inoltre, si osserva un numero di vincoli attivi, oltre che un valore delle response complessivamente più alto per tutte le condizioni di carico, nel caso dell'ottimizzazione lattice. Essendo entrambe le strutture all'interno dei limiti imposti dai vincoli di progettazione, si può affermare che questi risultati evidenziano la maggiore efficienza nella distribuzione del materiale per le strutture lattice. Come previsto, si evince che la migliore distribuzione del materiale è dovuta al

preservare delle densità intermedie derivate dal processo di ottimizzazione.

Tabella 7.2: Confronto del valore delle response e dei vincoli attivi per le due diverse tecniche di ottimizzazione.

Response	Valore della response		Vincoli attivi	
	Topologica	Lattice	Topologica	Lattice
2	5,96E-03	6,25E+00		A
3	5,57E-03	6,02E+00		A
4	5,74E-02	5,82E+01		
5	1,33E-02	2,06E+01		
6	4,70E-03	4,96E+00		A
7	2,96E-02	2,97E+01	A	A
8	3,98E-02	5,92E+01		
9	2,01E-02	2,14E+01		A
10	2,77E-02	2,82E+01		A
11	1,45E-02	2,20E+01		
12	6,12E-03	6,42E+00		A
13	2,40E-02	2,97E+01		
14	5,06E-03	6,40E+00		
15	3,30E-03	3,53E+00		A
16	2,65E-02	2,76E+01		A
17	5,50E-03	7,98E+00		
18	2,62E-03	2,73E+00		A
19	2,87E-02	3,57E+01		
20	1,43E-02	1,83E+01		
21	1,19E-02	1,25E+01		A
22	3,00E-02	3,51E+01		
23	4,35E-02	4,66E+01		A
24	4,97E-02	5,31E+01		A
25	3,82E-02	4,06E+01		
26	1,44E-02	1,59E+01		
27	8,12E-03	8,56E+00		A
28	4,17E-02	4,44E+01		
29	5,94E-03	6,22E+00		A
30	5,37E-03	5,59E+00		A
31	4,55E-02	4,68E+01		

Confrontando poi l'indice di discretezza (figure 7.1), che misura la percentuale di elementi appartenenti alla fascia di densità compresa tra $0.9 - 1$, si osserva che il valore ottenuto dall'ottimizzazione topologica è significativamente più alto rispetto a quello dell'ottimizzazione lattice, in conseguenza all'adozione di un fattore di penalizzazione maggiore, che

conduce ad una struttura più discreta. In maniera analoga, la percentuale di elementi appartenenti alla fascia di densità 0.0 – 0.1 è maggiore nel caso di ottimizzazione topologica classica. Al contrario, invece, le fasce intermedie di densità degli elementi sono meno popolate nel caso di ottimizzazione topologica.

Density	%	Density	%
0.0-0.1	51.8	0.0-0.1	49.2
0.1-0.2	3.2	0.1-0.2	9.0
0.2-0.3	3.7	0.2-0.3	7.2
0.3-0.4	2.0	0.3-0.4	5.2
0.4-0.5	2.5	0.4-0.5	4.2
0.5-0.6	7.1	0.5-0.6	7.2
0.6-0.7	2.1	0.6-0.7	3.8
0.7-0.8	3.8	0.7-0.8	3.8
0.8-0.9	3.6	0.8-0.9	3.3
0.9-1.0	20.2	0.9-1.0	7.0

(a) Topologica.

(b) Lattice.

Figura 7.1: Discreteness tables.

Per quanto riguarda gli algoritmi di gestione dei file, la riduzione dei tempi di esecuzione delle operazioni da parte dell'elaboratore non è quantificabile, in quanto la misurazione degli stessi, in assenza dell'algoritmo, dipende da fattori esterni associati alla qualità della macchina in uso. Tuttavia, in base all'esperienza, il guadagno è significativo quando si lavora con modelli che contengono un elevato numero di entità. L'utilizzo degli algoritmi sviluppati, infatti, è giustificato solo in presenza di modelli di grandi dimensioni, poiché bisogna considerare che tali algoritmi richiedono un certo tempo per eseguire le operazioni assegnate, e che lo svolgimento di più fasi in ambienti diversi implica tempi di configurazione maggiori rispetto all'esecuzione delle stesse operazioni in un unico ambiente. Tuttavia, nei modelli utilizzati in questa tesi, gli algoritmi si sono dimostrati indispensabili per la gestione efficiente delle operazioni.

Come accennato in precedenza, i vantaggi offerti dalle due tecniche di ottimizzazione possono essere sfruttati solo impiegando tecniche di fabbricazione additiva. Dunque, ottenuta la struttura ottimizzata, è necessario trasferire al reparto produzione tutte le informazioni geometriche necessarie alla manifattura dei vari componenti. Infatti, al termine delle fasi di ottimizzazione si implementano nel modello tutte quelle caratteristiche che in fase di modellazione risultano superflue. A tal fine è necessario, dopo aver completato tutti i

passaggi che portano al componente ottimizzato, convertire la struttura finale in un modello CAD che presenti geometrie continue, raccordando le beam nei punti di giunzione. Successivamente, occorre modificare il modello nei punti di applicazione del carico, che finora sono stati rappresentati mediante volumi di elementi inclusi nel non-design space. Questi punti vanno ridefiniti per garantire un alloggiamento corretto per gli elementi di collegamento tra il componente e gli elementi esterni ad esso, come le sospensioni o il corpo vettura in fibra di carbonio. Solo dopo aver perfezionato il file CAD, si può procedere alla conversione in formato STL. Questo viene poi importato in un software slicer, al fine di impostare i parametri di stampa e generare eventuali supporti. In ultimo, lo slicer genera un codice in grado di dare alla macchina delle istruzioni in sequenza riguardo le operazioni da eseguire per la creazione del componente fisico.

La produzione di componenti tramite le suddette tecniche di fabbricazione, tuttavia, presenta anche importanti limitazioni. Uno dei contro relativi allo sviluppo di strutture ottimizzate tramite topologica o lattice, consiste nel fatto che, con le tecnologie attualmente disponibili sul mercato, non sarebbe possibile produrre un telaio delle dimensioni considerate in un'unica soluzione, a causa delle limitazioni di volume delle camere di stampa. Una possibile soluzione a questo problema potrebbe essere la suddivisione della struttura in più elementi, così da distribuire la produzione su più macchine. Inoltre, i costi legati all'investimento iniziale per l'implementazione di tecniche di FA sono particolarmente elevati e non sempre giustificabili per le applicazioni nella produzione di massa. Ancora, la produttività delle macchine in uso è particolarmente più bassa rispetto alle classiche linee di produzione impiegate negli impianti industriali e l'uso di tali macchine richiede degli standard di formazione e sicurezza più elevati rispetto alle normali macchine utensili.

Tuttavia, i vantaggi in termini prestazionali di una struttura di tipo lattice sono particolarmente evidenti in base ai risultati ottenuti in questa sede. Inoltre, la direzione di sviluppo per le tecnologie di AM prevede un progressivo aumento dei volumi di stampa, della velocità e della precisione delle macchine, contemporaneamente alla riduzione dei prezzi di acquisto e delle problematiche dovute alla manipolazione dei materiali, attraverso sistemi sempre più automatizzati per la gestione dei pezzi prima e dopo il processo di stampa. Infine, un aspetto particolarmente vantaggioso sia in termini economici che ecologici della FA risiede nella possibilità di progettare il componente in un'unica sede, per poi produrlo ovunque sia disponibile una stampante 3D. Questo porterebbe, in un contesto di ampia diffusione delle tecnologie di fabbricazione additiva, ad una notevole riduzione dei costi e delle emissioni legate al trasporto dei componenti

Bibliografia

- [1] C. Chu, G. Graf e D. W. Rosen, “Design for additive manufacturing of cellular structures,” *Computer-Aided Design and Applications*, vol. 5, n. 5, pp. 686–696, 2008.
- [2] Istituto della Enciclopedia Italiana fondata da Giovanni Treccani S.p.A., *La Treccani*. indirizzo: <https://www.treccani.it/>.
- [3] n.d., *New Product Development Stages*. indirizzo: <https://productlifecyclestages.com/new-product-development-stages/>.
- [4] Altair Engineering Inc., *Practical Aspects of Structural Optimization - A Study Guide*. 2018. indirizzo: https://www.advanced-eng.cz/wp-content/uploads/2021/06/ebook_Practical_Aspects_of_Optimization_with_Altair_OptiStruct_2021.pdf.
- [5] A. Gugliotta, A. Somà e N. Zampieri, *Elementi Finiti*. Quine, 2022, p. 320, ISBN: 8831284061.
- [6] Wikipedia, *Wikipedia- The Free Enciclopedia*. indirizzo: https://en.wikipedia.org/wiki/Main_Page.
- [7] Altair Engineering Inc., *Practical Aspects of Finite Element Simulation - A Study Guide*. 2019. indirizzo: https://www.advanced-eng.cz/wp-content/uploads/2019/11/eBook_Practical-Aspects-of-Finite-Element-Simulation.pdf.
- [8] Altair Engineering Inc., *Altair HyperWorks 2023.1*. indirizzo: <https://2023.help.altair.com/2023.1/hwdesktop/hwx/index.htm>.
- [9] A. Schumacher, “Optimierung mechanischer Strukturen - Grundlagen und industrielle Anwendungen,” 2013.
- [10] P. Minetola et al., *Additive Manufacturing Systems and Materials*, Course slides.
- [11] Altair Engineering Inc., *Altair OptiStruct 2023.1*. indirizzo: <https://2023.help.altair.com/2023.1/hwsolvers/os/index.htm>.
- [12] Altair Engineering Inc., *Altair Compose 2023.1*. indirizzo: https://2023.help.altair.com/2023.1/compose/help/en_us/index.htm.

Ringraziamenti

