

Università degli Studi di Padova

DIPARTIMENTO DI INGEGNERIA CIVILE EDILE ED
AMBIENTALE

Corso di Laurea in Ingegneria Civile

PROCEDURA AUTOMATIZZATA PER LA
VALUTAZIONE DELLA VULNERABILITÀ
SISMICA DI TIPOLOGIE RICORRENTI DI
PONTI E VIADOTTI

Tesi di laurea di:

Diego Debortoli

Relatore:

Prof. Ing. **Carlo Pellegrino**

Correlatore:

Dr. Ing. **Riccardo Morbin**

Anno Accademico 2012-2013

Indice

1	INTRODUZIONE	4
2	SISMICA E DINAMICA STRUTTURALE	7
2.1	GENERALITÀ SUI FENOMENI SISMICI	7
2.2	PERICOLOSITÀ SISMICA DI RIFERIMENTO PER IL TERRITORIO NAZIONALE	11
2.3	GRANDEZZE FISICHE PER LA DESCRIZIONE DI UN TERREMOTO	13
2.3.1	Accelerogrammi	13
2.3.2	Spettro di risposta elastico	15
2.3.3	Spettro di risposta di progetto	18
2.4	METODI DI ANALISI PER IL CALCOLO DELLA RISPOSTA STRUTTURALE ALL'AZIONE SISMICA	22
2.4.1	Analisi statica lineare	23
2.4.2	Analisi dinamica lineare	24
2.4.3	Analisi statica non lineare	25
2.4.4	Analisi dinamica non lineare	26
3	VULNERABILITÀ SISMICA E CURVE DI FRAGILITÀ	27
3.1	RISCHIO SISMICO E PERICOLOSITÀ SISMICA	27
3.2	VULNERABILITÀ SISMICA	31
3.3	CURVE DI FRAGILITÀ EMPIRICHE METODO RISK-UE	34
3.4	CURVE DI FRAGILITÀ ANALITICHE	37
3.4.1	Definizione degli indici di danno	37
3.4.2	Analisi in Time-history: Metodo 1	39
3.4.3	Analisi in Time-history: Metodo 2	43
3.4.4	La distribuzione Normale e Log-normale	45

	3
4 PROCEDURA DI AUTOMATIZZAZIONE	49
4.1 PROGRAMMARE CON MATLAB	49
4.2 L'INTERFACCIA GRAFICA DI MATLAB, GUI (Graphical User Interface)	50
4.3 COSTRUZIONE DEL MODELLO TRIDIMENSIONALE	52
4.3.1 Dalla geometria del ponte al modello numerico	53
4.3.2 Geometria degli elementi strutturali	59
4.4 ANALISI DINAMICA NON LINEARE CON OPENSEES	63
4.4.1 OpenSees	63
4.4.2 Creazione dei file di comando	64
4.4.3 Lanciare l'analisi con OpenSees	70
4.5 COSTRUZIONE DELLE CURVE DI FRAGILITÀ	72
4.5.1 Distribuzione probabilistica dei materiali	72
4.5.2 Calcolo dell'indice di danno, la duttilità cinematica della sezione	78
4.5.3 Costruzione automatica delle curve di fragilità	82
5 RISULTATI	84
6 CONCLUSIONI E SVILUPPI FUTURI	89
BIBLIOGRAFIA	92
A APPENDICE DEGLI SCRIPT	95

Capitolo 1

INTRODUZIONE

L'Italia è un paese caratterizzato da un elevato rischio sismico. Tale conoscenza è storicamente resa nota dal grande numero di documenti e informazioni sugli effetti che nel passato i terremoti hanno provocato nelle diverse aree geografiche della nostra penisola. Per ciascuna zona sappiamo, almeno nell'intervallo di tempo per il quale sono disponibili le informazioni, il numero e l'intensità dei terremoti che l'hanno interessata. A partire dal diciannovesimo secolo, il fenomeno sismico è stato studiato ed analizzato mediante metodi scientifici rigorosi, ed ora siamo in grado di fornire delle carte di pericolosità sismica che propongono una classificazione di tutto il territorio italiano utilizzando come indicatore di pericolosità l'accelerazione orizzontale di picco al suolo, *Peak Ground Acceleration (PGA)* oppure un indicatore di intensità macrosismica basato sul grado di danneggiamento indotto dal terremoto.

Al giorno d'oggi sono disponibili tutta una serie di tecniche di progettazione e di costruzione che permettono di governare il problema legato all'azione sismica in maniera sempre più affidabile. Quindi, sebbene si possa ritenere ridimensionato il rischio sismico sui nuovi manufatti, dal momento che questi debbono soddisfare elevati standard di sicurezza, dettati in primo luogo dalle norme tecniche per le costruzioni, non è da trascurare, tuttavia, il pericolo generato dall'azione del terremoto su edifici e infrastrutture esistenti, o addirittura vetuste. Ora, data l'evidente importanza che riveste l'infrastruttura viaria qualora si dovesse far fronte ad un evento sismico di qualche rilievo, dipendendo dalla stessa il soccorso della popolazione all'indomani del terremoto, inoltre le attività economiche della zona colpita possono riprendere tanto più rapidamente quanto questa si danneggia in misura inferiore, si rende necessario lo studio di un metodo mediante il quale determinare in maniera oggettiva la vulnerabilità, rispetto ad un determinato evento sismico, delle strutture più deboli della rete di

trasporti, quali ponti e viadotti.

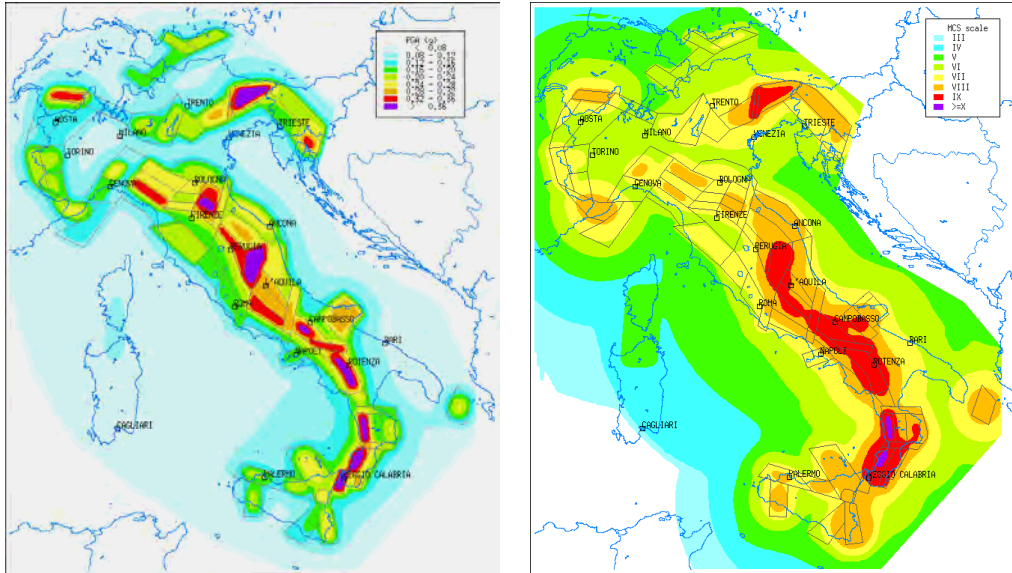


Figura 1.1: Mappe pericolosità sismica in Italia

Una volta determinata la vulnerabilità sismica degli elementi, unitamente a valutazioni di tipo strategico, sarà possibile redigere un piano, secondo opportune priorità, per la manutenzione ed eventualmente l'adeguamento di tali opere.

Il metodo di analisi più opportuno si è rivelato la costruzione delle curve di fragilità, le quali rappresentano la probabilità condizionata di eccedere un definito livello di danno per diversi valori di intensità dell'azione sismica. Come si può ben capire, l'approccio al problema è di tipo probabilistico: questa esigenza nasce dalla stessa natura aleatoria delle variabili in gioco, basti pensare ad esempio alle incertezze legate all'intensità del sisma e alle proprietà meccaniche dei materiali stessi.

Il presente lavoro di tesi si prefissa l'obiettivo di fornire all'utente uno strumento per la costruzione automatizzata delle curve di fragilità di ponti e viadotti in cemento armato. Più nello specifico è stata realizzata un'interfaccia grafica mediante Matlab che permette la costruzione di un modello strutturale tridimensionale di un ponte in cemento armato, con caratteristiche geometriche (lunghezza delle campate, larghezza dell'impalcato, numero e sezione dei traversi nell'impalcato) e strutturali (numero e disposizione dei ferri d'armatura) fornite dall'utente. Successivamente, la stessa interfaccia, richiamando un opportuno codice di Matlab, andrà a tradurre la geometria appena immessa in un linguaggio fruibile dal programma di analisi sismica, che nella fattispecie è *Opensees*. Viene quindi, in modo del tutto automatico, lanciata un'analisi dinamica non lineare in *Time-*

history, basandosi su accelerogrammi spettrocompatibili, impostati di default o forniti dall'utente. Infine, i risultati così ottenuti sono ulteriormente elaborati su base statistica, al fine di ottenere le curve di fragilità per il ponte assemblato.

In sintesi, si è voluto creare, mediante l'utilizzo di Matlab, un'interfaccia molto intuitiva e flessibile che permetta di ottenere una simulazione sismica e quindi le relative curve di fragilità in maniera del tutto automatica per ogni ponte o viadotto. Dal momento che non è possibile eseguire un'analisi dinamica dedicata per ogni singolo ponte presente su territorio nazionale, dato l'elevato numero, lo strumento sviluppato nell'ambito del seguente lavoro di tesi si pone come valida alternativa, poichè è in grado di fornire, con un basso onere computazionale, informazioni con buona approssimazione circa la vulnerabilità sismica di un determinato manufatto, questo con la finalità ultima di avere criteri oggettivi per redigere piani contenenti interventi di manutenzione e adeguamento su strutture esistenti con un'opportuna priorità.

Capitolo 2

SISMICA E DINAMICA STRUTTURALE

2.1 GENERALITÀ SUI FENOMENI SISMICI

Con il termine terremoto (dal latino *terrae motus*, cioè movimento della terra), detto anche sisma o scossa tellurica, intendiamo fenomeni vibrazionali improvvisi, più o meno intensi, che coinvolgono la crosta terrestre e che sono generati dallo scarico repentino di energia accumulatasi nel sottosuolo. Questo accumulo di energia può avere molteplici cause, ma nella maggior parte dei casi, è imputabile a fenomeni tettonici che comprendono il movimento relativo di placche di crosta terrestre, con forte generazione di attrito lungo le fratture tra suddette placche, dette *faglie*. L'energia accumulata dai fenomeni sopra eposti si libera sottoforma di onde che causano la vibrazione della superficie terrestre, tali perturbazioni, dette *onde sismiche*, si propagano in tutte le direzioni a partire dall'*ipocentro*, mentre il luogo posto sulla sua verticale, sulla superficie terrestre, si chiama *epicentro*, ed è di norma, il punto generalmente interessato in misura maggiore dal fenomeno. La fonte di origine di tali onde può essere considerata come un punto a superficie sferica dalla quale le onde si propagano in direzione radiale, mentre per i terremoti di maggiore entità, è ritenuto più corretto considerare il centro di formazione delle onde come un segmento lungo la faglia a superficie cilindrica.

Suddette perturbazioni si propagano all'interno del mezzo solido secondo molteplici tipi di onde sismiche: possiamo trovare le *onde di compressione o longitudinali (P)*, la cui velocità di propagazione è variabile, essendo funzione delle caratteristiche e della densità del materiale entro il quale si propagano, in genere però viaggiano ad una velocità compresa tra i 4-8 km/s, sono le onde più veloci

generate da un sisma e pertanto le prime ad essere avvertite in superficie, da cui il nome *onde P* (*primarie*). Tali perturbazioni fanno oscillare le particelle che incontrano parallelamente alla direzione di propagazione dell'onda. Generano, nel materiale attraversato, una serie di compressioni e decompressioni alternate. Poi esistono le *onde di taglio o trasversali (S)*, essendo più lente delle onde P sono anche dette *onde S* (*secondarie*). Esse si diffondono nel materiale attraversato in direzione perpendicolare alla loro propagazione, viaggiano ad una velocità variabile, solitamente compresa tra i 2-4 km/s, la particolarità di queste onde è che non comportano variazioni volumetriche nel materiale perturbato e che non possono propagarsi in fluidi (liquidi e gas) avendo questi ultimi resistenza a taglio nulla. Le due tipologie sopra esposte fanno parte della categoria denominata: *onde di corpo o volume (body waves)*, che si propagano dalla sorgente sismica attraverso il volume del mezzo interessato lungo tutte le direzioni. Quando queste raggiungono la superficie vengono rifratte e al contempo, a causa di questa discontinuità, generano una ulteriore categoria di onde, ovvero le *onde superficiali (surface waves)* che, a differenza di ciò che si potrebbe pensare, non si manifestano nell'epicentro, ma solo ad una certa distanza da questo. Esse si suddividono in *onde di Rayleigh (onde R)*, che muovono le particelle secondo orbite ellittiche in un piano verticale lungo la direzione di propagazione, con un moto molto simile all'increspatura della superficie d'acqua di uno stagno, ed in *onde di Love (onde L)*, le quali muovono invece le particelle trasversalmente alla direzione di propagazione ma unicamente sul piano orizzontale.

La misurazione delle onde sismiche avviene mediante sismografi, che ne rilevano la presenza sulla superficie terrestre, l'elaborazione incrociata dei dati di più sismografi sparsi su un territorio ad una certa distanza dal sisma consente di stimare in maniera abbastanza accurata l'epicentro, l'ipocentro e l'intensità del terremoto. Quest'ultima grandezza fisica può essere misurata secondo diverse scale, tra cui la scala Mercalli (1902), basata su effetti macroscopici del sisma su strutture esistenti, e la scala Richter basata sulla *magnitudo*, che è l'energia meccanica sprigionata dall'evento sismico all'ipocentro, calcolata in maniera indiretta basandosi sull'ampiezza delle onde sismiche registrate in superficie dai sismografi. La scala Mercalli, come detto, misura l'intensità del sisma basandosi sulla percezione soggettiva di movimento e danni agli edifici, e pertanto, al giorno d'oggi, con la crescente raffinatezza delle tecniche di costruzione antisismica, poco si presta ad un'oggettiva quantificazione della violenza del sisma. In maniera del tutto analoga, anche la *magnitudo* fornisce un'informazione poco fruibile dal punto di vista dell'ingegneria sismica nell'ottica della valutazione degli effetti

del terremoto nei confronti delle strutture.

Per tali applicazioni è necessario, infatti, disporre di grandezze fisiche opportune e tali sono l'accelerazione al suolo e la durata del sisma, per misurarle è necessario disporre di accelerografi *strong - motion*. Essi sono degli apparecchi che permettono di registrare in modo completo, come funzioni del tempo, le tre componenti di accelerazione del suolo, due orizzontali (nord -sud ed est – ovest) ed una verticale. L'apparato registratore di un accelerografo *strong - motion* di solito entra in funzione solo nel caso di fenomeno sismico di entità tale da indurre danni su edifici ed ambiente, in particolare quando l'accelerazione del suolo supera un predeterminato valore di soglia. L'accelerografo è formato da un trasduttore che nella sua forma elementare è costituito da un oscillatore semplice dotato di una propria frequenza e di un certo smorzamento, questi strumenti registrano l'andamento dell'accelerazione in funzione del tempo contenenti frequenze entro un certo *range*. All'interno di un accelerografo sono presenti tre trasduttori, al fine di registrare l'accelerogramma nelle tre direzioni, solitamente le componenti in direzione orizzontale hanno ampiezza dello stesso ordine di grandezza mentre la componente verticale presenta ampiezze più contenute. In un accelerogramma sono contenute le seguenti informazioni circa l'evento sismico registrato:

- ampiezza, caratterizzata dal picco (*PGA, Peak Ground Acceleration*) o dal numero di picchi che superano un livello prefissato;
- durata;
- contenuto in frequenze;

Sulla base di tali grandezze si ricorda una classificazione del passato in quattro tipologie [*Newmark e Rosenblueth (1971)*] :

1. Scossa di tipo impulsivo (praticamente un solo urto)

Generalmente si verifica non molto distante dall'epicentro, attraverso un suolo abbastanza compatto, come esempi si citano gli eventi accaduti a: *Agadir (1960), Skopje (1963), Port Hueneme (1957)*. Un esempio è riportato nella figura seguente:

2. Scossa moderatamente lunga, con movimento estremamente irregolare

Si tratta di un terremoto caratterizzato da una durata piuttosto lunga, con ampiezza molto irregolare, con un contenuto in frequenze molto largo. Generalmente si verifica ad una distanza media dall'epicentro e in un terreno compatto, l'evento più significativo di questa categoria è il terremoto di *El*

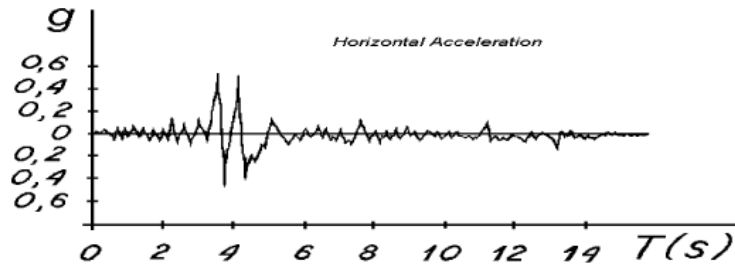


Figura 2.1: Scossa di tipo impulsivo, Port Hueneme (1957)

Centro, (California 1940) come mostrato in figura 2.2:

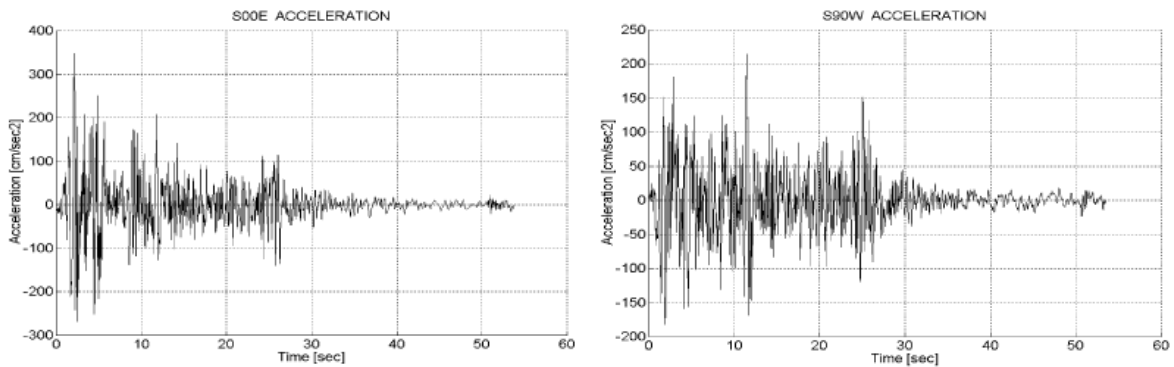


Figura 2.2: Scossa di El Centro, (California 1940)

3. Scossa di lunga durata, con periodi predominanti pronunciati

Questo tipo di sisma è caratterizzato da una frequenza predominante, il terreno svolge il ruolo di filtro nella riflessione delle onde di corpo quando raggiungono la superficie. Un classico esempio è il terremoto di *Mexico City* (1964).

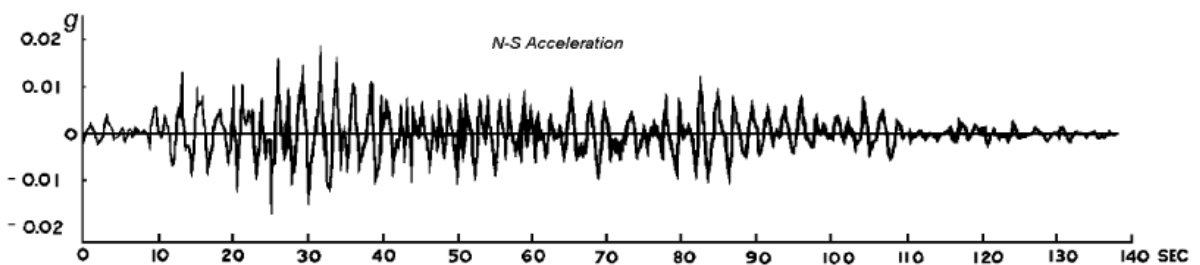


Figura 2.3: Scossa di Mexico City (1964)

4. Scossa che determina deformazioni permanenti del suolo su larga scala

Questo tipo di terremoto causa delle forti deformazioni del suolo, con eventuali scorrimenti macroscopici o liquefazioni, si citano ad esempio gli eventi: *Cile (1960); Alaska (1964); Niigata (1964)*.

2.2 PERICOLOSITÀ SISMICA DI RIFERIMENTO PER IL TERRITORIO NAZIONALE

In seguito all'ordinanza PCM 3274/2003 si è avviato, per la prima volta in Italia, un processo per la stima della pericolosità sismica del territorio, sono state delineate in maniera organica il tipo di procedure da seguire, la natura dei dati da considerare e il tipo di prodotti da rilasciare. Si è potuto, in questa maniera giungere ad una puntuale e approfondita assegnazione dei comuni italiani a zone sismiche. Con il termine pericolosità sismica si intende, in senso probabilistico, lo scuotimento atteso a livello del suolo (che può essere espresso in termini di PGA o di intensità macrosismica) con una certa probabilità di eccedenza in un dato intervallo di tempo. In altri termini la probabilità che, in un dato periodo di riferimento (*periodo di ritorno T_r*), possa, in media, verificarsi un sisma che ecceda una prefissata intensità:

$$T_r = -\frac{V_r}{\ln(1 - P_{V_r})} \quad (2.1)$$

dove:

- T_r è il tempo di ritorno per un prefissato livello di azione sismica;
- V_r è la vita di riferimento della costruzione;
- P_{V_r} è la probabilità che nell'intervallo di tempo della vita di riferimento vengano superati gli stati limite considerati.

Sono, quindi stati rilasciati dall'INGV (*Istituto Nazionale per la Geofisica e Vulcanologia*), una serie di mappe di pericolosità sismica per diverse probabilità di eccedenza con riferimento ad una vita utile di 50 anni, basate tutte sulla medesima metodologia statistica e su dati in input omogenei; le mappe riportano come valore dell'azione sismica: livelli di intensità macrosismica, PGA, oppure vengono espresse in termini di accelerazioni spettrali.

I temi sopra illustrati verranno approfonditi nel successivo capitolo al fine di introdurre il concetto di vulnerabilità sismica.

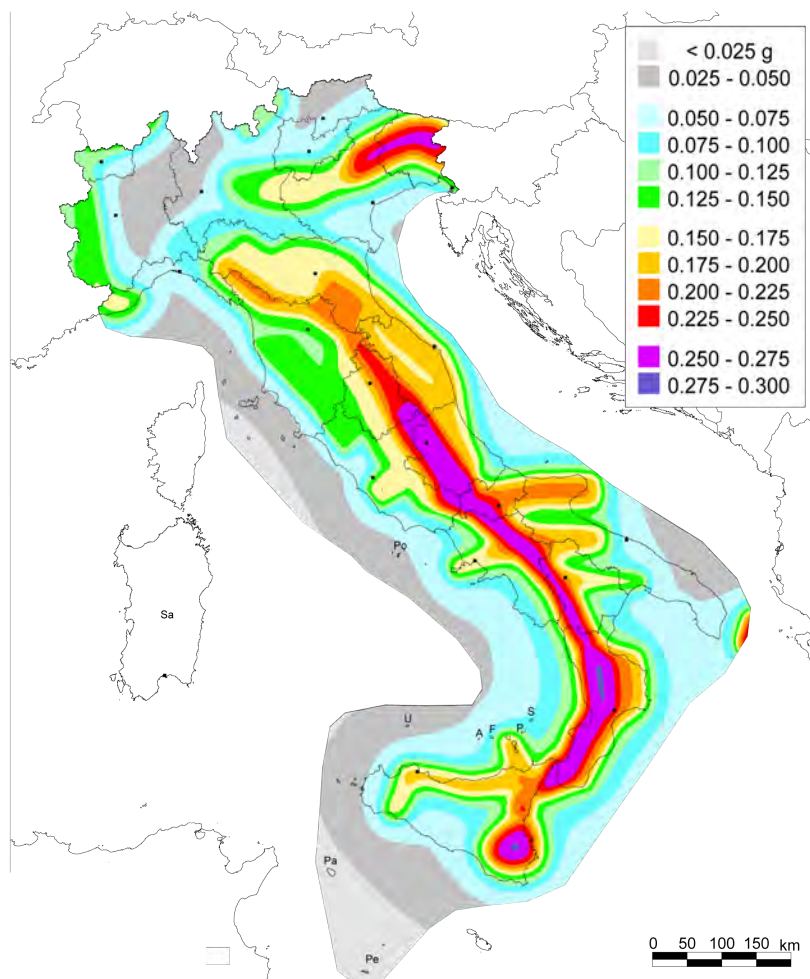


Figura 2.4: Mappa di pericolosità sismica del territorio nazionale espressa in termini di accelerazione massima del suolo con probabilità di eccedenza del 10% in 50 anni

2.3 GRANDEZZE FISICHE PER LA DESCRIZIONE DI UN TERREMOTO

2.3.1 Accelerogrammi

Come precedentemente accennato, nell'ambito dell'ingegneria sismica vengono presi in considerazione solo gli eventi sismici denominati strong motion, ovvero eventi che generano vibrazioni violente tali da produrre danni su edifici e ambiente. Questo in contrapposizione con l'attività microsismica, con la quale si intendono vibrazioni con frequenze molto basse e periodi ampi tali da non essere percepiti dai più comuni strumenti, ha rilievo soprattutto per i sismologi. Per caratterizzare in maniera oggettiva gli effetti dell'attività strong motion è necessario registrare in termini di accelerazione tali componenti, ottenendo in questa maniera dei diagrammi che ne illustrano la variazione in funzione del tempo (accelerogrammi). Come anticipato, il moto sismico viene descritto da ampiezza, contenuto in frequenza e durata.

La grandezza fisica che meglio descrive l'ampiezza è il *picco di accelerazione* o (*PGA*) che viene definito come il picco di valore più elevato dell'accelerazione durante tutta la manifestazione del fenomeno sismico. Esso si divide in:

- *PGHA* se riferito alle componenti orizzontali, è certamente la caratteristica più importante dal momento che sono proprio queste le componenti che inducono sollecitazioni sugli edifici esistenti, inoltre, è possibile correlare in maniera indiretta il picco di accelerazione orizzontale con l'intensità macroscopica del sisma (magnitudo), in maniera tale da poter stimare una *PGA* anche per zone di cui sono note solamente registrazioni di magnitudo;
- *PGVA* se riferito alle componenti verticali, che nell'ambito dell'ingegneria sismica rivestono un minore interesse essendo le strutture maggiormente resistenti nei confronti delle sollecitazioni verticali. Solitamente è un'approssimazione accettabile assumere:

$$PGVA = PGHA \cdot \frac{2}{3} \quad (2.2)$$

Altra grandezza di fondamentale importanza è la durata, infatti in ambito ingegneristico solo una porzione temporale dell'intero fenomeno contiene informazioni di qualche interesse. Per cui come durata si intende l'intervallo temporale

che comprende la porzione strong motion del sisma ed esso può essere calcolato secondo diversi criteri:

- porzione di tempo compresa tra il primo e l'ultimo superamento di un'arbitraria soglia di accelerazione (solitamente si assume 0.05 g);
- porzione di tempo compresa tra gli istanti in cui l'energia del segnale varia tra il 5% e 95% dell'energia complessiva registrata.

Infine, l'ultima ma forse la più importante caratteristica per la descrizione di un fenomeno sismico è il *contenuto in frequenza*, esso descrive come le varie ampiezze della scossa si suddividono tra le varie frequenze. Questa nozione è di fondamentale importanza perchè la risposta dinamica delle strutture dipende non solo dal valore assoluto delle sollecitazioni applicate (ciò viene rappresentato dall'ampiezza delle componenti di accelerazione) ma soprattutto dalla frequenza del loro manifestarsi, in rapporto con la frequenza di oscillazione propria della struttura in questione.

Ogni componente di accelerazione è una funzione periodica variabile con il tempo che può essere interpretata come una somma di funzioni armoniche semplici. Ogni generica, *n-esima*, funzione armonica $x(t)$ è caratterizzata da un periodo T_f , una frequenza ω_n , ampiezza C_n e fase ϕ_n , secondo la trasformazione di Fourier:

$$x(t) = C_0 + \sum_{n=1}^{\infty} C_n \sin(\omega_n t + \phi_n) \quad (2.3)$$

dove:

$$C_0 = a_0 \quad (2.4)$$

$$C_n = \sqrt{a_n^2 + b_n^2} \quad (2.5)$$

$$\phi_n = \tan^{-1}\left(\frac{a_n}{b_n}\right) \quad (2.6)$$

$$\omega_n = \frac{2\pi n}{T_f} \quad (2.7)$$

con:

$$a_0 = \frac{1}{T_f} \int_0^{T_f} x(t) dt \quad (2.8)$$

$$a_n = \frac{2}{T_f} \int_0^{T_f} x(t) \cos(\omega_n t) dt \quad (2.9)$$

$$b_n = \frac{2}{T_f} \int_0^{T_f} x(t) \sin(\omega_n t) dt \quad (2.10)$$

Ora, dal momento che l'accelerogramma registrato dagli strumenti è un insieme discreto di valori, di conseguenza anche la trasformazione di Fourier è ricavata in forma discreta e prende il nome di *DFT, Discrete Fourier Transformation*. Si ottiene, in tale maniera, la scomposizione del segnale dell'accelerogramma in una somma di funzioni armoniche caratterizzate da un insieme discreto di frequenze. Tali informazioni vengono raggruppate nello *spettro di Fourier* che rappresenta la variazione delle ampiezze e delle fasi delle singole armoniche, e in funzione del periodo o della frequenza permette di ricavare il periodo (o frequenza) fondamentale, quello, cioè, in corrispondenza al quale si ha l'ampiezza massima.

Gli accelerogrammi, infine, possono anche essere classificati sulla base del metodo con il quale vengono generati, si distinguono:

- accelerogrammi naturali, vengono registrati a partire dall'evento manifestatosi in natura e sono molto correlati alle caratteristiche di intensità del sisma, in termini di magnitudo e di distanza del sito di registrazione dall'epicentro. Il loro utilizzo, in ambito di analisi strutturale, è da farsi con qualche cautela, variando di molto, a parità di intensità dell'evento sismico, gli effetti sortiti sulle strutture;
- accelerogrammi spettro-compatibili, vengono calcolati sulla base di uno spettro di risposta atteso, a partire da registrazioni di eventi realmente accaduti in natura, vengono resi compatibili entro determinate tolleranze;
- accelerogrammi derivanti da modelli statistici, sono ottenuti in maniera del tutto artificiale attraverso degli algoritmi di modellazione stocastica che prevedono una simulazione dei reali meccanismi di generazione e propagazione del terremoto nel sottosuolo. Devono essere specificati parametri di tipo geologico come la dimensione della faglia, la natura e composizione del mezzo entro il quale si propagheranno le onde sismiche. Esistono anche algoritmi di previsione che generano accelerogrammi a loro volta spettro-compatibili con uno spettro di risposta atteso.

2.3.2 Spettro di risposta elastico

Ai fini della progettazione sismica è necessario, oltre che la descrizione matematica del moto sismico, anche conoscere la risposta di una struttura nei confronti di tale sollecitazione. Per fare ciò si considera un sistema semplice composto da

un oscillatore ad un singolo grado di libertà (SDOF), e si analizza la risposta al variare delle sue caratteristiche: *periodo proprio* e *smorzamento*.

Lo *spettro di risposta elastico* è un diagramma che esprime la risposta massima ad un particolare moto sismico di input, al variare delle caratteristiche dell'oscillatore stesso. L'utilità ai fini ingegneristici è quella di valutare in maniera diretta la risposta delle strutture, ancorchè semplificate, nei confronti di un sisma di assegnata entità. Si faccia riferimento, quindi ad una struttura schematizzabile da un oscillatore a singolo grado di libertà, soggetto ad una sollecitazione sismica, rappresentata dall'accelerazione $x(\ddot{g})$, indicando con:

- $u(t)$ lo spostamento della massa M rispetto al sistema di riferimento relativo;
- $x(t)$ lo spostamento della massa M nel sistema di riferimento assoluto O, che si può ricavare dallo spostamento relativo $u(t)$ e dallo spostamento del terreno $x_g(t)$ dalla relazione $x(t) = u(t) + x_g(t)$;
- ω la *frequenza propria* dell'oscillatore;
- ξ il *fattore di smorzamento*.

L'equazione del moto, nel sistema di riferimento relativo è:

$$M\ddot{u}(t) + C\dot{u}(t) + Ku(t) = -M\ddot{x}_g(t) \quad (2.11)$$

Da cui, dividendo per M, l'equazione 2.11 può essere scritta come:

$$\ddot{u}(t) + 2\xi\omega\dot{u}(t) + \omega^2u(t) = -\ddot{x}_g(t) \quad (2.12)$$

La precedente equazione differenziale, note le condizioni al contorno e l'equazione della forzante esterna, può essere risolta mediante integrazione per via numerica, data la natura casuale della forzante, che rappresenta il moto sismico. Tuttavia, in alcuni casi, per determinare gli effetti che un determinato terremoto può sortire sulle costruzioni, non è sempre necessario conoscere l'intera evoluzione temporale degli spostamenti indotti, spesso è sufficiente valutare gli effetti massimi. In particolare la risposta massima dell'oscillatore semplice considerato in precedenza può essere posta in termini di:

- $S_d(T, \xi) = |u_{\max}(t, T, \xi)|$, Spettro di risposta elastico in termini di spostamento;
- $S_v(T, \xi) = |\dot{u}_{\max}(t, T, \xi)| = S_d\xi$, Spettro di risposta elastico in termini di velocità;

- $S_a(T, \xi) = |\ddot{u}_{\max}(t, T, \xi)| = S_d \xi^2$, Spettro di risposta elastico in termini di accelerazione;

Mentre gli spettri di Fourier, espressi in ampiezza e in fase descrivono in maniera matematica completa l'andamento del moto sismico, lo spettro di risposta elastico fornisce, invece, una valutazione circa gli effetti indotti dal terremoto e ne indica la distribuzione in frequenza, nel senso che la risposta di un oscillatore sarà tanto più amplificata quanto la forzante ha una frequenza prossima a quella propria dell'oscillatore.

La ricerca dei massimi effetti indotti dal sisma, per un sistema a singolo grado di libertà, si traduce nella ricerca degli spostamenti massimi, l'equazione del moto (2.11) diventa, quindi, per essere nulla la velocità in corrispondenza dello spostamento massimo:

$$ku_{\max} = -M(\ddot{x}_g + \ddot{u}) \quad (2.13)$$

$$M\omega^2 u_{\max} = -M(\ddot{x}_g + \ddot{u}) \quad (2.14)$$

$$\omega^2 u_{\max} = -(\ddot{x}_g + \ddot{u}) \quad (2.15)$$

La quantità a primo membro nella 2.15, $\omega^2 u_{\max}$ ha le dimensioni di una accelerazione e prende, difatti, il nome di *pseudo-accelerazione* e, negli istanti in cui lo spostamento raggiunge il valore massimo, coincide con l'accelerazione assoluta dell'oscillatore ($\ddot{x}_g + \ddot{u}$). Nella pratica, e anche nei riferimenti normativi, è più pratico l'utilizzo di spettri di risposta espressi in pseudo-accelerazione, rispetto a spettri di risposta in accelerazione totale, poichè sono più semplici da ricavare essendo legati allo spostamento massimo dell'oscillatore dall'espressione:

$$S_{\text{pseudo-accelerazione}} = \omega^2 u_{\max}(t) \quad (2.16)$$

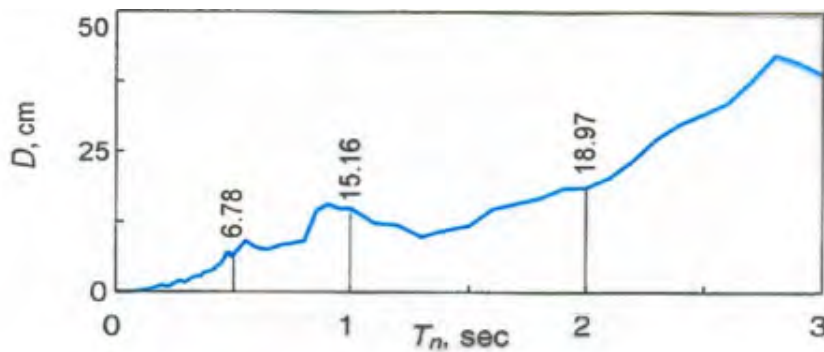


Figura 2.5: Spettro di risposta in termini di spostamento (D). Fonte: EL Centro, California

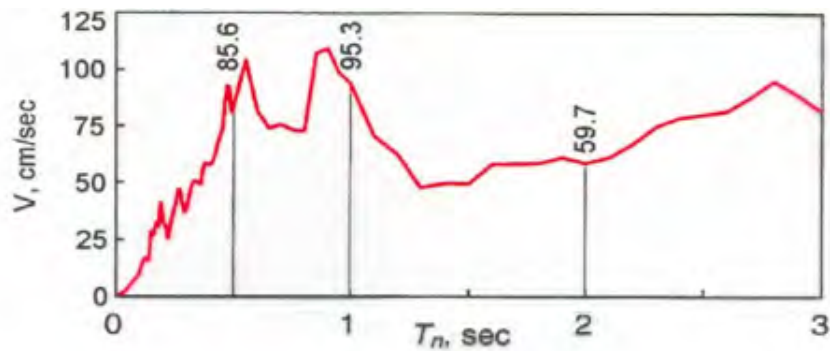


Figura 2.6: Spettro di risposta in termini pseudo-velocità (ωD). Fonte: EL Centro, California

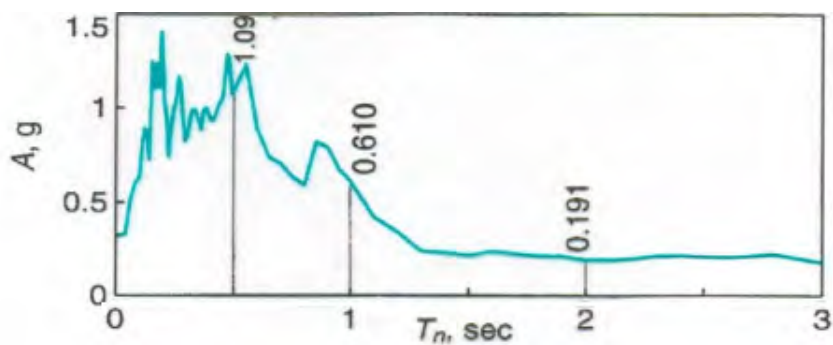


Figura 2.7: Spettro di risposta in termini di pseudo-accelerazione ($\omega^2 D$). Fonte: EL Centro, California

2.3.3 Spettro di risposta di progetto

Risulta chiaro che, qualora fosse necessario uno strumento per il calcolo delle azioni sismiche per il progetto di una nuova struttura o la verifica di costruzioni esistenti, non è sufficiente avere a disposizione uno spettro di risposta ricavato unicamente da un solo sisma, anche perchè non necessariamente diverse scosse devono presentare picchi di accelerazione nelle stesse frequenze. Quindi la normativa fornisce gli elementi per costruire uno spettro elastico di progetto inteso come combinazione di molti eventi attesi, sulla base di informazioni circa la sismicità locale del sito, che viene introdotta mediante l'accelerazione massima al suolo, e circa le caratteristiche dinamiche della struttura, individuate dal periodo proprio e dallo smorzamento. Lo spettro elastico di progetto si compone, pertanto, di una serie di curve combinate tra loro fino ad ottenere un risultato di questo tipo:

La normativa attuale fornisce le espressioni per la costruzione di detto spettro di risposta elastico. Il concetto alla base, per la costruzione di tale curva, è l'involuppo di due differenti spettri, il primo riferito a eventi attesi di modesta entità

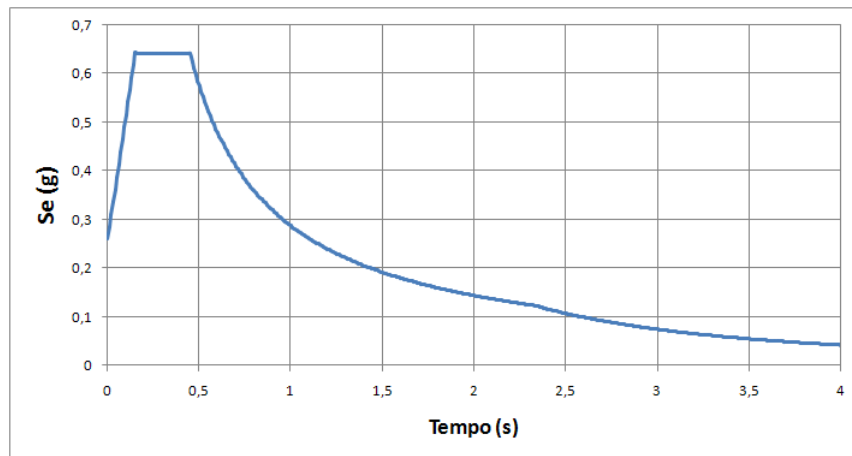


Figura 2.8: Spettro di risposta elastico, determinato tramite la normativa (N.T.C. 2008)

ma molto vicini al sito di progetto, il secondo invece si basa su eventi ad elevata intensità che però avvengono a grandi distanze dal sito, questo comporta che la curva risultante mostri due *plateaux*, il primo ad elevate frequenze, rappresentativo di eventi prossimi al sito, e il secondo, a basse frequenze, che testimonia l'effetto di terremoti ad elevata intensità che si propagano per lunghe distanze fino ad arrivare al sito di progetto.

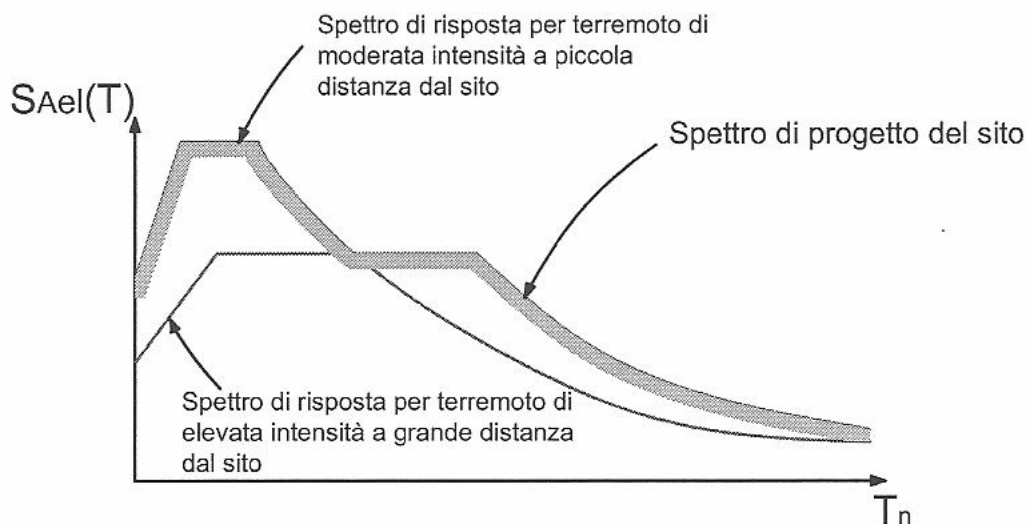


Figura 2.9: Spettro elastico di progetto a partire dall'involuppo di due spettri. (Chopra, 2001).

Come anticipato, la normativa italiana in vigore, fornisce delle espressioni per costruire spettri elastici di risposta *ad hoc* per l'analisi che si sta effettuando. Tali

relazioni prendono in input caratteristiche di sismicità del sito e caratteristiche meccaniche della struttura:

$$\begin{aligned}
 - 0 \leq T < T_B & \quad S_e(T) = a_g S \eta F_0 \left[\frac{T}{T_B} + \frac{1}{\eta F_0} \left(1 - \frac{T}{T_B} \right) \right] \\
 - T_B \leq T < T_C & \quad S_e(T) = a_g S \eta F_0 \\
 - T_C \leq T < T_D & \quad S_e(T) = a_g S \eta F_0 \left(\frac{T_C}{T} \right) \\
 - T_D \leq T & \quad S_e(T) = a_g S \eta F_0 \left(\frac{T_C T_D}{T^2} \right)
 \end{aligned}$$

Con il seguente significato dei termini:

a_g , accelerazione orizzontale massima su sito di riferimento rigido orizzontale;

T , periodo di vibrazione di un sistema lineare ad un grado di libertà;

$S_e(T)$ è l'accelerazione spettrale orizzontale;

S , coefficiente che tiene conto della categoria di sottosuolo e della posizione topografica;

η , fattore che altera lo spettro elastico per coefficienti di smorzamento viscosi ξ diversi dal 5% convenzionale;

F_0 , fattore che quantifica l'amplificazione spettrale massima, ha valore minimo pari a 2.2;

T_C è il periodo corrispondente all'inizio del tratto a velocità costante nello spettro, si calcola dalla relazione $T_C = C_C T^*_{*C}$ con T^*_{*C} valore tabellato e C_C coefficiente funzione della tipologia di sottosuolo;

T_B è il periodo corrispondente all'inizio del tratto dello spettro ad accelerazione costante, dato da $T_B = \frac{T_C}{3}$;

T_D è il periodo corrispondente all'inizio del tratto a spostamento costante dello spettro, si ricava dall'espressione $T_D = 4 \frac{a_g}{g}$ (valore in secondi).

Solitamente, dato il lungo periodo di ritorno, cioè l'intervallo di tempo col quale un terremoto di una certa intensità si manifesta mediamente, di scosse di elevata violenza, si ritiene ragionevole progettare le strutture sfruttando anche le loro capacità plastiche, cioè si ammette che per terremoti particolarmente forti, la struttura esca dal campo elastico e subisca danni plastici, quindi permanenti, sfruttando la sua duttilità, pur garantendo, tuttavia, la salvaguardia della vita delle persone. Per evitare di compiere analisi di tipo non lineare, che terrebbero conto in maniera coerente delle risorse plastiche delle strutture, a partire dallo spettro di risposta elastico, se ne ricava un secondo, detto *Spettro di progetto*, che risulta ridotto di un coefficiente, per tenere conto della capacità di dissipare

energia di una struttura, attraverso deformazioni plastiche di alcuni opportuni elementi. Tale coefficiente è detto *fattore di struttura*, q e sulla base dello spettro ridotto si possono compiere analisi lineari, dinamiche e statiche. Il fattore di struttura si calcola secondo l'espressione:

$$q = q_0 K_R$$

Con il seguente significato dei termini:

q_0 , è il valore massimo del fattore di struttura per una determinata costruzione, esso dipende dal livello di duttilità che ci si aspetta, dalla tipologia strutturale e dal rapporto α_u/α_1 , che è il quoziente tra il valore dell'azione sismica per il quale la struttura è resa labile e quello per il quale si ha la generazione della prima cerniera plastica;

K_R , è un fattore che tiene conto della riduzione della duttilità della struttura se questa non è regolare in altezza, vale 1 per strutture regolari, 0.8 viceversa.

Dal punto di vista grafico si ottiene un *abbassamento* della curva, nel senso che a parità di periodo, l'accelerazione che sollecita il sistema risulta ridotta:

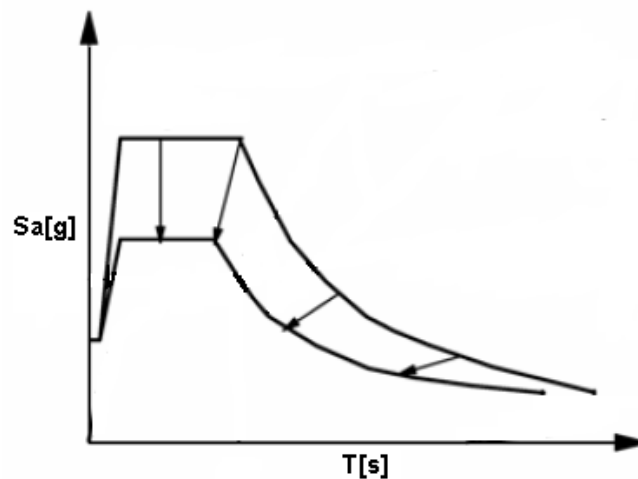


Figura 2.10: Spettro di risposta di progetto, si nota l'abbassamento della curva dovuto al fattore di struttura q (N.T.C. 2008)

Per il calcolo dello spettro di progetto, la normativa, fornisce un altro set di relazioni, di seguito riportate, con significato dei termini identico a quello sopra specificato per il caso dello spettro di risposta elastico.

$$- 0 \leq T < T_B \quad S_e(T) = a_g S \eta \frac{F_0}{q} \left[\frac{T}{T_B} + \frac{1}{\eta F_0} \left(1 - \frac{T}{T_B} \right) \right]$$

$$\begin{aligned}
 - T_B \leq T < T_C & \quad S_e(T) = a_g S \eta \frac{F_0}{q} \\
 - T_C \leq T < T_D & \quad S_e(T) = a_g S \eta \frac{F_0}{q} \left(\frac{T_C}{T} \right) \\
 - T_D \leq T & \quad S_e(T) = a_g S \eta \frac{F_0}{q} \left(\frac{T_C T_D}{T^2} \right)
 \end{aligned}$$

2.4 METODI DI ANALISI PER IL CALCOLO DELLA RISPOSTA STRUTTURALE ALL'AZIONE SISMICA

Esistono quattro grandi categorie per l'analisi strutturale nell'ambito dell'ingegneria sismica. Esse sono:

- Analisi statica lineare;
- Analisi dinamica lineare;
- Analisi statica non lineare;
- Analisi dinamica non lineare;

L'analisi statica lineare fornisce risultati attendibili solo nei casi in cui la struttura presenta elevata regolarità, in pianta e in elevazione. Comporta il vantaggio di una notevole semplicità e velocità di risoluzione. La sollecitazione sismica totale non è direttamente funzione delle caratteristiche elastiche della struttura, bensì dipende unicamente dalle masse in gioco e dall'accelerazione considerata.

Le analisi dinamiche lineari possono essere condotte mediante integrazione dell'equazioni del moto, illustrate in precedenza, oppure in forma modale, a partire da massa, rigidità e smorzamento della struttura e dallo spettro di risposta. Risulta chiaro che, nel caso di analisi dinamica, la forza sollecitante dovuta al sisma, dipende, tra le altre cose, anche dal periodo di oscillazione, e, di conseguenza, dalla rigidità dell'edificio.

Le procedure di analisi non lineari, invece, consistono nel monitoraggio delle deformazioni e negli spostamenti che si verificano nella struttura al variare della forzante sismica. Possono condursi mediante la risoluzione diretta dell'equazione del moto, oppure applicando delle forze orizzontali via via crescenti fino al raggiungimento di un limite prefissato (push-over). Il vantaggio delle analisi non lineari sta nel fatto che permettono una valutazione più coerente della risposta strutturale, tuttavia possono presentare alcune difficoltà, sia nella fase di taratura del modello, sia nell'interpretazione finale dei dati in output.

2.4.1 Analisi statica lineare

L'approccio al problema della progettazione sismica mediante valutazioni condotte con analisi statiche lineari è sicuramente il più semplice fra tutti. Esso porta a conclusioni buone in prima approssimazione, che possono servire come indicazioni orientative per le successive analisi più approfondite, o possono essere sufficienti come prescrizioni per la progettazione strutturale solo nei casi previsti dalla normativa, e cioè quando l'edificio sorge in un sito a bassa pericolosità sismica e qualora presenti notevoli caratteristiche di semplicità strutturale e regolarità.

In questo tipo di analisi la sollecitazione sismica viene tradotta in uno schema di forze orizzontali *statiche equivalenti*, e quindi vengono calcolate deformazioni e tensioni con il classico schema statico lineare, la semplicità di questa analisi deriva dal fatto che le rigidezze e le forze sono sempre costanti nel tempo, vengono pertanto del tutto ignorate la capacità di plasticizzazione della struttura.

Per il calcolo delle forze orizzontali equivalenti, con riferimento alla normativa italiana in particolare, bisogna osservare il primo modo di vibrare della struttura, e noto il periodo, si entra in ascissa dello spettro di progetto, espresso in accelerazione. Noto il peso della struttura è immediato calcolare la sollecitazione di taglio massimo alla base dei pilastri della struttura. L'espressione riportata in normativa ricalca la basilare formula $F = Ma$, definito con W il peso della struttura, T_1 il periodo di oscillazione del primo modo di vibrare, si ha:

$$V_{\text{base}max} = \frac{S_{A,\text{progetto}(T_1)}}{g} W \quad (2.17)$$

Se la costruzione è formata da più piani, le forze orizzontali sono decrescenti in altezza, secondo la relazione:

$$F_i = F_h \frac{Z_i W_i}{\sum_{j=1}^n (z_j W_j)} \quad (2.18)$$

Dove si è indicato con F_h la forza di taglio alla base, con W_i W_j i pesi di due piani successivi e con z_i z_j le altezze di due piani successivi rispetto alla base. Si ottiene, pertanto una distribuzione di azioni orizzontali equivalente riportata in Figura 2.11.

Questo tipo di analisi fornisce risultati accettabili solo nel caso in cui il primo modo è predominante, tipicamente quando coinvolge una massa partecipante di almeno il 90%, e se i successivi modi di vibrare non presentano periodi propri molto differenti dal primo. Solitamente le caratteristiche richieste vengono rispettate solo in caso di edifici che presentano notevole semplicità e regolarità strutturale,

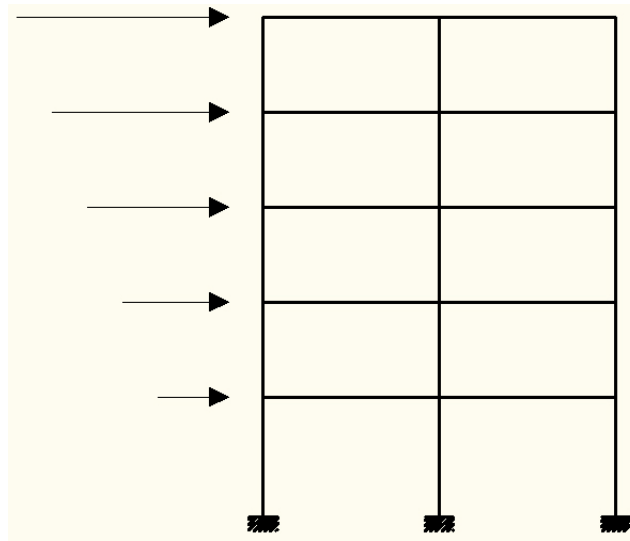


Figura 2.11: Distribuzione delle azioni orizzontali su strutture a telaio

restringendo, di fatto, la legittimità dell'applicabilità delle analisi statiche lineari solo in tale campo.

2.4.2 Analisi dinamica lineare

L'analisi dinamica lineare, in comune con l'analisi statica lineare, considera sempre costante la rigidezza degli elementi strutturali al variare dell'azione sismica, anche in questo tipo di analisi vengono considerate le riserve di deformazione plastica della struttura in maniera forfaitaria attraverso il coefficiente di struttura q . Tuttavia, in questo caso, non si considera un'azione orizzontale equivalente all'effetto del sisma costante, ma variabile in funzione dei modi di vibrare della struttura. Il procedimento prevede la creazione di un modello tridimensionale

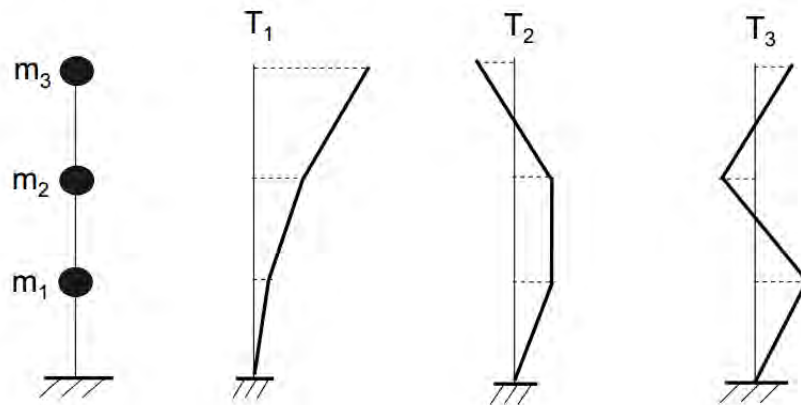


Figura 2.12: Modi di vibrare di una struttura semplice

della struttura e l'individuazione dei suoi modi di vibrare, sono ritenuti consistenti tutti i modi con massa partecipante superiore al 5%, e ne vanno considerati un numero tale per cui la somma della massa partecipante di ciascuno raggiunga l'85%. Poi la sollecitazione sismica per ciascun modo di vibrare viene calcolata a partire da uno spettro di risposta di progetto. Infine, i parametri massimi, siano essi di spostamento o di sollecitazione, sono calcolati mediante una combinazione statistica della risposta ottenuta per ciascun modo. Le formule di combinazione statistica sono due: *SRSS* (radice quadrata della somma dei quadrati delle risposte modali, *Square Root Sum of Squares*) e *CQC* (combinazione quadratica completa, *Complete Quadratic Combination*).

In questo tipo di analisi riveste molta importanza una modellazione ragionata della struttura, infatti, esasperando il dettaglio nel modello tridimensionale si potrebbero avere come unica conseguenza dei modi di vibrare improbabili.

Un altro tipo di procedura di analisi dinamica lineare prevede l'integrazione delle equazioni del moto, sempre considerando un comportamento elastico lineare degli elementi. Al modello tridimensionale vengono applicati degli accelerogrammi spettro compatibili (terne di accelerogrammi nelle tre direzioni) e si conduce un'analisi in *Time-history* che consente di determinare le storie temporali delle risposte nodali anche in termini di spettri di accelerazione. La normativa impone l'utilizzo di almeno tre terne di accelerogrammi, al fine di individuare le condizioni di maggiore sollecitazione data la suscettibilità della soluzione da tali spettri.

2.4.3 Analisi statica non lineare

L'analisi statica non lineare prevede di caricare la struttura con delle forze incrementali passo-passo e leggere di conseguenza gli spostamenti di alcuni punti della struttura, detti *punti di controllo*, ad ogni step il modello strutturale viene aggiornato per tenere conto dell'eventuale insorgenza di fenomeni non lineari, quali plasticizzazioni, rotture fragili, effetti P-delta, rotture locali nei nodi.

Il sistema di forze orizzontali e gravitazionali sarà opportunamente scalato ad ogni incremento di carico al fine di far crescere in maniera monotona lo spostamento orizzontale di ogni punto di controllo, fino al raggiungimento delle condizioni ultime. Il risultato finale sarà un diagramma nel quale verrà riportato in ascissa lo spostamento orizzontale del punto di controllo e in ordinata la forza orizzontale totale applicata, cioè il taglio alla base. Un'informazione molto sintetica e importante è data dalla *curva di capacità* della struttura, che è un diagram-

ma riportante lo spostamento orizzontale del baricentro del solaio di copertura in funzione della risultante delle forze orizzontali applicate. Infine si confronta la curva di capacità così ottenuta con la domanda del sisma che viene generata rappresentando gli spettri di progetto in un piano con le pseudo-accelerazioni in ordinata e gli spostamenti in ascissa. L'indubbio vantaggio di questo tipo di analisi è che riesce a riportare l'effettiva capacità plastica della struttura ed eventuali meccanismi di collasso. Tuttavia la principale limitazione è che per tutta la durata della simulazione viene considerato solo un modo di vibrare, come se la reale risposta sismica di una struttura fosse assimilabile a quella di un oscillatore semplice, ipotesi non sempre verificata, poichè in alcune strutture irregolari possono avere molta importanza anche modi di vibrare inferiori.

2.4.4 Analisi dinamica non lineare

Nell'analisi dinamica non lineare è prevista la variazione sia delle forze orizzontali introdotte dal sisma, sia della rigidità degli elementi strutturali per l'insorgere di fenomeni non lineari. Per rappresentare adeguatamente il problema è necessario fornire uno spettro di progetto, mediante accelerogrammi spettro-compatibili, e un modello tridimensionale della struttura nella quale ogni elemento è caratterizzato sia geometricamente sia meccanicamente, al fine di riportare nella maniera più fedele possibile eventuali comportamenti anelastici. La soluzione si ottiene pertanto, mediante algoritmi numerici che provvedono ad un'integrazione al passo delle equazioni generali del moto nel dominio del tempo, aggiornando il modello *event by event* se dovessero eventualmente insorgere fenomeni anelastici. La soluzione che si ottiene è la funzione di spostamento $x(t)$ al variare del tempo.

Questi tipi di analisi sono fortemente influenzate dalle terne di accelerogrammi utilizzate, per questo la normativa impone di utilizzarne almeno tre, e anche dalla accuratezza della modellazione della struttura, comportano un elevato carico computazionale ma rappresentano, se ben poste, l'approccio più valido per descrivere il problema sismico e la risposta di una struttura.

Capitolo 3

VULNERABILITÀ SISMICA E CURVE DI FRAGILITÀ

3.1 RISCHIO SISMICO E PERICOLOSITÀ SISMICA

L'azione sismica ha delle peculiarità specifiche se presa in confronto rispetto alle altre azioni: si tratta di un'accelerazione alla base piuttosto che un carico ben definito (come invece il peso proprio, il vento, la neve), la risposta della struttura è dinamica e deve essere correlata anche al danno che questa subisce dopo ripetuti cicli sismici, infine, l'effetto del sisma coinvolge l'intero sistema, laddove l'influenza degli altri carichi è quasi sempre di tipo locale. La filosofia alla base della progettazione anti-sismica è quella di conferire alla struttura, se sollecitata da un terremoto di forte entità, la capacità di salvaguardia della vita degli abitanti, accettando deformazioni plastiche anche gravi, ma comunque mantenendo un certo grado di sicurezza dal collasso, piuttosto che creare un edificio in grado di restare in campo elastico anche per azioni sismiche elevate.

Al fine di raggiungere tali obiettivi, tutte le grandezze in gioco, all'interno del processo di progettazione, sono state classificate in elementi più semplici, in termini di: parametri di misura per l'intensità e la definizione del terremoto (IMs), parametri per la domanda strutturale (EDPs), misure di danno (DMs) e variabili decisionali (DVs). Degli esempi comuni per i parametri sopra citati sono l'accelerazione di picco al suolo e la frequenza propria del primo modo di vibrare (IMs), le componenti di deformazione inelastica (EDPs), livelli di danno per elementi strutturali e non strutturali (DMs) e perdite economiche dirette e numero di vittime (DVs). Esistono dei metodi probabilistici consolidati che permettono di

quantificare in maniera rigorosa l'aleatorietà delle grandezze sopra riportate, ad esempio, la frequenza media annua dei parametri DV viene ottenuta applicando il teorema della probabilità totale (Cornell & Krawinkler, 2000; Yeo & Cornell, 2005):

$$\iiint P(DV|DM) \cdot |dP(DM|EDP)| \cdot |dP(EDP|IM)| \cdot |dH(IM)| \quad (3.1)$$

dove:

- **P(DV|DM)** è la probabilità che DV ecceda un valore specifico, condizionata dal danno strutturale, DM. Si può stimare il valore di DV a partire da analisi probabilistiche basate su perdite economiche dirette, con qualche difficoltà data l'incertezza sui dati in input;
- **P(DM|EDP)** è la probabilità che DM superi un determinato valore, assegnato una quantità al parametro EDP. Dal momento che questo termine considera diversi IMs, è anche chiamato *fragilità sismica*;
- **P(EDP|IM)** è la probabilità che EDP superi un certo valore assegnato al parametro IM;
- **H(IM)** è il rischio sismico del sito, viene ottenuto mediante analisi probabilistiche.

L'equazione 3.1 è comunemente nota come *PEER equation*, dal nome del dipartimento da cui è stata formulata (*Pacific Earthquake Engineering Research Center*). Ogni termine dell'equazione precedente è costruito in maniera tale da richiedere dati in ingresso da una specifica disciplina, per esempio H(IM) deve essere determinato da sismologi ed ingegneri geotecnici, P(EDP|IM) da ingegneri strutturisti e P(DV|DM) da specialisti in estimo e valutazione economica. Il metodo basato sulla valutazione dei parametri PEER è mostrato in Figura 3.1. L'equazione PEER è composta da coppie di quattro variabili aleatorie totali: questo per permettere una soluzione dove il singolo termine non è condizionato dai restanti, ma solo da uno. In altre parole, questo significa che $P[DM > x|EDP = y \text{ e } IM = z]$ è del tutto uguale a $P[DM > x|EDP = y]$, essendo tale probabilità unicamente funzione di y. Questo comporta che EDP viene determinato in modo univoco da DM e pertanto, non sarà necessario includere IM nell'equazione. La formula PEER è stata sviluppata considerando scosse sismiche di entità rilevante, modellate mediante processi omogenei alla Poisson con parametri indipendenti dal tempo. Inoltre, dal momento che le analisi probabilistiche per il rischio sismico del sito sono descritte usando i valori medi annuali dei terremoti, anche la conseguente

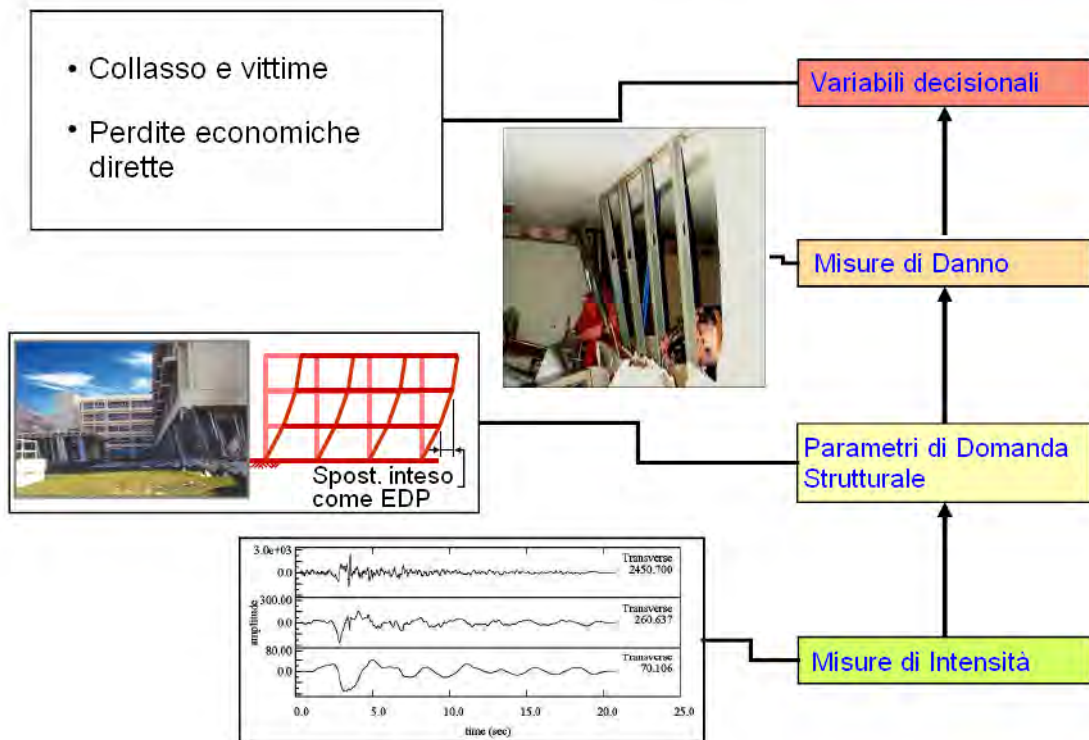


Figura 3.1: Metodo basato sui parametri PEER .

distribuzione probabilistica di DV risulterà su base annuale. Queste analisi sono compatibili con gli attuali studi sulla sicurezza sismica, dal momento che sono comunemente espressi in termini di frequenza annuale media del numero di vittime. Da quanto esposto si intende che l'equazione 3.1 è implicitamente basata sulla durata temporale annuale e, dato lo scarso numero di terremoti rilevanti che si manifestano in tale periodo, la probabilità del verificarsi di multipli eventi sismici di una certa entità è quasi nulla. Se, invece, si considerasse un intervallo temporale diverso dal singolo anno, ad esempio un tempo generico T , il numero medio di eventi che superino $DV = X$ nel tempo T diventa $P(DV)cT$. Se T fosse un intervallo molto esteso, ad esempio la vita nominale di una struttura, allora la probabilità del verificarsi di multipli eventi sismici significativi diventerebbe rilevante. In questo caso, il metodo PEER, assume che, in seguito ad un terremoto di una certa entità, la struttura venga ripristinata alle condizioni iniziali prima del verificarsi dell'evento sismico successivo.

Considerando tutti questi aspetti, il rischio sismico può essere visto come la probabilità di osservare una certa perdita economica in uno specifico sito durante un determinato periodo di tempo: questo dipenderà dalla pericolosità sismica,

dalla vulnerabilità della struttura e dal *livello di esposizione*. Quest'ultimo parametro rappresenta la stima economica delle strutture considerate, della popolazione e delle attività economiche e servizi pubblici soggetti alla minaccia sismica nel sito di osservazione. Alla luce di quanto esposto, l'applicazione del metodo per la valutazione del rischio sismico comporta non poche difficoltà, in particolare nella valutazione del livello di esposizione. Di conseguenza, si preferisce intendere il rischio sismico come la probabilità di osservare un certo livello di danno all'interno di un determinato periodo di tempo per uno specifico sito, così facendo si tiene conto solo della vulnerabilità e della pericolosità sismica. Il parametro così calcolato è solitamente chiamato *rischio sismico specifico*, l'equazione 3.1 risulta allora semplificata:

$$\int P(DV|DM) \cdot |dH(IM)| \quad (3.2)$$

dove, in questo caso, DV coincide con DM. Una volta stabiliti un set di *Performance Levels* (PLs), la valutazione del danno atteso si ottiene a partire dalla funzione densità di probabilità della pericolosità sismica:

$$\int_{IM} P(D > d_{PL}|IM) \cdot \left| \frac{dH(IM)}{dIM} \right| dIM \quad (3.3)$$

dove la funzione $P(D > d_{PL}|IM)$ è la curva di fragilità associata al livello di danno d_{PL} e $H(IM)$ è la funzione della pericolosità sismica. L'integrale nell'equazione 3.2 è noto come integrale di convoluzione. Se IM è una variabile discreta con un set di valori finito, applicando il teorema della probabilità totale, è possibile riscrivere l'equazione 3.3 come segue (Ellingwood & Kinali, 2009):

$$\sum_{IM} P(D > d_{PL}|IM) \cdot |\Delta H(IM)| \quad (3.4)$$

Passiamo ora ad analizzare brevemente il concetto di pericolosità sismica, *seismic hazard*. Questo parametro rappresenta la probabilità di osservare un certo livello di scuotimento del suolo, espresso mediante PGA o accelerazione spettrale ad 1 secondo, $S_a(1s)$, in un determinato intervallo temporale e in uno specifico sito. Occorre, tuttavia, distinguere fra pericolosità sismica standard e locale. La pericolosità sismica standard si riferisce a delle condizioni standard del terreno, mentre la pericolosità sismica locale identifica gli effetti locali determinati dalle condizioni geomorfologiche del sito in esame.

Questo parametro viene rappresentato mediante le curve di pericolosità sismica, che solitamente rappresentano la frequenza media annua con la quale viene

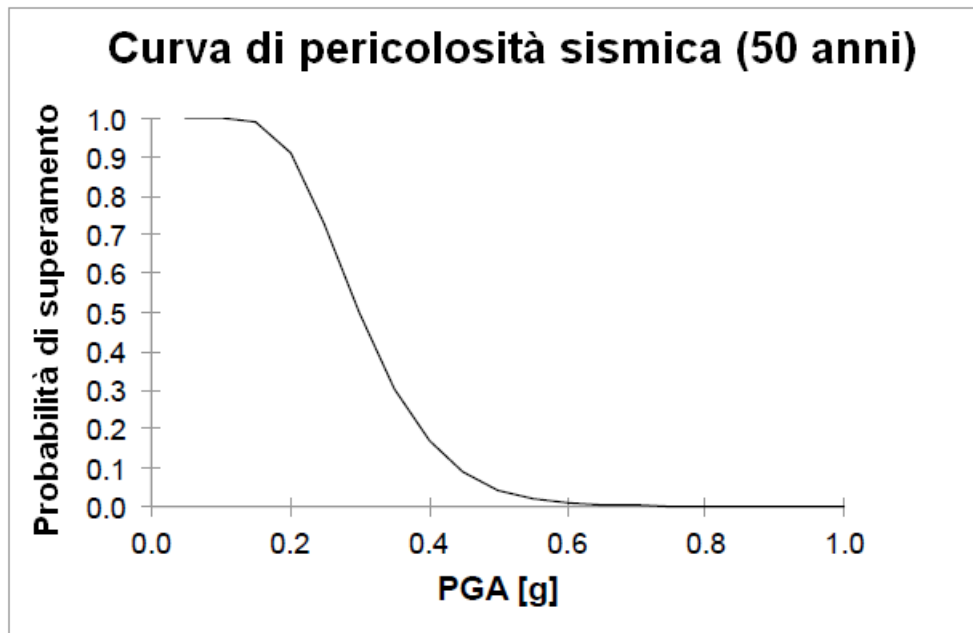


Figura 3.2: Esempio della curva della pericolosità sismica riferita ad un tempo $T=50$ anni.

eguagliato o superato un determinato valore di azione sismica, (Figura 3.2). È possibile ottenere tali diagrammi mediante un'analisi probabilistica per la pericolosità sismica, (*Probabilistic Seismic Hazard Analysis - PSHA*), utilizzando una distribuzione di probabilità di tipo lognormale. Al giorno d'oggi questa metodologia rappresenta ancora la strada seguita nella maggiore parte dei casi per la redazione delle mappe regionali della pericolosità sismica.

3.2 VULNERABILITÀ SISMICA

Con il termine vulnerabilità sismica di un sistema strutturale, intendiamo la conoscenza preventiva del livello di danno cui può andare incontro a seguito di un evento sismico di prefissata entità. Uno strumento molto valido per la valutazione della vulnerabilità di ponti e viadotti è rappresentato dalle *curve di fragilità*, esse non sono altro che delle curve parametriche che, data una certa intensità dell'evento sismico, riportano in ordinata la probabilità condizionata di eccedere o eguagliare un prefissato livello di danno. Esistono diverse curve di fragilità, distinte sulla base del meccanismo di rottura, che può essere di tipo duttile (rottura a flessione), il quale ammette molteplici livelli di danno, o di tipo fragile (rottura a taglio).

È evidente la natura probabilistica dell'intero problema, essa deriva, infatti, dalla aleatorietà delle variabili in gioco: il sisma in primo luogo ha caratteristiche difficilmente prevedibili, ed esso rappresenta la domanda a cui viene sottoposta la struttura, ma anche i materiali da costruzione, acciaio e calcestruzzo, hanno delle performance meccaniche soggette a qualche incertezza in fase di progettazione, ed essi rappresentano la capacità che è in grado di fornire il sistema. La figura seguente è molto esplicitiva da questo punto di vista, in quanto le curve, rispettivamente di capacità e domanda, sono caratterizzate da una loro distribuzione probabilistica, e la loro intersezione, il *punto di performance* della struttura, non è pertanto individuata da un singolo punto, ma da un intervallo.

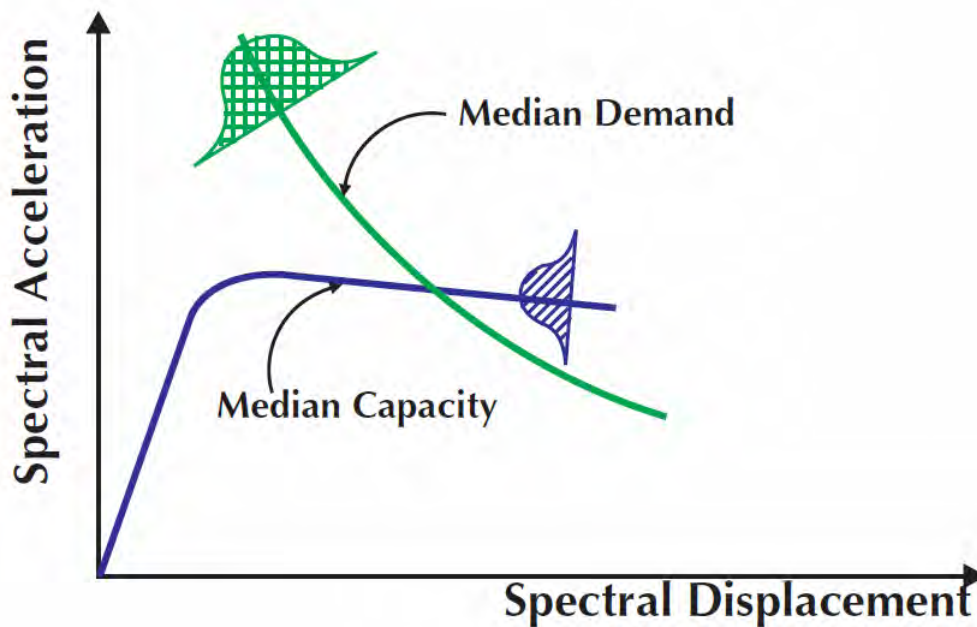


Figura 3.3: Aleatorietà delle curve di capacità della struttura e della domanda sismica [Mander et al. 1999]

Di seguito si riportano, sinteticamente, le fasi per la valutazione della affidabilità di strutture da ponte nei confronti dell'azione sismica mediante curve di fragilità:

- definizione di opportuni livelli di danno a cui può andare incontro la struttura (*PL*, *performance level*);
- costruzione delle curve di fragilità per ogni livello di danno;
- definizione delle probabilità di eguagliamento o superamento accettabili per ogni *PL*;

- confronto delle probabilità di superamento accettabili con le curve di fragilità, per ciascun PL a parità di intensità sismica.

Con riferimento alla costruzione delle curve di fragilità troviamo molti esempi in letteratura, specialmente in quella anglosassone e giapponese. Il primo procedimento è di natura empirica e si basa sull'osservazione di ciò che avviene sulle strutture a seguito di episodi sismici (Basoz, terremoto di Northridge, 1994 e Shinouzuka, terremoto di Kobe, 1995). A questo tipo di approccio si rifanno le normative americane (*hazus*) e europee (*Risk-UE*). Sul suolo italiano, data la scarsità di informazioni storiche circa il danneggiamento di strutture da ponte esistenti ad opera del terremoto, si avrebbe una qualche difficoltà nell'applicare metodi di tipo empirico, in tal caso esiste la necessità di seguire un approccio di tipo *analitico* mediante modelli numerici, che può essere brevemente riassunto nei passi seguenti:

- simulazione numerica dell'azione del terremoto mediante accelerogrammi registrati o spettro-compatibili;
- simulazione della risposta del ponte al terremoto mediante modelli a elementi finiti, costruendo diversi ponte campione per tenere conto della variabilità sui materiali;
- costruzione delle curve di fragilità a partire dagli spostamenti di opportuni punti di controllo, sperimentati dal modello del ponte durante la simulazione numerica;

I dati della risposta strutturale della struttura da ponte soggetta all'azione sismica possono essere reperiti attraverso differenti tipi di analisi. Sono state diffusamente impiegate analisi statiche non lineari (push-over), apprezzate per il modesto carico computazionale (Shinouzuka et al., 2000). Oppure, come nell'ambito del presente lavoro di tesi, possono essere impiegate analisi dinamiche non lineari in time-history, maggiormente gravose dal punto di vista computazionale, ma in grado di fornire risultati più attendibili, specialmente nel caso di modelli tridimensionali. Verrà di seguito approfondito, per completezza, il metodo empirico risk-UE e quindi il metodo analitico.

3.3 CURVE DI FRAGILITÀ EMPIRICHE METODO RISK-UE

In sintesi questo metodo, contemplato nel sistema americano di gestione delle infrastrutture Hazus, prevede la costruzione delle curve di fragilità basandosi su una banca dati di eventi sismici già successi su ponti esistenti, senza creare nessun modello analitico per il ponte in esame. Come è già stato accennato sia l'azione sismica che le performance meccaniche dei materiali non possono essere descritte in maniera deterministica, data l'incertezza sui valori da assumere, bensì si adattano bene ad essere rappresentati mediante una distribuzione di tipo logonormale. Di conseguenza le curve di capacità e di domanda saranno descritte dalla medesima funzione probabilistica, e per il teorema del limite centrale, anche il punto della loro intersezione, il punto di performance, sarà rappresentabile mediante tale distribuzione. Per identificare in maniera univoca la curva di fragilità sono necessari due parametri: il *valore medio* (valore con probabilità di accadimento del 50%) e la *deviazione standard*, quindi la funzione di probabilità cumulata viene descritta dalla seguente relazione:

$$F(S_a) = \Phi \left[\frac{1}{\beta_c} \ln \left(\frac{S_a}{A_i} \right) \right] \quad (3.5)$$

con il seguente significato dei termini:

- F è la funzione di distribuzione logonormale;
- S_a è l'accelerazione spettrale;
- A_i è l'accelerazione spettrale media per ottenere il livello di danno richiesto;
- β_c è la deviazione standard che incorpora tutti gli aspetti dovuti alla aleatorietà della capacità e della domanda; tale valore è stato per molto tempo oggetto di ricerche e di norma si utilizza il valore 0,6.

Questo metodo prevede lo sviluppo delle curve di fragilità basandosi su alcune caratteristiche macroscopiche, che derivano dalla struttura e dalla geometria del ponte. Per facilitare tale distinzione, si ha una classificazione dei ponti in categorie in funzione della loro tipologia e delle loro caratteristiche dimensionali, secondo la tabella seguente:

Per costruire la curva di fragilità di uno specifico ponte, è previsto l'adattamento di una curva preesistente di uno dei *ponti standard* elencati nella tabella

Material	Column bent type	Span continuity	Design	Category
All	Single Span	-	Conventional	1
			Seismic	2
Concrete bridges	Single	Simple Support	Conventional	3
			Seismic	4
		Continuous	Conventional	5
			Seismic	6
	Multiple	Simple Support	Conventional	7
			Seismic	8
Continuous	Conventional	9		
	Seismic	10		
Steel bridges	Multiple	Simple Support	Conventional	11
			Seismic	12
	All	Continuous	Conventional	13
			Seismic	14
	Other			15

Figura 3.4: Classificazione dei ponti in base alle loro caratteristiche geometrico-strutturali per il metodo RISK-EU (fonte manuale RISK-UE).

precedente. Con il termine ponte standard ci si riferisce ad un viadotto sufficientemente lungo in modo da rendere trascurabili gli effetti tridimensionali. Poi si individuano i valori medi delle accelerazioni spettrali a un secondo, $S_a(1s)$, che determinano sul ponte standard un certo livello di danno, questi valori vengono quindi modificati attraverso opportuni coefficienti per adattarli alle caratteristiche del ponte specifico in esame. Tali coefficienti tengono conto di caratteristiche geometriche del ponte e del suo sito di collocazione:

- $K_{skew} = \sqrt{\sin \alpha_{skew}}$ coefficiente che tiene conto della sghembatura del ponte in pianta;
- $K_{shape} = 2,5 \frac{S_a(1,0)}{S_a(0,3)}$ coefficiente che tiene conto del luogo ove sorge la struttura, le accelerazioni spettrali sono infatti ricavate in funzione del sito in esame;
- $K_{3D} = f(N)$ coefficiente che tiene conto di eventuali effetti tridimensionali che affliggono il ponte, dipende dal numero di campate;

I valori dei coefficienti sopra riportati e delle medie spettrali modificate sono espressi nella tabella seguente.

In sintesi, questo metodo prevede delle valutazioni circa il sito dove sorge il ponte in esame, infatti tali informazioni sono necessarie per la costruzione dello spettro e l'individuazione delle accelerazioni $S_a(1,0)$, $S_a(0,3)$, sia circa le peculiarità del ponte stesso: dalla tipologia fino ad arrivare a caratteristiche geometriche

Skew angle (K_{skew})	3-dimensional arch action in the deck (K_{3D})						
	EQ1	EQ2	EQ3	EQ4	EQ5	EQ6	EQ7
$\sqrt{\sin(90-\alpha)}$	$1+\frac{0.25}{N-1}$	$1+\frac{0.33}{N}$	$1+\frac{0.33}{N-1}$	$1+\frac{0.33}{N-1}$	$1+\frac{0.05}{N}$	$1+\frac{0.20}{N-1}$	$1+\frac{0.10}{N}$

Typology	Damage state			
	Minor	Moderate	Extensive	Complete
Category	Median SA at 1.0 s (g) with $\beta=0.6$			
1-2	$0.8 \times \min\left(1; 2.5 \times \frac{SA(1.0)}{SA(0.3)}\right)$	$1.0 \times K_{skew} \times EQ1$	$1.2 \times K_{skew} \times EQ1$	$1.7 \times K_{skew} \times EQ1$
3	0.25	$0.35 \times K_{skew} \times EQ1$	$0.45 \times K_{skew} \times EQ1$	$0.70 \times K_{skew} \times EQ1$
4	0.50	$0.80 \times K_{skew} \times EQ1$	$1.10 \times K_{skew} \times EQ1$	$1.70 \times K_{skew} \times EQ1$
5	0.35	$0.45 \times K_{skew} \times EQ2$	$0.55 \times K_{skew} \times EQ2$	$0.80 \times K_{skew} \times EQ2$
6	0.60	$0.90 \times K_{skew} \times EQ3$	$1.30 \times K_{skew} \times EQ3$	$1.60 \times K_{skew} \times EQ3$
7	0.25	$0.35 \times K_{skew} \times EQ1$	$0.45 \times K_{skew} \times EQ1$	$0.70 \times K_{skew} \times EQ1$
8	0.50	$0.80 \times K_{skew} \times EQ1$	$1.10 \times K_{skew} \times EQ1$	$1.70 \times K_{skew} \times EQ1$
9	$0.60 \times \min\left(1; 2.5 \times \frac{SA(1.0)}{SA(0.3)}\right)$	$0.90 \times K_{skew} \times EQ2$	$1.10 \times K_{skew} \times EQ2$	$1.50 \times K_{skew} \times EQ2$
10	$0.90 \times \min\left(1; 2.5 \times \frac{SA(1.0)}{SA(0.3)}\right)$	$0.90 \times K_{skew} \times EQ3$	$1.10 \times K_{skew} \times EQ3$	$1.50 \times K_{skew} \times EQ3$
11	0.25	$0.35 \times K_{skew} \times EQ4$	$0.45 \times K_{skew} \times EQ4$	$0.70 \times K_{skew} \times EQ4$
12	0.50	$0.80 \times K_{skew} \times EQ1$	$1.10 \times K_{skew} \times EQ1$	$1.70 \times K_{skew} \times EQ1$
13	$0.75 \times \min\left(1; 2.5 \times \frac{SA(1.0)}{SA(0.3)}\right)$	$0.75 \times K_{skew} \times EQ5$	$0.75 \times K_{skew} \times EQ5$	$1.10 \times K_{skew} \times EQ5$
14	$0.90 \times \min\left(1; 2.5 \times \frac{SA(1.0)}{SA(0.3)}\right)$	$0.90 \times K_{skew} \times EQ3$	$1.10 \times K_{skew} \times EQ3$	$1.50 \times K_{skew} \times EQ3$
15	0.80	1.00	1.20	1.70

Figura 3.5: Valori dei coefficienti e delle medie spettrali modificate per la costruzione di curve di fragilità specifiche (fonte manuale RISK-UE).

come il numero di campate (N), la larghezza dell'impalcato (W), la lunghezza massima delle campate (L_{max}) e la lunghezza complessiva (L). Con tutte queste informazioni è possibile calcolare i coefficienti e quindi le accelerazioni medie modificate per il ponte specifico, che permette l'adattamento delle curve di fragilità dei ponti standard della medesima tipologia.

Infine, assumendo 0,6 come deviazione standard, è possibile tracciare le curve di fragilità in funzione di ciascun livello di danno. I livelli di danno o *Performance Level* sono quattro: *Slight*, *Moderate*, *Extensive*, *Complete* e la loro descrizione nell'ambito dell'analisi RISK-UE è riportata nella Figura 3.6.

Di seguito, in Figura 3.7, vengono riportate le curve risultanti del metodo sopra esposto, si notano quattro curve relative ad altrettanti livelli di danno. La lettura del grafico è immediata: per ogni intensità del sisma atteso si riscontra in

Damage States	Description
No Damage (N)	No damage to a bridge.
Slight/Minor Damage (S)	Minor cracking and spalling to the abutment, cracks in shear keys at abutments, minor spalling and cracks at hinges, minor spalling at the column (damage requires no more cosmetic repair) or minor cracking to the deck.
Moderate Damage (M)	Any column experiencing moderate cracking and spalling (column structurally still sound), any connection having cracked shear keys or bent bolts, or moderate settlement of the approach.
Extensive Damage (E)	Any column degrading without collapse (column structurally unsafe), any connection losing some bearing support, or major settlement of the approach.
Complete Damage (C)	Any column collapsing and connection losing all bearing support, which may lead to imminent deck collapse.

Figura 3.6: Livelli di danno o Performance Level secondo il metodo RISK-EU (fonte manuale RISK-UE).

ordinata la probabilità di superamento di un determinato Performance Level per il ponte in esame.

3.4 CURVE DI FRAGILITÀ ANALITICHE

3.4.1 Definizione degli indici di danno

Una delle caratteristiche negative che maggiormente affligge i metodi empirici per la costruzione delle curve di fragilità risiede nel fatto che la funzione di danno così costruita è di tipo qualitativo, fornisce cioè sono delle indicazioni a livello descrittivo, senza fornire alcun valore numerico. Nell'ambito della ricerca di un metodo per la costruzione di curve di fragilità quantitative, cioè in grado di associare un valore numerico ad ogni livello di danno, si collocano i metodi di tipo *analitico*, che prevedono come principali parametri per la valutazione del danneggiamento strutturale la *duttilità richiesta* e *l'energia dissipata*.

Il primo parametro, la duttilità richiesta, intende uno spostamento massimo in corrispondenza del quale si verifica il collasso della struttura, tale valore può essere determinato attraverso una prova monotona o mediante il grafico momento-curvatura riportato dalla normativa. il secondo parametro, l'energia dissipata plasticamente, intende che la struttura ha una capacità di dissipare, mediante deformazioni plastiche, l'energia del sisma e che si raggiunga il collasso

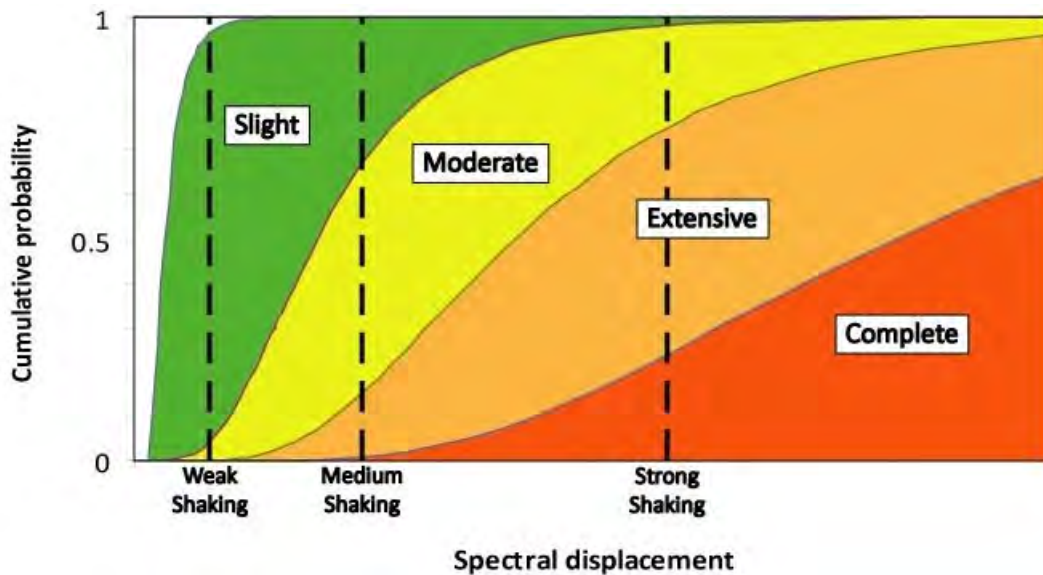


Figura 3.7: Esempio di curve di fragilità tracciate per uno specifico ponte mediante il metodo RISK-UE

quando, attraverso carichi ciclici, la struttura raggiunga tale valore di energia dissipata.

Nell'ambito di questo lavoro di tesi si assume come indicatore di danno la duttilità cinematica:

$$D = \mu_{\sigma} = \frac{x_{\max}}{x_y} \quad (3.6)$$

Dove:

- x_{\max} è lo spostamento massimo del punto di controllo durante la simulazione sismica. Nell'ambito della presente trattazione è stato assunto come punto di controllo la testa della pila e durante la simulazione numerica del terremoto ne vengono registrati gli spostamenti in direzione longitudinale e trasversale.
- x_y è lo spostamento a snervamento del medesimo punto di controllo.

In questo metodo di costruzione delle curve di fragilità un aspetto critico è la suddivisione dei livelli di danno, dovendo confrontare l'analisi non inerte di strutture esistenti con l'analisi del danno subito in seguito a terremoti realmente accaduti. In letteratura possiamo riscontrare numerosi studi che cercano di determinare i diversi livelli di danno mediante una corrispondenza ai livelli descrittivi di danno riportati dalla normativa anglosassone Hazus, (Choi, 2002):

Come già specificato la classificazione riportata nella tabella precedente fa riferimento alla normativa americana, per cui tali prescrizioni sarebbero da appli-

Definition of damage states for bridge components				
Damage state	Slight damage	Moderate damage	Extensive damage	Complete damage
Columns (μ)	$1.0 < \mu < 2.0$	$2.0 < \mu < 4.0$	$4.0 < \mu < 7.0$	$7.0 < \mu$
Steel bearings (δ , mm)	$1 < \delta < 6$	$6 < \delta < 20$	$20 < \delta < 40$	$40 < \delta$
Expansion bearings (δ , mm)	$\delta < 50$	$50 < \delta < 100$	$100 < \delta < 150$	$150 < \delta < 255$
Fixed dowels (δ , mm)	$8 < \delta < 100$	$100 < \delta < 150$	$150 < \delta < 255$	$255 < \delta$
Expansion dowels (δ , mm)	$\delta < 30$	$30 < \delta < 100$	$100 < \delta < 150$	$150 < \delta < 255$

Figura 3.8: Livelli di danno classificati secondo la duttilità cinematica di un ponte, in particolare la prima riga riporta quelli della pila [Choi 2002].

carsi con qualche riserva ai ponti esistenti in Italia, principalmente per le diverse tecniche di costruzione, ma data la carenza di informazioni sui danneggiamenti di ponti riscontrati in casi reali a seguito di un sisma nel territorio italiano, si è deciso di adottare i quattro valori di duttilità elencati in tabella, corrispondenti a livelli di duttilità crescenti in curvatura della sezione.

Lo schema delle pile, nei modelli che sarà possibile creare con l'interfaccia sviluppata, è di tipo *a telaio* per cui il loro schema statico sarà di *mensola incastrata alla base* se il carico agisce in direzione longitudinale, mentre sarà di *doppio incastro* se il carico agisce in direzione trasversale, data la presenza del traverso superiore. I quattro valori di danno per i relativi Performance Level sono:

- $D_1[PL1] : \mu = 1$ Danno lieve;
- $D_2[PL2] : \mu = 2$ Danno moderato;
- $D_3[PL3] : \mu = 4$ Danno esteso;
- $D_4[PL4] : \mu = 7$ Danno completo;

Il livello di danno che un certo ponte può subire per una determinata azione sismica attesa è correlato alla sua importanza strategica in condizioni di emergenza: per ponti di fondamentale importanza è ammesso solo un lieve danneggiamento, mentre in altri casi sono tollerati livelli di danno maggiori. Questa considerazione è strettamente legata all'efficacia delle curve di fragilità come strumento utile alla pianificazione delle priorità per interventi di adeguamento e rinforzo dei ponti esistenti. In nessun caso è tuttavia ammesso il collasso.

3.4.2 Analisi in Time-history: Metodo 1

Questa metodologia assume che le curve di fragilità siano delle distribuzioni di probabilità cumulata condizionata, esse esprimono la probabilità di superamento o di eguagliamento di un determinato livello di danno condizionato alla misura

di intensità del sisma, che solitamente viene espressa attraverso la PGA. Questa procedura è stata proposta in letteratura da *Monti e Nisticò, 2002*.

Il termine *fragilità*, $P_{f,PL}(a)$, è introdotto mediante la seguente espressione:

$$P_{f,PL}(a) = P[D > d_{PL}|a] \quad (3.7)$$

Dove D è la funzione di danno, che nel caso in esame viene rappresentata dalla duttilità della sezione della pila, come illustrato in precedenza. Dunque la fragilità è la probabilità condizionata all'intensità del sisma che la funzione di danno superi o eguagli un determinato PL.

La funzione di danno, è una funzione aleatoria, la cui casualità deriva, come accennato più volte, dalle incertezze sui materiali e sull'azione sismica. Dal momento che si assume la pila come elemento caratterizzante la vulnerabilità sismica dell'intero ponte: si indichi con \mathbf{G} il modello strutturale della stessa, con \mathbf{p} la funzione casuale che incorpora le incertezze sui materiali, cioè la resistenza a compressione f_c del calcestruzzo e la resistenza a snervamento f_y dell'acciaio da armatura, si indichi inoltre con \mathbf{S}_a lo spettro di risposta, che varia in funzione degli accelerogrammi considerati, si avrà, quindi, la seguente espressione per la funzione di danno:

$$D = D(\mathbf{G}(\mathbf{p}), \mathbf{S}_a|a) \quad (3.8)$$

Che, sostituita nella (3.7), permette di esplicitare la fragilità come:

$$P_{f,PL}(a) = P[D(\mathbf{G}(\mathbf{p}), \mathbf{S}_a|a) > d_{PL}] = \int_{D(a) > d_{PL}} f_D(d|a) dd \quad (3.9)$$

Ora si espone, in sintesi il procedimento necessario per determinare, dati i valori in ascissa espressi mediante l'accelerazione \mathbf{a} , il corrispondente valore in ordinata della curva di fragilità relativa al livello di danno \mathbf{d}_{PL} :

- si stimano media e varianza dei parametri casuali \mathbf{p} , la resistenza a compressione f_c del calcestruzzo e la resistenza a snervamento f_y dell'acciaio, e si scelgono opportune funzioni di densità di probabilità, pdf , per descriverne la distribuzione;
- si suddivide ogni pdf in un numero m di intervalli Δp_i^k , ogni intervallo ha un valore centrale la cui probabilità di occorrenza è uguale a: $f_{pi}(p_i^k)\Delta p_i^k$;

- si sceglie la k -esima combinazione dei parametri p^k e se ne calcola la probabilità di occorrenza mediante la relazione:

$$\prod_{i=1}^n f_{pi}(p_i^k) \Delta p_i^k \quad (3.10)$$

- si lancia l'analisi di simulazione sismica sulla struttura, secondo il modello strutturale \mathbf{G} adottato, e si calcola il danno sui punti di controllo, cioè se ne rileva la duttilità;
- una volta noti i valori di danno per la simulazione condotta, si astraggono per determinare la funzione di densità di probabilità del danno $f_D(d|a)$. Tale distribuzione è di tipo lognormale, secondo quanto riportato in letteratura, ma ha validità generale. Per rendere nota tale funzione è necessario stimare media:

$$E[D] = \sum_{k=1}^{m^n} d_k \left[\prod_{i=1}^n f_{pi}(p_i^k) \Delta p_i^k \right] \quad (3.11)$$

e varianza:

$$Var[D] = \sum_{k=1}^{m^n} (d_k - E[D])^2 \left[\prod_{i=1}^n f_{pi}(p_i^k) \Delta p_i^k \right] \quad (3.12)$$

- si calcola l'integrale $\int_{D(a) > d_{PL}} f_D(d|a) dd$ tramite un metodo numerico. L'estremo inferiore di integrazione varia a seconda del livello di danno che si sta considerando ed è pari al corrispettivo valore di μ .

Le curve di fragilità così costruite sono riferite per il meccanismo di rottura delle pile a *flessione*. Infatti si è presa in considerazione la duttilità della sezione della pila ammettendo che questa entri largamente in campo plastico con deformazioni notevoli. Diversamente, per alcune tipologie strutturali, può diventare determinante il meccanismo di rottura a *taglio* della pila. Per costruire le curve di fragilità riferite a tale comportamento è possibile operare con il medesimo procedimento, basta considerare l'indice di danno come il rapporto tra la sollecitazione tagliante che la sezione di controllo sperimenta durante la simulazione sismica, e la resistenza a taglio della stessa:

$$D = \frac{V_{soll}}{V_{res}} \quad (3.13)$$

Ovviamente, trattandosi di un meccanismo di rottura fragile, non è possibile individuare livelli di danno crescenti.

Con il metodo illustrato è possibile giungere alla costruzione di un set di curve di fragilità (quattro se il meccanismo di rottura considerato è quello a flessione, una soltanto se si considera il meccanismo di rottura a taglio) specifico per ogni pila. Ma nell'ottica di voler valutare la vulnerabilità sismica dell'intero ponte occorre combinare statisticamente i risultati ottenuti per ciascuna. Avremo, pertanto, in un ponte costituito da N pile, per un determinato Performance Level, la seguente probabilità di superamento (ipotesi di indipendenza):

$$P_{f,PL\text{ sistema}} = 1 - \prod_{pila=1}^N (1 - P_{f,PL\text{ pila}}(a)) \quad (3.14)$$

Il procedimento riportato è da effettuarsi sia per gli spostamenti in direzione trasversale, sia per quelli in direzione longitudinale. Nella modellazione del ponte, seguita anche in questa trattazione, l'impalcato viene considerato elastico semplicemente appoggiato, dal momento che si assume come elemento vulnerabile al sisma la pila. Pertanto, questa può essere schematizzata come un elemento a un singolo grado di libertà, il cui comportamento non viene influenzato dalla rigidità degli impalcati, ipotesi nella quale è valido il procedimento appena esposto.

Diversamente, se il modello strutturale accoppia la risposta delle pile al comportamento degli impalcati, in funzione delle loro rigidità, non è più corretto l'utilizzo di tale procedura. Si fa coincidere il collasso del sistema con la rottura della prima pila, dal momento che queste ultime solitamente giungono a collasso prima degli impalcati. Allora se si sta considerando un ponte continuo, dal momento che la probabilità di collasso del sistema è determinata dalla rottura della prima pila, si può definire un intervallo in cui l'intero ponte ha probabilità di collasso. L'intervallo è delimitato da un estremo superiore che rappresenta la probabilità di superamento di un certo PL computato come se il sistema fosse ad elementi indipendenti (cioè del tutto analogo al caso esposto in precedenza), e da un estremo inferiore che descrive la probabilità di superamento di un determinato livello di danno nell'ipotesi (più pessimistica) in cui tutti gli elementi della struttura fossero dipendenti gli uni dagli altri. Questa è la relazione, proposta da *Melchers, 1999*, per la valutazione della vulnerabilità di ponti con impalcati e pile

accoppiati:

$$\max[P_{f,PLpila}] \leq P_{f,PLsistema} \leq 1 - \prod_{pila=1}^N (1 - P_{f,PLpila}(a)) \quad (3.15)$$

3.4.3 Analisi in Time-history: Metodo 2

Verrà di seguito esposto un altro procedimento per la costruzione delle curve di fragilità, esso si divide nei seguenti passi:

- si calcolano, mediante modellazione numerica, i valori che rappresentano la *risposta massima della struttura al sisma*, nella seguente trattazione si considera come indice di danno la duttilità cinematica di sezioni di controllo;
- si assume che la *domanda sismica* sia descritta mediante una distribuzione logonormale, essa viene quantificata da un parametro IM *Intensity Measure*, solitamente espresso in PGA o S_a . Pertanto si può calcolare la *domanda strutturale media*, S_d mediante la seguente formula (Nielson & DesRoches, 2007):

$$S_d = e^A IM^B \quad (3.16)$$

Questa formula è rappresentabile mediante una retta nel piano bilogarithmico:

$$\ln(S_d) = A + B \ln(IM) \quad (3.17)$$

- Si calcolano i coefficienti A e B mediante regressione lineare sull'intero set di dati rappresentati nel piano $\ln(IM)-\ln(S_d)$, cioè si calcola la deviazione standard dei valori della domanda rispetto al valore medio λ dato dalla retta di regressione per un determinato IM (Cornell et al., 2002), graficamente la retta di regressione lineare presenta l'andamento riportato in Figura 3.9.
- Una volta determinati i coefficienti di regressione lineare A e B è possibile tracciare la curva di fragilità per ciascun PL come distribuzione cumulata logonormale, dove la probabilità di superamento di ciascun PL data l'azione sismica è:

$$P_{f,PL}(a) = P[D > d_{PL}|a] \quad (3.18)$$

e si calcola integrando numericamente:

$$\int_{D(a) > d_{PL}} f_D(d|a) dd \quad (3.19)$$

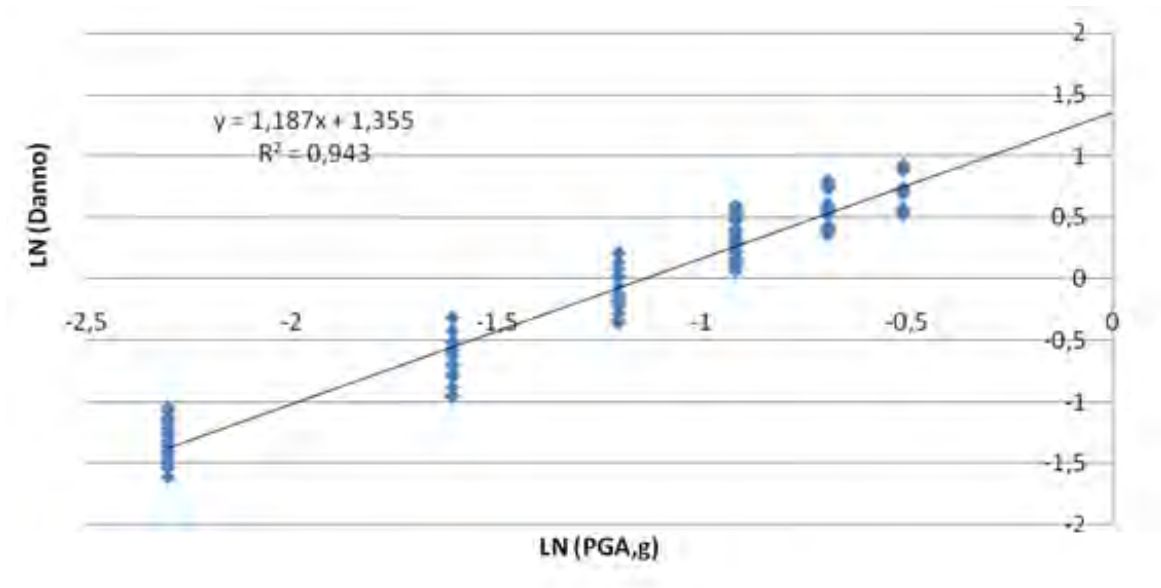


Figura 3.9: Esempio di retta di regressione nel piano $\ln(\text{PGA})$ vs. $\ln(\text{Danno})$

Dove l'argomento dell'integrale, cioè la funzione densità di probabilità del danno, che come già detto si assume descrivibile con una distribuzione logonormale, è data dalla seguente legge:

$$f_D(d) = \frac{1}{\sqrt{2\pi}\epsilon d} \exp\left[-\frac{1}{2}\left(\frac{\ln(d) - \lambda}{\epsilon}\right)^2\right] \quad (3.20)$$

dove si è indicato con $\lambda = A + B\ln(IM)$ il valore medio atteso calcolato sulla retta di regressione dato un certo IM (PGA), mentre ϵ è la deviazione calcolata sull'intero set di dati rappresentato nel piano bilogarithmico rispetto al valore medio dato dalla retta di regressione.

- La curva di fragilità così costruita rappresenta la vulnerabilità della singola pila $P_{f,PL,pila}$, considerando un ponte formato da N pile, occorre combinare statisticamente le probabilità delle singole pile, attraverso la medesima relazione del metodo 1:

$$\max[P_{f,PL,pila}] \leq P_{f,PL,sistema} \leq 1 - \prod_{pila=1}^N (1 - P_{f,PL,pila}(a)) \quad (3.21)$$

Si riporta un esempio di curve di fragilità, ottenute con un campione di dati che deriva da analisi dinamiche non lineari in Time-history per 6 valori di PGA (0.1g; 0.2g; 0.3g; 0.4g; 0.5g; 0.6g). La maggiore attendibilità del metodo 2 rispetto al

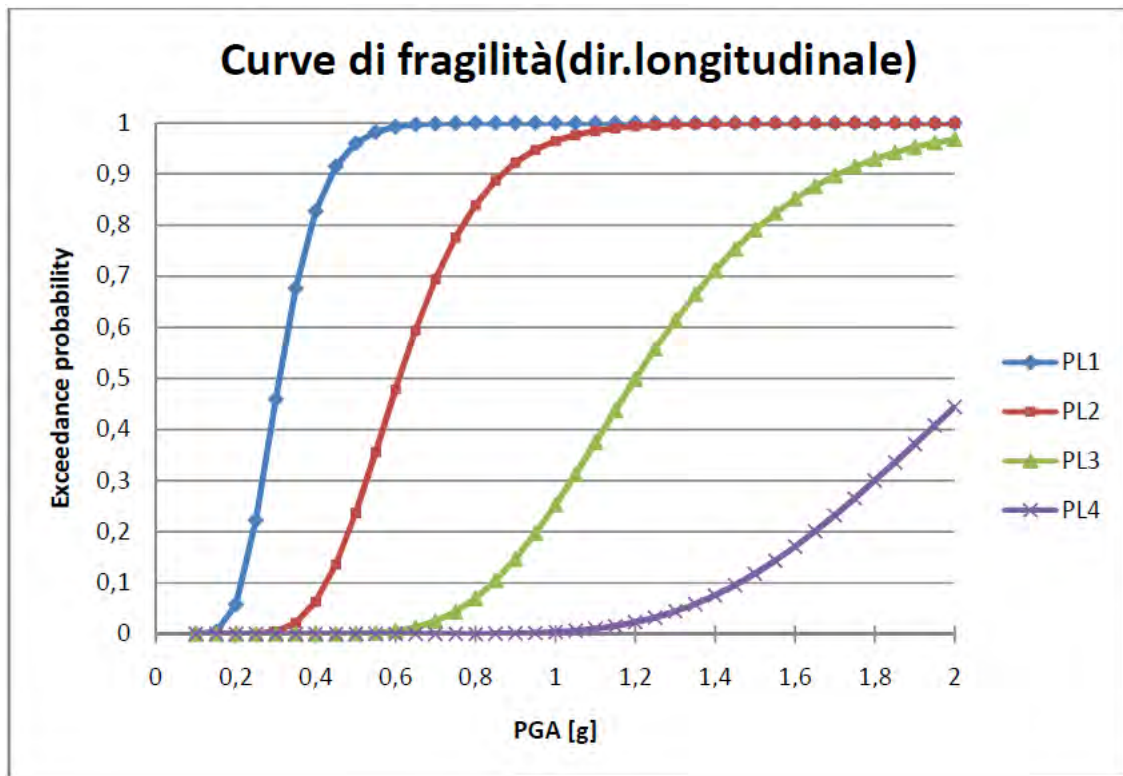


Figura 3.10: Esempio di curve di fragilità costruite con il metodo due partendo da analisi dinamiche non lineari su 6 valori di PGA

metodo 1 risiede nel fatto che è sufficiente un numero limitato di analisi a diversa PGA (nella fattispecie 6) per poter astrarre i risultati in generale ed avere curve continue che forniscono il valore della probabilità di superamento per ciascun livello di danno per un qualsiasi valore di accelerazione in ascissa.

3.4.4 La distribuzione Normale e Log-normale

Più volte si è fatto riferimento, in precedenza, alle distribuzioni di probabilità di tipo logonormale: si è assunta avere tale distribuzione la funzione di domanda sismica, S_d e le proprietà dei parametri p che definiscono la risposta sismica. Nel paragrafo seguente verranno brevemente esposte le caratteristiche della distribuzione di probabilità normale e logonormale.

Una variabile aleatoria viene definita come una variabile che non assume un valore costante, deterministico, bensì può assumere valori diversi, all'interno di un intervallo più o meno esteso, che prende il nome di spazio campionario, in dipendenza di fenomeni casuali. Ad ogni variabile aleatoria X si associa una distribuzione casuale, o funzione di probabilità, $P(X)$, che associa la probabilità che X assuma valori in un determinato intervallo del suo spazio campionario. Se

la variabile aleatoria assume valori reali la distribuzione di probabilità prende il nome di *funzione di ripartizione*, ed è definita come:

$$F(x) = P(X \leq x) \quad (3.22)$$

Se invece la variabile X può assumere valori discreti, cioè può valere solamente delle quantità multiple di un certo valore, si definisce la *funzione di probabilità discreta* come:

$$p(x) = P(X = x) \quad (3.23)$$

Se si sta considerando una variabile aleatoria continua, è possibile definire una funzione f , non negativa detta *funzione densità di probabilità*. La funzione di densità rappresenta quanto vale la probabilità nell'intorno di un determinato valore di X in rapporto all'ampiezza di tale intorno. Sia X una variabile aleatoria continua, sia A un intervallo all'interno del suo spazio campionario, la funzione densità di probabilità è definita come:

$$P(X \in A) = \int_A f(x) dx \quad (3.24)$$

Esistono varie forme di distribuzioni casuali, la più importante è senza dubbio quella *gaussiana* o *normale*, con la classica forma a campana simmetrica centrata sul valore medio, perchè ben si presta a rappresentare l'andamento di molte variabili aleatorie e perchè è alla base dell'inferenza statistica in virtù del teorema del limite centrale. La distribuzione normale è caratterizzata dalla seguente funzione di densità di probabilità:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2} \left(\frac{x - \mu}{\sigma}\right)^2\right) \quad (3.25)$$

Dove μ è il valore medio o *valore atteso* e σ^2 è la *varianza*. La variabile x può assumere valori: $-\infty < x < \infty$ come già anticipato la forma della curva è a tipo campanulare, fissato un valore x in ascissa l'area sottesa tra la curva e l'asse delle ascisse fino al valore fissato rappresenta la probabilità che la variabile aleatoria X possa assumere valori compresi tra meno infinito e x .

A partire dalla distribuzione normale si può ricavare la logonormale, una variabile aleatoria X si dice distribuita secondo la logonormale se la variabile $\ln(X)$ segue una distribuzione normale. La funzione densità di probabilità di una

logonormale segue la seguente legge:

$$f(x) = \frac{1}{x \sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2} \left(\frac{\ln(x) - \mu}{\sigma}\right)^2\right) \quad (3.26)$$

La forma di tale curva è campanulare asimmetrica, con il massimo della curva posizionato sopra il valore atteso, che è dato dalla seguente relazione:

$$E(X) = e^{\mu + \frac{\sigma^2}{2}} \quad (3.27)$$

mentre la varianza si calcola come:

$$Var(X) = e^{2\mu + \sigma^2} (e^{\sigma^2} - 1) \quad (3.28)$$

I parametri calcolati mediante le relazioni (3.26) e (3.27) sono riferiti alla variabile aleatoria $W = \ln(X)$, dove W segue una distribuzione logonormale, X una distribuzione normale. Qualora si disponesse della media e della varianza della variabile aleatoria normale, X , e si volesse rappresentare la funzione logonormale, si rende necessario risolvere un sistema lineare a due equazioni e a due incognite per determinare i parametri, $E(W)$ $Var(W)$, della variabile aleatoria logonormale, $W = \ln(X)$:

$$\mu = \ln(E(X)) - \frac{1}{2}\sigma^2 \quad (3.29)$$

$$\sigma^2 = \ln\left[\frac{Var(X)}{E(X)^2} + 1\right] \quad (3.30)$$

In conclusione, si riportano di seguito i diagrammi delle funzioni densità di probabilità utilizzati per descrivere l'incertezza a riguardo delle performance meccaniche dei materiali, cioè la resistenza a compressione del calcestruzzo, f_c , e la resistenza a snervamento dell'acciaio, f_y . Come si evince dalla letteratura in merito, la distribuzione probabilistica di tipo normale si adatta bene alla resistenza del calcestruzzo (Melchers, 1999), mentre la resistenza a snervamento dell'acciaio è ben descritta da una distribuzione probabilistica lognormale (Mirza e MacGregor, 1979, Melchers, 1979). Le curve della f_y sono state generate con uno script creato in Matlab, tale interfaccia permette di implementare la distribuzione logonormale in forma cumulata e come densità di probabilità. Mentre le curve della resistenza del calcestruzzo sono state tracciate mediante dei grafici in Excel. All'interno dell'interfaccia per la creazione del modello del ponte l'utente potrà scegliere tra diverse classi di materiale da costruzione, per ciascuna classe è stato effettuato

uno studio per determinarne la corretta distribuzione della resistenza, ma questo argomento verrà trattato più approfonditamente nel paragrafo apposito.

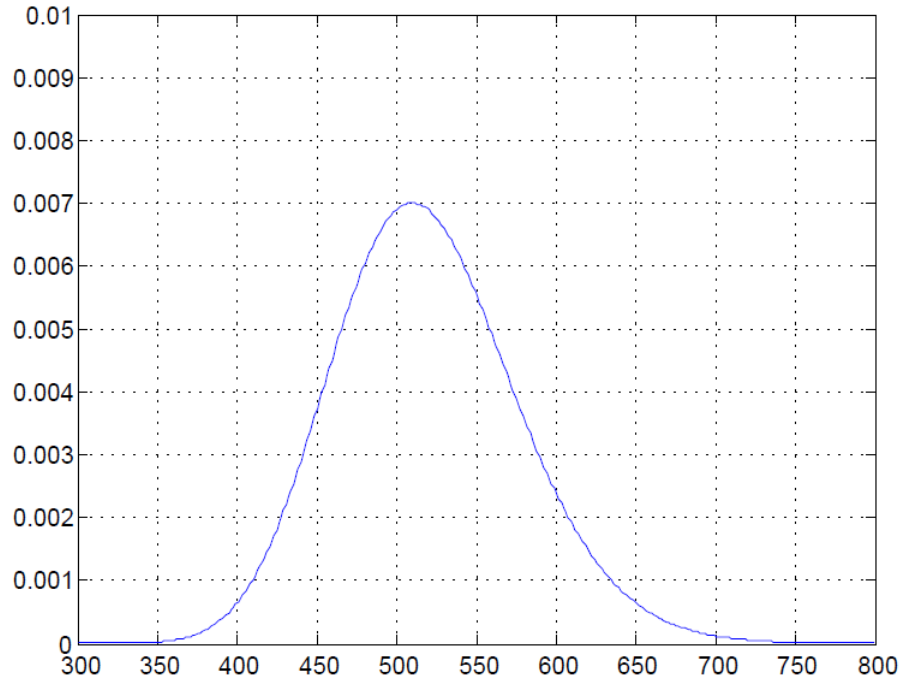


Figura 3.11: Distribuzione di probabilità lognormale per l'acciaio d'armatura FeB44k

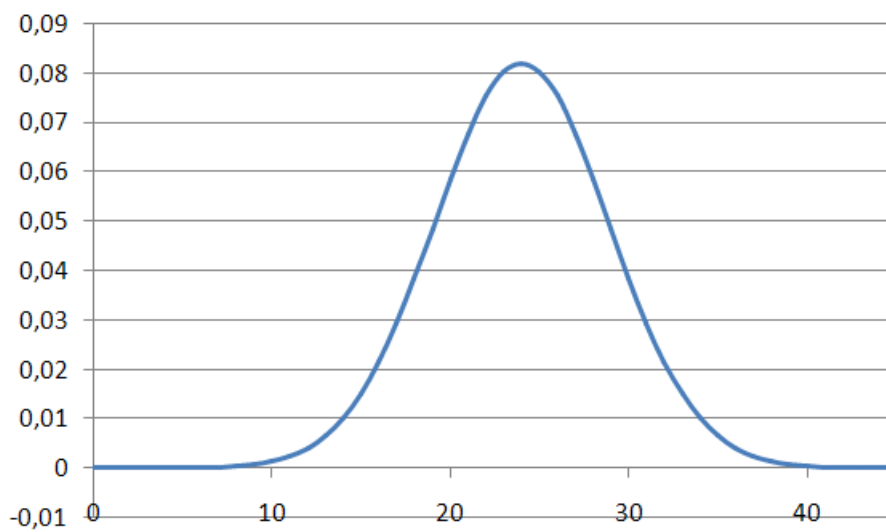


Figura 3.12: Distribuzione di probabilità normale per il calcestruzzo in classe C16/20

Capitolo 4

Procedura di automatizzazione per la valutazione della vulnerabilità sismica di un ponte

4.1 PROGRAMMARE CON MATLAB

Nell'ambito della seguente trattazione si è deciso di avvalersi del programma *Matlab* per la costruzione dell'interfaccia grafica per l'utente e per lo sviluppo del codice automatico ad essa associato. Tale scelta è giustificata dal fatto che Matlab rappresenta un software molto potente utilizzabile a più livelli, a partire da operazioni molto banali fino ad analisi di sistemi complessi. Il nome Matlab deriva da un'abbreviazione del termine *Matrix-Laboratory*, dal momento che la struttura base è la **matrice**: ogni variabile viene trattata come una matrice di dimensioni $m \times n$, ovviamente un scalare sarà una matrice di dimensioni 1×1 . Passiamo, in seguito, ad analizzare brevemente quali sono le caratteristiche salienti di questo linguaggio di programmazione:

- la caratteristica di maggior pregio è che si tratta di un linguaggio molto rapido (infatti è un linguaggio di altissimo livello) dal momento che Matlab è un interprete e non un compilatore, tuttavia questo potrebbe determinare una qualche lentezza in fase di esecuzione;
- si ha una notevole semplificazione all'atto della dichiarazione delle variabili, dal momento che non è necessario specificare nè la dimensione nè il tipo di valori che essa può assumere (numeri interi, reali, complessi, stringhe di testo,...),

- all'interno del software si trovano già predefinite una notevole serie di funzioni primitive, dette **built-in-functions**, che sono molto utili perchè permettono la risoluzione immediata di problemi complessi, ad esempio la ricerca degli zeri di una funzione, la risoluzione di sistemi lineari;
- crea file (script, function, GUI) che sono eseguibili esclusivamente attraverso Matlab stesso.
- è molto facile reperire manuali circa il funzionamento di aspetti specifici: oltre alla stretta programmazione anche la grafica o le toolboxes;

4.2 L'INTERFACCIA GRAFICA DI MATLAB, GUI (Graphical User Interface)

L'oggetto di questo lavoro di tesi, come più volte accennato, è quello di sviluppare un codice in grado di realizzare un modello strutturale di un ponte di tipologia ricorrente, lanciare un'analisi dinamica non lineare e infine di analizzare i risultati ottenuti in modo da fornire in output le curve di fragilità. Per rendere l'intero programma facilmente fruibile agli utenti è stato collegato al codice una interfaccia grafica, chiamata in ambiente Matlab *GUI, Graphical User Interface*.

L'interfaccia grafica è stata costruita mediante l'ambiente di progettazione di Matlab, il *GUIDE, Graphical User Interface Development Environment*, che è un editor visuale per la costruzione delle GUI.

Sul pannello di sinistra si possono notare tutti gli elementi interattivi che possono essere inclusi nella GUI e collegati al codice. Per realizzare l'interfaccia è sufficiente trascinare gli oggetti desiderati all'interno dell'area di progettazione e definirne dimensione, posizione e proprietà. In seguito, ad ogni elemento, sia che la sua funzione sia quella di ricevere dati in input dall'utente o stampare dati in output a schermo, va assegnato un opportuno *Tag*, in modo da poterlo includere nel codice dello script.

Di seguito si descrivono in particolare gli oggetti interattivi utilizzati nell'ambito della creazione dell'interfaccia per la costruzione del modello tridimensionale del ponte:

- **Push Button**, si tratta di un pulsante collegato con diverse parti del codice, che entra in funzione solo se premuto dall'utente;
- **Static Text**, è una finestra di testo statico, cioè che viene definito a priori, in fase di programmazione, e che non viene mutato durante l'esecuzione.

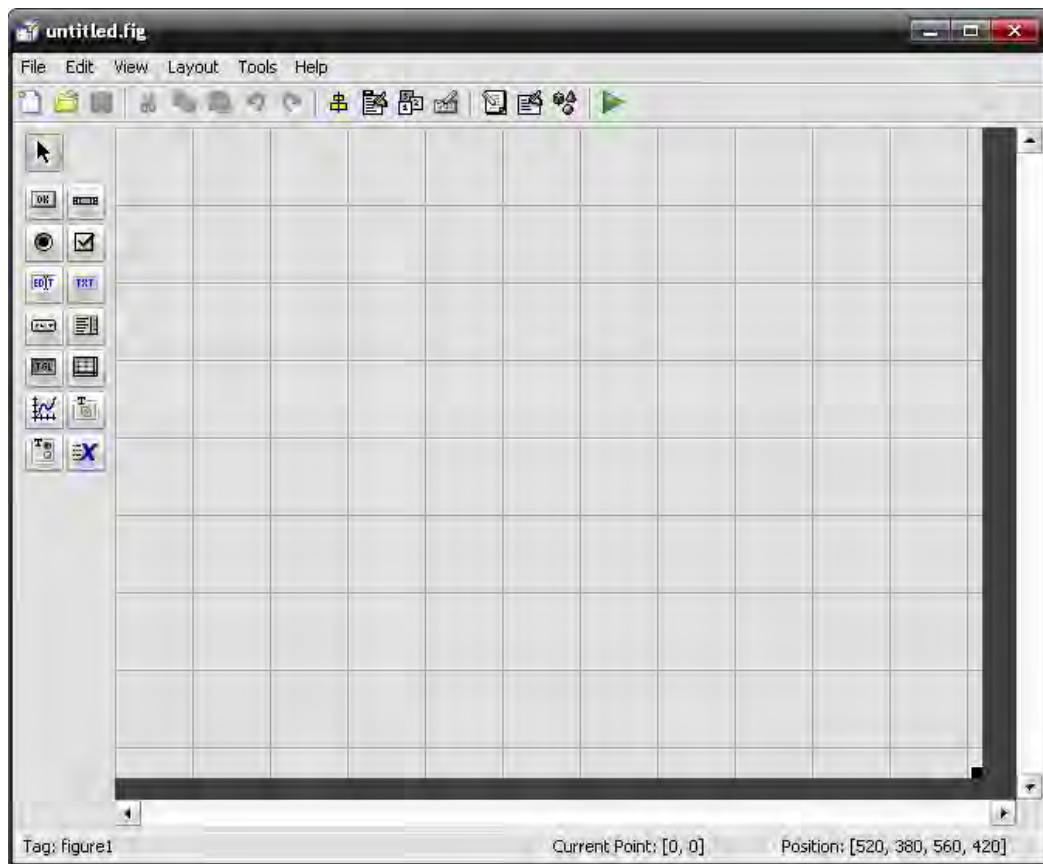


Figura 4.1: Ambiente visuale per l'editing delle interfacce grafiche di Matlab

ne dell'interfaccia, serve ad esempio per fornire istruzioni all'utente o per denominare campi per l'input di dati;

- **Edit Text**, è un campo per l'inserimento di dati, all'avvio della GUI contiene un testo assegnato di default, che può essere un indicazione sul contenuto o un valore standard, poi l'utente può modificarlo inserendo valori in input. Una precisazione sulla funzionalità dell'Edit Text: i numeri ricevuti in ingresso vengono letti dal codice come stringhe di testo, per cui si rende necessario richiamare una *function* che li converte in variabili di tipo numerico.
- **Axis**, questo oggetto permette di tracciare un'area all'interno dell'interfaccia nella quale sarà possibile tracciare dei grafici bidimensionali o tridimensionali, all'interno del codice si utilizzano opportuni comandi per il plot dei grafici che permettono di definire lo stile del diagramma.

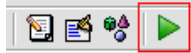
Una volta inseriti e disposti correttamente tutti gli oggetti necessari, possiamo salvare il progetto. Verranno così creati due file con lo stesso nome, uno avrà

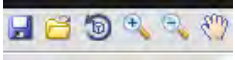
estensione *.fig* e contiene l'interfaccia vera e propria, l'alto avrà estensione *.m* e sarà proprio il codice di controllo per la gestione delle funzionalità dell'interfaccia. Vediamo, senza scendere nel dettaglio, da quali macro-sequenze di istruzioni è composto tale script:

- la *OpeningFcn* che contiene il codice per l'inizializzazione dell'interfaccia, è sempre generata in automatico all'atto del salvataggio e non va assolutamente modificata, (contiene per altro dei commenti con l'indicazione do not edit). A seguito della *OpeningFcn* si possono, se necessario inizializzare le variabili con dei valori standard;
- la *OutputFcn* che contiene le istruzioni necessarie per comunicare con la GUI al fine di restituire i valori in output, anche questa macro-sequenza è posta in testa al codice, viene automaticamente compilata e non va modificata;
- le funzioni *CallBack* sono delle *function* specifiche che vengono richiamate per ogni oggetto presente nell'interfaccia (push button, edit text,...). Il programmatore può inserire all'interno delle callback blocchi di istruzioni che vuole siano eseguite quando l'utente va ad interagire con il particolare oggetto presente nella GUI.

4.3 COSTRUZIONE DEL MODELLO TRIDIMENSIONALE

Nel seguente paragrafo sarà illustrata nello specifico l'interfaccia appositamente creata per la costruzione del modello tridimensionale del ponte.

Si deve lanciare la GUI premendo il bottone *run figure*  dall'ambiente di progettazione. Così facendo viene lanciato il codice sorgente legato all'interfaccia che restituisce a schermo una finestra come riportato in Figura 4.2.

Nella porzione in alto a sinistra possiamo trovare la barra del menù con oggetti a pulsante.  A partire da sinistra troviamo, in ordine, il pulsante

salva, che permette all'utente di salvare in ogni momento il progetto a cui si sta lavorando e i dati immessi in input, il tutto può essere successivamente richiamato mediante il pulsante apri. Gli altri quattro pulsanti servono per modificare la visualizzazione tridimensionale del modello una volta che lo si è creato, il bottone *rotate 3D* permette un pan tridimensionale della finestra, i restanti servono ad aumentare o diminuire lo zoom e a traslare il modello nel piano.

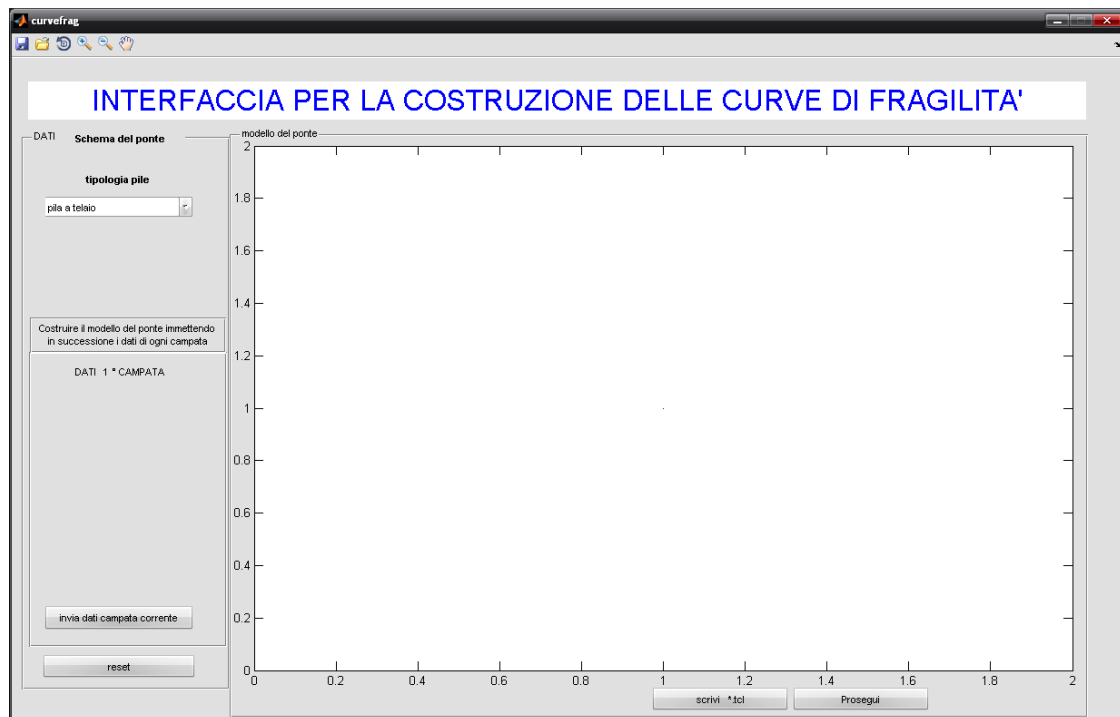


Figura 4.2: L'interfaccia grafica per la costruzione del modello tridimensionale del ponte

Il pannello a sinistra permette l'immissione, da parte dell'utente, dei dati geometrici riguardanti l'impalcato e le pile, mentre lo spazio a destra è l'ambiente dove verrà visualizzata la rappresentazione grafica del ponte.

4.3.1 Dalla geometria del ponte al modello numerico

Prima di analizzare nello specifico il procedimento per l'immissione dei dati e la costruzione campata per campate del ponte è necessario specificare lo schema statico adottato per il modello. Nell'ottica di voler creare un procedimento che si adattasse alle più ricorrenti tipologie di ponti e viadotti esistenti sul territorio nazionale si è scelto di dare all'utente la possibilità di generare un ponte di tipo rettilineo, ogni campata potrà essere diversa dalle altre per lunghezza e numero di traversi presenti al suo interno. In corrispondenza di ogni pila l'utente potrà decidere l'altezza del fusto delle stesse nonché del loro traverso superiore, il loro schema è del tipo pile a telaio.

Il modello numerico che verrà di conseguenza generato avrà le seguenti caratteristiche:

- ogni impalcato è considerato semplicemente appoggiato sia sulle spalle che sul traverso in testa ad ogni pila, le campate verranno modellate mediante

elementi elastici lineari, durante l'analisi sismica questi elementi restano in campo elastico e il loro comportamento non è determinante per lo studio della vulnerabilità sismica dell'intera struttura;

- ogni pila viene rappresentata mediante uno schema statico a portale incastrato alla base, pertanto la risposta nei confronti dell'azione orizzontale sarà diversa a seconda della direzione: la pila si comporta come una semplice mensola con incastro alla base se l'azione del sisma è diretta in senso longitudinale, mentre se l'azione è diretta trasversalmente si avrà un comportamento a telaio dal momento che il traverso superiore solitamente ha un'inerzia maggiore rispetto alle pile e che, quindi, impone un vincolo di incastro in sommità alle stesse. Le pile e il traverso vengono modellati attraverso elementi a fibre e ogni portale ha una massa di pertinenza pari all'intera campata, che viene rappresentata come un carico distribuito in sommità al traverso superiore della pila.
- il traverso superiore è modellato utilizzando elementi a fibre, la sua altezza viene computata come offset tra il livello dell'impalcato e la testa della pila mediante l'utilizzo di elementi *rigid link*;
- il vincolo di appoggio semplice mediante cerniere tra l'impalcato e il traverso superiore delle pile e tra impalcato e spalle del ponte viene rappresentato mediante elementi *zerolength* che sbloccano le rotazioni secondo opportune direzioni;

L'input dei dati dell'interfaccia è organizzato in modo tale che l'utilizzatore deve fornire dati geometrici campata per campata, in successione una dopo l'altra fino ad ottenere il risultato finito, questa scelta è motivata dal fatto che si voleva lasciare libero l'utente di creare il ponte con campate e pile diverse le une dalle altre, in modo da estendere l'adattabilità di questo preprocessore al più grande numero di casi possibili

Passiamo, quindi, ad analizzare in quale ordine l'utente dovrà inserire le caratteristiche dimensionali del ponte per costruire un modello fedele alla struttura esistente. Cominciamo con l'osservare che nel pannello d'input dei dati è presente un menù a tendina nel quale può essere scelto lo schema strutturale delle pile tra: **pile a telaio** e **pile a fusto unico**, nell'ambito del seguente lavoro di tesi si è deciso di approfondire solamente la prima tipologia, ma il codice si presta bene ad eventuali ulteriori estensioni che potrebbero inglobare ponti con pile a fusto unico od altri sistemi.

The figure shows two sequential screenshots of a software interface titled "DATI Schema del ponte".

Left Screenshot:

- Section: **tipologia pile**
- Dropdown menu: "pila a telaio" (selected), "pila a telaio", "pila a fusto unico"
- Text: "Costruire il modello del ponte immettendo in successione i dati di ogni campata"
- Section: **DATI 1 ° CAMPATA**
- Buttons: "invia dati campata corrente", "reset"

Right Screenshot:

- Section: **tipologia pile**
- Dropdown menu: "pila a telaio" (selected)
- Section: **larghezza impalcato**
- Field: **Ld=** 6 m
- Section: **numero di travi nell'impalcato**
- Field: **nb=** 3
- Text: "Costruire il modello del ponte immettendo in successione i dati di ogni campata"
- Section: **DATI 1 ° CAMPATA**
- Section: **lunghezza campata**
- Field: **Ls=** 10 m
- Section: **numero traversi per impalcato**
- Field: **nt=** 3 m
- Buttons: "invia dati campata corrente", "reset"

Figura 4.3: Campi per l'inserimento dei dati geometrici del ponte

Una volta selezionato lo schema di pile a telaio dunque, appaiono nel medesimo pannello ulteriori campi:

- **larghezza impalcato, Ld:** si dovrà inserire un numero che rappresenta la larghezza dell'impalcato, espressa in metri, che si ritiene costante per tutto lo sviluppo del ponte;
- **numero travi nell'impalcato, nb** in questo box va specificato il numero di travi longitudinali presenti nella struttura, si assume che tale numero non cambi per tutta la lunghezza del ponte;
- **lunghezza campata, Ls** in questo campo va inserita la lunghezza della campata corrente, ogni campata può avere lunghezza diversa dalle altre e può essere discretizzata in modo diverso;

- **numero traversi per impalcato, nt**: si dovrà specificare il numero di traversi presenti nella campata corrente, ogni campata potrà avere un numero di traversi diverso rispetto alle altre.

In seguito, l'utente, deve premere il pulsante invia dati campata ottenendo la stampa a schermo, nell'ambiente *axis*, del primo impalcato, inoltre, il pannello per l'inserimento dati si aggiorna per la campata successiva e compaiono anche i box per l'inserimento delle caratteristiche delle pile: bisognerà specificare l'altezza del fusto, (**Hp**) e del traverso superiore, (**Ht**). Mano a mano che l'utilizzatore

DATI **Schema del ponte**

tipologia pile

pila a telaio

larghezza impalcato

Ld= 6 m

numero di travi nell'impalcato

nb= 3

Costruire il modello del ponte immettendo in successione i dati di ogni campata

DATI 2 ° CAMPATA

lunghezza campata

Ls= 10 m

numero traversi per impalcato

nt= 2 m

altezza delle pile

Hp= 9 m

altezza traverso

Ht= 1 m

invia dati campata corrente

reset

Figura 4.4: Campi per l'inserimento dei dati geometrici delle campate successive alla prima

inserisce i dati relativi a ciascuna campata in successione, nell'ambiente *axis* il ponte si assembla contestualmente, fino ad ottenere un modello tridimensionale. Un esempio viene riportato in Figura 4.5, la visualizzazione del modello così

generato può essere modificata a piacere tramite i comandi di rotazione 3D e zoom presenti nel menù dell'interfaccia.

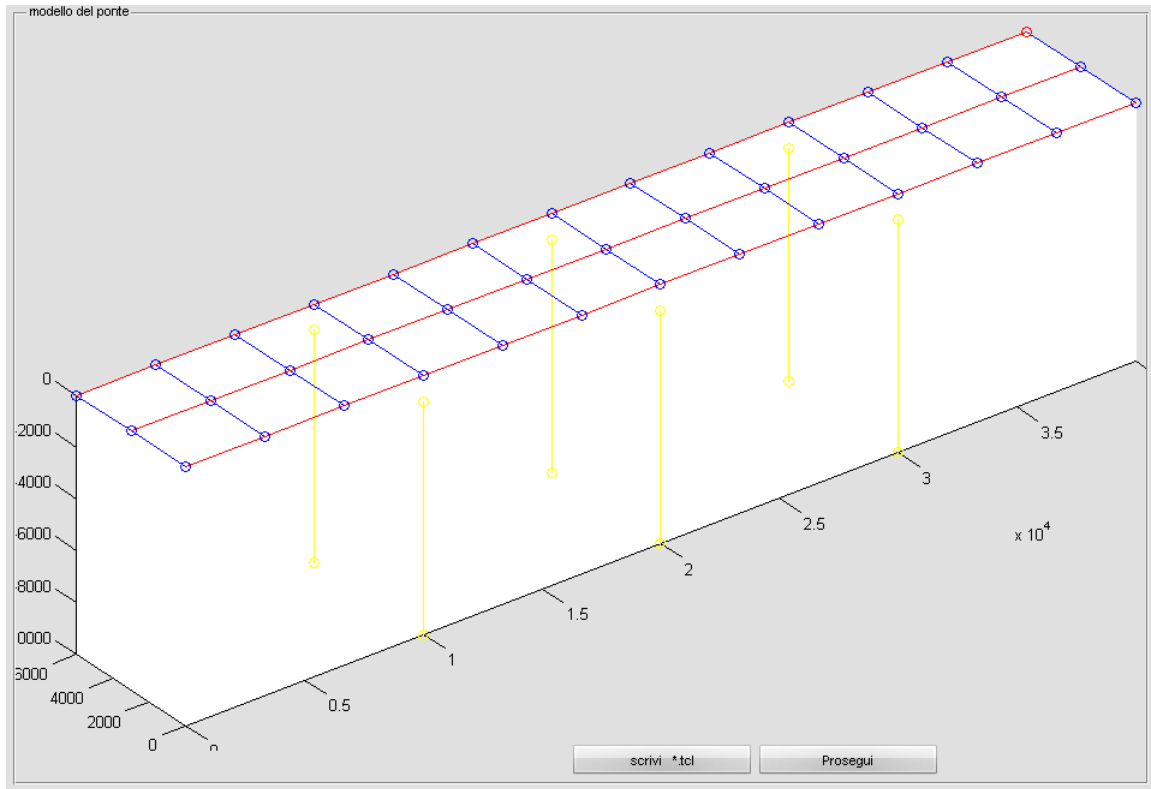


Figura 4.5: Modello tridimensionale del ponte generato dall'utente

Di seguito verrà trattato l'aspetto di come sia possibile passare da semplici valori geometrici immessi in input dall'utente al modello tridimensionale finito. Ciò è reso possibile da due script: il primo è il codice direttamente collegato all'interfaccia grafica, denominato *curvefrag.m*, che richiama successivamente un secondo script denominato *calcola.m*. Commentiamo ora nel dettaglio delle porzioni di questi due codici per capire come lavorano con riferimento alla funzionalità di tracciamento del modello 3D

- lo script *curvefrag.m* serve essenzialmente ad organizzare l'interfaccia dal punto di vista grafico. Ogni oggetto interattivo presente nella GUI viene salvato mediante una *structure* denominata di default *handles*, queste structure sono un tipo di dato che contiene al suo interno numerosi elementi individuali detti *campi*. Quindi ad esempio, la variabile che contiene la larghezza dell'impalcato verrà memorizzata con la seguente sintassi: *handles.edit_Ld* e così via anche per il resto dei dati. Il compito principale di questo script è sostanzialmente quello di rendere visibili o meno i campi per l'inserimento

dati nell'interfaccia quando opportuno, questo obiettivo viene reso possibile attraverso i comandi: `set(handles.edit_Ld,'Visible','off')`, e di reimpostarne i valori a quantità di default una volta inizializzata la GUI o qualora l'utente dovesse premere il pulsante di reset. Per settare un valore nullo, ad esempio nel campo della larghezza dell'impalcato, occorre fornire il seguente comando: `set(handles.edit_Ld,'String','')`. A questo punto il codice principale richiama lo script `calcola.m` che si occuperà dell'elaborazione dei valori in input, ad esso verrà passata la structure `handles` che contiene tutti i dati immessi. Si riporta in Figura 4.6, una porzione di testo del programma nell'editor di matlab che esegue i comandi sopra illustrati.

```

1
2
3     % --- Executes on button press in pushbutton_reset.
4     function pushbutton_reset_Callback(~, ~, handles)
5     % hObject      handle to pushbutton_reset (see GCBO)
6     % eventdata    reserved - to be defined in a future version of MATLAB
7     % handles      structure with handles and user data (see GUIDATA)
8
9     set(handles.edit_Ls,'String','0');
10    set(handles.edit_Ld,'String','0');
11    set(handles.edit_nb,'String','0');
12
13    set(handles.edit_Hp,'Visible','off');
14    set(handles.edit_Ht,'Visible','off');
15
16    set(handles.schemapile,'Value',1);
17
18    % --- Executes on button press in edit_invia.
19    function edit_invia_Callback(~, ~, handles)
20    % hObject      handle to edit_invia (see GCBO)
21    % eventdata    reserved - to be defined in a future version of MATLAB
22    % handles      structure with handles and user data (see GUIDATA)
23
24    [handles]=calcola(handles);

```

Figura 4.6: Porzione di codice per la gestione dell'interfaccia grafica

- Successivamente entra in funzione lo script `calcola.m` che recepisce in ingresso i dati contenuti nella structure `handles`, le principali funzioni di questo codice sono dapprima effettuare un controllo sul valore delle quantità in input, stampando degli avvisi nell'interfaccia qualora l'utente abbia inserito valori nulli o privi di significato, per qualche valore geometrico del ponte. Effettuato questo primo check, il programma passa alla costruzione vera e propria del modello: questo avviene mediante la creazione di tre terne di matrici di punti, X_t , Y_t e Z_t è la terna di matrici che individua i punti degli elementi longitudinali dell'impalcato, X_t , Y_t e Z_t è la terna di matrici che

individua i punti degli elementi trasversali mentre X_p , Y_p e Z_p è la terna di matrici delle coordinate dei punti delle pile.

Tali matrici vengono aggiornate per ogni campata successiva, tutti i punti dell'impalcato hanno coordinata lungo l'asse delle z nulla, mentre i punti che individuano le pile hanno coordinata z negativa, questa scelta si è rivelata necessaria dal momento che si è voluto dare l'opportunità all'utente di realizzare un modello di ponte con pile ad altezza differente tra loro. Nella parte finale dello script è presente il comando per la stampa e schermo delle tre terne di matrici, i comandi aggiuntivi `'-ro'`, `'-bo'` `'-yo'` servono, rispettivamente, per specificare il colore e lo stile dei punti, mentre il comando `axis equal` è necessario per avere la stessa scala sui tre assi.

```

255 %----plotto le tre terne di matrici di punti-----
256 - plot3 (xl, yl, zl, '-ro', xt, yt, zt, '-bo', xp, yp, zp, '-yo' );
257 - axis equal;
258

```

Figura 4.7: Estratto del codice per la stampa a schermo del modello

4.3.2 Geometria degli elementi strutturali

Una volta determinata la geometria complessiva del viadotto occorre specificare le caratteristiche dei singoli elementi strutturali. Per fare ciò l'utente dovrà premere il pulsante *proseguì* posto in basso a destra nella GUI, in questa maniera viene richiamata una function che genera a schermo un'ulteriore interfaccia di dialogo, che permette all'utente di assegnare valori specifici salvandoli in una *structure* apposita, (GUI-dialog *settingsdlg()*, Rody Oldenhuis, 2010).

Vediamo, nel dettaglio, i singoli campi di input presenti in tale ambiente. In primo luogo vanno specificate le caratteristiche geometriche e meccaniche degli elementi strutturali presenti nell'impalcato, cioè traversi e travi longitudinali, l'utente dovrà inserire il valore del modulo di elasticità E , e il modulo di elasticità tangenziale, G , del calcestruzzo, nonché l'area delle rispettive sezioni e il momento d'inerzia rispetto alle direzioni principali della sezione. L'utente dovrà porre attenzione alle unità di misura dei dati, in particolare, il valore dei momenti d'inerzia è inteso espresso in notazione scientifica e andranno inserite nell'apposito campo solamente le cifre significative, che poi verranno automaticamente moltiplicate per 10^{10} dal programma stesso.

The dialog box is titled "dati geometrici elementi strutturali" and contains several sections for data entry:

- Dati degli elementi strutturali della campata**
 - modulo di elasticità E calcestruzzo [N/mm²]: 0
 - modulo di elasticità tangenziale G calcestruzzo [N/mm²]: 0
- traverso in campata**
 - area sezione [mm²]: 0
 - Ix [*10¹⁰][mm⁴]: 0
 - Iy [*10¹⁰][mm⁴]: 0
- trave longitudinale in campata**
 - area sezione [mm²]: 0
 - Ix [*10¹⁰][mm⁴]: 0
 - Iy [*10¹⁰][mm⁴]: 0
- Dati geometrici del traverso superiore delle pile [mm]**
 - Lb: 0
 - La: 0
 - Hb: 0
 - Ha: 0
- Dati armatura del traverso superiore delle pile [mm]**
 - numero barre armatura superiore: 0
 - diametro barre armatura superiore: 0
 - numero barre armatura inferiore: 0
 - diametro barre armatura inferiore: 0
 - numero barre armatura minima: 0
 - diametro barre armatura minima: 0
 - copriferro: 0
- Dati geometrici delle pile [mm]**
 - diametro esterno pila: 0
 - numero barre armatura longitudinale: 0
 - diametro barre armatura: 0
 - copriferro: 0
- Caratteristiche meccaniche dei materiali**
 - Classe di calcestruzzo: C16/20
 - Classe di acciaio: FeB32k fy=31

Buttons: Cancel, OK

Figura 4.8: Interfaccia di dialogo per l'inserimento delle caratteristiche geometriche e meccaniche degli elementi strutturali del ponte

Il secondo blocco di campi fa riferimento alle caratteristiche del traverso superiore delle pile, per facilitare la comprensione dei vari termini da parte dell'utente

è stato disposto un'ulteriore script, che parte in automatico all'atto dell'apertura del *settingsdlg()* e che genera a schermo una figura che chiarisce le quote del traverso delle pile. La geometria di quest'ultimo è organizzata in maniera tale da dare la possibilità all'utente di creare una trave di forma rettangolare o a T.

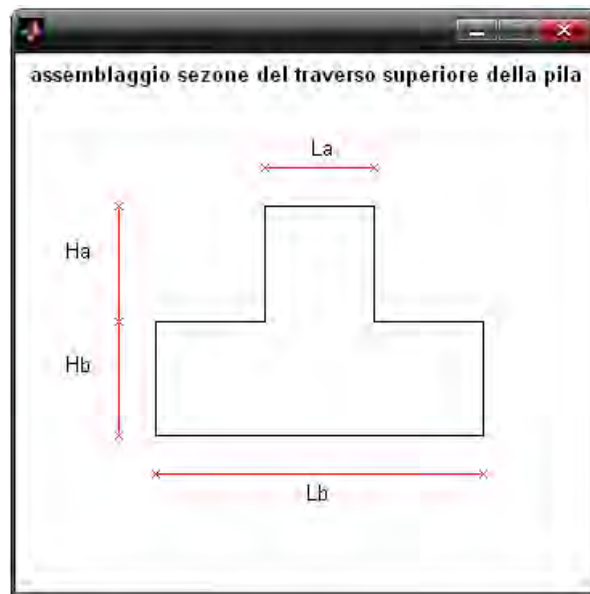


Figura 4.9: Figura che riporta le quote geometriche del traverso superiore delle pile

Oltre a valori strettamente dimensionali, è necessario specificare anche il numero di barre e il loro diametro; le barre verranno poste ad una distanza dal bordo pari al copriferro, con *armatura minima* si intendono i ferri posti lateralmente nella sezione della trave.

Il penultimo blocco di termini riguarda la pila, l'utente dovrà dichiararne diametro, numero di barre di armatura e il loro diametro. Si ritiene il caso di specificare che le barre si intendono poste radialmente nella sezione e ad una distanza dal bordo esterno pari al copriferro.

Infine, troviamo due menù a tendina che permettono di scegliere le proprietà dei materiali: la resistenza a compressione del calcestruzzo f_c e la resistenza a snervamento dell'acciaio, f_y . È stato deciso di fornire all'utente una scelta di cinque classi di calcestruzzo, e tre di acciaio, considerando le classi di resistenza dei materiali utilizzati con maggior frequenza nella costruzione di ponti e viadotti nel recente passato, dal momento che sono proprio queste le strutture verso le quali è rivolto lo strumento per la valutazione della vulnerabilità sismica mediante le curve di fragilità oggetto della presente trattazione. Ad ogni classe di resistenza

numero barre armatura longitudinale:	C16/20
diametro barre armatura:	C20/25
copriferro:	C25/30
Caratteristiche meccaniche dei materiali	
Classe di calcestruzzo:	C28/35
Classe di acciaio:	C32/40
	C16/20
	FeB32k fy=31 ...

Figura 4.10: Menù per la scelta delle classi di resistenza del calcestruzzo.

numero barre armatura longitudinale:	U
diametro barre armatura:	0
copriferro:	0
Caratteristiche meccaniche dei materiali	
Classe di calcestruzzo:	FeB32k fy=315 MPa
Classe di acciaio:	FeB38k fy=375 MPa
	FeB44k fy=430 MPa
	FeB32k fy=31 ...

Figura 4.11: Menù per la scelta delle classi di resistenza dell'acciaio d'armatura.

del materiale scelto corrisponde una precisa suddivisione in intervalli per tenere conto della aleatorietà di tali proprietà, ma questo argomento verrà trattato più nello specifico nei prossimi paragrafi.

Una volta compilati opportunamente tutti i campi presenti, alla pressione del pulsante ok la finestra di dialogo verrà chiusa e con essa, in modo automatico, si chiuderà anche la figura per l'assemblaggio della sezione del traverso, tutti i dati forniti in input vengono salvati nella structure *handles* e l'intero blocco viene passato allo script principale, che effettua un controllo sui valori introdotti dall'utente, qualora si dovesse verificare l'immissione di valori negativi o nulli, viene stampato a schermo, nell'interfaccia principale, un avviso riportante la variabile con valore errato. In tal caso l'utente dovrà premere il pulsante prosegui, si riaprirà la finestra di dialogo e dovranno essere immessi nuovamente tutti i dati. Per effettuare tale controllo, il codice principale, si avvale di confronti logici *if - end*.

4.4 ANALISI DINAMICA NON LINEARE CON OPEN-SEES

4.4.1 OpenSees

OpenSees è l'acronimo di *Open System for Earthquake Engineering Simulation*, si tratta di un software a licenza gratuita, (*open-source*), orientato agli oggetti, impiegato nell'ambito dell'ingegneria sismica per la simulazione dell'azione del terremoto su modelli numerici di strutture o sistemi geotecnici. È un programma finanziato dall'istituto PEER (Università della California, Berkley) ma ad accesso libero. Infatti la grande potenzialità di questo codice è che fino ad oggi molti ricercatori lo hanno arricchito con componenti aggiuntive che consentono di eseguire sofisticate simulazione della risposta delle strutture nei confronti dell'azione sismica. Tali ulteriori implementazioni riguardano principalmente la fase di costruzione del modello, la formulazione analitica di ciascun elemento, modelli di materiale, procedure di analisi, sottoprogrammi che si occupano della risoluzione numerica di equazioni non lineari, strumenti per la gestione dei dati. OpenSees è principalmente scritto in linguaggio C++, ma all'occorrenza richiama delle librerie implementate in linguaggio *Fortran* o *C*, è composto da una struttura modulare e gli sviluppatori possono inserire o modificare singolarmente specifici moduli.

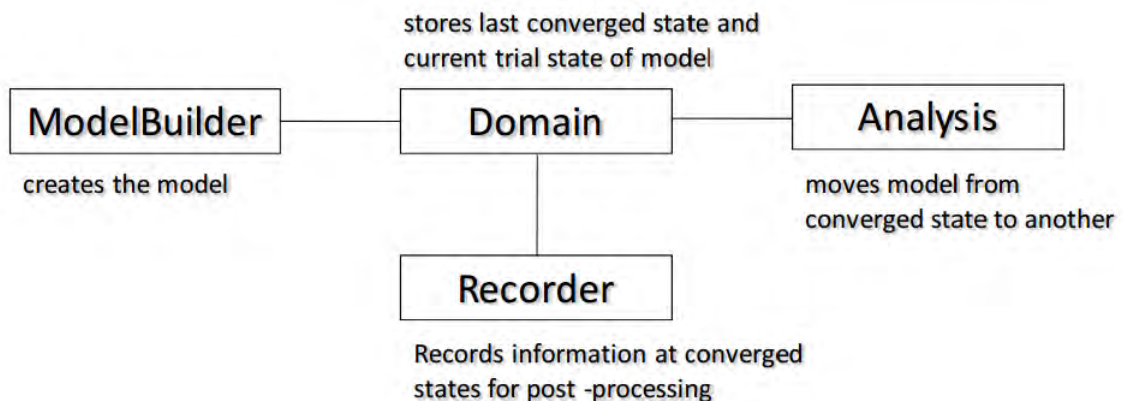


Figura 4.12: Principali oggetti del software OpenSees.

Si riportano in Figura 4.12 i principali oggetti e la loro organizzazione in una struttura gerarchica all'interno del software. Ad esempio, l'oggetto *Domain*, creato a sua volta dall'oggetto *ModelBuilder*, fornisce lo stato dell'elemento finito e varia al variare dell'oggetto *Analysis*. Invece, l'oggetto *Recorder* si occupa di

reperire opportune informazioni utili in seguito per il post-processore e per la visualizzazione a schermo dei risultati.

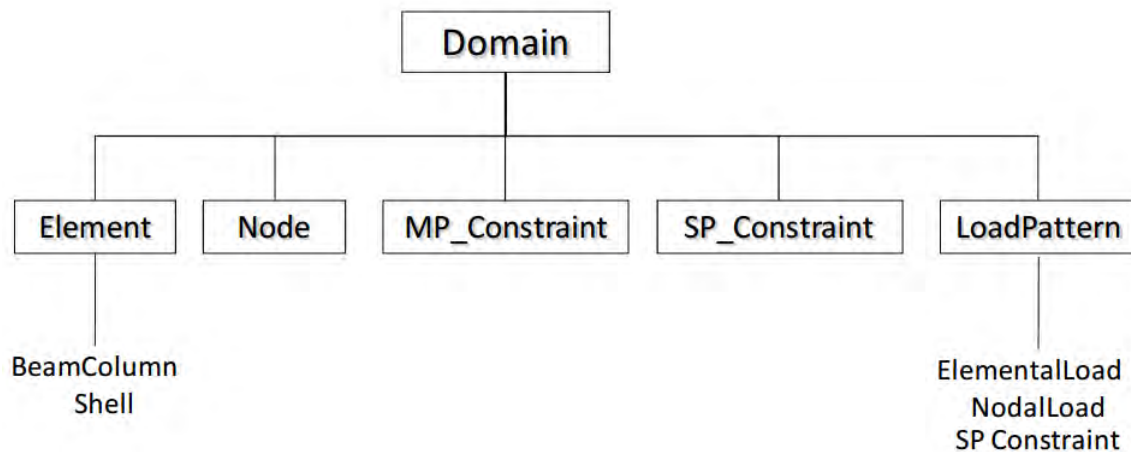


Figura 4.13: Struttura gerarchica dell'oggetto *Domain*.

A sua volta, l'oggetto *Domain* è formato da altri elementi, così come riportato in Figura 4.13: il tipo di elemento finito, i nodi, le condizioni al contorno, i carichi, le condizioni di vincolo interne ed esterne. Sul sito <http://opensees.berkeley.edu/> è molto semplice reperire informazioni e circa la sintassi corretta per costruire gli oggetti, anche attraverso esempi di modelli preconfezionati, o discussioni all'interno della community del forum di Opensees.

4.4.2 Creazione dei file di comando

Giunti a questo punto della procedura l'utente è stato in grado di assemblare un modello tridimensionale del ponte in esame, di specificarne geometria e caratteristiche meccaniche degli elementi strutturali e di visualizzare il risultato in un ambiente grafico 3D, il passo successivo è effettuare la simulazione sismica mediante analisi dinamica non lineare. Per fare ciò ci si avvale di un software gratuito, il già citato *OpenSees*, tuttavia tale programma deve ricevere in input dei precisi files di comando, redatti con opportuna sintassi. Allora, in questo caso entra in gioco il cuore dell'intero lavoro della presente tesi, ovvero un insieme di script che, a partire dai dati immessi dall'utente e dal modello 3D, compila una serie di files di comando, con estensione *.tcl*, in modo che siano fruibili da *OpenSees* per lanciare l'analisi sismica. I *.tcl* necessari sono molteplici, per la compilazione di ciascuno dei quali è stato creato un opportuno programma denominato *editor*. Nell'ambito di uno script di matlab il comando specifico per creare

un generico file di testo è `fopen('nomefile.tcl','w')`, il flag `'w'` (*writable*), serve per renderlo accessibile e modificabile, in seguito, per scrivere all'interno di tale file si usa il comando `fprintf(fileID,'set','corpo del testo')`, il `'set'` è una serie di istruzioni per determinare la spaziatura e la distribuzione dei caratteri riportati nel corpo del testo. Una volta finita la compilazione del file di testo, questo va chiuso, attraverso l'istruzione: `fclose(fileID)` inoltre viene salvato in modo automatico nella cartella di lavoro corrente di matlab.

Vediamo, nel dettaglio, per ogni file di comando di OpenSees, la sua funzione e le sue caratteristiche:

- **NodeCoord.tcl** è il file che contiene la numerazione di tutti i nodi con le relative coordinate lungo gli assi x, y e z. Le coordinate di ciascun nodo vengono ottenute a partire dalle medesime terne di matrici ($X_p, Y_p, Z_p, X_t, Y_t, Z_t$ e X_l, Y_l, Z_l) che sono state utilizzate per costruire il grafico 3D all'interno dell'interfaccia di matlab. Per comprendere meglio il criterio di numerazione dei nodi si riporta un modello molto semplice di ponte e il relativo file *NodeCoord.tcl*, per garantire l'omogeneità dimensionale con il resto delle grandezze, sebbene un viadotto abbia tipicamente dimensioni dell'ordine delle decine di metri, è stato necessario esprimere le coordinate dei punti in millimetri. Come si può meglio notare nell'immagine, alcuni nodi hanno una doppia o tripla numerazione, questo è necessario dal momento che se si vuole modellare, ad esempio, l'appoggio dell'impalcato mediante cerniere, bisogna creare in uno stesso punto due nodi coincidenti e unirli con elementi zerolenght, mentre in corrispondenza del traverso superiore delle pile ogni nodo ha una tripla numerazione perchè sullo stesso punto sussistono tre nodi coincidenti: una coppia serve per l'elemento zerolenght che modella la cerniera di una campata, mentre l'altra coppia rappresenta l'appoggio della campata successiva.

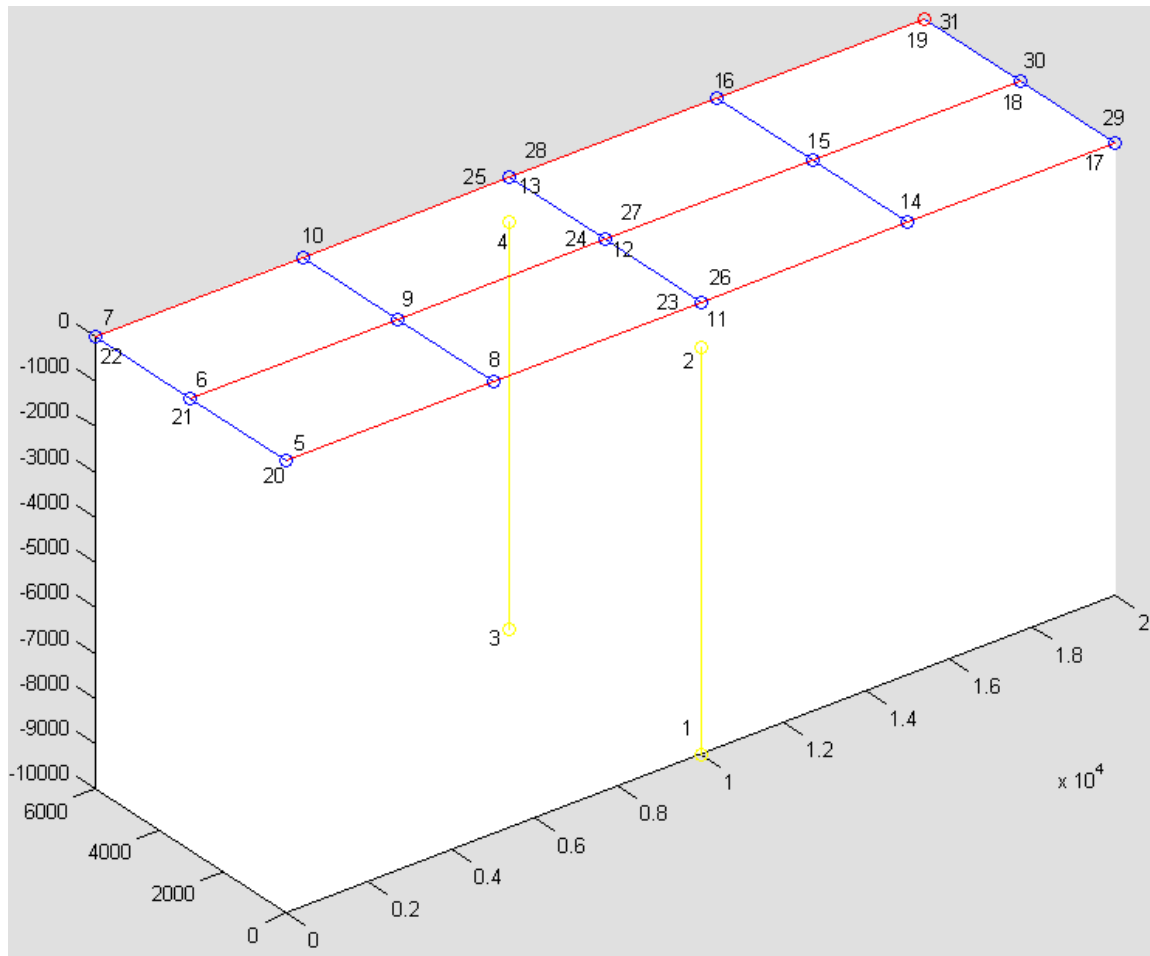


Figura 4.14: Numerazione dei nodi del modello.

```

1 # NodeCoord.tcl
2
3 # Node      tag      xCrd      yCrd      zCrd
4 node        1 +1.000000E+004 +0.000000E+000 -1.000000E+004
5 node        2 +1.000000E+004 +0.000000E+000 -1.000000E+003
6 node        3 +1.000000E+004 +6.000000E+003 -1.000000E+004
7 node        4 +1.000000E+004 +6.000000E+003 -1.000000E+003
8 node        5 +0.000000E+000 +0.000000E+000 +0.000000E+000
9 node        6 +0.000000E+000 +3.000000E+003 +0.000000E+000
10 node       7 +0.000000E+000 +6.000000E+003 +0.000000E+000
11 node       8 +5.000000E+003 +0.000000E+000 +0.000000E+000
12 node       9 +5.000000E+003 +3.000000E+003 +0.000000E+000
13 node      10 +5.000000E+003 +6.000000E+003 +0.000000E+000
14 node      11 +1.000000E+004 +0.000000E+000 +0.000000E+000
15 node      12 +1.000000E+004 +3.000000E+003 +0.000000E+000
16 node      13 +1.000000E+004 +6.000000E+003 +0.000000E+000
17 node      14 +1.500000E+004 +0.000000E+000 +0.000000E+000
18 node      15 +1.500000E+004 +3.000000E+003 +0.000000E+000
19 node      16 +1.500000E+004 +6.000000E+003 +0.000000E+000
20 node      17 +2.000000E+004 +0.000000E+000 +0.000000E+000
21 node      18 +2.000000E+004 +3.000000E+003 +0.000000E+000
22 node      19 +2.000000E+004 +6.000000E+003 +0.000000E+000
23 node      20 +0.000000E+000 +0.000000E+000 +0.000000E+000
24 node      21 +0.000000E+000 +3.000000E+003 +0.000000E+000
25 node      22 +0.000000E+000 +6.000000E+003 +0.000000E+000
26 node      23 +1.000000E+004 +0.000000E+000 +0.000000E+000
27 node      24 +1.000000E+004 +3.000000E+003 +0.000000E+000
28 node      25 +1.000000E+004 +6.000000E+003 +0.000000E+000
29 node      26 +1.000000E+004 +0.000000E+000 +0.000000E+000
30 node      27 +1.000000E+004 +3.000000E+003 +0.000000E+000
31 node      28 +1.000000E+004 +6.000000E+003 +0.000000E+000
32 node      29 +2.000000E+004 +0.000000E+000 +0.000000E+000
33 node      30 +2.000000E+004 +3.000000E+003 +0.000000E+000
34 node      31 +2.000000E+004 +6.000000E+003 +0.000000E+000

```

Figura 4.15: File NodeCoord.tcl, si nota la corrispondenza con la numerazione riportata nel modello.

- **Materials.tcl**, in questo file di comando bisogna scegliere la natura dei materiali, a partire da delle tipologie presenti nella libreria di OpenSees, e poi specificarne le proprietà. Per il calcestruzzo si è scelto il materiale *Concrete01*, mentre per l'acciaio si usa lo *Steel01*. Naturalmente le proprietà meccaniche di ciascun materiale sono stabilite nel rispetto della classe di resistenza scelta dall'utente e dei vari intervalli determinati per tenere conto dell'incertezza sui materiali.
- **MPconstraint.tcl**, in questo .tcl vengono specificati i vincoli multipunto, nell'ambito del modello in esame vengono creati solo dei *rigid link*, che risultano necessari per collegare il nodo in testa alla pila con il traverso superiore, dal momento che questi elementi sono sfalsati di una distanza pari all'altezza del traverso.

- **SPconstraint.tcl**, in questo file di comando bisogna dichiarare i vincoli attribuiti ai singoli nodi, nel caso in esame sono soggetti al vincolo di incastro le basi delle pile e i nodi che identificano le spalle del ponte.
- **NodeMass.tcl** in questo file viene specificata la massa nodale. Ora, dal momento che pile e traversi superiori vengono rappresentati da elementi a fibre, che hanno una densità propria, si rende necessario specificare la massa solamente dei nodi che fanno parte dell'impalcato. Si è deciso di calcolare separatamente la massa di ogni campata, sommando il peso proprio degli elementi strutturali in essa presenti più il sovraccarico della soletta, per il quale si è assunto un valore ragionevole di $10\text{KN}/\text{m}^2$, poi la massa dell'intera campata viene divisa per il numero di nodi presenti nella stessa, ottenendo così la massa nodale.
- **Sections.tcl**, in questo file viene specificata la geometria delle sezioni degli elementi a fibre del modello, ovvero pile e traversi superiori. Per quanto riguarda la sezione della pila questa viene suddivisa in due regioni concentriche, il *core*, che sarebbe il calcestruzzo confinato all'interno delle barre di armatura e la *cover*, che sarebbe il copriferro di calcestruzzo non confinato, il valore di resistenza a compressione di quest'ultimo viene ridotto di un fattore di 1.2. Diverso è il discorso per quanto riguarda la sezione del traverso superiore, dal momento che viene lasciata libertà all'utente di creare sezioni rettangolari o a T. Allora, per dichiarare la sezione di tale elemento nel file .tcl si ricorre all'assemblaggio di otto rettangoli, che rappresentano il copriferro di calcestruzzo non confinato e altri due rettangoli che rappresentano il nocciolo confinato, secondo uno schema che viene meglio esemplificato nella figura seguente.
- **Elements.tcl**, è il file nel quale vengono specificate le caratteristiche di ogni elemento strutturale, nell'ottica di automatizzare il processo di compilazione dei file di comando, lo script editor dichiara tutti gli oggetti in ordine raggruppandoli per tipologia.

Per primi vengono elencati gli elementi a *fibre*, cioè le pile e il loro traverso superiore, che sono oggetti di tipo *non linear Beam Column*, poi gli elementi elastici dell'impalcato, cioè i traversi e le travi longitudinali, che sono del tipo *elastic Beam Column*, infine vengono specificati gli *zero length elements* che rappresentano le cerniere che collegano l'impalcato rispettivamente con le spalle e con le pile.

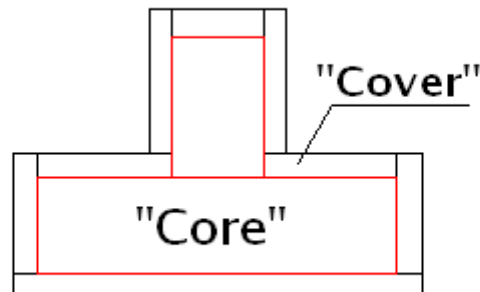


Figura 4.16: Elementi rettangolari per l'assemblaggio di sezioni generiche create dall'utente.

- **GeoTran.tcl**, questo comando effettua una trasformazione lineare di coordinate degli elementi strutturali, dal sistema locale a quello globale, considerando gli effetti $P - \Delta$ per tenere conto della non linearità data dai momenti di secondo ordine.
- **AnalysisRecorder.tcl**, questo comando specifica quali sono i risultati che OpenSees deve fornire in output per ogni tipologia di analisi che viene lanciata.
- **LoadPattern.tcl**, questi .tcl specificano l'entità e la distribuzione del carico per ogni tipologia di analisi che viene lanciata. Il *LoadPattern1* si riferisce all'analisi statica e considera il peso proprio di pile e traversi, mentre il carico gravitazionale dell'impalcato viene distribuito sullo stesso mediante strisce di carico longitudinali. Poi ci sono altri tre LoadPattern (ognuno dei quali specifica quale accelerogramma usare per la direzione trasversale, longitudinale e verticale) per ciascuna analisi dinamica.
- **PontePC.tcl**, questo è il file di comando principale, l'unico che deve essere caricato nel prompt di OpenSees. Durante la sua esecuzione richiama, con ordine opportuno, tutti i file .tcl sopra riportati ed esegue, per ogni ponte campione tutte le analisi, che sono, in ordine: analisi statica con solo il carico gravitazionale e sei analisi dinamiche con rispettivi valori di PGA pari a: 0.1g, 0.2g, 0.3g, 0.4g, 0.5g, 0.6g.

Tutti i files i comando sopra illustrati vengono compilati in modo del tutto automatico dal codice collegato all'interfaccia, all'atto della pressione, da parte

dell'utente, del pulsante *scrivi file .tcl*. Dal momento che, come verrà trattato nello specifico in seguito, dalla combinazione dei valori di resistenza dei materiali risultano 15 ponti campioni e dal momento che la normativa (T.U.2008) impone l'utilizzo di almeno tre terne di accelerogrammi per le analisi in Time-history, l'interfaccia organizza i file di comando così creati in 15 sottocartelle denominate *PC1*, *PC2*, ... ,*PC15* all'interno di 3 cartelle, *TERNA1*, *TERNA2*, *TERNA3*, per ciascuna terna di accelerogrammi spettrocompatibili. L'interfaccia propone di default un set di accelerogrammi, ma l'utente è libero di fornirne degli altri semplicemente sostituendoli a quelli già presenti nella cartella denominata *file comuni*, avendo però l'accortezza di nominarli nello stesso modo di quelli già presenti e di non alterare gli altri file presenti in tale cartella.

4.4.3 Lanciare l'analisi con OpenSees

Una volta che l'interfaccia ha compilato correttamente tutti i file di comando, è possibile lanciare le analisi dinamiche non lineari, il tutto avviene in maniera completamente automatica, l'utente dovrà solo premere, all'interno dell'interfaccia il pulsante *Lancia l'analisi*. Il codice collegato a tale pulsante fa partire l'eseguibile

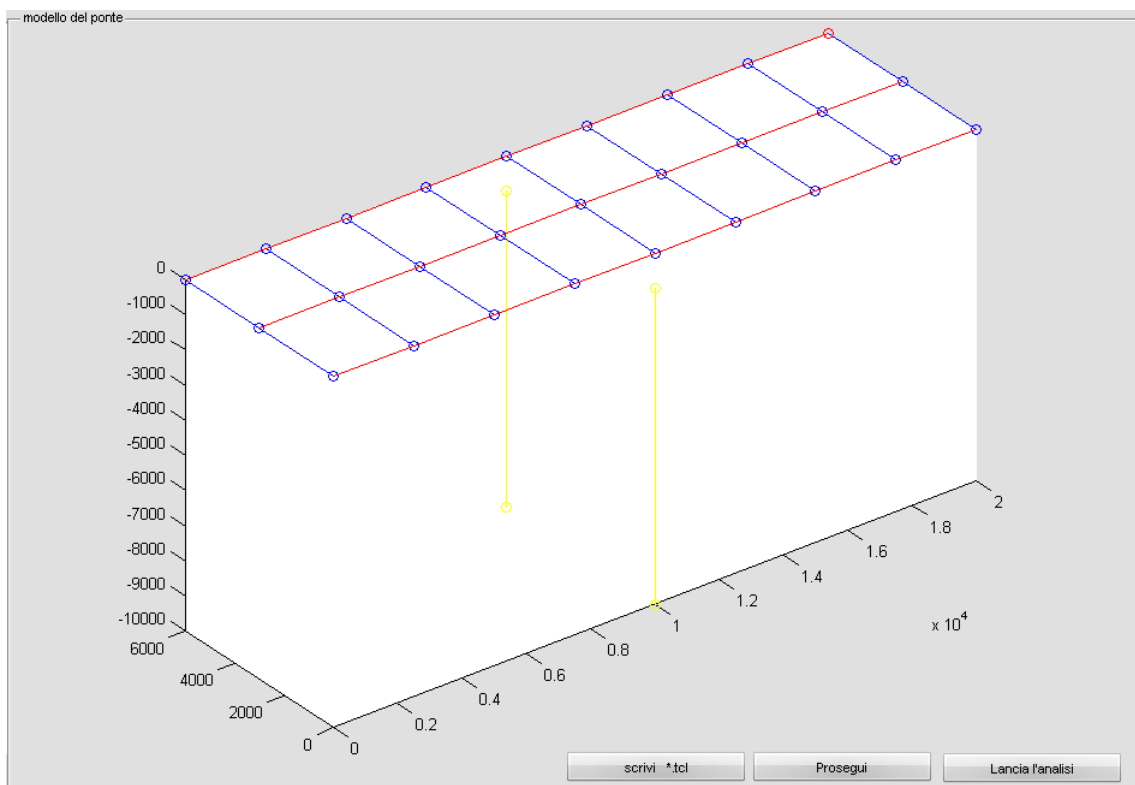


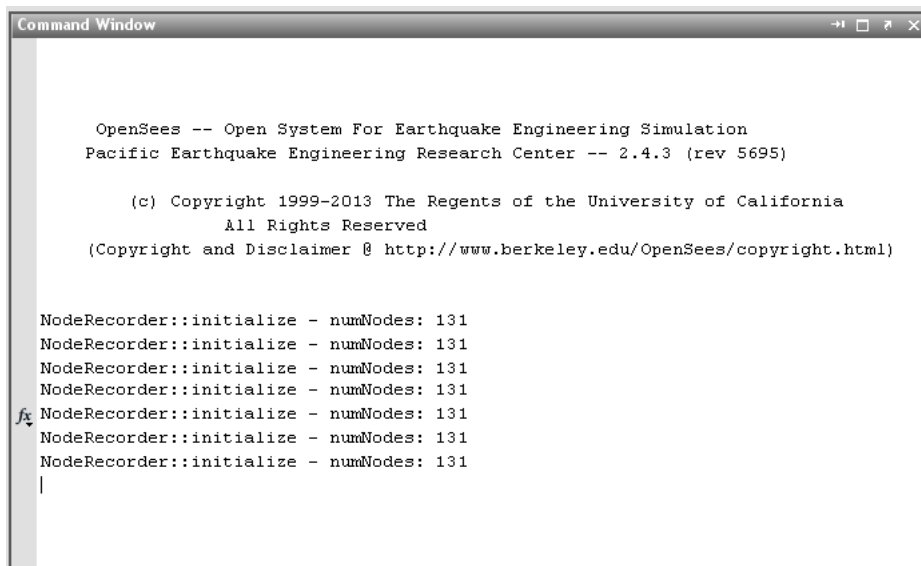
Figura 4.17: Interfaccia grafica con il modello assemblato, pronta per lanciare le analisi dinamiche.

tramite il comando `!Opensees.exe main.tcl`, dove il file di comando, `main.tcl`, che si trova già preimpostato nella cartella principale dell'interfaccia, si occuperà, in maniera automatica, di lanciare ogni analisi dinamica per ogni ponte campione di ciascuna terna.

Questa fase della procedura si rivela sicuramente la più onerosa in termini computazionali e di tempo, dal momento che, sebbene il modello numerico del ponte sia volutamente realizzato con uno schema piuttosto semplice, il numero di analisi da effettuare è molto elevato.

Infatti, per ogni ponte campione, per avere una densità campionaria adeguata, si effettuano dei analisi dinamiche in Time-history, con una PGA di 0.1g, 0.2g, 0.3g, 0.4g, 0.5g, 0.6g, rispettivamente, per ciascuna l'intervallo temporale è di 20 secondi con step del centesimo di secondo. Come già specificato i ponti campione sono 15 e da moltiplicare per 3 terne di accelerogrammi spettrocompatibili, secondo quanto imposto da normativa. Per tanto, in totale, saranno da sostenere: $6 \cdot 15 \cdot 3 = 270$ analisi dinamiche.

Lo stato di avanzamento di OpenSees può essere monitorato in qualsiasi momento attraverso la finestra *Command Window*, all'interno dell'ambiente di matlab, la dicitura `numNodes 131` è riferita all'analisi corrente e riporta il numero di nodi del modello creato.



```
Command Window

OpenSees -- Open System For Earthquake Engineering Simulation
Pacific Earthquake Engineering Research Center -- 2.4.3 (rev 5695)

(c) Copyright 1999-2013 The Regents of the University of California
All Rights Reserved
(Copyright and Disclaimer @ http://www.berkeley.edu/OpenSees/copyright.html)

NodeRecorder::initialize - numNodes: 131
NodeRecorder::initialize - numNodes: 131
NodeRecorder::initialize - numNodes: 131
NodeRecorder::initialize - numNodes: 131
fx NodeRecorder::initialize - numNodes: 131
NodeRecorder::initialize - numNodes: 131
NodeRecorder::initialize - numNodes: 131
|
```

Figura 4.18: Ambiente di matlab dal quale si può monitorare l'avanzamento delle analisi dinamiche effettuate da OpenSees.

4.5 COSTRUZIONE DELLE CURVE DI FRAGILITÀ

4.5.1 Distribuzione probabilistica dei materiali

Per costruire le curve di fragilità si è tenuto conto dell'incertezza sulle proprietà meccaniche dei materiali considerando la distribuzione di probabilità delle loro resistenze, inoltre, al fine di garantire la maggiore adattabilità possibile dell'interfaccia nel rappresentare adeguatamente i ponti oggetto di analisi, si è deciso di fornire un set di cinque classi di resistenza per il calcestruzzo, e tre classi per l'acciaio, come già specificato in precedenza durante l'illustrazione della finestra di dialogo dell'interfaccia.

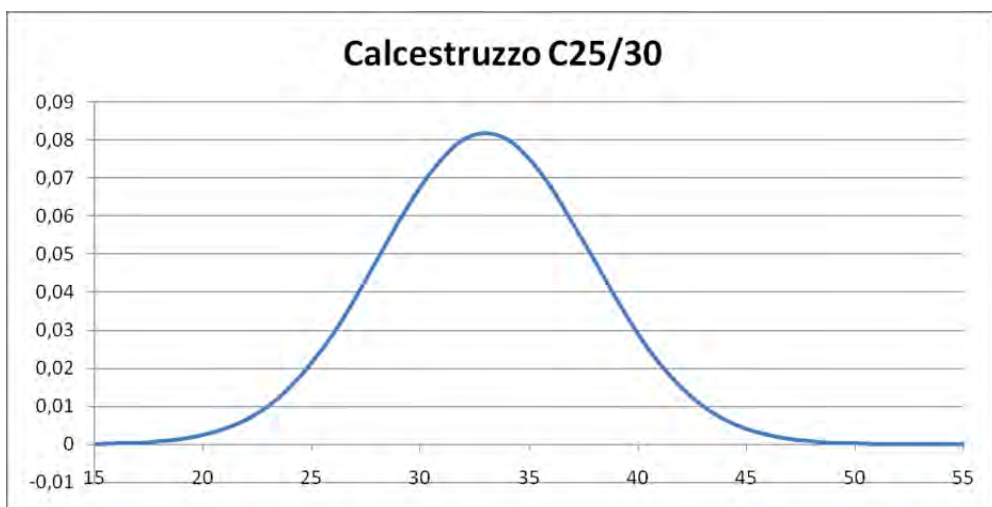
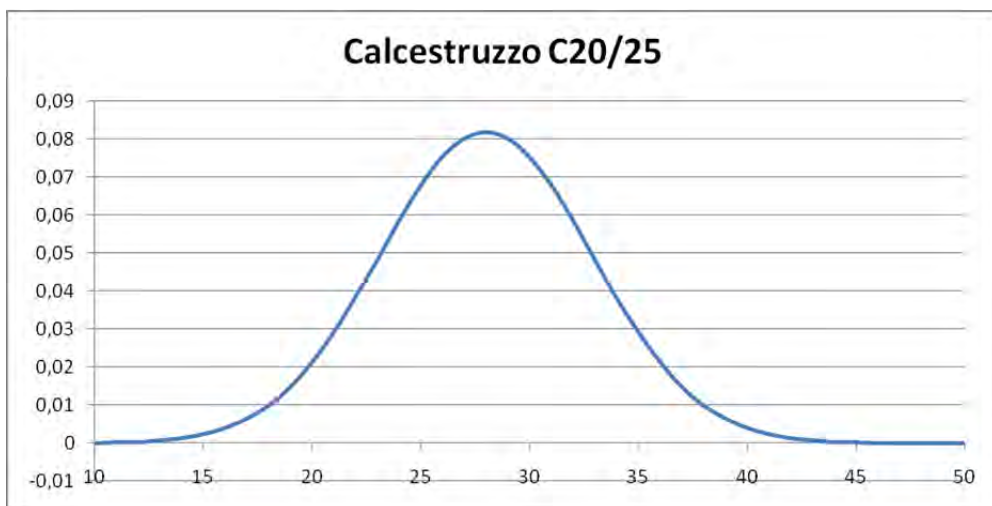
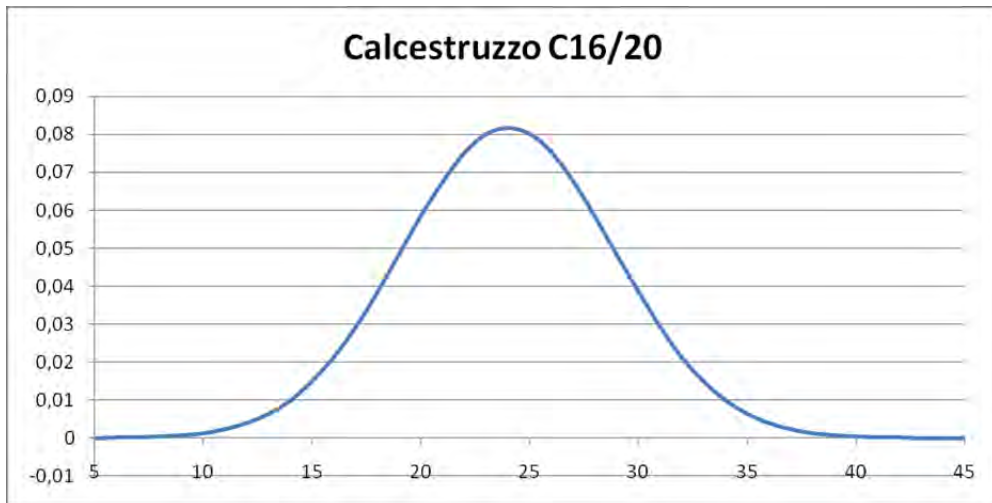
CLASSI DI RESISTENZA DEI MATERIALI	
CALCESTRUZZO	ACCIAIO
C16/20	FeB32k
C20/25	FeB38k
C25/30	FeB44k
C28/35	
C32/40	

Figura 4.19: Classi di resistenza dei materiali disponibili.

Ad esempio per il calcestruzzo in classe C16/20 il primo valore indica la resistenza caratteristica a compressione cilindrica, f_{ck} , mentre il secondo quella cubica, R_{ck} , entrambe espresse in MPa, dalla letteratura (Melchers, 1999) si è potuto constatare che la resistenza non confinata del calcestruzzo sia meglio descritta da una distribuzione probabilistica normale della resistenza cilindrica. Il valore caratteristico coincide con il quinto percentile di tale distribuzione, (ovvero un valore di resistenza che verrà superato con il 95% delle probabilità), inoltre la resistenza cilindrica media si calcola con la seguente relazione (T.U. 2008):

$$f_{cm} = f_{ck} + 8 \text{ [MPa]}$$

Quindi, dal momento che abbiamo a disposizione il 5° e il 50° percentile di una distribuzione probabilistica normale è immediato calcolare σ , cioè la deviazione standard, che risulta: $\sigma = 4,8785$. Quindi è possibile tracciare il grafico della funzione densità di probabilità per ogni classe di resistenza del calcestruzzo.



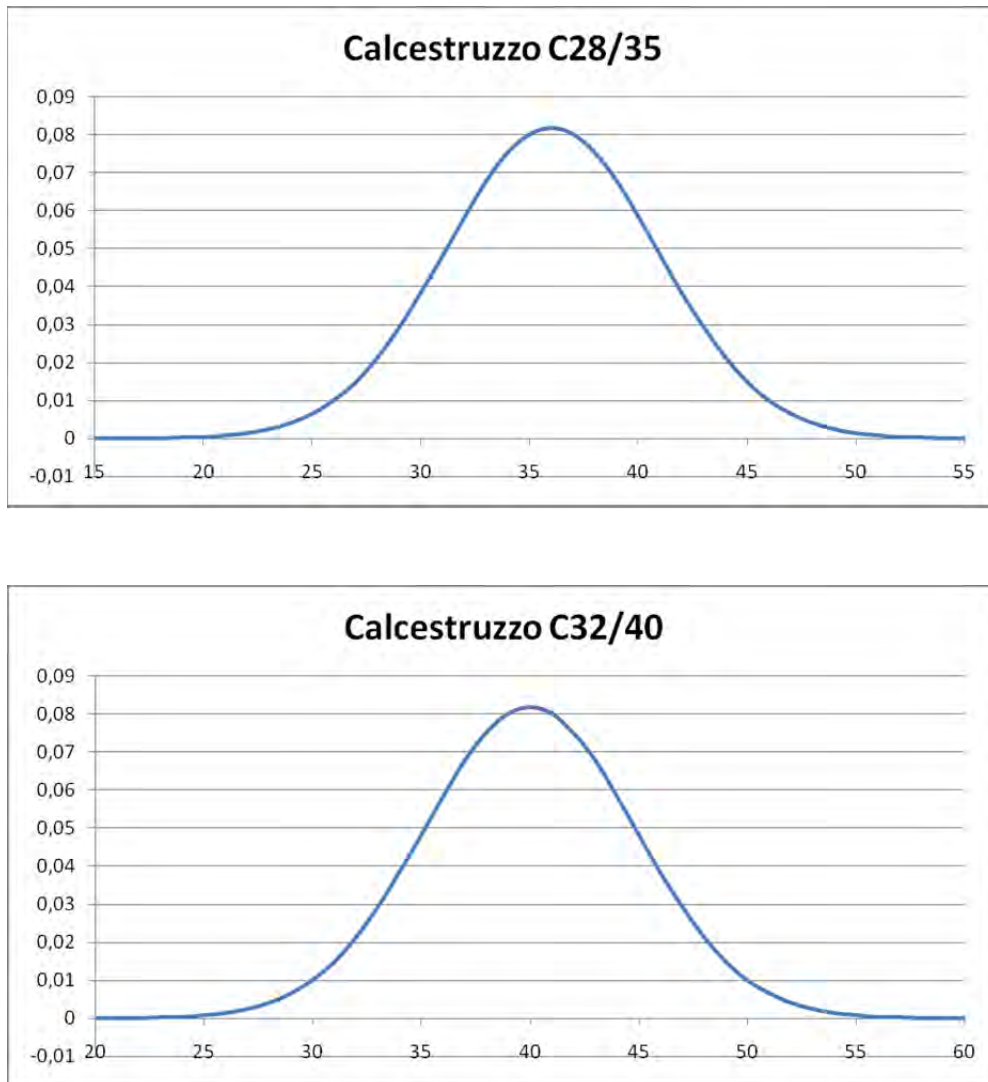
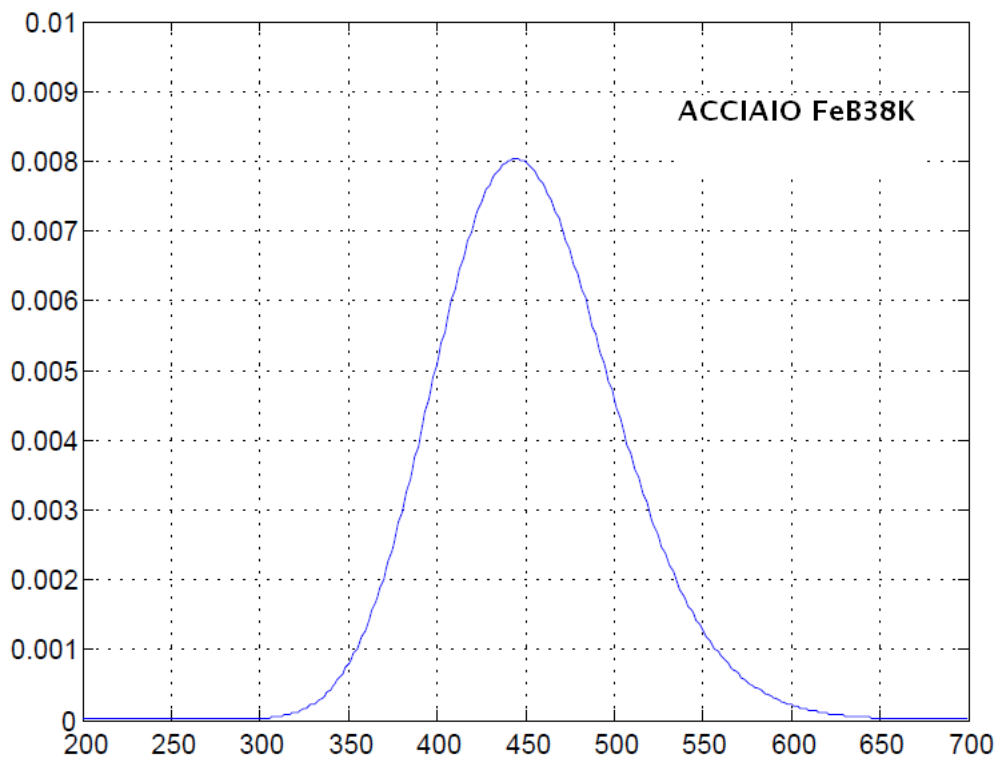
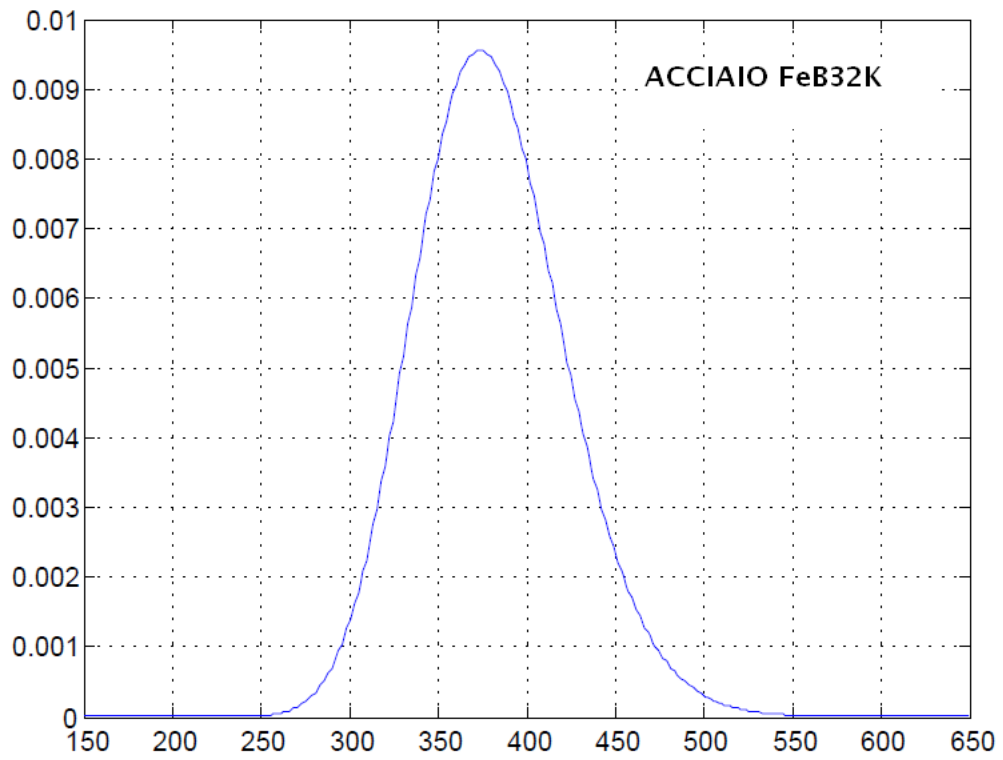


Figura 4.20: Distribuzioni di probabilità normale per le varie classi di resistenza del materiale calcestruzzo.

Per l'acciaio d'armatura, ad esempio in classe FeB32K, invece, da quanto si può riscontrare in letteratura, (Mirza e MacGregor, 1979, Melchers, 1979), si è utilizzata una distribuzione di tipo logonormale, con una deviazione standard pari a 42 MPa. Il valore della resistenza a snervamento caratteristico, f_{yk} , è invece fornito da normativa (D.M. 9 gennaio 1996, par. 2.2), che fornisce altresì la relazione per il calcolo del valore medio:

$$f_{ym} = f_{yk}\gamma_{RD} \quad \text{con } \gamma_{RD} \text{ (fattore di sovrarresistenza)} = 1.2$$

Quindi, anche per l'acciaio, noto il valore medio e la deviazione standard, è immediato tracciare la funzione densità di probabilità per ogni classe di resistenza:



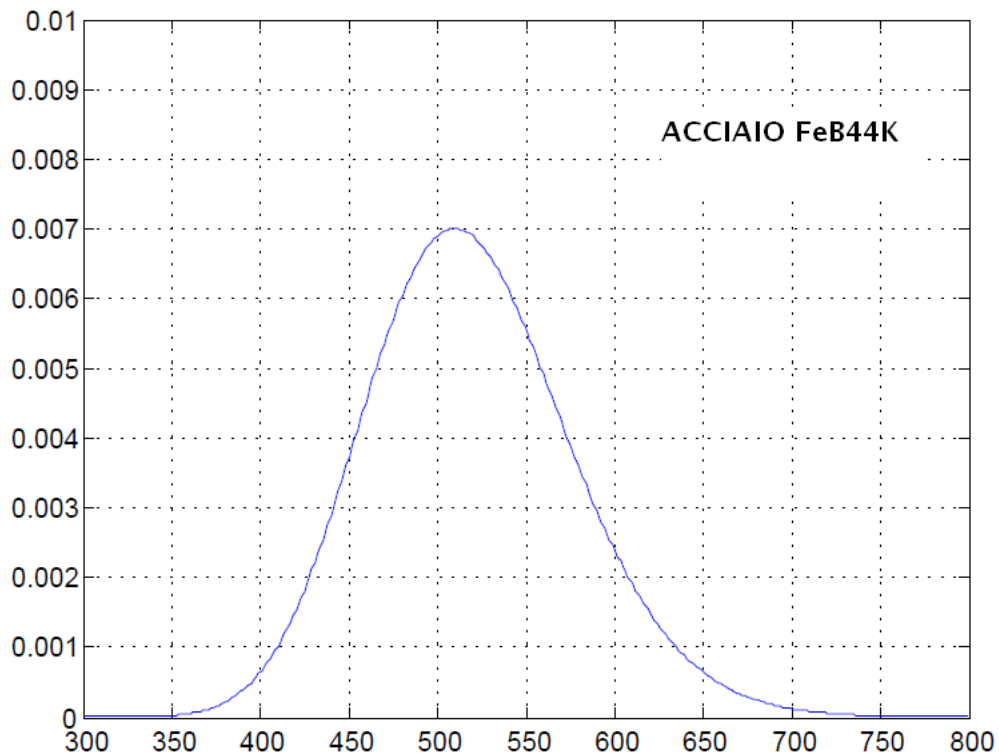


Figura 4.21: Distribuzioni di probabilità logonormale per le varie classi di resistenza del materiale acciaio.

Per considerare l'incertezza sui materiali è stata operata una discretizzazione delle funzioni di probabilità: sono stati individuati 5 intervalli per il calcestruzzo e tre per l'acciaio, in questa maniera, per ogni coppia di valori di resistenza dei materiali scelta dall'utente, si vengono a creare 15 *ponti campioni*, *PC*, che derivano dalla combinazione di questi intervalli. Si riportano di seguito due tabelle contenenti la suddivisione in intervalli di ciascun materiale:

CLASSI DI RESISTENZA	f_{ck}	f_{cm}	intervalli				
C16/20	16	24	10	17	24	31	38
C20/25	20	28	14	21	28	35	42
C25/30	25	33	19	26	33	40	47
C28/35	28	36	22	29	36	43	50
C32/40	32	40	26	33	40	47	54

CLASSI DI RESISTENZA	f_{yk}	f_{ym}	intervalli		
FeB32k	315	378	278	378	478
FeB38k	375	450	350	450	550
FeB44k	430	516	416	516	616

Figura 4.22: Discretizzazione in intervalli per le varie classi di resistenza.

Si riporta, infine, a titolo del tutto esemplificativo, le caratteristiche meccaniche dei materiali per ciascuno dei 15 ponti campione, qualora l'utente scegliesse la combinazione: calcestruzzo in classe C16/20 e acciaio FeB32K. Ciascuno dei ponti campione ha elementi strutturali nominalmente identici, ma statisticamente differenti in base alla resistenza dei materiali e alla probabilità associata alla combinazione delle resistenze, che viene ottenuta mediante il prodotto della probabilità di occorrenza del valore centrale per l'ampiezza del suo intervallo, che è stata appositamente scelta in modo che la somma delle probabilità delle singole combinazioni restituisca valore unitario.

PONTI CAMPIONE	f_c [Mpa]	Prob (f_c)	f_y [Mpa]	Prob (f_y)	Prob ($f_c:f_y$)
PC1	10	0,0144	278	0,0283	0,0004
PC2	17	0,2045	278	0,0283	0,0058
PC3	24	0,5724	278	0,0283	0,0162
PC4	31	0,2045	278	0,0283	0,0058
PC5	38	0,0093	278	0,0283	0,0003
PC6	10	0,0144	378	0,9493	0,0137
PC7	17	0,2045	378	0,9493	0,1941
PC8	24	0,5724	378	0,9493	0,5434
PC9	31	0,2045	378	0,9493	0,1941
PC10	38	0,0093	378	0,9493	0,0088
PC11	10	0,0144	478	0,0808	0,0012
PC12	17	0,2045	478	0,0808	0,0165
PC13	24	0,5724	478	0,0808	0,0463
PC14	31	0,2045	478	0,0808	0,0165
PC15	38	0,0093	478	0,0808	0,0008
				somma	1,000

Figura 4.23: Caratteristiche dei 15 ponti campioni per la coppia di materiali: calcestruzzo C16/20 e acciaio FeB32K.

4.5.2 Calcolo dell'indice di danno, la duttilità cinematica della sezione

Come già trattato nel capitolo precedente, nella metodologia per la costruzione delle curve di fragilità, si è scelto come indice di danno la duttilità cinematica della sezione:

$$D = \mu_{\sigma} = \frac{x_{\max}}{x_y} \quad (4.1)$$

Ma se il valore di spostamento a numeratore viene trovato a partire dai risultati in output che fornisce OpenSees, riguardo lo spostamento dei nodi, come fare, invece, per determinare il denominatore x_y , ovvero lo spostamento a snervamento della testa della pila? Normalmente si determina tale valore mediante una prova monotona sull'elemento considerato (analisi di tipo Push-over), ma in questo caso, dal momento che la sezione della pila, ancorchè circolare, non è fissa ma dipende dall'input dell'utente, si è reso necessario un metodo facilmente estendibile ad una generica sezione circolare.

Andiamo ad analizzare, per semplicità quanto accade per la sezione rettangolare, salvo poi estendere i risultati al caso in esame di sezione circolare. Si consideri una generica sezione rettangolare, di larghezza b , altezza h e area A , si assuma per il calcestruzzo un diagramma tensione-deformazione di tipo triangolo-rettangolo:

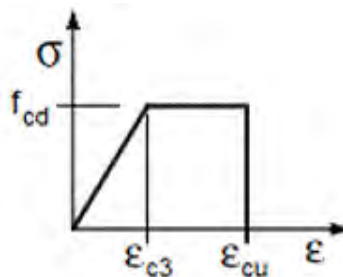


Figura 4.24: Modello $\sigma - \epsilon$ del calcestruzzo (T.U. 2008 par. 4.1.1 (b))

Il momento agente sulla sezione si calcola mediante integrazione delle tensioni sulla medesima:

$$M = \int_A \sigma(y)y dA \quad (4.2)$$

Allora, all'aumentare del momento agente sulla sezione si avranno i seguenti stati tensionali:

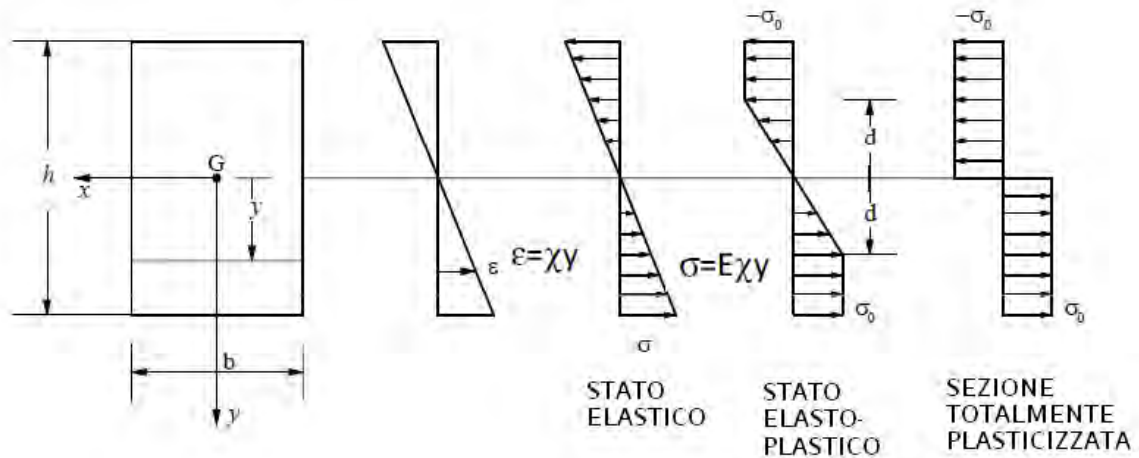


Figura 4.25: Andamento delle tensioni al variare del momento agente sulla sezione

Dal momento che la tensione σ dipende linearmente dalla deformazione ϵ mediante la relazione:

$$\sigma = E\chi y \quad (4.3)$$

Dove χ è la curvatura della sezione, è immediato calcolare il momento agente nella fase elastica:

$$M = b \int_{-\frac{h}{2}}^{\frac{h}{2}} E\chi y y dy = EJ\chi \quad (4.4)$$

dove $J = \int_A y^2 dA$ momento d'inerzia della sezione.

Se si indica con χ_{el} la curvatura della sezione corrispondente al raggiungimento della tensione massima $\sigma_0 = f_{cd}$ all'estradosso e all'intradosso, possiamo chiamare *momento limite elastico* la quantità:

$$M_{el}(\chi_{el}) = EJ\chi_{el} = \frac{2J\sigma_0}{h} = \frac{\sigma_0 b h^2}{6} = \sigma_0 W_{el} \quad (4.5)$$

Con riferimento allo stato deformativo elasto-plastico di Figura 4.25, è immediato giungere al seguente risultato per il calcolo del momento agente sulla sezione:

$$M = 2b \left[\int_0^d E\chi y^2 dy + \int_d^{\frac{h}{2}} \sigma_0 y dy \right] = M_{el} \left[\frac{3}{2 - \frac{1}{2} \left(\frac{\chi_{el}}{\chi} \right)^2} \right] \quad (4.6)$$

Al tendere ad infinito del valore di curvatura della sezione il momento agente raggiunge un valore limite, che indichiamo con M_0 e che prende il nome di

Momento limite plastico o di snervamento:

$$\lim_{\chi \rightarrow \infty} M = M_0 = \frac{3}{2} M_{el} \quad (4.7)$$

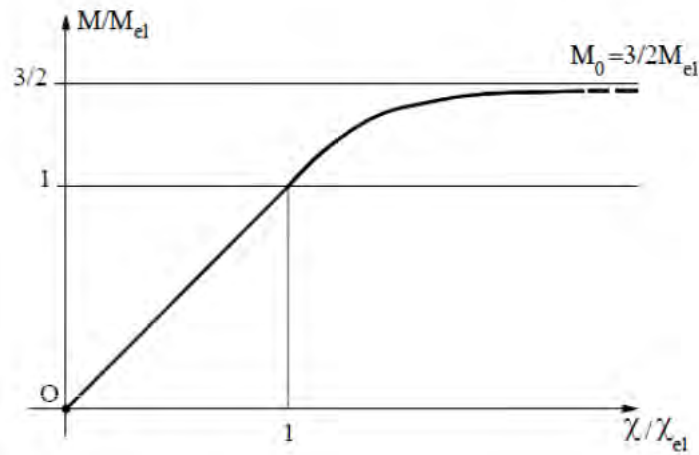


Figura 4.26: Diagramma momento curvatura per la generica sezione rettangolare.

Ora, dal momento che M_0 e M_{el} dipendono linearmente dalla tensione massima σ_0 , se ne ricava che il loro rapporto, M_0/M_{el} , è unicamente funzione della forma della sezione, ed è pari ad un *coefficiente di forma*, α .

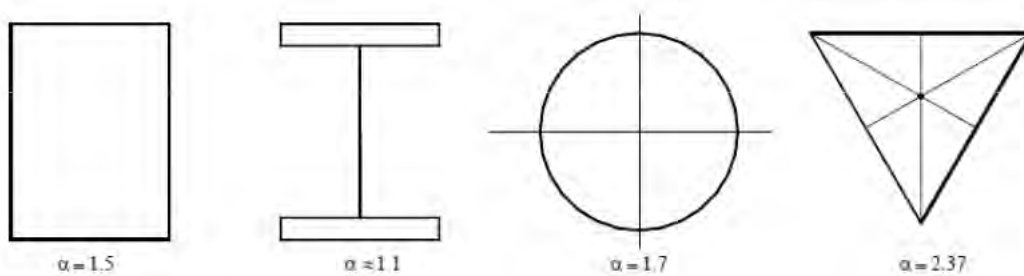


Figura 4.27: Valore del coefficiente di forma per un generico tipo di sezione.

Estendendo la precedente trattazione al caso in esame di sezione circolare, il modulo di resistenza per tale forma, secondo quanto riportato nei prontuari è:

$$W_{el} = \frac{\pi d^3}{32} \quad (4.8)$$

dove con d si indica il diametro della pila espresso in mm. Giunti a questo punto è immediato il calcolo del momento di snervamento per la pila assegnata dall'utente

mediante la relazione:

$$M_0 = \alpha f_{cd} \frac{\pi d^3}{32} \quad (4.9)$$

dove:

- α è pari a 1.7 per la pila a sezione circolare;
- $f_{cd} = \frac{f_{yk}\alpha_{cc}}{\gamma_c}$ secondo la formula (4.1.4) N.T.C 2008, dove i coefficienti valgono rispettivamente: $\alpha_{cc} = 0.85$ $\gamma_c = 1.5$

Ora, si immagini la pila, che ha uno schema statico di una mensola incastrata alla base, sollecitata da una forza concentrata sull'estremo libero:

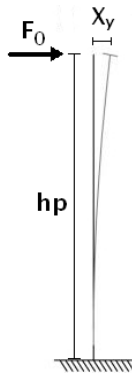


Figura 4.28: Schema statico della pila.

Al crescere di tale forza F , l'elemento giungerà a snervamento quando il momento nella sezione più sollecitata, cioè la base, raggiunge il valore M_0 .

In tale situazione il carico in sommità vale pertanto $F_0 = \frac{M_0}{h_p}$ e dal momento che siamo ancora in condizioni di elasticità, seppure al limite, valgono ancora le relazioni da prontuario che pongono lo spostamento massimo pari a:

$$x_y = \frac{F_0 h_p^3}{3EI} = \frac{M_0 h_p^2}{3EI} \quad (4.10)$$

Dove, il momento d'inerzia della sezione circolare vale (tutte le grandezze geometriche si intendono espresse in mm):

$$I_{pila} = \frac{\pi d^4}{64} \quad (4.11)$$

Pertanto, nell'ottica di automatizzare il procedimento per il calcolo dell'indice di danno, x_y , il codice legato all'interfaccia si avvale della relazione 4.10 che è funzione delle caratteristiche geometriche e meccaniche scelte dall'utente.

4.5.3 Costruzione automatica delle curve di fragilità

Una volta ultimate tutte le analisi dinamiche non lineari da parte di OpenSees, si può passare all'ultima fase, ovvero la costruzione delle curve di fragilità, premendo il pulsante *traccia le curve di fragilità*.

Il programma di analisi sismica restituisce come output dei file di testo contenenti 2000 righe, pari alla discretizzazione temporale adottata, e sei colonne, corrispondenti agli altrettanti gradi di libertà, per ciascun nodo che costituisce il modello del ponte. Dal momento che sono necessari, nell'ambito della presente trattazione, solo gli spostamenti dei punti di controllo, ovvero dei nodi in testa alla pila, sono stati compilati degli appositi script di matlab che convertono i file di output in matrici, e ne estraggono solamente delle opportune colonne. A partire da tali vettori verrà condotta l'opportuna analisi statistica, secondo il procedimento noto come *Metodo 2* e illustrato nel capitolo precedente, per la determinazione delle curve di fragilità in direzione longitudinale e trasversale.

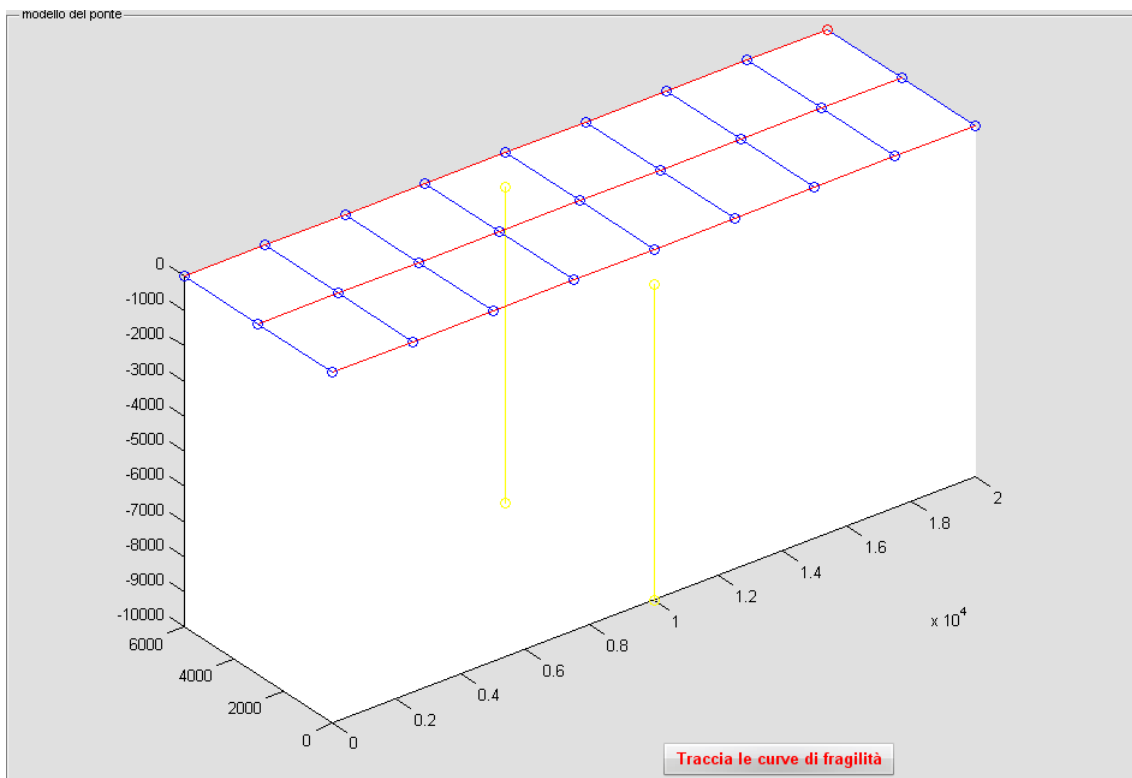


Figura 4.29: Una volta terminate la analisi dinamiche, l'interfaccia è pronta a lanciare gli script per il tracciamento delle curve di fragilità

Si ritiene il caso di fare una puntualizzazione circa il tempo di esecuzione: il comando *load*, che in ambiente di programmazione matlab, carica i file di testo

convertendoli in opportune matrici, impiega un tempo non trascurabile, e dal momento che gli output sono numerosi, l'ultima parte del lavoro dell'interfaccia, cioè la creazione delle curve di fragilità, potrebbe impiegare un tempo dello stesso ordine di grandezza rispetto a quello impiegato per le analisi dinamiche. Pertanto, al fine di facilitare la comprensione dello stato di avanzamento della procedura, si è deciso di implementare una funzione built-in di matlab, la *waitbar*. Durante la fase di caricamento dei risultati compare a schermo una finestra che mostra la barra di avanzamento riferita, rispettivamente, al caricamento dei file di output per il tracciamento delle curve di fragilità in direzione longitudinale e trasversale.

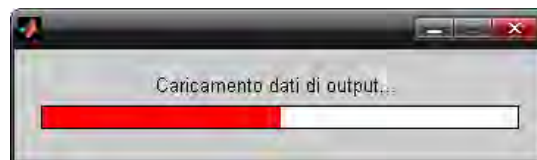


Figura 4.30: Barra di avanzamento per il caricamento dei file di output.

Capitolo 5

RISULTATI

Nel seguente capitolo verranno presentate le curve di fragilità risultanti dall'intera procedura esposta nella presente trattazione, nei confronti di due modelli di ponte presi come esempio. Il primo modello rappresenta un semplice ponte a due campate, con una pila a telaio, i due impalcati hanno uguali dimensioni, ognuno ha una larghezza di 6 metri per una lunghezza di 10 metri, sono stati discretizzati mediante tre traversi per ogni campata e tre travi longitudinali. Le pile sono alte 9 metri, e l'offset tra la testa della pila e l'impalcato è pari ad un metro, lo schema del modello è riportato in Figura 5.1.

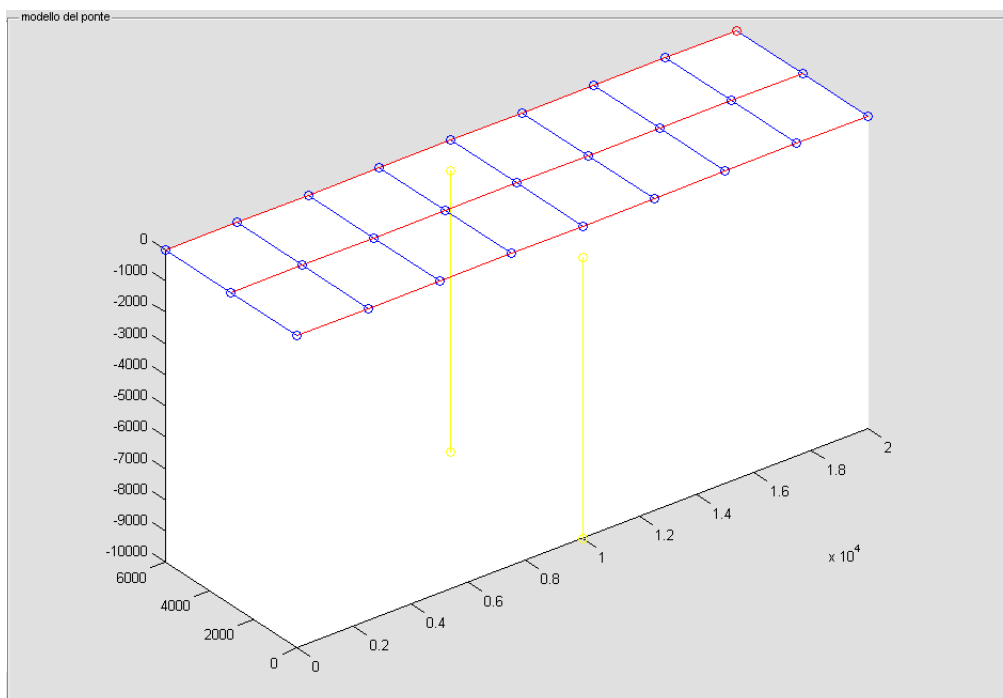


Figura 5.1: Modello numerico del ponte considerato come esempio.

La geometria fornita per gli elementi strutturali è la seguente:

- pile a sezione circolare, aventi diametro di 1,5 metri e armate con 20 barre dal diametro di 24 mm, con copriferro di 50 mm;
- traverso superiore delle pile a T rovescia, avente altezza di 2 metri e base di 2 metri, armata superiormente con 4 barre di diametro 20 mm e inferiormente con 12 barre dello stesso diametro;
- gli elementi strutturali nell'impalcato (traversi e travi longitudinali) hanno medesima area, pari a 500000mm^2 ;
- per il calcestruzzo si è fornito un modulo di elasticità $E = 35000\text{N/mm}^2$, e un modulo di elasticità tangenziale $G = 14000\text{N/mm}^2$;
- si è scelto, per il calcestruzzo una classe di resistenza C25/30 e per l'acciaio d'armatura si è selezionata la classe FeB32K.

Dopo avere compilato i file di comando, e una volta terminate le analisi dinamiche, le curve di fragilità ottenute, rispettivamente per la direzione longitudinale e trasversale, sono mostrate nelle figure seguenti:

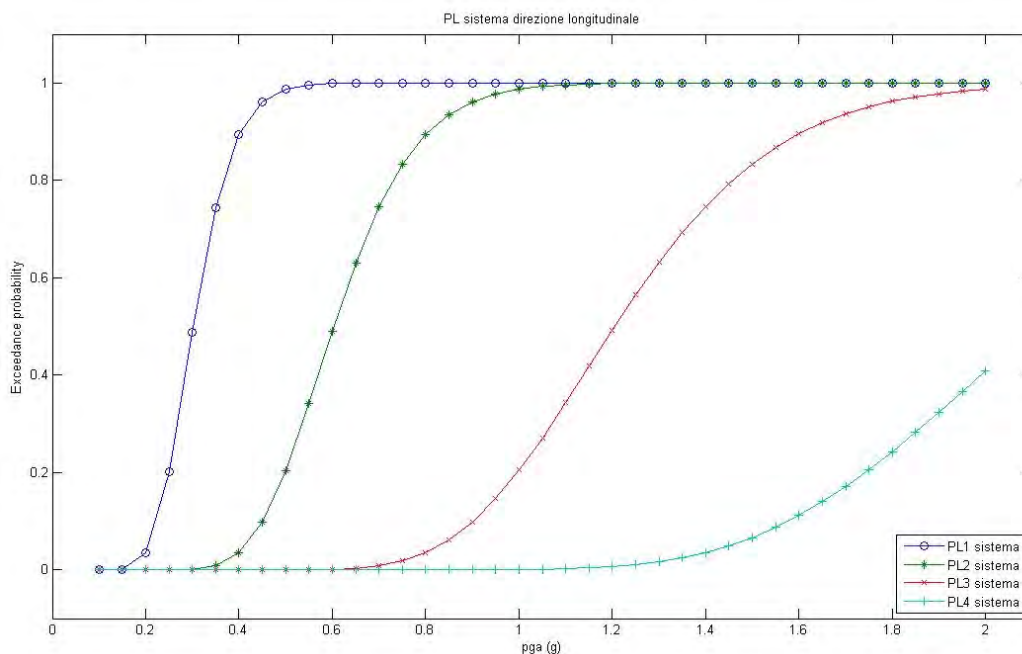


Figura 5.2: Curve di fragilità in direzione longitudinale per il modello di ponte a due campate.

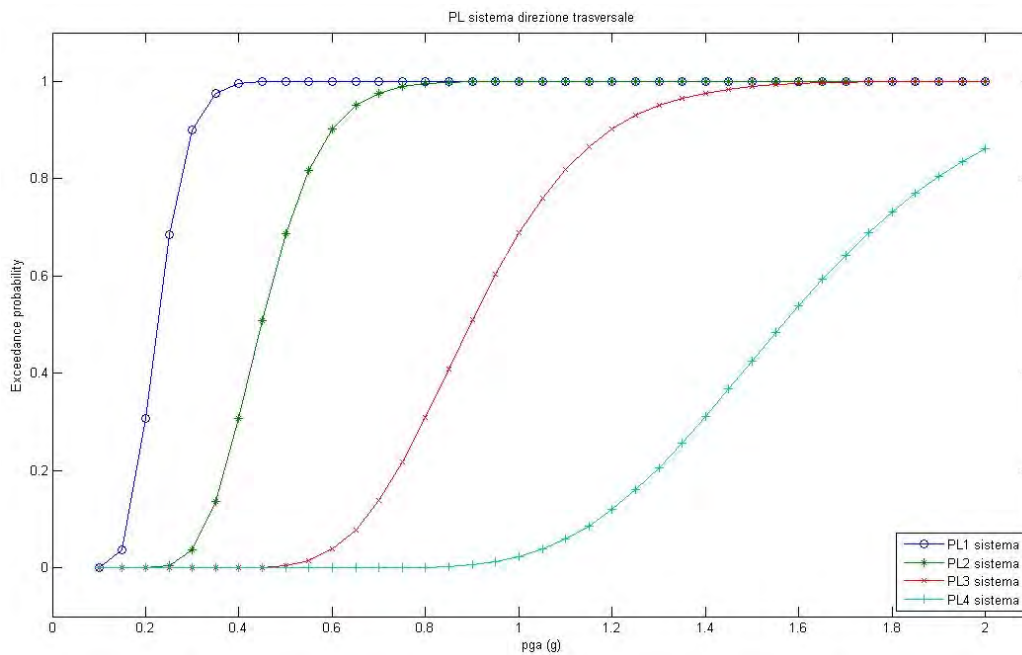


Figura 5.3: Curve di fragilità in direzione trasversale per il modello di ponte a due campate.

Il secondo modello rappresenta un ponte rettilineo a tre campate e due pile a telaio, i tre impalcati hanno medesima dimensione, ognuno è largo 6 metri ed ha una lunghezza pari a 20 metri. In ogni campata troviamo quattro traversi e tre travi longitudinali. Le pile sono alte 9 metri ed è stato posto un interasse tra la testa delle pile e l'impalcato pari ad un metro. Lo schema statico del ponte assume le pile incastrate alla base e l'impalcato in semplice appoggio, modellato mediante cerniere, sulle spalle e in corrispondenza delle pile. La figura seguente riporta la stampa a schermo del suddetto modello creato con l'interfaccia grafica:

La geometria fornita per gli elementi strutturali è la seguente:

- pile a sezione circolare, aventi diametro di 1,5 metri e armate con 20 barre dal diametro di 24 mm, con copriferro di 50 mm;
- traverso superiore delle pile a T rovescia, avente altezza di 2 metri e base di 2 metri, armata superiormente con 4 barre di diametro 20 mm e inferiormente con 12 barre dello stesso diametro;
- gli elementi strutturali nell'impalcato (traversi e travi longitudinali) hanno medesima area, pari a 500000mm^2 ;

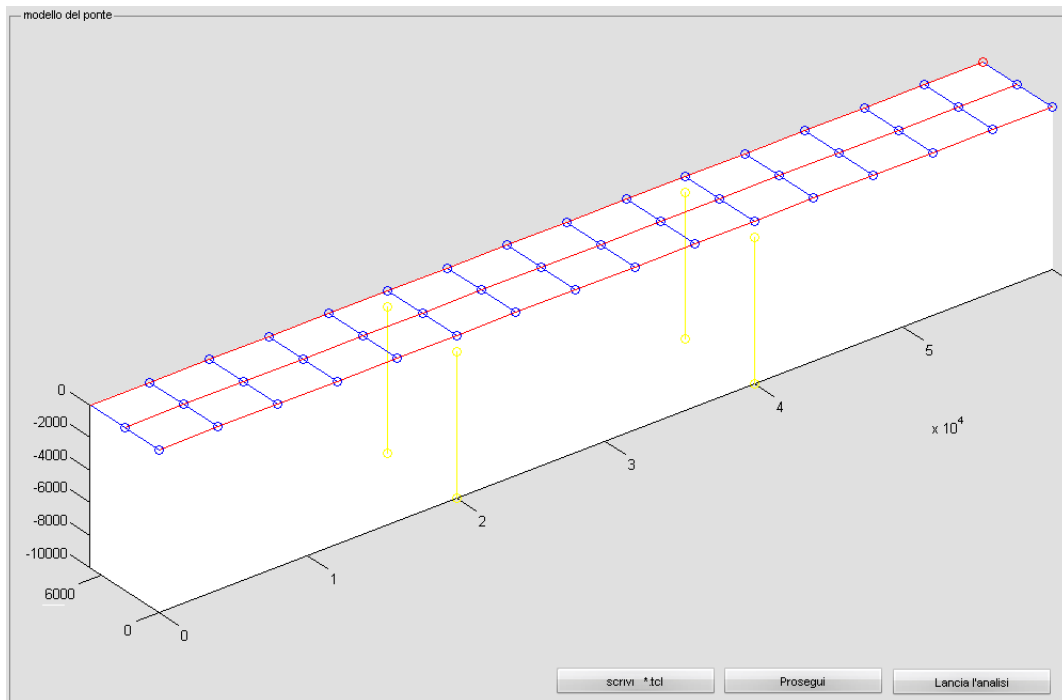


Figura 5.4: Modello numerico del ponte considerato come esempio.

- per il calcestruzzo si è fornito un modulo di elasticità $E = 35000N/mm^2$, e un modulo di elasticità tangenziale $G = 14000N/mm^2$;
- si è scelto, per il calcestruzzo una classe di resistenza C25/30 e per l'acciaio d'armatura si è selezionata la classe FeB32K.

Dopo avere compilato i file di comando, e una volta terminate le analisi dinamiche, le curve di fragilità ottenute, rispettivamente per la direzione longitudinale e trasversale, sono mostrate nelle figure seguenti:

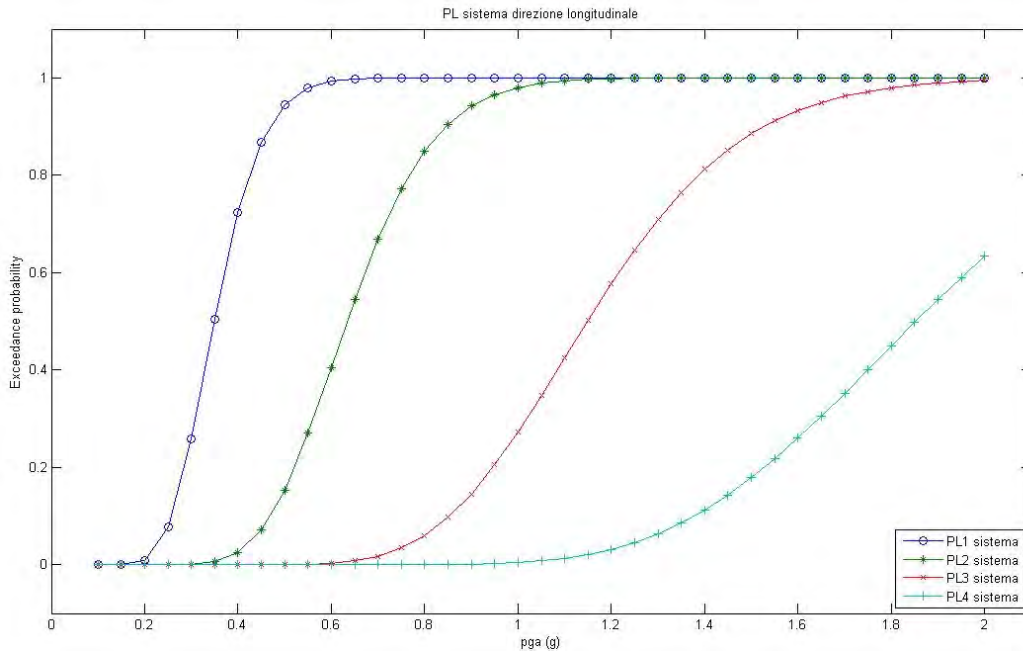


Figura 5.5: Curve di fragilità in direzione longitudinale per il modello di ponte a tre campate.

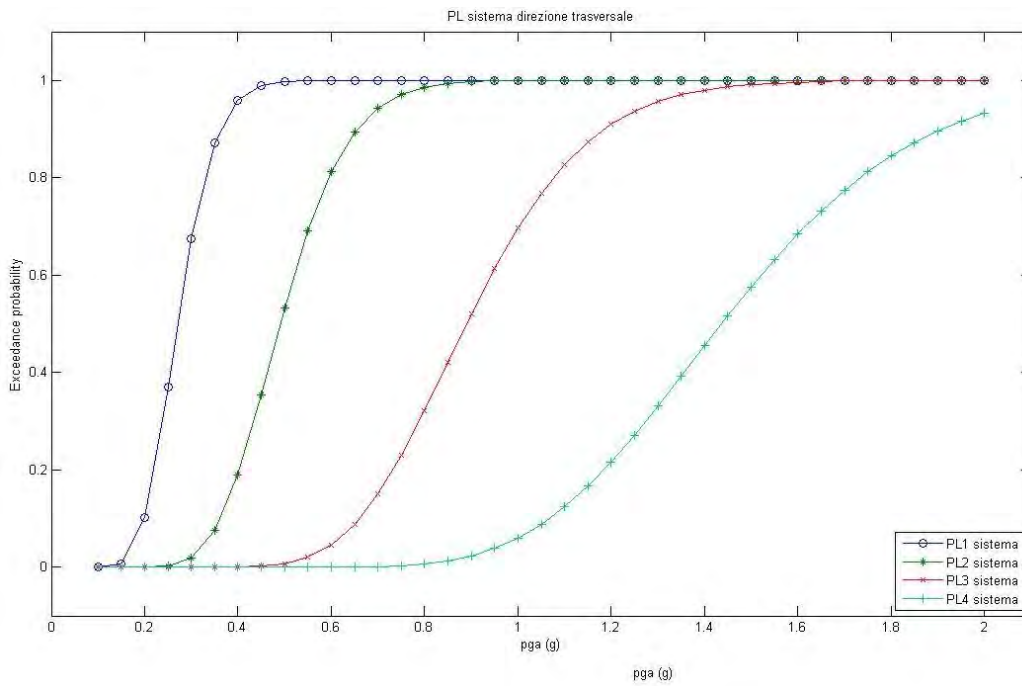


Figura 5.6: Curve di fragilità in direzione trasversale per il modello di ponte a tre campate.

Capitolo 6

CONCLUSIONI E SVILUPPI FUTURI

In questo lavoro di tesi si è provveduto alla creazione di un'interfaccia che potesse rappresentare un rapido ed efficace strumento per la valutazione delle curve di fragilità di tipologie ricorrenti di ponti e viadotti esistenti. Senza dubbio la corretta stima della vulnerabilità sismica di un manufatto è molto importante al fine della redazione di piani, con un adeguato ordine di priorità, per la manutenzione ed eventualmente l'adeguamento di ponti e viadotti della rete stradale. Ma dato l'elevato patrimonio di opere esistenti, risulterebbe un onere non sostenibile eseguire per ciascuno di essi un'analisi dettagliata e approfondita. Pertanto, nell'ottica di sopperire a questa difficoltà si colloca il codice sviluppato nella presente trattazione, fornendo dei risultati con buona approssimazione e con un onere, in termini di tempo e di impegno computazionale, ragionevole, non volendosi, tuttavia, sostituire ad analisi più approfondite, con risultati sicuramente più aderenti alla realtà, laddove lo si ritenga necessario. Un'adeguato piano per la manutenzione dei ponti della rete viaria è sicuramente indispensabile allo scopo di evitare perdite di vite umane, per garantire la necessaria mobilità ai soccorsi e per la ripresa delle attività economiche sulla zona che ha subito il terremoto.

Dall'analisi delle curve di fragilità che risultano dalla procedura di automatizzazione, presentante nel precedente paragrafo, si può valutare l'affidabilità dello strumento proposto dal momento che i diagrammi risultanti sono compatibili con le curve di fragilità, reperibili in letteratura, che derivano da analisi di ponti simili a quelli presi come esempio. In particolare, dal confronto dei risultati tra il primo e il secondo modello, assunti come esempio, risulta che l'analisi del secondo ponte, avente campate con luce maggiore, fornisce delle curve di fragilità più distese, il che significa una maggiore fragilità del manufatto nei confronti del sisma.

Le conclusioni e gli spunti per ulteriori sviluppi futuri che si possono trarre dal seguente lavoro di tesi sono i seguenti:

- La scelta del numero di terne di accelerogrammi influenza pesantemente i risultati dell'analisi dinamica, e di conseguenza anche l'andamento delle curve di fragilità, pertanto, maggiore è il numero di terne che si prendono in considerazione, più la stima della vulnerabilità del manufatto sarà corretta. La normativa vigente prevede di prenderne in considerazione almeno tre, e tale numero è stato considerato nel presente lavoro, tuttavia non si esclude la possibilità di una ulteriore implementazione del codice al fine di poter considerare un numero maggiore di terne di accelerogrammi;
- Al fine di avere un campione consistente per l'analisi probabilistica applicata ai risultati delle analisi dinamiche, ogni ponte campione è stato sottoposto all'azione di 6 valori di PGA, ma questo non esclude la possibilità di implementare un diverso tipo di analisi che preveda un numero maggiore di valori di PGA (ad esempio 0.1g 0.2g 0.3g 0.4g 0.5g 0.6g 0.7g 0.8g 0.9g 0.1g) o intervallati in maniera diversa. Un'applicazione di questo tipo potrebbe fornire valutazioni circa la sensibilità delle curve di fragilità, per uno stesso ponte e per le medesime terne di accelerogrammi, al variare della dimensione del campione statistico;
- La scelta delle classi di resistenza per i materiali da costruzione è stata limitata a 5 classi per il calcestruzzo e 3 per l'acciaio, in particolare, per quest'ultimo sono state prese in considerazione delle classi obsolete, che oggi non vengono più utilizzate, ma che si è ritenuto corretto implementare dal momento che erano quelle maggiormente impiegate nella costruzione di ponti e viadotti fino a qualche decennio fa, tuttavia, non è esclusa la possibilità di aggiornare la libreria dei materiali;
- Il metodo di analisi dinamica non lineare in Time-history è sicuramente il più corretto per questa tipologia di problemi, ma d'altra parte impone un carico computazionale non trascurabile, e dal momento che le analisi da effettuare, per ogni ponte, sono molto numerose, (6 valori di PGA * 15 ponte campione * 3 terne di accelerogrammi=270 analisi dinamiche!), è stato necessario semplificare lo schema statico del modello. Tuttavia, se in alcuni casi lo si ritenesse necessario, è possibile, mediante OpenSees, aggiungere dei dettagli allo schema statico, come ad esempio: tenere in considerazione l'interazione suolo-pila, considerare la rigidità dell'impalcato nella risposta

della struttura all'azione sismica oppure considerare diversi meccanismi di non linearità per le pile, andando ovviamente incontro ad un aumento, anche importante, dell'onere computazionale: rendendo necessario valutare i tempi richiesti da questo tipo di analisi in relazione al livello di accuratezza richiesto dall'indagine di vulnerabilità;

- Il codice legato all'interfaccia è stato, per quanto possibile, sviluppato con una struttura di tre *blocchi*, uno che compone lo strumento pre-processore per la creazione e la visualizzazione del modello tridimensionale, uno per la traduzione di quanto visto a schermo in file di comando comprensibili ad OpenSees, e l'ultimo che si occupa della rielaborazione dei risultati dell'analisi dinamica per la creazione delle curve di fragilità. Questi blocchi sono indipendenti tra loro, in maniera tale da consentire, in futuro, che la modifica di uno di essi non vada ad intaccare le funzionalità del restante programma;

BIBLIOGRAFIA

Testi consultati

- [1] Autori vari, *RISK-UE An advanced approach to earthquake risk scenarios with applications to different European towns*, 2004.
- [2] Battistella S., *Influenza dei meccanismi a taglio nella valutazione della vulnerabilità sismica di tipologie ricorrenti di ponti stradali*, tesi di laurea, relatore Prof. Ing. Pellegrino Carlo, correlatore Ing. Morbin R., Università di Padova, 2013.
- [3] Cornell C. A., Krawinkler H., *Progress and challenges in seismic performance assessment*, Pacific Earthquake Engineering Research Center (PEER), 2000.
- [4] Cosenza E., Manfredi G., *Indici e misure di danno nella progettazione sismica*, CNR-GNDT, 2000.
- [5] Fardis M.N., Carvahlo E.C., Elnashai A., Faccioli E., Pinto P.E., Plumier A., *Guida all'eurocodice 8: progettazione delle strutture in zona sismica*, Epc Libri, Roma, 2011.
- [6] Majorana C., Modena C., Franchetti P., Grendene M., Secchi S., *Fondamenti di dinamica e ingegneria sismica*, ed. McGraw-Hill, 2007.
- [7] Mazzoni S., McKenna F., Scott M.H., Gregory F.L., et. al, *OpenSees Command Language Manual*, The Regents of the University of California, 2007.
- [8] Montgomery D.C., Rungen D.C., Hubele N.F., *Statistica per ingegneria*, ed. Egea, 2004.
- [9] Morbin R., *Strategies for seismic assesment of common existing reinforced concrete bridge typologies*, tesi PhD, tutor Prof. Ing. Pellegrino C., Università di Trento, 2013.

- [10] Morbin R., *Strategie di valutazione della vulnerabilità sismica di tipologie ricorrenti di ponti stradali*, tesi di laurea, relatori Prof. Ing. Modena C. e Prof. Ing. Pellegrino Carlo, correlatore Ing. Grendene M., Università di Padova, 2009.
- [11] *Norme Tecniche per le Costruzioni*, D.M. 14 gennaio 2008 e successivi aggiornamenti.
- [12] Pavese A., *Valutazione della vulnerabilità sismica dei ponti di Catania*, CNR-GNDT, 2000.
- [13] Vitaliani R., Scotta R., Saetta A., *Il calcolo agli Stati Limite delle strutture incalcestruzzo armato*, ed. Libreria Progetto, Padova, 2002.
- [14] Yeo G. L., Cornell C. A., *Stochastic Characterization and Decision Bases under Time-Dependent Aftershock Risk in Performance-Based Earthquake Engineering*, Pacific Earthquake Engineering Research Center, PEER, 2005.

Articoli consultati

- [15] Carvalho J. C. M., Ceccarelli M., *Seismic motion simulation based on Cassino Parallel Manipulator*, Journal of the Brazilian Society of Mechanical Sciences, pp. 1202-1207, 2002.
- [16] Codermatz R., Nicolich R., Slejko D., *Seismic risk assessment and GIS technology: applications to infrastructures in the Friuli-Venezia Giulia region (NE Italy)*, Earthquake Engineering and Structural Dynamics, pp. 1677-1690, 2003
- [17] Cornell C., Jalayer F., Hamburger R. O., Foutch D. A., *Probabilistic Basis for 2000 SAC Federal Emergency Management Agency Steel Moment Frame Guidelines*, Journal of structural Engineering, 128, pp. 526-533, 2002.
- [18] Cornell C. A., *Engineering seismic risk analysis*, Bulletin of the Seismological Society of America, 58, pp. 1583-1606, 1968.
- [19] Ellingwood B. R., Kinali K., *Quantifying and communicating uncertainty in seismic risk assessment*, Structural Safety, 31, pp. 179-187, 2009.
- [20] DesRoches R., Padgett J.E., *Methodology for the development of analytical fragility curves for retrofitted bridges*, Earthquake Engineering and Structural Dynamics, 37, pp. 1157-1174, 2008.

- [21] DesRoches R., Padgett J.E., *Bridge Functionality Relationships for Improved Seismic Risk Assessment of Transportation Networks*, Earthquake Spectra, vol. 23, No. 1, pp. 115-130, 2007.
- [22] DesRoches R., Choi E., Nielson B., *Seismic fragility of typical bridges in moderate seismic zones*, Engineering Structures, 26, pp. 187-199, 2004.
- [23] DesRoches R., Nielson B. G., *Analytical Seismic Fragility Curves for Typical Bridges in the Central and Southeastern United States*, Earthquake Spectra, Volume 23, No. 3, pages 615–633, 2007.
- [24] Franchin P., Noto F., *emphBRI.T.N.E.Y. BRIdge auTomatic NItha-based Earthquake fragilitY*, Dipartimento italiano della protezione civile, pp. 103-110, 2009.
- [25] Lupoi G., Franchin P., Lupoi A., Pinto P. E., *Seismic Fragility Analysis of Structural Systems*, Journal of Engineering Mechanics, pp. 385-395, 2006.
- [26] Monti G., Nisticò N., *Simple Probability-Based Assessment of Bridges under Scenario Earthquakes*, Journal of bridge Engineering, pp. 104-114, 2002.
- [27] Shinozuka M., Feng M. Q., Lee J., Naganuma T., *Statistical analysis of fragility curves*, Journal of Engineering Mechanics, pp. 1224-1231, 2000.
- [28] Shinozuka M., Feng M. Q., Kim H., Kim S., *Nonlinear static procedure for fraility curve development*, Journal of Engineering Mechanics, pp. 1287-1295, 2000.
- [29] Zanini M. A., Pellegrino C., Modena C., Morbin R., *Seismic vulnerabilty of bridges in transport networks subjected to environmental deterioration*, Bulletin of Earthquake Engineering, 11, 561-579, 2013.

Appendice A

APPENDICE DEGLI SCRIPT

Di seguito si riporta il codice di alcuni tra gli script più significativi collegati all'interfaccia.

Il primo, denominato *curvefrag.m* è il codice necessario per la gestione dell'interfaccia grafica, serve per lanciare tutte le function collegate ai pulsanti e per recepire i dati in input. Il secondo, chiamato *calcola.m*, è la porzione di programma che si occupa della creazione del modello tridimensionale del ponte, una volta che l'utente ha immesso i dati. L'ultimo codice riportato in appendice, *editorElements*, è uno fra i tanti script che si occupa di generare i file .tcl, nella fattispecie *Elements.tcl*, necessari per lanciare le analisi dinamiche tramite OpenSees.

```

function varargout = curvefrag(varargin)
%CURVEFRAG M-file for curvefrag.fig
%   CURVEFRAG, by itself, creates a new CURVEFRAG or raises the existing
%   singleton*.
%
%   H = CURVEFRAG returns the handle to a new CURVEFRAG or the handle to
%   the existing singleton*.
%
%   CURVEFRAG('Property','Value',...) creates a new CURVEFRAG using the
%   given property value pairs. Unrecognized properties are passed via
%   varargin to curvefrag_OpeningFcn. This calling syntax produces a
%   warning when there is an existing singleton*.
%
%   CURVEFRAG('CALLBACK') and CURVEFRAG('CALLBACK',hObject,...) call the
%   local function named CALLBACK in CURVEFRAG.M with the given input
%   arguments.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help curvefrag

% Last Modified by GUIDE v2.5 02-Apr-2014 14:27:06

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @curvefrag_OpeningFcn, ...
                  'gui_OutputFcn',  @curvefrag_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before curvefrag is made visible.
function curvefrag_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   unrecognized PropertyName/PropertyValue pairs from the
%            command line (see VARARGIN)

% Choose default command line output for curvefrag
handles.output = hObject;

```



```

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes curvefrag wait for user response (see UIRESUME)
% uiwait(handles.figure1);

set(handles.analisi,'Visible','off');
set(handles.frag,'Visible','off');
set(handles.text10,'Visible','off');
set(handles.text11,'Visible','off');
set(handles.edit_Hp,'Visible','off');
set(handles.text12,'Visible','off');
set(handles.text13,'Visible','off');
set(handles.text14,'Visible','off');
set(handles.edit_Ht,'Visible','off');
set(handles.text15,'Visible','off');
% --- Outputs from this function are returned to the command line.
function varargout = curvefrag_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

%spengo tutto all'inizio della gui
set(handles.text10,'Visible','off');
set(handles.text31,'Visible','off');
set(handles.text5,'Visible','off');
set(handles.text6,'Visible','off');
set(handles.text7,'Visible','off');
set(handles.text44,'Visible','off');
set(handles.edit_Ls,'Visible','off');
set(handles.edit_nt,'Visible','off');
set(handles.text11,'Visible','off');
set(handles.edit_Hp,'Visible','off');
set(handles.text12,'Visible','off');
set(handles.text14,'Visible','off');
set(handles.edit_Ht,'Visible','off');
set(handles.text15,'Visible','off');
set(handles.text45,'Visible','off');
set(handles.text46,'Visible','off');
set(handles.text47,'Visible','off');
set(handles.edit_L,'Visible','off');
set(handles.text10,'Visible','off');
set(handles.text11,'Visible','off');
set(handles.edit_Hp,'Visible','off');
set(handles.text12,'Visible','off');
set(handles.text13,'Visible','off');
set(handles.text14,'Visible','off');
set(handles.edit_Ht,'Visible','off');
set(handles.text15,'Visible','off');
set(handles.text22,'Visible','off');
set(handles.text23,'Visible','off');
set(handles.text24,'Visible','off');

```

```
set(handles.text26,'Visible','off');
set(handles.text27,'Visible','off');
set(handles.text30,'Visible','off');
set(handles.edit_Ld,'Visible','off');
set(handles.edit_nb,'Visible','off');
set(handles.edit_Ht,'Visible','off');
```

```
function edit_Ls_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to edit_Ls (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_Ls as text
%        str2double(get(hObject,'String')) returns contents of edit_Ls as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function edit_Ls_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to edit_Ls (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
% See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit_Hp_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to edit_Hp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit_Hp as text
```

```
%        str2double(get(hObject,'String')) returns contents of edit_Hp as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function edit_Hp_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to edit_Hp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
% See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit_Ht_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to edit_Ht (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```

% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_Ht as text
%         str2double(get(hObject,'String')) returns contents of edit_Ht as a double

% --- Executes during object creation, after setting all properties.
function edit_Ht_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit_Ht (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton_reset.
function pushbutton_reset_Callback(hObject, eventdata, handles)

set(handles.edit_Ls,'String','0');
set(handles.edit_Ld,'String','0');
set(handles.edit_nb,'String','0');
set(handles.edit_nt,'String','0');
set(handles.edit_Hp,'String','0');
set(handles.edit_Ht,'String','0');
set(handles.edit_L,'String','0');
set(handles.edit_contatore,'String','1');

set(handles.war_Ls,'String','');
set(handles.war_Ld,'String','');
set(handles.war_nb,'String','');
set(handles.war_nt,'String','');
set(handles.war_Hp,'String','');
set(handles.war_Ht,'String','');
set(handles.war_L,'String','');
a=1;
plot(a);
set(handles.text10,'Visible','off');
set(handles.text11,'Visible','off');
set(handles.edit_Hp,'Visible','off');
set(handles.text12,'Visible','off');
set(handles.text13,'Visible','off');
set(handles.text14,'Visible','off');
set(handles.edit_Ht,'Visible','off');
set(handles.text15,'Visible','off');

set(handles.schemapile,'Value',1);

```

```
function edit_Ld_Callback(hObject, eventdata, handles)
% hObject    handle to edit_Ld (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_Ld as text
%        str2double(get(hObject,'String')) returns contents of edit_Ld as a double

% --- Executes during object creation, after setting all properties.
function edit_Ld_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_Ld (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit_nb_Callback(hObject, eventdata, handles)
% hObject    handle to edit_nb (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_nb as text
%        str2double(get(hObject,'String')) returns contents of edit_nb as a double

% --- Executes during object creation, after setting all properties.
function edit_nb_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_nb (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit_nt_Callback(hObject, eventdata, handles)
% hObject    handle to edit_nt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_nt as text
%        str2double(get(hObject,'String')) returns contents of edit_nt as a double

% --- Executes during object creation, after setting all properties.
```

```

function edit_nt_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_nt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in schemapile.
function schemapile_Callback(hObject, eventdata, handles)
% hObject    handle to schemapile (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.valore = get(handles.schemapile,'Value');

switch handles.valore
case 1
    %l'utente ha scelto la pila a telaio devo disporre visibile tutto il
    %pannello per questa tipologia
    %azzerò i warning
    set(handles.war_Ls,'String','');
set(handles.war_Ld,'String','');
set(handles.war_nb,'String','');
set(handles.war_nt,'String','');
set(handles.war_Hp,'String','');
set(handles.war_Ht,'String','');
set(handles.war_L,'String','');
    %nascondo ciò che non interessa
set(handles.war_L,'Visible','off');
set(handles.text45,'Visible','off');
set(handles.text46,'Visible','off');
set(handles.text47,'Visible','off');
set(handles.edit_L,'Visible','off');
set(handles.text10,'Visible','off');
set(handles.text11,'Visible','off');
set(handles.edit_Hp,'Visible','off');
set(handles.text12,'Visible','off');
set(handles.text13,'Visible','off');
set(handles.text14,'Visible','off');
set(handles.edit_Ht,'Visible','off');
set(handles.text15,'Visible','off');
    %accendo quello che mi serve
set(handles.text30,'Visible','on');
set(handles.text22,'Visible','on');
set(handles.text23,'Visible','on');
set(handles.text24,'Visible','on');
set(handles.text26,'Visible','on');
set(handles.text27,'Visible','on');
set(handles.edit_Ld,'Visible','on');
set(handles.edit_nb,'Visible','on');
set(handles.war_Ld,'Visible','on');
set(handles.war_nb,'Visible','on');

```

```
set(handles.text31,'Visible','on');
set(handles.text5,'Visible','on');
set(handles.text6,'Visible','on');
set(handles.text7,'Visible','on');
set(handles.text44,'Visible','on');
set(handles.edit_Ls,'Visible','on');
set(handles.edit_nt,'Visible','on');
```

%ora rinomino i titoli dei testi sulla base di ciò che mi serve

```
set(handles.text14,'String','Ht=');
set(handles.text5,'String','lunghezza campata');
set(handles.text44,'String','m');
set(handles.text6,'String','Ls=');
set(handles.text31,'String','nt=');
set(handles.text7,'String','m');
set(handles.text13,'String','altezza traverso');
%sposto i campi in una nuova posizione
set(handles.text44,'Position',[27.8 16.6 5.2 1.077]);
set(handles.edit_nt,'Position',[9.6 13.154 16.6 1.231]);
set(handles.text31,'Position',[4.8 12.15 4.6 1.923]);
set(handles.war_nt,'Position',[32.2 16.769 11.4 1.462]);
guidata (handles.figure1,handles);
```

case 2

```
%l'utente ha scelto la pila a fusto devo rendere visibile solo il
%azzerò i warning
set(handles.war_Ls,'String','');
set(handles.war_Ld,'String','');
set(handles.war_nb,'String','');
set(handles.war_nt,'String','');
set(handles.war_Hp,'String','');
set(handles.war_Ht,'String','');
set(handles.war_L,'String','');
%pannello per questo tipo
set(handles.text22,'Visible','off');
set(handles.text23,'Visible','off');
set(handles.text24,'Visible','off');
set(handles.text26,'Visible','off');
set(handles.text27,'Visible','off');
set(handles.text30,'Visible','off');
set(handles.edit_Ld,'Visible','off');
set(handles.edit_nb,'Visible','off');
set(handles.edit_Ht,'Visible','off');
set(handles.war_Ld,'Visible','off');
set(handles.war_nb,'Visible','off');
set(handles.war_Ht,'Visible','off');
%rendo visibile ciò che mi interessa
set(handles.war_L,'Visible','on');
set(handles.text31,'Visible','on');
set(handles.text5,'Visible','on');
set(handles.text6,'Visible','on');
set(handles.text7,'Visible','on');
set(handles.text44,'Visible','on');
set(handles.text47,'Visible','on');
set(handles.edit_Ls,'Visible','on');
set(handles.edit_nt,'Visible','on');
set(handles.edit_L,'Visible','on');
```

```

set(handles.text45,'Visible','on');
set(handles.text46,'Visible','on');

set(handles.war_Ht,'Visible','on');
set(handles.war_Hp,'Visible','on');
%ora rinomino i titoli dei testi sulla base di ciò che mi serve
set(handles.text14,'String','H=');
set(handles.text5,'String','caratteristiche beam equivalente');
set(handles.text6,'String','E=');
set(handles.text44,'String','m^4');
set(handles.text31,'String','J=');
set(handles.text7,'String','N/mm^2');
set(handles.text13,'String','offset impalacto/pile');
%sposto i campi in una nuova posizione
set(handles.text44,'Position',[27.8 19 5.2 1.077]);
set(handles.edit_nt,'Position',[9.6 15.354 16.6 1.231]);
set(handles.text31,'Position',[4.8 14.5 4.6 1.923]);
set(handles.war_nt,'Position',[32.2 18.7 11.4 1.462]);
guidata(handles.figure1,handles);
end;

% Hints: contents = cellstr(get(hObject,'String')) returns schemapile contents as cell array
%         contents{get(hObject,'Value')} returns selected item from schemapile

% --- Executes during object creation, after setting all properties.
function schemapile_CreateFcn(hObject, eventdata, handles)
% hObject    handle to schemapile (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in edit_invia.
function edit_invia_Callback(hObject, eventdata, handles)
% hObject    handle to edit_invia (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
camp_s=get(handles.edit_contatore,'String');
camp=str2num(camp_s);

set(handles.text10,'Visible','off');
set(handles.text11,'Visible','off');
set(handles.edit_Hp,'Visible','off');
set(handles.text12,'Visible','off');
set(handles.text13,'Visible','off');
set(handles.text14,'Visible','off');
set(handles.edit_Ht,'Visible','off');
set(handles.text15,'Visible','off');

```

```
[handles]=calcola(handles);
```

```
function edit_L_Callback(hObject, eventdata, handles)
% hObject    handle to edit_L (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_L as text
%        str2double(get(hObject,'String')) returns contents of edit_L as a double

% --- Executes during object creation, after setting all properties.
function edit_L_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_L (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in prosegui.
function prosegui_Callback(hObject, eventdata, handles)
% hObject    handle to prosegui (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%azzerare i warning
schema;
set(handles.warning,'String','');
settings = settingsdlg(...
    'separator' , ' Dati degli elementi strutturali della campata',...
    { '      modulo di elasticità E calcestruzzo [N/mm²]', 'E'},0,...
    { '      modulo di elasticità tangenziale G calcestruzzo[N/mm²]', 'G'},0,...
    'separator' , '      traverso in campata',...
    { '      area sezione [mm²]', 'AreaDellaSezioneDelTraversoInCampata'},0,...
    { '      Ix [*10^10][mm^4]', 'Ixtrav'},0,...
    { '      Iy [*10^10][mm^4]', 'Iytrav'},0,...
    'separator' , '      trave longitudinale in campata',...
    { '      area sezione [mm²]', 'AreaDellaSezioneDellaTraveLongitudinale'},0,...
    { '      Ix [*10^10][mm^4]', 'Ixlong'},0,...
    { '      Iy [*10^10][mm^4]', 'Iylong'},0,...
    'separator' , ' Dati geometrici del traverso superiore delle pile [mm]',...
    { '      Lb', 'LarghezzaBaseTraverso'},0,...
    { '      La', 'LarghezzaAnimaTraverso'},0,...
    { '      Hb', 'AltezzaBaseDelTraverso'},0,...
    { '      Ha', 'AltezzaAnimaDelTraverso'},0,...
    'separator', 'Dati armatura del traverso superiore delle pile [mm]',...
    '      numero barre armatura superiore',0,...
    '      diametro barre armatura superiore',0,...
    '      numero barre armatura inferiore',0,...
    '      diametro barre armatura inferiore',0,...
```



```

'        numero barre armatura minima',0,...
'        diametro barre armatura minima',0,...
{'        copriferro','copriferrotrav'},0,...
'separator' , ' Dati geometrici delle pile [mm]',...
'        diametro esterno pila',0,...
'        numero barre armatura longitudinale',0,...
'        diametro barre armatura',0,...
'        copriferro',0,...
'separator' , ' Caratteristiche meccaniche dei materiali',...
'        Classe di calcestruzzo' ,{'C16/20','C20/25','C25/30','C28/35','C32/40'},...
'        Classe di acciaio' ,{'FeB32k fy=315 MPa','FeB38k fy=375 MPa','FeB44k fy=430 MPa'
},...
'Description','inserire le dimensioni degli elementi strutturali del ponte',...
'WindowWidth', 500,...
'title','dati geometrici elementi strutturali');
%faccio un controllo su quello che mi ha dato in input l'utente
if settings.E==0
    set(handles.warning,'String','modulo di elasticità E nullo');

end;
if settings.G==0
    set(handles.warning,'String','modulo di elasticità G nullo');

end;
if settings.AreaDellaSezioneDelTraversoInCampata==0
    set(handles.warning,'String','area sezione traverso in campata nulla');

end;
if settings.Ixtrav==0
    set(handles.warning,'String','momento Ix nullo');

end;
if settings.Iytrav==0
    set(handles.warning,'String','momento Iy nullo');

end;
if settings.AreaDellaSezioneDellaTraveLongitudinale==0
    set(handles.warning,'String','area sezione trave da ponte nulla');

end;
if settings.Ixlong==0
    set(handles.warning,'String','momento Ix nullo');

end;
if settings.Iylong==0
    set(handles.warning,'String','momento Iy nullo');

end;

if settings.LarghezzaBaseTraverso==0
    set(handles.warning,'String','larghezza base traverso nulla');

end;
if settings.LarghezzaAnimaTraverso==0
    set(handles.warning,'String','larghezza anima traverso nulla');

end;
if settings.AltezzaBaseDelTraverso==0

```

```
set(handles.warning,'String','altezza base del traverso nulla');

end;
if settings.AltezzaAnimaDelTraverso==0
    set(handles.warning,'String','altezza anima traverso nulla');

end;
if settings.Copriferro==0
    set(handles.warning,'String','Copriferro del traverso nullo');

end;
if settings.DiametroEsternoPila==0
    set(handles.warning,'String','diametro esterno pila nullo');

end;
if settings.DiametroBarreArmatura==0
    set(handles.warning,'String','diametro barre armatura nullo');

end;
if settings.NumeroBarreArmaturaLongitudinale==0
    set(handles.warning,'String','numero barre armatura nullo');

end;
if settings.Copriferro>(0.4*(settings.DiametroEsternoPila))
    set(handles.warning,'String','Copriferro troppo elevato rispetto al diametro');

end;
if settings.Copriferro==0
    set(handles.warning,'String','Copriferro della pila nullo');

end;
if settings.NumeroBarreArmaturaSuperiore==0
    set(handles.warning,'String','Numero barre armatura superiore nullo');

end;
if settings.DiametroBarreArmaturaSuperiore==0
    set(handles.warning,'String','Diametro barre armatura superiore nullo');

end;
if settings.NumeroBarreArmaturaInferiore==0
    set(handles.warning,'String','Numero barre armatura inferiore nullo');

end;
if settings.DiametroBarreArmaturaInferiore==0
    set(handles.warning,'String','Diametro barre armatura inferiore nullo');

end;
if settings.NumeroBarreArmaturaMinima==0
    set(handles.warning,'String','Numero barre armatura minima nullo');

end;
if settings.DiametroBarreArmaturaMinima==0
    set(handles.warning,'String','Diametro barre armatura minima nullo');

end;

%multiplico per 10 alla 10 i momenti d'inerzia
```

```

Iextrav=(settings.Iextrav)*10000000000;
Iytrav=(settings.Iytrav)*10000000000;
Ixlong=(settings.Ixlong)*10000000000;
Iylong=(settings.Iylong)*10000000000;

```

```
%salvo tutti gli input dell'utente
```

```

handles.E=settings.E;
handles.G=settings.G;
handles.Iextrav=Iextrav;
handles.Iytrav=Iytrav;
handles.Ixlong=Ixlong;
handles.Iylong=Iylong;
handles.copriferrotrav=settings.copriferrotrav;
handles.classe=settings.ClasseDiCalcestruzzo;
handles.classeacc=settings.ClasseDiAcciaio;
handles.diametropila=settings.DiametroEsternoPila;
handles.diametrobarre=settings.DiametroBarreArmatura;
handles.copriferro=settings.Copriferro;
handles.numerobarre=settings.NumeroBarreArmaturaLongitudinale;
handles.larghezzabasetraverso=settings.LarghezzaBaseTraverso;
handles.larghezzaanimatraverso=settings.LarghezzaAnimaTraverso;
handles.altezzabasedeltraverso=settings.AltezzaBaseDelTraverso;
handles.altezzaanimadeltraverso=settings.AltezzaAnimaDelTraverso;
handles.Atraverso=settings.AreaDellaSezioneDelTraversoInCampata;
handles.Atrave=settings.AreaDellaSezioneDellaTraveLongitudinale;
handles.numbarsup=settings.NumeroBarreArmaturaSuperiore;
handles.diabarsup=settings.DiametroBarreArmaturaSuperiore;
handles.numbarinf=settings.NumeroBarreArmaturaInferiore;
handles.diabarinf=settings.DiametroBarreArmaturaInferiore;
handles.numbarmin=settings.NumeroBarreArmaturaMinima;
handles.diabarmin=settings.DiametroBarreArmaturaMinima;

```

```
[handles]=salva(handles);
```

```
% --- Executes on button press in scrivi.
```

```
function scrivi_Callback(hObject, eventdata, handles)
```

```
% hObject handle to scrivi (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
%creo le cartelle dei 15 ponti campione dentro le cartelle delle terne
```

```

mkdir('TERNAl','PC1');
mkdir('TERNAl','PC2');
mkdir('TERNAl','PC3');
mkdir('TERNAl','PC4');
mkdir('TERNAl','PC5');
mkdir('TERNAl','PC6');
mkdir('TERNAl','PC7');
mkdir('TERNAl','PC8');
mkdir('TERNAl','PC9');
mkdir('TERNAl','PC10');
mkdir('TERNAl','PC11');

```

```
mkdir('TERNA1','PC12');
mkdir('TERNA1','PC13');
mkdir('TERNA1','PC14');
mkdir('TERNA1','PC15');
```

```
mkdir('TERNA2','PC1');
mkdir('TERNA2','PC2');
mkdir('TERNA2','PC3');
mkdir('TERNA2','PC4');
mkdir('TERNA2','PC5');
mkdir('TERNA2','PC6');
mkdir('TERNA2','PC7');
mkdir('TERNA2','PC8');
mkdir('TERNA2','PC9');
mkdir('TERNA2','PC10');
mkdir('TERNA2','PC11');
mkdir('TERNA2','PC12');
mkdir('TERNA2','PC13');
mkdir('TERNA2','PC14');
mkdir('TERNA2','PC15');
```

```
mkdir('TERNA3','PC1');
mkdir('TERNA3','PC2');
mkdir('TERNA3','PC3');
mkdir('TERNA3','PC4');
mkdir('TERNA3','PC5');
mkdir('TERNA3','PC6');
mkdir('TERNA3','PC7');
mkdir('TERNA3','PC8');
mkdir('TERNA3','PC9');
mkdir('TERNA3','PC10');
mkdir('TERNA3','PC11');
mkdir('TERNA3','PC12');
mkdir('TERNA3','PC13');
mkdir('TERNA3','PC14');
mkdir('TERNA3','PC15');
```

```
%lancio in ordine tutti gli editor
[handles]=editorNodeCoord(handles);
editorSPconstraint(handles);
editorMPconstraint(handles);
[handles]=editorElements(handles);
editorSections(handles);
[handles]=editorNodeMass(handles);
editorLoadPattern(handles);
editorAnalysisRecorder;
editorGeoTran(handles);
editorfilecomuni;
```

```
%rendo visibile il bottone per lanciare l'analisi
set(handles.analisi,'Visible','on');
```

```
%creo uno switchcase che sulla base della classe di calcestruzzo scelta
%scriva i 15 opportui tcl dei materiali
classe=handles.classe;
switch classe
    case 'C16/20'
```

```
[handles]=editorMaterialsC1620(handles);  
case 'C20/25'  
[handles]=editorMaterialsC2025(handles);  
case 'C25/30'  
[handles]=editorMaterialsC2530(handles);  
case 'C28/35'  
[handles]=editorMaterialsC2835(handles);  
case 'C32/40'  
[handles]=editorMaterialsC3240(handles);
```

end

%creo uno switchcase che sulla base della classe di acciaio scelta
%vada a completare i 15 tcl dei materiali precedentemente iniziati
classeacc=handles.classeacc;

switch classeacc

```
case 'FeB32k fy=315 MPa'  
    editorMaterialsFeB32k(handles);  
case 'FeB38k fy=375 MPa'  
    editorMaterialsFeB38k(handles);  
case 'FeB44k fy=430 MPa'  
    editorMaterialsFeB44k(handles);
```

end

% --- Executes on button press in analisi.

function analisi_Callback(hObject, eventdata, handles)

% hObject handle to analisi (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

```
set(handles.analisi,'Visible','off');
```

```
set(handles.scrivi,'Visible','off');
```

```
set(handles.proseguì,'Visible','off');
```

```
set(handles.frag,'Visible','on');
```

```
!Opensees.exe main.tcl
```

% --- Executes on button press in frag.

function frag_Callback(hObject, eventdata, handles)

% hObject handle to frag (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

```
curvefraglong(handles);
```

```
curvefragtrasv(handles);
```

```
function(handles)= calcola(handles)
switch handles.valore
case 1

%azzerò i warning

set(handles.war_Ls,'String','');
set(handles.war_Ld,'String','');
set(handles.war_nb,'String','');
set(handles.war_nt,'String','');
set(handles.war_Hp,'String','');
set(handles.war_Ht,'String','');

%salvo i dati inseriti nelle edit_text nella struttura (dati)

Ls_s=get(handles.edit_Ls,'String');
Ld_s=get(handles.edit_Ld,'String');
nb_s=get(handles.edit_nb,'String');
nt_s=get(handles.edit_nt,'String');
Hp_s=get(handles.edit_Hp,'String');
Ht_s=get(handles.edit_Ht,'String');
camp_s=get(handles.edit_contatore,'String');

%convertò le stringhe in numeri

Ls=str2num(Ls_s);
Ld=str2num(Ld_s);
nb=str2num(nb_s);
nt=str2num(nt_s);
Hp=str2num(Hp_s);
Ht=str2num(Ht_s);
camp=str2num(camp_s);

%convertò i metri in mm
Ls=Ls*1000;
Ld=Ld*1000;

Hp=Hp*1000;
Ht=Ht*1000;

%se Ld=0 la funzione si arresta
if Ld==0
    set(handles.war_Ld,'String','Ld >0!');
return
end
%se nb=0 la funzione si arresta
if nb<=1
    set(handles.war_nb,'String','nb >2!');
return
end
%se Ls=0 la funzione si arresta
if Ls==0
    set(handles.war_Ls,'String','Ls>0!!');
return
```

```

end
%se nt=0 la funzione si arresta
if nt==0
    set(handles.war_nt,'String','nt >0!');
return
end
%visualizzo l'oggetto axes3
%accendo i campi per l'inserimento dei dati riguardanti la pila
set(handles.axes3,'Visible','on');
set(handles.text10,'Visible','on');
set(handles.text11,'Visible','on');
set(handles.edit_Hp,'Visible','on');
set(handles.text12,'Visible','on');
set(handles.text13,'Visible','on');
set(handles.text14,'Visible','on');
set(handles.edit_Ht,'Visible','on');
set(handles.text15,'Visible','on');

if camp~=1,
%se Hp=0 la funzione si arresta
if Hp==0
    set(handles.war_Hp,'String','Hp>0!');
return
end
%se Ht=0 la funzione si arresta
if Ht==0
    set(handles.war_Ht,'String','Ht>0!');
return
end
end;

%disegno il ponte
%----VOGLIO COSTRUIRE 3 MATRICI DI PUNTI (X Y Z), OGNI MATRICE CONTIENE LE
%COORDINATE X O Y O Z DELLA COPPIA DI PUNTI DEL SEGMENTO CONSIDERATO'-----

%---stabilisco tre indici che mi danno il numero di punti complessivo nel
%piano dell'impalcato che sto considerando----

i=Ls/(nt+1);
j=Ld/(nb-1);
k=nt+2;

%----MATRICE X E Y DELL'IMPALACATO fibre longitudinali-----
% questo gira solo alla prima iterazione-----
if camp==1
l=1;
xl= zeros(k,nb);
yl= zeros(k,nb);

%--comincio a compilare le matrici x e y

for l=1:k,
    xl(l,:)=0+(l-1)*i;
end;
yl= zeros(l,nb);

```

```

for m=1:nb,
    yl(:,m)=0+(m-1)*j;
end;

end

%questo gira se non siamo alla prima iterazione-----
if camp~=1

% apro i salvataggi
xl=handles.xl;
yl=handles.yl;
zl=handles.zl;
l=handles.l;

%----MATRICE X E Y DELL'IMPALACATO fibre longitudinali-----
%--comincio a compilare le matrici x e y
l=l+1;
for l=1:(l+k-2),
    xl(l,:)=xl((l-1),1)+i;
end;

yl= zeros(l,nb);
for m=1:nb,
    yl(:,m)=0+(m-1)*j;
end;

end
%-----matrice Z dell'impalcato, ogni punto dell'impalcato sta alla stessa quota zero--
zl= zeros(l,nb);

%----MATRICE X E Y DELL'IMPALACATO fibre trasversali-----

if camp==1
    t=1;

% questo gira solo alla prima iterazione-----
xt= zeros(nb,k);
yt= zeros(nb,k);

%--comincio a compilare le matrici x e y

for t=t:k,
    xt(:,t)=0+(t-1)*i;
end;
yt= zeros(nb,t);
for m=1:nb,
    yt(m,:)=0+(m-1)*j;
end;

end

```



```

%questo gira se non siamo alla prima iterazione-----
if camp~=1
% apro i salvataggi
xt=handles.xt;
yt=handles.yt;
zt=handles.zt;
t=handles.t;

%----MATRICE X E Y DELL'IMPALCATO fibre trasversali-----
%--comincio a compilare le matrici x e y
t=t+1;
for t=t:(t+k-2),
    xt(:,t)=xt(1,(t-1))+i;
end;

yt= zeros(nb,t);
for m=1:nb,
    yt(m,:)=0+(m-1)*j;
end;

end

%-----matrice Z dell'impalcato, ogni punto dell'impalcato sta alla stessa quota zero--
zt= zeros(nb,t);

%----MATRICE X E Y E Z DELLE PILE-----
if camp==1,
    xp=zeros(2,2);
    yp=zeros(2,2);
    zp=zeros(2,2);
    Ls_old=Ls;
    p=0;
    handles.xp=xp;
    handles.yp=yp;
    handles.zp=zp;
    handles.p=p;
    handles.Ls_old=Ls_old;
end;

%comincio a ostruire le pile partendo dalla seconda campata-----
if camp~=1
    xp=handles.xp;
    yp=handles.yp;
    zp=handles.zp;
    p=handles.p;

    Ls_old=handles.Ls_old;

%--comincio a compilare le matrici x e y e z
p=p+1;
yp(:,p)=0;
yp(:,p+1)=Ld;

if camp==2,

    for p=p:(p+1),

```

```

xp(:,p)=Ls_old;
zp(1,p)=-Ht-Hp;
zp(2,p)=-Ht;

```

```
end;
```

```
end;
```

```

if camp~=2,
    val=xp(1,(p-1));
    for p=p:(p+1),
        xp(:,p)=val+Ls_old;
        zp(1,p)=-Ht-Hp;
        zp(2,p)=-Ht;
    end;
end;

```

```
end;
```

```
Ls_old=Ls;
```

```
end;
```

```

handles.p=p;
handles.Ls_old=Ls_old;

```

```

%----plotto le tre terne di matrici di punti-----
plot3(xl,yl,zl,'-ro',xt,yt,zt,'-bo',xp,yp,zp,'-yo');
axis equal;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%CONTATORE PER SAPERE A CHE CAMPATA MI STO RIFERENDO%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

if camp>2,
    recordpile=handles.recordpile;
end;
if camp~=1,
    recordnt=handles.recordnt;
    recordLs=handles.recordLs;
    recordpile(camp-1)=Hp;
    handles.recordpile=recordpile;
    %save recordpile.mat recordpile
end;

```

```

recordLs(camp)=Ls;
recordnt(camp)=nt;
handles.recordnt=recordnt;
handles.recordLs=recordLs;

```

```

camp=camp+1;
camp_s=num2str(camp);
set(handles.edit_contatore,'String',camp_s);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%salvo le matrici dei punti e i contatori%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

handles.xl=xl;
handles.yl=yl;
handles.zl=zl;
handles.xt=xt;
handles.yt=yt;
handles.zt=zt;
handles.xp=xp;
handles.yp=yp;
handles.zp=zp;
handles.l=l;
handles.t=t;
handles.p=p;
handles.Ls_old=Ls_old;
handles.camp=camp;
handles.nb=nb;
guidata (handles.figure1,handles);

```

```

%esporto i dati in un file di testo txt
%save xt.mat xt
%save yt.mat yt
%save zt.mat zt
%save xl.mat xl
%save yl.mat yl
%save zl.mat zl
%save xp.mat xp
%save yp.mat yp
%save zp.mat zp
%save camp.mat camp

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% CASE 2 %

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

case 2

```

```

%%disegna il ponte con pile a fusto unico
%azzerò i warning

```

```
set(handles.war_Ls,'String','');
set(handles.war_nt,'String','');
set(handles.war_Hp,'String','');
set(handles.war_Ht,'String','');
set(handles.war_L,'String','');

%salvo i dati inseriti nelle edit_text nella struttura (dati)

E_s=get(handles.edit_Ls,'String');
J_s=get(handles.edit_nt,'String');
Hp_s=get(handles.edit_Hp,'String');
Ht_s=get(handles.edit_Ht,'String');
L_s=get(handles.edit_L,'String');
camp_s=get(handles.edit_contatore,'String');

%converto le stringhe in numeri
E=str2num(E_s);
J=str2num(J_s);
Hp=str2num(Hp_s);
Ht=str2num(Ht_s);
L=str2num(L_s);
camp=str2num(camp_s);

%se E=0 la funzione si arresta
if E==0
    set(handles.war_Ls,'String','E>0!');
return
end
%se J=0 la funzione si arresta
if J==0
    set(handles.war_nt,'String','J>0!');
return
end
%se L=0 la funzione si arresta
if L==0
    set(handles.war_L,'String','L>0!!');
return
end

L=L*1000;

Hp=Hp*1000;
Ht=Ht*1000;

%visualizzo l'oggetto axes3
%accendo i campi per l'inserimento dei dati riguardanti la pila
set(handles.axes3,'Visible','on');
set(handles.text10,'Visible','on');
set(handles.text11,'Visible','on');
set(handles.edit_Hp,'Visible','on');
set(handles.text12,'Visible','on');
set(handles.text13,'Visible','on');
set(handles.text14,'Visible','on');
set(handles.edit_Ht,'Visible','on');
set(handles.text15,'Visible','on');
if camp~=1,
```

```

%se Hp=0 la funzione si arresta
if Hp==0
    set(handles.war_Hp,'String','Hp>0!');
return
end
%se H=0 la funzione si arresta
if Ht==0
    set(handles.war_Ht,'String','H>0!');
return
end
end;

```

```

%disegno il ponte
%----VOGLIO COSTRUIRE 3 MATRICI DI PUNTI (X Y Z), OGNI MATRICE CONTIENE LE
%COORDINATE X O Y O Z DELLA COPPIA DI PUNTI DEL SEGMENTO CONSIDERATO'-----

```

```

%----MATRICE X E Y DELL'IMPALACATO fibre longitudinali-----
% questo gira solo alla prima iterazione-----
if camp==1
    l=1;
    xl= zeros(2,1);
    yl= zeros(2,1);

    %--comincio a compilare le matrici x e y

    for l=1:2,
        xl(l,1)=0+(l-1)*L;
    end;
    %-----matrice Z dell'impalcato, ogni punto dell'impalcato sta alla stessa quota zero--
    zl= zeros(1,1);

end

```

```

%questo gira se non siamo alla prima iterazione-----
if camp~=1

```

```

    % apro i salvataggi
    xl=handles.xl;
    yl=handles.yl;
    zl=handles.zl;
    l=handles.l;

```

```

%----MATRICE X E Y DELL'IMPALACATO fibre longitudinali-----
%--comincio a compilare le matrici x e y
l=l+1;
xl(l,1)=xl((l-1),1)+L;
yl= zeros(1,1);

```

```

%-----matrice Z dell'impalcato, ogni punto dell'impalcato sta alla stessa quota zero--

```

```

z1= zeros(1,1);
end

%----MATRICE X E Y E Z DELLE PILE-----
if camp==1,
    xp=zeros(2,1);
    yp=zeros(2,1);
    zp=zeros(2,1);
    L_old=L;
    p=0;
    handles.xp=xp;
    handles.yp=yp;
    handles.zp=zp;
    handles.p=p;
    handles.L_old=L_old;
end;
%comincio a ostruire le pile partendo dalla seconda campata-----
if camp~=1
    xp=handles.xp;
    yp=handles.yp;
    zp=handles.zp;
    p=handles.p;

    L_old=handles.L_old;

%--comincio a compilare le matrici x e y e z
p=p+1;
yp=zeros(2,(camp-1));

if camp==2
    xp(1,(camp-1))=L_old;
    xp(2,(camp-1))=L_old;
end
if camp~=2
    xp(1,(camp-1))=xp(1,(camp-2))+L_old;
    xp(2,(camp-1))=xp(2,(camp-2))+L_old;
end
zp(1,(camp-1))=-Ht;
zp(2,(camp-1))=-Ht-Hp;

L_old=L;

handles.p=p;
handles.L_old=L_old;
end

%----plotto le tre terne di matrici di punti-----
plot3(xl,y1,z1,'-ro',xp,yp,zp,'-yo');
axis equal;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%CONTATORE PER SAPERE A CHE CAMPATA MI STO RIFERENDO%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

camp=camp+1;
camp_s=num2str(camp);

```

```
set(handles.edit_contatore,'String',camp_s);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%salvo le matrici dei punti e i contatori%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
handles.xl=xl;  
handles.yl=yl;  
handles.zl=zl;  
handles.xp=xp;  
handles.yp=yp;  
handles.zp=zp;  
handles.l=l;  
handles.p=p;  
handles.L_old=L_old;  
handles.camp=camp;
```

```
guidata (handles.figure1,handles);
```

```
end
```

```
function(handles)= editorElements(handles)
```

```
%richiamo i file salvati che mi servono
```

```
recordnt=handles.recordnt;
recordpile=handles.recordpile;
lunghpile=recordpile;
camp=handles.camp;
nb=handles.nb;
```

```
Ld_s=get(handles.edit_Ld,'String');
Ld=str2num(Ld_s);
```

```
largbasetrav=handles.larghezzabasetravverso;
larganimatrav=handles.larghezzaanimatravverso;
altezzabatrav=handles.altezzabasedeltraverso;
altezzaanimatrav=handles.altezzaanimadeltraverso;
Atraverso=handles.Atraverso;
Atrave=handles.Atrave;
diametropila=handles.diametropila;
```

```
fileID = fopen('Elements.tcl','w');
fprintf(fileID,'%6s\n','# Elements.tcl');
fprintf(fileID,'\n');
```

```
%faccio prima tutte le pile
%stampo l'element per ogni coppia di pile
%calcolo la massa di ogni coppia di pile
```

```
z=length(recordpile);
area((((diametropila/1000)/2)^2)*3.14;
for i=1:z,
    recordpile(i)=recordpile(i)*area*(2.5);
end;
```

```
eletag=1;
i=1;
j=2;
```

```
sectag=2;
geotrantag=1;
z=1;
NIP=3;
g=1;
for k=1:(camp-2)
```

```
    fprintf(fileID,'%5s %9s %8s %8s %6s %9s %13s %7s %11s %6s\n','# Element "Pila":','eleTag'
    , 'NodeI', 'NodeJ', 'NIP', 'secTag', 'GeoTranTag', 'Mass', 'MaxIters', 'Tol');
    fprintf(fileID,'%5s %7i %7i %7i %5i %5i %5i %6s %+14.6E %11s %+15.6E\n','element
    nonlinearBeamColumn',eletag,i,j,NIP,sectag,geotrantag,'-mass', recordpile(z),'-iter    20'
    ,0.001 );
    fprintf(fileID,'\n');
    loadtagpile(g)=eletag;
    i=i+2;
```



```

j=j+2;
g=g+1;
eletag=eletag+1;
fprintf(fileID,'%5s %9s %8s %8s %6s %9s %13s %7s %11s %6s\n','# Element "Pila":','eleTag'
,'NodeI','NodeJ','NIP','secTag','GeoTranTag','Mass','MaxIters','Tol');
fprintf(fileID,'%5s %7i %7i %7i %5i %5i %5i %6s %+14.6E %11s %+15.6E\n','element
nonlinearBeamColumn',eletag,i,j,NIP,sectag,geotrantag,'-mass', recordpile(z),'-iter 20'
,0.001 );
fprintf(fileID,'\n');
i=i+2;
j=j+2;
z=z+1;
loadtagpile(g)=eletag;
eletag=eletag+1;
g=g+1;
end;

```

%ora faccio i traverso fibre (è il top pila)

% calcolo la massa del top pila che è sempre uguale

%calcolo area traverso

area=(largbasetrav*altezzabatrav)+(altezzaanimatrav*larganimatrav);

%massa traverso

mass=area*Ld*(0.0000025);

sectag=4;

i=0;

cont=0;

g=1;

geotrantag=geotrantag+1;

for k=1:(camp-2)

for n=0:(nb-2)

i=cont+(recordnt(k))*nb +1+n;

if k==1

i=4*(camp-2) + (1+recordnt(1))*nb +1+n;

end

j=i+1;

fprintf(fileID,'%5s %9s %8s %8s %6s %9s %13s %7s %11s %6s\n','# Element
"TraversoFibre":','eleTag','NodeI','NodeJ','NIP','secTag','GeoTranTag','Mass',
'MaxIters','Tol');

fprintf(fileID,'%5s %7i %7i %7i %5i %5i %5i %6s %+14.6E %11s %+15.6E\n','element
nonlinearBeamColumn',eletag,i,j,NIP,sectag,geotrantag,'-mass', mass,'-iter 20',0.001
);

fprintf(fileID,'\n');

loadtagtop(g)=eletag;

geotrantag=geotrantag+1;

g=g+1;

eletag=eletag+1;

end

cont=j;

end;

%ora faccio il travetto campata (sono tutti i pezzi delle travi del ponte)

A=Atrave;

E=handles.E; %modulo di young del calcestruzzo

```

G=handles.G; %modulo di taglio
% per gli elementi della campata l'utente mi da un'area, io ipotizzo una
% base di 500 mm e calcolo tutto di conseguenza
Iy=handles.Ixlong;
Iz=handles.Iylong;
J=Iy+Iz;

%blocco di istruzioni che segue nella prima campata, creo dei vettori i e
%j opportuni NodeI=i NodeJ=j
i(1)=(camp-2)*4+1;
j(1)=(camp-2)*4+1+nb;
%calcolo il numero totale di punti
ntot=0;
for k=1:(camp-1)
ntot=ntot+recordnt(k)*nb;
end;

ntot=ntot+camp*nb-nb;

%creo il vettore i "grezzo"
for n=2:(ntot)
    i(n)=i(n-1)+1;
end;
%creo il vettore j "grezzo"
for n=2:(ntot)
    j(n)=j(n-1)+1;
end;

%costruisco il vettore SpallaI
spallaI(1)=i(ntot)+nb+nb+nb+1;
l=2;

for k=1:(camp-2)
    if k==1
        for m=2:nb
            spallaI(l)=spallaI(l-1)+1;
            l=l+1;
        end;
    end
    if k~=1
        spallaI(l)=spallaI(l-1)+nb+1;
        l=l+1;
        for m=2:nb
            spallaI(l)=spallaI(l-1)+1;
            l=l+1;
        end;
    end
end

%costruisco il vettore SpallaJ
spallaJ(1)=j(ntot)+nb+1;
l=2;

for k=1:(camp-2)
    if k==1
        for m=2:nb

```

```

    spallaJ(l)=spallaJ(l-1)+1;
    l=l+1;
end;
end
if k~=1
    spallaJ(l)=spallaJ(l-1)+nb+1;
    l=l+1;
    for m=2:nb
        spallaJ(l)=spallaJ(l-1)+1;
        l=l+1;
    end;
end
end;

% sostituisco il tag dei nodi in corrispondenza dei "top pila" del vettore i
for k=1:(camp-2)
    if k==1
        n=recordnt(1)*nb +nb+1;
        l=1;
        for m=1:nb
            i(n)=spallaI(l);
            n=n+1;
            l=l+1;
        end;
    end
    if k~=1
        n=n+recordnt(k)*nb ;
        for m=1:nb
            i(n)=spallaI(l);
            n=n+1;
            l=l+1;
        end;
    end
end;

% sostituisco il tag dei nodi in corrispondenza dei "top pila" del vettore j
for k=1:(camp-2)
    if k==1
        n=recordnt(1)*nb +1;
        l=1;
        for m=1:nb
            j(n)=spallaJ(l);
            n=n+1;
            l=l+1;
        end;
    end
    if k~=1
        n=n+recordnt(k)*nb ;
        for m=1:nb
            j(n)=spallaJ(l);
            n=n+1;
            l=l+1;
        end;
    end
end;

%stampo

```

```

lunghezza=size(i);
g=1;
for m=1:(lunghezza(2))

    fprintf(fileID,'%5s %9s %8s %8s %4s %4s %4s %4s %5s %5s %13s\n','# Element
    "TravettoCampata":','eleTag','NodeI','NodeJ','A','E','G','J','Iy','Iz','GeoTranTag');
    fprintf(fileID,'%5s %6i %7i %7i %15.6E %15.6E %15.6E %15.6E %15.6E %15.6E %5i\n',
    'element elasticBeamColumn ','eletag,i(m),j(m),A,E,G,J,Iy,Iz,geotrantag);
    fprintf(fileID,'\n');
    loadtagtrav(g)=eletag;
    geotrantag=geotrantag+1;
    g=g+1;
    eletag=eletag+1;

end;

```

```

%ora faccio il traverso campata che sono tutti gli elementi trasversali al
%ponte tranne i top pila
% per gli elementi della campata l'utente mi da un'area, io ipotizzo una
% base di 500 mm e calcolo tutto di conseguenza
A=Atraverso;
Iy=handles.Ixtrav;
Iz=handles.Iytrav;
J=Iy+Iz;

```

```

%creo i vettori i e j che corrispondono ai NodeI nodeJ

```

```
i=1;
```

```
j=1;
```

```
i(1)=(camp-2)*4+1;
```

```
n=2;
```

```
for k=1:(camp-1)
```

```
    if k==1
```

```
        for p=1:(recordnt(1)+1)
```

```
            for l=2:(nb-1)
```

```
                i(n)=i(n-1)+1;
```

```
                n=n+1;
```

```
            end;
```

```
            if p~=1
```

```
                i(n)=i(n-1)+2;
```

```
                n=n+1;
```

```
            end;
```

```
        end;
```

```
    end;
```

```
    if k~=1
```

```
        i(n)=i(n-1)+2+nb;
```

```
        n=n+1;
```

```
        for p=1:recordnt(k)
```

```
            for l=2:(nb-1)
```

```
                i(n)=i(n-1)+1;
```

```
                n=n+1;
```

```
            end;
```

```
            if p~=1
```

```
                i(n)=i(n-1)+2;
```

```

        n=n+1;
    end;
end;
end;
end;

%aggingo l'ultima spalla
i(n)=i(n-1)+2;
n=n+1;
for z=1:(nb-2),
    i(n)=i(n-1)+1;
    n=n+1;
end;

% creo il vettore j
lunghezza=size(i);
for m=1:(lunghezza(2))
    j(m)=i(m)+1;
end;

%stampo

for m=1:(lunghezza(2))

    fprintf(fileID,'%5s %9s %8s %8s %4s %4s %4s %4s %5s %5s %13s\n','# Element
    "TraversoCampata":','eleTag','NodeI','NodeJ','A','E','G','J','Iy','Iz','GeoTranTag');
    fprintf(fileID,'%5s %6i %7i %7i %+15.6E %+15.6E %+15.6E %+15.6E %+15.6E %+15.6E %5i\n',
    'element elasticBeamColumn ',eletag,i(m),j(m),A,E,G,J,Iy,Iz,geotrantag);
    fprintf(fileID,'\n');
    eletag=eletag+1;
    geotrantag=geotrantag+1;
end;

%faccio tutti i link cerniera
%questo è un valore per le direzioni degli assi locali degli elementi
%zerolength
a=6.123234*(10^(-17));
% costruisco il vettore traversofibre

n=1;
for k=1:(camp-2)
    if k==1
        traversofibre(1)=(camp-2)*4+1+(recordnt(1)+1)*nb;
        n=n+1;
        for l=2:nb
            traversofibre(n)=traversofibre(n-1)+1;
            n=n+1;
        end;
    end;
    if k~=1
        traversofibre(n)=traversofibre(n-1)+(recordnt(k))*nb+1;
        n=n+1;
        for l=2:nb
            traversofibre(n)=traversofibre(n-1)+1;
            n=n+1;
        end;
    end;
end;

```

```

    end;
end;
end;

%stampo
lunghezza=size(traversofibre);
for m=1:(lunghezza(2))

    fprintf(fileID,'%5s %9s %8s %8s %7s %4s %7s %4s %30s\n','# Element "LinkCerniera":',
        'eleTag','NodeI','NodeJ','-mat','matTag1 matTag2 ...','-dir','dir1 dir2 ...','<-orient
x1 x2 x3 y1 y2 y3>');
    fprintf(fileID,'%5s %6i %7i %7i %5s %1i %2i %2i %2i %2i %2i %5s %1i %2i %2i %2i %2i
%2i %8s %+14.6E %+14.6E %+14.6E %+15.6E %+14.6E %+14.6E\n','element zeroLength ',
        eletag,traversofibre(m),spallaI(m),'-mat',2,3,4,1,6,5,'-dir',1,2,3,4,5,6,'-orient',1,0,
0,0,-1,a);
    fprintf(fileID,'\n');
    eletag=eletag+1;
end;
for m=1:(lunghezza(2))

    fprintf(fileID,'%5s %9s %8s %8s %7s %4s %7s %4s %30s\n','# Element "LinkCerniera":',
        'eleTag','NodeI','NodeJ','-mat','matTag1 matTag2 ...','-dir','dir1 dir2 ...','<-orient
x1 x2 x3 y1 y2 y3>');
    fprintf(fileID,'%5s %6i %7i %7i %5s %1i %2i %2i %2i %2i %2i %5s %1i %2i %2i %2i %2i
%2i %8s %+14.6E %+14.6E %+14.6E %+15.6E %+14.6E %+14.6E\n','element zeroLength ',
        eletag,traversofibre(m),spallaJ(m),'-mat',2,3,4,1,6,5,'-dir',1,2,3,4,5,6,'-orient',1,0,
0,0,-1,a);
    fprintf(fileID,'\n');
    eletag=eletag+1;
end;

%creo i due link spalla
%creo i vettori ad hoc che contengono i tag dei punti opportuni

%creo traverso sx

traversosx(1)=(camp-2)*4+1;
n=2;
for k=2:nb
    traversosx(n)=traversosx(n-1)+1;
    n=n+1;
end;
ntot=traversosx(n-1);
for k=1:(camp-1)
    ntot=ntot+recordnt(k)*nb;
end;

ntot=ntot+(camp-2)*nb+1;

traversodx(1)=ntot;
n=2;
for k=2:nb
    traversodx(n)=traversodx(n-1)+1;
    n=n+1;
end;

spallasx(1)=traversodx(n-1)+1;

```

```

n=2;
for k=2:nb
    spallasx(n)=spallasx(n-1)+1;
    n=n+1;
end;

spalladx(1)=spallasx(n-1)+(camp-2)*2*nb+1;
n=2;
for k=2:nb
    spalladx(n)=spalladx(n-1)+1;
    n=n+1;
end;

%stampo
lunghezza=size(traversosx);
for m=1:(lunghezza(2))

    fprintf(fileID,'%5s %9s %8s %8s %7s %4s %7s %4s %30s\n','# Element "LinkSpalla":',
        'eleTag','NodeI','NodeJ','-mat','matTag1 matTag2 ...','-dir','dir1 dir2 ...','<-orient
x1 x2 x3 y1 y2 y3>');
    fprintf(fileID,'%5s %6i %7i %7i %5s %1i %3i %3i %3i %3i %3i %5s %1i %2i %2i %2i %2i
%2i %8s %+14.6E %+14.6E %+14.6E %+15.6E %+14.6E %+14.6E\n','element zeroLength ',
        eletag,spallasx(m),traversosx(m),'-mat',11,3,4,1,6,5,'-dir',1,2,3,4,5,6,'-orient',1,0,0
,0,-1,a);
    fprintf(fileID,'\n');
    eletag=eletag+1;
end;
lunghezza=size(traversosx);
for m=1:(lunghezza(2))

    fprintf(fileID,'%5s %9s %8s %8s %7s %4s %7s %4s %30s\n','# Element "LinkSpalla":',
        'eleTag','NodeI','NodeJ','-mat','matTag1 matTag2 ...','-dir','dir1 dir2 ...','<-orient
x1 x2 x3 y1 y2 y3>');
    fprintf(fileID,'%5s %6i %7i %7i %5s %1i %3i %3i %3i %3i %3i %5s %1i %2i %2i %2i %2i
%2i %8s %+14.6E %+14.6E %+14.6E %+15.6E %+14.6E %+14.6E\n','element zeroLength ',
        eletag,spalladx(m),traversodx(m),'-mat',11,3,4,1,6,5,'-dir',1,2,3,4,5,6,'-orient',1,0,0
,0,-1,a);
    fprintf(fileID,'\n');
    eletag=eletag+1;
end;

fclose(fileID);

handles.loadtagtrav=loadtagtrav;
handles.loadtagtop=loadtagtop;
handles.loadtagpile=loadtagpile;
handles.mass=mass;
handles.recordpile=recordpile;
handles.lunghpile=lunghpile;
%save loadtagtrav.mat loadtagtrav;
%save loadtagtop.mat loadtagtop;
%save loadtagpile.mat loadtagpile;
%save mass.mat mass;
%save recordpile.mat recordpile;

%copio il tcl nelle sottocartelle
copyfile('Elements.tcl','TERNAL/PC1');

```

```
copyfile('Elements.tcl','TERNA1/PC2');
copyfile('Elements.tcl','TERNA1/PC3');
copyfile('Elements.tcl','TERNA1/PC4');
copyfile('Elements.tcl','TERNA1/PC5');
copyfile('Elements.tcl','TERNA1/PC6');
copyfile('Elements.tcl','TERNA1/PC7');
copyfile('Elements.tcl','TERNA1/PC8');
copyfile('Elements.tcl','TERNA1/PC9');
copyfile('Elements.tcl','TERNA1/PC10');
copyfile('Elements.tcl','TERNA1/PC11');
copyfile('Elements.tcl','TERNA1/PC12');
copyfile('Elements.tcl','TERNA1/PC13');
copyfile('Elements.tcl','TERNA1/PC14');
copyfile('Elements.tcl','TERNA1/PC15');
```

```
copyfile('Elements.tcl','TERNA2/PC1');
copyfile('Elements.tcl','TERNA2/PC2');
copyfile('Elements.tcl','TERNA2/PC3');
copyfile('Elements.tcl','TERNA2/PC4');
copyfile('Elements.tcl','TERNA2/PC5');
copyfile('Elements.tcl','TERNA2/PC6');
copyfile('Elements.tcl','TERNA2/PC7');
copyfile('Elements.tcl','TERNA2/PC8');
copyfile('Elements.tcl','TERNA2/PC9');
copyfile('Elements.tcl','TERNA2/PC10');
copyfile('Elements.tcl','TERNA2/PC11');
copyfile('Elements.tcl','TERNA2/PC12');
copyfile('Elements.tcl','TERNA2/PC13');
copyfile('Elements.tcl','TERNA2/PC14');
copyfile('Elements.tcl','TERNA2/PC15');
```

```
copyfile('Elements.tcl','TERNA3/PC1');
copyfile('Elements.tcl','TERNA3/PC2');
copyfile('Elements.tcl','TERNA3/PC3');
copyfile('Elements.tcl','TERNA3/PC4');
copyfile('Elements.tcl','TERNA3/PC5');
copyfile('Elements.tcl','TERNA3/PC6');
copyfile('Elements.tcl','TERNA3/PC7');
copyfile('Elements.tcl','TERNA3/PC8');
copyfile('Elements.tcl','TERNA3/PC9');
copyfile('Elements.tcl','TERNA3/PC10');
copyfile('Elements.tcl','TERNA3/PC11');
copyfile('Elements.tcl','TERNA3/PC12');
copyfile('Elements.tcl','TERNA3/PC13');
copyfile('Elements.tcl','TERNA3/PC14');
movefile('Elements.tcl','TERNA3/PC15');
```

end