

POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Civile

Tesi di Laurea Magistrale

**UTILIZZO DEL CODICE FEM CODE-ASTER
PER LA MODELLAZIONE DI STRUTTURE IN
INGEGNERIA CIVILE**



Relatore

Firma del relatore

prof. Stefano Invernizzi

.....

Candidato

Firma del candidato

Marco Borgna

.....

Dicembre 2018

1	Introduzione.....	5
2	Inquadramento Generale	7
2.1	Introduzione al software libero	7
2.2	Le origini della filosofia “Open-Source”	7
2.3	Il movimento “Freeware”.....	8
2.4	Il progetto e la creazione della rete “Open-Source”	9
3	Il sistema operativo ed i Software	11
3.1	L’ambiente GNU/Linux	11
3.2	L’ambiente Virtual Box ed il pacchetto Ubuntu	12
3.3	La piattaforma Salomè-Meca.....	13
3.4	Il Software ad elementi finiti: Code-Aster.....	14
3.5	Il modulo di elaborazione grafica dei dati “ParaVis”	18
4	Il metodo degli Elementi Finiti (FEM).....	19
4.1	Gli aspetti generali del metodo FEM	19
4.1.1	Sistemi ad un grado di libertà.....	19
4.1.2	Sistemi a più gradi di libertà	20
4.2	Esempio pratico di analisi all’interno del software FEM.....	22
4.3	Modellazione della geometria e “Meshing”	24
4.4	Funzioni di forma e procedimento del software	26
4.5	Proprietà del materiale	27
4.6	Condizioni al contorno, di carico e di stato.....	27
5	Code-Aster: guida alla lettura dei manuali	29
5.1	La documentazione di Code-Aster.....	29
5.2	Il fenomeno meccanico all’interno di Code-Aster	30
5.3	Alcuni preziosi consigli prima di iniziare l’analisi	31
5.4	Il manuale Utilisation.....	33
5.5	Il manuale Reference.....	35
5.6	Il manuale Validation	44
5.7	I campi di analisi possibili e le funzioni che li richiamano.....	47
5.7.1	Le grandezze associate ai campi di forze	47
5.7.2	Calcolo degli Stati di Tensione (Opzioni di calcolo)	49
5.7.3	Calcolo delle forze nodali e delle reazioni generalizzate	52
5.8	Il metodo di analisi FEM del software	54
5.8.1	Discretizzazione	54
5.8.2	Tipologia di elementi e modello di calcolo	55
5.8.3	Orientamento degli elementi trave	56
6	Analisi di elementi strutturali tramite Code-Aster	59

6.1	Introduzione sui moduli e le analisi effettuate	59
6.2	Avviamento del programma e definizione degli ambienti di lavoro.....	60
6.3	Mensola soggetta a carichi verticali	61
6.3.1	Mensola incastrata: generazione della geometria.....	62
6.3.2	Mensola incastrata: generazione della Mesh.....	65
6.3.3	Creazione del Command file tramite il modulo Asterstudy.....	67
6.3.4	Valutazione dei risultati all'interno del modulo Paravis	77
6.4	Trave a due campate soggetta a carichi verticali	84
6.4.1	Trave a due campate: geometria, Mesh e Command File	85
6.4.2	Trave a due campate: analisi dei risultati e verifica	90
6.5	Portale soggetto a carichi verticali	93
6.5.1	Portale: geometria e Mesh.....	94
6.5.2	Portale: generazione del Command File	94
6.5.3	Portale: analisi dei risultati e verifica.....	99
7	Conclusioni.....	101
8	Allegati	102
8.1	Mensola soggetta a carichi verticali: grafici	102
8.2	Mensola soggetta a carichi verticali: "Command File"	113
8.3	Trave a due campate soggetta a carichi verticali: grafici.....	120
8.4	Trave a due campate soggetta a carichi verticali: "Command File"	127
8.5	Portale soggetto a carichi verticali: "Command File".....	132
9	Bibliografia.....	141
10	Ringraziamenti	143

1 Introduzione

Nella presente Tesi di Laurea viene studiato il comportamento di elementi strutturali schemi soggetti a differenti combinazioni di carico tramite l'ausilio del software agli elementi finiti *Code-Aster*; un software *Open-Source* rilasciato dalla società elettrica transalpina EDF (*Electricité De France*). Il software viene rilasciato all'interno di una piattaforma multi-operativa di nome *Salomé-Mecà* la quale integra al suo interno diversi programmi applicativi, sempre *Open-Source*, come ad esempio il modulo *Geometry*, il modulo *Mesh* ed il modulo *Paravis*; questi moduli saranno il mezzo impiegato per generare, calcolare e visualizzare le azioni e gli effetti delle azioni agenti sui nostri casi studio, i quali verranno infine sottoposti a verifica attraverso le NTC 2018 (Norme Tecniche per le Costruzioni D.M. 17/01/2018).

Nel capitolo 1 viene spiegata la nascita e gli ideali del movimento *Open-Source*, un movimento che ha permesso la nascita di software ad accesso libero e gratuiti per tutti, creando una rete di sviluppo che consente a programmi, come quelli noi useremo, di essere costantemente aggiornati tanto da poter essere considerati una valida alternativa ai programmi a pagamento in commercio.

Nel capitolo 2 si entra nel cuore della piattaforma operativa *Salome-Mecà* spiegando il funzionamento dei moduli che utilizzeremo per la modellazione geometrica CAD (*Geometry*), per la discretizzazione degli elementi strutturali (*Mesh*) ed il modulo per la visualizzazione grafica (*Paravis*). Tutti i programmi saranno eseguibili dopo aver installato il Sistema Operativo Linux all'interno di un applicativo di supporto grafico, Virtual Box, che ci consentirà di lavorare in un ambiente parallelo all'ambiente Windows consentendoci di poter utilizzare i sistemi operativi, anche contemporaneamente, senza creare problemi di esecuzione né conflitti tra i programmi.

Nel capitolo 3 viene spiegata la teoria che consente ai software agli elementi finiti di poter analizzare e governare lo studio degli elementi strutturali; viene spiegata la metodologia di analisi computazionale e le procedure per la discretizzazione degli elementi strutturali che ci consentiranno di creare il nostro modello geometrico.

Nel capitolo 4 viene trattata tutta la documentazione ed i manuali che ci hanno permesso di utilizzare *Code-Aster* descrivendo solamente le parti utilizzate nelle analisi dei nostri casi studio; tali manuali si suddividono per argomenti e descrivono le metodologie di utilizzo degli applicativi, le funzioni e i campi di analisi da impiegare per il corretto utilizzo e conclude descrivendo alcuni esempi effettuati da utenti (e imprese) messi a disposizione come supporto pratico.

Nel capitolo 5 vengono descritti i casi studio analizzati; vengono trattati elementi strutturali soggetti a carichi esterni in combinazione tra loro; partendo dall'analisi di una mensola incastrata soggetta a combinazioni di carico differenti, vengono calcolate le sollecitazioni agenti e confrontate analiticamente tramite calcoli effettuati mediante Excel per verificarne la correttezza dei risultati. Un secondo caso studio tratta una trave a più campate in semplice appoggio soggetta ad azioni verticali, su di essa vengono valutati i diagrammi delle sollecitazioni, e dopo un confronto tramite Excel viene eseguita la verifica per mezzo di un diagramma d'interazione. Infine viene studiata una struttura costituita da telai che si sviluppa in altezza su tre piani; l'obiettivo sarà generare un codice di calcolo in grado di poter effettuare una verifica agli stati limiti ultimi su una porzione

della struttura; il codice di calcolo sviluppato dovrà essere in grado di potersi adattare ad una struttura qualunque generata dalle caratteristiche geometriche e fisiche che l'utente digiterà all'interno del codice.

Lo scopo principale è quello di guidare il lettore alla conoscenza di questo linguaggio di programmazione che, tramite semplici ma inevitabili modifiche, può essere adattato ad una moltitudine di casi studio più o meno complessi.

2 *Inquadramento Generale*

2.1 *Introduzione al software libero*

Nell'universo informatico troviamo una grande quantità di software di qualsiasi genere e tipo, dove i detentori dei diritti d'autore perfezionano e rilasciano versioni dei suddetti software sempre più elaborate, complesse e ricercate in modo da soddisfare le richieste del mercato; questo mondo deve ovviamente ragionare con le regole del mercato in modo da sviluppare programmi che soddisfino le esigenze degli utenti finali per far sì che questi comprino la licenza d'uso; tuttavia esiste anche un'alternativa, ossia i programmi *Open-Source* a licenza gratuita.

I suddetti software "*Open-Source*", ovvero tutti quei software che vengono appunto definiti letteralmente ad accesso libero prevedono non solo la possibilità di usufruirne gratuitamente e liberamente, ma anche la possibilità di partecipare allo sviluppo del codice sorgente di programmazione del software stesso. Vi sono parecchi software che appartengono a questo mondo che non richiede licenze e permessi a pagamento, e di questi ve ne sono alcuni che sfruttano le potenzialità di grandi aziende pubbliche, private o community di programmatori che investono sullo sviluppo di un software "*Open-Source*" in sinergia con il contributo degli utenti utilizzatori finali che, tramite l'utilizzo del software stesso in ambiente lavorativo o accademico, possono contribuire allo sviluppo del codice sorgente fornendo informazioni ed eventuali problemi in seguito all'esperienza diretta.

E' importante non confondere il concetto di "*Open-Source*" con quello di "*freeware*". I programmi "*freeware*" pur essendo liberamente utilizzabili, restano comunque di proprietà dell'azienda o della persona fisica che ha prodotto il codice sorgente, ne consegue che un software "*freeware*", pur restando gratuito, sarà sviluppabile e modificabile solo ed esclusivamente da chi ne detiene i diritti. Anche per questa ragione il concetto di "*Open-Source*" risulta essere un concetto più potente, in quanto non solo viene seguita la filosofia di rendere il software gratuito e scaricabile da chiunque, ma diviene una vera e propria scelta di campo in favore della condivisione, consentendo all'utente finale di correggere eventuali errori nel codice sorgente e partecipare allo sviluppo del software nelle versioni future.

2.2 *Le origini della filosofia "Open-Source"*

La filosofia "*Open-Source*" deve le sue origini al movimento per il software libero capeggiato dal suo portavoce ed appassionato Richard Stallman.

L'idea del software libero trova le sue radici nei primi anni '80, quando lo sviluppo del codice cominciò a circolare dalle università alle aziende, ponendo un pesante freno alla collaborazione che caratterizzava il lavoro di gran parte dei programmatori e dei sistemisti dell'epoca, soprattutto con i patti di non divulgazione che le aziende facevano obbligatoriamente firmare ai programmatori e sviluppatori che assumevano.

Nonostante ciò il software venne sviluppato come bene di consumo per il mercato globale molti anni prima, precisamente si può dire che il software e l'hardware siano due facce della stessa medaglia, e quindi nati come un'unica entità.

I maggiori produttori di hardware dei primi anni '60 e '70 vendevano i loro costosissimi componenti hardware corredati dai programmi software specifici per quei componenti, siccome durante le fasi embrionali dello sviluppo globale dei sistemi informatici non era ancora stato imposto uno standard delle architetture hardware. A queste condizioni, il software risultava privo di un valore commerciale di rilievo e veniva quindi distribuito con il codice sorgente insieme ai componenti hardware, il tutto veniva commercializzato in maniera tale che chiunque li utilizzasse potesse personalizzarlo e anche condividere eventuali perfezionamenti del software acquistato.

Con l'evoluzione dell'industria lineare, le linee di produzione di prodotti informatici erano ormai una realtà, questo portò ad un sostanziale ridimensionamento dei prezzi che consentì anche alle comuni famiglie di acquistare un Personal Computer, mentre prima i principali clienti potevano essere laboratori di ricerca, università o grandi aziende.

La diffusione dei compilatori e degli interpreti del codice sorgente rese più facile la comunicazione tra componenti hardware di produttori diversi, e quindi apparve chiaro che il ruolo del software non poteva più essere limitato o precluso dall'hardware: nacque in questo modo un nuovo prodotto commerciale; questa fase vide dunque la nascita di tante aziende informatiche che occupandosi dello sviluppo dell'area "software" crearono sempre più complessi ed innovativi livelli di codice, che col passare del tempo vennero brevettati e messi in vendita segnando così l'inizio del cosiddetto "*Closed-Source*", ovvero il periodo in cui si rese il codice sorgente un linguaggio non più accessibile e modificabile dall'utente finale. Nasce in questo modo un nuovo prodotto commerciale in un mercato ancora acerbo ma con immense potenzialità.

2.3 Il movimento "*Freeware*"

Negli ambienti accademici il concetto di condivisione e cooperazione aperta del codice rimase tuttavia ancora vivo, forte del fatto che la condivisione delle idee è un vantaggio per tutti ed in particolare per la ricerca. Su questo entusiasmo nei primi anni '80, uno studente del MIT, Richard Stallman fondò la "Free Software Foundation" (FSF), un'organizzazione no-profit per lo sviluppo, la promozione e la divulgazione del software libero in tutte le aree informatiche: dalla ricerca al linguaggio codice e dai mezzi di divulgazione ai componenti di protezione. Stallman chiamò il progetto GNU, un sistema operativo ispirato a Unix, il cui scopo fu di rendere libero ed accessibile il sistema operativo.

L'obiettivo di tale progetto venne protetto tramite lo sviluppo di una licenza pubblica la "GNU-General Public License" (GNU-GPL) e risulta essere il tipo di licenza più diffusa nei progetti "*Open-Source*". Fu sotto questa licenza che all'inizio degli anni '90 prese corpo il sistema operativo GNU/Linux di Linus Torvalds divenendo il pilastro su cui si basarono svariati progetti legati alla creazione di programmi ad accesso libero.

Linux rappresenta il progetto di software libero di maggior complessità e diffusione. Il successo di Linux lo si deve a tutti quei programmatori, ingegneri informatici o semplici appassionati che hanno nel corso degli anni sviluppato il codice sorgente, e che tutt'ora continuano a dare il loro prezioso supporto.

2.4 Il progetto e la creazione della rete “Open-Source”

Fin dalla prima comparsa dei Personal Computer il software libero ebbe una crescita costante, ma ancora non aveva fatto breccia nel mondo del Business e nelle grandi compagnie che vedevano questo filone culturale con occhio critico; il motivo era il basso appeal commerciale, in quanto alla parola “Free” poteva essere associato un doppio significato: “gratis” e “libero”, e questa ambiguità causò parecchie controversie all’interno delle grandi comunità che consideravano, il prodotto gratuito, ma inesorabilmente di bassa qualità o comunque poco sviluppato in termini di sicurezza ed affidabilità.

Per questa ragione alcuni esponenti del software libero crearono il termine “Open-Source” e nel 1998 diedero vita al consorzio OSI (*Open-Source Initiative*) per iniziare una nuova fase di promozione e di pubblicità.

Ben presto questo consorzio raggiunse l’obiettivo prefissato, infatti si era aperta una breccia nel mondo dell’informatica in quanto venivano tralasciate le questioni etiche e filosofiche, ed il concetto vero di software libero assumeva un nuovo significato più fedele alle richieste commerciali degli utenti finali.

La posizione del consorzio OSI si può riassumere nelle seguenti frasi, citando il pensiero che si trova nella home page del sito del consorzio: “...mantenere la stessa licenza del software libero, mantenere le stesse pratiche collaborative e di conoscenza comune, ma estraniare il concetto di libertà e ideologia liberale di cui si faceva e si fa ancora oggi portavoce Stallman.” (cit. www.Open-Source.org).

L’idea di creare un qualcosa che a livello di marketing e di commerciabilità fosse più accattivante creò non poche divisioni, tuttavia in termini di benefici si raggiunsero risultati eccezionali. Infatti a partire dai primi anni 2000 tantissime aziende, multinazionali e non solo, si affacciarono a questo mondo che prima poteva sembrare una chimera, ma poi si rivelò affascinante.

Un esempio lampante di come l’*Open-Source* abbia dato linfa vitale a nuovi prodotti di successo globale può essere ad esempio il browser “Mozilla-Firefox”, nato dal rilascio del codice sorgente, sotto licenza *Open-Source*, del famoso browser Netscape da parte del colosso statunitense America Online (AOL).

Nonostante dunque le divergenze tra i due consorzi e i dibattiti accesi, le due filosofie corrono sullo stesso binario, e questo ha prodotto diverse collaborazioni tra i due enti che promuovono il software libero.

Su questa filosofia si basa l’idea della EDF (*Électricité de France*), la società elettrica francese, ovvero sviluppare un software aziendale all’interno dei suoi uffici e di renderlo disponibile in formato *Open-Source*; in modo che si possa capire come non vi siano più interessi filosofici o politici, ma la libertà di progredire allo sviluppo del linguaggio sorgente tramite l’aiuto di ogni singolo utente.

3 Il sistema operativo ed i Software

3.1 L'ambiente GNU/Linux

Nel 1991 prende vita il progetto Linux, un Kernel sviluppato negli ambienti accademici che inizia ad essere disponibile tramite la rete internet via FTP; da quel momento il Kernel Linux (o più propriamente GNU/Linux) diverrà la base per la creazione di molti progetti *Open-Source* e software sviluppato all'interno del progetto GNU.

Da allora il progetto di kernel Linux inizia a crescere esponenzialmente e, grazie alla grande possibilità di condivisione tramite il Web, da sistema operativo di nicchia diviene ben presto accessibile anche a quella branchia di utenti da sempre indirizzati sui sistemi operativi a pagamento.

Come già accennato le potenzialità di Linux iniziano ad essere considerate una meravigliosa occasione per lo sviluppo di prodotti di qualità e sempre più performanti; lo sviluppo viene considerato decentralizzato in cui ogni autore non ha un ruolo definito e dunque questo ha portato il sistema operativo ad essere rilasciato con licenza "GNU-General Public License" (GNU-GPL) basato sui principi del *Copyleft* (permessi d'autore) che assicurano quattro importanti libertà:

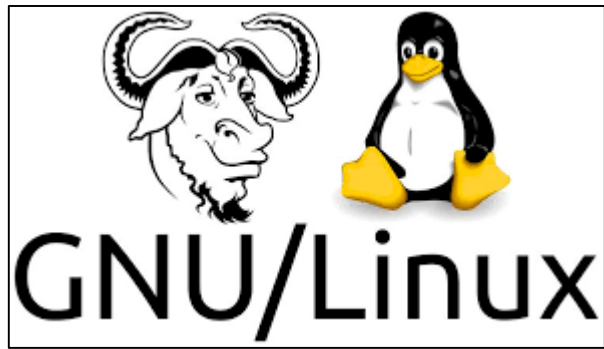


Figura 3.1 Logo GNU/Linux

- 1) Libertà di eseguire un programma per qualunque scopo;
- 2) Libertà di studiare come funziona il programma e di adattarlo ai propri bisogni;
- 3) Libertà di redistribuzione ad altri;
- 4) Libertà di migliorare il programma e ridistribuire il miglioramento ad altri.

Un programma si dice software libero se consente tutte queste libertà.

Sotto queste prerogative gli ultimi anni hanno visto una moltitudine di software di calcolo agli elementi finiti basati essenzialmente sulla piattaforma kernel Linux, e non solo, che ha permesso lo sviluppo del codice sorgente degli stessi (trattandosi per la maggior parte di software *Open-Source*), fra questi se ne possono ricordare alcuni ad esempio: *Elmer* (<https://www.csc.fi>), *OOFEM* (<http://oofem.org/doku.php>), *Open-Sees* (<http://opensees.berkeley.edu/>) e *Code-Aster* (www.code-aster.org).

In questo quadro generale di lavoro questo scritto intende mostrare alcune esperienze di modellazione numerica in campo lineare svolte con uno di questi codici, ovvero il codice di calcolo *Code-Aster*, il quale risulterà integrato con la piattaforma di *Salomè-Meca* e con il software di visualizzazione grafica *Paraview* (tramite l'ausilio del tool *ParaVis*), il tutto sarà fruibile attraverso un programma di visualizzazione virtuale,

Virtual Box, sul terminale che consentirà al Kerner *Linux* di poter essere operativo sul computer senza influire sulla velocità di esecuzione del software.

3.2 L'ambiente *Virtual Box* ed il pacchetto *Ubuntu*

Al fine di non appesantire la memoria del computer, di non creare interferenze o possibili conflitti tra software *Windows* e software *Open-Source*, si è deciso di operare tramite l'ambiente di lavoro *Virtual Box*.

Virtual Box viene definita letteralmente la macchina virtuale, ovvero si tratta di un programma che una volta installato sul proprio computer lavora e ragiona letteralmente come un lettore CD/DVD; infatti quando si vuole utilizzare la macchina virtuale la prima fase consiste nel cliccare sull'icona, che corrisponderà all'apertura del cassetto del lettore CD/DVD, in seguito la seconda fase consiste nel scaricare all'interno della macchina programmi, sistemi operativi, software di vario genere senza che questi attacchino o danneggino il sistema operativo di base del computer, questo corrisponderà all'inserimento di un CD/DVD all'interno del lettore, infine la terza fase prevede l'uso del S.O., del software,... installati sulla macchina, che corrisponde all'ascolto del CD/DVD.

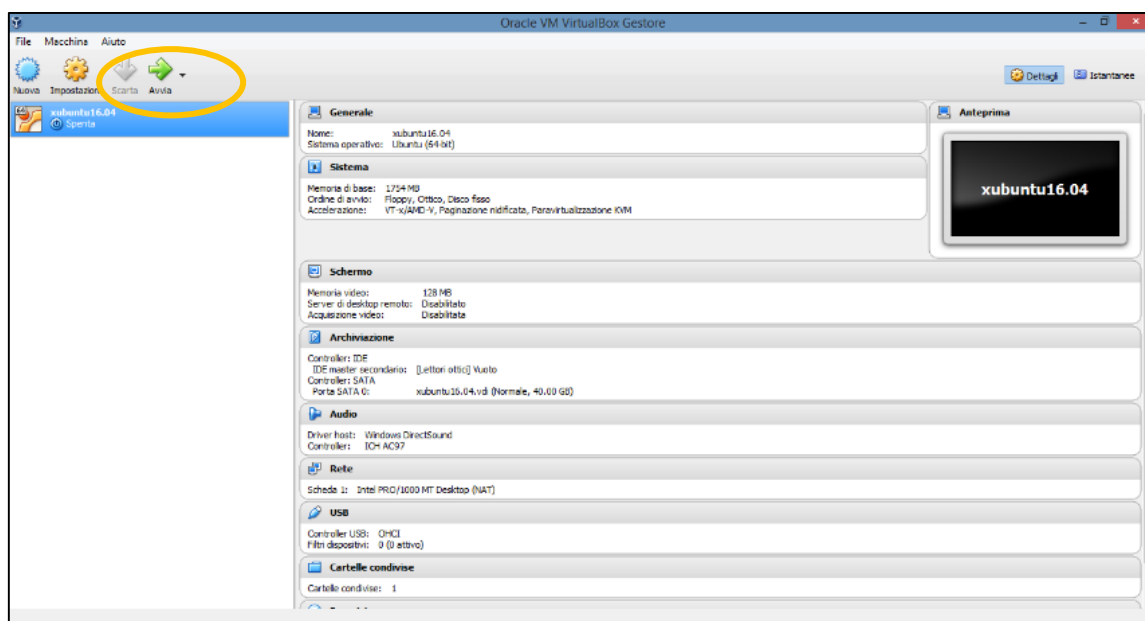


Figura 3.2 Schermata di apertura dell'ambiente *Virtual-Box*

Questo ambiente di lavoro è quindi un ambiente virtuale, all'interno del quale si può lavorare proprio come se noi accendessimo un altro computer. Ed è proprio quello che è stato fatto; infatti si è installato la versione del sistema operativo *Linux* all'interno della macchina virtuale in modo da operare in questo ambiente virtuale.

Per poter installare *Linux* all'interno della macchina si è scelto un percorso molto semplice e ben definito; essendo *Linux* un sistema operativo *Open-Source*, con tutti i vantaggi già ampiamente descritti nel capitolo 1, si è deciso di scaricare tramite la pagina web di www.ubuntu.org, la versione *Ubuntu 16.04*; il suddetto pacchetto è stato installato su *Virtual Box*, in quanto al suo interno si trova il sistema operativo *Linux* con alcuni programmi e funzioni, sempre *Open-Source*, già preinstallate all'interno del pacchetto.

Nella fattispecie abbiamo installato un nuovo S.O. con preinstallati i programmi e le applicazioni di base.

3.3 La piattaforma Salomè-Meca

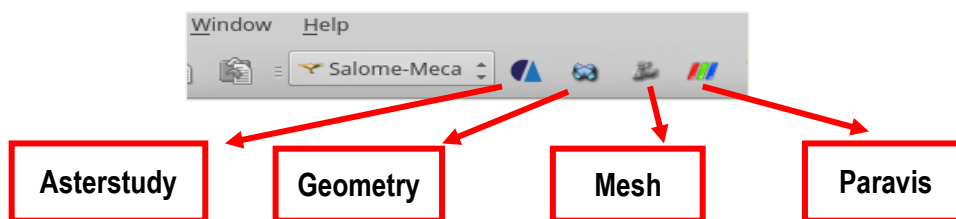
Con il termine *Salomè-Meca* (<http://www.Salome-platform.org>) si vuole indicare una piattaforma di lavoro integrata per simulazioni fisiche che coopera strettamente con il nostro software di calcolo agli elementi finiti *Code-Aster*. Questa multi-piattaforma dispone di un architettura aperta e flessibile, composta da *tools* e moduli *Open-Source* che riescono in questo modo a comunicare all'interno di un solo ed unico spazio di lavoro in maniera efficace e senza interferenze. *Salomè-Meca* è dunque una piattaforma integrata, che viene impiegata principalmente come pre/post processing da diversi *tools* di modellazione multi-fisica; anche questa piattaforma è rilasciata sotto licenza GNU/GPL, ciò significa che il codice sorgente risulta essere accessibile e modificabile.



Figura 3.3 Logo Code-Aster / Salomè-Meca

Le particolarità di questa piattaforma sono un'interfaccia grafica e testuale in linguaggio *Python* (www.python.org). L'interfaccia grafica supporta l'interoperabilità CAD/CAE (modellazione CAD) per la creazione geometrica dei modelli strutturali, o per l'importazione e l'esportazione tramite il formato predisposto dal programma, inoltre offre strumenti sufficientemente semplici per l'integrazione di nuove componenti (funzioni di calcolo matematico e/o geometrico) gestendo in maniera efficace problemi di diversa natura.

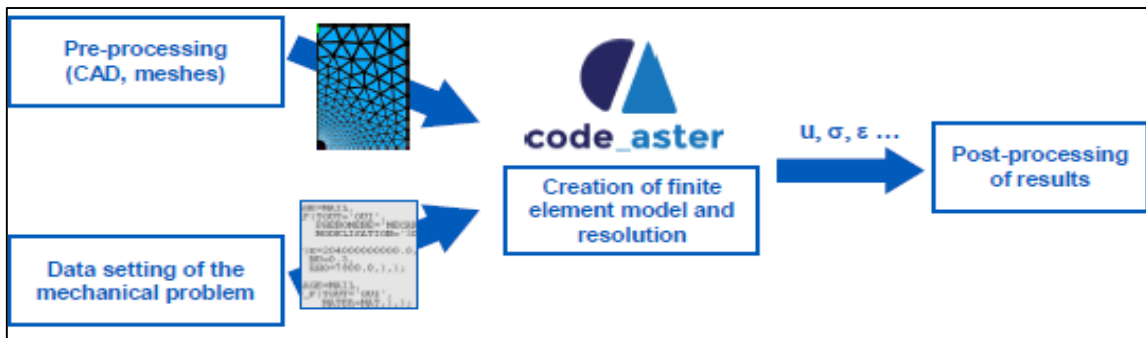
Le caratteristiche principali del software si basano generalmente su un'interfaccia utente generica (GUI), *User-Friendly*, che aiuta a ridurre i tempi di calcolo nel passaggio da un'ambiente all'altro, senza dimenticare che è possibile poter accedere alla console di programmazione in linguaggio *Python* integrata (TUI).



Le funzioni principali della piattaforma possono essere sintetizzate nei seguenti punti:

- Definizione del modello geometrico tramite il modulo *Geometry* (in fase di creazione e/o modifica), oppure importarlo ed esportarlo utilizzando i formati *BREP*, *IGES* e *STEP*;

- Definizione della maglia (modulo *Mesh*) degli elementi geometrici (la stessa può essere anche importata/esportata dall'esterno come file *.med*);
- Gestione delle proprietà fisiche e delle quantità associate agli elementi geometrici (definendo il materiale, il tipo di analisi, le forze, i vincoli, ...);
- Esecuzione dei calcoli utilizzando un risolutore (modulo *Asterstudy*);
- Visualizzazione dei campi dei risultati in 1D, 2D e 3D ed esportazione delle immagini della loro visualizzazione in un formato appropriato (*.rmed*, *.med*), utilizzando un modulo di visualizzazione grafica (ParaVis);
- Visualizzazione dei risultati studiati tramite diversi formati di file (*.resu*, *.rmed*, ...).



2.4 Schema di analisi all'interno di Salomé-Meca

3.4 Il Software ad elementi finiti: Code-Aster

In questo capitolo non ci soffermeremo troppo sugli aspetti, le procedure e le funzioni di *Code-Aster*, tratteremo piuttosto aspetti che introducono al software, alle sue potenzialità ed ai suoi moduli di lavoro.

Code-Aster è un software *Open-Source* di simulazione e calcolo basato sulla tecnica degli elementi finiti (FEM) ed è stato sviluppato principalmente dal dipartimento “*Analyses Mécaniques et Acoustiques*” del servizio R&D (Ricerca & Sviluppo) di EDF, Électricité de France, dove appunto “*ASTER*” è l’acronimo di “*Analyses des Structures et Thermo-mécanique pour des Études et des Recherches*” (www.edf.fr).

Il codice ha origine negli ambienti aziendali della società; il quale, a partire dal 1989, per rispondere alle necessità di progettazione e modellazione si sviluppa esponenzialmente all’interno dell’azienda stessa, successivamente viene reso disponibile sotto la licenza GNU/GPL nel 2001.

Nella versione in ambiente *CAELinux* (www.caelinux.com) *Code-Aster* è direttamente integrato con la piattaforma *Salomé-Meca*; in questa versione si può altresì notare come il codice risulti decisamente robusto, infatti contiene circa 1,500,000 linee di codice (scritte sia in linguaggio *Fortran* sia in *Python*) ed è, come nella logica degli *Open-Source*, in continuo sviluppo ed aggiornamento. La parte di validazione è profondamente curata (con molti confronti con risultati sperimentali e *Benchmarks* con altri codici), ed è stata svolta da soggetti indipendenti da EDF. La documentazione consta di oltre 14,000 pagine dove è possibile trovare tutto ciò che può servire per l’utilizzo del software, anche se il codice non può dirsi essere perfetto, infatti all’interno della documentazione ufficiale in alcune particolari metodologie vengono sconsigliati alcuni passaggi ed operazioni che

possono inevitabilmente creare qualche conflitto interno nella modellazione/risoluzione; la documentazione, è molto ingente e complessa, questo ha reso complicato la gestione di funzioni e operazioni di carattere fisico diverso, ma rispettando tutti i campi di analisi e le procedure passo dopo passo è possibile trovare all'interno della documentazione la soluzione nella maggioranza dei casi. Bisogna infine evidenziare come la documentazione, essendo in lingua francese, a volte può essere fraintesa nella sua traduzione "scolastica" della lingua scritta.

Code-Aster offre una gamma completa di analisi multi-fisica, metodi che spaziano in tutti i campi della fisica e dell'ingegneria; nel seguito sono rappresentati in maniera sintetica i campi di interesse del software in oggetto (*fonte www.code-aster.org*):

■ Phenomena

> Mechanical

- Static, quasi-static, linear or otherwise
- Dynamic, linear or otherwise, on a physical or modal basis
- Fracture, damage and fatigue
- Soil-Structure, Fluid-Structure and Soil-Fluid-Structure interactions

> Thermal

- Stationary, transient, linear or otherwise
- Fixed or moving reference coordinate system

> Associated phenomena

- Acoustics
- Metallurgy
- Hydration and drying

■ Analysis Types

> Standard

> Decomposition into Fourier modes

> Substructuring

> Model superposing, multiscale

> Adaptive mesh

> Sensitivity calculation

> Fitting and optimization

> Mechanical reliability calculation

■ Multiphysical

> Internal links with thermics

- Hydration, drying
- Metallurgy

> Internal links with mechanics

- Thermal
- Metallurgy
- Hydration and drying

> Internal couplings

- Thermo-hydro mechanical
- Fluid-Structure

> External coupling with other codes

- Soil-(Fluid)-Structure (MISS3D)

> External links with other codes

- Hydraulics: Code_Saturne
- Thermal: SYRTHES
- Electromagnetism: Flux2D/3D
- Fast dynamics: EUROPLEXUS

■ Loadings

> Mechanical

- Nodal or distributed forces
- Pressure
- Inertia loading
- Centrifugal acceleration
- Imposed movements
- Anelastic strain
- Effect of wind

> Thermal

- Temperature
- Flows, linear or otherwise
- Forced convection
- Exchange between walls
- Heating by Joule effect

> Specific loads (following forces, electromagnetic forces, initial states)

■ Nonlinearities in static and dynamic

> Geometrics

- Geometric updating, large displacements, large rotations
- Following forces
- Continuation methods: in displacement, by arc length, in strain, by criterion output
- Load discharge and non-radiality indicators
- Contact and friction: by a discrete contact method (active stresses, penalization, conjugated projected gradient or by an augmented Lagrangian method.)
- First order buckling

> Materials (95 constitutive laws)

- Linear and nonlinear elasticity
- Nonlinear hyperelasticity
- Local elastoplasticity and elastoplasticity with gradient formulation
- Nonlinear viscoelasticity
- Local and with gradient formulation damage
- Elastoviscoplasticity
- Metallurgical effects
- Material data dependent on temperature, metallurgical condition, hydration, drying and fluence
- Progressive strain
- Hydration, shrinkage and creep of concrete
- Geomaterials

■ Dynamics

> Modal analysis

- With or without damping (viscous, hysteretic, modal)
- Direct or by substructuring
- Normalization, filtering, modal parameters

> Linear transient response

- Direct
- On modal basis
- By substructuring

> Transient response with local nonlinearities (on modal basis)

- Shocks
- Friction
- Fluid blade

> Harmonic response

- Direct
- On modal basis
- By substructuring

> Random response

- Parametric and nonparametric probabilistic
- Stochastics

> Direct nonlinear analysis

- Implicit
- Explicit
- Shocks
- Plasticity, damage
- Contact and friction

> Substructuring

- Conventional or cyclic
- Modal, transient or harmonic analysis

> Seismic analysis

- With shocks or multi-support
- Spectral or transient direct linear or on modal basis
- Modal damping calculation (RCC-G)

> Extrapolation of experimental measurements

- Temporal or frequential

■ Interactions

> Fluid-Structure

- Structure-incompressible flow interaction; turbulent stresses
- Vibro-acoustics (free surface)

> Soil-Structure and Soil-Fluid-Structure

- Absorbent boundary elements
- Frequential coupling with MISS3D

■ Thermal analysis

- > Linear and nonlinear thermics
 - Phase change
 - Hydration and drying
 - Mobile coordinate system resolution
- > Metallurgical changes
 - Steels, Zircaloy
 - Phase hardness calculation
- > Thermal treatments and welding

■ Geotechnical Civil Engineering

- > Constitutive laws for concrete (reinforced or pre-stressed), geomaterials
- > Hydration, drying and basic creep at different time scales
- > Passive reinforcement or pre-stressing effect with elastoplastic behaviour: bar, grids and membrane
- > Creep-cracking coupling
- > Thermohydromechanics (porous media, formulation in effective stresses, constitutive laws in kit form, etc.)
- > Specific loads (hydraulic and gaseous flows)
- > Excavation procedure
- > Fracture, damage, fatigue and collapse of structures
- > Global release rate
 - thermoelasticity: G
 - thermoelastoplasticity: GP and GTP
- > Local release rate in 3D
- > Stress intensity factors
- > Models of brittle and ductile fracture, initiation and instability
- > Specific load drive
- > Local and non-local damage
- > Crack modelling: joint element and X-FEM
- > Decoupled damage

- > Fatigue analysis
 - Loading history
 - Counting methods
 - Specific criteria applications
- > Verification of RCC-M criteria
- > Zarka-Casier method in cyclic loading
- > Progressive wear
- > Limit analysis
- > Micro-macro approach: polycrystalline model

■ Survey quality

- > Spatial error indicators
 - Mécanique (en résidu pur, par lissage)
 - Thermique (en résidu pur)
- > Mesh refinement/unrefinement via HOMARD
- > More robust finite elements
 - Mechanics (underintegrated, incompressible)
 - Thermics (lumped modelling)
- > Mesh diagnosis
- > Thermomechanical time step redistribution
- > Sensitivity calculations
 - Mechanical
 - Thermal
 - Depending on materials, loadings and domain variations
- > Parameter fitting
 - Materials or loading
 - Depending on test sampling or digital results.
- > Reliability calculations: probability of exceeding threshold using a FORM type method

■ Modelling

- > Catalogue of material data
- > Connecting incompatible meshes.
- > Superimposing models using the Arlequin method.
- > Modelling connection (3D-shell, beam-pipe, etc.)
- > Plane stress condition adaptable to all models
- > Beam characteristics calculation
- > Homogenization (composites, repetitiveness, etc.)

■ Elements library (400 finite elements)

- > Mechanical
 - 2D, 2D axi (with or without Fourier decomposition), 3D, under-integrated, incompressible
 - Bars, beams (simple or multi-fibre), pipes, plates, shells, membranes, cables, discrete or non-distortable elements
- > Thermics: 2D, 2D axi (with or without Fourier decomposition), 3D, shells
- > Hydration-Drying: 2D, 2D axi, 3D
- > THM Coupling: 2D, 2D axi, 3D

■ Solvers

- > Linear (LDLT, multifrontal, PCG, MUMPS, FETI)
- > Nonlinear (Newton, etc.)
- > Integration schemes (Runge-Kutta, Newmark, adaptatives, etc.)
- > Modals (Power, Lanczos, IRAM, etc.)
- > Extended parameter setting. Several strategies for re-numbering, storage, pre-conditioning, post-verification, etc.

■ Dedicated tools

- > Pipework stacks and elbows ASPIC/ASCOUF
- > Steam Generator Tubes: GEVIBUS

■ Software environment

- > Integration in Salomé
- > Pre- / post processing: I-DEASTM, GIBI, Gmsh, ENSIGHT, Xmgrace
- > Data exchange in MED format
- > Survey management tool and engineering application workshop: ASTK
- > Command File Editor and Syntax Analyser: EFICAS
- > Portable nature of bases
- > Developed command language: PYTHON
 - Loop, test, checking structures, etc.
 - Method, class, etc.
 - Interactive calculation and visualization (mathematic libraries, GUI, scale drawings, etc.)

■ Not forgetting...

- > Software under QA (independent validations, reference of 2,000 test cases, 13,000 pages of documentation, source management, qualification of version, etc.).
- > code-aster.org website (downloads, online documentation, forum, FAQ, examples, etc.)
- > Communication and network (quarterly ASTER 'echos' magazine; User Club, its network of correspondents and the annual day; Free Code_Aster under GPL licence, etc.)

3.5 Il modulo di elaborazione grafica dei dati “ParaVis”

In questo capitolo descriveremo brevemente alcune delle funzioni che sono possibili sfruttando le caratteristiche di *ParaVis*, un modulo all’interno del pacchetto *Salomè-Meca*, il quale risulterà uguale al suo “fratello maggiore” *ParaView* (versione del software più completa e potente), la differenza si ritrova in alcuni piccoli dettagli, ma che in ogni caso il software *Code-Aster* e *Salomè-Meca* non trattano (come ad esempio alcune opzioni legate ai software GIS e QGIS).

ParaView è una multiplatforma *Open-source* progettata per visualizzare insiemi di dati di varie dimensioni. Gli obiettivi del software comprendono il supporto e lo sviluppo di modelli computazionali distribuiti per elaborare set di dati di grandi dimensioni. Ha un'interfaccia utente aperta, flessibile e intuitiva che consente una rielaborazione dei dati sotto diverse forme di rappresentazione fisica, che varia dalle più conosciute (ad es. visualizzazione degli spostamenti o degli sforzi) a quelle più ricercate (interazioni acustiche o termiche).

ParaView gira su sistemi di memoria paralleli distribuiti e condivisi e su sistemi a singolo processore ed è stato testato con successo su Windows, Linux e Mac OS X.

Gli utenti che utilizzano *ParaView* (e quindi anche *ParaVis*) possono creare rapidamente visualizzazioni per analizzare i propri dati impiegando tecniche qualitative e quantitative.

La base di codice *ParaView* è concepita in modo tale che tutti i suoi componenti possano essere riadoperati per sviluppare rapidamente applicazioni verticali. Questa adattabilità consente agli sviluppatori *ParaView* di sviluppare rapidamente applicazioni con funzionalità specifiche per uno specifico dominio problematico.

Sotto il cofano, *ParaView* utilizza il *Visualization Toolkit* (VTK) come motore di elaborazione e rendering dei dati. Gli obiettivi del team *ParaView* includono quanto segue (www.paraview.org):

- Sviluppare un'applicazione di visualizzazione multi-piattaforma *Open-source*;
- Supportare modelli di calcolo distribuiti per elaborare set di dati di grandi dimensioni;
- Creare un'interfaccia utente aperta, flessibile e intuitiva;
- Sviluppare un'architettura estensibile basata su standard aperti.

ParaView è un ottimo strumento per visualizzare la *Mesh* di input e i risultati dei solutori basati sul metodo degli elementi finiti (FEM), e sarà esattamente ciò che utilizzeremo noi tramite *Code-Aster*.

Questi risolutori sono utilizzati in molti domini ma particolarmente spesso nei settori dell'ingegneria meccanica, civile e automobilistica. “*Element*” e “*Nodes*” in FEM si associano in modo abbastanza naturale a “celle” e “Punti” nella terminologia VTK, e una volta trovato il campo specifico da visualizzare, *ParaView* (o *ParaVis*) rappresenterà tutte le *Mesh* inserite, troverà e evidenzierà le cellule con caratteristiche specifiche, deformerà le *Mesh* in base alle grandezze vettoriali derivate e mapperà i valori dei risultati sui nodi e sugli elementi attraverso tabelle di colori specifiche per il suddetto elemento considerato.

4 Il metodo degli Elementi Finiti (FEM)

4.1 Gli aspetti generali del metodo FEM

Il metodo degli Elementi Finiti (FEM), o molte volte semplicemente ricordato come il metodo di Analisi agli Elementi Finiti (FEA), è un metodo di calcolo computazionale che viene utilizzato per ricercare delle soluzioni approssimate in grado di risolvere problemi ingegneristici e non solo.

L'idea di partenza di questo metodo è quella di ricercare la soluzione di un problema considerato complicato utilizzando un metodo di analisi che ci porti ad una soluzione facile ed accettabile, trovando in questo modo una soluzione che sarà non esatta ma approssimata in maniera ottimale da renderla molto simile alla soluzione esatta.

4.1.1 Sistemi ad un grado di libertà

Se analizziamo il comportamento di un carrello (rappresentante un punto materiale) soggetto ad una forza esterna F ed alla forza elastica di richiamo di una molla lineare avente rigidità k , considerando le leggi di proporzionalità tra elongazione della molla e forza si può arrivare all'equazione di equilibrio statico (dal libro “*Scienza delle costruzioni vol.2*”, cap.11.1, A. Carpinteri).

$$F - kx = 0 \quad (3.1)$$

ed alla posizione di equilibrio che ne consegue

$$x = F/k \quad (3.2)$$

Tale risultato è ottenibile sfruttando anche l'*energia potenziale* del sistema

$$W(x) = \frac{kx^2}{2} - Fx \quad (3.3)$$

Sfruttando le relazioni di meccanica razionale, derivando e ponendo la derivata a zero troviamo la condizione per cui il sistema risulta essere in equilibrio questo risulta essere anche il punto minimo di energia.

$$\text{forza totale} = -\frac{dW}{dx} = F - kx = 0 \quad (3.4)$$

Tale posizione è ottenibile sfruttando anche il principio dei lavori virtuali, infatti imponendo uno spostamento infinitesimo Δx il lavoro finale, nella condizione di stazionarietà sarà nullo.

Con queste premesse si può introdurre il metodo degli elementi finiti, rivolto a strutture a più gradi di libertà; le due vie indicate possono essere le medesime per il sistema ad un grado di libertà appena introdotto:

- Principio di minimo dell'energia potenziale totale;
- Principio dei lavori virtuali;

Il metodo agli elementi finiti vuole essere un metodo di discretizzazione considerando invece della funzione di spostamento $\{\eta\}$, gli spostamenti stessi $\{\delta\}$ come incognite per un certo numero di nodi.

Una volta trovati gli spostamenti questi vengono interpolati tramite funzioni sufficientemente regolari (funzioni di forma). Il problema dunque risulta essere la determinazione della posizione di equilibrio $\{\delta\}$, da cui per interpolazione si ricavano gli spostamenti $\{\eta\}$, per derivazione si ricavano le deformazioni $\{\epsilon\}$ ovvero $\{q\}$, e quindi tramite le equazioni costitutive le tensioni $\{\sigma\}$ ovvero le caratteristiche statiche $\{Q\}$.

Le grandezze principali saranno i *gradi di libertà* g (grandezza del vettore spostamento $\{\eta\}$), la dimensione d dei *vettori delle caratteristiche deformative* $\{\epsilon\}$ ovvero $\{q\}$, e statiche $\{\sigma\}$ ovvero $\{Q\}$.

In tabella sono rappresentati i numeri caratteristici per tutti i solidi elastici unidimensionali e tridimensionali sinora analizzati.

Tabella 3.1. g e d per i solidi elastici (dal libro “*scienza delle costruzioni vol.2*”, A.Carpinteri).

<i>Tipologia</i>	<i>g</i>	<i>d</i>
Trave nel piano	3	6
Trave nello spazio	3	6
Lastra piana inflessa	3	5
Lastra a doppia curvatura	5	8
Solido bidimensionale	2	3
Solido tridimensionale	3	6

4.1.2 Sistemi a più gradi di libertà

Come già descritto nel precedente capitolo si possono estendere le formulazioni inerenti al metodo agli elementi finiti inerenti ai sistemi ad un grado di libertà, a sistemi a più gradi di libertà (dal libro “*Scienza delle costruzioni vol.2*”, cap.11, A. Carpinteri).

Su queste basi si deve procedere definendo il tipo di elemento strutturale (trave, lastra, ...) che si vuole descrivere, e di questo elemento avremo un numero di nodi che lo definiscono.

Definiti gli spostamenti $\{\eta\}$ (per ogni n gdl dell'elemento), le forze nodali F (per ogni n gdl dell'elemento) e gli spostamenti interni $\mathbf{u}(P)$ (per ogni k spostamenti interni definiti) si valuteranno le funzioni interne con cui descrivere il campo interno di

spostamenti (mediante la loro combinazione). Si precisa che la combinazione delle funzioni scelte, in genere polinomiali, forniranno una soluzione approssimata e non esatta.

Identificate le funzioni interpolanti $\varphi_i(P)$ per ogni punto interno dell'elemento si può scrivere:

$$\begin{Bmatrix} u_1(P) \\ \vdots \\ u_k(P) \\ \vdots \end{Bmatrix} = \begin{Bmatrix} \alpha_1 \varphi_{11}(P) + \alpha_2 \varphi_{12}(P) + \cdots + \alpha_n \varphi_{1n}(P) \\ \vdots \\ \alpha_1 \varphi_{k1}(P) + \alpha_2 \varphi_{k2}(P) + \cdots + \alpha_n \varphi_{kn}(P) \\ \vdots \end{Bmatrix} \quad (3.5)$$

Ora si applicano le interpolazioni sui nodi considerando tutti i gdl definiti

$$\begin{Bmatrix} \vdots \\ f_1 \\ \vdots \end{Bmatrix} = \begin{Bmatrix} \vdots \\ \alpha_1 \varphi_{k1}(P_i) + \alpha_2 \varphi_{k2}(P_i) + \cdots + \alpha_n \varphi_{kn}(P_i) \\ \vdots \end{Bmatrix} \quad (3.6)$$

Una volta note le posizioni dei nodi nel sistema di riferimento locale si può osservare che **A** è una matrice nota e definita.

Facendo coincidere i coefficienti α con i n gdl di ogni elemento, **A** risulterà quadrata, quindi si ricaverà:

$$\{u(P)\} = [\varphi(P)][A]^{-1}\{f\} \quad (3.7)$$

$$[N(P)] = [\varphi(P)][A]^{-1} = \textbf{funzioni di forma} \quad (3.8)$$

Num. Righe $N \rightarrow$ numero di spostamenti interni definiti

Nume.colonne $N \rightarrow$ numero di gdl dell'elemento

Si può osservare innanzitutto che $N(P)$ non è quadrata; le funzioni di forma possono far descrivere una comb. Lineare nella quale i coefficienti sono gli spostamenti nodali **f**.

Note le funzioni di forma si possono ricavare il campo delle deformazioni interne tramite le relazioni differenziali:

$$\{\varepsilon(P)\} = Diff [u(P)] \quad (3.9)$$

Ora è possibile esplicitare le tensioni tramite le equazioni costitutive utilizzando la matrice di rigidezza del materiale **D**

$$\{\sigma(P)\} = [D] \{\varepsilon(P)\} \quad (3.10)$$

A questo punto è possibile applicare il *principio dei lavori virtuali* per la determinazione di **K**, matrice di rigidezza.

Per le formule appena esplicitate era richiesta la continuità sul confine del dominio, ma parlando ora in termini di deformazioni e tensioni, la continuità non sempre sarà garantita.

L'equilibrio quindi sarà valutato in senso globale, attraverso un funzionale determinato dal lavoro virtuale.

Principio dei Lavori Virtuali: Se una struttura, soggetta a vincoli, si trova in equilibrio, allora per ogni spostamento virtuale compatibile con i vincoli, il lavoro delle forze esterne è uguale all'energia immagazzinata dal campo di tensioni interno, per il tramite delle deformazioni virtuali.

In questo senso si esplicitano le grandezze virtuali (indicate con *) $\{f^*\}$; $\{\varepsilon^*\}$:

$$W_{est} = \{f^*\}^T \cdot \{F\} = \text{forma quadratica} \quad (3.11)$$

$$dW_{int}(P) = \{\varepsilon^*(P)\}^T \cdot \{\sigma(P)\} dV = \text{forma quadratica} \quad (3.12)$$

Integrando sul volume dell'elemento, ricordando le posizioni precedenti e portando fuori dall'integrale le grandezze discrete si ottiene:

$$W_{int} = \{f^*\}^T \int_V [\mathbf{B}]^T \cdot [\mathbf{D}][\mathbf{B}] dV \{f\} \quad (3.13)$$

Uguagliando lavoro interno con il lavoro esterno si eliminano i valori virtuali, in virtù delle compatibilità coi vincoli:

$$F = \int_V [\mathbf{B}]^T \cdot [\mathbf{D}][\mathbf{B}] dV \{f\} \quad (3.14)$$

$$K = \int_V [\mathbf{B}]^T \cdot [\mathbf{D}][\mathbf{B}] dV \quad (3.15)$$

Dove K risulterà un valore approssimato; Per il calcolo occorre integrare le matrici su tutto l'elemento.

Infine applicando la stessa procedura per elementi caricati con forze di volume

$$\int_V [\mathbf{B}]^T \cdot [\mathbf{D}][\mathbf{B}] dV \{f\} = \{F\} + \{F_{vol}\} + \{F_{supl}\} - \{F_0\} \quad (3.16)$$

In seguito è possibile ricavare la matrice delle masse inserendo la densità di volume ottenendo:

$$[\mathbf{M}] = \int_V [\mathbf{N}]^T \rho(P) [\mathbf{N}] dV \quad (3.17)$$

A calcolo avvenuto è possibile allo stesso modo ricavare tensioni e deformazioni con le medesime relazioni già esplicitate in precedenza.

4.2 Esempio pratico di analisi all'interno del software FEM

Come si può osservare ciò che verrà fatto in seguito si riferisce alla semplice trattazione di un caso pratico generico per mostrare in maniera più generale e a livello grafico il procedimento che avviene all'interno del software di modellazione *Code-Aster*.

Avendo a disposizione una funzione (che rappresenta il nostro elemento monodimensionale, o bidimensionale,...) si deve cercare un metodo per approssimare la funzione e trattare con gli elementi che descrivono il comportamento dei nodi di discretizzazione.

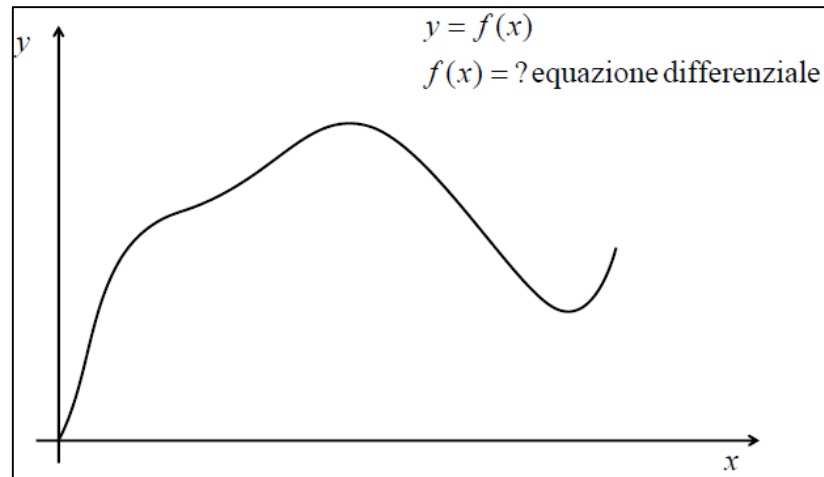


Figura 3. 1 funzione di esempio generica

A questo punto si ricerca un metodo per descrivere al meglio il comportamento della funzione, e dunque posso immaginare che la mia incognita venga ad essere h_i (aliquota dell'asse y) in riferimento al punto appartenente alla funzione sul piano xy .

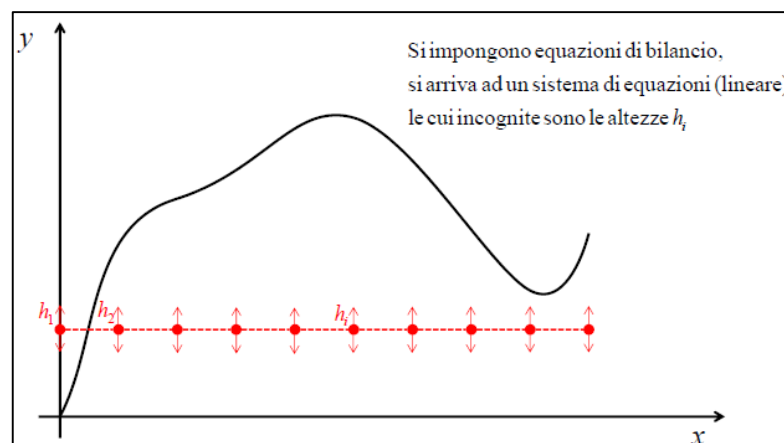


Figura 3. 2 Punti di riferimento

Una volta trovato il metodo con cui si andrà a rappresentare la funzione, si pongono le altezze dall'asse delle X come valori che, coincidendo con la funzione, descrivono in maniera approssimata (ma sufficiente) il comportamento della funzione stessa.

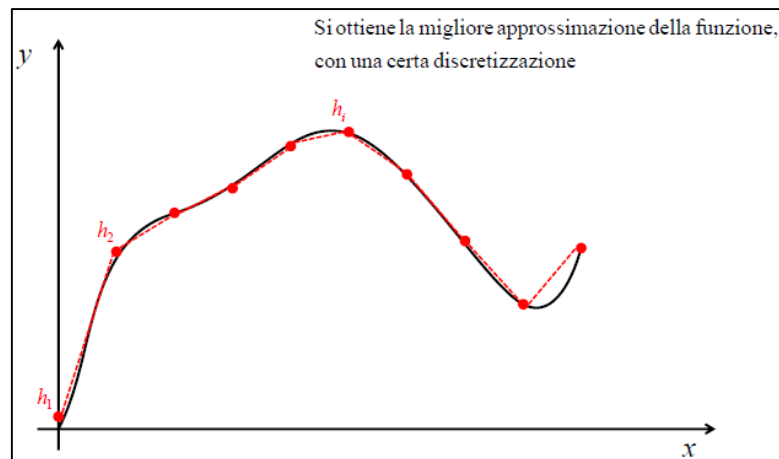


Figura 3.3 Funzione discretizzata

Nonostante l'esistenza di diversi strumenti matematici, anche di accuratezza elevata, questi non sono in grado di arrivare ad una soluzione completamente esatta, per questa ragione in assenza di altri strumenti di comprovata validità che ci consentano di giungere alla soluzione esatta, dobbiamo riferirci ad una soluzione approssimata, che in ogni caso sarà una soluzione accettabile per i problemi ingegneristici.

In ogni caso il punto di forza di questo metodo è quello di poter affinare in maniera sempre più accurata la soluzione aumentando il grado computazionale del problema.

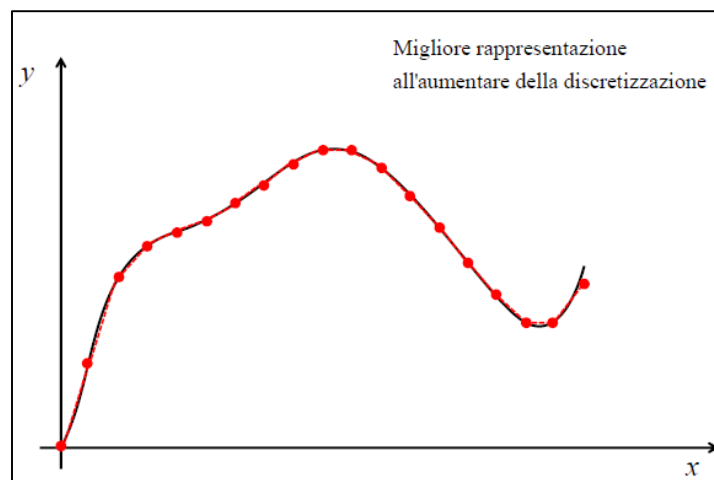


Figura 3.2 Funzione discretizzata con maggiore accuratezza

4.3 Modellazione della geometria e "Meshing"

Le strutture nell'ingegneria civile possono avere una geometria talvolta complessa e di difficile rappresentazione; per poter eseguire i calcoli per mezzo della teoria degli elementi finiti la struttura viene discretizzata in elementi e nodi che possano descrivere fedelmente il comportamento reale della struttura stessa.

Le strutture curve, o che presentano irregolarità, possono essere modellate utilizzando curve e superfici curve; tuttavia come si può osservare nella grande maggioranza dei casi, queste particolarità sono rappresentate in maniera approssimativa con linee rette e superfici piatte, se si utilizzano elementi lineari.

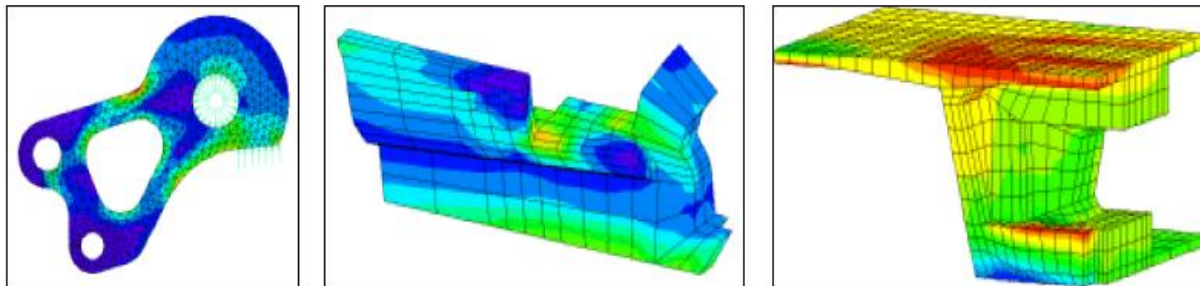


Figura 3.3 Tipologie di Mesh

A seconda del software che si utilizza nella modellazione progettuale, vi sono molti strumenti che consentono di creare una geometria del modello, ovvero una *Mesh*; i punti sono creati attraverso la digitazione di coordinate, le linee e le curve possono essere connesse sfruttando la presenza di punti o nodi. Le superfici inoltre possono essere create tramite la rotazione o la modellazione delle curve esistenti, mentre i solidi possono essere creati collegando, ruotando o modellando le superfici esistenti. In generale le interfacce grafiche aiutano gli utenti nella creazione di questo ambiente virtuale di progettazione geometrica, utilizzando numerosi pacchetti software di assistenza che si basano su protocolli di file CAD, notoriamente utilizzati nella progettazione tecnica. Nella maggior parte dei casi questi file possono essere letti da moduli o altri software, i quali sfruttando la modellazione al CAD ed impartendo un determinato percorso computazionale riescono a produrre in maniera autonoma una *Mesh* in pochi semplici passi.

Tuttavia molte volte possono esserci irregolarità o piccole accortezze da valutare nel file CAD, e queste possono creare problemi nella creazione della *Mesh* (“*error meshing*”), per questa ragione si deve sempre osservare, ed in certi casi, correggere la geometria ancora prima di lanciare il calcolo della discretizzazione computazionale del nostro modello geometrico.

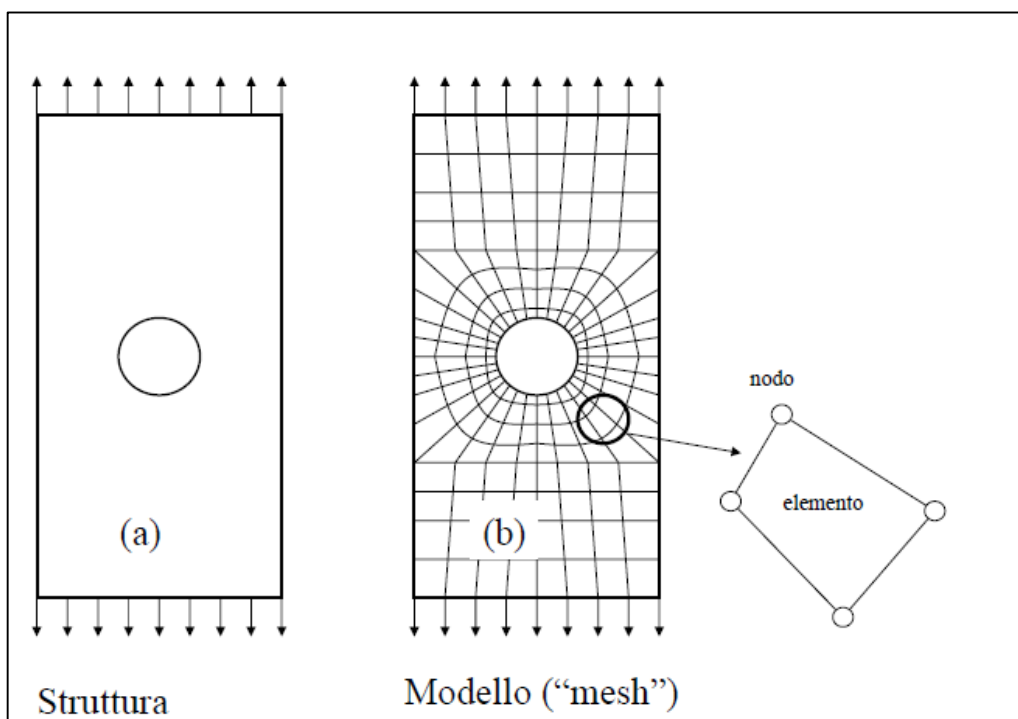


Figura 3.4 Elemento appartenente alla Mesh

La procedura ideale si avrebbe con un generatore di *Mesh* automatizzato, ma sfortunatamente i software informatici pongono dei limiti su questo argomento; al momento i software in commercio presentano un pre-processore semi-automatizzato che crea la *Mesh* con poche indicazioni dall'utente; inoltre vi sono *tools* e pacchetti progettati essenzialmente per la creazione della *Mesh*, generando di conseguenza file in diversi formati che possono essere letti da software di modellazione e simulazione di vari autori.

4.4 Funzioni di forma e procedimento del software

La *Mesh* viene eseguita per discretizzare l'elemento geometrico al fine di poter creare piccoli pezzi chiamati elementi o celle; In un mezzo continuo di qualsivoglia dimensione, cioè in un corpo o in una regione dello spazio in cui abbia luogo un particolare fenomeno, la variabile di campo, come la pressione, lo spostamento, la temperatura, la velocità o la densità, è funzione di un generico punto del dominio di definizione. Di conseguenza il problema presenta un numero infinito di incognite. La procedura di discretizzazione agli elementi finiti lo riduce ad un problema con un numero finito di incognite, suddividendo il dominio in elementi finiti ed esprimendo il campo incognito in termini di funzioni approssimanti, definite all'interno di ogni elemento.

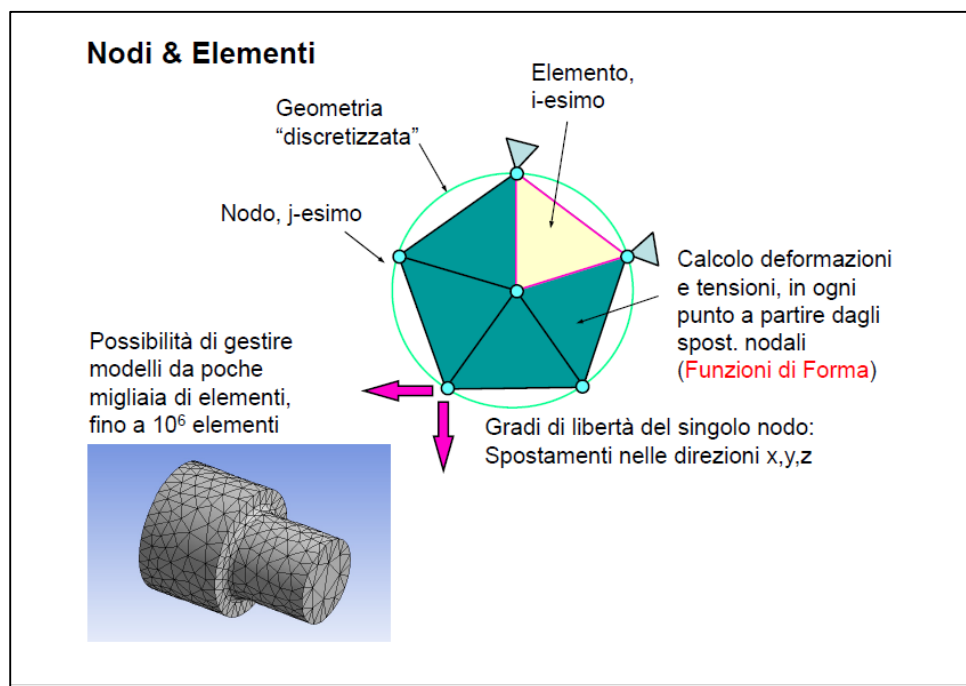


Figura 3.5 Metodo di discretizzazione

Le funzioni approssimanti, chiamate anche *funzioni di forma*, vengono individuate mediante i valori che la variabile dipendente assume in punti specifici detti nodi. I nodi sono posti di solito sul contorno degli elementi, in punti comuni a due o più elementi. Oltre ai nodi sul contorno un elemento può presentare dei nodi al suo interno. I valori che la variabile di campo assume sui nodi, ne definiscono univocamente l'andamento all'interno dell'elemento. Nella rappresentazione agli elementi finiti di un problema, i valori nodali della variabile di campo rappresentano le nuove incognite. Come anticipato,

l'idea base dell'approssimazione usata nel metodo agli elementi finiti è quella di approssimare il vero andamento della funzione incognita con quello di alcune funzioni particolari ad andamento noto: generalmente polinomiali, ma anche trigonometriche ed esponenziali. Vengono presi in considerazione un numero limitato di punti (chiamati anche *nodi*) interni al dominio di integrazione, per i quali i valori della funzione approssimata risulteranno identici a quelli della funzione approssimante. La scelta di base delle funzioni di forma influenza la fase di suddivisione in elementi finiti e viceversa, infatti l'utilizzo di funzioni di forma lineari modella l'andamento degli spostamenti all'interno dei singoli elementi finiti attraverso funzioni lineari; l'utilizzo di funzioni di forma lineari richiederà pertanto un infittimento della discretizzazione nelle zone di analisi in cui si possa prevedere un valore elevato del gradiente degli sforzi.

4.5 *Proprietà del materiale*

Le proprietà dei materiali possono essere definite per un singolo elemento o per un gruppo di elementi, se necessario. In genere per diversi fenomeni da valutare saranno richiesti le specifiche dei materiali di tutti gli elementi considerati, in questo modo si vuole conoscere il comportamento specifico di ogni singola particella di materiale. Per fare un esempio concreto il modulo di elasticità ed il modulo del taglio possono essere richiesti per la valutazione delle caratteristiche di sollecitazione di strutture e solidi in genere, oppure il coefficiente di dilatazione termica può essere richiesto come parametro nella valutazione dei problemi di scambio termico; l'inserimento di questi moduli e coefficienti in fase di *pre-processing* è solitamente semplice tramite la semplice immissione dei dati in apposite caselle, e questo avviene per ogni singolo software di elaborazione agli elementi finiti, per architetture più complesse di modellazione dei materiali possono essere sostituite leggi, caratteristiche o altre proprietà del materiale o dei materiali coinvolti indicando in maniera dettagliata a quale porzione dell'oggetto geometrico corrisponde la legge o il parametro inserito.

4.6 *Condizioni al contorno, di carico e di stato*

Queste specifiche condizioni giocano un ruolo chiave nell'elaborazione di simulazione al fine di conoscere le caratteristiche stesse che identificano il problema e, di conseguenza, poter arrivare ad una soluzione del problema medesimo.

L'inserimento di queste condizioni viene effettuato generalmente seguendo le linee guida dell'interfaccia grafica del programma utilizzato, il quale è spesso interfacciato con la grafica. Gli utenti utilizzatori possono specificare le condizioni di carico, le caratteristiche geometriche (punti, linee o curve, superfici o solidi) tramite inserimento di elementi o tramite punti in griglia di inserimento, ancora una volta tuttavia occorre ricordare che queste caratteristiche di creazione di elementi geometrici all'interno dell'interfaccia grafica devono essere inseriti correttamente e questo processo apparentemente semplice richiede abilità ed esperienza al fine di non sbagliare la modellazione, anche in questo percorso le ipotesi e le condizioni variano a seconda del tipo di problema da analizzare.

5 *Code-Aster: guida alla lettura dei manuali*

5.1 *La documentazione di Code-Aster*

All'interno del presente capitolo si parlerà del software *Code-Aster*, e più nel dettaglio, di tutto quello che dovremmo affrontare per eseguire le analisi dei casi studio che vorremmo affrontare. Come si può capire il software risulta essere molto complicato per chi non ha molta dimestichezza col i codici ad elementi finiti, tuttavia grande quantità di documentazione reperibile su sito www.code-aster.org Ci consentirà di avere informazioni dettagliate sulle procedure da seguire per la corretta compilazione. E' comunque necessario ricordare che bisogna necessariamente fissare un obiettivo, e dopo aver capito quali output vogliamo dal programma, creare un linguaggio di codice che istruisca il programma in maniera opportuna, senza andare fuori tema, e rispettando le regole di compilazione del codice.

Il software *Code-Aster* come già ampiamente discusso è un software agli elementi finiti per la modellazione e la risoluzione di strutture semplici e/o complesse; come già detto il software in questione non è un semplice software con licenza GPL, ma essendo un software con caratteristiche *Open-Source*, risulta essere molto più articolato e difficile da comprendere. Per tali ragioni le versioni che vengono messe a disposizione dall' EDF sono versioni che vengono sviluppate sempre in ambito interno, oppure anche in ambito internazionale tramite lo scambio di informazioni di alcuni gruppi di esperti che si adoperano per risolvere problemi di natura reale commissionati da enti privati (o dalla stessa EDF), e dopo stringenti controlli vengono rilasciate le versioni aggiornate; inoltre queste versioni vengono sottoposte ad ulteriori test, e fintanto che i test non sono conclusi non possono considerarsi versioni di software stabili. Noi ovviamente partiremo da una versione stabile ovvero: ***Code-Aster v3.14 Stable***.

Una volta scaricato tutto il pacchetto di ***Salomè-Meca 8.3.0*** ci si baserà su questa interfaccia GUI per accedere ai vari livelli di modellazione fino a confluire nella risoluzione FEM in *Code-Aster*.

Occorre preventivare che per l'approccio al software sono necessarie ore di pratica e molta pazienza, ma nonostante la documentazione reperibile sul sito internet si affianca a documentazione in cui vengono spiegate le caratteristiche di analisi, le caratteristiche di modellizzazione e vengono inoltre affiancati alcuni esempi semplici per ogni singolo tema di analisi che aiuterà l'utente a sviluppare correttamente il proprio progetto.

Per comprendere meglio questo software possiamo cominciare a dire che i manuali classificati come **U**, **R** e **V**, sono quelli più largamente utilizzati per le operazioni di base e per la comprensione del software, mentre gli altri due manuali sono largamente indirizzati per sviluppatori del codice sorgente ASCII e calcoli complessi, o per tutti gli utenti che vogliano anche solo conoscere le componenti informatiche profonde su cui si struttura il software. Verrà fatto riferimento quasi esclusivamente ai manuali più vicini all'utente base e a questi verranno affiancati alcuni scritti e manuali trovati in rete ed in biblioteca che potranno essere utili allo sviluppo dei casi studio.

La documentazione è suddivisa in 5 Manuali (*Handbooks*) facilmente accessibili:

Tabella 4.1. *Suddivisione per macro-argomenti dei manuali a disposizione (dal sito code-aster.org)*

<i>Handbook</i>	<i>Acronimo di riferimento</i>	<i>Descrizione del contenuto</i>
<i>Utilisation</i>	U	Utilizzo dei controlli e delle strutture di dati
<i>Reference</i>	R	Formulazione di fenomeni modellati, metodi di analisi ed algoritmi numerici
<i>Descriptif informatique</i>	D	Strutture dei dati, algoritmi, architetture dati e ambiente
<i>Validation</i>	V	Test di valutazione di tutti i campi possibili di modellazione
<i>Administration</i>	A	Piani di qualità, procedure di sviluppo e risorse dei servizi offerti dalle versioni

5.2 Il fenomeno meccanico all'interno di Code-Aster

Code-Aster è un software che appropria principalmente meccanica dei solidi e strutturale, ma tutto questo avviene come supporto d'integrazione per la rappresentazione dell'ambiente di sfruttamento dei componenti meccanici; su questa prospettiva è stata presa la decisione di includere in *Code-Aster* le caratteristiche che consentono la modellizzazione dei fenomeni che frequentemente si associano anche ai fenomeni meccanici: ovvero i fenomeni termici e acustici.

Il fenomeno meccanico sarà il punto cardine che andremo a trattare; esso è modellato per raggiungere due obiettivi principali: la determinazione delle sollecitazioni interne, per poi risalire allo stato di tensioni in ogni punto di una struttura, sotto le molteplici combinazioni di carico richieste al programma.

La conoscenza dello stato di tensione all'interno del nostro elemento rende possibile il proseguo delle analisi all'interno del comportamento meccanico permettendoci di arrivare a poter appurare altri elementi nel nostro mezzo meccanico, in particolare è possibile:

- Verificare il rispetto delle regole di costruzione strutturale per ogni tipo di struttura, in particolare le regole di progettazione, di costruzione, ...;
- Analizzare la nocività delle imperfezioni e la loro possibile propagazione;
- Studiare il comportamento nel carico ciclico e analisi con fatica;
- Prevedere le analisi dei carichi di lavoro con l'evoluzione dello stato interno.

Inoltre all'interno di *Code-Aster* è possibile riferirsi alle più svariate combinazioni di carico possibile, e agli effetti prodotti: come la determinazione della configurazione deformata indotta da un carico permanente (statico) o derivante da un'evoluzione lenta (quasi-statica) o più veloce (dinamica) dei carichi o condizioni al contorno. E' inoltre possibile conoscere la configurazione deformata, la velocità e le accelerazioni

corrispondenti, consentendoci di continuare l'analisi del comportamento meccanico fino ad arrivare alla presa di coscienza dei seguenti effetti:

- comportamento vibratorio o acustico;
- trasmissione di effetti ad altre strutture o componenti;
- analisi delle interazioni con le strutture vicine per determinare il danno, gli effetti del danno o i parametri di usura che possono derivarne.

I livelli di modellizzazione che intervengono per lo studio di questo fenomeno sono: la possibilità di una rappresentazione della struttura partendo dalla forma geometrica, con diverse possibili modalità di rappresentazione in grado di rappresentare lo stato di analisi della struttura.

Tali modalità in grado di coesistere possono essere elencate nel seguito: mezzo continuo corrispondente a una geometria tridimensionale o bidimensionale con varie assunzioni (deformazioni piane, simmetriche o adattato alla decomposizione dei carichi nelle modalità di Fourier), elementi strutturali corrispondenti a un mezzo con strato medio, un mezzo con fibra media o un mezzo discretizzato (travi, piastre, scafi, tubi, ...). La rappresentazione del comportamento secondo le leggi dei materiali che compongono gli elementi studiati, a volte comprendenti anche più di una legge, in qualunque punto di una struttura; tramite le leggi individuate sarà possibile descrivere il comportamento dei materiali (ad esempio legge elastica lineare, visco-elastica lineare, plastica, ...).

I parametri che descrivono il comportamento possono essere variabili comunemente conosciute (come la temperatura), oppure meno conosciute ma adatte alla descrizione di analisi maggiormente specifiche e più avanzate (come l'essiccazione del calcestruzzo, il grado di idratazione, e molto altro).

5.3 Alcuni preziosi consigli prima di iniziare l'analisi

Prima di cominciare ad utilizzare il software, all'interno della piattaforma di *Salomè-Meca*, si introdurranno alcuni preziosi consigli che è bene ricordare soprattutto se non si ha familiarità con i software ad elementi finiti in generale:

- Si consiglia di inserire tutti i file relativi ad un singolo studio all'interno di una cartella che segua il medesimo percorso di file (in parole povere avere una sola *directory*), ricordandosi di avere i permessi di lettura, scrittura ed esecuzione della stessa;
- In questa *directory* non nominare nessun file o cartella con caratteri speciali, come “spazio” o “\$”, che potrebbero dare problemi nella lettura di altri file all'interno della stessa *directory*;
- Nel file sorgente di comando (.comm), qualunque sia il formato, ricordarsi di scrivere commenti o piccoli appunti che possano servire a ricordarsi tutte le operazioni che si vengono svolte, in modo tale che ogni volta che verrà letto il file con i vostri commenti (che non verranno ovviamente eseguiti dal software in quanto preceduti da “#”) saprete il perché e il come avete affrontato il problema e con quale metodo;
- Se si lavora tramite Gmsh tenere sotto controllo “Message Console” all'interno del quale verranno visualizzati eventuali messaggi di errore (nelle nostre analisi non è stata utilizzata questa applicazione);

- Eseguire un controllo sulle dimensioni e le unità di misura utilizzate nella creazione del CAD, oppure a seguito di una importazione del file dall'esterno, evita di avere disagi durante la modellazione;
- Eseguire un controllo sulla *Mesh* per eliminare nodi ed elementi doppi tramite i seguenti comandi (“*merge nodes*”, “*merge elements*”);
- Quando è possibile, effettuare confronti sui calcoli effettuati dal software e calcoli fatti con altri software, oppure “a mano” (ricordarsi che un piccolo errore sistematico dovuto a diversi fenomeni sarà sempre presente);
- Fare controlli sui risultati in output delle masse, per modelli generati o per gruppi di elementi;
- Controllare la coerenza tra la somma delle reazioni trovate con la somma delle reazioni applicate;
- *Code-aster* non si basa su un determinato sistema di unità di misura, ma si auto adatta al nostro sistema di misura utilizzato, pertanto è necessario adoperare le medesime unità di misura dall'inizio alla fine dell'analisi (*vedi tabella 4.2*).

Tabella 5.1 Unità di misura trattate

	<i>Grandezza fisica</i>	<i>Dimensione</i>	<i>mm.t.s</i>	<i>SI</i>
Unità di base	Lunghezza	L	mm	m
	Massa	M	T	kg
	Tempo	T	s	s
	Temperatura	°	K	K
Unità derivate	Angolo	l	Rad	Rad
	Frequenza	T ⁻¹	Hz	Hz
	Forza	MLT ⁻²	N	N
	Pressione/Sforzo	ML ⁻¹ T ⁻²	N/mm ²	N/m ²
	Densità di massa	ML ⁻³	t/mm ³	kg/m ³
Esempio	Densità di massa	ML ⁻³	7.8E-9	7800
	Accelerazione	LT ⁻²	9810	9.81
	Modulo di Young	ML ⁻¹ T ⁻²	210000	2.1E11

Quando si effettua l'analisi, ovvero si esegue il *Run* del programma, l'esito positivo/negativo è determinato da un segnale di colore rosso, arancione o verde. Ovviamente si intuisce cosa significhino i segnali, tuttavia è necessario ricordare che con l'output viene prodotto un file di messaggio (*.mess*) prodotto come output all'interno del quale sono spiegate dettagliatamente le operazioni di calcolo e, nell'eventualità, gli errori. La lettura del file è abbastanza laboriosa, ma è necessaria perché attraverso la lettura si può stabilire il punto esatto in cui la compilazione si è interrotta ed eventualmente intervenire per apportare la correzione. Di conseguenza si consiglia di leggere attentamente tutti gli output del file.

- Se vi sono messaggi di errore, ci sono anche suggerimenti per la risoluzione del problema;
- Se otteniamo un risultato, la spiegazione del perché possa essere privo di significato spesso viene spiegato al suo interno;
- In generale, controllare tutte le linee dei messaggi di avviso che iniziano con <A>;
- Non si deve dare per scontato le didascalie colorate ed i range di valori associati alle barre scalari prodotte dallo strumento di post-elaborazione (ParaVis) in quanto possono essere un buon strumento di verifica dei risultati;
- E' inoltre sempre bene che i file *.mess*, *.resu*, *.comm*, *.med*, che stiamo leggendo o verificando appartengano alla stessa analisi; prima di incolpare il software, provare a scoprire se vi siano errori latenti. Nel caso vi sia il sospetto di qualche anomalia del software segnalarlo nel forum (<https://www.code-aster.org/forum2/>) di supporto;
- Infine utilizzare sempre la documentazione fornita dal sito ufficiale con discrezione, infatti tale documentazione è stata ottenuta dal traduttore automatico (ad esempio Google translator) e per questa ragione vi possono essere delle incoerenze, degli errori o delle dimenticanze all'interno dei documenti tradotti (dal Francese all'Inglese).
- Si consiglia all'interno dell'ambiente *Geometry* di inspessire le linee per avere una maggiore chiarezza degli elementi strutturali.
- Si consiglia di definire minuziosamente la geometria, la mesh ed i gruppi prima di aprire Asterstudy, infatti lo zapping tra un modulo e l'altro può causare errori imprevisti.
- Si consiglia di definire i carichi, i vincoli e i file di output separatamente, quando possibile, per non sovraccaricare l'analisi di calcolo.

5.4 Il manuale Utilisation

Ogni manuale è suddiviso in 10 parti (da 0→9), le parti sono suddivise in fascicoli (da 00→99) e i fascicoli in documenti (da 00→99).

Generalmente l'ordine con cui sono indicati i documenti corrispondono ad avere:

- **[Parte del manuale].[numero del fascicolo].[Numero del documento]**

(ad esempio U1.01.22)

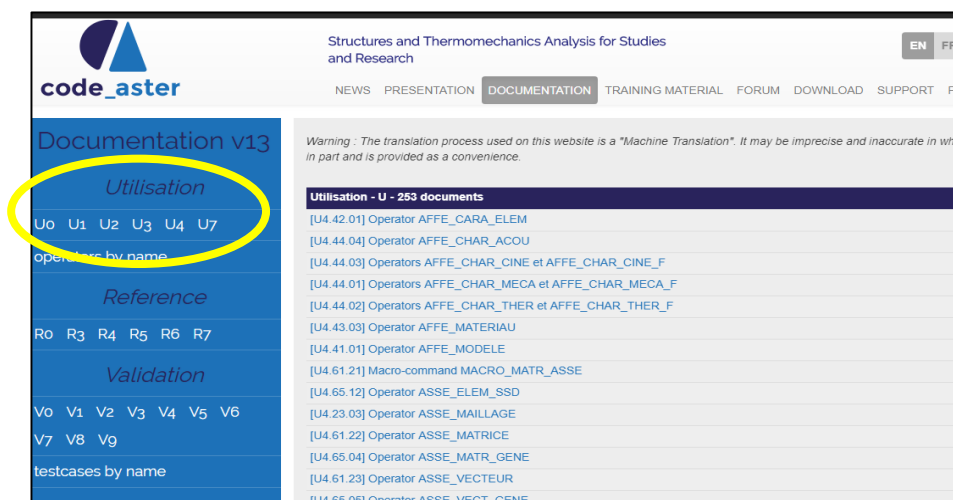


Figura 4.1 Documentazione disponibile (dal sito www.code-aster.org)

Solitamente i documenti sono suddivisi o per macro-argomenti, ed è il caso dei manuali U, oppure per i *test-case* effettuati; occorre comunque verificare di volta in volta il giusto percorso concettuale.

Tra le molteplici informazioni occorre ricordare che in testa ad ogni documento si troveranno: la versione del software di riferimento (molto importante in quanto ogni anno viene rilasciata una *release* dello stesso), il numero identificativo di revisione del documento, l'autore e la data di pubblicazione della revisione. Queste brevi indicazioni sono utili a reindirizzarsi correttamente nella scelta della versione di riferimento rilasciata su cui si sta lavorando.

Per il singolo utente i documenti necessari risulteranno essere suddivisi in determinate categorie che non verranno trattate tutte in quanto queste risultano essere diverse migliaia di pagine; queste categorie, soprattutto per i *test-case*, devono essere visualizzate per poter comprendere se esistono casistiche simili su cui basarsi per poter procedere all'analisi che si vuole affrontare, dunque a volte è consigliabile riferirsi alla sezione *Validation*.

All'interno del documento di presentazione di *Code-Aster* viene consigliata la lettura di tre documenti che presentano le principali linee guida e regole del software, nonché i principali comandi per iniziare a capire come eseguire le analisi, questi documenti sono [U1.02.00]. [U1.03.00] e [U1.05.00].

Un chiaro esempio sintetico di organizzazione del manuale *Utilisation U* viene espresso nel seguito, in cui si può notare che l'organizzazione sistematica della documentazione sarà generalmente la parte compresa tra U4 ed U7, ovvero:

- U4.1- Assegnazione di risorse in memoria;
- U4.2-, U7.01. - in U7.03. - Acquisizione dei dati dalla mesh;
- U4.3- e U4.4- Modellazione (assegnazione degli elementi finiti, materiali, carichi dei contenuti, ecc ...);
- U4.5- Risoluzione del sistema di equazioni (calcolo);
- U4.6-, U4.7-, U4.8-, U7.03 con U7.05. – Post-processamento ed analisi dei risultati;

Infine il documento [U4.01.00] descrive come capire alcuni comandi; spiega in particolare il significato dei metacaratteri, le funzioni e i percorsi da seguire una volta definito il tipo di analisi.

Generalmente la scelta degli elementi finiti per un tipo di modellazione è una responsabilità dell'utente, così come la descrizione matematica delle modellazioni supportate dagli elementi finiti, che si può ritrovare nel manuale *Référence R*.

La descrizione dei gradi di libertà di questi elementi come le loro possibilità di modellazione (metodologie di carico supportate, campi di output prodotti, possibilità di analisi non lineari, ecc ...) sono localizzati nei documenti:

- U3.1-: *modellazioni meccaniche*;
- U3.2-: *modellazioni termiche*;
- U3.3-: *modellazioni acustiche*.

E' molto importante ricordare che gli operatori *orders* creano delle strutture di dati che è essenziale conoscere e ricordare, per poter sapere le forme di risultati che vengono emesse dal processo di calcolo; di questo si parla all'interno di alcuni documenti molto importanti, e questi saranno i formati più usati durante le nostre analisi:

[U4.91.01] Procedure IMPR_RESU (FORMAT = 'RESULT' and 'ASTER')

[U7.05.21] Procedure IMPR_RESU (FORMAT=' MED').

5.5 Il manuale Reference

Come già detto, all'interno del manuale *Reference* possiamo trovare una serie di informazioni e nozioni utili alla formulazione del fenomeno di modellazione e dei metodi digitali di analisi e degli algoritmi; elementi essenziali per procedere nelle nostre analisi, infatti se grazie al manuale *Utilisation* possiamo sapere le principali istruzioni di comandi e funzioni, tramite questo manuale possiamo sapere gli incroci tra i vari comandi, le gerarchie dei dati da analizzare e le istruzioni di modellazione degli stessi. Per evitare di generare troppa confusione si consiglia di analizzare e focalizzarsi su un singolo obiettivo di analisi e di raccogliere tutta la documentazione necessaria procedendo a piccoli passi, in questo modo si evita di andare fuori argomento. All'interno di questo capitolo verrà fornita in maniera sintetica l'indicizzazione che viene fornita dal sito di *Code-Aster* riguardo il manuale *Reference*, ovvero uno dei tre Manuali che maggiormente verranno considerati.

Come si potrà osservare i capitoli sono suddivisi per argomentazione, scavalcando l'indice alfanumerico, in questo modo si vuole aiutare l'utente a ricercare con maggiore rapidità ed efficacia il campo di applicazione e, conseguentemente, anche i documenti che possano essere utili in particolari casi.

Si può vedere che ogni singolo documento è sicuramente molto corposo, strutturato e articolato, questo dovuto prevalentemente alla presenza di vari autori che hanno redatto i documenti, a volte scavalcando o ritrattando vari argomentazioni (*fonte www.code-aster.org*):

1 Algoritmi e metodologie

1.1 Analisi modale

[R5.01.01] Risolutore modale e risoluzione del problema generalizzato (GEP)

[R5.01.02] Risoluzione del problema modale quadratico (QEP)

[R5.01.03] Parametri modali e norma di autovettori

[R5.01.04] Procedura di enumerazione degli autovalori

[R5.07.01] Calcolo dei modi non lineari

1.2 Analisi dinamica transitoria

[R4.05.01] Risposta sismica mediante analisi transitoria

[R5.05.01] Soluzione di un'equazione differenziale del secondo ordine con il metodo di NIGAM

[R5.05.02] Algoritmi di integrazione diretta dell'operatore DYNA_LINE_TRAN

[R5.05.04] Modellazione dello smorzamento in dinamica lineare

[R5.05.05] Algoritmo dinamico non lineare

[R5.06.03] Modellazione degli shock e dell'attrito nell'analisi transitoria mediante ricombinazione modale

[R5.06.04] Algoritmi di integrazione temporale dell'operatore DYNA_TRAN_MODAL

1.3 Analisi dinamica frequente

[R5.05.03] Risposta armonica

1.4 Analisi spettrale e risposta casuale

[R4.05.02] Approccio stocastico per l'analisi sismica

[R4.05.03] Risposta sismica con metodo spettrale

[R7.10.01] Conteggio delle risposte aleatorie

1.5 Sotto-strutturazione dinamica

[R4.06.02] Calcolo modale per la sotto-strutturazione dinamica classica

[R4.06.03] Sotto-strutturazione dinamica ciclica

[R5.06.01] Riduzione del modello in dinamica lineare e non lineare: Metodo di RITZ

1.6 Estrapolazione di misure sperimentali

[R7.20.02] Estrapolazione di misure su un modello numerico in dinamica

1.7 Interazione Terreno-Struttura

[R4.05.04] Interazione terreno-struttura con variabilità spaziale (operatore DYNA_ISS_VARI)

[R4.05.05] Generazione di segnali sismici

[R4.05.06] Metodi lineari equivalenti per la propagazione delle onde in 1D

[R4.05.07] Interazione non lineare terreno-struttura tramite la teoria di Laplace-Temps

[R4.05.08] Metodo di presa in considerazione l'interazione Plancher-Materiel

[R 5.05.0 9] Calcolo dei segnali ricostituiti e della matrice funzione di trasferimento

1.8 Meccanica della rottura

[R7.02.01] Tasso di resa dell'energia in termoelasticità lineare

[R7.02.03] Tasso di resa dell'energia in termoelasticità non lineare

[R7.02.04] Modello di Beremin

[R7.02.05] Calcolo dei coefficienti di intensità dei vincoli nella termoelasticità lineare del piano

[R7.02.06] Modelli di Bordet e Rice and Tracey

[R7.02.07] Tasso di resa dell'energia in termo-elastoplasticità - Approccio GTP

[R7.02.08] Calcolo dei fattori di intensità dei vincoli mediante estrapolazione del campo di spostamento

[R7.02.09] Identificazione del modello di Weibull

- [R7.02.10] Analisi semplificata della nocività del difetto con il metodo K-beta
- [R7.02.13] Algoritmi di propagazione di fessure con X-FEM
- [R7.02.16] Metodo Gp: un approccio energetico alla predizione dello sfaldamento
- [R7.02.17] Fessura equivalente a partire da un campo di danneggiamento
- [R7.02.18] Elemento idro-meccanico accoppiato con XFEM
- [R7.02.19] Elementi coesivi con X-FEM
- [R7.02.20] Calcolo degli sforzi di taglio per estrapolazione dal campo degli spostamenti

1.9 Retiming

- [R4.03.06] Algoritmo di retiming

1.10 Metodi probabilistici

- [R4.03.05] Modelli probabilistici parametrici e non parametrici in dinamica

1.11 Condizioni al contorno e connessioni

- [R3.03.01] Dualità delle condizioni al contorno
- [R3.03.02] Condizioni di connessioni solide del corpo
- [R3.03.03] Collegamento trave 2D e trave 3D
- [R3.03.05] Eliminazione della dualità delle condizioni al contorno
- [R3.03.06] Collegamento guscio-trave
- [R3.03.08] Relazioni cinematiche lineari di tipo RBE3
- [R3.03.09] Raccordo Arlequin 3D – trave

1.12 Meccanica non lineare

- [R5.03.01] Algoritmo non lineare quasi-statico (operatore STAT_NON_LINE)
- [R5.03.03] Presa in conto degli sforzi piani nei comportamenti non lineari
- [R5.03.14] Integrazione implicita ed esplicita delle relazioni di comportamento non lineare
- [R5.03.80] Metodi di pilotaggio del carico
- [R5.03.81] Metodo IMPLEX
- [R7.01.02] Modellazione dei cavi di precompressione

1.13 Risolutori algebrici

- [R6.01.02] Informazioni generali sul gradiente combinato: GCPC Aster e uso di PETSc
- [R6.02.01] Informazioni sui metodi di decomposizione del tipo GAUSS
- [R6.02.02] Risolutore lineare con il metodo multi-frontale MULT_FRONT
- [R6.02.03] Panoramica sui risolutori lineari diretti e uso di MUMPS
- [R6.03.01] Risoluzione di sistemi non regolari mediante metodo di decomposizione a valore singolare

1.14 Fatica

[R7.04.01] Stima della durata di vita in affaticamento per un gran numero di cicli e per oligo-ciclico

[R7.04.02] Stima della fatica su sollecitazione aleatoria

[R7.04.04] Criteri di innesco della fatica multi-assiale

1.15 Termo-idro-meccanica accoppiata

[R7.01.10] Modelli THHM: generalità ed algoritmi

[R7.01.34] Schemi e volumi SUSHI per la modellazione dei flussi miscibili insaturi

1.16 Contatto e attrito

[R5.03.50] Formulazione discreta dell'attrito da contatto

[R5.03.52] Elementi di contatto derivati da una formulazione ibrida continua

[R5.03.53] Contatto in grandi scivolamenti con X-FEM

[R5.03.54] Contatto in piccoli scivolamenti con X-FEM

[R5.03.55] Metodo di LAC – Local Average Contact

1.17 Grandi deformazioni e grandi spostamenti

[R3.03.04] Sforzi esterni di pressione in grandi spostamenti

[R3.03.07] Pressione seguente per gli elementi voluminosi di gusci

[R5.03.21] Modellazione elasto-visco-plastica con indurimento isotropico con grandi deformazioni

[R5.03.22] Legge di comportamento in grandi rotazioni e piccole deformazioni

[R5.03.24] Modello di grandi deformazioni GDEF_LOG

[R5.03.40] Modellazione statica e dinamica delle travi in grandi rotazioni

1.18 Inarcamento

[R7.05.01] Criteri di stabilità strutturale

1.19 Termica

[R3.06.07] Diagonalizzazione della matrice di massa termica

[R5.02.01] Algoritmo termici lineari transitori

[R5.02.02] Termica non lineare

[R5.02.04] Termica non lineare nel riferimento mobile

1.20 Interazione Fluido-Struttura

[R4.07.02] Modellazione delle eccitazioni turbolente

[R4.07.03] Calcolo della matrice di massa aggiunta su base modale

[R4.07.04] Accoppiamento fluido-struttura per le strutture tubolari e gusci coassiali

[R4.07.05] Omogeneizzazione di una rete di travi immerse in un fluido

[R4.07.07] Identificazione degli sforzi dei fluidi

1.21 Analisi limitativa

[R7.07.01] Calcolo dei limiti di carico con il metodo di Norton-Hoff-Friaâ, comportamento NORTON_HOFF

1.22 Stimatori di errore

[R4.10.01] Stimatore di errore di ZHU-ZIENKIEWICZ

[R4.10.02] Stima dell'errore nel residuo

[R4.10.03] Indicatore di errore spaziale nel residuo per la termica transitoria

[R4.10.04] Rilevamento delle singolarità e calcolo di una mappa di dimensioni degli elementi

[R4.10.05] Indicazione dell'errore residuo per i modelli HM

[R4.10.06] Stimatori di errore in quantità di interesse

[R4.10.07] Analisi degli errori in relazione al comportamento in dinamica sotto una formulazione di frequenza

1.23 Post-trattamento

[R3.06.03] Calcolo dei vincoli ai nodi mediante livellamento locale

[R4.09.01] Bilancio energetico in termomeccanica

[R4.20.01] Indicazione dello scarico e perdita di proporzionalità del carico in elastoplasticità

[R7.04.03] Post-trattamento secondo RCC-M

[R7.04.05] Algoritmo di calcolo della densità di armatura

[R7.04.10] Operatore del calcolo dell'usura

[R7.10.02] Post-trattamento dei calcoli modali con shock

[R7.10.03] Post-trattamento dei calcoli di linea ad albero: diagramma di Campbell

[R7.20.01] Proiezione di un campo su una Mesh

1.24 Strumenti di lavoro

[R7.06.01] Modellazione della deformazione dei gruppi di combustibili con lo strumento MAC3

2 Leggi di comportamento

2.1 Generalità

[R5.03.27] Comportamenti meccanici per le simulazioni numeriche

2.2 Elementi discreti e travi

[R5.03.09] Relazioni del comportamento non lineare 1D

[R5.03.17] Relazioni del comportamento degli elementi discreti

2.3 Termo-elasticità

[R4.01.02] Elasticità anisotropa

[R4.08.01] Calcolo della deformazione termica

[R5.03.19] Legge del comportamento Iper-elastico: materiale quasi incompressibile

[R5.03.20] Relazione di comportamento elastico non lineare in grandi spostamenti

2.4 Elasto-plasticità incrementale

[R5.03.16] Comportamento elasto-plastico con indurimento misto isotropo e cinematico lineare

2.5 Elasto-visco-plasticità sotto irradiazione

[R5.03.23] Comportamento elasto-plastico sotto irradiazione di metalli: applicazione agli interni del serbatoio

2.6 Comportamento dei metalli policristallini

[R5.03.11] Comportamenti elasto-viscoplastici mono e policristallini

2.7 Geo-materiali e ambienti porosi

[R7.01.11] Modelli di comportamento THHM

[R7.01.13] Legge CJS in geo-meccanica

[R7.01.14] Legge di comportamento CAM_CLAY

[R7.01.15] Legge di comportamento di LAIGLE

[R7.01.16] Integrazione dei comportamenti di Drucker-Prager DRUCK_PRAGER e DRUCK_PRAG_N_A e post-trattamento

[R7.01.17] Legge di comportamento degli ambienti porosi: modello di BARCELONA

[R7.01.18] La legge di comportamento di HOEK_BROWN modificata

[R7.01.22] Legge di comportamento visco-plastica VISC_DRUC_PRAG

[R7.01.23] Legge ciclica di comportamento di Hujieux per i terreni

[R7.01.24] Legge di comportamento visco-plastica LETK

[R7.01.28] Legge di Mohr-Coulomb

[R7.01.38] Legge d'Iwan per il comportamento ciclico del materiale granulare

[R7.01.39] Legge di Rankine

[R7.01.40] Modello di comportamento LKR

2.8 Comportamento dei calcestruzzi

[R7.01.01] Relazione di comportamento BETON_TO ARRANGE per lo fluage del calcestruzzo

[R7.01.03] Legge di comportamento BETON_DOUBLE_DP con doppio criterio Drucker Prager per la fessurazione e la compressione del calcestruzzo

[R7.01.04] Legge di comportamento ENDO_ISOT_BETON

[R7.01.06] Relazione di comportamento BETON_UMLV per il fluage del calcestruzzo

[R7.01.08] Modello di danneggiamento di MAZAR

[R7.01.09] Legge di comportamento ENDO_ORTH_BETON

[R7.01.12] Modellizzazione della termo-idratazione, essiccazione e restringimento del calcestruzzo

[R7.01.19] Modellazione dell'accoppiamento fluage/plasticità del calcestruzzo

[R7.01.26] Legge di comportamento BETON_RAG

[R7.01.27] Legge di comportamento BETON_REGLE_PR

[R7.01.31] Legge di comportamento delle lastre in cemento armato GLRC_DAMAGE

[R7.01.32] Legge di comportamento GLRC_DM

[R7.01.35] Relazione di comportamento BETON_BURGER per il fluage del calcestruzzo

[R7.01.36] Relazione di comportamento KIT_RGI

[R7.01.37] Dissipative Homogenised Reinforced Concrete (DHRC) Modello costitutivo del cemento armato dedicato al rinforzo di piastre in calcestruzzo

2.9 Comportamento dei rinforzi nelle strutture in cemento armato

[R3.08.10] Elemento CABLE_GAINE

[R7.01.20] Comportamento dell'acciaio sottoposto a corrosione

[R7.01.21] Legge di comportamento (in 2D) per il collegamento acciaio/calcestruzzo: JOINT_BA

2.10 Danni

[R5.03.25] Legge di danneggiamento regolarizzata ENDO_SCALAIRE

[R5.03.26] Legge di danneggiamento quadratica regolarizzata ENDO_CARRE

[R5.03.28] Legge di danneggiamento del gradiente ENDO_FISS_EXP

[R5.04.01] Modellazione non locale con gradiente di variabili interne

[R5.04.03] Secondo gradiente di modellazione

[R5.04.04] Modellazione non locale con gradienti di danno nodale GVNO

[R7.01.29] Legge di comportamento ENDO_HETEROGENE

[R7.01.33] Accoppiamento elasto-plastico-danno

2.11 Modelli metallurgici

[R4.04.01] Modello del comportamento metallurgico degli acciai

[R4.04.02] Modellazione elastica(visco)plastica che tiene in conto delle trasformazioni metallurgiche

[R4.04.03] Legge di comportamento elasto(visco)plastica in grandi deformazioni con trasformazioni metallurgiche

[R4.04.04] Modello di comportamento metallurgico dello zircaloy

[R4.04.05] Modello del comportamento elasto-viscosa META_LEMA_ANI tenendo conto della metallurgia per i tubi della guaina del perno del carburante

2.12 Visco-elasticità

[R5.03.08] Integrazione delle relazioni visco-elastiche di comportamento

[R5.0 5.10] Analisi dinamica di strutture con materiali visco-elastici aventi proprietà dipendenti dalla frequenza

2.13 Elasto-visco-plasticità dei metalli

[R5.03.02] Integrazione delle relazioni di comportamento elasto-plastico di Von Mises

[R5.03.04] Relazioni di comportamento Elasto-visco-plastico di Chaboche

[R5.03.05] Relazione di comportamento viscoplastico TAHERI

[R5.03.06] Modello di Rousselier in grandi deformazioni

[R5.03.07] Modello di Rousselier per la rottura duttile

[R5.03.12] Comportamento visco-plastico con effetto di memoria e restaurazione di Chaboche

[R5.03.13] Comportamento visco-plastico con danno di HAYHURST

[R5.03.15] Comportamento visco-plastico con danno di Chaboche

[R5.03.32] Legge di comportamento dell'assemblaggio ASSE_CORN

2.14 Leggi coesive e di congiunzioni

[R3.06.09] Elementi finiti di giunzioni meccaniche ed elementi finiti di giunzioni idraulici accoppiati

[R7.01.25] Regole di comportamento dei giunti di arresto: JOINT_MECA_RUPT e JOINT_MECA_FROT

[R7.02.11] Leggi di comportamento coesivo CZM e pilotaggio del carico

3 elementi finiti

3.1 Elementi finiti

[R3.01.00] metodo degli elementi finiti iso-parametrico

[R3.01.01] Funzioni di forma e punti di integrazione degli elementi finiti

[R3.06.02] Modellazione lineare degli elementi del mezzo continuo in modelli termici

[R3.06.04] Elementi di Fourier per le strutture assial-simmetriche

[R3.06.10] Elemento quadrangolare in un punto di integrazione, stabilizzato dal metodo "Assumed Strain"

[R3.06.11] Elemento esaedrico in un punto di integrazione, stabilizzato dal metodo "Assumed Strain"

[R4.02.01] Elementi finiti in acustica

[R4.02.02] Elementi vibro-acustici

[R4.02.04] Giunto fluido - Struttura con superficie libera

[R4.02.05] Elementi di confine assorbente

3.2 Elementi di struttura

[R3.07.02] Modellazione numerica delle strutture medie: gusci termoelasto-plastici assial-simmetrici e 1D

[R3.07.03] Elementi della piastra: modelli DKT, DST, DKTG e Q4g

[R3.07.04] Elementi finiti di gusci voluminosi

[R3.07.05] Elementi di gusci voluminosi non lineari

[R3.07.06] Trattamento della compensazione per gli elementi della piastra

[R3.07.07] Elemento di guscio voluminoso SHB ad 8 nodi

[R3.07.08] Elemento di guscio voluminoso SHB ad 6,15 e 20 nodi

[R3.07.09] Elementi della piastra: modellazione Q4GG

[R3.08.01] Elementi "esatti" di travi

[R3.08.02] Modellazione dei cavi

[R3.08.03] Calcolo delle caratteristiche di una sezione trasversale qualunque di una trave

[R3.08.04] Elemento di trave a 7 gradi di libertà per la presa in considerazione della deformazione

[R3.08.05] Un elemento finito di cavo-puleggia

[R3.08.06] Elementi finiti di tubo dritto o curvo

[R3.08.07] Elementi MEMBRANE e GRILLE_MEMBRANE

[R3.08.08] Elementi multi-fibre della trave (dritta)

[R3.08.09] Trave multi fibra in grandi spostamenti

[R3.11.01] Formulazione di un modello termico per gusci sottili

[R4.01.01] Pre e post-processing per gli gusci sottili "compositi"

[R4.02.03] travi elasto-acustiche

[R5.05.07] Matrice giroscopica di travi dritte e di dischi

[R5.05.08] Modellazione dei rotori incrinati con rigidità equivalente a seconda dell'angolo di rotazione

3.3 Elementi finiti incompressibili

[R3.06.08] Elementi finiti che trattano la quasi incompressibilità

3.4 Elementi di articolazioni, interfaccia e discontinuità

[R3.06.09] Elementi finiti di giunzioni meccaniche ed elementi finiti di giunzioni idromeccaniche accoppiate

[R3.06.13] Elemento finito ad interfaccia mista per modelli di zone coesive (xxx_INTERFACE e xxx_INTERFACE_S)

[R7.02.12] eXtended Finite Element Method

[R7.02.14] Elemento con discontinuità interna

[R7.02.15] Modellazione delle fessure con accoppiamento idro-meccanico in ambiente poroso-saturo.

5.6 *Il manuale Validation*

Il manuale *Validation* di tutta la documentazione di *Code-Aster* può essere ritenuto quello più diretto e rappresentativo, infatti al suo interno vi sono numerosi test-case di svariati argomenti; ogni test si compone di una breve descrizione, non troppo specifica, ma necessaria a spiegare il percorso di funzioni adottate all'interno del test.

Questi test di esempio costituiscono un valido supporto per affrontare problemi che siano vagamente simili, infatti da questi è possibile stabilire a priori quali strade possano essere escluse per il conseguimento del risultato prefissato.

Ogni opuscolo è identificato da una chiave relativa a tutte le pagine, tale parola chiave consente una gestione progressiva della documentazione.

Parola chiave: ***Parte del Manuale. N° fascicolo [N° documento]***

Ad ogni test è associato un riferimento codificato, questa codifica formata da 8 caratteri alfanumerici considera:

- Le prime 4 lettere riferite al tipo di problema da affrontare;
- Il numero progressivo dei test nella lista dei test-case del medesimo tipo;

I quattro caratteri alfabetici indicano nell'ordine:

1_ *L'ambito di applicazione;*

2_ *Il tipo di analisi;*

3_ *Il tipo di comportamento;*

4_ *Il tipo di modello.*

Nel seguito viene indicata la metodologia di indicizzazione dei test-case:

Tabella 5.2 Indicizzazione alfanumerica dei test-case

1	2	3	4	5 6 7
L'ambito di applicazione	Il tipo di analisi	Il tipo di comportamento	Il tipo di modello	Sequenza Numerica
S Meccanica delle strutture	S Statica	L Lineare	D Discreto	01 to 999
T Termica	D Dinamica	NR Non lineare	L Lineare (trave)	
H Termomeccanica	P Stazionario	C Condizioni al contorno non-lineari	S Superficie (guscio piastra)	
F Fluidi	T Transitorio		P Piano (2D)	
A Acustica	H Armonico		A Assial-simmetrico	
W Ambiente poroso	Z Fatica		V Voluminoso (3D)	
W Metallurgia			X Modelli mixati	

Manuale di Validazione

V0: Informazioni generali

V1: Test di elaborazione dati

V2: Dinamica lineare

V3: Statica lineare

V4: Termica stazionaria, transitorio, con radiazioni, metallurgia

V5: Dinamica non lineare

V6: statica non lineare

V7: Termo-idro-meccanica

V8: Fluido, Acustica

V9: Qualsiasi altra attività (Fatica, ...)

Test-Case disponibili

Parte [V0]: informazioni generali

-Opuscolo [V0.00]: informazioni generali sulla documentazione di convalida

Parte [V1]: test di elaborazione dati (prestazioni, ...)

-Libretto [V1.01]: Test di validità dei comandi

-Libretto [V1.10]: test di performance

Parte [V2]: dinamica lineare

- Libretto [V2.01]: Dinamica lineare dei sistemi discreti
- Libretto [V2.02]: Dinamica lineare delle travi
- Libretto [V2.03]: Dinamica lineare degli gusci e dei piastre
- Libretto [V2.04]: Dinamica lineare delle strutture tridimensionali
- Libretto [V2.05]: Dinamica lineare delle strutture assemblate
- Libretto [V2.06]: Risposta armonica delle strutture lineari
- Libretto [V2.07]: Risposta armonica delle strutture tridimensionali

Parte [V3]: statica lineare

- Libretto [V3.01]: Statica lineare delle strutture lineari
- Libretto [V3.02]: Statica lineare dei sistemi piani
- Libretto [V3.03]: Statica lineare degli gusci e delle piastre
- Libretto [V3.04]: Statica lineare delle strutture tridimensionali
- Libretto [V3.05]: Statica lineare delle strutture assemblate
- Libretto [V3.90]: Riferimenti teorici di test in statica lineare

Parte [V4]: Termica stazionaria, transitoria, con radiazioni, metallurgica

-Libretto [V4.01]: Analisi Termica stazionaria delle strutture assial-simmetriche

- Libretto [V4.02]: Analisi Termica stazionaria delle strutture lineari
- Libretto [V4.03]: Analisi Termica stazionaria delle piastre e dei gusci
- Libretto [V4.04]: Analisi Termica stazionaria delle strutture tridimensionali
- Libretto [V4.21]: Analisi Termica transitoria delle strutture lineari
- Libretto [V4.22]: Analisi Termica transitoria non lineare delle strutture lineari
- Libretto [V4.23]: Analisi Termica transitoria dei sistemi piani
- Libretto [V4.25]: Analisi Termica transitoria delle strutture tridimensionali
- Libretto [V4.41]: Analisi Termica stazionaria con radiazioni
- Libretto [V4.61]: Metallurgia

Parte [V5]: dinamica non lineare

- Libretto [V5.01]: Dinamica non lineare dei sistemi discreti
- Libretto [V5.02]: Dinamica non lineare delle strutture lineari
- Libretto [V5.03]: Dinamica non lineare delle strutture tridimensionali

Parte [V6]: statica non lineare

- Libretto [V6.01]: Statica non lineare in assial-asimmetria
- Libretto [V6.02]: Statica non lineare delle strutture lineari
- Opuscolo [V6.03]: Statica non lineare dei sistemi piani
- Libretto [V6.04]: Statica non lineare delle strutture tridimensionali

Parte [V7]: Termomeccanica

-Libretto [V7.01]: Termomeccanica lineare stazionaria delle strutture assial-simmetriche

-Libretto [V7.02]: termomeccanico lineare stazionaria dei sistemi piani

-Libretto [V7.03]: termomeccanico lineare stazionaria dei sistemi tridimensionali

-Libretto [V7.11]: termomeccanica statica lineare delle piastre e dei gusci

-Libretto [V7.21]: termomeccanica statica non lineare delle strutture lineari

-Libretto [V7.22]: termomeccanica statica non lineare delle strutture tridimensionali

-Libretto [V7.31]: termo-idro-meccanica in ambienti porosi saturi

-Libretto [V7.41]: transitorio termomeccanico non lineare dei piani di sistema

-Libretto [V7.90]: Riferimenti teorici di test in termomeccanica

Parte [V8]: Fluidi, Acustica

-Libretto [V8.01]: Fluidi

-Libretto [V8.21]: Acustica modale

-Libretto [V8.22]: acustica armonica

5.7 I campi di analisi possibili e le funzioni che li richiamano

In questo capitolo si analizzeranno le grandezze che tratteremo nell'analisi degli elementi strutturali ed indicheremo le funzioni che richiamano tali grandezze.

Per i campi di analisi ci riferiremo principalmente a grandezze reali (DEPL_R, SIEF_R, ...) e verranno elencati le grandezze che si possono estrapolare da questi campi di analisi. Ogni singola grandezza ha una cifratura definita e deve essere sempre collegata al tipo di analisi effettuata. Ad esempio, rimanendo nel campo di risultati reali (DEPL_R), possiamo effettuare un'analisi che generi gli spostamenti richiamati tramite la funzione spostamento (DEPL); ovvero questi saranno gli spostamenti in campo reale della struttura da cui è possibile estrarre le rotazioni e gli spostamenti. Un altro esempio può riguardare il campo delle tensioni/sollecitazioni in campo reale della struttura (SIEF_R) da cui è possibile estrapolare le tensioni e le sollecitazioni calcolate richiamando la funzione che vogliamo (EFGE_ELNO, EFGE_ELGA, ...) che rappresenta il metodo di analisi scelto a monte.

Si ricorda che in *Code-Aster* il campo di risultati deve essere associato ad una funzione di richiamo corretta (vedi cap 4.7.2) al fine di poter estrapolare la corretta grandezza.

5.7.1 Le grandezze associate ai campi di forze

SIEF_R: rappresenta lo stato di tensione della struttura, e quindi contiene al suo interno almeno le seguenti grandezze:

Campi di tensione dei mezzi continui (SR globale):

SIXX SIYY SIZZ SIXY SIXZ SIYZ

Sollecitazioni della trave e dei mezzi discreti (SR locale per trave e per i mezzi discreti):

N VY VZ MT MFY MFZ

Per Le travi irregolari si aggiunge un doppio momento (necessario nel SR locale con fibre):

BX

Sollecitazioni per piastre (SR locale sulla superficie):

NXX NYY NXY MXX MYX MXY QX QY

A volte è conveniente utilizzare le sollecitazioni per travi, e per i discreti, nel SR globale:

FX FY FZ MX MY MZ

E' comunque utile alcune volte utilizzare i campi di tensione nel sistema di riferimento locale per elementi trave e per e piastre; in questi casi *Code-Aster* utilizza le medesime diciture usate nel SR globale anche se può esserci confusione.

FORC_F e **FORC_R**: questi campi rappresentano le forze applicate sull'interfaccia di una struttura. Per:

- Per un mezzo continuo consiste un vettore di forze;
- Per una trave un tensore di forze;
- Per una lastra un tensore di forze;

Questi campi devono avere le seguenti grandezze:

Per un mezzo continuo:

FX FY FZ

Inoltre per travi e piastre:

MX MY MZ

DEPL_R: Siccome in *Code-Aster*: un campo può essere collegato ad una sola grandezza, che i metodi ad elementi finiti possono mescolarsi (mescolandosi in maniera sconosciuta con spostamenti standard e fattori di forze nodali sconosciuti), che la dualità con le condizioni al contorno comporta avere un componente del vettore di Lagrange che sono forze nodali nel verso che è stato specificato, che è necessario essere in grado di eseguire qualunque combinazione lineare con le forze nodali, che la numerazione delle incognite deve essere la medesima di quelli a secondo membro, allora le forze nodali le quali sono duali nel senso energetico W agli spostamenti nodali, avranno necessariamente le medesime componenti degli spostamenti:

DX DY DZ DRX DRY DRZ

Inoltre, per le travi curve si aggiunge il doppio momento: **GRX**

5.7.2 Calcolo degli Stati di Tensione (Opzioni di calcolo)

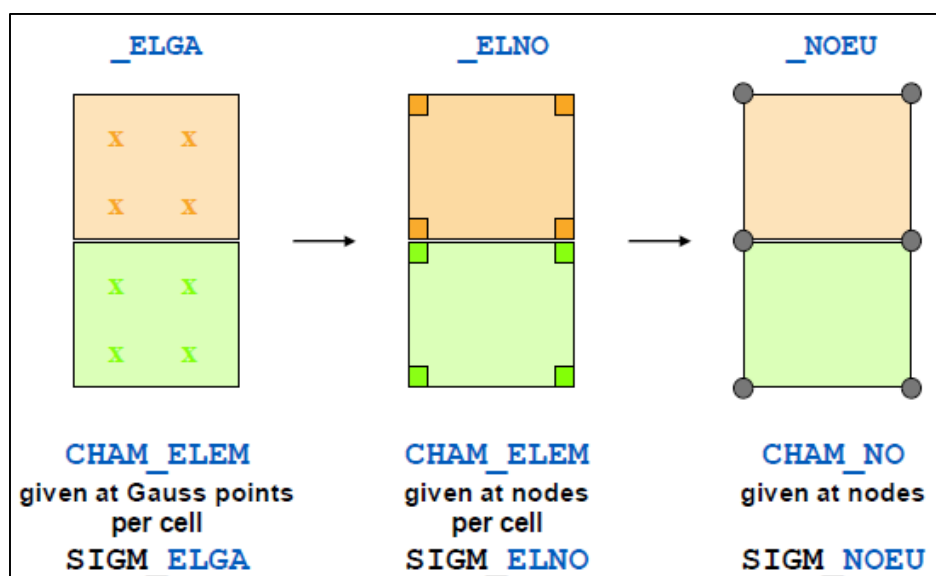


Figura 5.1 Campi di forze utilizzati

SIEF_ELGA: E' il campo rappresentativo degli stati di tensione e permette di proseguire i calcoli (rigidezza, forze nodali, ...). Viene espresso nei punti di Gauss (e nei sottopunti per gli elementi strutturali). Il prefisso del campo è **SIEF**, perché in base agli elementi, contiene sollecitazioni (E_{ffort}) o tensioni (Sigma) per elementi di Gauss.

Tabella 5.3 Campo, analisi e risultati del campo SIEF_ELGA

Opzioni di calcolo	Simbolo di riferimento del concetto RESULT	Calcoli che si vogliono ottenere	3D, 2D, Gusci 1D, Tubolari, Travi Multifibre	Travi: POU_D_E, POU_D_T, POU_D_TG, Discrete	Piastre: DKT, DST, Q4G, Q4GG, setti
SIEF_ELGA	SIEF_ELGA	Partendo dal campo spostamenti in analisi elastica-lineare	σ	[M, F] SR locale	σ SR locale (*)

In questo modo è possibile calcolare le tensioni nei mezzi contini 2D e 3D, e gli elementi a comportamento locale: COQUE_3D, piastre, gusci 1D, tubi, travi multifibre, ed in ogni integrazione dei sotto-punti. La terna nel SR locale per piastre e gusci deve essere specifico per ogni elemento.

SIGM_ELGA: questo campo rappresenta lo stato di tensione nei punti di Gauss (e nei sottopunti). Il prefisso è **SIGM** poiché all'interno di questo campo vi sono solo tensioni (quindi sono un'estrazione di grandezze dal campo **SIEF_ELGA**).

Tabella 5.4 Campo, analisi e risultati del campo SIGM_ELGA

Opzioni di calcolo	Simbolo di riferimento del concetto RESULT	Calcoli che si vogliono ottenere	3D, 2D, Gusci 1D, Tubolari, Travi Multifibre	Travi: POU_D_E, POU_D_T, POU_D_TG, POU_D_GD Discrete	Piastre: DKT, DST, Q4G, Q4GG, setti
SIGM_ELGA	SIGM_ELGA	Partendo dal campo SIEF_ELGA, estrazione delle tensioni	σ	Non calcolabile	σ SR locale (*)

SIEF_ELNO – SIEF_NOEU: questo campo rappresenta lo stato di tensioni/sollecitazione al fine di calcolare/visualizzare per elementi con nodi agli estremi (o ai sottopunti) e ai nodi dell'elemento. Dunque può contenere tensioni o sollecitazioni.

Tabella 5.5 Campo, analisi e risultati del campo SIEF_ELNO / SIEF_NOEU

Opzioni di calcolo	Simbolo di riferimento del concetto RESULT	Calcoli che si vogliono ottenere	3D, 2D, Gusci 1D, Tubolari, Travi Multifibre	Travi: POU_D_E, POU_D_T, POU_D_TG, POU_D_GD Discrete	Piastre: DKT, DST, Q4G, setti
SIEF_ELNO	SIEF_ELNO	Estrapolando i risultati dai nodi delle quantità riferite ai punti di Gauss	σ	[<i>M</i> , <i>F</i>] SR locale	σ SR locale (*)
x	SIEF_NOEU	Aritmeticamente con calcoli ai nodi riferendosi alle quantità dei nodi degli elementi	σ	[<i>M</i> , <i>F</i>] SR locale	σ SR locale (*)

SIGM_ELNO – SIGM_NOEU: questo campo rappresenta lo stato di tensioni a fine di calcolare/visualizzare per elementi con nodi agli estremi (o ai sottopunti) e ai nodi dell'elemento. Il prefisso SIGM indica che in questo campo sono contenute solo tensioni.

Tabella 5.6 Campo, analisi e risultati del campo SIGM_ELNO / SIGM_NOEU

Opzioni di calcolo	Simbolo di riferimento del concetto RESULT	Calcoli che si vogliono ottenere	3D, 2D, Gusci 1D, Tubolari, Travi Multifibre	Travi: POU_D_E, POU_D_T, POU_D_TG, POU_D_GD Discrete	Piastre: DKT, DST, Q4G, setti
SIGM_ELNO	SIGM_ELNO	Estrapolando i risultati dai nodi delle quantità riferite ai punti di Gauss	σ	Non calcolabile	σ SR locale (*)
x	SIGM_NOEU	Aritmeticamente con calcoli ai nodi riferendosi alle quantità dei nodi degli elementi	σ	Non calcolabile	σ SR locale (*)

EFGE_ELGA, EFGE_ELNO and EFGE_NOEU: questo campo rappresenta lo stato di sollecitazione generalizzato per le travi o per i gusci al fine di calcolare/visualizzare per i punti di Gauss, per elementi con nodi agli estremi o ai nodi dell'elemento le caratteristiche della sollecitazione o le forze.

Tabella 5.7 Campo, analisi e risultati del campo EFGE_ALGA / EFGE_ELNO

Opzioni di calcolo	Simbolo di riferimento del concetto RESULT	Calcoli che si vogliono ottenere	3D, 2D	Travi, tubolari, travi multi-fibre, Discrete	Piastre, lastre
EFGE_ELGA	EFGE_ELGA	Dalle integrazioni delle tensioni	Non calcolabile	[M , F] SR locale	[N , M , V] SR locale (*)
EFGE_ELNO	EFGE_ELNO	Partendo dal campo spostamenti in analisi elastica-lineare	Non calcolabile	[M , F] SR locale	[N , M , V] SR locale (*)
EFGE_ELNO	EFGE_ELNO	Dalle integrazioni delle tensioni non lineari	Non calcolabile	[M , F] SR locale	[N , M , V] SR locale (*)
x	EFGE_NOEU	Aritmeticamente con calcoli ai nodi riferendosi alle quantità dei nodi degli elementi	Non calcolabile	[M , F] SR locale	[N , M , V] SR locale (*)

5.7.3 Calcolo delle forze nodali e delle reazioni generalizzate

Le reazioni nodali sono calcolate solo tramite un procedimento, partendo dallo stato di tensione, ed è possibile solo un'opzione.

Tabella 5.8 Campo, analisi e risultati del campo FORC_NODA

Opzioni di calcolo	Simbolo di riferimento del concetto RESULT	Calcoli che si vogliono ottenere	3D, 2D	Travi, Elementi discreti	Lastre
FORC_NODA	FORC_NODA	Partendo dal campo SIEF_ELGA	Forze	[M , F] SR locale	[M , F] SR locale (*)

Le forze nodali, duali in termini energetici W agli spostamenti, hanno le medesime componenti nominative degli spostamenti:

DX DY DZ DRX DRY DRZ

L'opzione **REAC_NODA** all'interno della funzione **CALC_CHAMP** effettua una chiamata a **FORC_NODA** e sottrae:

- Carichi statici;
- Carichi, le forze inerziali e viscosi in dinamica.

In generale tutti i valori ottenuti da **FORC_NODA** partono dalle dimensioni e dalle caratteristiche del vincolo a cui essi sono associati.

Per le travi, e per nessun'altro elemento, il campo **SIPO_ELNO** presenta alcune caratteristiche: il campo indica le tensioni sulla trave (Sigma POutre), per elementi con nodi agli estremi. Questo campo particolare rappresenta le tensioni prodotto da ognuna delle tre forze e momenti definite in **SIEF_ELNO** se queste agissero da sole sulla sezione della trave. Per esempio **SMFY** è la tensione duale al singolo momento flettente **MFY**, il quale è calcolato partendo da appunto il momento duale e le caratteristiche della sezione definite in **AFFE_CARA_ELEM**.

Infine sempre e solo per le travi **SIPM_ELNO** (Sigma Poutre Maximum) mi restituisce le componenti della tensione estrema (massima o minima), nella direzione indicata dalle ultime 2 lettere del nome ed è riferito ai nodi dell'elemento. Il più importante **SIXX**, può essere visto come estremo (massimo o minimo) di tensione normale, ad esempio come il risultato dovuto ad una forza normale e un doppio momento applicato nella direzione XX.

Per le travi è importante inoltre un'altra condizione, la tensione duale al momento torcente è vera solo se la condizione di "*Saint Venant*" sono dichiarate, questo indicherà che non vi sono deformazioni della sezione. Questo tuttavia è falso in quanto la sezione si differenzia da una curvilinea chiusa ad una sezione aperta e/o se la trave deformata è limitata anche alle condizioni al contorno. La tensione può quindi arrivare ad essere molto più alta, anche di dieci volte o più, il problema non può essere risolto rigorosamente da questo tipo di analisi della trave.

Una caratteristica molto importante nell'analisi in *Paravis* consiste nella rappresentazione dei risultati (di Default) tramite l'opzione "magnitude"; questo setting produce il modulo del vettore formato dalla somma delle tre componenti. Per gli spostamenti questo risultato è la somma dei tre spostamenti DX, DY e DZ ed è utile; Per **SIEF_ELNO** o **SIPO_ELNO** questo risultato non ha molta importanza in termini ingegneristici.

Per le lastre i campi **SIEF_ELGA** e **SIEF_ELNO** forniscono 9 componenti, i primi tre sono gli sforzi normali (N, ...) gli altri tre i momenti (M, ...) e gli ultimi tre sono gli sforzi di taglio (V, ...) riferito agli assi locali dell'elemento.

N.B. attenzione a non fare confusione tra il SR locale e il SR globale.

(*) per gli elementi di piastre, gusci il SR locale è definito in base alle assunzioni definite all'interno della funzione **AFFE_CARA_ELEM** (vedi cap. 4.8.3)

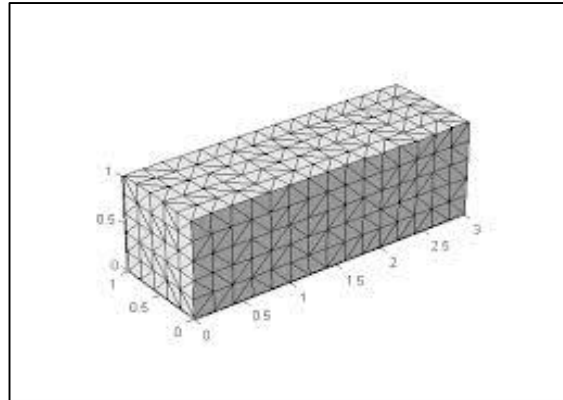


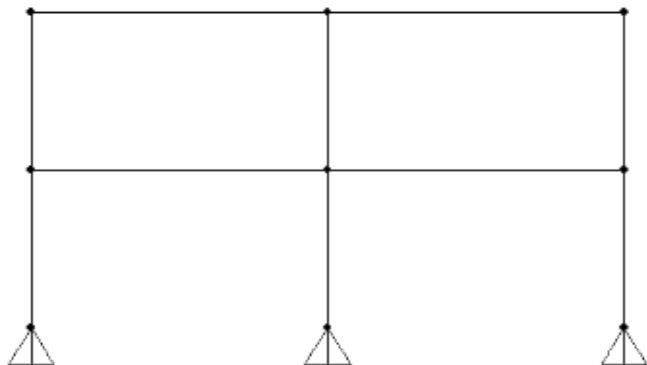
Figura 5.2 Elemento discretizzato

5.8 Il metodo di analisi FEM del software

In questo capitolo ci si dedicherà allo studio di un'analisi FEM che venga eseguita tramite il linguaggio macchina, ovvero tramite la piattaforma *Salomé-Meca*, non si tratteranno i temi nello specifico poiché già è stato fatto nei precedenti capitoli, piuttosto ci si concentrerà ad evidenziare gli elementi più importanti e più essenziali per effettuare l'analisi all'interno del software in maniera da rendere il progetto finale un progetto eseguibile e comprensibile anche per chi non ha mai utilizzato *Code-Aster*.

5.8.1 Discretizzazione

Il metodo di analisi strutturale agli elementi finiti FEM si fonda sostanzialmente sul concetto di discretizzazione: ovvero descrivere la struttura attraverso un numero finito di punti significa dividerla in un sistema equivalente di elementi strutturali (e non) più piccoli, tali che il loro assemblaggio dia luogo alla struttura originale.



La discretizzazione può variare dal più semplice degli elementi strutturali, come ad esempio il telaio, fino ad arrivare ad elementi più complessi, l'importante è che si rispettino le condizioni al contorno e di carico per tutti gli elementi strutturali.

Il metodo agli elementi finiti si basa dunque di correlare a questa discretizzazione, una funzione di spostamento che servirà per descriverne lo stato di deformazione del sistema, associando dunque ogni singola soluzione per tutti gli elementi discretizzati ad un sistema di riferimento locale il quale farà da cardine per espandere le soluzioni ed assemblarle a tutti gli elementi ed al sistema di riferimento globale, in modo da ottenere una soluzione per ogni singolo elemento finito che è correlata a tutto il sistema nel suo complesso.

5.8.2 Tipologia di elementi e modello di calcolo

Gli elementi che nella modellazione useremo sono vari, ma i più comuni sono il linguaggio base della modellazione strutturale. Le più utilizzate sono:

FRAME: elemento asta con sollecitazioni N , T_X , T_Y , M_X , M_Y , M_Z .

SHELL: è un elemento bidimensionale nelle sue varianti piastra (regime degli sforzi flettenti), membrana (sforzi solo nel piano dell'elemento, N , T .) o guscio (somma dei due).

SPRING: elemento molla di rigidezza k , usato in alcune varianti della modellazione strutturale, ed adoperato nella modellazione geotecnica (utile per modellare suolo elastico alla winkler).

RIGEL – CONSTRAINT: elementi di legame rigido. Consentono di legare con infinita rigidezza uno o più nodi secondari a uno principale; in questo modo si consente di inserire delle condizioni di vincolo ausiliarie.

Analizzando maggiormente nel dettaglio l'elemento Frame, ovvero l'elemento più semplice e comune potremmo osservare le seguenti caratteristiche, l'elemento frame è una generica asta tridimensionale, che include gli effetti di deformazione assiale, taglio, torsione e momento flettente. E' definito da una linea retta che congiunge due punti nodali.

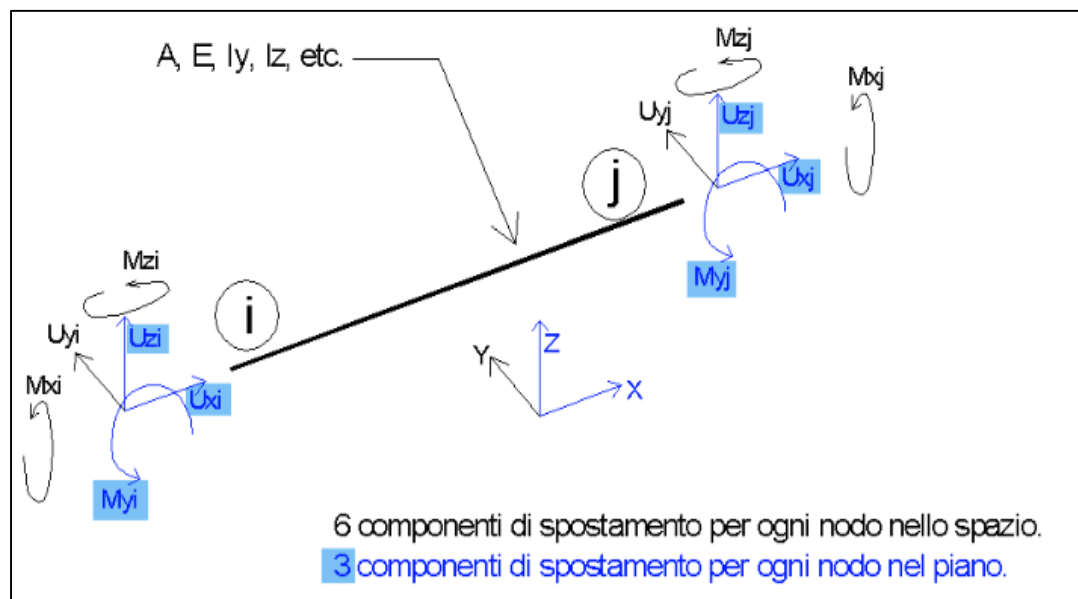


Figura 5.3 Direzione e versi delle caratteristiche geometriche e di sollecitazione

Utilizzato nella modellazione spaziale, e nella modellazione lineare a questo elemento vengono inglobate anche le leggi della meccanica e della fisica che ci consentono di definire il comportamento del materiale, e quindi dell'elemento, quando esso è sottoposto a carico:

- E modulo di Young (MPa);
- G modulo di taglio (MPa);

A questi possono aggiungersi anche:

- ❖ **Coefficiente di espansione termica:** Per analisi di sforzi derivanti da dilatazioni termiche;
- ❖ **Massa per unità di volume:** Per analisi sismica;
- ❖ **Peso per unità di volume:** Per avere il calcolo automatico del peso proprio della struttura, o gli sforzi di compressione sui pilastri calcolati automaticamente dal programma di calcolo.

Oltre a questo è necessario aggiungere alcune proprietà geometriche della sezione che abbiamo già riscontrato frequentemente in altri corsi e non solo:

A Area della sezione (m^2)

I_x Momento d'inerzia (mm^4)

I_y Momento d'inerzia (mm^4)

J Costante torsionale

E' importante notare che deve essere mantenuta una necessaria congruenza tra le unità di misura data in input al programma.

Una volta definite le sue proprietà geometriche e meccaniche, e i vincoli con l'esterno l'elemento è perfettamente congruente all'elemento originale nelle sue proprietà strutturali e meccaniche.

E' importante ricordare che nella modellazione strutturale si può pensare di analizzare qualunque elemento strutturale, ma è necessario che questo elemento sia opportunamente verificato per i carichi che andrà a sopportare.

Una buona impostazione del modello deve essere fatta anche considerando tutte le caratteristiche della struttura in essere, ma di questo ne abbiamo parlato già nei precedenti capitoli indicando come e quali elementi siano favorevoli ad una modellazione efficace.

Nello specifico ora analizziamo gli elementi che ci consentono di giungere ad una modellazione efficace utilizzando elementi propedeutici:

Joints: Punti nodali che definiscono la geometria e la collocazione nello spazio della struttura;

Frames (o altro): Tipo di elemento finito esistente tra i joints;

Material properties: Caratteristiche meccaniche dei materiali utilizzati (E,G, massa, ecc.);

Frame properties: Caratteristiche geometriche della sezione (A, I_x , I_y , ecc);

Restraints: vincoli da applicare ai joints per rendere il sistema iso o iperstatico;

Loads: Carichi applicati (concentrati o ripartiti, statici o dinamici), sia applicati ai nodi che alle superfici degli elementi finiti;

Go: Avvio della procedura di calcolo.

5.8.3 *Orientamento degli elementi trave*

In questo capitolo verrà descritto brevemente il concetto di orientamento, molto caro a *Code-Aster*, a cui ci si dovrà riferire per la generazione e modellazione, il calcolo e la stampa dei risultati in maniera corretta ed esaustiva.

Come si vede nell'immagine la "TopBeam" di riferimento ha l'asse Y locale che giace sul piano OXY globale. La stessa cosa vale per "Mast", tuttavia mentre gli elementi si estendono lungo Z, gli assi Y locale ed Y globale coincidono. I calcoli verranno comunque riferiti alla terna globale.

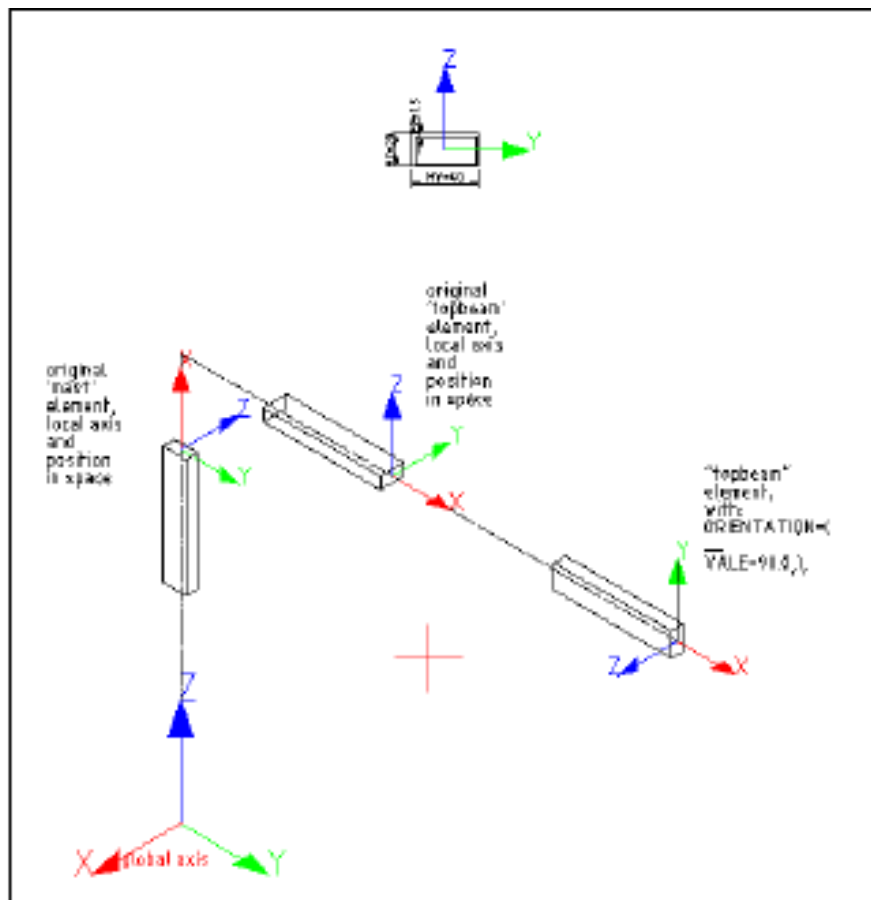


Figura 5.4 Esempio di orientamento della SR locale

Per avere questo elemento trave modificato all'interno di *Asterstudy*, siccome *Asterstudy* provvede ad essere specifico, si può intervenire all'interno di `AFFE_CARA_ELEM` ed intervenire in `ORIENTATION` andando a specificare in `VECT_Y` i valori che posizioneranno l'asse Y dell'elemento nel SR globale.

Si precisa che la seguente operazione posiziona l'asse delle Y locale sull'asse delle X globale.

```
ORIENTATION=_F(
#selezioniamo l'elemento a cui corrisponderà la nostra terna di rotazione
GROUP_MA="Mast",)
#la seguente funziona indica che noi vogliamo spostare il vettore Ylocale
CARA='VECT_Y,
VALE=(1.0, 0.0, 0.0),
),
```

In sintesi la regola per orientare correttamente gli assi di una trave possono essere definite in queste:

- L'asse locale X giace lungo l'asse della trave;
- L'asse locale Y giace di Default nel piano XOY globale;
- L'asse Z completa la terna;

Un altro metodo per l'orientamento che viene suggerito è quello di mantenere l'asse Z come su cui agisce il vettore delle forze peso, X è l'asse di direzione della *Mesh* o di un veicolo in moto e Y completa la terna.

Si ricorda inoltre che variare l'orientamento della trave, comporta il cambiamento della terna locale, che a sua volta comporta il cambiamento della direzione di forze e momenti. Questo ovviamente vale anche se consideriamo le forze applicate lungo la trave, nel momento in cui indicheremo la proprietà REPER='LOCAL' ed otteniamo l'orientamento che ci aspetteremo.

E' importante come una volta capito l'orientamento della trave sia possibile operare in *Code-Aster* senza fare errori, ma nel dubbio è consigliabile, prima di effettuare uno studio a tutto campo, quello di eseguire uno studio strettamente locale e limitato ad una porzione dell'elemento.

6 Analisi di elementi strutturali tramite Code-Aster

6.1 Introduzione sui moduli e le analisi effettuate

In questo capitolo tratteremo l'analisi di alcuni casi studio che riguarderanno elementi strutturali semplici; osserveremo il comportamento degli elementi strutturali sotto l'azione di carichi verticali, inoltre valuteremo il comportamento del software *Code-Aster* per la risoluzione dei modelli strutturali e valuteremo le funzioni ed i comandi impiegati per svolgere i casi studio esaminati.

Come prima cosa è necessario descrivere gli ambienti di lavoro all'interno del quale verranno prodotti i file; quando si parla di ambiente di lavoro ci riferiremo ai moduli che ritroviamo all'apertura della piattaforma *Salome-Meca*, all'interno di ogni singolo modulo è possibile svolgere determinate operazioni che consentono di distinguere le fasi di analisi; gli ambienti di lavoro che utilizzeremo saranno pertanto: il modulo *Geometry*, il modulo *Mesh* ed il modulo *Asterstudy* (modulo che si avvale del software FEM *Code-Aster*), questi ambienti di lavoro rappresentano la base per affrontare la modellazione, la discretizzazione e l'analisi degli elementi.

In questi tre ambienti di lavoro verranno eseguite tutte le analisi che precedono la fase di calcolo, in seguito i risultati ottenuti saranno rappresentati tramite il modulo di visualizzazione grafica *Paravis*.

Ogni singolo modulo adoperato dovrà essere del formato corretto (*.med*, *.comm*,...), questo verrà specificato nel seguito, ma soprattutto tramite la piattaforma *Salome-Meca* sarà possibile produrre tutti i file all'interno di un'unica interfaccia. Questo non esclude la possibilità di importare file, immagini *Raster* o elementi da un ambiente esterno, ma l'importazione di un qualunque file sarà possibile solo se questo avrà un formato di lettura supportato dalla piattaforma stessa.

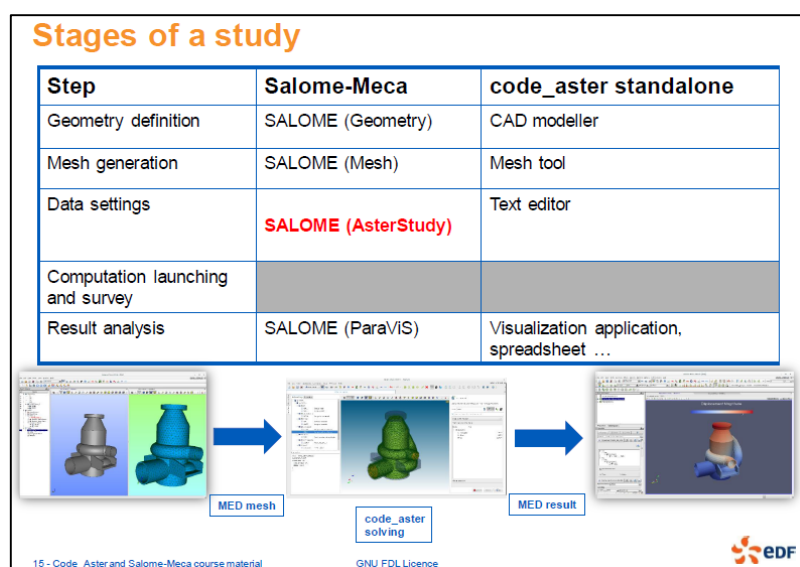


Figura 6.1 Stage di studio

6.2 Avviamento del programma e definizione degli ambienti di lavoro

La macchina virtuale si presenta come una semplice icona sul nostro computer denominata *Virtual-Box*; essa consiste in un applicativo grafico, definito nel gergo comune “lettore CD” o “Macchina Virtuale di lettura”, in grado di poter alloggiare all’interno di una porzione di memoria del nostro computer un *Kernel Linux*. Una volta che il sistema operativo è alloggiato in una porzione di memoria, sarà compito dell’utente definire quanta RAM e memoria grafica dedicare alla macchina che la metterà a disposizione del sistema operativo; la procedura non è difficile da applicare, e sicuramente può essere considerata come la migliore soluzione per far alloggiare all’interno di uno stesso computer due sistemi operativi, infatti in questo modo si generano due percorsi differenti creando due sistemi “*end to end*” differenziati che non entreranno mai in conflitto tra di loro e potranno comunicare tramite una sola cartella di file in condivisione (che consisterà nella nostra cartella condivisa) per traslocare i file da Linux a Windows.

Una volta avviato il pacchetto *Salomè-Meca* andremmo a cliccare immediatamente nel menu a tendina sull'icona *Geometry*, sarà quello il nostro modulo di partenza, ovvero il modulo all’interno del quale verrà generato il modello geometrico CAD.

All’interno di questo ambiente di lavoro troveremo una serie di comandi e di applicazioni utilizzabili, come avviene nella maggior parte dei software CAD, ma ci limiteremo a descrivere i comandi solo quando andremo a richiamarli.

E’ bene ribadire che la geometria che noi andremo a creare sarà costruita passo a passo, ma esiste anche la possibilità di importare altri formati all’interno del modulo *Geometry*; i formati supportati (*.BREP*, *.COM*,...) possono essere un’alternativa veloce per la definizione della geometria del nostro elemento. Una volta aperta la finestra del modulo *Geometry* per poter lavorare con maggiore efficienza e avere una migliore padronanza degli strumenti si consiglia di impostare lo sfondo del modello di colore nero a tinta unita:

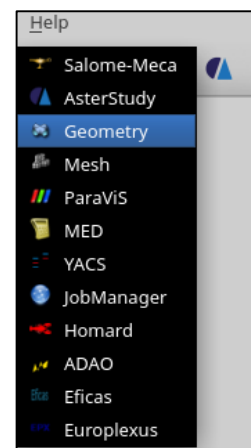


Figura 6.2 Modulo *Geometry*

Tasto destro [mouse] → Change Background → Single color → [nero].

Nel seguito tratteremo una serie di casi studio, e soprattutto per il primo si descriveranno maggiormente nel dettaglio i comandi e le funzioni impiegati per eseguire le operazioni volute; come già mostrato l’accesso ad un qualunque modulo di lavoro avverrà attraverso il menu a tendina prima mostrato. *Code-Aster* in queste prime operazioni non verrà direttamente richiamato poiché ci occuperemo di creare la geometria, ed in seguito la *Mesh*, che servirà poi al software di analisi strutturale.

Si raccomanda di salvare file, cartelle e tutti i lavori relativi al suddetto caso studio in un’apposita cartella senza cambiare mai il nome di nessun file, questo eviterà disagi ed errori in apertura e scrittura dei file; dunque salveremo nella medesima directory tutti i documenti, incluso il file di studio (*.hdf*), il file *Mesh* (*.med*) e i file di output (*.rmed*).

6.3 Mensola soggetta a carichi verticali



Figura 6.3 Mensola incastrata

Andremo ora a studiare un semplice caso strutturale, ovvero una mensola con un estremo libero ed un estremo incastrato soggetta a due carichi distribuiti e ad una forza puntuale in combinazione tra di loro. La mensola in questione sarà lunga 10 metri e sarà costituita interamente da calcestruzzo, materiale che sarà opportunamente indicato all'interno del software di analisi; in questo caso imporrò tutti i coefficienti di sicurezza pari a 1.0:

$$\gamma=1.0$$

$$L= 10.00 \text{ m}$$

$$\Gamma_{CLS}= 25 \text{ kN/m}^3$$

$$NU= 0.3$$

$$E_c= 31447 \text{ MPa}$$

$$Q_{k1}=Q_{k2}= 50 \text{ KPa}$$

$$F= 500 \text{ KN}$$

Di seguito sono rappresentate le differenti combinazioni di carico studiate:

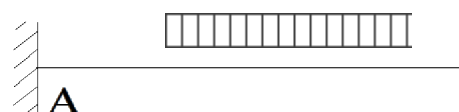
COMBINAZIONE 1

F



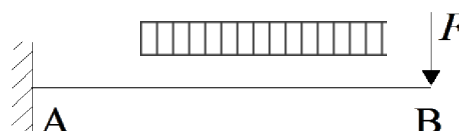
COMBINAZIONE 2

(Q_{k1})

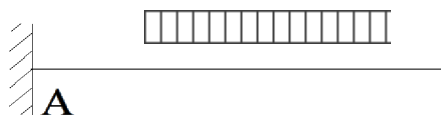


COMBINAZIONE 3

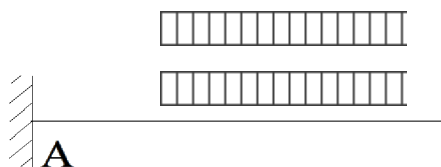
(F+Q_{k1})



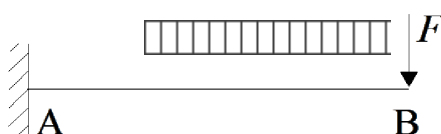
COMBINAZIONE 4
(Qk2)



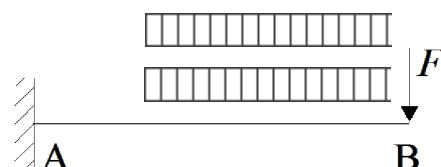
COMBINAZIONE 5
(Qk2+Qk1)



COMBINAZIONE 6
(F+Qk2)



COMBINAZIONE 7
(F+Qk1+Qk2)



6.3.1 Mensola incastrata: generazione della geometria

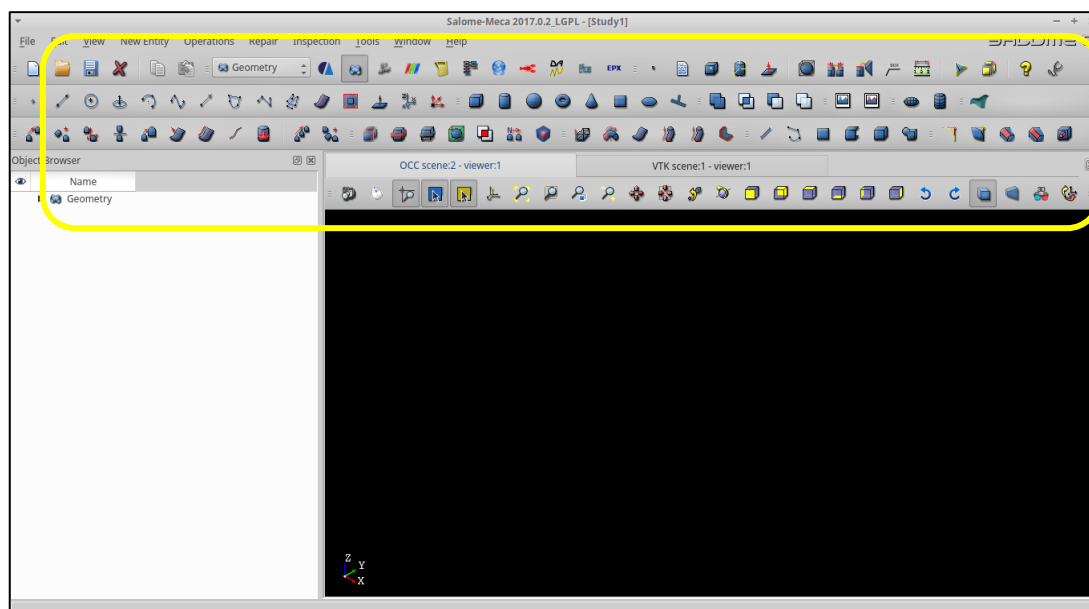


Figura 6.4 Comandi all'interno del modulo Geometry

L'ambiente *Geometry* al suo avvio si presenta come un semplice modulo di generazione della geometria CAD molto comune a software in commercio; nella parte alta

si ritrovano i comandi specifici alla realizzazione del nostro caso studio, mentre lo spazio modello (in nero) sarà l'ambiente ove verrà creata la nostra geometria.

I comandi di ingrandimento, selezione, orientamento dello spazio modello sono posizionati poco sopra lo spazio modello. Mentre sul lato sinistro risiede l'*Object Browser* che sarà la nostra finestra di controllo delle operazioni eseguite.

A questo punto per generare la nostra mensola è necessario indicare i nodi degli elementi; sceglieremo di scrivere le coordinate di ogni singolo nodo, in seguito verrà definito quale dei due sarà l'incastro e quale l'estremo libero caricato dalla forza puntuale. La mensola viene posizionata sul piano XY con asse baricentrico coincidente con l'asse delle X. Definiamo a questo punto le caratteristiche geometriche:

Nodo_1 = (0;0;0) ; Nodo_2 = (10;0;0) ;

Points → (Nome_nodo) → (0;0;0) → “Apply and Close”

A questo punto potremmo trovare gli elementi geometrici appena creati all'interno dell'*Object Browser*; nel caso in seguito si vogliano selezionare o eseguire qualunque operazione sui nodi o sugli oggetti creati, sarà quindi necessario selezionarlo all'interno dell'*Object Browser* e poi impartire i comandi.

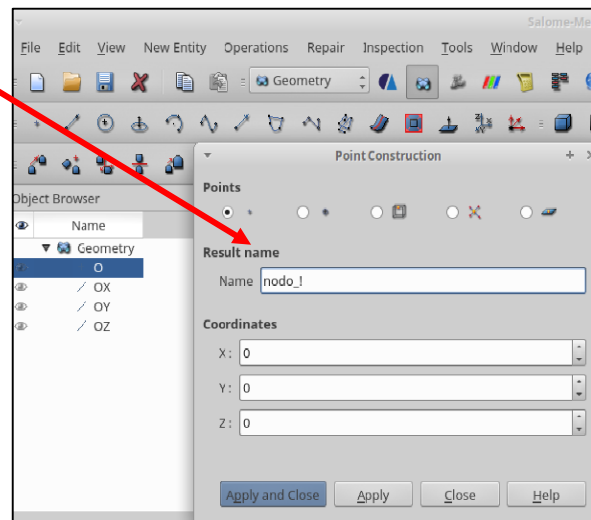


Figura 6.5 Inserimento delle coordinate

Ora generiamo la linea tra i due vertici creati:

Line Costruction → (Nodo_1;Nodo_2) → “Apply and Close”

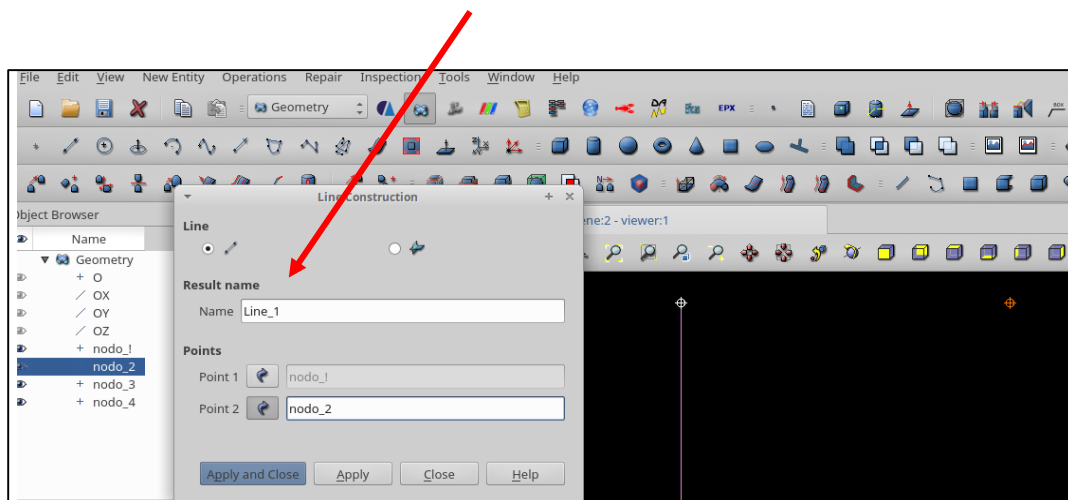


Figura 6.6 Generazione della Linea

Avremo così generato la nostra mensola di 10 metri. Abbiamo scelto volutamente di lavorare in metri e lo faremo per tutti i casi studio, rispettando le unità di misura (*vedi cap.4.3*) e le gerarchie imposte da *Code-Aster*. Ora in “*New Entity*” procediamo a generare

i gruppi. All'interno di tre distinti gruppi inseriremo il nodo vincolato, quello caricato e l'intera trave studiata.

Si seleziona la tipologia di gruppo voluto (punto, linea, ...), si nomina il gruppo, e si seleziona dallo spazio modello il punto voluto; si clicca “*Apply and Close*”, e all'interno dell'*Object Browser* compariranno sotto la voce “Line” i gruppi ad essa associati.

New Entity → Create Group → Shape_Type → (Nome_Gruppo) → (seleziono dall'ambiente) → “Add → Apply”

Eseguo la medesima operazione per ogni singolo nodo e per la linea.

Si otterrà la seguente rappresentazione all'interno dell'*Object Browser*.

A questo punto avremmo nella seguente immagine la nostra mensola; per semplicità abbiamo aumentato lo spessore della linea per facilitarne la visualizzazione tramite i comandi:

(Tasto destro) → Edge Width → “5”

Come si può osservare la trave risulta dunque essere sul piano XY, il motivo per cui abbiamo generato gruppi che potrebbero non servire per la nostra analisi, risiede nel fatto che possono sempre essere utilizzati per qualunque ragione in seguito, infatti è possibile applicare un carico puntuale, un carico distribuito, o un vincolo aggiuntivo, e la presenza dei gruppi ne faciliterebbe il processo; si potrebbe comunque tornare indietro tra un modulo di lavoro e l'altro, ma si appesantirebbe troppo il lavoro per il processore rischiando di avere un “*fatal error*” dal software.

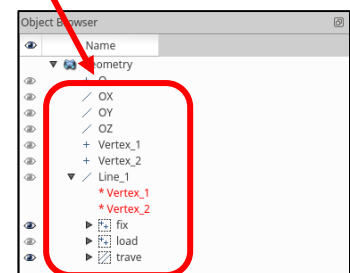
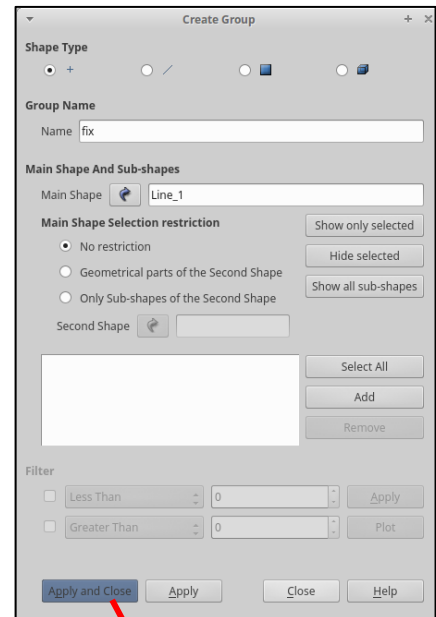


Figura 6.7
Creazione dei gruppi

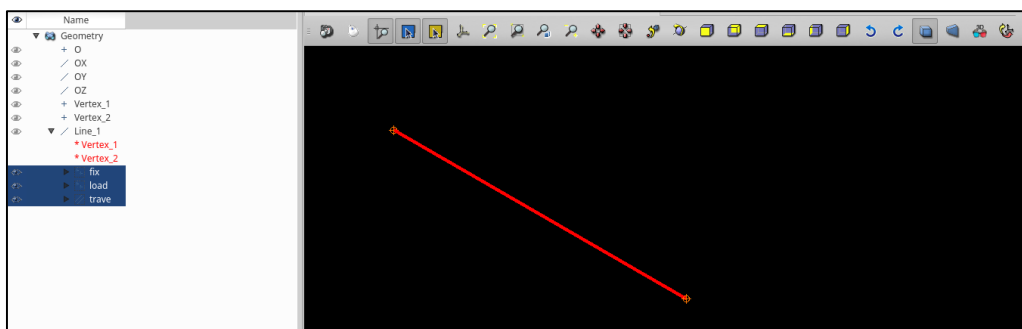
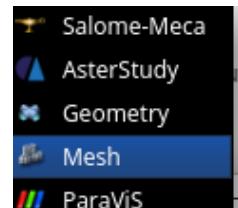


Figura 6.8 Mensola

Nell'immagine si possono anche osservare al di sopra dello spazio modello i comandi necessari a gestire la geometria, utili per ingrandire, selezionare o per orientare la mensola. Come si può vedere selezionando un tratto qualunque della trave apparirà tutta evidenziata, segno di aver creato continuità tra gli elementi.

Definita la Geometria ora ci occupiamo di discretizzare il nostro elemento strutturale eseguendo il calcolo della *Mesh*. Per fare questo clicchiamo dal menu a tendina il modulo *Mesh*.



6.3.2 Mensola incastrata: generazione della Mesh

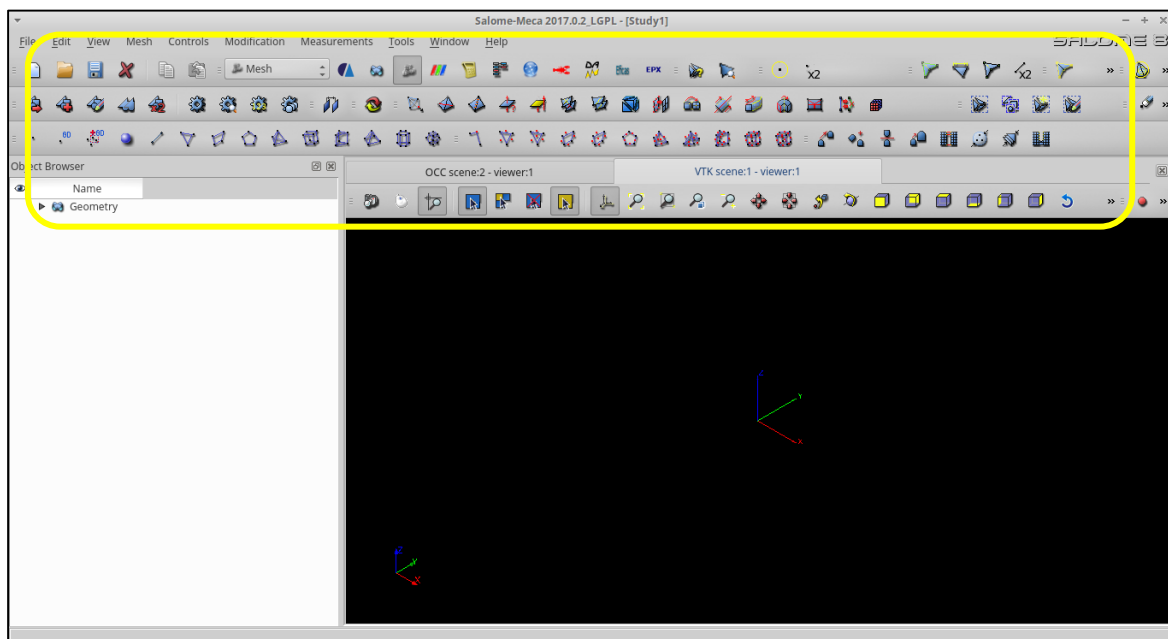


Figura 6.9 Modulo Mesh

L'immagine dell'ambiente di lavoro è del tutto simile all'ambiente *Geometry*, non vi sono quindi molte differenze sulle prime operazioni da eseguire, infatti come prima operazione si consiglia di cambiare lo sfondo dello schermo allo stesso modo di come è stato fatto in precedenza.

Come si può osservare vi sarà sempre una parte sulla sinistra dello schermo in cui si trova l'*Object Browser*, mentre nella parte alta vi sono tutti comandi necessari alla creazione della *Mesh* ed ella manipolazione della stessa.

L'obiettivo che ci poniamo all'interno del seguente modulo sarà quello di discretizzare l'elemento creato generando elementi lineari e punti che andranno a formare la nostra *Mesh* sul quale eseguiremo i calcoli

Come già spiegato (vedi capitolo 4.8.1) la creazione della *Mesh* viene generata attraverso l'analisi dell'elemento geometrico, precedentemente definito, tramite funzioni di discretizzazione contenute all'interno del modulo. Dunque sarà necessario stabilire il

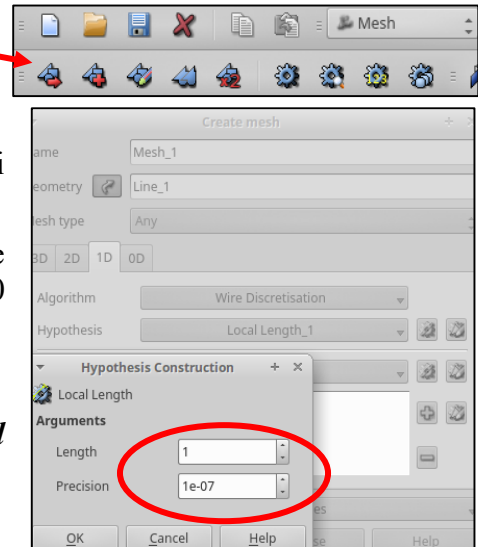
tipo ed il metodo di discretizzazione da effettuare e capire se questa possa essere sufficiente o debba essere affinata.

Una volta aperto il modulo *Mesh*, cliccando sul comando “*Create Mesh*”.

E nella finestra successiva dopo aver definito l’elemento geometrico ed il nome si definiscono i parametri della *Mesh*.

In questo caso eseguiamo una *Mesh* che suddividerà l’elemento in 10 parti uguali, ovvero 10 elementi di lunghezza 1 metro.

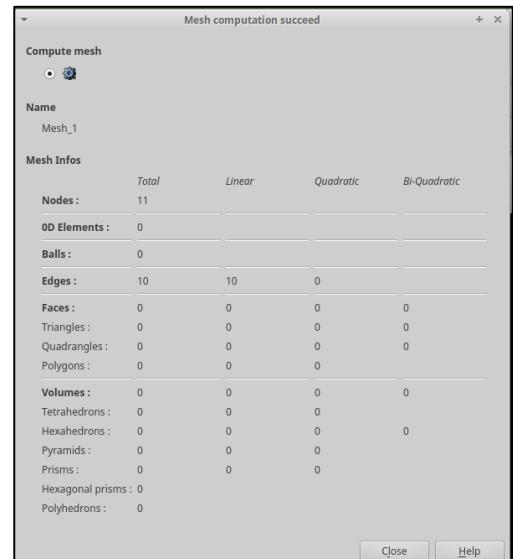
Create New Mesh → “wire discretisation → “Local Length” → “1” → OK



Dopo si clicca sul tasto “*Compute*” per effettuare il calcolo della *Mesh* e verrà proiettata la finestra seguente che rappresenta il numero dei nodi e di elementi creati. Come si può vedere avendo eseguito l’analisi su un elemento lineare senza curvature, senza elementi a due dimensioni la discretizzazione sarà molto semplice. Questa discretizzazione viene eseguita come esempio per far capire il funzionamento del software.

A questo punto per le medesime ragioni che ci hanno portato a generare i gruppi all’interno del modulo *Geometry*, importiamo i gruppi dal medesimo modulo al modulo *Mesh* per avere dunque gruppi di elementi su cui poter eseguire le nostre analisi.

Mesh_1 → (tasto destro) → Create Group from Geometry → “Apply and Close”



Una volta definiti i gruppi anch’essi si potranno ritrovare nell’Object Browser della *Mesh* sulla sinistra dello schermo. Come si vede noi avremmo 10 elementi con 11 nodi per la nostra trave di 10 metri.

La *Mesh* ovviamente potrà essere ancora infittita e ricalcolata con grado discretizzazione maggiore, a questo punto si esegue il salvataggio della *Mesh* e si esporta il file .med all’esterno dell’ambiente per sicurezza.

L’immagine successiva definisce esattamente i nodi che sono stati realizzati e i gruppi all’interno della *Mesh*; ovviamente i gruppi saranno formati dai nodi appartenenti alla

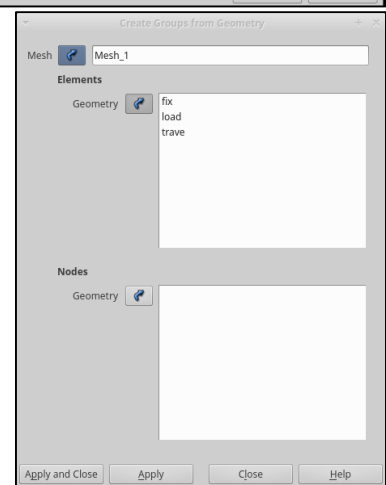


Figura 6.10
Generazione della Mesh

Mesh nelle entità definite tramite la geometria in precedenza; quindi il nodo “*fix*” sarà il primo nodo da sinistra della *Mesh*.

A questo punto si esegue un controllo su eventuali nodi ed elementi doppi/orfani cliccando con il tasto destro “*node controls*”, inoltre per poter visualizzare nodi ed elementi clicchiamo tasto destro sul modello e “*show nodes*”; nell’immagine che segue viene mostrata la mensola discretizzata.

A questo punto se non vi sono correzzioni da fare si può procedere con il modulo *Asterstudy*.

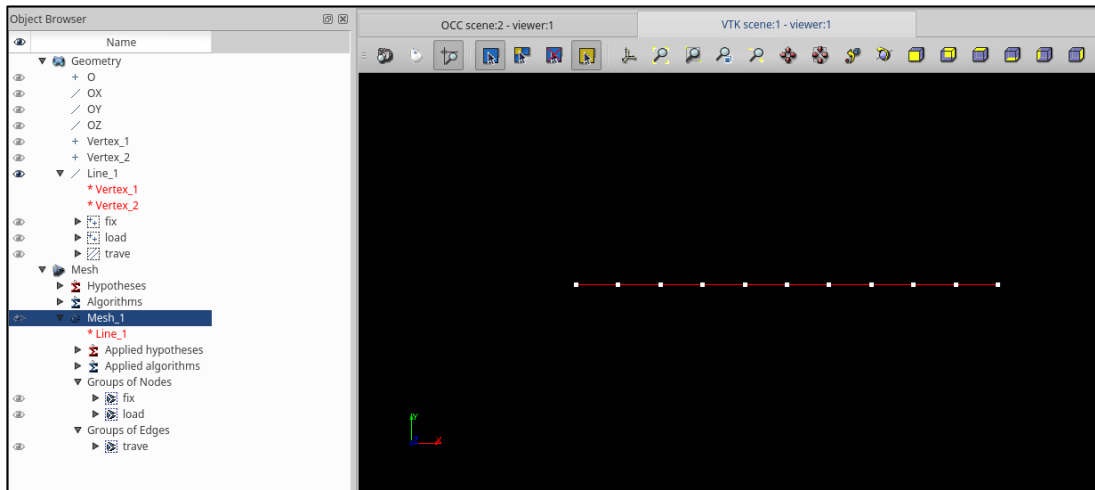


Figura 6.11 Mensola discretizzata

6.3.3 Creazione del Command file tramite il modulo Asterstudy

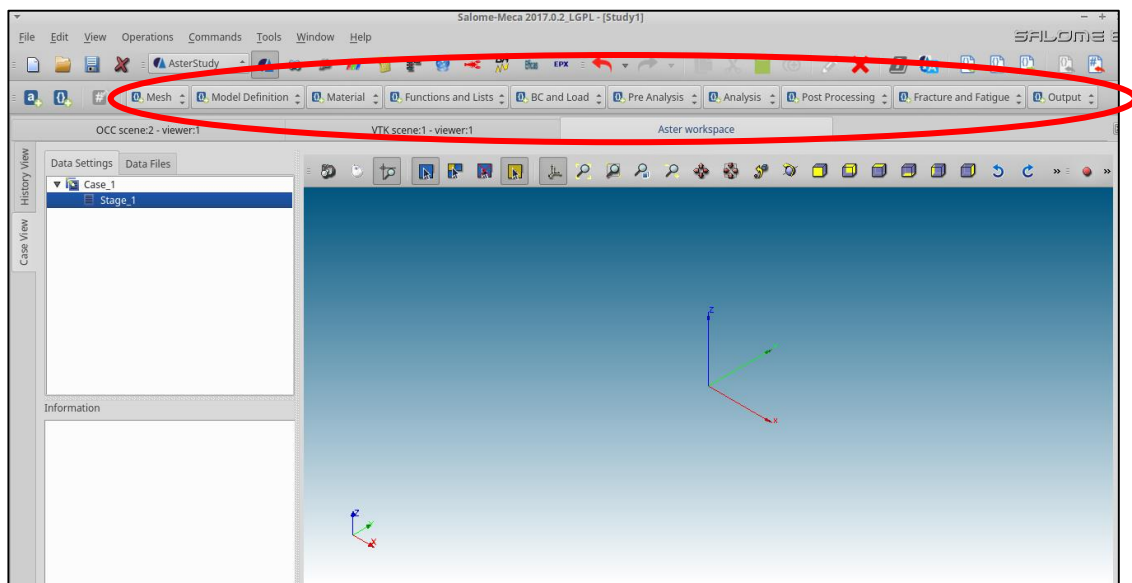


Figura 6.12 Il modulo Asterstudy

Come si può osservare l'ambiente del modulo *Asterstudy* sarà molto differente dai precedenti moduli, infatti si può osservare che nella parte alta dello schermo vi saranno tutte le icone relativi ad operazioni semplici (copia incolla, cancella, ...), ma subito sotto questi si osservano i menu a tendina che rappresentano le funzioni da richiamare all'interno del "*Command file*" per poter eseguire i calcoli. Mentre sulla sinistra vi sono le finestre delle informazioni ed il "*Case View*" all'interno del quale saranno rappresentate tutte le funzioni richiamate da noi durante l'esecuzione.

Sarà in questo modulo che verranno definiti carichi condizioni al contorno, orientamento e tutte le caratteristiche necessarie ad eseguire le analisi sulla mensola; tutte le caratteristiche verranno definite seguendo l'ordine dei menu a tendina (cerchiati), quindi si eseguiranno le funzioni definite che si vorranno e le si potranno gestire tramite il *ToolBox* sulla destra dello schermo. La compilazione del *ToolBox* genera un file (.comm) che è possibile generare anche manualmente tramite un semplice file (.txt) adoperando le regole di invocazione dei comandi e delle funzioni di *Code-Aster*, è possibile inoltre importarlo dall'esterno oppure scriverlo direttamente in alternativa al metodo grafico.

Tutto questo è possibile cliccando su "*Add Stage*" per iniziare a definire le funzioni, e cliccando con il tasto destro "*text mode*". La generazione del file è comunque indifferente che si faccia in un modo o nell'altro, l'importante è che vi sia il rispetto di tutte le regole di compilazione del file.

"Stage_1" → [tasto destro mouse] → text Mode → edit command (scrittura diretta).

N.B. *Le funzioni maggiormente impiegate nella risoluzione dei problemi vengono espresse dagli sviluppatori in carattere minuscolo rispetto alle altre.*

Ci si adopera a completare il *ToolBox* di destra inserendo tutti i dati che comporranno le funzioni finali; per ogni dato inserito, cliccando su "*Apply*" verrà riaggiornata la casella "*information*" con le informazioni inserite; per quanto riguarda il box *Data Setting* questo verrà aggiornato per ogni funzione inserita.

Una volta iniziato lo Stage si procede a definire le variabili, ed eventuali commenti che possano essere utili allo svolgimento del problema.

La definizione delle variabili si basa sostanzialmente per il nostro caso nella definizione dei coefficienti di sicurezza e dei coefficienti di combinazione definite all'interno delle NTC 2018, vengono quindi elencati tutti i valori definiti nella normativa in base al tipo di categoria appartenente; oltre a questi è possibile anche inserire eventuali altre variabili che possano servire nel proseguo dell'esercizio richiamandole all'interno del file .comm, e noi in questo caso inseriremo il modulo elastico del calcestruzzo calcolato in riferimento ad una classe C25/30, oltre al modulo elastico vengono definiti i carichi che agiranno sulla trave in esame, ovvero due carichi variabili e una forza puntuale; tutto questo viene doverosamente fatto all'inizio del file in quanto questo file potrà essere riutilizzato per ulteriori analisi tramite semplice modifica dei parametri appena descritti.

La formula per calcolare il modulo elastico medio del calcestruzzo sarà:

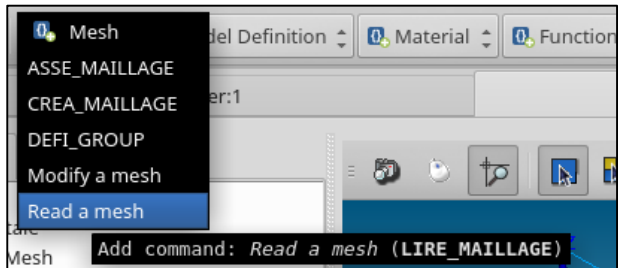
$$E=(\text{pow}((f_{ck}+8/10);0.3))*22000000000 \text{ [N]};$$

Viene inoltre inserita una formula che spiegheremo nel seguito e che ci consentirà di effettuare le analisi sulla nostra mensola utilizzando due carichi variabili in quanto alcune limitazioni di *Code-Aster* purtroppo hanno impedito una corretta analisi.

Per una più chiara esposizione si rimanda al file *.comm* in allegato (vedi Cap.7)

- Si comincia con l'analisi andando nel menu a tendina: **Mesh → Read mesh**

LIRE_MAILLAGE: in questa fase ci si occupa di leggere la *Mesh* generata e di definirne la posizione, il formato e le informazioni di base. Questa funzione permette di acquisire le informazioni di tutti gli elementi e i nodi appartenenti alla *Mesh*.



MESH FILE LOCATION: Viene inserito il percorso del file per caricare il file *Mesh* generato in precedenza e salvato nella medesima *Directory*.

FORMAT: Viene specificato il formato della *Mesh*: [*MED* o *ASTER*]; il formato *Med* risulta il maggiormente utilizzato in quanto compatibile con altre piattaforme, a differenza del formato *ASTER* che risulta rilegato alla piattaforma madre.

VERI_MAIL: se la casella viene spuntata sul "SI" viene effettuata una check sui valori delle *Mesh*, in caso di valori doppi, o valori errati viene riportato un file di errore.

INFORMATION: [1 o 2] vengono definite le informazioni di default (1) oppure (2) maggiormente dettagliate della *Mesh* in cui vengono specificate le coordinate di tutti i nodi, le dimensioni, il tipo di nodo, il tipo di *Mesh*,...

NOM_MED: Nel caso vi siano più di una *Mesh* è possibile indicare il nome della *Mesh* che si vuole invocare; la funzione risulta utile nel caso vi siano due o più *Mesh* confinanti, in caso contrario non risulta necessario.

INFO_MED: nel caso di più *Mesh* vengono stampate le informazioni aggiuntive che legano la *Mesh* selezionata e le altre *Mesh* non selezionate, ma presenti all'interno del file; **1:** non vi sono riferimenti con le seconde/terze/... *Mesh*; **2:** solo le informazioni relativi a gruppi ed elementi in comune; **3:** tutte le informazioni vengono stampate. In questo modulo indicheremo il nome della *Mesh*, il formato e faremo un check di controllo, le altre opzioni possono essere lasciate impostate di default.

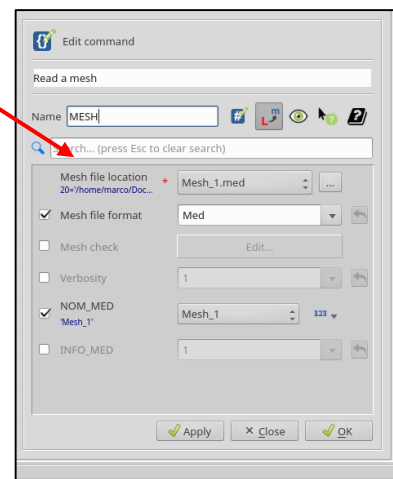


Figura 6.13 Lettura della mesh

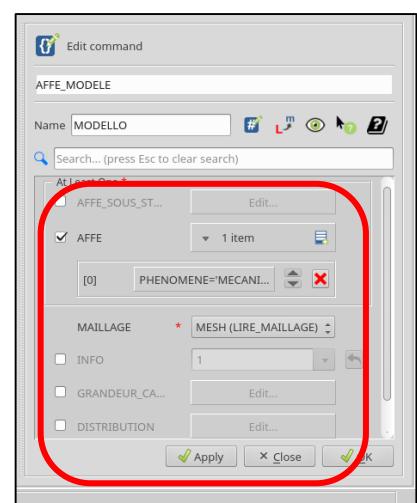


Figura 6.14 Definizione del modello

- Si apre ora il seguente modulo: **model definition** → **assign finite element**

AFFE_MODELE: In questa fase ci si occupa di assegnare il corretto modello alla nostra *Mesh*, oppure ai nostri elementi dichiarati, in base al tipo di materiale che si è scelto di studiare facendone corrispondere un preciso modello correlabile con il comportamento del materiale.

Per la nostra mensola definiremo un modello meccanico legato alla teoria di Eulero-Bernoulli in cui l'azione delle forze di taglio sulla rotazione delle sezioni trasversali viene trascurata.

AFFE_SOUS_STRUCT/AFFE_MODELE: si assegna la condizione ad una sotto-struttura/struttura → si sceglie se assegnare la condizione a tutta la sub-struttura/struttura ["TOUT"], oppure a determinati elementi della *Mesh* che devono essere stati preventivamente indicati all'interno della *Mesh* [GROUP_MA] → si sceglie il tipo di fenomeno [MECANIQUE] → si sceglie il tipo di modellizzazione associato al tipo di discretizzazione [POU_D_E].

Nella denominazione i primi tre caratteri POU stanno per "*Poutre*", ('trave' in francese); il quinto carattere determina il supporto: D rappresenta '*Droite*', (disteso lungo una linea retta), E rappresenta la teoria di Eulero-Bernoulli.

MAILLAGE: rappresenta la nostra *Mesh*, viene rappresentata di Default.

INFO: [1 o 2] vengono definite le informazioni di default (1) oppure (2) maggiormente dettagliate della *Mesh* in cui vengono specificate le coordinate di tutti i nodi, le dimensioni, il tipo di nodo, il tipo di *Mesh*,... a cui viene assegnato il modello.

GRANDEUR_CAR: valore legato all'errore ricavabile da una grandezza fisica (longeur, pression, temperature).

DISTRIBUTION: consente di distribuire il modello FEM tramite funzioni parallele e di controllarle nella loro elaborazione.

VERY_JACOBIEN: operatore che verifica la bontà del modello e che non vi siano troppe distorsioni tra la *Mesh* e il modello medesimo.

- Si rimane sempre nel medesimo modulo e si apre: **model definition** → **Affe_cara_elem**

AFFE_CARA_ELEM: All'interno di questa funzione si definiscono le caratteristiche geometriche della sezione della nostra mensola indicando le dimensioni e le caratteristiche rispetto agli assi di riferimento.

POUTRE: Si seleziona tra le opzioni possibili il tipo di struttura, e quindi *Poutre* (=trave), ed al suo definiamo le caratteristiche della sezione; si ricorda che nel caso non si conoscessero le caratteristiche della sezione è possibile calcolarle, per sezioni generiche, attraverso la funzione MACR_CARA_POUTRE che calcola tutte le caratteristiche della sezione a seconda dell'orientamento degli assi.

SECTION: Definiamo la sezione rettangolare della nostra mensola.

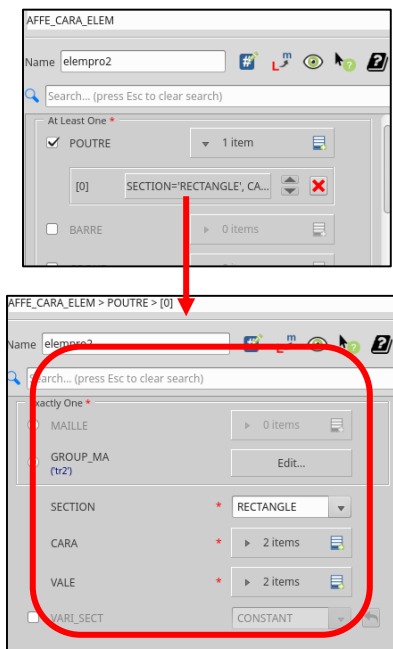


Figura 6.15 Dimensioni della sezione

GROUP_MA: Definiamo la *Mesh* a cui appartiene la sezione, nel nostro caso la *Mesh* di tutta la mensola.

CARA: Definiamo gli elementi geometrici che conosciamo, nel nostro caso conosciamo le dimensioni della sezione (HY; HZ); qualunque sia l'orientamento dell'elemento strutturale si dovranno considerare gli assi locali che giacciono sulla sezione della trave.

VALE: Si definiscono i valori delle grandezze della sezione (0.4 ; 0.4) m.

Si procede con l'analisi: **Material** → **Define materials**

DEFI_MATERIAU: in questa fase è necessario definire il tipo di materiale che verrà associato alla nostra *Mesh*. Come si può già osservare nell'immagine l'elenco dei materiali a nostra scelta sono parecchi, dai non lineari ai fluidi, dai materiali rocciosi ai comportamenti pseudo-elastici, di conseguenza è necessario conoscere il materiale e le relative proprietà meccaniche/fisiche/chimiche per poter scegliere la corretta opzione che descriva al meglio il comportamento del materiale e le sue specificità.

Tra le varie scelte disponibili si opta per “ELAS” materiale con caratteristiche elastiche-lineari isotropiche. Ovvero a cui faremo corrispondere il calcestruzzo.

Definiamo le caratteristiche del materiale elastico-lineare, ovvero il modulo di Young ed il coefficiente di Poisson per il calcestruzzo ($\nu=0.3$); in entrambi i casi gli inserimenti sono avvenuti mediante le variabili definite in precedenza.

Oltre ai valori indicati è possibile anche definire i seguenti valori:

RHO: valore relativo alla massa volumica degli elementi (2500 kg/m^3)

ALPHA: coefficiente di dilatazione termica isotropica valutato sulla base di una temperatura media.

AMOR_ALPHA: coefficiente legato alla costruzione della matrice di rigidezza.

AMOR_BETA: coefficiente legato alla costruzione della matrice delle masse.

AMOR_HYST: coefficiente isteretico legato ad un materiale visco-elastico per generare una matrice complessa di rigidezza all'interno di un'analisi di oscillazione armonica.

LONG_CARA: Coefficiente corrispondente ad una dimensione caratteristica L associato ai coefficienti di Lamé; questo coefficiente interviene nell'espressione del termine aggiunto del vettore forza integrato lungo il confine del campo degli elementi finiti, che corrisponde alle rigidità distribuite lungo questo confine del campo degli elementi finiti.

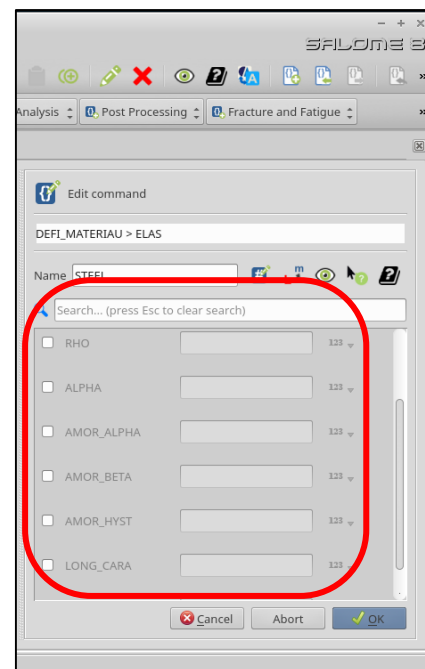


Figura 6.16 Definizione del materiale

COEF_AMOR: Coefficiente che interviene nell'espressione del vettore forza integrato lungo il bordo del campo degli elementi finiti e che corrisponde agli ammortizzatori viscosi distribuiti lungo questo confine.

- Si assegna ora il materiale appena definito: **Material** → **Assign a material**

AFFE_MATERIAU: In questa sezione ci si occupa di assegnare il materiale al nostro modello definito.

In seguito si definisce a quale porzione di struttura (modello e/o *Mesh*) verrà fatta l'assegnazione:

Assign a material → **"Everywhere"** → **[yes]** → **Material** → **"mater"**.

Le opzioni successive sono rivolte al comportamento che potrebbe essere assegnato al materiale stesso: **"Behavior assignement"** ovvero comportamento del materiale o di alcune componenti di materiale, ad esempio è possibile assegnare un comportamento non lineare ad alcune componenti del

modello/*Mesh*; **"External State variable assignement"** ovvero una eventuale variabile di stato esterna (correlata a temperatura, elettricità, idraulica,...); **"Verbosity"** ovvero ulteriori informazioni riguardo al materiale assegnato. Noi decidiamo di fermarci nell'indicare solamente il comportamento del materiale senza entrare in ulteriori dettagli.

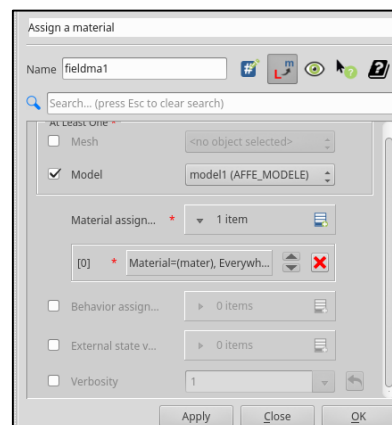


Figura 6.17 Assegnazione del materiale

- Si procede ora con l'analisi selezionando dal menu a tendina: **Function and List** → **Define Function**

DEFI_FUNCTION: La suddetta funzione risulterà essere molto importante per quanto concerne la generazione delle combinazioni di carico; infatti all'interno di tale funzione definiremo alcune istanti di analisi temporale relazionate rispettivamente ad ogni singolo carico applicato sulla struttura che all'innescarsi di un determinato istante moltiplicherà i parametri che imporranno in input per dei coefficienti (ovvero saranno i nostri coefficienti di sicurezza dichiarati in apertura del file).

VALE: All'interno di questo valore vengono rappresentati i valori degli istanti temporali ed i valori dei coefficienti corrispondenti; tali valori dei coefficienti costituiranno i valori della funzione ad ogni singolo intervallo temporale. Nel seguito sono rappresentati i valori della forza e dei due carichi distribuiti; come si può vedere avremmo 7 combinazioni di carico ed ogni combinazione imporranno i coefficienti di sicurezza (γ_p) pari a 1:

- COMBINAZIONE 1: (F) applicato;
- COMBINAZIONE 2: (Qk1) applicato;
- COMBINAZIONE 3: (F + Qk1) applicato;
- COMBINAZIONE 4: (Qk2) applicato;
- COMBINAZIONE 5: (Qk1 + Qk2) applicato;

COMBINAZIONE 6: (F + Qk2) applicato;

COMBINAZIONE 7: (F + Qk1 + Qk2) applicato;

	Time	Function
1	1.0	1.0
2	2.0	0.0
3	3.0	1.0
4	4.0	0.0
5	5.0	0.0
6	6.0	1.0
7	7.0	1.0

	Time	Function
1	1.0	0.0
2	2.0	1.0
3	3.0	1.0
4	4.0	0.0
5	5.0	1.0
6	6.0	0.0
7	7.0	1.0

	Time	Function
1	1.0	0.0
2	2.0	0.0
3	3.0	0.0
4	4.0	1.0
5	5.0	1.0
6	6.0	1.0
7	7.0	1.0

NOM_PARA: All'interno della seguente istruzione definiamo il valore che abbiamo definito essere assunto dalla funzione, ed nel nostro caso è un istante INST.

INTERPOL: Nella casella seguente viene definito il tipo di interpolazione assunta dalla funzione, che nel nostro caso sarà "LIN" cioè lineare.

PROL_DROITE/PROLE_GAUCHE: Questa proprietà definisce il valore della funzione a destra e a sinistra, ovvero prima e dopo, degli istanti temporali che abbiamo precedentemente definito.

Prol_Droite/Gauche → Constant

TITRE: Infine daremo un titolo.

Figura 6.18 Definizione dei Time-Step di analisi

- Si procede sempre all'interno del medesimo menu a tendina: **Define Function**

DEFI_LIST_REEL: All'interno di questo comando viene definito il passo di analisi della funzione, ovvero se si deve considerare tutta, parte o se non si deve considerare per determinati valori prefissati.

DEBUT: Viene inserito il valore di inizio degli istanti temporali.

INTERVALLE: All'interno di questa funzione vengono definiti il valore massimo di dove far arrivare il termine temporale ed anche il passo da assumere, il passo di avanzamento.

TITRE: Infine si definisce il titolo

Figura 6.19 Definizione del passo temporale

- Si avvia ora il modulo di assegnazione di forze e condizioni al contorno: **BC and LOADS → Assign mechanical load**

AFFE_CHAR_MECA: All'interno di questo modulo vengono richiamati tutti gli elementi a cui assegnare eventuali forze, carichi, coppie o accelerazioni ed agli stessi possono essere anche assegnati valori inerenti ai vincoli o condizioni al contorno, e vi sono numerose combinazioni di forze/carichi e condizioni vincolari da poter applicare, ma per la suddetta analisi ci limiteremo ad eseguire le combinazioni di carico tra i due carichi distribuiti variabili la forza puntuale applicata all'estremo libero.

DDL_IMPO: La dicitura sta per vincoli imposti; sostanzialmente all'interno della suddetta funzione indicheremo i vincoli che interesseranno la nostra mensola, ovvero al nodo incastro bloccheremo tutte le rotazioni e gli spostamenti.

DDL_impo → Group_no → ("Fix") → "Ecastre"

FORCE_NODALE: Faremo la stessa cosa ma questa volta si tratterà della forza nodale di 500 kN applicata all'estremo libero.

FORCE_POUTRE: Invece all'interno della suddetta funzione si devono inserire i carichi variabili che abbiamo assegnato alla struttura e che sono di 50 KPa, nell'assegnazione andremo ad indicare il tratto di *Mesh* che avevamo definito in precedenza e che comprendeva tutta la trave nella sua lunghezza.

- Si avvia ora dal menu a tendina: **Analysis → Static Mechanical Analysis**

MECA_STATIQUE: Questo comando ha molta importanza all'interno di *Code-Aster*, infatti grazie a questo comando è possibile definire il tipo di analisi da effettuare. Infatti si può definire all'interno del suddetto comando come debbano essere considerati i carichi e i vincoli nell'elemento strutturale.

CHAM_MATER: Viene richiamato il tipo di materiale precedentemente definito che interviene nel computo di analisi.

CARA_ELEM: Viene richiamato le caratteristiche della sezione trasversale definite in precedenza.

MODELE: Si indica il tipo di modello definito in precedenza (elastico lineare).

EXCIT: Vengono ora definiti i tipi di vincoli e carichi agenti sulla struttura andando ad elencarli tutti quanti; all'interno di ogni singola opzione selezionata si definisce il tipo di carico (variabile, forza, ...) e gli si associa la funzione definita in "*Function and List*"; così

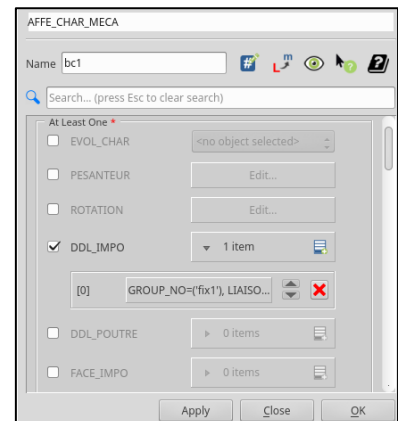


Figura 6.20 Definizione dei carichi e dei vincoli

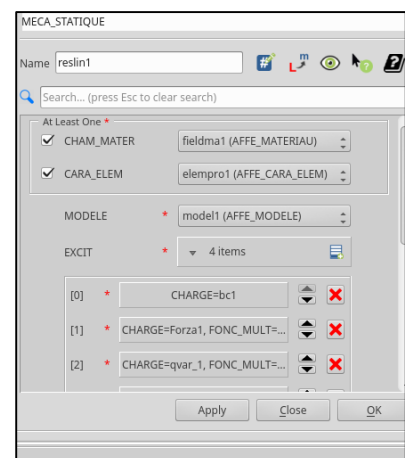


Figura 6.21 Definizione del metodo di analisi

facendo stiamo creando un'analisi meccanica con forze e vincoli applicati in determinati istanti temporali.

ORIENTATION: Si definiscono ora gli assi di riferimento che orienteranno la trave per poter effettuare il calcolo delle caratteristiche della sollecitazione; all'interno di questa funzione si definisce la *Mesh* dell'elemento strutturale da orientare, il metodo di orientamento VECT_Y (posiziona l'asse Y locale sul sistema di riferimento Globale) e si indicano i valori (0;0;1) in questo valuteremo il momento flettente MFZ e il taglio VY.

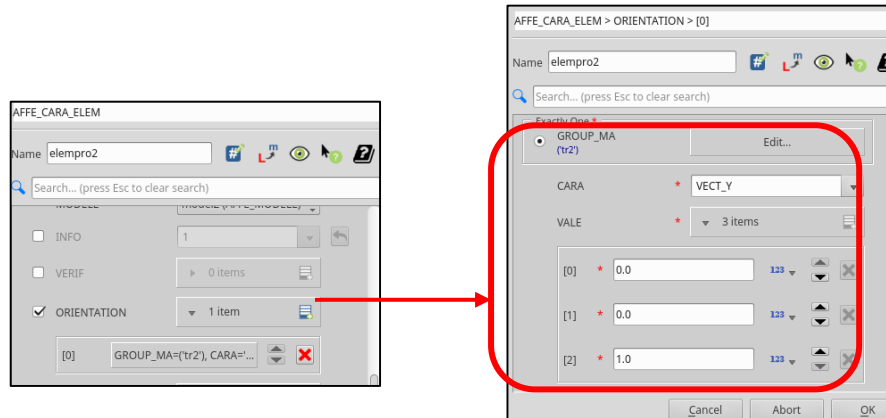


Figura 6.22
Orientamento della trave

- Si procede ora ad attivare dal menu a tendina: **Post-Processing → Calc_Champ**

CALC_CHAMP: Questa funzione viene impiegata nelle analisi per arricchire ulteriormente di informazioni i risultati; infatti eseguendo un'analisi generica su una struttura si invoca (di default) il campo dei risultati SIEF_R (campo di output Reale di Σ e Effort, tensioni e sollecitazioni) all'interno di questo campo dunque vi sono parecchi valori che possono essere richiamati tramite una funzione, ed utilizzando CALC_CHAMP si può fare.

Si osserva nella figura a fianco che vengono richiamati MODELE (il modello), AFFE_CARA (caratteristiche geometriche) e RESULT (il tipo di analisi effettuata che produce un determinato risultato).

CONTRAINTE: Il tipo di risultato con cui io voglio arricchire la mia analisi, nel nostro caso selezioniamo EFGE_NOEU, EFGE_ELGA ed EFGE_ELNO (rispettivamente gli sforzi generalizzati nei nodi, nei punti di Gauss e nei nodi degli elementi), calcoliamo inoltre in un'altra funzione di CALC_CHAMP anche le tensioni massime prodotte sulla sezione trasversale ricavate dualmente alle caratteristiche di sollecitazione saranno SIPM_ELNO, SIPO_ELNO e SIPO_NOEU (rispettivamente le tensioni della trave massime/minime sulla sezione, le tensioni sulla trave medie calcolate ai nodi degli elementi e le tensioni sulla trave medie calcolate ai nodi). (vedi capitolo 4.7.1)

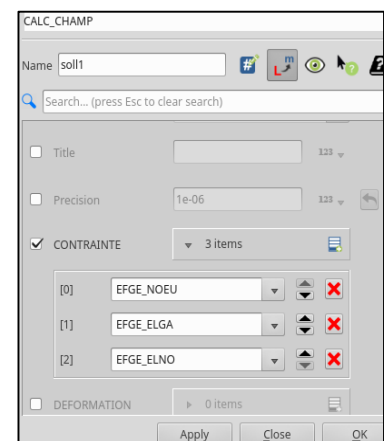
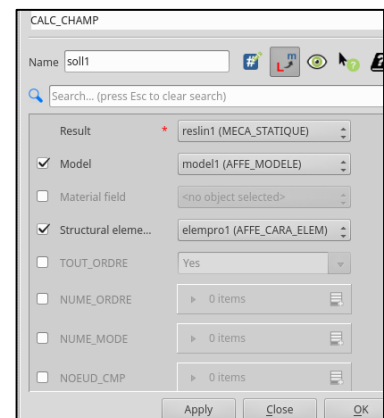


Figura 6.23 Funzione
Calc_Champ

LISTR: definisco il richiamo alla funzione per impostare il calcolo tramite i time-step.

- Ora si accede dal menu a tendina alla funzione: **Output→ Setup Output Results**

IMPR_RESU: All'interno del suddetto file si andranno a definire le caratteristiche di stampa di un risultato, ovvero devo definire le caratteristiche del file di output.

Nell'immagine si osserva che viene definito il nome e la directory, ricordando che il file per essere letto in *Paravis* dovrà essere del formato *.rmed*.

RESU: In questa sezione si richiamano solo gli elementi da stampare quindi si definiscono al suo interno: la *Mesh*, il modello, il tipo di analisi (standard o arricchita) e l'eventuale uso del time-step. All'interno di questa sezione è possibile anche raffinare il risultato su un nodo, un elemento o una fibra oppure si possono stampare in output solamente alcune componenti specifiche. Si fa notare che i file di output in cui andiamo a richiamare le analisi ed campi voluti, vengono creati separatamente in modo da non creare conflitti interni al software.

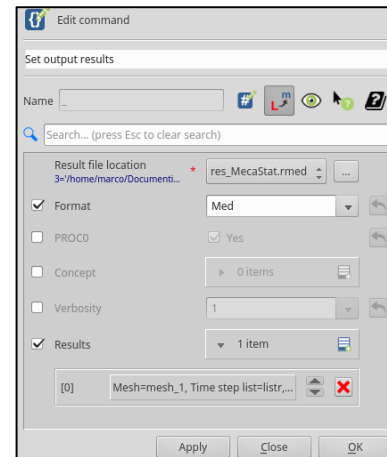


Figura 6.24 Definizione dell'output

Adesso ci occupiamo del passaggio cruciale infatti una volta definito il nostro “*Command File*” è necessario verificare che esso sia corretto e farlo calcolare dal software di calcolo. Si clicca l'icona sulla sinistra del ToolBox “*History view*” e si avrà la presente schermata:

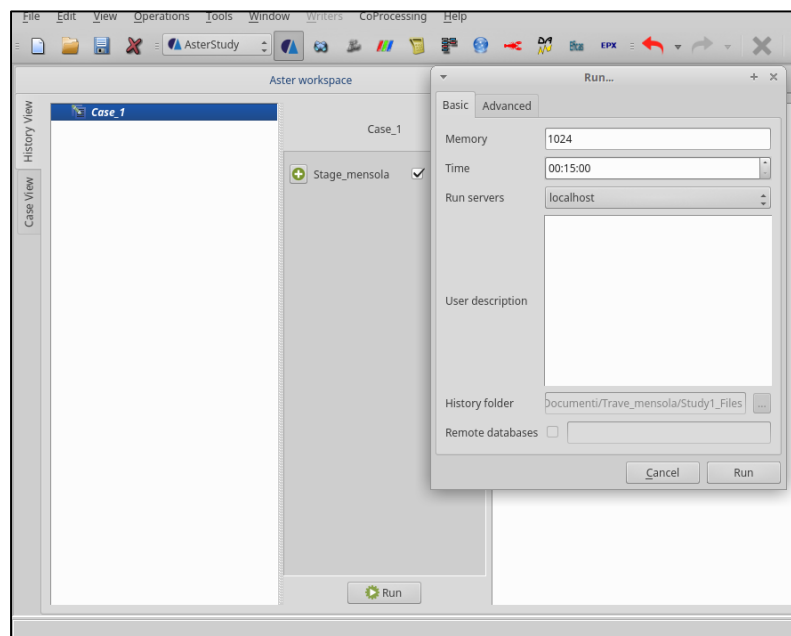


Figura 6.25 Avvio del calcolo di Code-Aster

Cliccando su “Run” e di nuovo “Run” sulla successiva finestra il calcolo inizierà, il risultato potrà essere negativo (ROSSO), positivo con Riserva ed in tal caso dovremmo fare alcune correzioni (ARANCIONE) oppure positivo (VERDE):

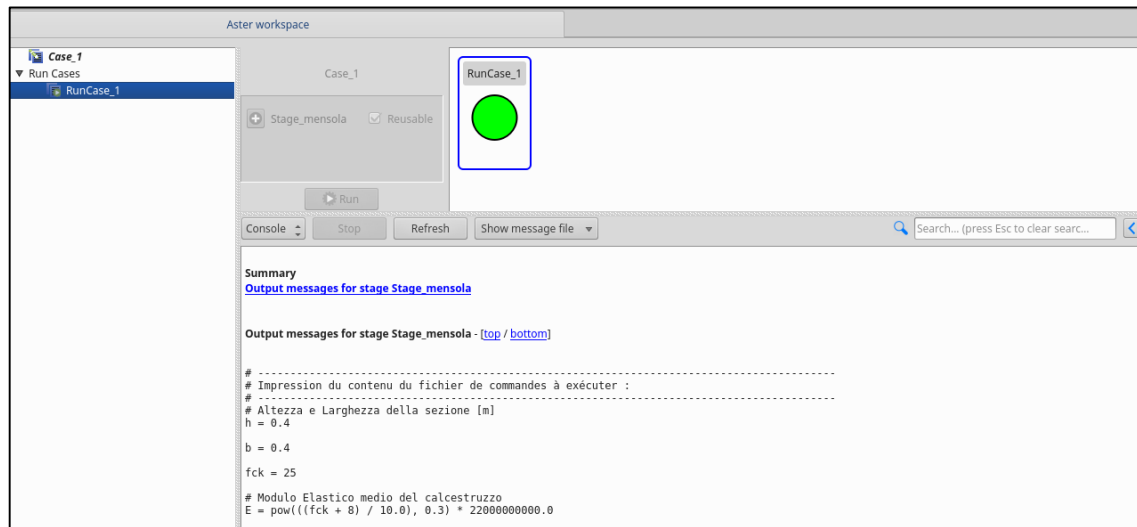


Figura 6.26 Esito del calcolo effettuato

Nel nostro caso il risultato è positivo dunque L’analisi si è conclusa e si procede nel successivo modulo di visualizzazione grafica *Paravis*.

Con questa ultima operazione abbiamo concluso l’analisi tramite il modulo *Asterstudy* (Si allega il “Coomand File” nel Cap.4), ora ritornando nel menu a tendina dei moduli principali si procede all’ultimo step di analisi, ovvero la rappresentazione dei risultati all’interno del modulo di visualizzazione grafica *Paravis*.

6.3.4 Valutazione dei risultati all’interno del modulo *Paravis*

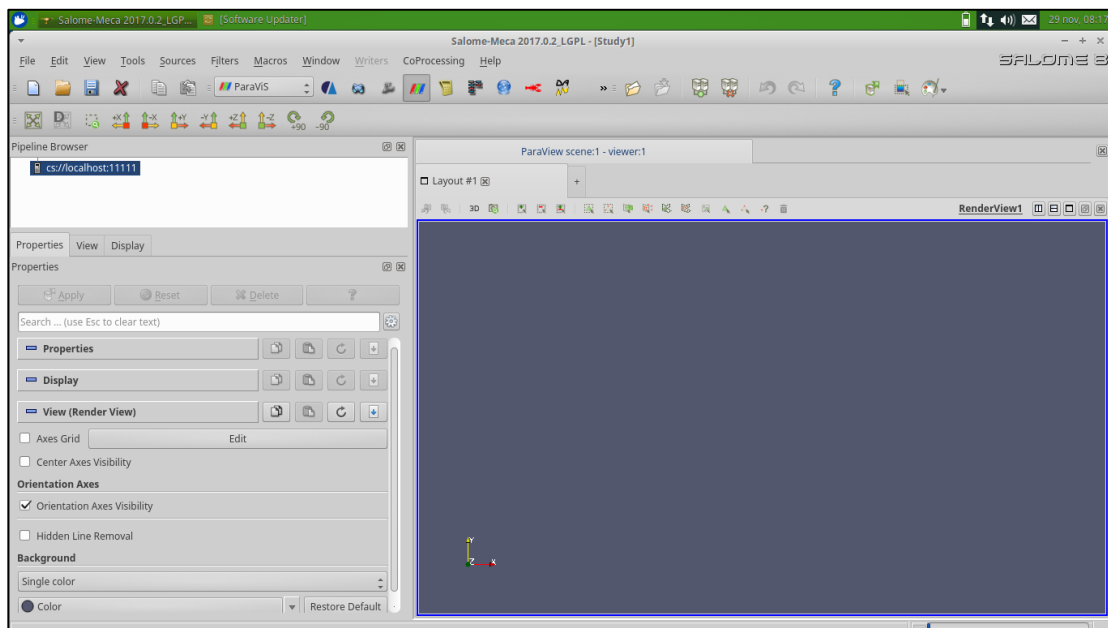


Figura 6.27 Modulo *Paravis*

All'interno dell'ultimo modulo dove visualizzeremo i risultati calcolati attraverso *Asterstudy* verranno spiegati i passaggi ed i filtri utilizzati per poter procedere alla visualizzazione dei risultati, inoltre verranno creati dei grafici che andranno a rappresentare l'andamento delle sollecitazioni e delle tensioni.

Un volta selezionato il modulo dal menu a tendina principale, la schermata di avvio risulterà completamente diversa dalle precedenti, ci troveremo all'interno di un modulo che non eseguirà operazioni di calcolo relativi alla computazione di un risultato, ma al più operazioni di calcolo volte alla visualizzazione grafica di un risultato.

La prima operazione consiste nel cliccare col tasto sulla casella di apertura dei file nella Toolbar a sinistra, e cliccare “*open*”; a quel punto dovremmo scegliere all'interno della nostra directory il percorso per arrivare ad aprire il file *.rmed* creato in precedenza.

Ora si selezionano i campi da voler visualizzare e si clicca su “*Apply*”.

Si prosegue selezionando “*Generate Vectors*” per generare le grandezze vettoriali di ogni singolo step di analisi, con questo comando attivo, aprendo il file relativo all'analisi statica (MECA_STATIQUE) si osserverà che verranno calcolati di Default i campi DEPL (spostamento) e il campo SIEF_ELGA (tensioni/sollecitazioni nei punti di Gauss). Si può scegliere di spuntare i comandi “*Time*” o “*Mode*” per suddividere i time step o per selezionare volutamente un modo di analisi, ovvero un solo timestep (ovviamente per il nostro caso non cambierà nulla, ma nelle analisi sismiche può essere un fattore cruciale).

Ora selezionando dalla barra dei comandi la vista corrente, possiamo settare i parametri all'interno del Toolbox a sinistra, selezionando quale componente del campo vogliamo visualizzare (DEPL, SIEF_ELGA) verranno rappresentati i risultati sul “*Render View*” del modello, e potremmo visualizzare le componenti di spostamento e rotazione (DZ, DX, DY, DRZ, DRY, DRX) del campo spostamenti (DEPL).

Per poter cambiare lo schema dei colori dei grafici, le dimensioni dei caratteri o la scala si potrà lavorare sempre tramite il Tolbox a sinistra.

Adesso cliccando su ogni singolo step di analisi e cliccando sul pulsante “*Scalar*”, per scalare i risultati visualizzati, vedere finalmente i valori relativi agli spostamenti o alle caratteristiche di sollecitazione nei punti di Gauss.

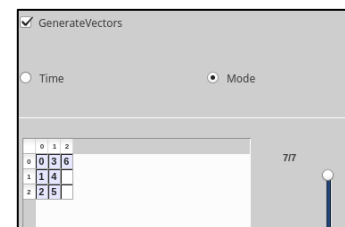
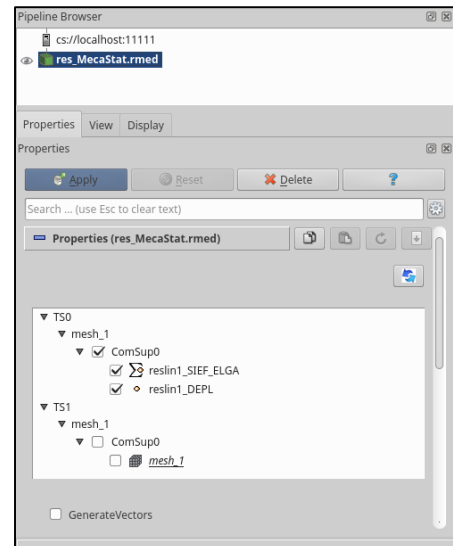


Figura 6.28
Definizione del tipo di output e del timestep

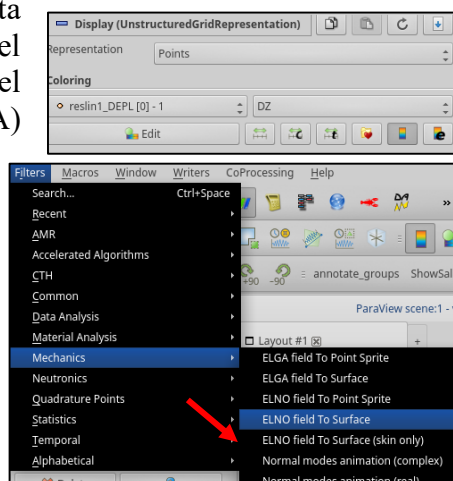


Figura 6.29 Selettore dei filtri

Per poter visualizzare tutti i risultati definiti rispetto ai punti di Gauss o ai nodi degli elementi è necessario cliccando dal menu “*Filters*” l’icona “*Mechanics*” e in seguito “*ELNO/ELGA field to surface*”, cliccando su “*Apply*” sarà in questo modo possibile visualizzare tutti i campi riferiti ad ELNO (e quindi anche a NOEU in quanto ricavati partendo per interpolazione dai valori di ELNO).

Un'altra possibilità che fornisce *Paravis* è la visualizzazione della deformata, questo è possibile attraverso il campo dei risultati DEPL e cliccando sulla funzione “*Warp by vector*”



Le medesime operazioni si possono effettuare anche con il calcolo delle caratteristiche di sollecitazioni e delle tensioni, questo però caricando un file *.rmed* diverso in quanto si è volutamente scelto di stampare i risultati eseguiti sul modulo *Asterstudy* su più file *.rmed* differenziandoli per tipo di risultato; in questo modo avremo spostamenti separati da sollecitazioni e da tensioni in modo da non creare problemi durante l’analisi. Viene infine visualizzata la schermata principale con tutti i comandi più significativi.

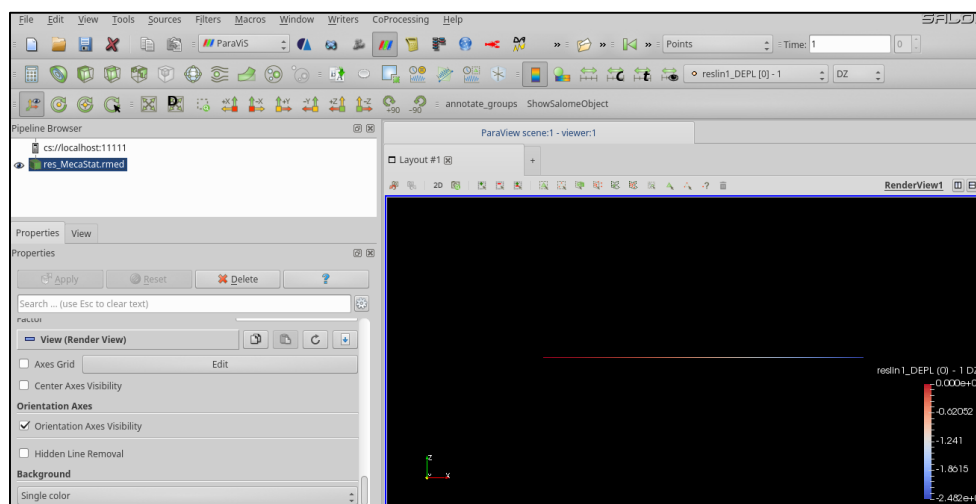


Figura 6.30 Mensola

Dunque abbiamo spiegato le tecniche per ricavare i risultati e i passaggi necessari per la visualizzazione di questi; ovviamente le tecniche di definizione di funzioni, di approccio e di visualizzazione possono variare in base alle esigenze e al problema, è dunque opportuno avere sempre la documentazione necessaria sia per *Code-Aster* sia per *Paravis* in modo da risolvere efficacemente i problemi.

Per questa analisi il nostro obiettivo era visualizzare i risultati calcolati per le caratteristiche di sollecitazione nei punti di Gauss, nei nodi e nei nodi degli elementi sotto le diverse combinazioni di carico (7 combinazioni effettuate); nel seguito vengono rappresentati alcuni risultati ricavati (*per i restanti grafici si rimanda al Cap.7*):

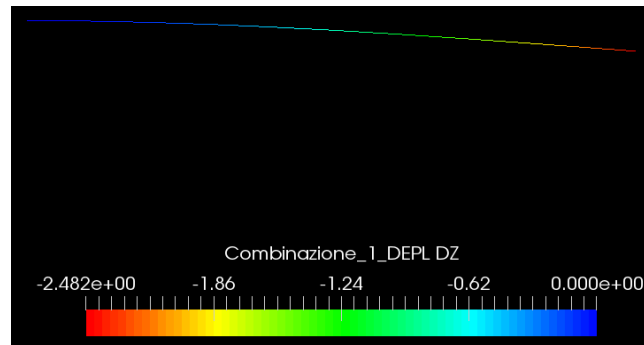


Figura 6.34 Freccia e deformata DZ Comb.1

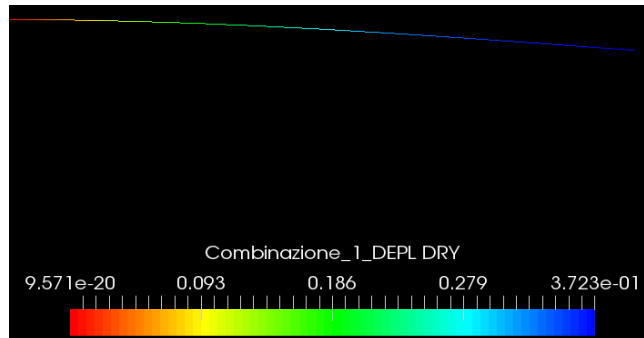


Figura 6.33 Rotazioni DRY comb.1

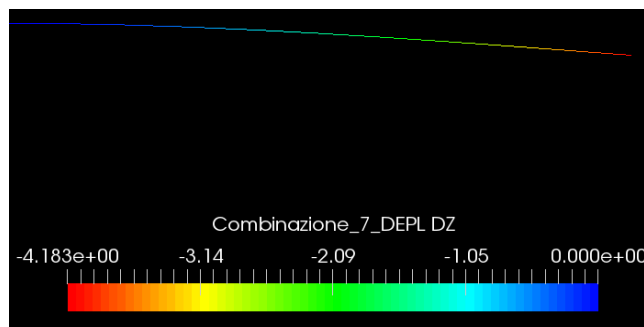


Figura 6.32 Freccia e deformata Comb.7

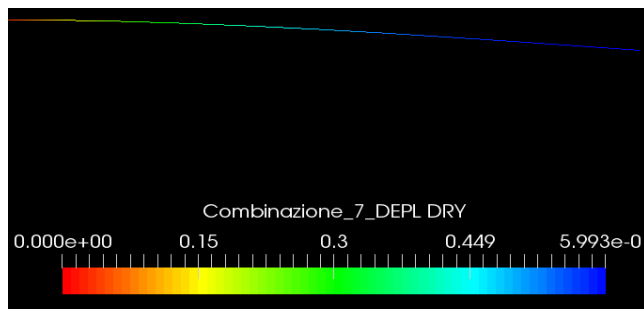


Figura 6.31 Rotazioni DRY Comb.7

Dopo aver rappresentato la mensola soggetta a combinazioni di carico 1 e 7, ed averne visualizzato gli spostamenti in direzione DZ e le rotazioni DRY rappresentati insieme alla deformata della trave, proseguiamo l'analisi in *Paravis* mostrando ora le sollecitazioni MFZ agenti in combinazioni 1 e 7 nel campo EFGE_ELNO:

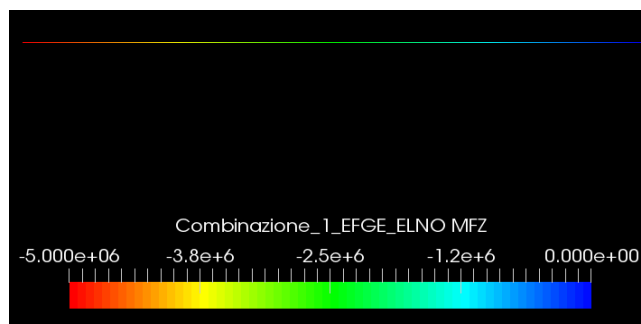


Figura 6.36 Momento Flettente MFZ Comb.1

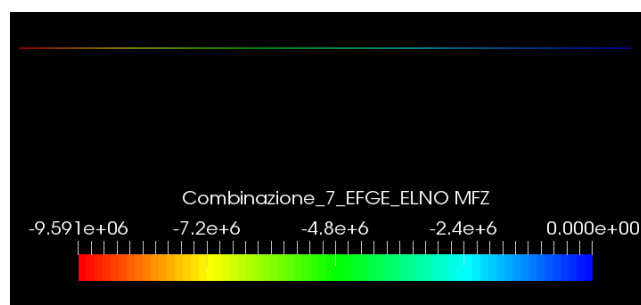


Figura 6.35 Momento Flettente MFZ Comb.7

Vista la scarsa qualità delle immagini si è scelto di proseguire l'analisi plottando graficamente i risultati ricavati per mezzo di una funzione all'interno di "Filters", cliccando su "Data analysis" e in seguito "Plot over Line".

La funzione permette di creare una linea immaginaria sulla trave estrapolando le coordinate dei punti e diagrammando i valori del momento con le coordinate. Questa funzione permette di rappresentare i nostri risultati in maniera più chiara rappresentando tutte le 7 diverse combinazioni di carico volute sia per il momento che per il taglio.

Nel seguito vengono quindi rappresentati i grafici ricavati attraverso tale procedura e vengono plottate le combinazioni 1 e 7 inerenti al momento rimandando tutte le altre combinazioni agli allegati (vedi Cap.7).

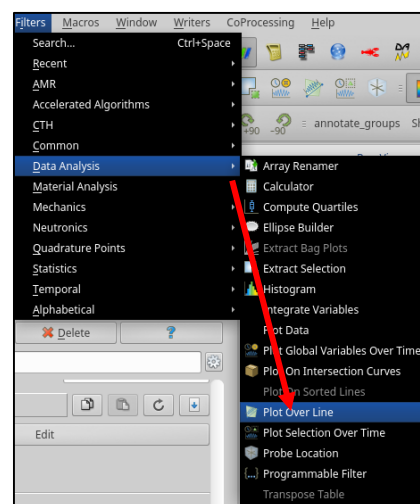


Figura 6.37 Plot over line

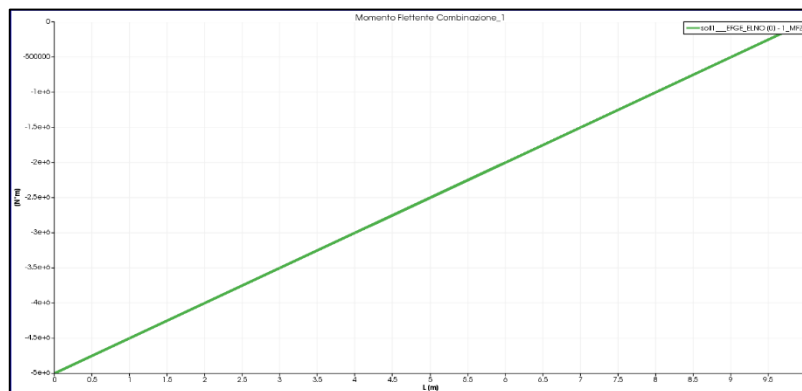


Figura 6.38 Momento Flettente comb.1

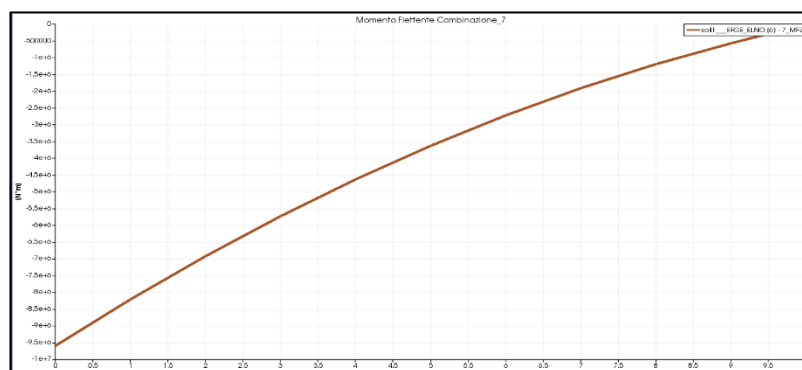


Figura 6.39 Momento Flettente Comb.7

Per dimostrare la veridicità del metodo e fugare dubbi sulla corretta esecuzione della rappresentazione grafica sono state eseguite delle analisi analitiche della mensola per mezzo di un semplice foglio di calcolo di Excel; all'interno del foglio di calcolo sono stati eseguiti i medesimi calcoli e sono state diagrammate in funzione della lunghezza della mensola i momenti flettenti e i tagli sotto le diverse combinazioni di carico. Nel seguito vengono mostrati i diagrammi di sollecitazione ricavati per le medesime combinazioni come mezzo di confronto per la bontà dei risultati.

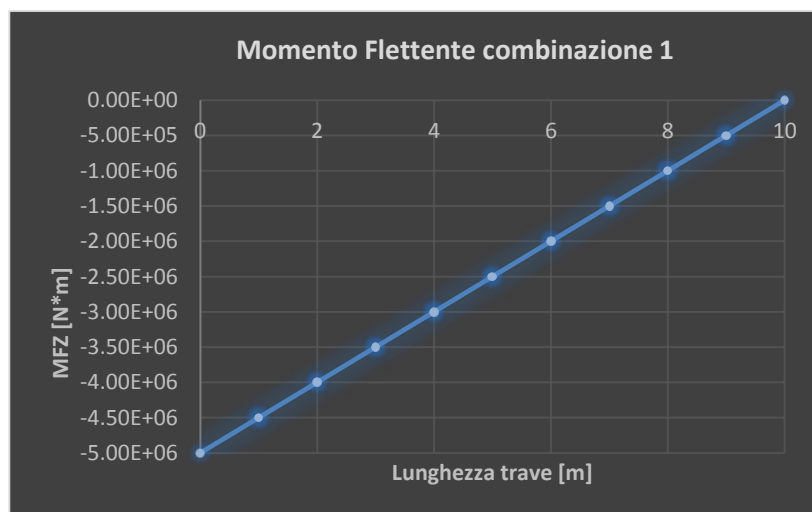


Figura 6.40 Momento flettente Comb.1 (Excel)

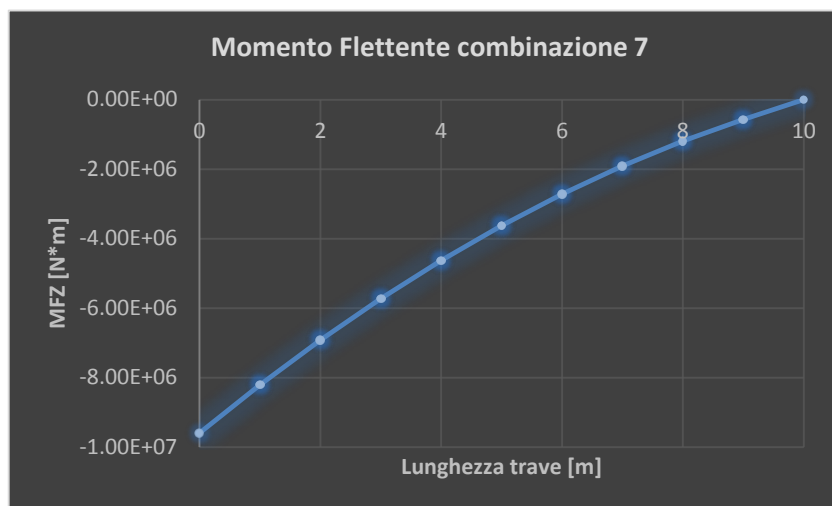


Figura 6.41 Momento flettente Comb.7 (Excel)

Dunque le analisi condotte hanno dimostrato che il software agli elementi finiti *Code-Aster* è in grado di eseguire dei calcoli strutturali e che le rappresentazioni grafiche e numeriche risultano verificate dalle analisi analitiche condotte come mezzo di confronto. In questo capitolo ci si è soffermati prevalentemente sulla metodologia di analisi e sulle funzioni invocate per eseguire una corretta analisi.

Unica constatazione che viene fatta riguarda una procedura che è stata necessaria per eseguire i calcoli appena mostrati. Infatti il software non permette all'utente di posizionare più di un carico distribuito su un elemento strutturale, e per questa ragione è stato necessario utilizzare una tecnica indicata nel libro di *Jean-Pierre Aubry "Beginning with Code_Aster"* nel quale si consiglia di creare una forza puntuale in sede di analisi e suddividerla su ogni punto dell'elemento per camuffare una forza puntuale in un carico distribuito.

$$Qvar = (F * L_{trave}) / (n^{\circ} nodi_{trave})$$

Analiticamente si è mostrato che il metodo è comunque non perfetto, ma i valori risultano essere molto vicini a quelli reali con un errore irrisorio.

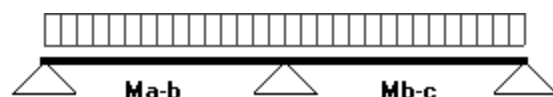
Nei prossimi esercizi casi studio affronteremo altre situazioni di carico e cercheremo di superare il problema appena descritto.

6.4 Trave a due campate soggetta a carichi verticali

Il prossimo caso studio che verrà affrontato riguarda una trave a due campate soggetta a carichi verticali in combinazione tra di loro. L'obiettivo di questo caso studio è riuscire a gestire le combinazioni dei carichi verticali che agiscono sulle due campate utilizzando sempre la tecnica del time-step, già eseguita per la mensola in precedenza, ricavandone i valori di massimo momento flettente sia a positivo che a negativo disponendo i carichi variabili sulle campate in combinazione tra di loro.

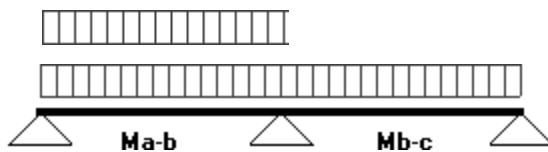
Nel presente caso non andremo a descrivere tutta la procedura eseguita per la mensola in quanto si suppone che la descrizione delle operazioni e delle funzioni eseguibili dal software siano già conosciute, ma ci limiteremo a descrivere i passaggi nell'analisi condotta rappresentando i risultati trovati e verificandoli di conseguenza.

La trave si presenta come nella figura seguente e consiste in una trave a due campate in semplice appoggio soggetta a peso proprio con un carico variabile applicato; si vuole trovare i massimi momenti flettenti agenti sulla trave. La trave ha una sezione pari a (30x40) cm ed è composta da calcestruzzo C25/30. Si definiscono ora le seguenti combinazioni di carico:



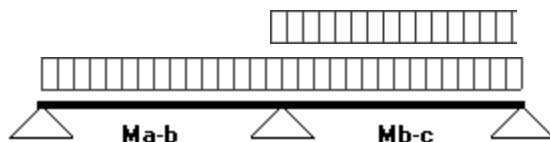
COMBINAZIONE 1

G_p



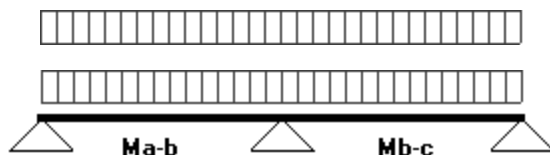
COMBINAZIONE 2

(G_p, Q_{k1}) 1° campata



COMBINAZIONE 3

(G_p, Q_{k1}) 2° campata



COMBINAZIONE 4

(G_p, Q_{k1})

Si precisa inoltre che per le suddette combinazioni di carico verranno utilizzati i coefficienti di sicurezza:

Carichi Permanenti: $\gamma_p = 1.3$

Carichi Variabili: $\gamma_q = 1.5$

Ovviamente tali vengono presi considerando la condizione più sfavorevole di carico.

6.4.1 Trave a due campate: geometria, Mesh e Command File

Una volta aperto il modulo *Geometry* all'interno di Salomè-Meca si procede a definire la geometria nel nostro caso studio.

La trave in questione avrà una lunghezza di $L = 6$ m per ogni campata ed una sezione pari a (30x40) cm.

Il metodo di creazione del modello geometrico è analogo a quello della mensola con l'aggiunta di una campata; in seguito alla creazione della trave verranno creati i gruppi all'interno dell'ambiente *Geometry* ed alla fine avremo il gruppo riguardante la campata_1, il gruppo per la campata_2, il gruppo dei nodi per le cerniere esterne ed il gruppo nodi per l'appoggio centrale:

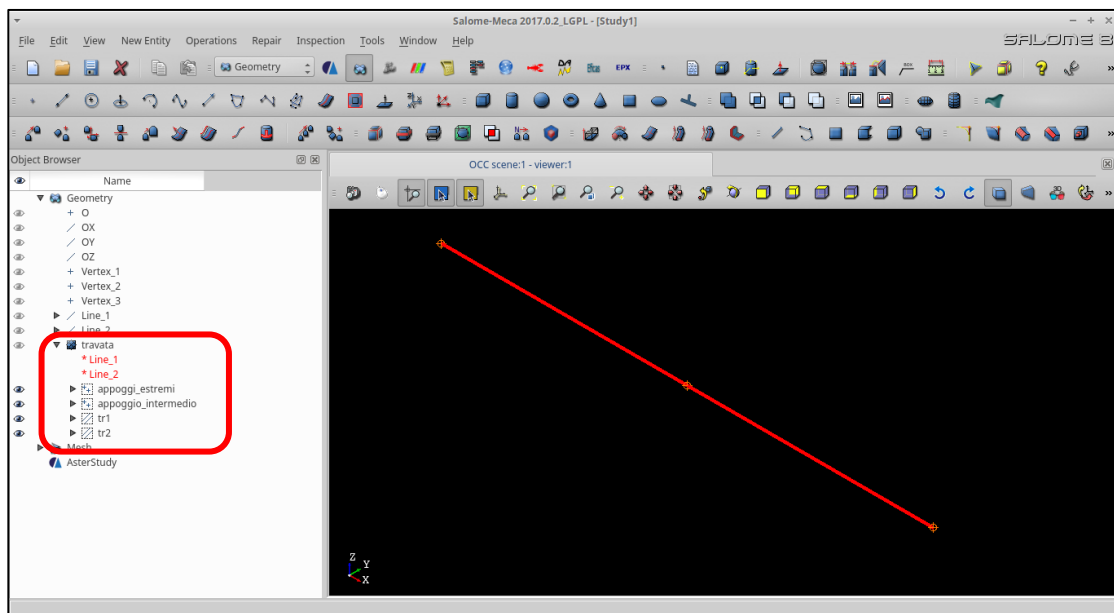


Figura 6.42 Trave a due campate

Una volta definita la geometria della trave a due campate si procede a discretizzare la trave in elementi, la *Mesh* viene creata tramite le stesse procedure con cui è stata creata per la mensola, ovvero elementi di 1 m:

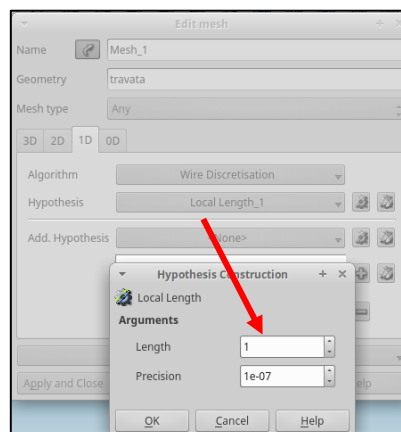


Figura 6.43
Discretizzazione della trave

Una volta discretizzato la nostra trave a due campate si creano i gruppi dei nodi uguali a quelli definiti in *Geometry*, e dopo un gruppo di tutti nodi presenti sulla trave; infine si ricorda di fare sempre un controllo sui nodi e sugli elementi tramite i comandi “merge nodes” e “double nodes”:

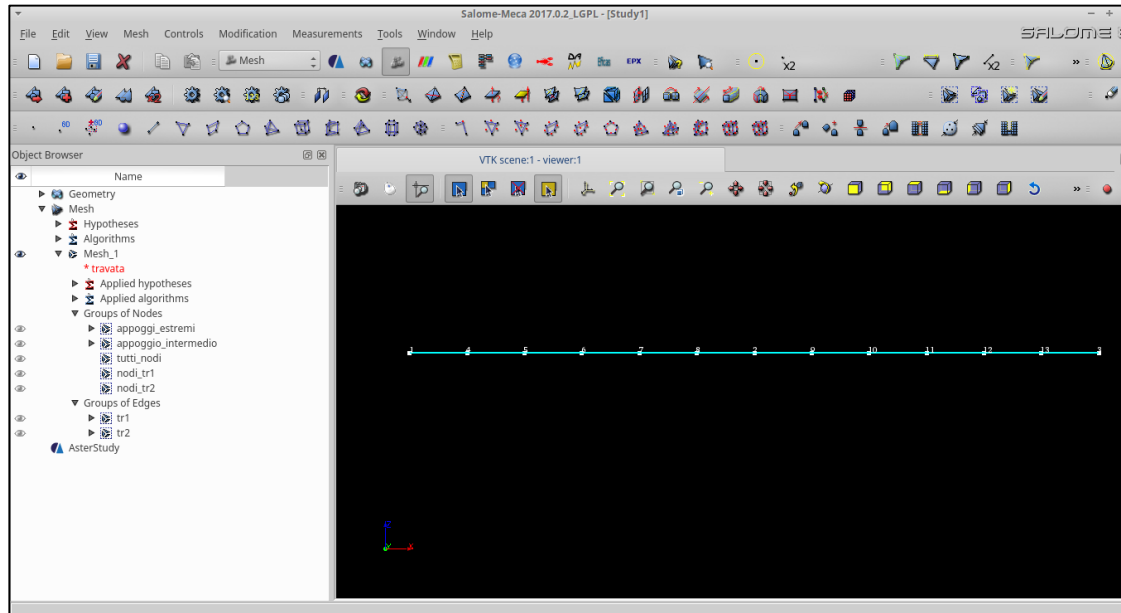


Figura 6.44 Trave a due campate discretizzata

A questo punto abbiamo ottenuto una *Mesh* per la trave a due campate lungo la sua lunghezza. Ora possiamo procedere ed avanzare con l’analisi andando a compilare il “*Command File*” all’interno del modulo *Asterstudy*; per differenziare l’operazione rispetto a prima faremo vedere la creazione del suddetto file tramite file di testo, ovvero esplicheremo direttamente tutte le funzioni che invocheremo per eseguire il calcolo delle sollecitazioni, ovviamente mostreremo le funzioni principali mentre esporremo il file completo all’interno degli allegati (vedi cap.7.5).

Si procede alla lettura della *Mesh* invocandone il nome ed il formato:

```
DEBUT ( )
```

```
mesh = LIRE_MALLAGE (FORMAT='MED', INFO_MED=2, NOM_MED='Mesh_1',  
UNITE=20)
```

Si definisce il modello meccanico della trave Eulero-bernoulli a tutti gli elementi:

```
model = AFFE_MODELE (  
    AFFE=_F (MODELISATION=('POU_D_E', ), PHENOMENE='MECANIQUE',  
TOUT='OUI'),  
    MAILLAGE=mesh  
)
```

In questo caso viene orientato l’elemento strutturale con asse Y locale sull’asse Z globale, nel seguito sono definite le dimensioni geometriche della sezione:

```
elemprop = AFFE_CARA_ELEM (  
    MODELE=model,
```

```

ORIENTATION=_F(
    CARA='VECT_Y', GROUP_MA=('tr1', 'tr2'), VALE=(0.0, 0.0, 1.0)
),
POUTRE=_F(
    CARA=('HY', 'HZ'),
    GROUP_MA=('tr1', 'tr2'),
    SECTION='RECTANGLE',
    VALE=(0.3, 0.4)
)

```

Ora si mostra la creazione della funzione temporale relativa al solo peso proprio (ma lo faremo per tutti i carichi) che assume un determinato valore in relazione all'istante d'invocazione (Si osserva come il peso proprio sarà mantenuto per tutte le combinazioni con coefficiente di sicurezza pari a 1.3):

```

func_pp = DEFI_FONCTION(
    INTERPOL=('LIN', ),
    NOM_PARA='INST',
    PROL_DROITE='CONSTANT',
    PROL_GAUCHE='CONSTANT',
    TITRE='combinazioni pp_var1',
    VALE=(1.0, 1.3, 2.0, 1.3, 3.0, 1.3, 4.0, 1.3)
)

```

Definite tutte le funzioni, si genera il passo di avanzamento del time-step:

```

listr = DEFI_LIST_REEL(
    DEBUT=1.0,
    INTERVALLE=_F(JUSQU_A=4.0, PAS=1.0),
    TITRE='combinazioni carichi'
)

```

Si dichiara ora il carico relativo al peso proprio, quindi orientato con verso opposto all'asse Z:

```

pp = AFFE_CHAR_MECA(
    MODELE=model,
    PESANTEUR=_F(
        DIRECTION=(0.0, 0.0, -1.0), GRAVITE=g, GROUP_MA=('tr1',
'tr2')
    )
)

```

Come fatto con i carichi, si definiscono i vincoli; nel nostro caso avremo tutte cerniere:

```

bc = AFFE_CHAR_MECA(
    DDL_IMPO=_F(
        DRX=0.0,
        DRZ=0.0,
        DX=0.0,
        DY=0.0,
        DZ=0.0,
        GROUP_NO=('appoggi_estremi', 'appoggio_intermedio')
    )
)

```

```

    ),
    MODELE=model
)

```

Con la seguente funzione si definisce il tipo di analisi meccanica-statica; viene invocato il materiale e la geometria della sezione, dopo sono definiti i carichi applicati correlati con la funzione temporale con cui sono stati associati; questa operazione permette di moltiplicare un carico per il coefficiente assegnato al prefissato istante temporale:

```

reslin = MECA_STATIQUE(
    CARA_ELEM=elemprop,
    CHAM_MATER=fieldmat,
    EXCIT=(
        _F(CHARGE=bc), _F(CHARGE=pp, FONC_MULT=func_pp),
        _F(CHARGE=qv1, FONC_MULT=func_q1), _F(CHARGE=qv2,
FONC_MULT=func_q2)
    ),
    LIST_INST=listr,
    MODELE=model
)

```

La funzione CALC_CHAMP consente di arricchire i calcoli effettuati sulla struttura, invocando i relativi campi di analisi vengono calcolate le caratteristiche di sollecitazione per i punti di Gauss, per i nodi degli elementi (ELNO) e per i nodi (NOEU):

```

MNV = CALC_CHAMP(
    CARA_ELEM=elemprop,
    CONTRAINTE=('EFGE_ELGA', 'EFGE_ELNO', 'EFGE_NOEU'),
    LIST_INST=listr,
    MODELE=model,
    RESULTAT=reslin
)

```

Infine vengono stampati i risultati che verranno poi visualizzati tramite *Paravis*, si noti come sia importante il termine “UNITE 80” poiché è il termine che consente di assegnare la dimensione di memoria al file di output:

```

IMPR_RESU(
    FORMAT='MED',
    RESU=_F(
        CARA_ELEM=elemprop, LIST_INST=listr, RESULTAT=reslin,
TOUT_CHAM='OUI'
    ),
    UNITE=80
)

```

Il file appena creato non è completo, ma vi sono solo alcuni parti di codice che rappresentano l'ossatura del file.

Per generare la combinazione dei carichi anche in questa analisi non è stato possibile generare una combinazione che comprenda due carichi variabili applicati sulla medesima campata, questa limitazione di *Code-Aster* è stata superata tramite l'utilizzo della medesima formula impiegata per le analisi della mensola, ovvero:

$$Qvar = (F * L_{trave}) / (n^{\circ} nodi_{trave})$$

Anche in questo caso i risultati saranno corretti, ma con errore marginale.

Conclusa la creazione *Command File* si procede nel visualizzare i risultati ricavati tramite *Paravis*; nel seguito vengono mostrati le immagini del *RenderView* ricavate, per ulteriori grafici ed immagini si rimanda agli allegati (*cap.8.3*)

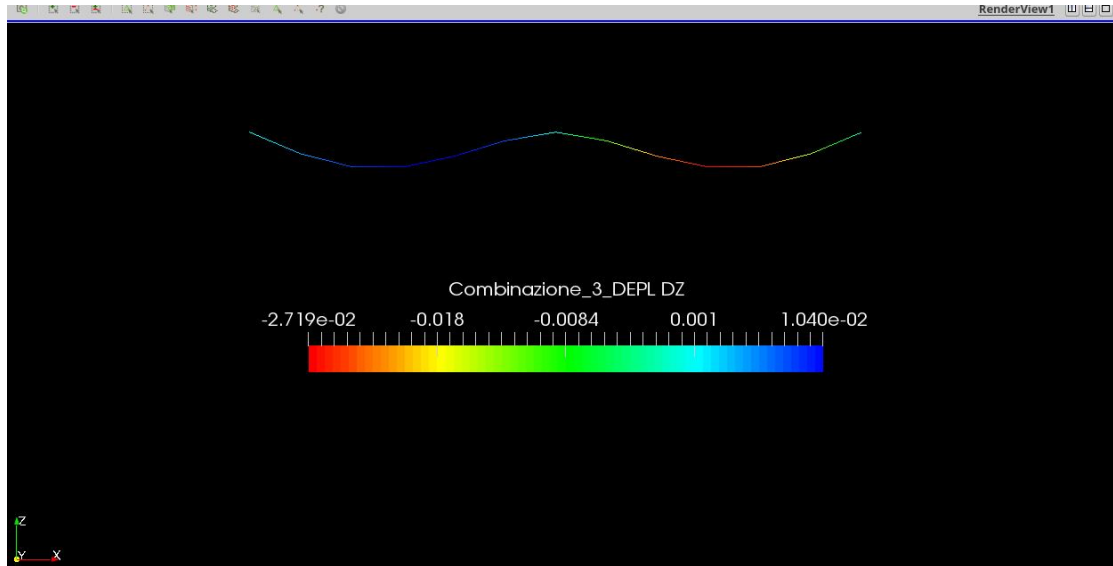


Figura 6.45 Spostamenti e deformata Comb.3

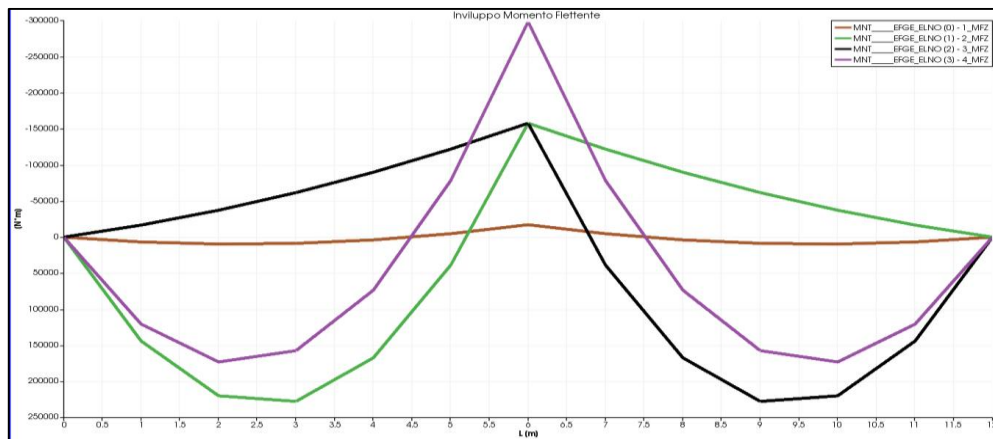


Figura 6.46 Involuppo momento flettente

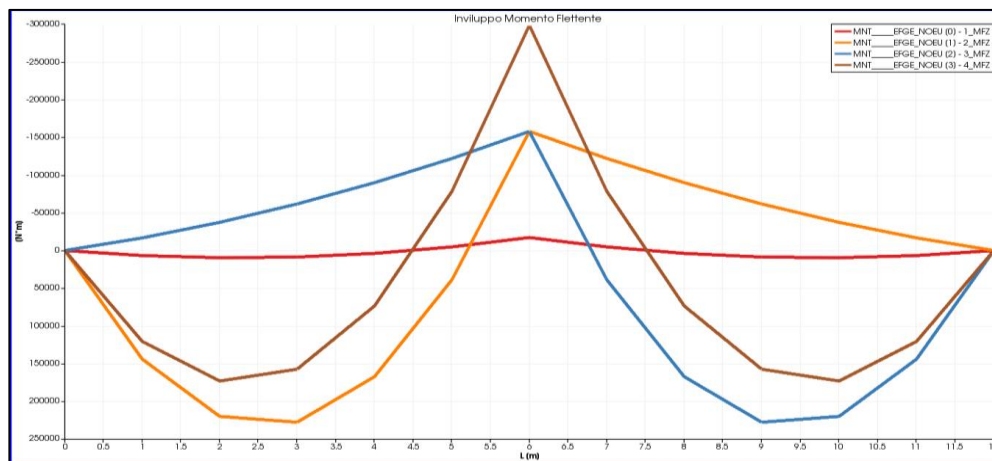


Figura 6.47 Involuppo momento flettente

Come si evince dai precedenti grafici si può osservare come i campi rappresentati EFGE_ELNO ed EFGE_NOEU siano due campi che rappresentino risultati molto simili tra di loro, ma questo era presumibile dall'ipotesi che le sollecitazioni sui nodi sono ricavate a partire dalle sollecitazioni sui nodi degli elementi, quindi NOEU è funzione di ELNO.

6.4.2 Trave a due campate: analisi dei risultati e verifica

Il calcolo delle sollecitazioni ha prodotto dei valori che non sono dissimili dai valori EFGE_ELNO e EFGE_NOEU, per questa ragione abbiamo provveduto a impostare la verifica dei valori di massimo momento flettente ai valori ai nodi; una volta calcolati i valori massimi di momento nelle due campate sotto le differenti combinazioni di carico, si procede a valutare tramite i digrammi d'interazione se tali valori risultano verificati.

Dunque si procede calcolando i limiti dei campi del diagramma d'interazione e per fare questo, ovviamente, si ipotizza una quantità di armatura longitudinale pari a (5+5)Φ24 per la trave studiata. Nel seguito si riportano i valori ricavati:

f_{ck} [MPa]	25
f_{yk} [MPa]	450
B [mm]	300
H [mm]	400
A_s [mm ²]	2260.8
A'_s [mm ²]	2260.8
$c+\phi/2$ [mm]	40
α_{cc} [-]	0.85
ϵ_{uk} [‰]	75.0
ϵ_{ud} [‰]	68.0
E_s [MPa]	200000

f_{cd} [MPa]	14
f_{yd} [MPa]	391
γ_s	1.15
γ_c	1.5
d [mm]	360
d' [mm]	40
A [mm ²]	120000
γ_G [mm]	200.0
ϵ_{yd} [‰]	1.96
ϵ_{cu2} [‰]	3.5
ϵ_{c2} [‰]	2.0

Tabella 6.1 Limiti dei campi

	CAMPO 1 (iniz.)	CAMPO 1-2	CAMPO 2-3	CAMPO 3-4	CAMPO 4- 4a	CAMPO 4a-5	CAMPO 5 (final.)
N_{Rd} [kN]	1769.3	1769.3	1708.7	-794.5	-2123.2	-2324.2	-3469.3
M_{Rd} [kNm]	0.0	0.0	11.7	365.7	203.8	177.7	0.0
N_{Rd} [kN]	1769.3	1769.3	1708.7	-794.5	-2123.2	-2324.2	-3469.3
M_{Rd} [kNm]	0.0	0.0	-11.7	-365.7	-203.8	-177.7	0.0

Diagrammando i limiti dei campi e inserendovi i valori dei massimi momenti flettente si ottiene:

Tabella 6.2 Massimi momenti agenti

INST	Mmax [kNm]	N [kN]
1	9.56475	0
2	227.581	0
3	227.581	0
4	172.958	0
INST	Mmin [kNm]	N [kN]
1	-17.2165	0
2	-157.842	0
3	-157.842	0
4	-298.467	0

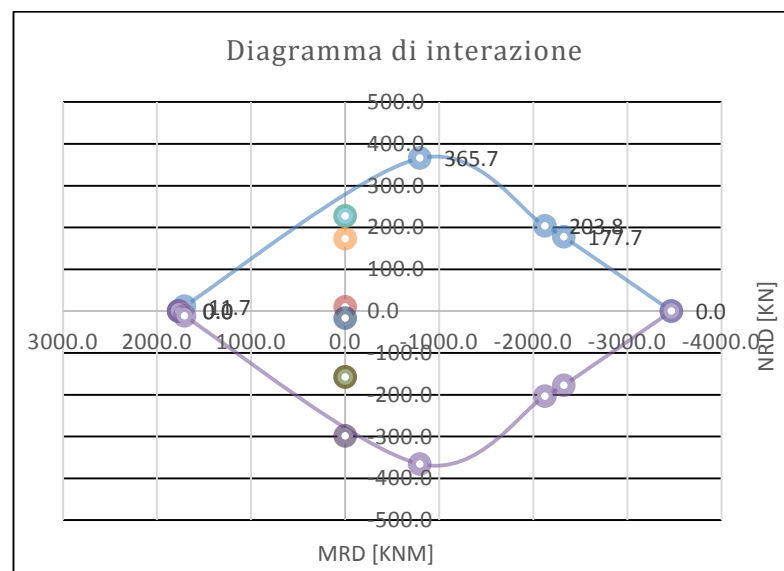


Figura 6.48 Diagramma di interazione

Come si può osservare risulta esserci solo un punto al di fuori del diagramma, e secondo i nostri calcoli la sezione risulta essere la sezione di appoggio tra le due campate; pertanto sarà necessario incrementare la dimensione della sezione della trave in oggetto ed eventualmente anche il quantitativo di armatura.

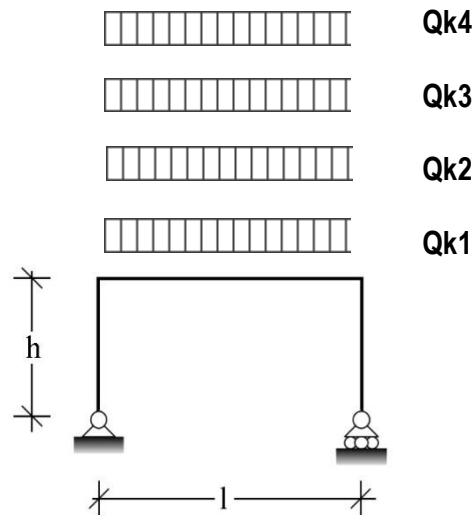
Per arrivare ad ottenere i valori dei momenti massimi e minimi per ogni punto si è provveduto ad estrarre dal modulo di *Paravis* la tabella “*SpreadSheet*” che consente di ottenere in formato tabellare i valori dei campi che abbiamo indicato in sede di analisi; la tabella viene estrapolata in formato .csv compatibile con Excel.

6.5 Portale soggetto a carichi verticali

Nel presente capitolo verrà trattato il caso studio inerente ad una portale caricato da carichi distribuiti permanenti e variabili. Lo studio vuole dimostrare che il comportamento assunto dal portale sotto l'azione di carichi verticali agenti trasversalmente rispetto alla direzione dell'asta orizzontale risulta essere consono rispetto ad uno studio condotto con altri programmi o analisi analitiche precedenti.

Questa volta si cercherà di fare un passo avanti ulteriore rispetto al caso studio precedente, infatti si proverà a caricare il portale senza l'ausilio della procedura time step, superando questa limitazione ed eseguendo i calcoli attraverso un'altra tecnica che, si spera, possa portare ad un risultato favorevole ed affine ai calcoli effettuati analiticamente.

Lo studio del portale ci porterà a sviluppare un “*Command File*” che con alcune funzioni nuove, che introdurremo strada facendo; verranno inoltre spiegati alcuni metodi per ottenere una stampa dei risultati in modo tabellare, filtrando le componenti selezionate, infine i carichi variabili che interesseranno il portale saranno quattro e verranno trattati tramite coefficiente di sicurezza (escludendo i coefficienti di combinazione):



Il portale in figura avrà le seguenti caratteristiche:

$h = 4 \text{ m}$;

$L = 4 \text{ m}$;

Inoltre la sezione trasversale sarà una sezione quadrata di (50X50) cm;

Sarà costituito interamente di calcestruzzo C25/30, quindi andremo a considerare anche il peso proprio oltre ai carichi variabili agenti in combinazione tra di loro ed aventi valore di:

$$Q_{k1} = 50 \text{ KPa}; \quad Q_{k2} = 50 \text{ KPa}; \quad Q_{k3} = 10 \text{ KPa}; \quad Q_{k4} = 0 \text{ KPa};$$

Carichi Permanenti: $\gamma_p = 1.3$

Carichi Variabili: $\gamma_q = 1.5$

Definiti ora i carichi variabili si procede nel creare il modello geometrico.

6.5.1 Portale: geometria e Mesh

All'interno dell'ambiente *Geometry* si procede nella medesima maniera delle precedenti definendo la geometria, i gruppi e tutte le caratteristiche relative alla geometria.

Si indicano i gruppi relativi ai nodi incastro al piede, ai nodi in testa al portale, il gruppo dell'asta orizzontale, il gruppo delle aste verticali ed infine si definisce l'elemento *Compound_1* che ingloba tutto il portale.

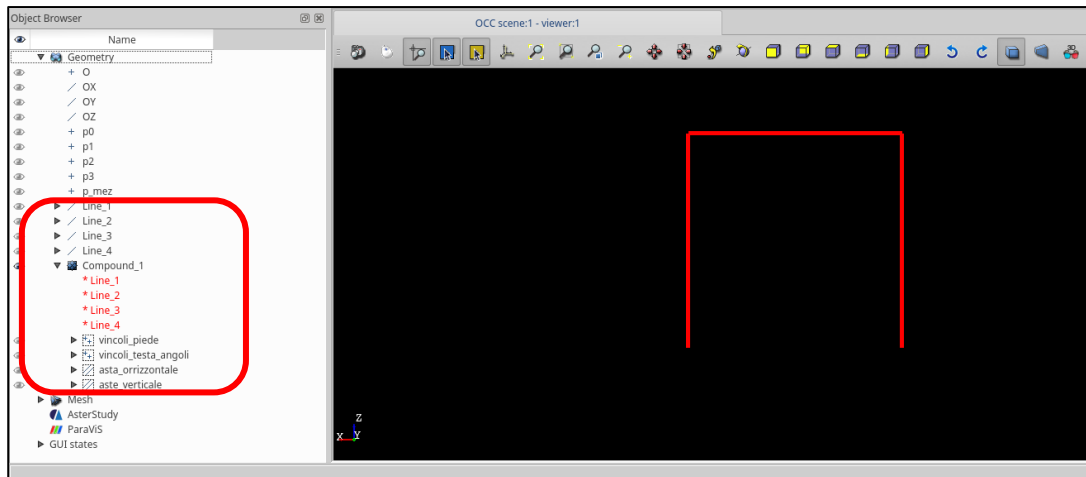


Figura 6.49 Portale

Una volta definita la geometria si procede all'interno del modulo *Mesh* per discretizzare l'elemento. La procedura segue le medesime consegne già eseguite in precedenza, e si discretizza il portale unidimensionalmente suddividendolo in elementi della lunghezza di 0.5 m e si riportano i medesimi gruppi definiti in precedenza.

A questo punto si procede nel calcolo delle sollecitazioni all'interno del modulo *Asterstudy*.

6.5.2 Portale: generazione del Command File

Una volta aperto il modulo di *Asterstudy* si procede come sempre nel caricare la *Mesh* e nel settare le funzioni che abbiamo già descritto in precedenza. Dopo aver definito la *Mesh*, il materiale e il modello meccanico alla Eulero-Bernoulli una caratteristica da sottolineare sarà l'orientamento differente rispetto ai precedenti esercizi. Infatti ora il vettore Y locale è orientato con il vettore Y globale, per questa ragione si dovranno considerare MFY e VZ come parametri del momento e del taglio.

```
elemprop = AFFE_CARA_ELEM(  
  MODELE=MODELLO,  
  ORIENTATION=_F(  
    CARA='VECT_Y', GROUP_MA=('Compound_1', ), VALE=(0.0, 1.0,  
0.0)  
  ),  
  POUTRE=_F(  
    CARA=('HY', 'HZ'),
```

```

GROUP_MA=('aste_verticale', 'asta_orrizzontale'),
SECTION='RECTANGLE',
VALE=(0.5, 0.5)
)
)

```

Come già detto la formulazione sarà differente dal time step, infatti dopo aver definito tutti i carichi come di consueto, si esegue l'analisi meccanica con tutti i carichi definiti e i vincoli definiti (che saranno solamente due incastri ai nodi al piede), quindi:

```

resq1 = MECA_STATIQUE(
    CARA_ELEM=elemprop,
    CHAM_MATER=As_CLS,
    EXCIT=(_F(CHARGE=BC2), _F(CHARGE=caricoq1)),
    MODELE=MODELLO
)

```

In seguito si calcolano le sollecitazioni riferite ad nei nodi degli elementi e nei punti di Gauss per tutti i carichi esistenti (il calcolo non può essere fatto nei nodi poiché il software non riconosce l'orientamento nei nodi in testa):

```

R_q1 = CALC_CHAMP(
    CARA_ELEM=elemprop,
    CONTRAINTE=('EFGE_ELNO', 'EFGE_ELGA'),
    MODELE=MODELLO,
    RESULTAT=resq1
)

```

Si continua definendo una nuova funzione CREA_CHAMP questa funzione ha la particolarità di estrarre una componente da un campo (come EFGE_ELNO) per poi stamparlo o effettuare altre operazioni (di per sé l'operazione non avrebbe senso poiché si estrarrebbe un campo di risultati avendolo già invocato):

```

c_q1 = CREA_CHAMP(
    NOM_CHAM='EFGE_ELNO',
    OPERATION='EXTR',
    RESULTAT=R_q1,
    TYPE_CHAM='ELNO_SIEF_R'
)

```

L'operazione precedente poteva sembrare illogica, ma l'estrazione del campo di risultati consente di isolare il campo e di assemblarlo tramite combinazione lineare e applicando i coefficienti di sicurezza ai carichi distribuiti applicati:

```

SLU = CREA_CHAMP(
    ASSE=(
        _F(CHAM_GD=c_pp, COEF_R=g_g1s, CUMUL='OUI', TOUT='OUI'), _F(
            CHAM_GD=c_q1, COEF_R=g_qs, CUMUL='OUI', TOUT='OUI'
        ), _F(CHAM_GD=c_q2, COEF_R=g_qs, CUMUL='OUI', TOUT='OUI'),
        _F(
            CHAM_GD=c_q3, COEF_R=g_qs, CUMUL='OUI', TOUT='OUI'
        ), _F(CHAM_GD=c_q4, COEF_R=g_qs, CUMUL='OUI', TOUT='OUI')
    ),
    MODELE=MODELLO,
)

```

```

        OPERATION='ASSE',
        TYPE_CHAM='ELNO_SIEF_R'
    )

```

Nel seguito si rappresenta un'altra nuova funzione, ovvero la funzione POST_RELIEVE_T, questa funzione consente di generare una tabella di output all'interno del quale noi andremo a stampare i valori nei due gruppi dei nodi e nel nodo di mezzzeria.

In questo caso noi prenderemo in considerazione il peso proprio del portale, estrarremo i valori del campo indicato nei gruppi dei punti indicati (o elementi) e stamperemo solamente le componenti del campo che indicheremo (MFY e VZ). Il tutto servirà per avere una tabella di confronto tra le sollecitazioni agenti nei punti più considerevoli.

```

Tabella = POST_RELEVE_T(
    ACTION=(
        _F(
            GROUP_NO=('punto_mezz', 'vincoli_piede',
'vincoli_testa_angoli'),
            INTITULE='Peso Proprio agente sul traverso
orizzontale',
            NOM_CHAM='EFGE_ELNO',
            NOM_CMP=('MFY', 'VZ'),
            OPERATION=('EXTRACTION', ),
            RESULTAT=R_pp
        ), _F(
            GROUP_NO=('Forza_mezz', ),
            INTITULE='Peso proprio',
            NOM_CHAM='EFGE_ELNO',
            OPERATION=('EXTRACTION', ),
            REPERE='GLOBAL',
            RESULTAT=R_pp,
            TOUT_CMP='OUI'
        )
    ),
    TITRE='Tabella di confronto dei valori M, N e V nel punto di
mezzzeria'
)

```

Infine per stampare la tabella dovremmo richiamare la funzione all'interno di:

```

IMPR_TABLE(
    FORMAT='TABLEAU',
    FORMAT_R='E12.3',
    SEPARATEUR=" ",
    TABLE=Tabella,
    UNITE=11
)

```

Questa tabella sarà organizzata per il semplice confronto dei valori nei nodi; ora possiamo procedere e rappresentare le immagini dei valori che si ritroveranno all'interno del modulo *Paravis*. Si rappresentano in tal senso i grafici relativi agli spostamenti (con deformata) lungo DZ e lungo DX e le rotazioni DRY; si prosegue con i grafici plottati delle sollecitazioni MFY, N e VZ.

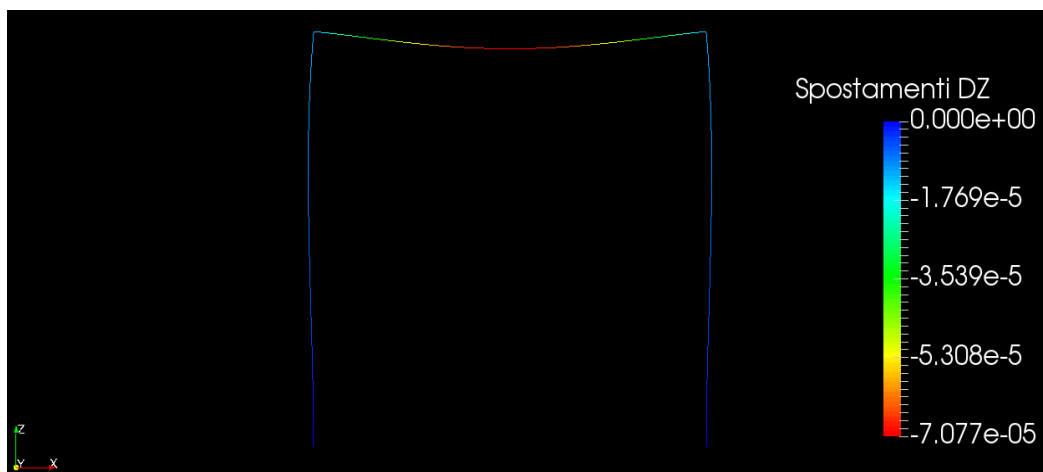


Figura 6.50 Spostamenti e deformata DZ

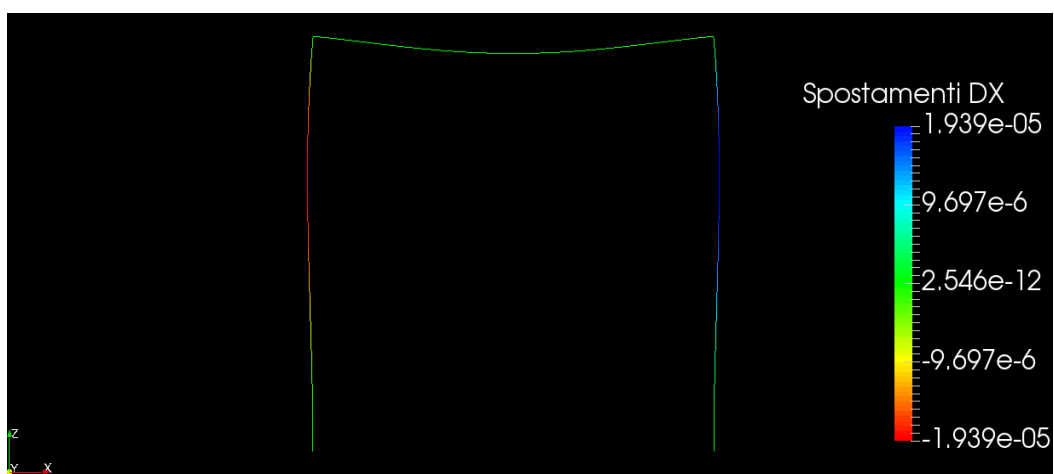


Figura 6.51 Spostamenti e deformata DX

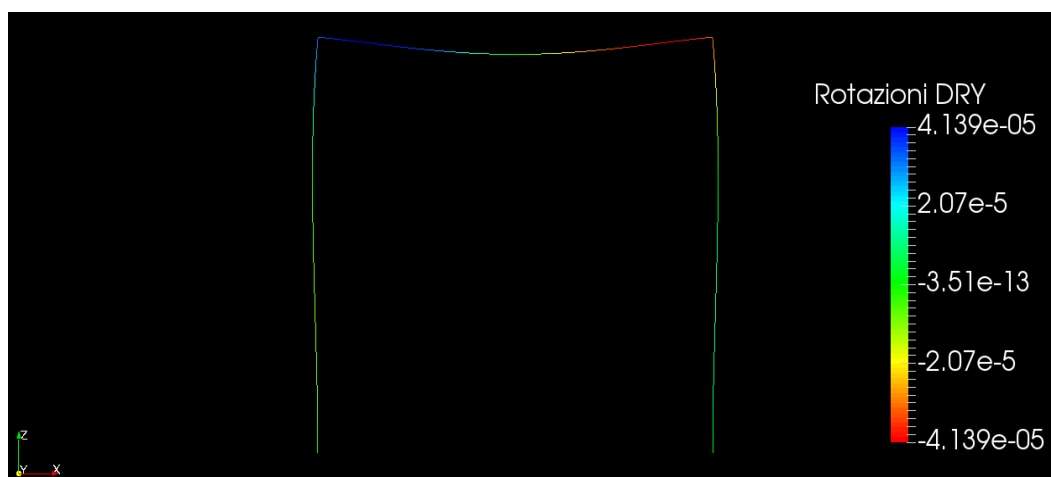


Figura 6.52 Rotazioni DRY

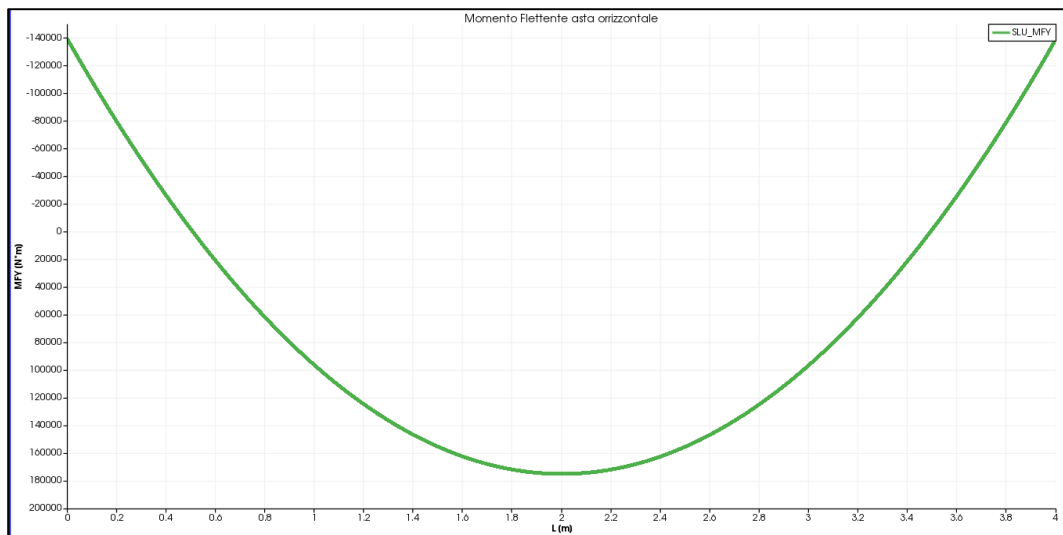


Figura 6.53 Momento Flettent sull'asta orrizzontale

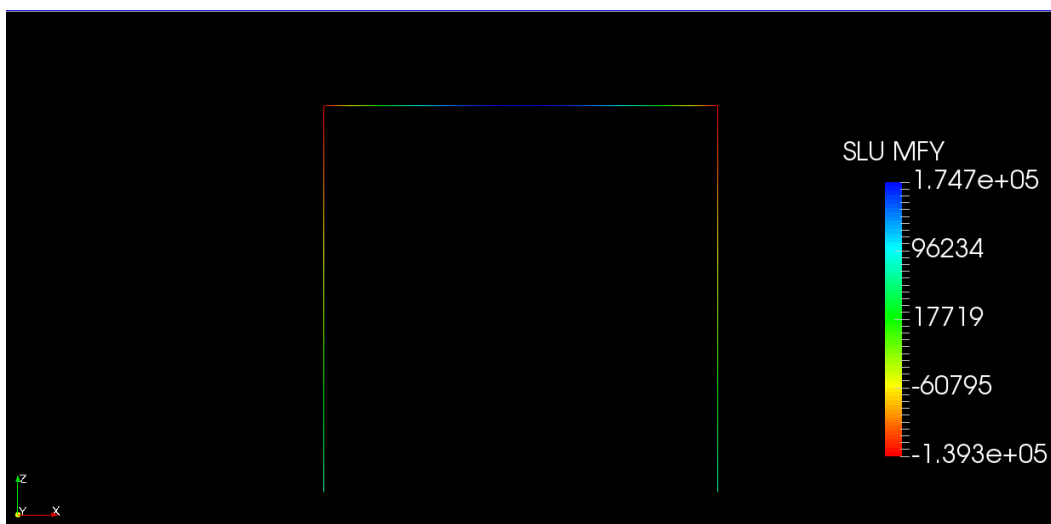


Figura 6.54 Momento Flettente sollecitante

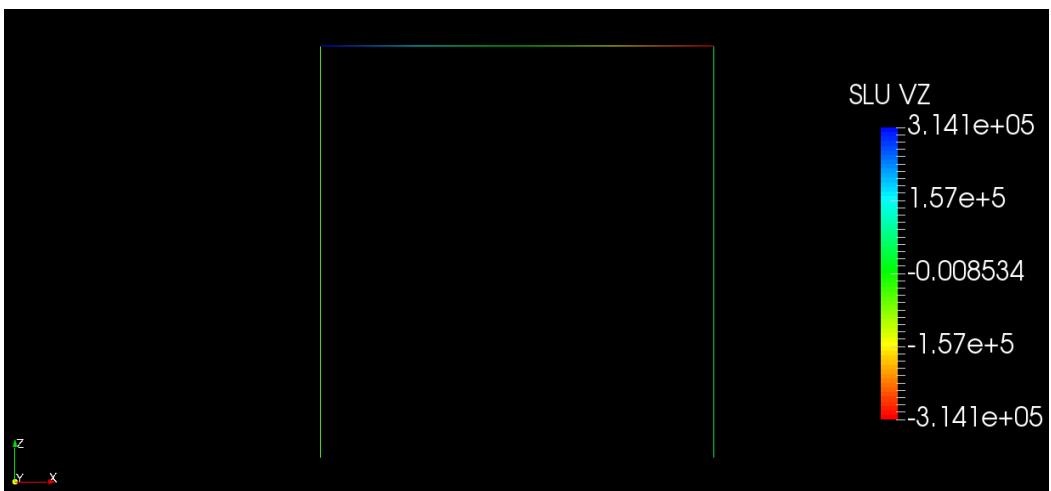


Figura 6.55 Taglio sollecitante

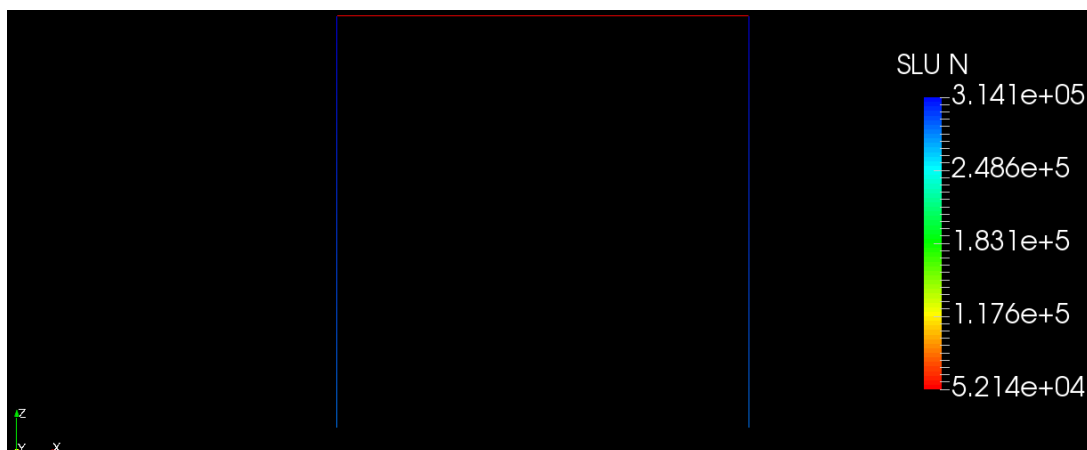


Figura 6.56 Sforzo normale agente

La cosa che viene subito evidenziata è il comportamento corretto del portale sotto l'azione dei carichi distribuiti ovvero le aste verticali tendono a deformarsi in direzione X e il traverso orizzontale in direzione DZ.

6.5.3 Portale: analisi dei risultati e verifica

Nel presente capitolo andremo a verificare i valori ricavati nell'esercizio precedente attraverso la medesima analisi condotta per la trave a due campate, ovvero andremo a verificare il valore del massimo momento flettente positivo/negativo all'interno dei diagrammi di interazione.

Il metodo di analisi sarà il medesimo spiegato in precedenza e ne verranno riprese le peculiarità per il calcolo dei momenti flettenti.

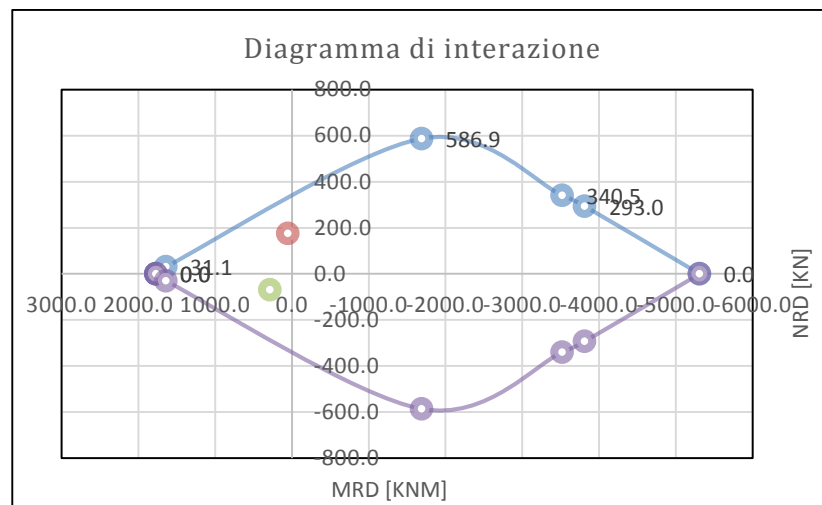
f_{ck} [MPa]	25
f_{yk} [MPa]	450
B [mm]	500
H [mm]	500
A_s [mm ²]	2260.8
A'_s [mm ²]	2260.8
$c+\phi/2$ [mm]	40
α_{cc} [-]	0.85
ϵ_{uk} [‰]	75.0
ϵ_{ud} [‰]	68.0
E_s [MPa]	200000
ϵ_{cu2} [‰]	3.5
ϵ_{c2} [‰]	2.0

f_{cd} [MPa]	14
f_{yd} [MPa]	391
γ_s	1.15
γ_c	1.5
A [mm ²]	250000
ϵ_{yd} [‰]	1.96
d [mm]	460
d' [mm]	40

Mmax	N	Mmin	N
175	5.21E+01	-69.25	2.82E+02

	CAMPO 1 (iniz.)	CAMPO 1- 2	CAMPO 2-3	CAMPO 3-4	CAMPO 4- 4a	CAMPO 4a-5	CAMPO 5 (final.)
N_{Rd} [kN]	1769.3	1769.3	1640.2	-1691.9	-3522.4	-3815.0	-5311.0
M_{Rd} [kNm]	0.0	0.0	31.1	586.9	340.5	293.0	0.0
N_{Rd} [kN]	1769.3	1769.3	1640.2	-1691.9	-3522.4	-3815.0	-5311.0
M_{Rd} [kNm]	0.0	0.0	-31.1	-586.9	-340.5	-293.0	0.0

Dunque si procede calcolando i limiti dei campi del diagramma d'interazione e per fare questo, ovviamente, si ipotizza una quantità di armatura longitudinale pari a $(5+5)\Phi 24$ per il portale. Si verificheranno che i valori momento Massimo e Minimo momento flettente rientrano all'interno del diagramma di interazione:



Come si può osservare il digramma contiene i punti di momento flettente accoppiati con i valori dello sforzo normale, in quanto essendo un portale sarà anche soggetto a queste sollecitazioni.

Dunque a differenza del precedente esercizio, in questo caso la verifica è soddisfatta e non è necessario incrementare la sezione trasversale.

7 Conclusioni

Al termine del nostro percorso che ci ha portato a conoscere un software FEM agli elementi finiti *Open-Source* adoperato per lo studio di elementi strutturali semplici sottoposti a differenti combinazioni di carico, possiamo concludere che il programma risulta essere una valida alternativa ai software commerciali a pagamento in quanto ci ha permesso di risolvere i casi studio esaminati.

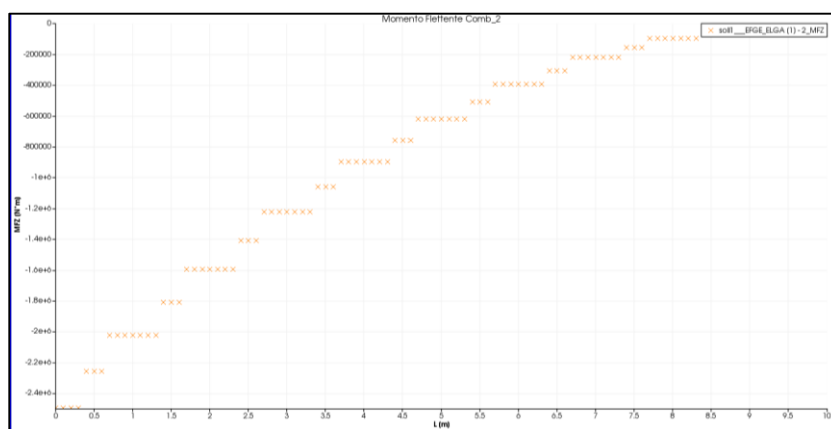
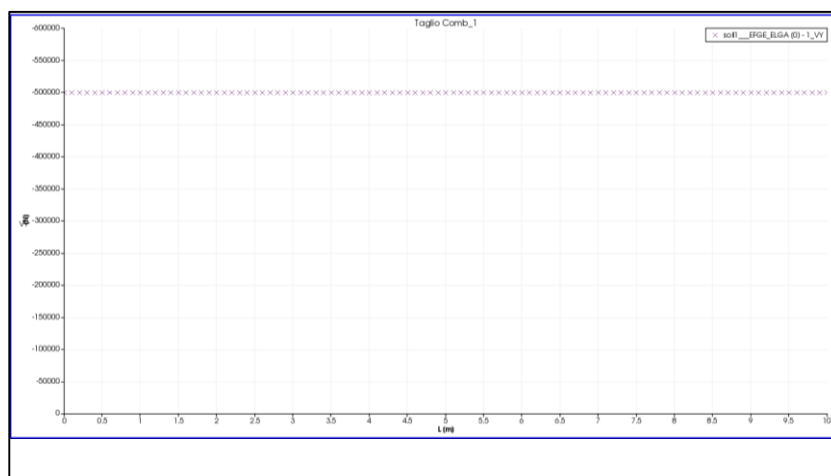
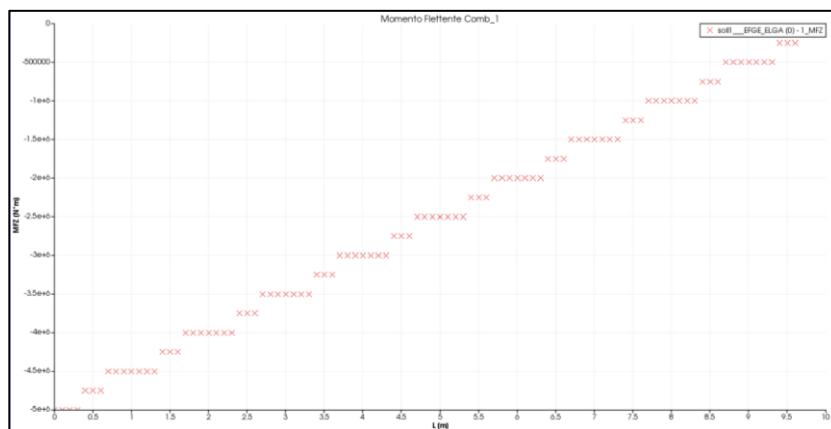
Le analisi condotte sono state affrontate in maniera critica, e sono state evidenziate alcune carenze: nella possibilità di incrementare il numero di combinazioni di carico, alcuni problemi legati alla visualizzazione grafica e alcuni problemi legati alla lettura della *Mesh*, tuttavia è stato possibile creare un file denominato “*Command File*” sviluppato tramite le funzioni messe a disposizione dalla libreria ufficiale di *Code-Aster* rendendo il file adattabile per ulteriori analisi su differenti elementi strutturali. E’ stato possibile sviluppare il codice arrivando a superare l’ostacolo riguardante le combinazioni dei carichi consentendo al software di poter eseguire i calcoli senza errori, o con un margine di errore accettabile. I valori ricavati sono stati verificati da un’analisi analitica eseguita tramite un foglio elettronico di Excel che ne ha confermato la correttezza; inoltre all’interno della stessa analisi analitica viene effettuata una verifica per mezzo di diagrammi di interazione.

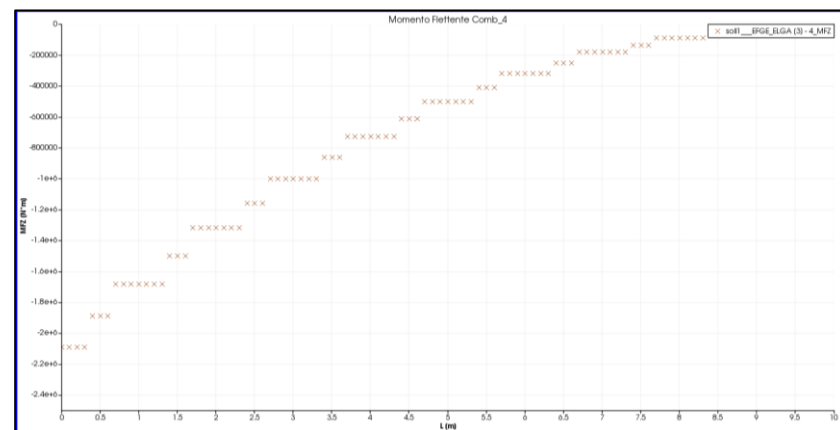
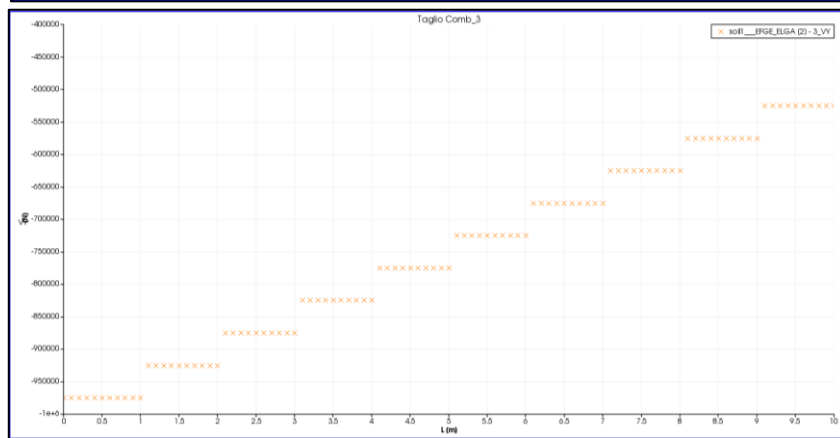
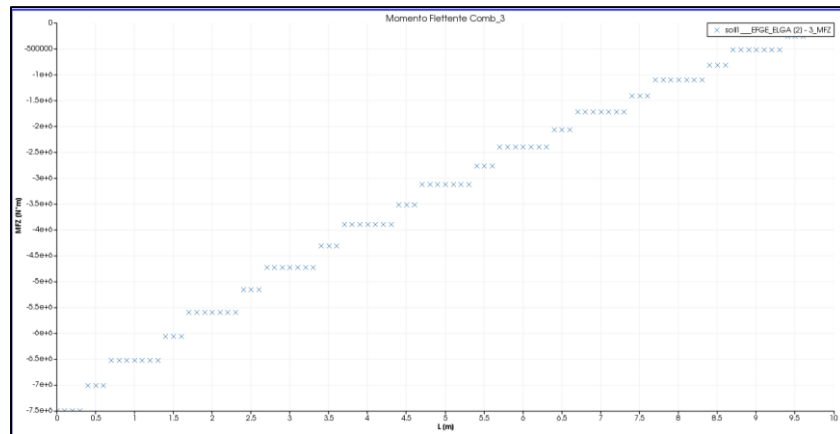
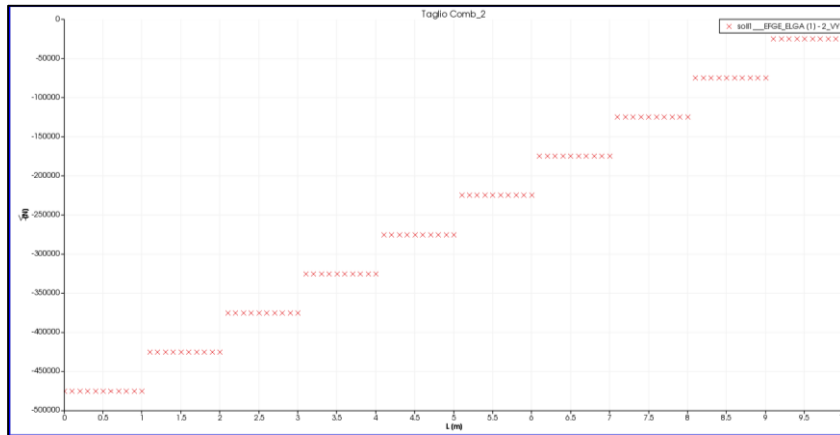
Il codice creato potrà essere impiegato su qualunque struttura in quanto è stato generato in forma parametrica consentendo a qualunque utente di utilizzarlo immettendo i propri dati di Input relativi alla geometria e ai carichi agenti sulla struttura.

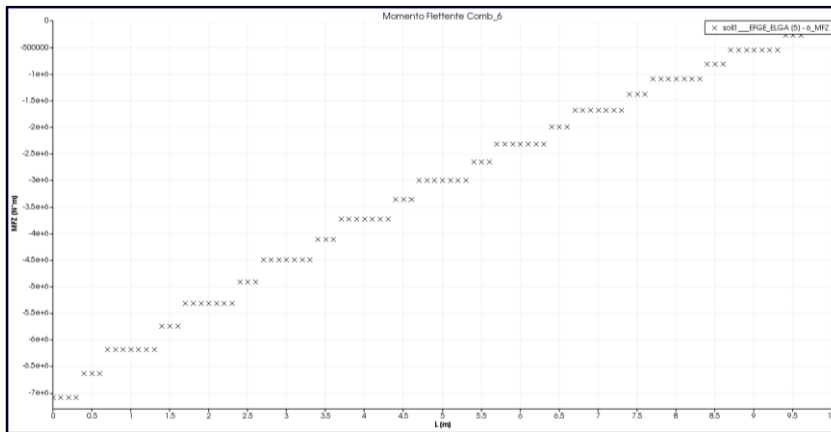
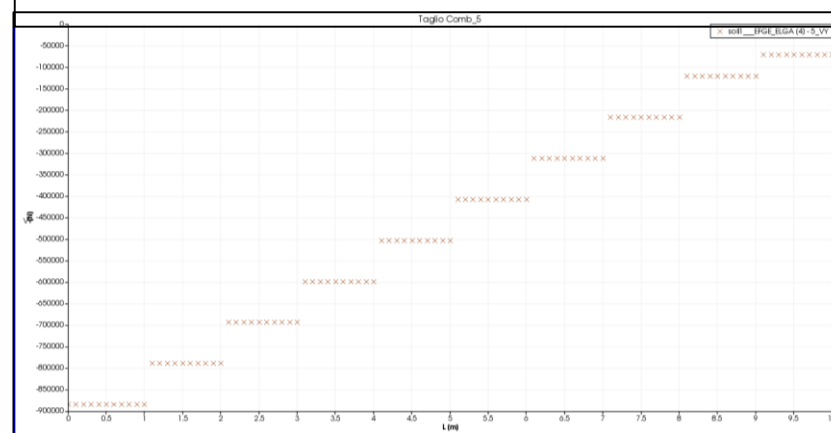
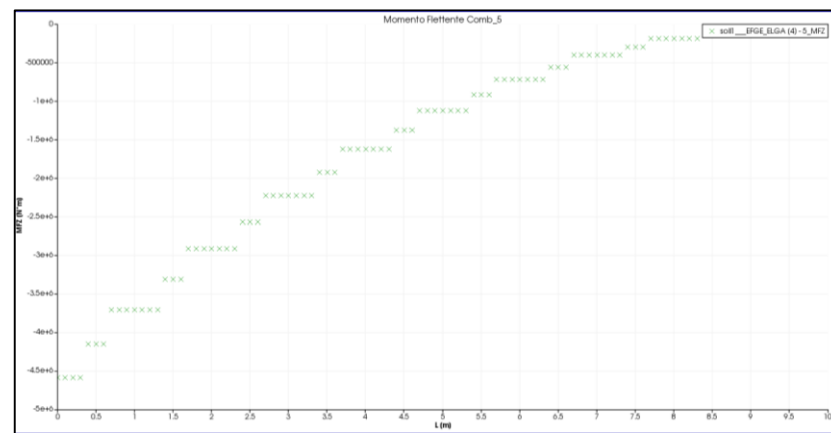
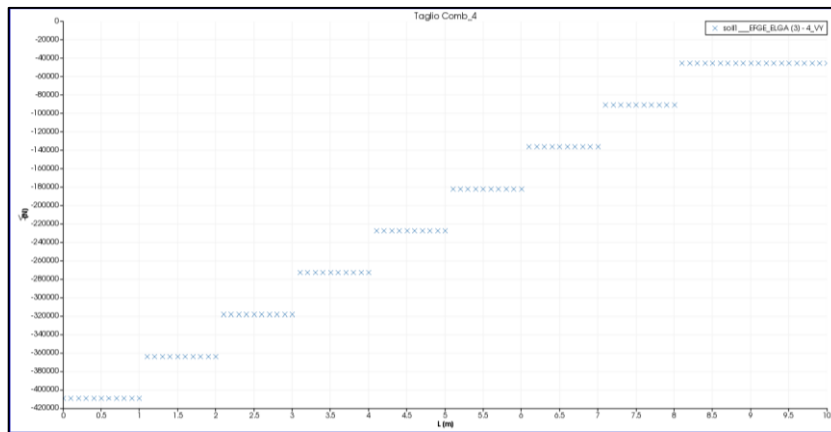
Nonostante il software durante il suo utilizzo abbia presentato limiti e difficoltà di esecuzione, rimane un programma gratuito con grandi potenzialità ed in costante sviluppo; la possibilità di accedere alle funzioni di programmazione e la numerosa disponibilità di strumenti a disposizione consente di sviluppare analisi strutturali di vario genere e complessità. In questo modo si spera di aver fornito al lettore le informazioni necessarie a condurre autonomamente un’analisi strutturale per mezzo di *Code-Aster*.

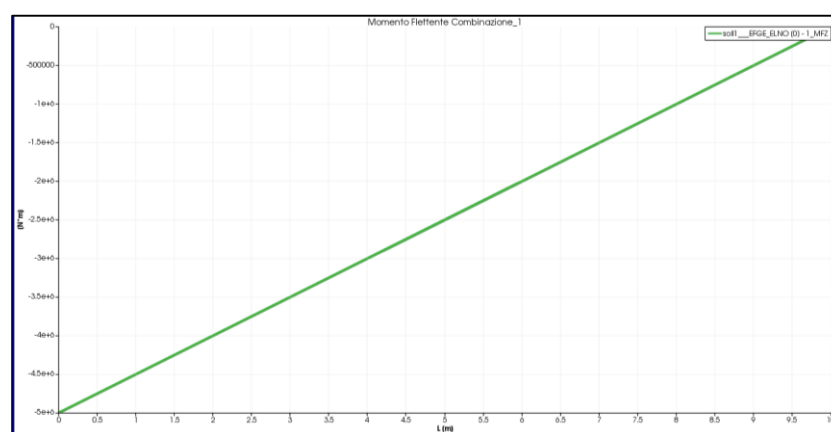
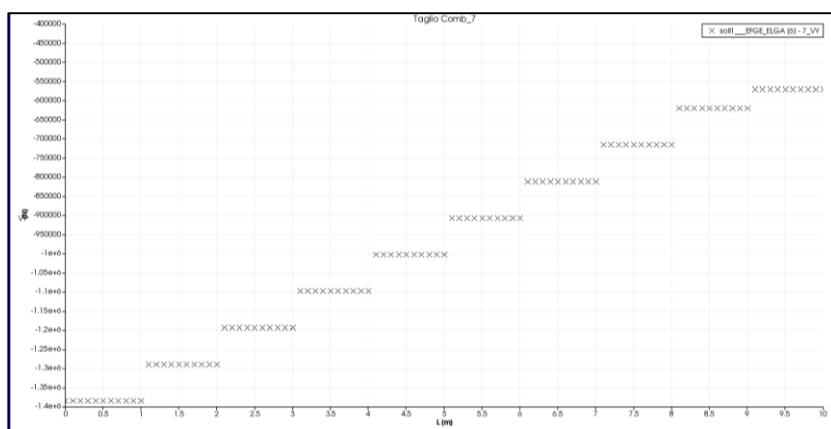
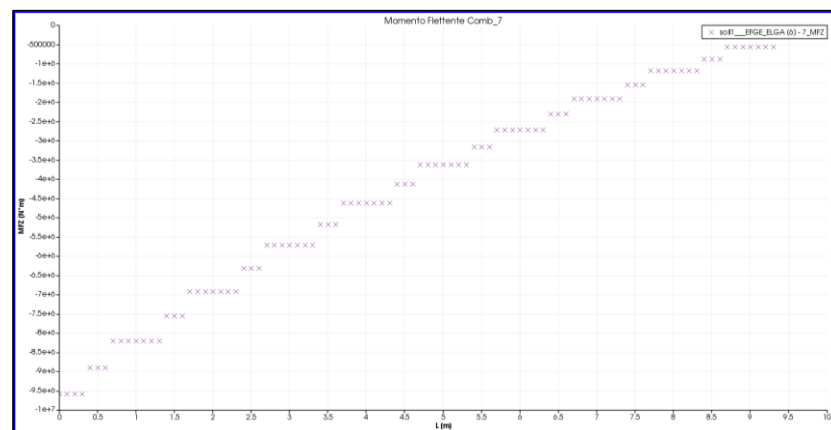
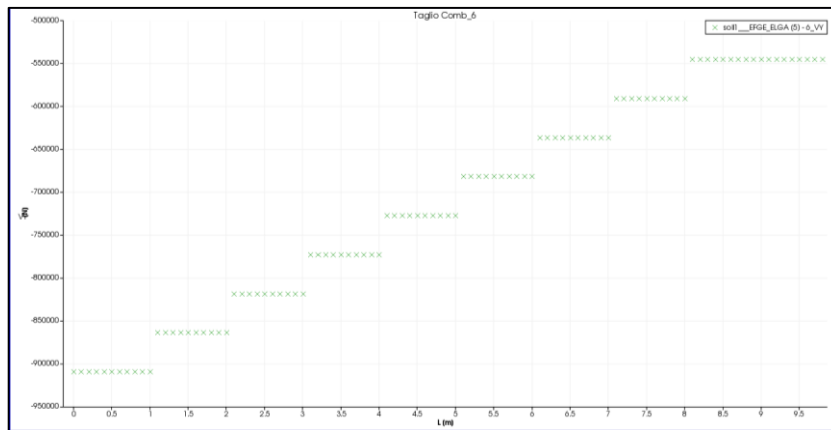
8 Allegati

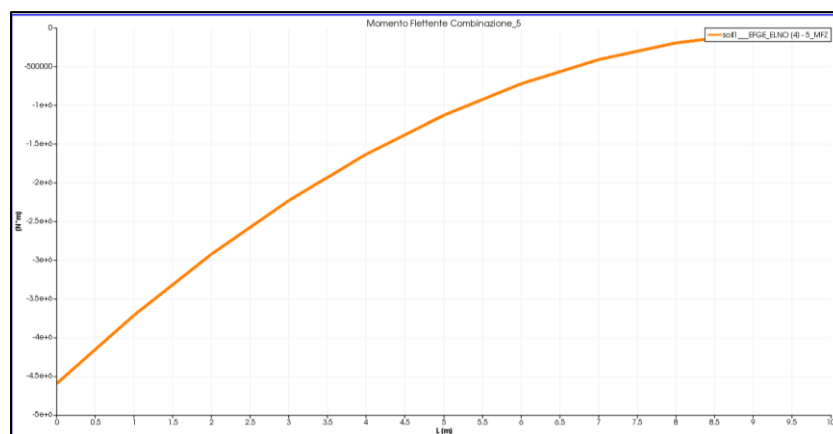
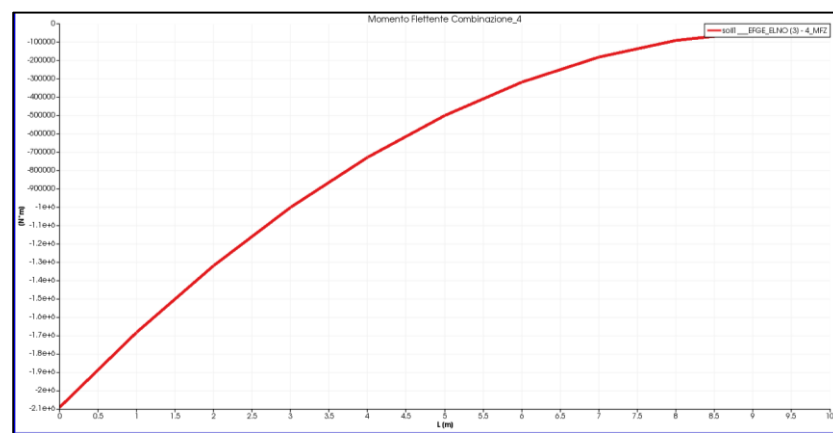
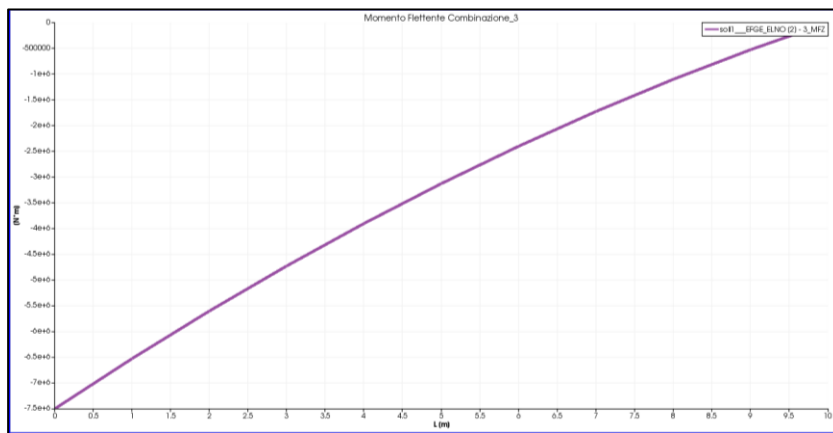
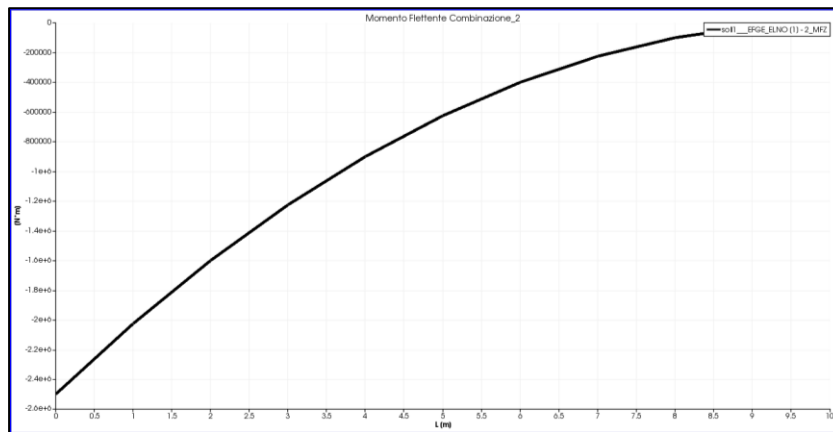
8.1 Mensola soggetta a carichi verticali: grafici

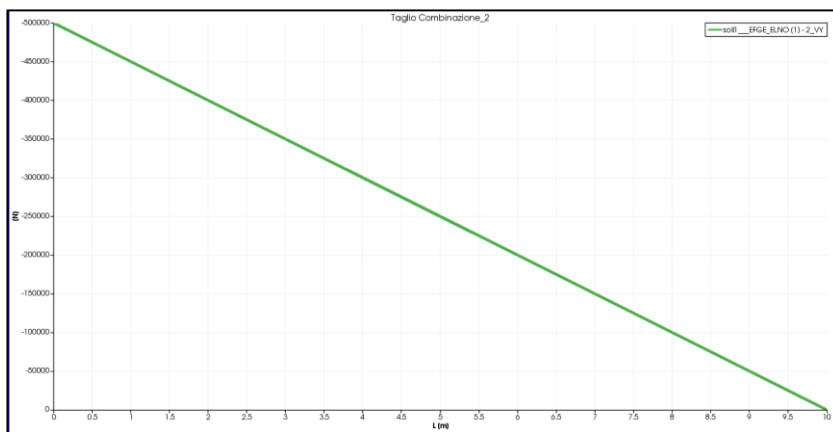
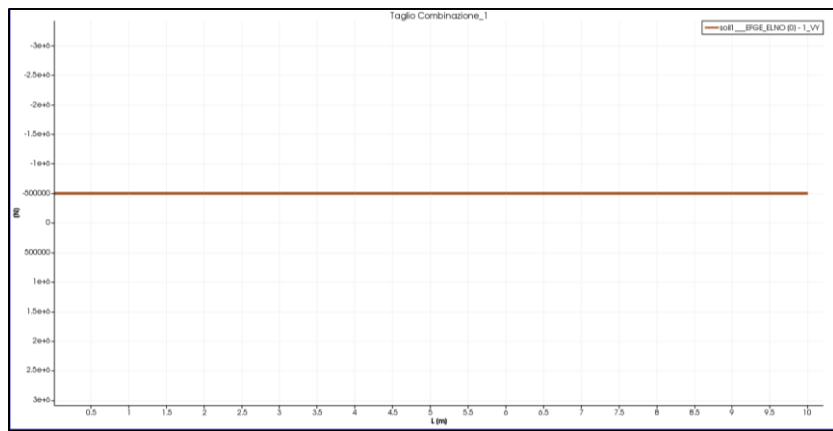
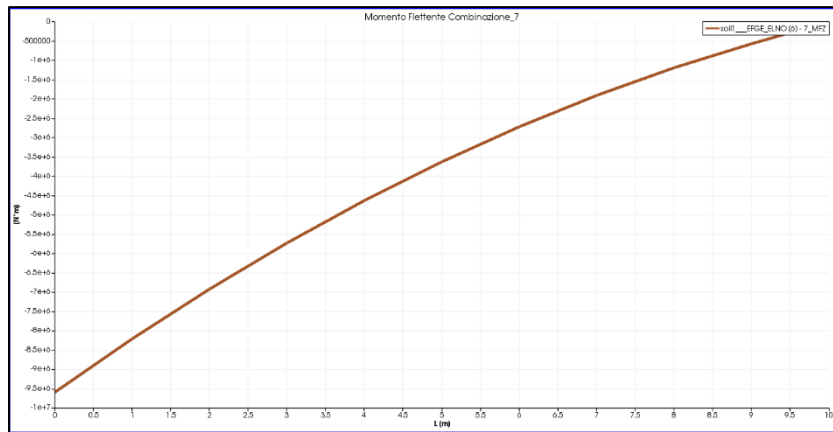
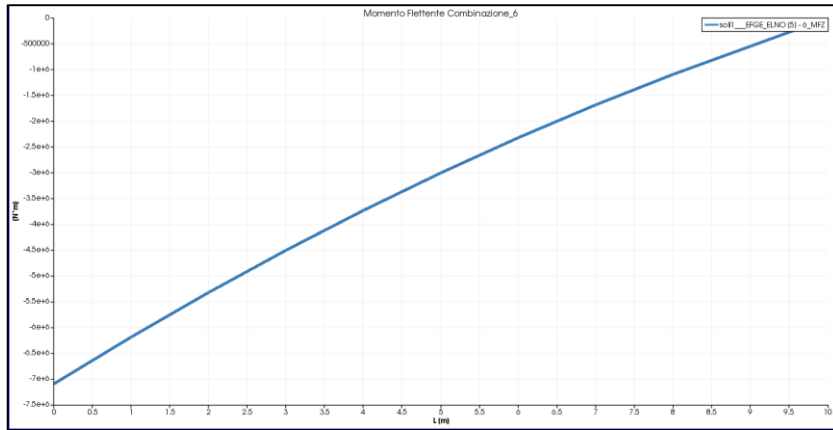


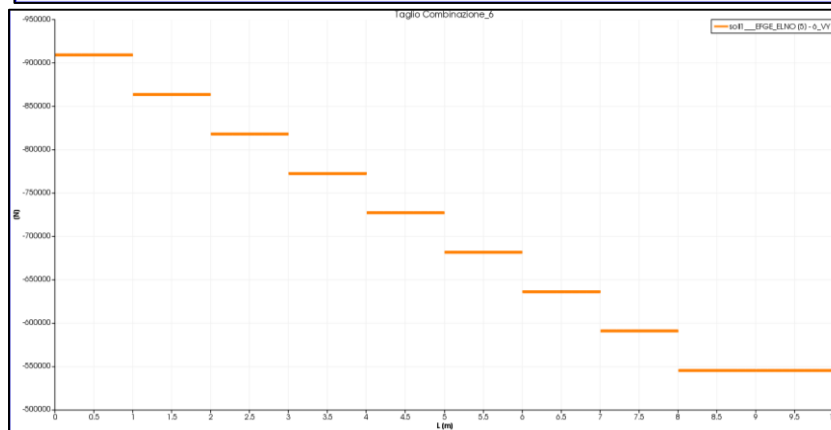
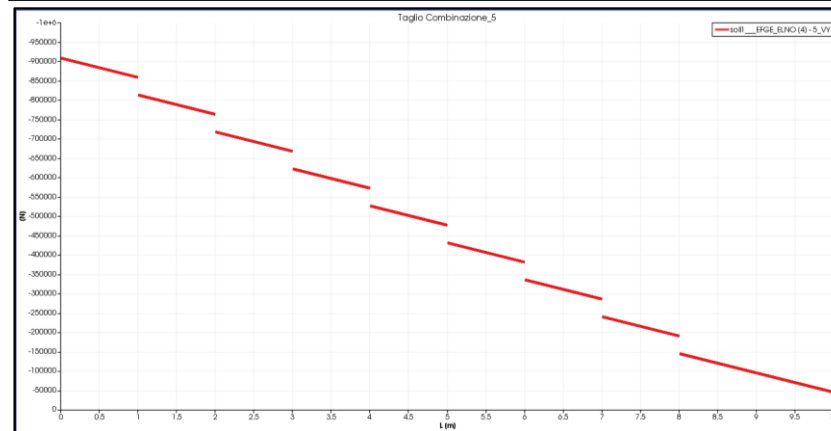
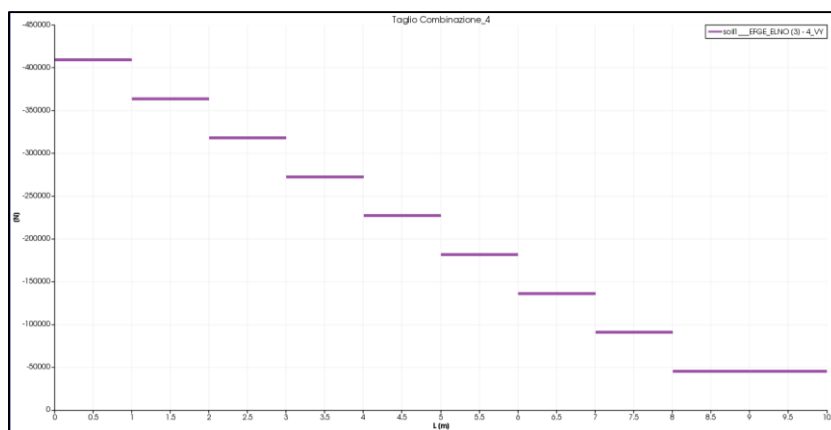
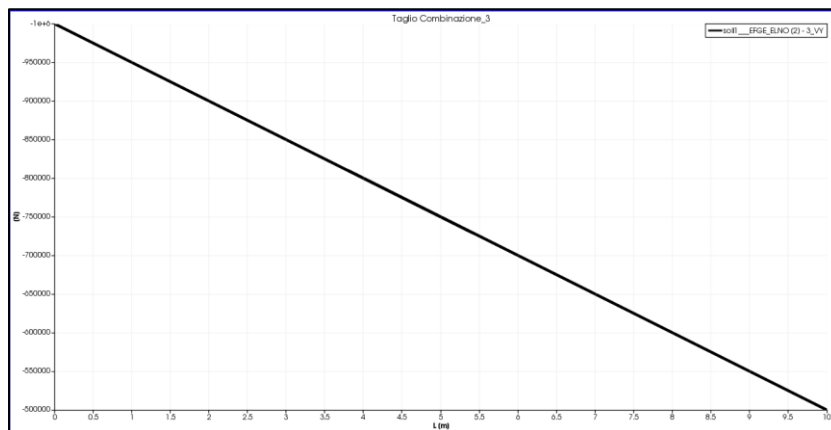


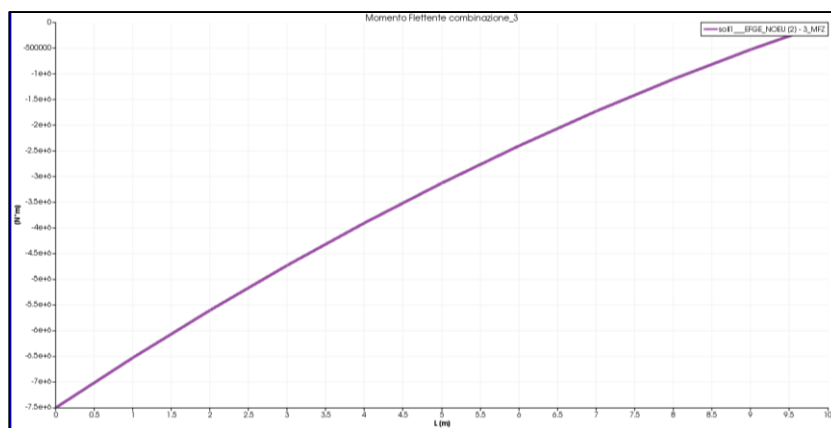
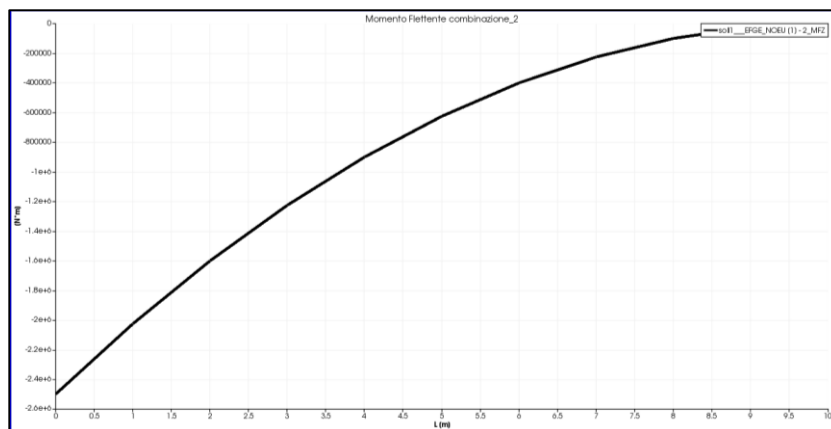
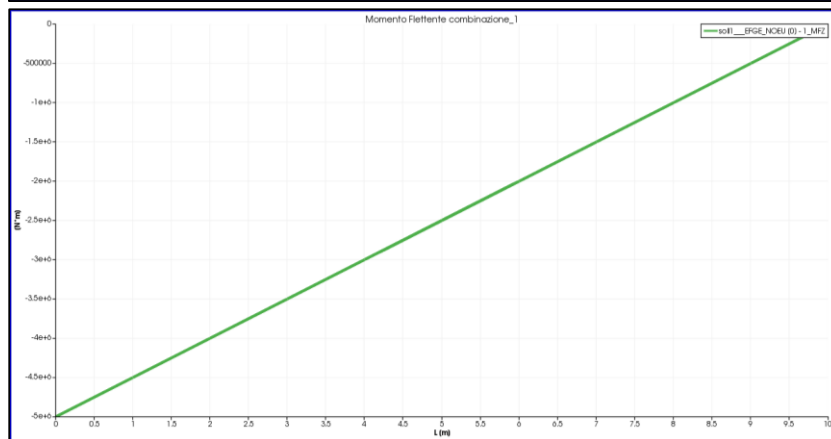
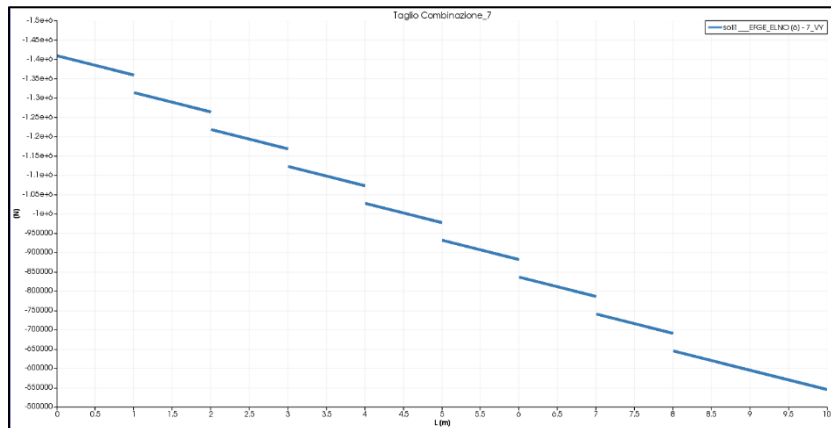


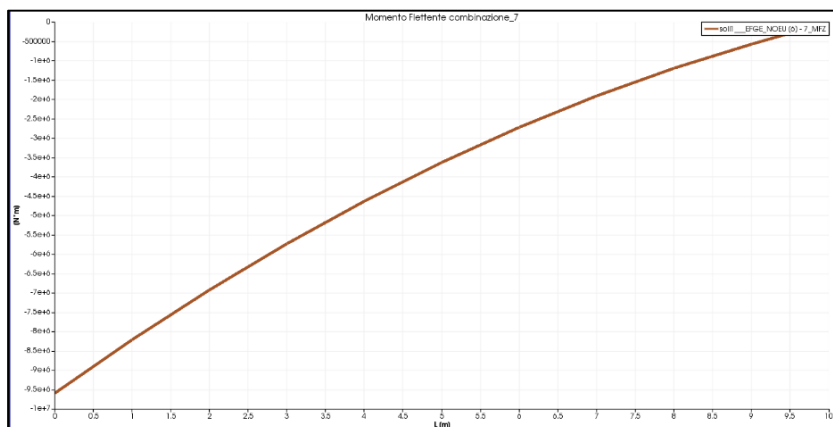
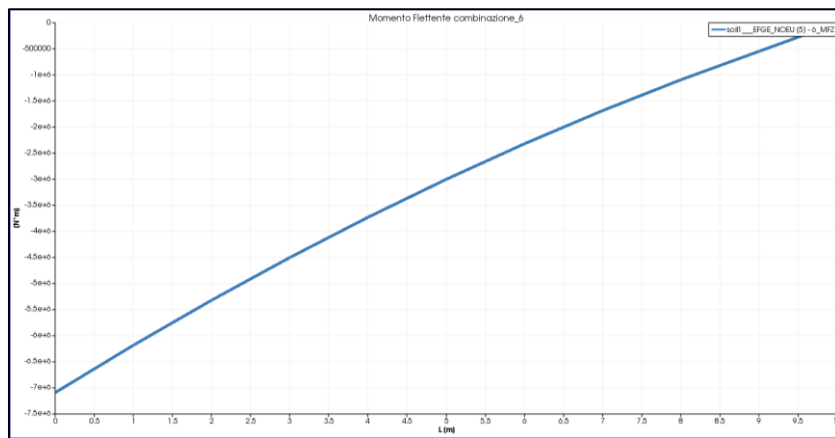
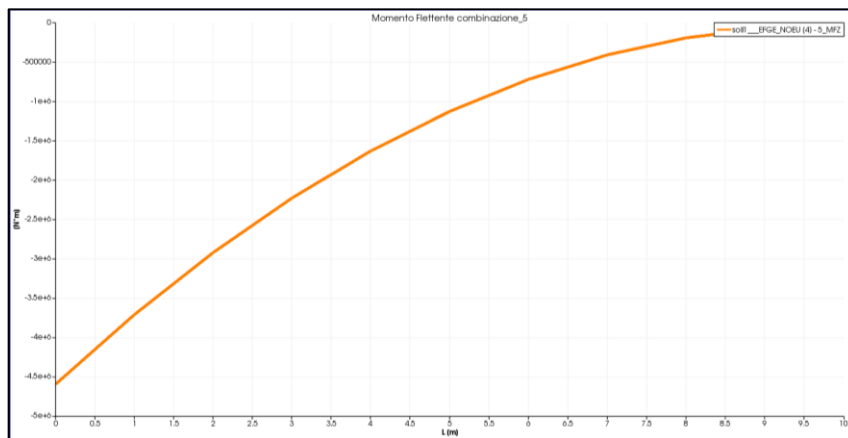
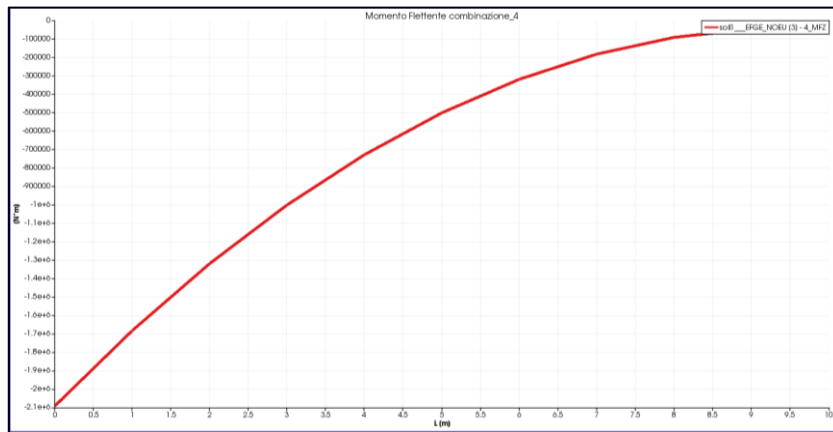


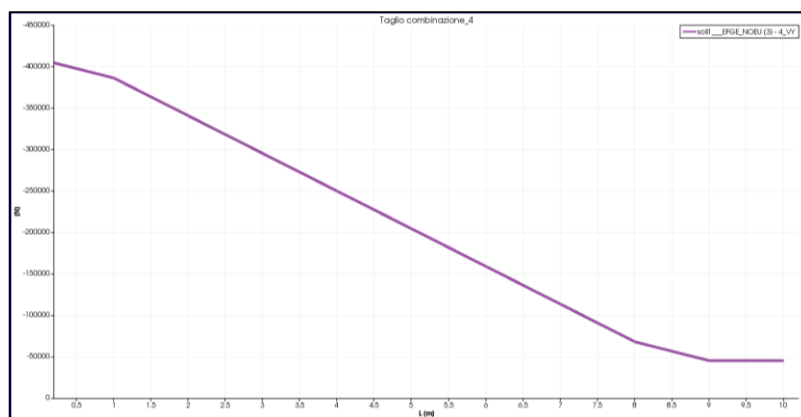
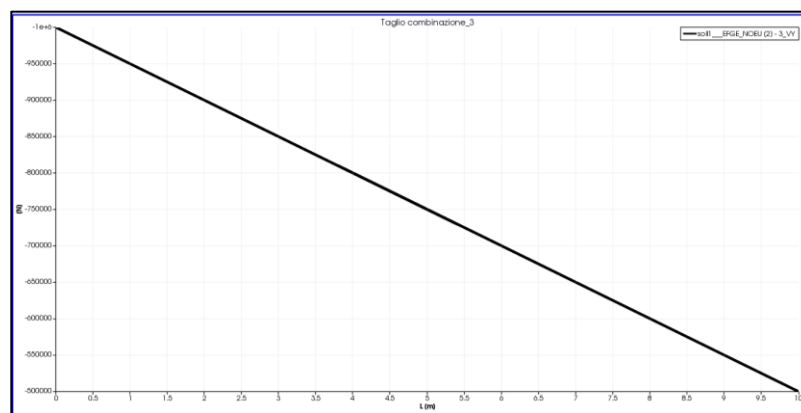
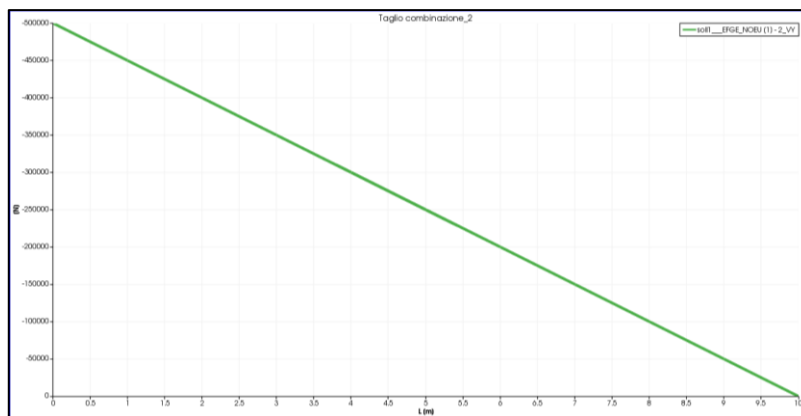
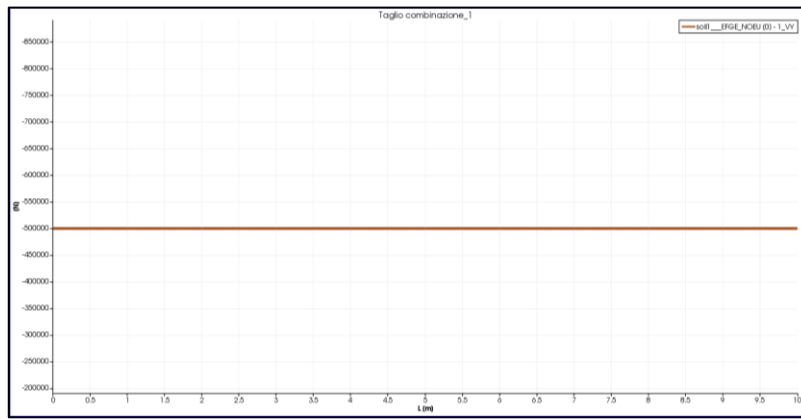


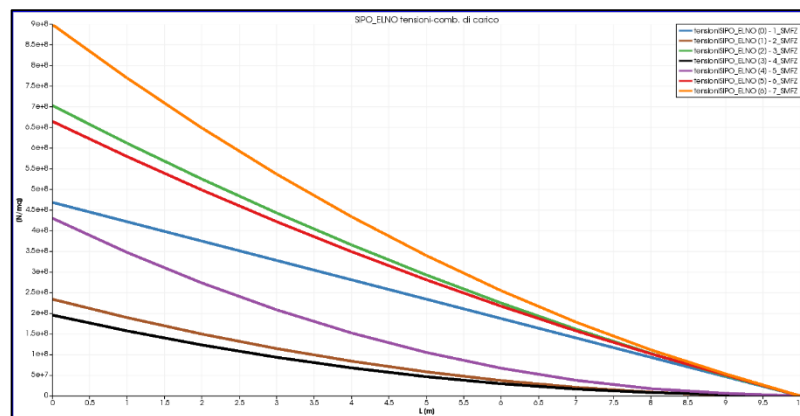
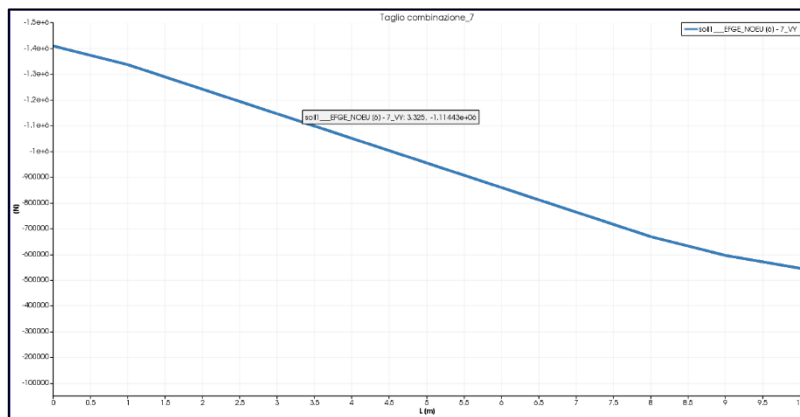
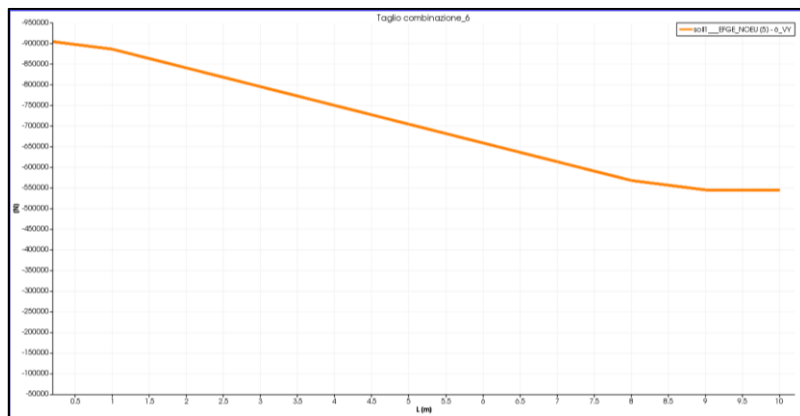
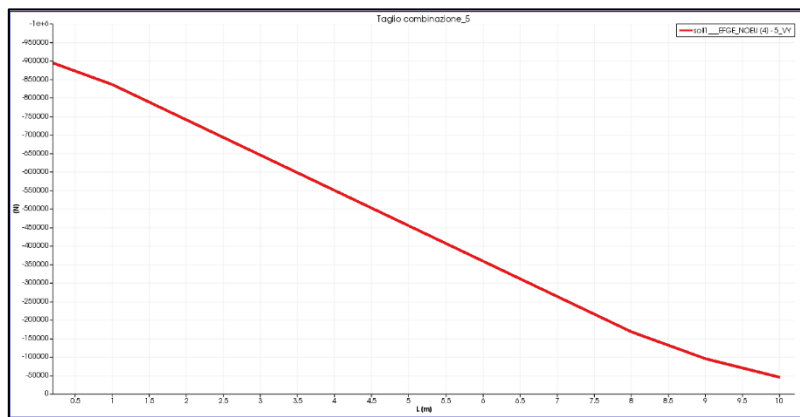


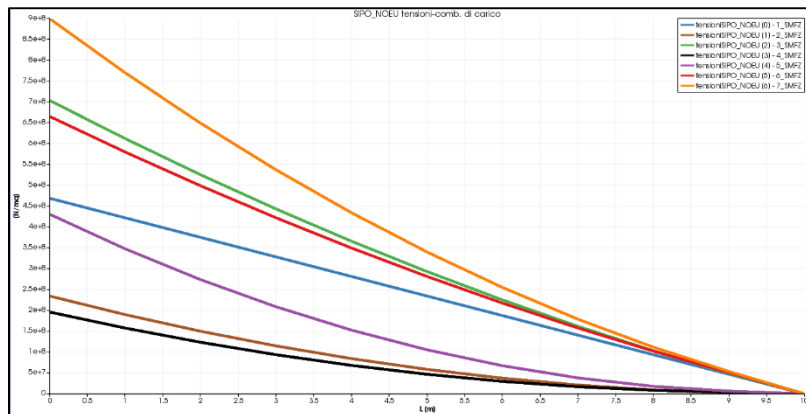












8.2 Mensola soggetta a carichi verticali: “Command File”

```
# Altezza e Larghezza della sezione [m]
h = 0.4

b = 0.4

fck = 25

# Modulo Elastico medio del calcestruzzo
E = pow(((fck + 8) / 10.0), 0.3) * 22000000000.0

# Definizione dei coefficienti parziali (paragr. 2.6.1 NTC 2018)
g_glf = 1.0

g_gls = 1.3

g_g2f = 0.0

g_g2s = 1.5

g_qf = 0.0

g_qs = 1.5

# Definizione dei coefficienti di combinazione (paragrafo 2.5.3 NTC
2018)
# Categoria A, Ambienti ad uso residenziale
psi_0A = 0.7

psi_1A = 0.5

psi_2A = 0.3

# Categoria B, uffici
psi_0B = 0.7
```

```

psi_1B = 0.5

psi_2B = 0.3

# Categoria C, ambienti suscettibili di affollamento
psi_0C = 0.7

psi_1C = 0.7

psi_2C = 0.6

# Categoria D, ambienti ad uso commerciale
psi_0D = 0.7

psi_1D = 0.7

psi_2D = 0.6

# Categoria E, biblioteche, archivi, magazzini e ambienti ad uso
industriale
psi_0E = 1.0

psi_1E = 0.9

psi_2E = 0.8

# Categoria F, rimesse e parcheggi (per autoveicoli di peso <= 30
kN)
psi_0F = 0.7

psi_1F = 0.7

psi_2F = 0.6

# Categoria G, Rimesse e parcheggi (per autoveicoli di peso > 30 kN)
psi_0G = 0.7

psi_1G = 0.5

psi_2G = 0.3

# Categoria H, Coperture
psi_0H = 0.0

psi_1H = 0.0

psi_2H = 0.0

# Vento
psi_0V = 0.6

psi_1V = 0.2

psi_2V = 0.0

```

```

# Neve (a quota <= 1000 m s.l.m.)
psi_0N1 = 0.5

psi_1N1 = 0.2

psi_2N1 = 0.0

# Neve (a quota > 1000 m s.l.m.)
psi_0N2 = 0.7

psi_1N2 = 0.5

psi_2N2 = 0.2

# Variazioni termiche
psi_0T = 0.6

psi_1T = 0.5

psi_2T = 0.0


# Definizione della costante di accelerazione gravitazionale
g = 9.81


# Definizione dei carichi agenti [Newton]
F1 = -500000.0

q1 = -50000.0

q2 = -50000.0

q3 = -3000

q4 = -6000


# Lunghezza_trave
L = 10.0

# num_nodi
n_nodi = 11

Q2var = (q2) * L / n_nodi

DEBUT()

mesh_1 = LIRE_MAILLAGE(FORMAT='MED', INFO_MED=2, NOM_MED='Mesh_1',
UNITE=11)

mesh_2 = LIRE_MAILLAGE(FORMAT='MED', INFO_MED=2, NOM_MED='Mesh_2',
UNITE=2)

mesh_3 = LIRE_MAILLAGE(FORMAT='MED', INFO_MED=2, NOM_MED='Mesh_3',
UNITE=20)

```

```

mesh_4 = LIRE_MAILLAGE(FORMAT='MED', INFO_MED=1,
NOM_MED='Mesh_face', UNITE=5)

model1 = AFFE_MODELE(
    AFFE=_F(MODELISATION=('POU_D_E', ), PHENOMENE='MECANIQUE',
TOUT='OUI'),
    MAILLAGE=mesh_1
)

model2 = AFFE_MODELE(
    AFFE=_F(MODELISATION=('POU_D_E', ), PHENOMENE='MECANIQUE',
TOUT='OUI'),
    MAILLAGE=mesh_2
)

model3 = AFFE_MODELE(
    AFFE=_F(MODELISATION=('POU_D_E', ), PHENOMENE='MECANIQUE',
TOUT='OUI'),
    MAILLAGE=mesh_3
)

elempro2 = AFFE_CARA_ELEM(
    MODELE=model2,
    ORIENTATION=_F(CARA='VECT_Y', GROUP_MA=('tr2', ), VALE=(0.0,
0.0, 1.0)),
    POUTRE=_F(
        CARA=('HY', 'HZ'),
        GROUP_MA=('tr2', ),
        SECTION='RECTANGLE',
        VALE=(0.4, 0.4)
    )
)

elempro3 = AFFE_CARA_ELEM(
    MODELE=model3,
    ORIENTATION=_F(CARA='VECT_Y', GROUP_MA=('tr3', ), VALE=(0.0,
0.0, 1.0)),
    POUTRE=_F(
        CARA=('HY', 'HZ'),
        GROUP_MA=('tr3', ),
        SECTION='RECTANGLE',
        VALE=(0.4, 0.4)
    )
)

elempro1 = AFFE_CARA_ELEM(
    MODELE=model1,
    ORIENTATION=_F(CARA='VECT_Y', GROUP_MA=('tr1', ), VALE=(0.0,
0.0, 1.0)),
    POUTRE=_F(
        CARA=('HY', 'HZ'),
        GROUP_MA=('tr1', ),
        SECTION='RECTANGLE',
        VALE=(0.4, 0.4)
    )
)

```



```

    )
)

mater = DEFI_MATERIAU(ELAS=_F(E=E, NU=0.3, RHO=2500.0))

fieldma1 = AFFE_MATERIAU(AFFE=_F(MATER=(mater, ), TOUT='OUI'),
MODELE=model1)

fieldma2 = AFFE_MATERIAU(
    AFFE=_F(MATER=(mater, ), TOUT='OUI'), MAILLAGE=mesh_2,
MODELE=model2
)

fieldma3 = AFFE_MATERIAU(
    AFFE=_F(MATER=(mater, ), TOUT='OUI'), MAILLAGE=mesh_3,
MODELE=model3
)

func_fz = DEFI_FONCTION(
    INTERPOL=('LIN', ),
    NOM_PARA='INST',
    PROL_DROITE='CONSTANT',
    PROL_GAUCHE='CONSTANT',
    TITRE='combinazioni F_var1',
    VALE=(1.0, 1.0, 2.0, 0.0, 3.0, 1.0, 4.0, 0.0, 5.0, 0.0, 6.0,
1.0, 7.0, 1.0)
)

func_q1 = DEFI_FONCTION(
    INTERPOL=('LIN', ),
    NOM_PARA='INST',
    PROL_DROITE='CONSTANT',
    PROL_GAUCHE='CONSTANT',
    TITRE='combinazioni q_var1',
    VALE=(1.0, 0.0, 2.0, 1.0, 3.0, 1.0, 4.0, 0.0, 5.0, 1.0, 6.0,
0.0, 7.0, 1.0)
)

func_q2 = DEFI_FONCTION(
    INTERPOL=('LIN', ),
    NOM_PARA='INST',
    PROL_DROITE='CONSTANT',
    PROL_GAUCHE='CONSTANT',
    TITRE='combinazioni q_var2',
    VALE=(1.0, 0.0, 2.0, 0.0, 3.0, 0.0, 4.0, 1.0, 5.0, 1.0, 6.0,
1.0, 7.0, 1.0)
)

listr = DEFI_LIST_REEL(
    DEBUT=1.0,
    INTERVALLE=_F(JUSQU_A=7.0, PAS=1.0),
    TITRE='combinazioni istanti'
)

bcl = AFFE_CHAR_MECA(

```

```

        DDL_IMPO=_F(GROUP_NO=('fix1', ), LIAISON='ENCASTRE'),
MODELE=model1
)

Forza1 = AFFE_CHAR_MECA(
    FORCE_NODALE=_F(FZ=F1, GROUP_NO=('pload1', )), MODELE=model1
)

qvar_1 = AFFE_CHAR_MECA(
    FORCE_POUTRE=_F(FZ=q1, GROUP_MA=('tr1', )), MODELE=model1
)

QVAR_2 = AFFE_CHAR_MECA(
    FORCE_NODALE=_F(FZ=Q2var, GROUP_NO=('Group_1', )), MODELE=model1
)

caricoq2 = AFFE_CHAR_MECA(
    FORCE_POUTRE=_F(FZ=-50000.0, GROUP_MA=('tr2', )), MODELE=model2
)

forzafz2 = AFFE_CHAR_MECA(
    FORCE_NODALE=_F(FZ=-500000.0, GROUP_NO=('pload2', )),
MODELE=model2
)

caricoq3 = AFFE_CHAR_MECA(
    FORCE_POUTRE=_F(FX=50000.0, GROUP_MA=('tr3', )), MODELE=model3
)

forzafz3 = AFFE_CHAR_MECA(
    FORCE_NODALE=_F(FX=500000.0, GROUP_NO=('pload3', )),
MODELE=model3
)

bc2 = AFFE_CHAR_MECA(
    DDL_IMPO=_F(GROUP_NO=('fix2', ), LIAISON='ENCASTRE'),
MODELE=model2
)

bc3 = AFFE_CHAR_MECA(
    DDL_IMPO=_F(GROUP_NO=('fix3', ), LIAISON='ENCASTRE'),
MODELE=model3
)

reslin1 = MECA_STATIQUE(
    CARA_ELEM=elempro1,
    CHAM_MATER=fieldma1,
    EXCIT=(
        _F(CHARGE=bc1), _F(CHARGE=Forza1, FONC_MULT=func_fz), _F(
            CHARGE=qvar_1, FONC_MULT=func_q1
        ), _F(CHARGE=QVAR_2, FONC_MULT=func_q2)
    ),
    LIST_INST=listr,
    MODELE=model1
)

```

```

sol11 = CALC_CHAMP(
    CARA_ELEM=elempro1,
    CONTRAINTE=('EFGE_NOEU', 'EFGE_ELGA', 'EFGE_ELNO'),
    LIST_INST=listr,
    MODELE=modell1,
    RESULTAT=reslin1
)

tensioni = CALC_CHAMP(
    CARA_ELEM=elempro1,
    CHAM_MATER=fieldmal,
    CONTRAINTE=('SIPO_ELNO', 'SIPM_ELNO', 'SIPO_NOEU'),
    LIST_INST=listr,
    MODELE=modell1,
    RESULTAT=reslin1
)

IMPR_RESU(
    FORMAT='MED',
    RESU=_F(
        CARA_ELEM=elempro1, LIST_INST=listr, MAILLAGE=mesh_1,
    RESULTAT=reslin1
    ),
    UNITE=3
)

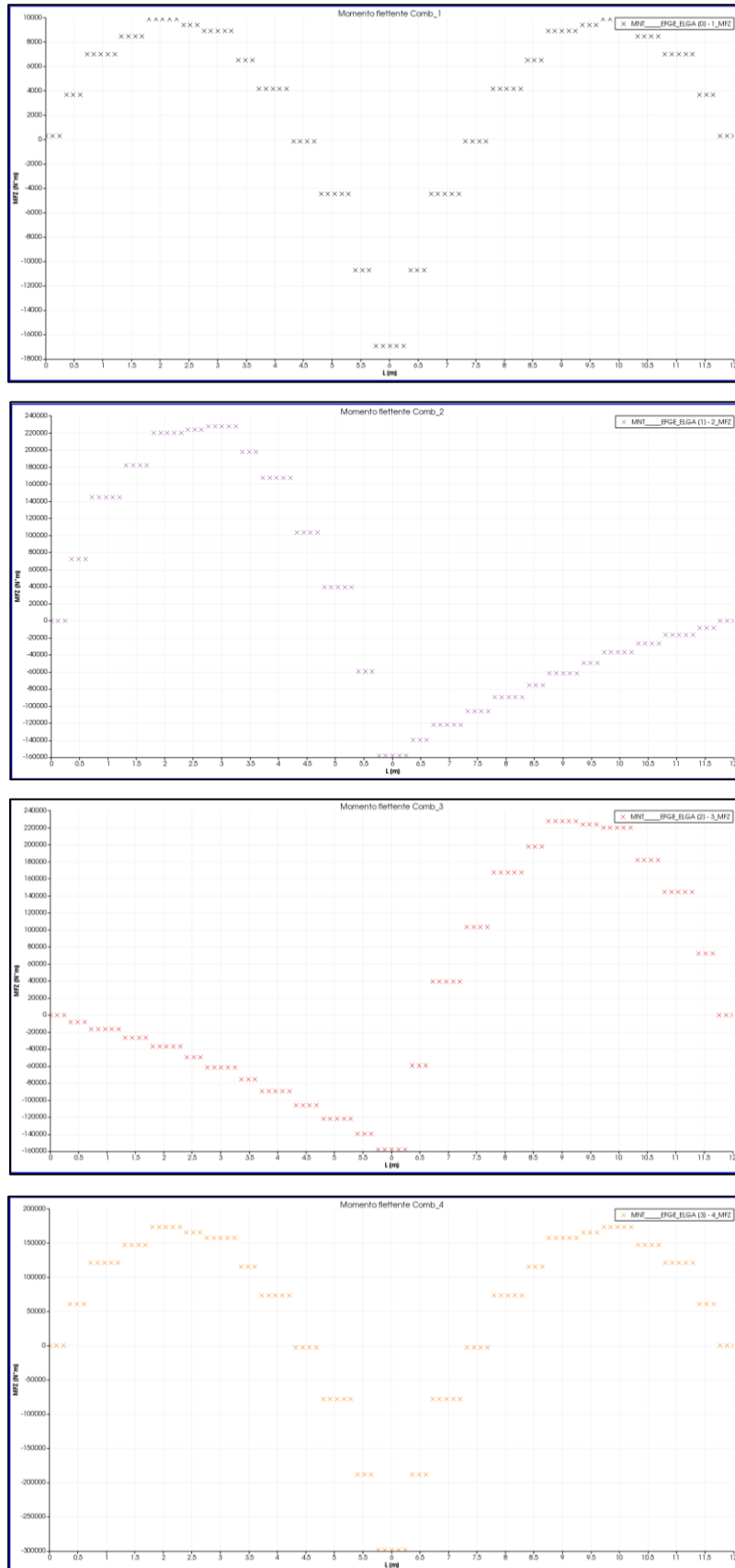
IMPR_RESU(
    FORMAT='MED',
    RESU=_F(
        CARA_ELEM=elempro1,
        LIST_INST=listr,
        MAILLAGE=mesh_1,
        RESULTAT=sol11,
        TOUT_CHAM='OUI'
    ),
    UNITE=80
)

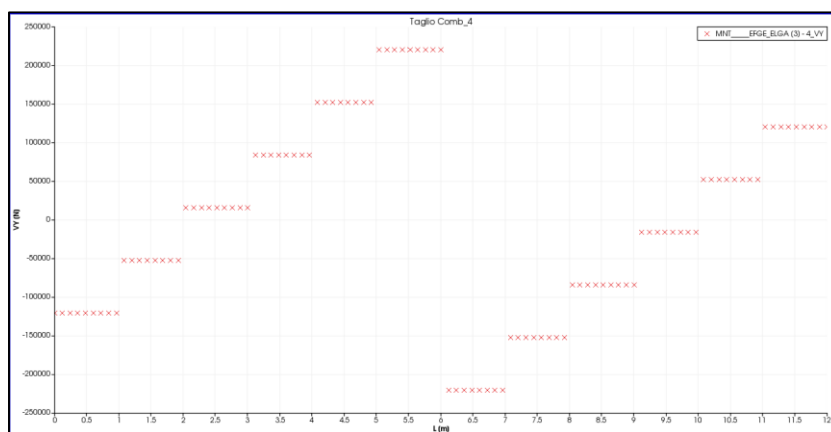
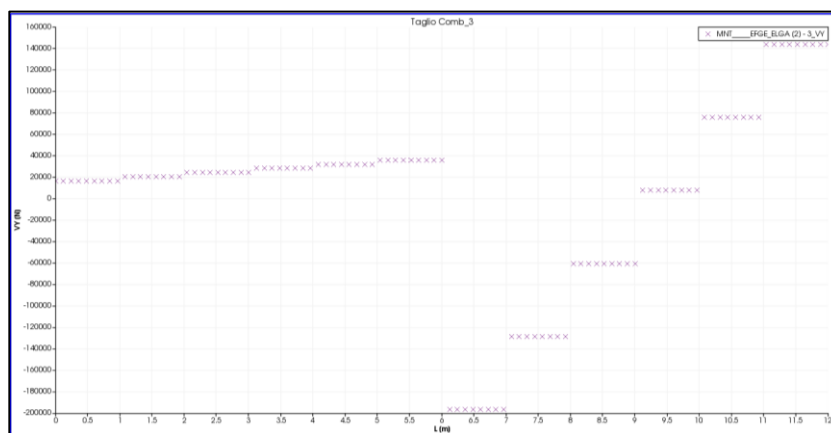
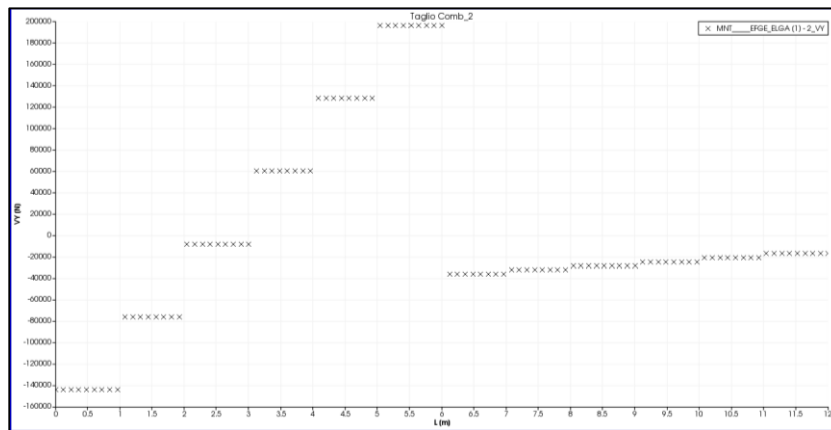
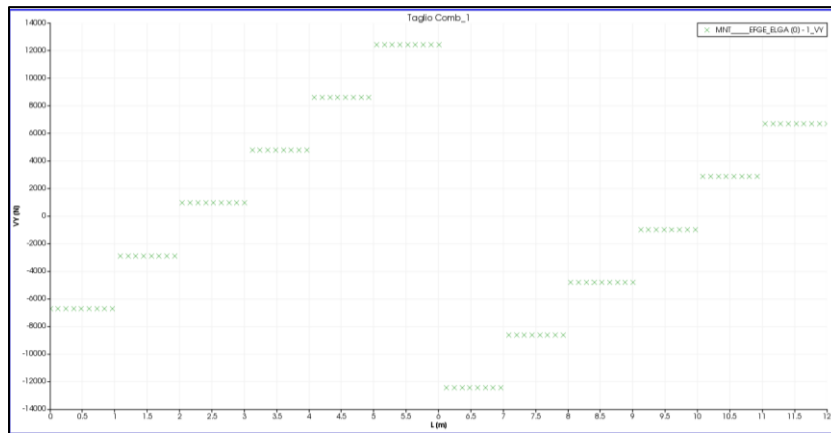
IMPR_RESU(
    FORMAT='MED',
    RESU=_F(
        CARA_ELEM=elempro1,
        LIST_INST=listr,
        MAILLAGE=mesh_1,
        RESULTAT=tensioni,
        TOUT_CHAM='OUI'
    ),
    UNITE=4
)

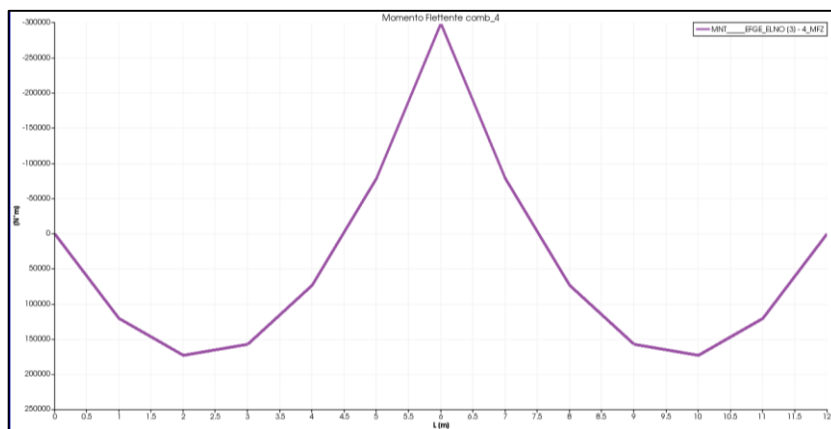
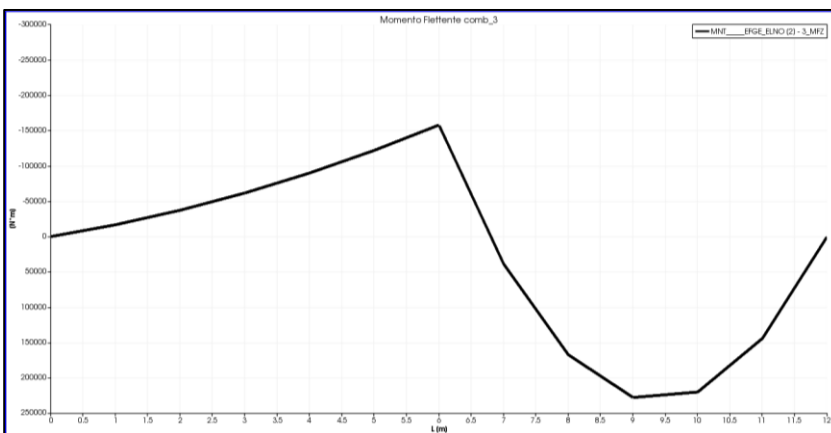
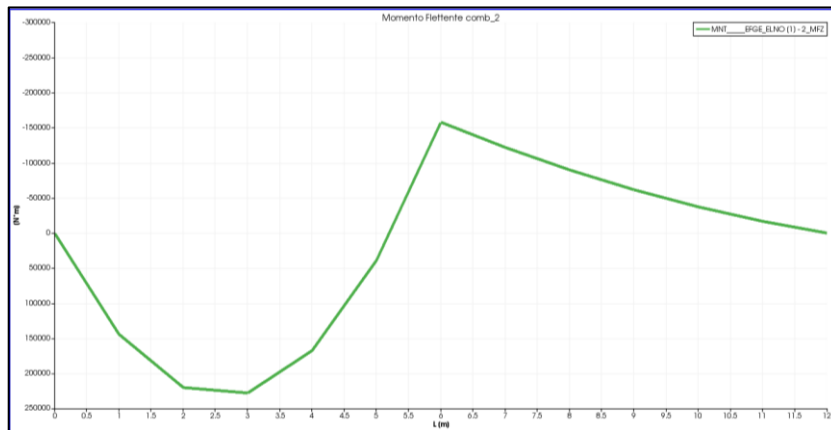
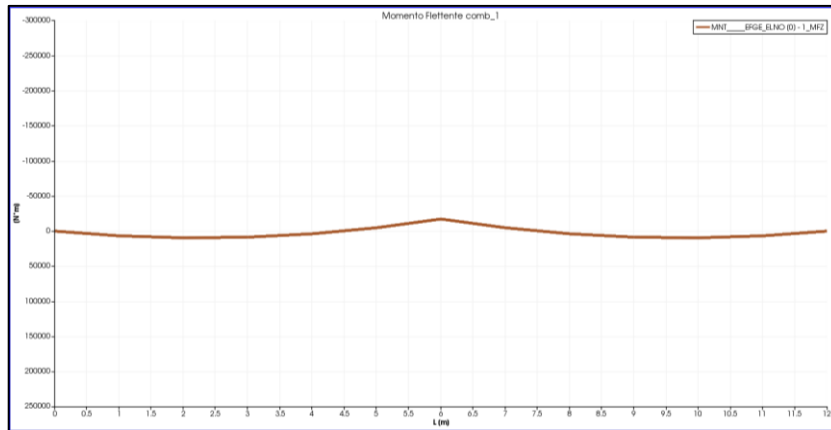
FIN()

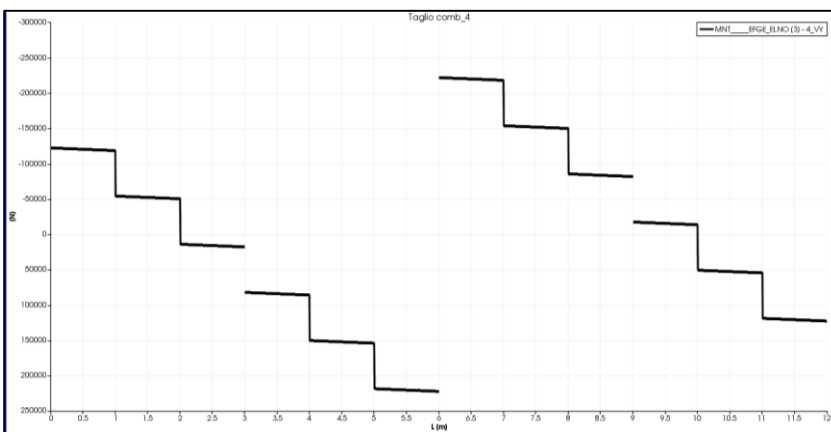
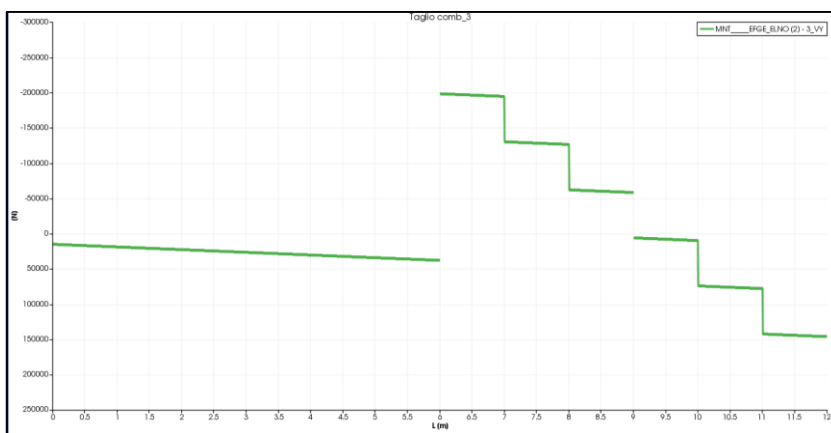
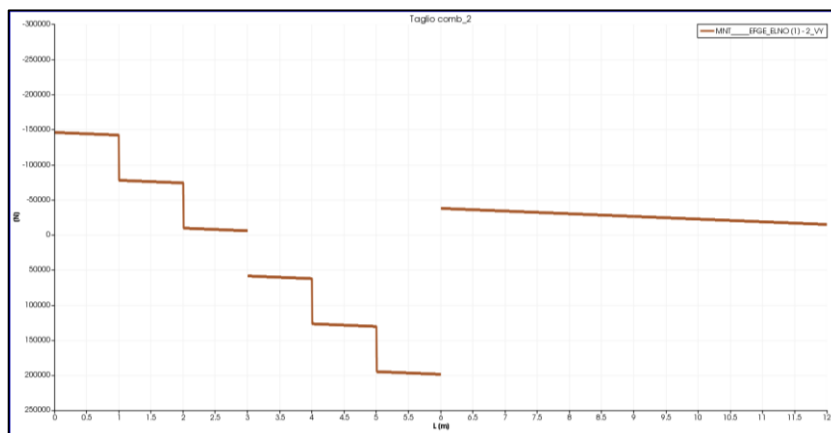
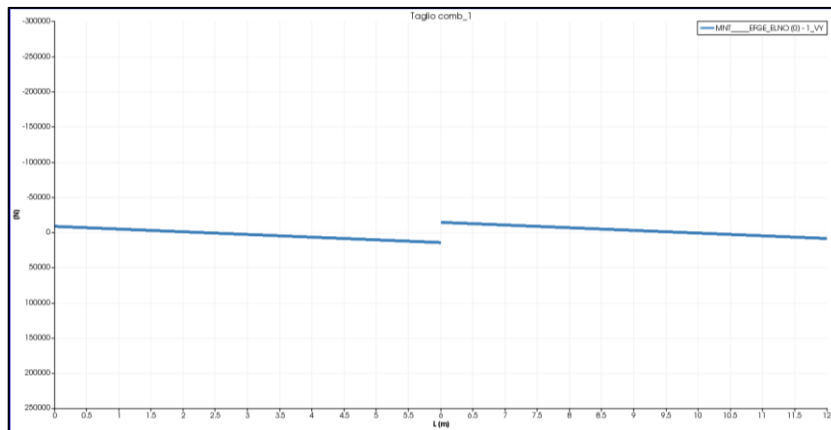
```

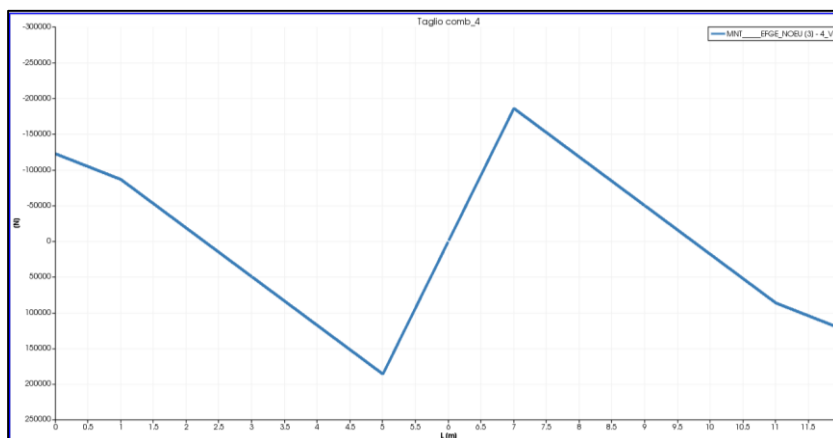
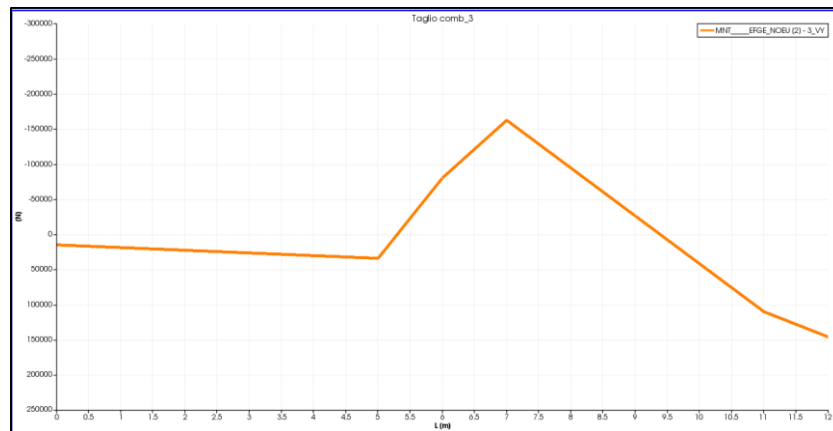
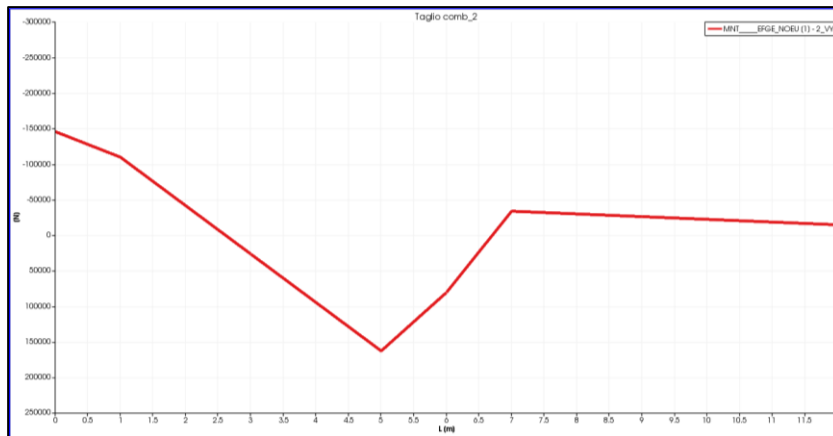
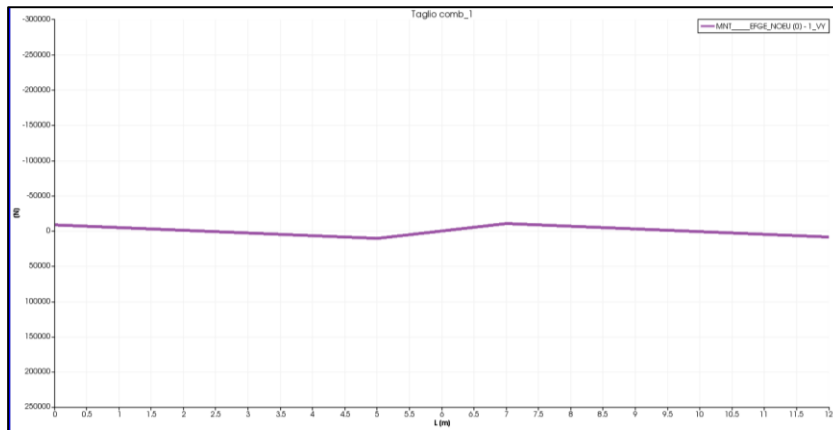
8.3 Trave a due campate soggetta a carichi verticali: grafici

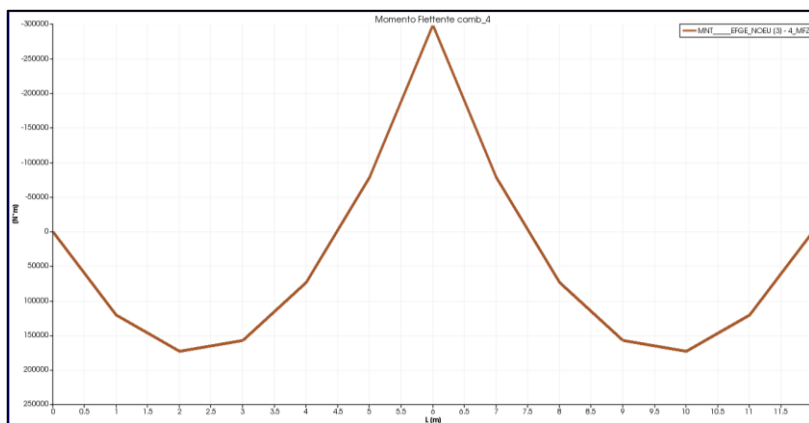
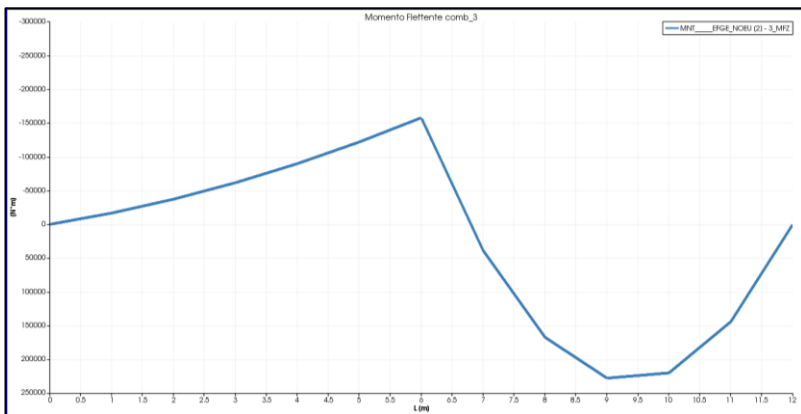
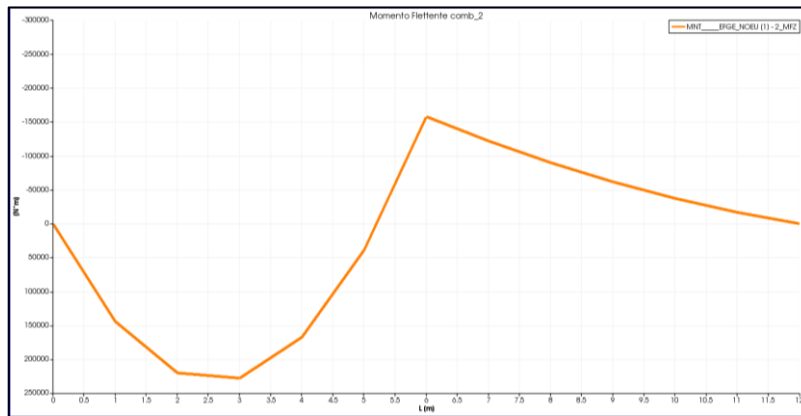
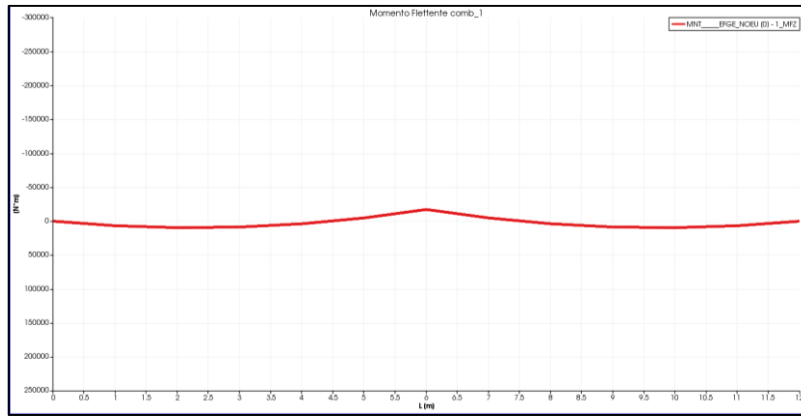


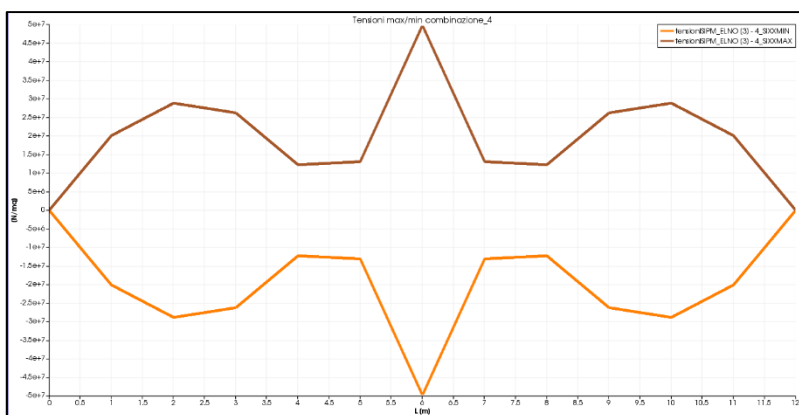
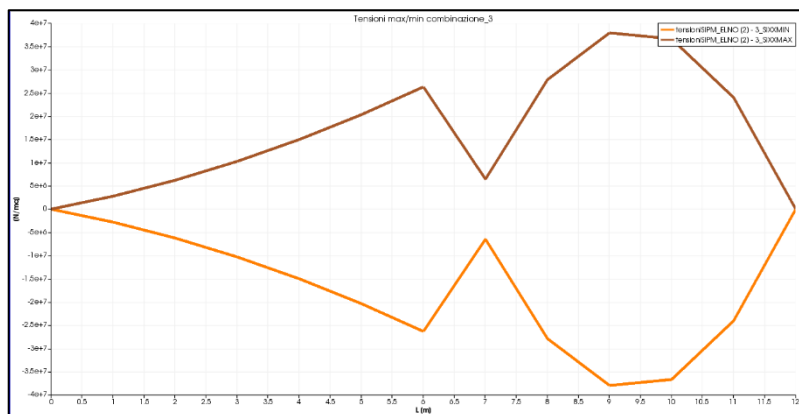
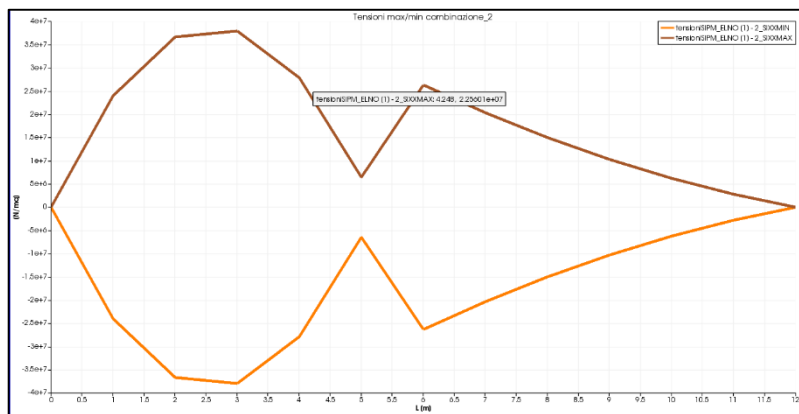
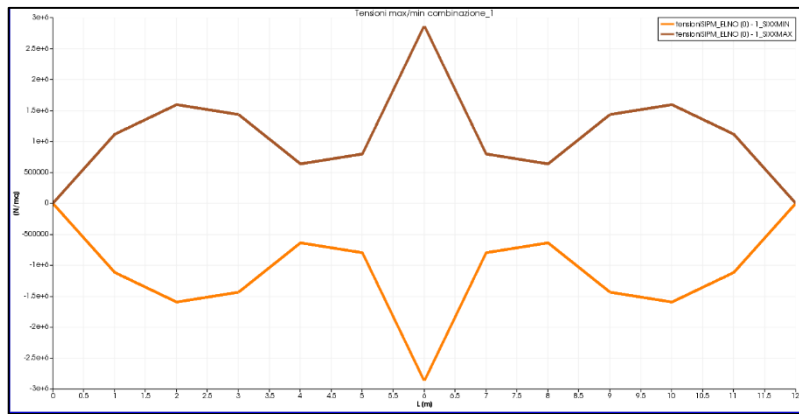












8.4 Trave a due campate soggetta a carichi verticali: “Command File”

```
# Altezza e Larghezza della sezione [m]
h = 0.4

b = 0.3

fck = 25

# Modulo Elastico medio del calcestruzzo
E = pow(((fck + 8) / 10.0), 0.3) * 22000000000.0

# Definizione dei coefficienti parziali (paragr. 2.6.1 NTC 2018)
g_glf = 1.0

g_gls = 1.3

g_g2f = 0.0

g_g2s = 1.5

g_qf = 0.0

g_qs = 1.5

# Definizione dei coefficienti di combinazione (paragrafo 2.5.3 NTC
2018)
# Categoria A, Ambienti ad uso residenziale
psi_0A = 0.7

psi_1A = 0.5

psi_2A = 0.3

# Categoria B, uffici
psi_0B = 0.7

psi_1B = 0.5

psi_2B = 0.3

# Categoria C, ambienti suscettibili di affollamento
psi_0C = 0.7

psi_1C = 0.7

psi_2C = 0.6

# Categoria D, ambienti ad uso commerciale
psi_0D = 0.7

psi_1D = 0.7
```

```

psi_2D = 0.6

# Categoria E, biblioteche, archivi, magazzini e ambienti ad uso
industriale
psi_0E = 1.0

psi_1E = 0.9

psi_2E = 0.8

# Categoria F, rimesse e parcheggi (per autoveicoli di peso <= 30
kN)
psi_0F = 0.7

psi_1F = 0.7

psi_2F = 0.6

# Categoria G, Rimesse e parcheggi (per autoveicoli di peso > 30 kN)
psi_0G = 0.7

psi_1G = 0.5

psi_2G = 0.3

# Categoria H, Coperture
psi_0H = 0.0

psi_1H = 0.0

psi_2H = 0.0

# Vento
psi_0V = 0.6

psi_1V = 0.2

psi_2V = 0.0

# Neve (a quota <= 1000 m s.l.m.)
psi_0N1 = 0.5

psi_1N1 = 0.2

psi_2N1 = 0.0

# Neve (a quota > 1000 m s.l.m.)
psi_0N2 = 0.7

psi_1N2 = 0.5

psi_2N2 = 0.2

# Variazioni termiche
psi_0T = 0.6

```

```

psi_1T = 0.5

psi_2T = 0.0

# Definizione della costante di accelerazione gravitazionale
g = 9.81

# Definizione dei carichi agenti [Newton]

q1 = -50000.0

q2 = -50000.0

q3 = 0

q4 = 0

# Lunghezza_trave
L = 6.0

# num_nodi
n_nodi = 7

Q2var = ((q2) * L) / n_nodi

Q1var = ((q1) * L) / n_nodi

DEBUT()

mesh = LIRE_MALLAGE(FORMAT='MED', INFO_MED=2, NOM_MED='Mesh_1',
UNITE=20)

model = AFFE_MODELE(
    AFFE=_F(MODELISATION=('POU_D_E', ), PHENOMENE='MECANIQUE',
TOUT='OUI'),
    MAILLAGE=mesh
)

elemprop = AFFE_CARA_ELEM(
    MODELE=model,
    ORIENTATION=_F(
        CARA='VECT_Y', GROUP_MA=('tr1', 'tr2'), VALE=(0.0, 0.0, 1.0)
    ),
    POUTRE=_F(
        CARA=('HY', 'HZ'),
        GROUP_MA=('tr1', 'tr2'),
        SECTION='RECTANGLE',
        VALE=(0.3, 0.4)
    )
)

cls = DEFI_MATERIAU(ELAS=_F(E=E, NU=0.3, RHO=2500.0))

```

```

fieldmat = AFFE_MATERIAU(AFFE=_F(MATER=(cls, ), TOUT='OUI'),
MODELE=model)

func_pp = DEFI_FONCTION(
    INTERPOL=('LIN', ),
    NOM_PARA='INST',
    PROL_DROITE='CONSTANT',
    PROL_GAUCHE='CONSTANT',
    TITRE='combinazioni pp_var1',
    VALE=(1.0, 1.3, 2.0, 1.3, 3.0, 1.3, 4.0, 1.3)
)

func_q1 = DEFI_FONCTION(
    INTERPOL=('LIN', ),
    NOM_PARA='INST',
    PROL_DROITE='CONSTANT',
    PROL_GAUCHE='CONSTANT',
    TITRE='combinazioni q_var1',
    VALE=(1.0, 0.0, 2.0, 1.5, 3.0, 0.0, 4.0, 1.5)
)

func_q2 = DEFI_FONCTION(
    INTERPOL=('LIN', ),
    NOM_PARA='INST',
    PROL_DROITE='CONSTANT',
    PROL_GAUCHE='CONSTANT',
    TITRE='combinazioni q_var2',
    VALE=(1.0, 0.0, 2.0, 0.0, 3.0, 1.5, 4.0, 1.5)
)

listr = DEFI_LIST_REEL(
    DEBUT=1.0,
    INTERVALLE=_F(JUSQU_A=4.0, PAS=1.0),
    TITRE='combinazioni carichi'
)

pp = AFFE_CHAR_MECA(
    MODELE=model,
    PESANTEUR=_F(
        DIRECTION=(0.0, 0.0, -1.0), GRAVITE=g, GROUP_MA=('tr1',
'tr2')
    )
)

qv1 = AFFE_CHAR_MECA(
    FORCE_NODALE=_F(FZ=Q1var, GROUP_NO=('nodi_tr1', )), MODELE=model
)

qv2 = AFFE_CHAR_MECA(
    FORCE_NODALE=_F(FZ=Q2var, GROUP_NO=('nodi_tr2', )), MODELE=model
)

bc = AFFE_CHAR_MECA(
    DDL_IMPO=_F(
        DRX=0.0,

```

```

        DRZ=0.0,
        DX=0.0,
        DY=0.0,
        DZ=0.0,
        GROUP_NO=('appoggi_estremi', 'appoggio_intermedio')
    ),
    MODELE=model
)

reslin = MECA_STATIQUE(
    CARA_ELEM=elemprop,
    CHAM_MATER=fieldmat,
    EXCIT=(
        _F(CHARGE=bc), _F(CHARGE=pp, FONC_MULT=func_pp),
        _F(CHARGE=qv1, FONC_MULT=func_q1), _F(CHARGE=qv2,
FONC_MULT=func_q2)
    ),
    LIST_INST=listr,
    MODELE=model
)

MNT = CALC_CHAMP(
    CARA_ELEM=elemprop,
    CONTRAINTE=('EFGE_ELGA', 'EFGE_ELNO', 'EFGE_NOEU'),
    LIST_INST=listr,
    MODELE=model,
    RESULTAT=reslin
)

tensioni = CALC_CHAMP(
    CARA_ELEM=elemprop,
    CONTRAINTE=('SIPM_ELNO', 'SIPO_ELNO', 'SIPO_NOEU'),
    LIST_INST=listr,
    MODELE=model,
    RESULTAT=reslin
)

IMPR_RESU(
    FORMAT='MED',
    RESU=_F(
        CARA_ELEM=elemprop, LIST_INST=listr, RESULTAT=reslin,
TOUT_CHAM='OUI'
    ),
    UNITE=80
)

IMPR_RESU(
    FORMAT='MED',
    RESU=_F(
        CARA_ELEM=elemprop, LIST_INST=listr, RESULTAT=tensioni,
TOUT_CHAM='OUI'
    ),
    UNITE=2
)

```

```

IMPR_RESU(
    FORMAT='MED',
    RESU= F(CARA_ELEM=elemprop, LIST_INST=listr, RESULTAT=MNT,
    TOUT_CHAM='OUI'),
    UNITE=3
)

FIN()

```

8.5 Portale soggetto a carichi verticali: “Command File”

```

# Altezza e Larghezza della sezione [m]
h = 0.5

b = 0.5

fck = 25

# Modulo Elastico medio del calcestruzzo
E = pow(((fck + 8) / 10.0), 0.3) * 22000000000.0

# Definizione dei coefficienti parziali (paragr. 2.6.1 NTC 2018)
g_glf = 1.0

g_gls = 1.3

g_g2f = 0.0

g_g2s = 1.5

g_qf = 0.0

g_qs = 1.5

# Definizione dei coefficienti di combinazione (paragrafo 2.5.3 NTC
2018)
# Categoria A, Ambienti ad uso residenziale
psi_0A = 0.7

psi_1A = 0.5

psi_2A = 0.3

# Categoria B, uffici
psi_0B = 0.7

psi_1B = 0.5

psi_2B = 0.3

# Categoria C, ambienti suscettibili di affollamento
psi_0C = 0.7

```



```

psi_1C = 0.7

psi_2C = 0.6

# Categoria D, ambienti ad uso commerciale
psi_0D = 0.7

psi_1D = 0.7

psi_2D = 0.6

# Categoria E, biblioteche, archivi, magazzini e ambienti ad uso
industriale
psi_0E = 1.0

psi_1E = 0.9

psi_2E = 0.8

# Categoria F, rimesse e parcheggi (per autoveicoli di peso <= 30
kN)
psi_0F = 0.7

psi_1F = 0.7

psi_2F = 0.6

# Categoria G, Rimesse e parcheggi (per autoveicoli di peso > 30 kN)
psi_0G = 0.7

psi_1G = 0.5

psi_2G = 0.3

# Categoria H, Coperture
psi_0H = 0.0

psi_1H = 0.0

psi_2H = 0.0

# Vento
psi_0V = 0.6

psi_1V = 0.2

psi_2V = 0.0

# Neve (a quota <= 1000 m s.l.m.)
psi_0N1 = 0.5

psi_1N1 = 0.2

psi_2N1 = 0.0

```

```

# Neve (a quota > 1000 m s.l.m.)
psi_0N2 = 0.7

psi_1N2 = 0.5

psi_2N2 = 0.2

# Variazioni termiche
psi_0T = 0.6

psi_1T = 0.5

psi_2T = 0.0

# Definizione della costante di accelerazione gravitazionale
g = 9.81

# Definizione dei carichi agenti [Newton]

q1 = 50000

q2 = 50000

q3 = 10000

q4 = 0

# Lunghezza_
L = 4.0

# num_nodi
n_nodi = 25

Q2var = ((q2) * L) / n_nodi

Q1var = ((q1) * L) / n_nodi

DEBUT()

mesh = LIRE_MALLAGE(FORMAT='MED', INFO_MED=3, NOM_MED='Mesh_1',
UNITE=2)

MODELLO = AFFE_MODELE(
    AFFE=_F(MODELISATION=('POU_D_E', ), PHENOMENE='MECANIQUE',
TOUT='OUI'),
    MAILLAGE=mesh
)

elemprop = AFFE_CARA_ELEM(
    MODELE=MODELLO,
    ORIENTATION=_F(
        CARA='VECT_Y', GROUP_MA=('Compound_1', ), VALE=(0.0, 1.0,
0.0)
    ),
    POUTRE=_F(

```

```

        CARA=('HY', 'HZ'),
        GROUP_MA=('aste_verticale', 'asta_orrizzontale'),
        SECTION='RECTANGLE',
        VALE=(0.5, 0.5)
    )
)

CLS = DEFI_MATERIAU(ELAS=_F(E=E, NU=0.3, RHO=2500.0))

As_CLS = AFFE_MATERIAU(
    AFFE=_F(MATER=(CLS, ), TOUT='OUI'), MAILLAGE=mesh,
    MODELE=MODELLO
)

# definizione dei carichi e delle forze
pp = AFFE_CHAR_MECA(
    MODELE=MODELLO,
    PESANTEUR=_F(
        DIRECTION=(0.0, 0.0, -1.0),
        GRAVITE=g,
        GROUP_MA=('aste_verticale', 'asta_orrizzontale')
    )
)

BC2 = AFFE_CHAR_MECA(
    DDL_IMPO=(
        _F(GROUP_NO=('vincoli_piede', ), LIAISON='ENCASTRE'),
        _F(DRX=0.0, DY=0.0, GROUP_NO=('vincoli_testa_angoli', ))
    ),
    MODELE=MODELLO
)

# carico variabile qk1
caricoq1 = AFFE_CHAR_MECA(
    FORCE_POUTRE=_F(FZ=q1, GROUP_MA='asta_orrizzontale'),
    MODELE=MODELLO
)

# carico variabile qk2
caricoq2 = AFFE_CHAR_MECA(
    FORCE_POUTRE=_F(FZ=q2, GROUP_MA='asta_orrizzontale'),
    MODELE=MODELLO
)

# carico variabile qk3
caricoq3 = AFFE_CHAR_MECA(
    FORCE_POUTRE=_F(FZ=q3, GROUP_MA='asta_orrizzontale'),
    MODELE=MODELLO
)

# carico variabile qk4
caricoq4 = AFFE_CHAR_MECA(
    FORCE_POUTRE=_F(FZ=q4, GROUP_MA='asta_orrizzontale'),
    MODELE=MODELLO
)

```

```

resq1 = MECA_STATIQUE(
    CARA_ELEM=elemprop,
    CHAM_MATER=As_CLS,
    EXCIT=( _F(CHARGE=BC2), _F(CHARGE=caricoq1)),
    MODELE=MODELLO
)

resq2 = MECA_STATIQUE(
    CARA_ELEM=elemprop,
    CHAM_MATER=As_CLS,
    EXCIT=( _F(CHARGE=BC2), _F(CHARGE=caricoq2)),
    MODELE=MODELLO
)

resq3 = MECA_STATIQUE(
    CARA_ELEM=elemprop,
    CHAM_MATER=As_CLS,
    EXCIT=( _F(CHARGE=BC2), _F(CHARGE=caricoq3)),
    MODELE=MODELLO
)

resq4 = MECA_STATIQUE(
    CARA_ELEM=elemprop,
    CHAM_MATER=As_CLS,
    EXCIT=( _F(CHARGE=BC2), _F(CHARGE=caricoq4)),
    MODELE=MODELLO
)

# #####
# calcolo del peso proprio pp
respp = MECA_STATIQUE(
    CARA_ELEM=elemprop,
    CHAM_MATER=As_CLS,
    EXCIT=( _F(CHARGE=BC2), _F(CHARGE=pp)),
    MODELE=MODELLO
)

# #####
# estrazione di una tabella degli spostamenti in direzione z
# #####
R_pp = CALC_CHAMP(
    CARA_ELEM=elemprop,
    CONTRAINTE=('EFGE_ELNO', 'EFGE_ELGA'),
    MODELE=MODELLO,
    RESULTAT=respp
)

R_q1 = CALC_CHAMP(
    CARA_ELEM=elemprop,
    CONTRAINTE=('EFGE_ELNO', 'EFGE_ELGA'),
    MODELE=MODELLO,
    RESULTAT=resq1
)

```

```

R_q2 = CALC_CHAMP(
    CARA_ELEM=elemprop,
    CONTRAINTE=('EFGE_ELNO', 'EFGE_ELGA'),
    MODELE=MODELLO,
    RESULTAT=resq2
)

R_q3 = CALC_CHAMP(
    CARA_ELEM=elemprop,
    CONTRAINTE=('EFGE_ELNO', 'EFGE_ELGA'),
    MODELE=MODELLO,
    RESULTAT=resq3
)

R_q4 = CALC_CHAMP(
    CARA_ELEM=elemprop,
    CONTRAINTE=('EFGE_ELNO', 'EFGE_ELGA'),
    MODELE=MODELLO,
    RESULTAT=resq4
)

c_pp = CREA_CHAMP(
    NOM_CHAM='EFGE_ELNO',
    OPERATION='EXTR',
    RESULTAT=R_pp,
    TYPE_CHAM='ELNO_SIEF_R'
)

c_q1 = CREA_CHAMP(
    NOM_CHAM='EFGE_ELNO',
    OPERATION='EXTR',
    RESULTAT=R_q1,
    TYPE_CHAM='ELNO_SIEF_R'
)

c_q2 = CREA_CHAMP(
    NOM_CHAM='EFGE_ELNO',
    OPERATION='EXTR',
    RESULTAT=R_q2,
    TYPE_CHAM='ELNO_SIEF_R'
)

c_q3 = CREA_CHAMP(
    NOM_CHAM='EFGE_ELNO',
    OPERATION='EXTR',
    RESULTAT=R_q3,
    TYPE_CHAM='ELNO_SIEF_R'
)

c_q4 = CREA_CHAMP(
    NOM_CHAM='EFGE_ELNO',
    OPERATION='EXTR',
    RESULTAT=R_q4,
    TYPE_CHAM='ELNO_SIEF_R'
)

```

```

SLU = CREA_CHAMP(
  ASSE=(
    _F(CHAM_GD=c_pp, COEF_R=g_gls, CUMUL='OUI', TOUT='OUI'), _F(
      CHAM_GD=c_q1, COEF_R=g_qs, CUMUL='OUI', TOUT='OUI'
    ), _F(CHAM_GD=c_q2, COEF_R=g_qs, CUMUL='OUI', TOUT='OUI'),
  _F(
    CHAM_GD=c_q3, COEF_R=g_qs, CUMUL='OUI', TOUT='OUI'
  ), _F(CHAM_GD=c_q4, COEF_R=g_qs, CUMUL='OUI', TOUT='OUI')
  ),
  MODELE=MODELLO,
  OPERATION='ASSE',
  TYPE_CHAM='ELNO_SIEF_R'
)

Tabella = POST_RELEVE_T(
  ACTION=(
    _F(
      GROUP_NO=('Forza_mezz', 'vincoli_piede',
'vincoli_testa_angoli'),
      INTITULE='Peso Proprio agente sul traverso
orizzontale',
      NOM_CHAM='EFGE_ELNO',
      NOM_CMP=('MFY', 'VZ'),
      OPERATION=('EXTRACTION', ),
      RESULTAT=R_pp
    ), _F(
      GROUP_NO=('Forza_mezz', ),
      INTITULE='Peso proprio',
      NOM_CHAM='EFGE_ELNO',
      OPERATION=('EXTRACTION', ),
      REPERE='GLOBAL',
      RESULTAT=R_pp,
      TOUT_CMP='OUI'
    )
  ),
  TITRE='Tabella di confronto dei valori M, N e V nel punto di
mezzeria'
)

IMPR_RESU(
  FORMAT='MED',
  RESU=(
    _F(
      CARA_ELEM=elemprop,
      CHAM_GD=SLU,
      GROUP_MA=('Compound_1', ),
      TOUT_CMP='OUI'
    ), _F(CARA_ELEM=elemprop, RESULTAT=respp, TOUT_CMP='OUI')
  ),
  UNITE=3
)

IMPR_TABLE(
  FORMAT='TABLEAU',

```

```

        FORMAT_R='E12.3',
        SEPARATEUR="  ",
        TABLE=Tabella,
        UNITE=11
    )

    IMPR_RESU(
        FORMAT='RESULTAT',
        MODELE=MODELLO,
        RESU=_F(
            CARA_ELEM=elemprop,
            CHAM_GD=SLU,
            FORMAT_R='1PE21.3',
            FORM_TABL='EXCEL',
            GROUP_NO=('Forza_mezz', 'vincoli_piede',
'vincoli_testa_angoli'),
            IMPR_COOR='OUI',
            NOM_CMP=('N', 'VZ', 'MFY')
        ),
        UNITE=12
    )

    FIN()

```


9 Bibliografia

- www.code-aster.org;
- www.python.org;
- www.otebac.it;
- www.paso.wordpress.com;
- www.opensource.org;
- www.gnu.org;
- www.virtualbox.org;
- www.salome-platform.org;
- “Norme Tecniche per le Costruzioni” D.M. 17/1/2018;
- www.wikipedia.org;
- www.paraview.org;
- www.edf.fr;
- www.kitware.com
- “*Scienza delle costruzioni Vol.2*”, Alberto Carpinteri, Ed. Pitagora, Bologna;
- “*Beginning with Code_Aster A practical introduction to finite element method using Code_Aster Gmsh and Salome*”, Jean-Pierre Aubry, Framabook;
- “*Finite element analysis using Open-source software first edition*”, Dharmit A. Thakore, Moonish Enterprises Pty Ltd;
- “*Finite element analysis using Open-source software second edition*”, Dharmit A. Thakore, Moonish Enterprises Pty Ltd;

10 Ringraziamenti

Desidero ringraziare il Prof. Invernizzi per la pazienza, le indicazioni ed i preziosi consigli nella stesura di questa Tesi. Inoltre un pensiero è dedicato ai miei genitori per il sostegno, ai miei amici ed ai miei compagni di corso per il supporto in questi anni universitari.