POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Civile

Tesi di Laurea Magistrale

Sviluppo di un software FEM open source per l'analisi statica nonlineare con metodo puramente incrementale di strutture in c.a.

Relatore:
Marco Gherlone

Candidato:
Filippo Cinquia

Dicembre 2020

English title of the thesis

Development of an open source FEM software for nonlinear static analysis with purely incremental method of RC structures

Abstract

L'argomento di questa tesi è un programma di calcolo open source, agli elementi finiti, per l'analisi statica nonlineare di strutture in cemento armato. Il programma è in fase di sviluppo: è stato implementato soltanto lo stretto necessario per poter fare delle prime valutazioni sul suo funzionamento.

Il codice è scritto in FORTRAN, ed è basato su quanto illustrato in *Programming the finite element method*, di Griffiths et al. (2014). È stato scelto un metodo puramente incrementale (forward Euler method) per l'approssimazione delle leggi costitutive nonlineari. Il modello di calcestruzzo adottato è basato sul lavoro di Kotsovos (2015): *Finite-element modelling of structural concrete*.

L'idea alla base di questo lavoro è quella di creare un software che abbia delle caratteristiche chiave di semplicità e affidabilità. E arrivare quindi ad un programma snello (che implementi solo ciò che è veramente necessario), con una modellazione semplificata (che non rinunci a risultati buoni/ottimi) e con un codice open source semplice e di qualità (conciso, pulito, facilmente leggibile/modificabile e testato passo-passo).

Il programma è stato testato per confronto con analisi sperimentali e numeriche effettuate da vari autori su semplici strutture isostatiche in c.a. quali travi, colonne e muri a taglio: esso mostra un buon funzionamento. Nei risultati, gli ordini di grandezza sono stati rispettati sotto quasi tutti i punti di vista: carichi di rottura, spostamenti, modalità di rottura, duttilità delle strutture, comportamento a fessurazione, capacità di cogliere effetti quali il confinamento del calcestruzzo. Le valutazioni sul metodo puramente incrementale sono molto incoraggianti.

Un importante obiettivo per il futuro è quello di arrivare ad un programma che possa eseguire delle complete analisi di pushover su qualsiasi tipo di struttura in cemento armato.

Abstract

The subject of this thesis is an open source finite element program for nonlinear static analysis of reinforced concrete structures. The program is under development: only the strictly necessary has been implemented to be able to make initial assessments of its operation.

The code is written in FORTRAN, and is based on what is illustrated in *Programming the finite element method*, by Griffiths et al. (2014). A purely incremental method (forward Euler method) was chosen for the approximation of nonlinear constitutive laws. The concrete model adopted is based on the work of Kotsovos (2015): *Finite-element modelling of structural concrete*.

The idea behind this work is to create a software that has key characteristics of simplicity and reliability. And therefore to arrive at a simple program (which implements only what is really necessary), with a simplified modeling (without giving up good/excellent results) and with a simple and quality open source code (concise, clean, easily readable/editable and tested step-by-step).

The program was tested by comparison with experimental and numerical analysis carried out by various authors on simple rc isostatic structures such as beams, columns and shear walls: it shows good functioning. In the results, the orders of magnitude were respected from almost all points of view: failure loads, displacements, failure modality, ductility of the structures, cracking behavior, ability to grasp effects such as concrete confinement. The evaluations on purely incremental method are very encouraging.

An important goal for the future is to arrive at a program that can perform comprehensive pushover analysis on any type of reinforced concrete structures.

Indice

Introduzione

L'argomento di questa tesi è un programma di calcolo open source, agli elementi finiti, per l'analisi statica nonlineare di strutture in cemento armato. Il programma è in fase di sviluppo: è stato implementato soltanto lo stretto necessario per poter fare delle prime valutazioni sul suo funzionamento.

Il codice è scritto in FORTRAN, con la versione del 2003, ed è basato su quanto illustrato in Griffiths et al. (2014). Questo libro permette a mio parere di scrivere programmi non solo open source, con tutti i vantaggi che ne derivano, ma anche con un codice particolarmente semplice e di qualità. E' stato usato un metodo puramente incrementale, il *forward Euler method* (Metodo di Eulero (2020)), per l'approssimazione delle leggi costitutive nonlineari: esso permette di semplificare il codice in maniera molto significativa, e questo probabilmente realizza l'idea innovativa più importante alla base di questo lavoro. Il modo in cui il metodo degli elementi finiti è stato implementato nel programma è oggetto del primo capitolo della tesi.

Il modello di calcestruzzo è basato sul lavoro di Kotsovos (2015): oltre ad essere un ottimo libro (anche per i motivi che vedremo tra poco) mi risulta sia l'unico a fornire informazioni non solo teoriche, ma anche su come implementarle in un programma di calcolo. L'implementazione dei materiali è descritta nel capitolo due.

L'idea alla base di questo lavoro è quindi quella di creare un software che abbia delle caratteristiche chiave di semplicità e affidabilità. E arrivare quindi ad un programma snello (che implementi solo ciò che è veramente necessario), con una modellazione semplificata (che non rinunci a risultati buoni/ottimi) e con un codice open source semplice e di qualità (conciso, pulito, facilmente leggibile/modificabile e testato passo-passo).

Sono stato spinto a questo lavoro innanzi tutto dall'interesse verso la materia: il metodo di calcolo agli elementi finiti, con particolare applicazione al cemento armato, attraverso la scrittura di un codice di calcolo open source. Si tratta di un argomento vastissimo, su cui è presente una mole di pubblicazioni. Un report sullo stato dell'arte è quello della Fédération internationale du béton (Fib (2008)), anche se i libri che ho maggiormente consultato sono quelli di Griffiths et al. (2014), per il metodo degli elementi finiti, e di Kotsovos (2015), sulla sua applicazione al cemento armato. Delle informazioni di base sull'open source possono essere trovate in Open Source (2020). Nei primi due capitoli fornirò le informazioni fondamentali sullo stato dell'arte, all'interno di ogni paragrafo.

I programmi di calcolo nonlineare sul cemento armato sono molto numerosi: alcuni di tipo libero, altri commerciali. Per fare alcuni esempi cito quelli di cui ho esperienza: ANSYS, ADINA, ATENA, DIANA FEA, OpenSees (vedi in bibliografia per i riferimenti), dei quali solo l'ultimo è un software open source. La bontà e l'accuratezza delle analisi eseguite con questi software sono oggetto di discussione. La Fib, nel citato report sullo stato dell'arte, fa notare che si possono ottenere risultati tutto sommato sufficientemente buoni/accettabili, ma solo a determinate condizioni. Queste sono: una buona/ottima esperienza da parte dell'operatore, la scelta del modello di calcestruzzo e la calibrazione preliminare di diversi parametri, entrambi in funzione della simulazione che si sta svolgendo. I parametri da calibrare possono essere numerosi e riguardare qualsiasi aspetto dell'analisi: la discretizzazione della struttura, la modellazione delle condizioni al contorno, delle scelte sui materiali (ad esempio i valori dei parametri delle leggi costitutive di calcestruzzo e acciaio) e numeriche (numero di step di carico, tipologia di metodo iterativo, criteri e valori di convergenza, ecc).

Kotsovos (2015) punta il dito proprio contro questa mancanza di generalità, affermando di aver sviluppato un modello di calcestruzzo valido per ogni tipo di analisi (statica, sia monotona che ciclica, e dinamica, sia da terremoto che da impatto), su qualsiasi tipo di struttura, e che consenta simulazioni che non richiedano alcun tipo di calibrazione. Vedremo al capitolo sui materiali il motivo fondamentale cui Kotsovos attribuisce il buon funzionamento del suo modello rispetto agli altri. Il modello è stato implementato in diversi programmi di calcolo commerciali, quali ad esempio ADINA e DIANA FEA, ma non è presente, per quanto mi risulta, in nessun programma di calcolo open source: questo ha costituito un'ulteriore motivazione al mio lavoro. Nella discussione sul metodo puramente incrementale (al primo capitolo), vedremo anche un'applicazione di uno di questi programmi commerciali alla legge costitutiva del calcestruzzo.

Per quanto riguarda la mia esperienza, l'impressione che mi sono fatto sullo stato dell'arte di queste analisi è complessivamente abbastanza negativa: in particolare sotto carichi ciclici, ma per diversi aspetti anche sotto carichi monotoni, mi è quasi sempre capitato di vedere risultati poco buoni, fortemente discordanti nel confronto con le stesse analisi eseguite sperimentalmente. E questo sempre a fronte di uno sforzo e di un impegno notevole (mio, e delle persone con cui ho avuto modo di lavorare) nel cercare di impostare i modelli e di correggerli in base ai risultati. E c'è un altro aspetto su cui sono critico. Anche a fronte di evidenti difficoltà, i programmi di calcolo per questo tipo di analisi non rinunciano a implementazioni a mio avviso eccessivamente e inutilmente complesse e minuziose della modellazione, anche su aspetti solitamente secondari di essa. Per fare un esempio, viene in genere data la possibilità di modellare il comportamento a trazione del calcestruzzo con molti tipi di leggi costitutive, addirittura a volte più di una decina (lineare, eventualmente con tratto di softening, elastoplastica, elastofragile, multilineare, esponenziale, iperbolica, più tutta una serie di leggi proposte

da autori o norme tecniche, a loro volta spesso molto complesse). Ora, un concetto molto importante che ho imparato in questo corso di laurea, è che tanto più i calcoli e i modelli sono complessi, tanto più è alta la probabilità di commettere errori. Viceversa la semplicità, la snellezza, o anche il solo fatto di rinunciare ad offrire una moltitudine di alternative diverse per la modellazione dello stesso comportamento/fenomeno, portano di fatto a una maggiore affidabilità e qualità dei programmi di calcolo. Al limite, alcuni dettagli e aspetti secondari potrebbero non essere colti, ma diminuisce drasticamente la probabilità di commettere errori grossolani sugli aspetti fondamentali del calcolo. Alla luce di tutto questo penso che il programma proposto in questa tesi, con un prossimo sviluppo, possa trovare il suo spazio nel panorama dei programmi di calcolo nonlineari per cemento armato. Penso anche che l'intero settore beneficerebbe di alcune delle caratteristiche chiave con cui è stato pensato questo software.

Il programma è in fase di sviluppo e questo è dovuto a diversi motivi. È stato fatto un tentativo di scrittura del codice di calcolo nella migliore maniera possibile, ma sia esso che il file con cui vengono forniti i dati di input al programma sono in una versione abbastanza rudimentale. Ad esempio, nel codice non è stato fatto uso di subroutine (sarebbero fondamentali per migliorare il programma) mentre, per fare un esempio sul file di input, per inserire una barra di acciaio è necessario specificare i nodi a cui deve essere vincolato ognuno degli elementi asta in cui la barra è suddivisa (non è cioè sufficiente inserire i punti di inizio e fine della barra) e questo è evidentemente abbastanza scomodo. Ritengo importante a questo punto una precisazione. Si è parlato inizialmente di un programma con un codice avente caratteristiche di semplicità e qualità. Ebbene queste caratteristiche non sono ancora del tutto soddisfatte dalla versione *attuale* del programma. Il modo in cui esso è *impostato* potrà portalo in futuro, con un successivo sviluppo, a soddisfare appieno tali caratteristiche.

Ci sono poi delle limitazioni. Come detto, il criterio che ho scelto per decidere cosa implementare è stato quello di creare un programma avente il minimo indispensabile per poter essere testato (senza rinunciare a nessuna delle caratteristiche fondamentali della modellazione). Testare il programma è consistito nell'effettuare confronti con i risultati sperimentali e con le analisi numeriche effettuate da vari autori: tale confronto è oggetto del capitolo 3.

Di seguito elenco i principali limiti del programma.

- L'attuale implementazione è in due dimensioni, per soli stati di tensione piana.
- La nonlinearità presa in considerazione è soltanto dei materiali ed è stata quindi trascurata la nonlinearità geometrica. Il libro di Kotsovos ha fatto questa scelta, per non "appesantire la trattazione", e personalmente sono stato del tutto d'accordo.

- Per quanto riguarda la geometria, è possibile inserire soltanto strutture di forma rettangolare. Questo costituisce ovviamente un enorme limite, ma lascia la possibilità di analizzare elementi come travi, colonne e pareti a taglio. Le barre possono essere poste soltanto in orizzontale e in verticale, quindi non in direzione qualsiasi.
- Gli elementi utilizzabili sono soltanto i due scelti, entrambi del prim'ordine: quadrangolare piano, di forma rettangolare e a quattro nodi, e monodimensionale rettilineo a due nodi.
- I materiali a disposizione sono solo tre: il calcestruzzo e un lineare elastico (eventualmente infinitamente rigido) per gli elementi bidimensionali, e un elastoplastico bilineare per le barre. È possibile utilizzare un solo tipo di calcestruzzo per l'intera struttura ma diversi tipi di barre.
- I carichi devono essere monotoni crescenti (non sono quindi previste analisi cicliche) e le analisi possono essere solo in controllo di forza e non di spostamento. Gli step di carico possono variare di intensità ma le forze devono rimanere proporzionali a loro stesse durante tutta l'analisi. In altre parole, deve esistere un unico moltiplicatore dei carichi.
- La fessurazione del calcestruzzo è stata modellata con delle semplificazioni: è stata trascurata la resistenza a trazione del calcestruzzo ed è stato usato un modello di fessurazione diffusa e ortogonalmente fissa, ossia con possibilità di formazione delle fessure nelle sole direzioni assiali e quindi non in direzione qualsiasi.
- La modellazione della rottura dei materiali (splitting del calcestruzzo e rottura della barre di acciaio) non è stata implementata. Questo comporta, tra l'altro, che le analisi terminano non appena un elemento raggiunge il proprio criterio di rottura.

Nelle conclusioni della tesi indicherò, in maniera qualitativa, la strada da intraprendere per eliminare ognuna di queste limitazioni. Un importante obiettivo per il futuro è quello di arrivare ad un software che possa eseguire delle complete analisi di pushover (vedi ad esempio CEN, *Eurocode 8* (2004), Abunama (2017) e Parducci (2016)). Intendo quindi su qualsiasi tipo di struttura in cemento armato, nelle tre dimensioni, con qualsiasi tipo di elementi (almeno i più comuni), senza semplificazioni riguardo alla fessurazione (o al limite trascurando la sola resistenza a trazione) e col calcestruzzo di Kotsovos (2015) (preferibilmente), o con altri modelli.

Introduction

The subject of this thesis is an open source, finite element program for nonlinear static analysis of reinforced concrete structures. The program is in a development phase: only what is strictly necessary has been implemented in order to make initial assessments of its operation.

The code is written in FORTRAN, with 2003 version, and it's based on what is illustrated in Griffiths et al. (2014). In my opinion, this book allows you to write not only open source programs, with all the advantages that come with it, but also with a particularly simple and quality code. A purely incremental method has been used, the *forward Euler method* (Metodo di Eulero (2020)), for the approximation of nonlinear constitutive laws: it allows to simplify the code in a very significant way, and this probably realizes the most important innovative idea behind this work. The way the finite element method has been implemented in the program is the subject of the first chapter of the thesis.

The concrete model is based on the work of Kotsovos (2015): as well as being an excellent book (also for reasons we'll see in a moment) it is the only one to provide not only theoretical information but also on how to implement them in a calculation program. The implementation of materials is described in chapter two.

The idea behind this work is therefore to create a software that has key characteristics of simplicity and reliability. And therefore to arrive at a simple program (that implements only what is really necessary), with simplified modelling (that doesn't give up good/best results) and with a simple and quality open source code (concise, clean, easily readable/modifiable and step-by-step tested).

I was driven to this work first of all by my interest in the subject: the finite element method, with particular application to reinforced concrete, through the writing of an open source code. This is a vast subject, on which there Is a huge number of publications. A report on the state of the art is that of the Fédération internationale du béton (Fib (2008)), although the books I've most consulted are those by Griffiths et al. (2014), for the finite element method, and by Kotsovos (2015), on its application to reinforced concrete. Basic information on open source can be found in Open Source (2020). In the first two chapters i'll provide basic information on the state of the art, within each paragraph.

Nonlinear programs on reinforced concrete are very numerous: some of them free type, others commercial. To give some examples, i mention those i have experience with: ANSYS, ADINA, ATENA, DIANA FEA, OpenSees (see bibliography for references), of which only the last one is open source. The goodness and accuracy of the analysis performed with these software are subject of discussion. Fib, in

the above mentioned state of the art report, points out that good/acceptable results can be obtained, but only under certain conditions. These are: a good/optimal experience of the operator, the choice of concrete model and preliminary calibration of several parameters, both depending on the simulation being carried out. The parameters to be calibrated can be numerous and concern any aspect of the analysis: discretization of the structure, modeling of boundary conditions, some choices on materials (for example the parameter values of constitutive laws of concrete and steel) and numerical (number of load steps, type of iterative method, criteria and convergence values, etc.).

Kotsovos (2015) points the finger precisely against this lack of generality, claiming to have developed a concrete model valid for any type of analysis (static, both monotonic and cyclic, and dynamic, both earthquake and impact), on any type of structure, and allowing simulations that do not require any kind of calibration. We'll see in the chapter on materials the fundamental reason Kotsovos attributes the good functioning of its model with respect to the others. The model has been implemented in several commercial programs, such as ADINA and DIANA FEA, but is not present, as far as I know, in any open source program: this has been a further motivation for my work. In the discussion on purely incremental method (in the first chapter), we will also see an application of one of these commercial software to concrete constitutive law.

As far as my experience is concerned, my impression of the state of the art of these analysis is overall quite negative: in particular under cyclic loads, but for various aspects also under monotonic loads, i almost always happened to see little good results, strongly discordant in the comparison with the same analysis performed experimentally. And this always in the face of a considerable effort and commitment (mine, and the people with whom I had the opportunity to work) in trying to set the models and correct them based on the results. And there's another aspect I'm critical of. Even with obvious difficulties, the calculation programs for this type of analysis do not give up implementations in my opinion excessively and unnecessarily complex and detailed of the modelling, even on usually secondary aspects of it. For example, it is generally possible to model the tensile behaviour of concrete with many types of constitutive laws, sometimes more than a dozen (linear, possibly with softening, elastoplastic, elastofragile, multilinear, exponential, hyperbolic, plus a whole series of laws proposed by authors or code of practice, in their turn often very complex). Now, one very important concept that i learned in this graduation course is that the more complex the calculations and models are, the higher the probability of making mistakes. On the other hand, simplicity, slenderness, or even the mere fact of giving up to offer a multitude of different alternatives for modeling the same behavior/ phenomenon, lead in fact to greater reliability and quality of calculation programs. To the limit, some details and secondary aspects may not be read, but drastically decreases the probability of making gross errors on fundamental aspects of calculation. In the light of all this i think that the program

proposed in this thesis, with a future development, can find its space in the panorama of nonlinear calculation programs for reinforced concrete. I also think that the entire industry would benefit from some of the key features with which this software was designed.

The program is in a development phase and this is due to several reasons. An attempt has been made to write the calculation code in the best possible way, but both it and the file with which the input data is provided to the program are in a fairly rudimentary version. For example, no subroutines have been used in the code (they would be fundamental to improve the program) while, to give an example on the input file, to insert a steel bar it is necessary to specify the nodes to which each of the rod elements, into which the bar is divided, is bound (i.e. it is not sufficient to insert the start and end points of the bar) and this is obviously quite uncomfortable. I think a clarification is important at this point. At the beginning we talked about a program with a code that has characteristics of simplicity and quality. Well, these characteristics are not yet fully satisfied by the *current* version of the program. The way it is *set up* may in the future, with subsequent development, lead it to fully satisfy these features.

There are also limitations. As said, the criterion i chose to decide what to implement was to create a program with the minimum necessary to be tested (without giving up any of the fundamental characteristics of the modelling). Testing the program consisted in making comparisons with experimental results and numerical analysis carried out by various authors: this comparison is the subject of chapter 3.

Below i list the main limitations of the program.

- The current implementation is in two dimensions, only for plane stress analysis.
- Nonlinearity taken into consideration is only of materials and therefore geometric nonlinearity has been neglected. Kotsovos' book made this choice, in order not to burden the discussion, and i personally agreed completely.
- As far as geometry is concerned, only rectangular structures can be inserted. This is obviously a huge limitation, but it leaves the possibility to analyse elements such as beams, columns and shear walls. The bars can only be placed horizontally and vertically, therefore not in any direction.
- The elements that can be used are only the two chosen, both of the first order: 2D quadrangular, rectangular in shape and with four nodes, and one-dimensional straight with two nodes.
- There are only three materials available: concrete and a linear elastic (possibly infinitely rigid) for two-dimensional elements, and a bilinear elastoplastic for the bars. Only one type of concrete can be used for the entire structure but different types of bars.
- The loads must be increasing monotonic (no cyclic analysis is therefore foreseen) and the analysis can only be in force control and not in displacement control. Load steps may vary in intensity but forces

must remain proportional to themselves throughout the analysis. In other words, there must be a single load multiplier.

- Concrete cracking has been modelled with simplifications: tensile strength of concrete has been neglected and a diffuse and orthogonally fixed crack model has been used, i.e. with the possibility of cracking in axial directions only and therefore not in any direction.
- The modelling of material failure (concrete splitting and steel bar rupture) has not been implemented. This means, among other things, that the analysis ends as soon as an element reaches its failure criterion.

In the conclusions of the thesis i will indicate, in a qualitative manner, the way to eliminate each of these limitations. An important goal for the future is to arrive at a software that can perform comprehensive pushover analysis (see for example CEN, *Eurocode 8* (2004), Abunama (2017) and Parducci (2016)). I mean therefore on any type of reinforced concrete structure, in three dimensions, with any type of elements (at least the most common), without simplifications regarding cracking (or at least neglecting only tensile strength) and with Kotsovos concrete (2015) (preferably), or with other models.

Capitolo

1 Implementazione del metodo degli elementi finiti

In questo capitolo descrivo il modo in cui il metodo degli elementi finiti è stato implementato all'interno del programma.

Per quanto riguarda le analisi lineari, tutta l'implementazione del metodo ricalca fedelmente quanto riportato in Griffiths et al. (2014). In particolare, il programma di questa tesi è stato scritto a partire dal *Program p51* del libro appena citato: quest'ultimo permette l'analisi statica di un solido lineare elastico. Entrambi i programmi sono riportati in appendice. Per questa prima parte, per diversi motivi, mi è sembrato più opportuno limitarmi a quanto segue, invece di fornire una descrizione generale del metodo. Darò quindi per scontata la conoscenza, da parte del lettore, di alcuni concetti fondamentali del metodo stesso.

Nel paragrafo sulle analisi non lineari, dopo una breve descrizione del metodo di Newton – Raphson (probabilmente il più famoso metodo per questo tipo di analisi), illustro il metodo puramente incrementale usato in questa tesi.

Chiude un paragrafo sul criterio usato per dare il termine alle analisi.

1.1 Analisi lineari

In questo paragrafo parlo inizialmente del tipo di elementi finiti che sono stati scelti e del modo in cui è stata discretizzata la struttura. Dopodiché vediamo il modo in cui sono state ottenute le matrici di rigidezza dei singoli elementi, come queste sono state assemblate per ottenere la matrice di rigidezza dell'intera struttura, e come sono state introdotte le condizioni al contorno all'interno di tale matrice. Infine la risoluzione dell'equazione di equilibrio globale permetterà di calcolare gli spostamenti dei nodi, e quindi le tensioni e le deformazioni all'interno di ciascun elemento. Come detto, l'implementazione è basata sul libro di Griffiths et al. (2014), e molte delle scelte effettuate sono dovute semplicemente al fatto di aver seguito fedelmente tale riferimento.

Sono stati scelti solo due elementi, entrambi del prim'ordine, e sono stati scelti perché sono i più semplici che potevano essere presi in considerazione. Il primo è quadrangolare piano, di forma rettangolare e a quattro nodi, il secondo è monodimensionale rettilineo a due nodi. L'elemento monodimensionale è di tipo asta, ossia è presa in considerazione la rigidezza assiale e sono trascurate le rigidezze flessionali e a taglio: questo è in genere del tutto accettabile per il cemento armato vista la snellezza tipica delle barre di acciaio al suo interno. Preciso anche che a rigore bisognerebbe chiamare *aste* gli elementi finiti e *barre* le barre (fisiche) di acciaio; spesso invece userò il termine di *barre* anche per gli elementi finiti (non è rigoroso ma mi è sembrato meno fonte di possibile confusione).

La discretizzazione della struttura avviene in maniera molto semplice, con suddivisione in elementi lungo gli assi x e y:

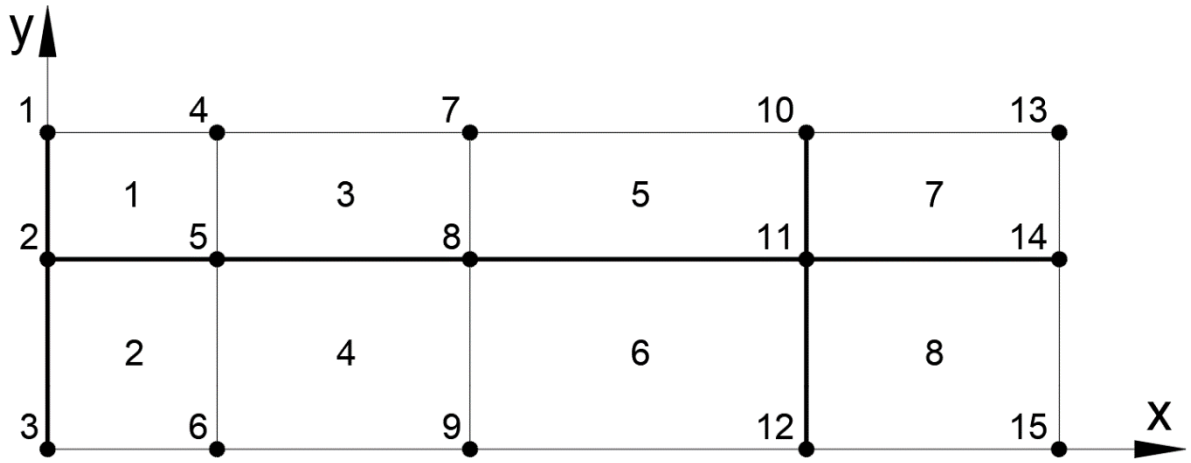

Figura 1 Esempio di discretizzazione di una struttura (gli elementi monodimensionali sono rappresentati dalle linee in grassetto) e di numerazione globale di nodi ed elementi bidimensionali

Anche la numerazione dei nodi e degli elementi bidimensionali avviene con un algoritmo molto semplice: viene numerata ogni colonna di nodi ed elementi lungo l'asse negativo delle y, poi si passa alla colonna successiva a destra (asse positivo delle x) e così via. Nella figura gli elementi monodimensionali non hanno invece numerazione perché essa non avviene in automatico, con un algoritmo, ma dipende dal modo in cui le barre sono inserite all'interno del file di testo che fornisce i dati di input (vedi appendice B).

Si può anche vedere che non è necessario che la spaziatura degli elementi lungo gli assi sia uniforme. Gli elementi monodimensionali devono invece essere posti sempre in orizzontale o in verticale e ciascuno di essi deve collegare due nodi dello stesso elemento bidimensionale. Tra elementi bidimensionali e aste si fa sempre l'ipotesi di ancoraggio perfetto.

Il sistema di riferimento locale degli elementi quadrilateri fa uso delle coordinate adimensionalizzate ξ ed η:

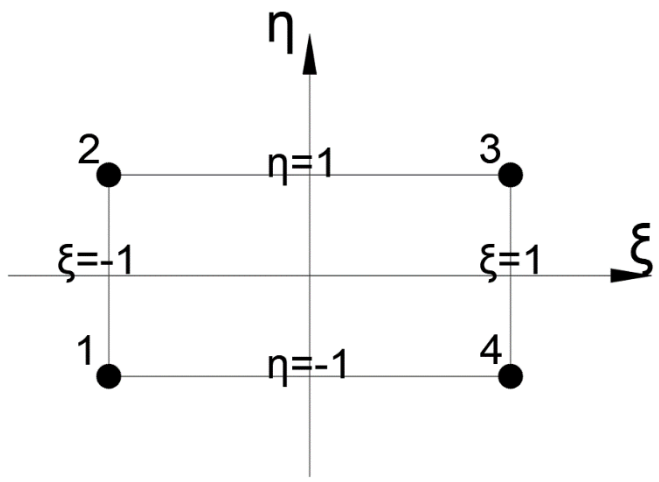

Figura 2 Sistema di rifermento locale degli elementi bidimensionali (anche la numerazione dei nodi è locale)

Il calcolo della matrice di rigidezza di questi elementi avviene per integrazione su quattro punti di Gauss. Essi hanno coefficienti di peso W_i pari a 1 e coordinate:

$$(\xi_i, \eta_i) = (\pm 1/\sqrt{3}, \pm 1/\sqrt{3})$$

Le funzioni di forma sono:

$$N_1 = {}^1/_4\,(1-\xi)(1-\eta)$$

$$N_2 = {}^1/_4\,(1-\xi)(1+\eta)$$

$$N_3 = {}^1/_4\,(1+\xi)(1+\eta)$$

$$N_4 = {}^1/_4\,(1+\xi)(1-\eta)$$

Raccolte nella relativa matrice:

$$[N] = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix}$$

Le derivate delle funzioni di forma rispetto alle coordinate globali sono contenute nella matrice:

$$[B] = [A][N]$$

essendo infatti [A] la matrice delle derivate rispetto alle coordinate globali:

$$[A] = \begin{bmatrix} \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{bmatrix}$$

La matrice jacobiana contiene invece le derivate delle coordinate globali rispetto alle coordinate locali:

$$[J] = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix}$$

La matrice costitutiva di un materiale lineare elastico è, per stati di tensione piana:

$$[D] = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \dfrac{1-\nu}{2} \end{bmatrix}$$

Dove E e ν sono il modulo di Young e il coefficiente di Poisson.

Per integrazione discreta, si può quindi calcolare la matrice di rigidezza:

$$[k] = \sum_{i=1}^{4} W_i\, det|J|_i([B]^T[D][B])_i$$

Per un elemento a quattro nodi in due dimensioni, la matrice di rigidezza sarà ovviamente una matrice 8 x 8.

Per gli elementi monodimensionali ho utilizzato un procedimento più diretto per il calcolo della matrice di rigidezza. Sia data un asta di lunghezza l, sezione A e rigidezza E_m in uno spazio unidimensionale (il pedice m indica appunto che si tratta degli elementi monodimensionali):

Figura 3 Elemento asta in uno spazio a una sola dimensione

Se impongo uno spostamento unitario prima ad una estremità e poi all'altra si ottengono le seguenti forze:

$$u_1 = 1 \Rightarrow f_1 = -f_2 = \frac{E_m A}{l}$$

$$u_2 = 1 \Rightarrow f_1 = -f_2 = -\frac{E_m A}{l}$$

I valori delle forze sono proprio gli elementi della matrice di rigidezza (per $u_1 = 1$ la prima colonna e per $u_2 = 1$ la seconda):

$$[k]_m = {}^{E_m A}/_{l} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

È possibile prendere in considerazione questa matrice di rigidezza nello spazio unidimensionale perché le barre sono poste sempre orizzontalmente o verticalmente, quindi lungo uno dei due assi.

Di fatto, per sommare le due matrici, è sufficiente addizionare i quattro termini della matrice di rigidezza dell'elemento monodimensionale ad altrettanti termini della matrice di rigidezza dell'elemento bidimensionale. Un semplice algoritmo permette di individuare tali termini all'interno della matrice 8 x 8. Quest'operazione di somma dei quattro scalari va fatta per ognuna delle barre presenti all'interno di un singolo elemento bidimensionale. Si otterrà quindi un'unica matrice di rigidezza, di dimensioni 8 x 8, che terrà conto sia della rigidezza dell'elemento bidimensionale, che di quella delle eventuali barre in esso presenti.

L'operazione di assemblaggio permette di passare dalle singole matrici di rigidezza alla matrice di rigidezza [K] dell'intera struttura (ho usato la lettera minuscola per le matrici degli elementi e quella maiuscola per la matrice della struttura). Essendo in due dimensioni, essa sarà una matrice 2n x 2n, con n pari al numero di nodi della struttura. Tale matrice ha due importanti proprietà: è simmetrica e

a banda; quest'ultima vuol dire che i termini diversi da zero sono concentrati intorno alla diagonale principale. Queste proprietà permettono al programma di memorizzare non l'intera matrice, ma soltanto la parte contenente i termini diversi da zero e, per simmetria, relativa a metà matrice. Inoltre, una particolarità del software è che memorizza tali valori non in una matrice ma all'interno di un vettore (per non fare confusione continuerò a parlare di *matrice* di rigidezza).

La subroutine che genera tale matrice incorpora di fatto anche la condizione al contorno relativa ai nodi che hanno uno o entrambi i gradi di libertà vincolati. Tali gradi di libertà vengono infatti meno in questo caso e le relative componenti di rigidezza possono essere scartate. Di fatto, la matrice di rigidezza della struttura diventa di dimensioni (2n-m) x (2n-m), con m numero di gradi di libertà vincolati. Anche i vettori spostamento dei nodi {U} e forze esterne {F}, inizialmente di lunghezza 2n:

$$\{U\} = \{u_{x1}, u_{y1}, \dots u_{xn}, u_{yn}\}^T$$

$$\{F\} = \{f_{x1}, f_{y1}, \dots f_{xn}, f_{yn}\}^T$$

diventano per lo stesso motivo di dimensione pari a 2n-m, con gli spostamenti e le forze associati ai gradi di libertà vincolati eliminati dal calcolo.

L'altra informazione relativa alle condizioni al contorno riguarda gli spostamenti imposti. Essa viene introdotta nel calcolo andando a modificare la matrice di rigidezza e il vettore contenente le forze esterne da applicare alla struttura. Al termine della diagonale principale, in corrispondenza del grado di libertà cui è applicato lo spostamento imposto, viene aggiunto un valore numericamente molto alto (nel programma è 10^{20}):

$$k_{i,i} = k_{i,i} + 10^{20}$$

con i indice del grado di libertà in questione. Il valore di forza esterna relativo a tale grado di libertà viene invece posto pari al valore di spostamento imposto moltiplicato per lo stesso valore numericamente molto grande:

$$f_i = 10^{20} s_i$$

dove s_i è, appunto, lo spostamento imposto.

Infine, sarà sufficiente risolvere l'equazione di equilibrio globale:

$$[K]\{U\} = \{F\}$$

per trovare gli spostamenti dei nodi della struttura:

$$\{U\} = [K]^{-1}\{F\}$$

Noti gli spostamenti è possibile calcolare i vettori delle deformazioni {ε} e delle tensioni {σ} di ogni singolo elemento bidimensionale:

$$\{\varepsilon\} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = [B]\{u\}$$

$$\{\sigma\} = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = [D]\{\varepsilon\}$$

con il vettore {u] spostamento dei nodi del dato elemento:

$$\{u\} = \{u_{x1}, u_{y1}, \dots u_{x4}, u_{y4}\}^T$$

La matrice di rigidezza degli elementi bidimensionali era stata calcolata usando quattro punti di integrazione. Le deformazioni, e quindi le tensioni, degli stessi elementi sono state invece computate soltanto al centro di ogni elemento.

Per la deformazione e la tensione delle barre ho usato le seguenti formule:

$$\varepsilon_m = (l - l_0)/l_0$$

$$\sigma_m = E_m \varepsilon_m$$

dove l_0 ed l sono la lunghezza dell'asta iniziale e dopo l'applicazione del carico.

1.2 Analisi non lineari

Nelle analisi strutturali la nonlinearità viene solitamente gestita con metodi incrementali e iterativi (Fib (2008), Crisfield et al. (2012), Kotsovos (2015)). Uno dei più conosciuti è quello di Newton – Raphson, che illustro solo qualitativamente, non avendolo implementato nel programma:

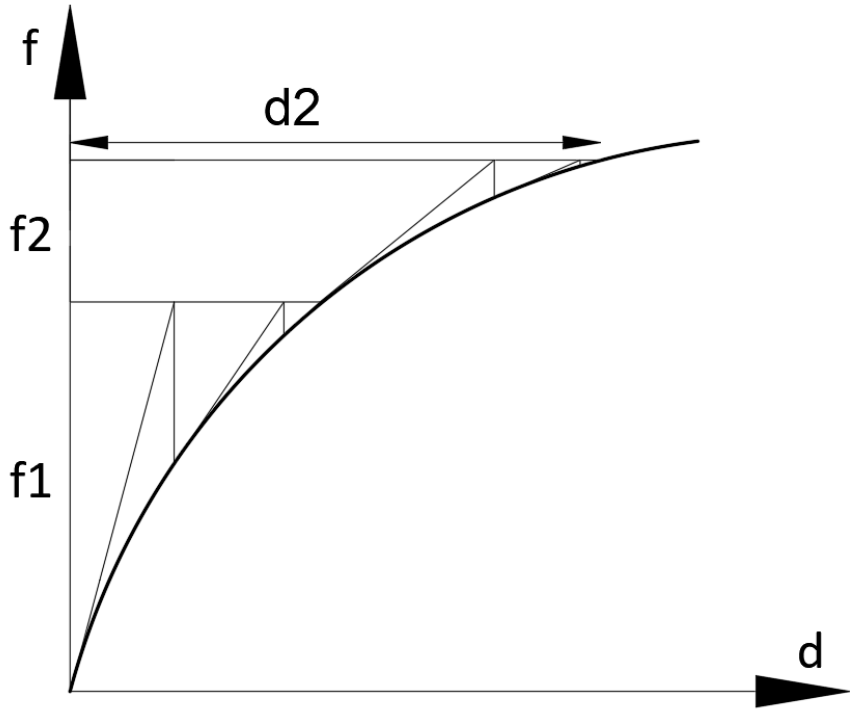

Figura 4 Metodo di Newton – Raphson: lo scostamento tra predizione e legge costitutiva del materiale può essere piccolo a piacere

Ogni step di carico viene eseguito con una o più iterazioni, fino al raggiungimento di uno scostamento sufficientemente piccolo (valore imposto di convergenza) tra deformazione predetta e legge costitutiva del materiale. Come si può vedere anche dal disegno, la matrice di rigidezza tangenziale della struttura viene aggiornata ad ogni iterazione.

Anche il libro di Kotsovos suggerisce l'uso di metodi incrementali e iterativi. In particolare ne vengono illustrati tre: quello di Newton – Raphson e altri due molto simili, ottenuti con piccole modifiche al NR stesso.

Nel programma è stato invece usato un metodo puramente incrementale. È conosciuto come metodo di Eulero in avanti (*forward Euler method*) o metodo di Eulero esplicito (vedi "Metodo di Eulero" (2020)), anche se la maggior parte dei riferimenti che ho consultato parla semplicemente di metodo puramente incrementale.

Non essendoci quindi iterazioni è probabilmente il metodo incrementale più semplice che esista:

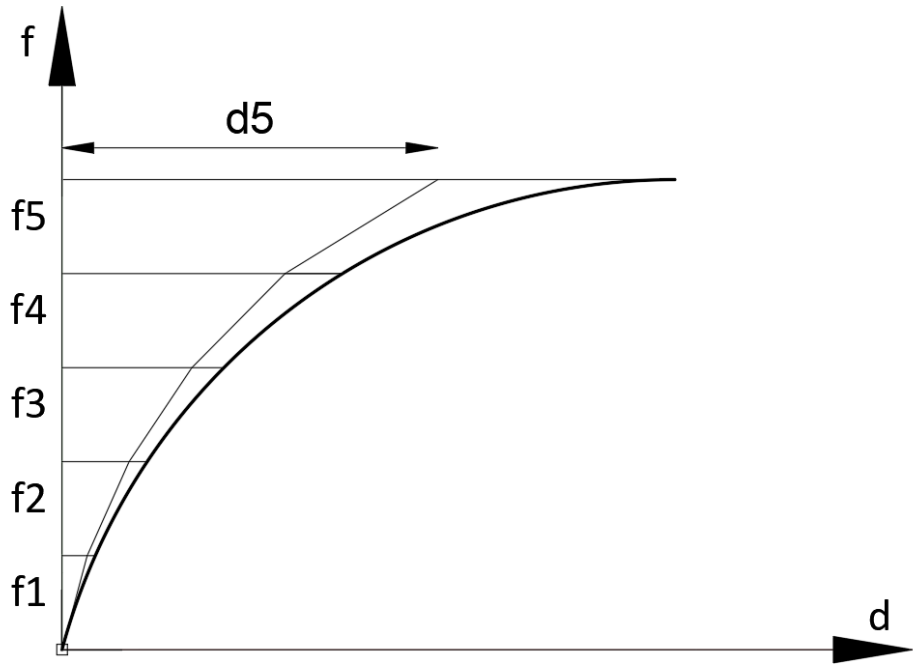

Figura 5 Metodo incrementale puro: è facile vedere lo scostamento tra curva numerica e reale nel caso in cui vengano usati soltanto pochi step di carico

Il riferimento grafico (anche nel caso del metodo di Newton – Raphson) è al caso unidimensionale, ma il metodo ha ovviamente validità generale.

Gli step di carico sono così eseguiti:

$$\{U\}_n = \{U\}_{n-1} + [K]^{-1}_{t,n-1}\{F\}_n$$

Per il primo step di carico (n=1), il vettore degli spostamenti $\{U\}_{n-1}$ e la matrice di rigidezza $[K]_{t,n-1}$ hanno pedice pari a zero: il vettore è ovviamente nullo mentre la matrice di rigidezza è quella iniziale della struttura.

Il pedice t alla matrice di rigidezza della struttura sta ad indicare che si tratta della rigidezza tangenziale. Il metodo consiste quindi nell'eseguire sempre la stessa analisi lineare, andando soltanto ad aggiornare, ad ogni step di carico, le matrici di rigidezza degli elementi e quindi della struttura: a volte sarà sufficiente modificare i valori di alcune proprietà dei materiali, altre volte (nel caso di elementi fessurati) bisognerà intervenire ulteriormente in maniera diretta sulle matrici costitutive.

Il calcolo delle deformazioni e delle tensioni nei singoli elementi avviene in maniera simile:

$$\{\varepsilon\}_n = \{\varepsilon\}_{n-1} + [B](\{u\}_n - \{u\}_{n-1})$$

$$\{\sigma\}_n = \{\sigma\}_{n-1} + [D]_{n-1}(\{\varepsilon\}_n - \{\varepsilon\}_{n-1})$$

$$\varepsilon_{mn} = \varepsilon_{m,n-1} + (l_n - l_{n-1})/l_0$$

$$\sigma_{mn} = \sigma_{m,n-1} + E_{m,n-1}(\varepsilon_n - \varepsilon_{n-1})$$

L'unica probabile applicazione che ho trovato di questo metodo puramente incrementale ad analisi strutturali su cemento armato è nel software commerciale ATENA, di Červenka Consulting. Il manuale di teoria del software (Červenka 2013) parla infatti di una *purely incremental formulation* per uno dei materiali (CC3DNonLinCementitious2).

Il vantaggio di questo metodo è a mio parere l'importante semplificazione del programma. Lo svantaggio, intuitivo, è che l'errore si cumula ad ogni step di carico. La quasi totalità dei riferimenti che ho consultato sull'uso di questo metodo in analisi agli elementi finiti (ad esempio Crisfield 2012, Ghaboussi et al. (2016) o lo stesso Kotsovos (2015)) parla in maniera molto sbrigativa di questo metodo. Viene dato per scontato, direttamente o indirettamente, che si tratta di un metodo inutilizzabile o comunque sconveniente rispetto ai metodi incrementali e iterativi.

L'idea di questa tesi è che il metodo possa invece raggiungere un buona approssimazione, o che sia quantomeno utilizzabile. La mia opinione va anche oltre perché personalmente intuisco che in ogni analisi si possa raggiungere un'approssimazione molto buona o anche trascurabile, sempre a fronte di uno sforzo computazionale ragionevole. Sto parlando in ogni caso delle analisi su cemento armato, o comunque su materiali aventi leggi costitutive debolmente nonlineari.

Per quanto riguarda dei valori quantitativi non sono riuscito a trovare riferimenti con informazioni sui risultati del metodo in analisi bidimensionali o tridimensionali, mentre ce ne sono diversi per il caso unidimensionale, come Bourne (2018) e Dawkins (2018). In questi elementari calcoli viene in effetti mostrato che il metodo può raggiungere buone/ottime approssimazioni, ma non si tratta di applicazioni ad elementi finiti.

Preferisco quindi riportare un paio di grafici ottenuti da me, anch'essi banali ma riguardanti direttamente la legge costitutiva del calcestruzzo o una curva ad essa molto simile. Non costituiscono ovviamente una dimostrazione del funzionamento del metodo ma ne possono dare un'idea. Il primo è il grafico di tre curve che approssimano una parabola, ottenute variando il numero di step di carico (10, 100 e 1000 step).

In questo caso il “carico” è in x, come si può vedere dal fatto che tutte le curve arrivano esattamente allo stesso valore di 0.0035:

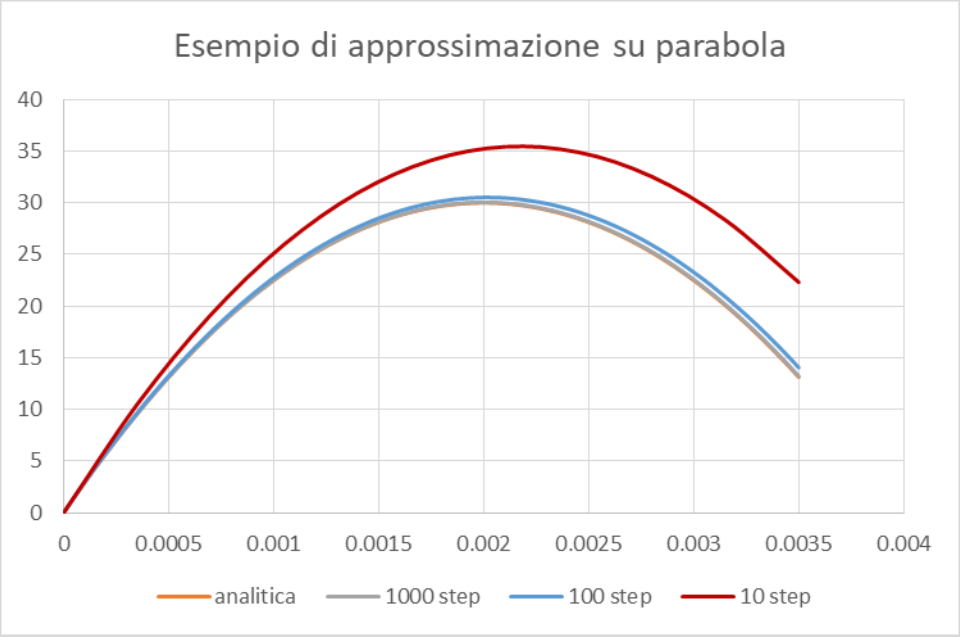

Figura 6 Metodo incrementale puro: esempio di applicazione su una curva parabolica

L’equazione della parabola è:

$$y = -7.5 * 10^6 x^2 + 3 * 10^4 x$$

Scelta semplicemente per avere valori tipici di un grafico σ – ε di calcestruzzo a compressione con σ in MPa. Come si può vedere, l’unica curva che si discosta marcatamente dall’analitica è quella con 10 step di carico. Quella con 100 step ha uno scostamento molto piccolo ma ancora visibile mentre quella con 1000 step è praticamente indistinguibile dalla curva analitica. Gli errori finali in y, rapportati al valore massimo della curva analitica (30), sono stati di circa 30%, 3% e 0.3%.

Il secondo è invece il confronto diretto con un’analisi unidimensionale effettuata con uno dei software commerciali che ha implementato il modello di Kotsovos (DIANA FEA). Tale programma usa un metodo di calcolo iterativo e la tolleranza scelta è stata talmente bassa (inferiore a 10^{-6} sulla deformazione) da far sì che la curva ottenuta possa essere considerata a tutti gli effetti come la curva teorica del materiale. Entrambe le analisi constano di 25 step di carico.

Non ho fatto variare in questo caso il numero di step (mi sono cioè fermato alla prima delle analisi che ho fatto) per quanto vedremo dai risultati:

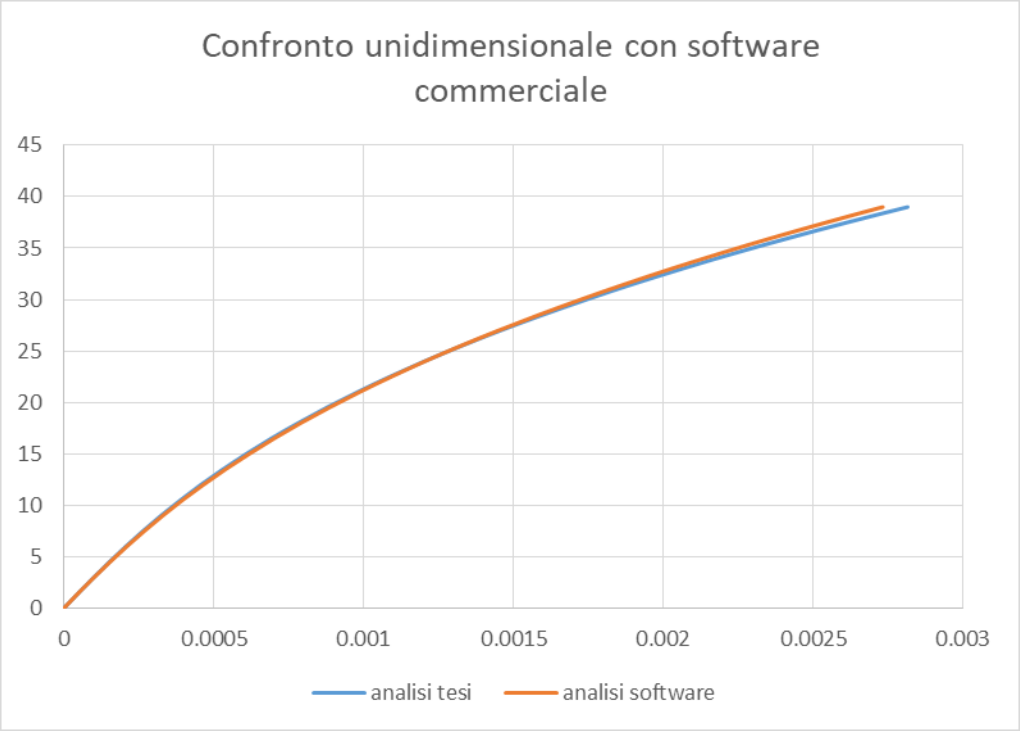

Figura 7 Metodo incrementale puro: esempio di applicazione su legge costitutiva del calcestruzzo di Kotsovos

Come si può vedere, succede una cosa inaspettata e cioè che la curva approssimata con metodo puramente incrementale si trova "a destra" di quella ottenuta col software (entrambe le analisi raggiungono invece la rottura al termine dello stesso step di carico). Questo dovrebbe essere dovuto a quanto segue. Il metodo puramente incrementale non prevede l'uso diretto della legge costitutiva del calcestruzzo ma solo delle sue derivate, ossia delle rigidezze tangenziali. Viceversa, un programma che fa uso di un metodo incrementale e iterativo (quindi predittivo e correttivo) usa le rigidezze tangenziali per effettuare le predizioni ma riesce ad arrivare alla legge costitutiva effettuando le correzioni. Ora, il libro di Kotsovos fornisce delle formule, per le rigidezze tangenziali, che sono state ottenute *separatamente*, cioè non per derivazione diretta, rispetto a quelle che esprimono la legge costitutiva, pertanto la loro primitiva *non coincide* perfettamente con la legge costitutiva stessa. Tale scostamento è aleatorio e può quindi portare la legge costitutiva a trovarsi anche "a sinistra", come nel grafico appena riportato, rispetto a una curva ottenuta per inviluppo di tangenti.

Ad ogni modo considero questo risultato come un ottimo indicatore della bontà del metodo (seppur limitatamente al caso unidimensionale): persino in un'analisi così semplice, e con pochi step di carico, la componente di errore dovuta all'utilizzo del metodo approssimato, ovviamente presente, è più piccola di quella dovuta alle altre incertezze di modello.

I risultati presentati al capitolo sugli esempi di calcolo possono invece dare, indirettamente, informazioni sul funzionamento del metodo nel caso bidimensionale.

Riassumo adesso il funzionamento del programma nel seguente diagramma di flusso:

Calcolo della matrice di rigidezza iniziale:

$$[K]_{t0} = f(geometria, materiali\ (E_0, \nu_0, E_{m0}), condizioni\ al\ contorno)$$

↓

Applicazione dell'n-esimo step di carico e calcolo del relativo stato tensionale:

$$\{U\}_n = \{U\}_{n-1} + [K]^{-1}_{t,n-1}\{F\}_n$$

$$\{\varepsilon\}_n = [B]\{u\}_n$$

$$\{\sigma\}_n = [D]_{n-1}\{\varepsilon\}_n$$

$$\varepsilon_{mn} = (l_n - l_0)/l_0$$

$$\sigma_{mn} = E_{m,n-1}\varepsilon_{mn}$$

↓

Controllo dello stato tensionale per vedere se si è raggiunta rottura:

$\tau_{0n} > \tau_{0un}$ oppure $|\sigma_{mn}| > \sigma_{ult}$

SI: STOP

NO: vai avanti

↓

Calcolo delle rigidezze tangenziali in funzione dello stato tensionale:

$$E_n, \nu_n = f(\{\sigma\}_n)$$

$$[D]_n = f(E_n, \nu_n)$$

$$E_{mn} = f(\sigma_{mn})$$

↓

Controllo dello stato tensionale per vedere se si è raggiunta fessurazione:

SI: Modifica delle matrici costitutive degli elementi fessurati:

$$[D]_n = f(stato\ di\ fessurazione, E_n, \nu_n, \beta)$$

NO: vai avanti

↓

Calcolo della matrice di rigidezza tangenziale:

$$[K]_{tn} = f(geometria, materiali\ ([D]_n, E_{mn}), condizioni\ al\ contorno)$$

TORNA A: applicazione dell'n-esimo step di carico

All’interno del diagramma di flusso, le matrici [K] e i vettori {U} ed {F} si riferiscono all’intera struttura. Tutte le altre grandezze si riferiscono ai singoli elementi. Per evitare di appesantire la notazione, non ho aggiunto nessun pedice per fare questa distinzione.

1.3 Termine delle analisi

Le analisi terminano quando almeno un elemento raggiunge la rottura. Per gli elementi in calcestruzzo si tratta della superficie di rottura descritta al capitolo 2, per le barre in acciaio si tratta della resistenza ultima della legge costitutiva bilineare.

È evidente che questo criterio non può essere usato per analisi di pushover su strutture iperstatiche (vedi ad esempio Parducci (2016) o Massonet e Save (2008)). Infatti, per tali strutture, il raggiungimento della rottura in un determinato punto della stessa non porta, in genere, al collasso globale.

Anche per strutture isostatiche in cemento armato non è sempre detto che in questo modo si riesca a catturare correttamente la rottura della struttura (Parducci (2016)). È infatti possibile che la rottura di alcune barre o del calcestruzzo, in un punto della struttura, possa non portare immediatamente al collasso globale. Tuttavia dovrebbero essere casi abbastanza rari.

Il criterio in questione è invece sempre valido per analisi di verifica (CEN (2004) e Aicap (2008)), su qualsiasi tipo di struttura, isostatica o iperstatica. Queste analisi servono infatti proprio a controllare che la rottura non si raggiunga in nessun punto della struttura (verifica soddisfatta). Se c'è invece almeno un punto in cui la rottura è stata raggiunta la verifica non è soddisfatta e l'analisi può terminare perché non ha interesse capire cosa succede dopo la rottura avvenuta in quel determinato punto della struttura.

Capitolo

2 Modello di calcolo dei materiali

In questo capitolo descrivo il modo in cui sono stati modellati i materiali, ossia il calcestruzzo e l'acciaio. Il primo è stato modellato secondo quanto descritto in Kotsovos (2015), l'acciaio con un comportamento elastoplastico incrudente descritto da una legge costitutiva bilineare.

Nell'introduzione abbiamo accennato a dei problemi nello stato dell'arte delle analisi nonlineari, ad elementi finiti, sul cemento armato: la mancanza di modelli di calcestruzzo che abbiano validità generale (per qualsiasi tipo di analisi/struttura), e il fatto che le analisi richiedano fasi preliminari abbastanza elaborate di calibrazione. Kotsovos afferma di aver superato questi problemi, e di aver sviluppato un modello di calcestruzzo valido per ogni tipo di analisi (statiche e dinamiche), su qualsiasi tipo di struttura, e senza necessità di alcun tipo di calibrazione. Una delle idee fondamentali del libro in questione è che il calcestruzzo è un materiale fragile, in rottura sia a compressione che a trazione (l'autore giustifica quest'osservazione sulla base di diversi esperimenti). La quasi totalità degli altri modelli non prevede invece questo comportamento fragile, ed è fondamentalmente alla risoluzione di questa incomprensione che l'autore attribuisce il miglior funzionamento del suo modello.

Per ogni paragrafo faccio prima un cenno sullo stato dell'arte della modellazione, con alcuni esempi, e poi descrivo il modo in cui il determinato comportamento o fenomeno è stato implementato all'interno del programma di calcolo.

Mi è sembrato importante aggiungere il paragrafo finale sul comportamento a taglio anche se non riguarda direttamente la modellazione dei materiali ma il comportamento dell'intera struttura.

2.1 Calcestruzzo non fessurato

I modelli di calcestruzzo solitamente implementati dai programmi di calcolo commerciali sono numerosi. Per quanto riguarda le leggi costitutive del calcestruzzo a compressione, solo per fare alcuni esempi abbiamo il comportamento elastico perfettamente plastico, elastico con tratto plastico incrudente (per questi primi due vedi ad esempio Ottosen e Ristinmaa (2005)), parabolico (Manie et al. (2017)), i modelli ricavati dalle normative, come l'eurocodice 2 (CEN, *Eurocode 2 – Part 1-1* (2004)), e quelli proposti da alcuni autori, come Maekawa et al. (2019) e Kotsovos (2015).

Prima di descrivere il modello di calcestruzzo implementato nel programma di questa tesi (Kotsovos (2015)), ne descrivo altri due: quello a comportamento parabolico e quello dell'eurocodice 2.

I primi due modelli, di cui mi limiterò a una descrizione qualitativa, hanno i seguenti andamenti:

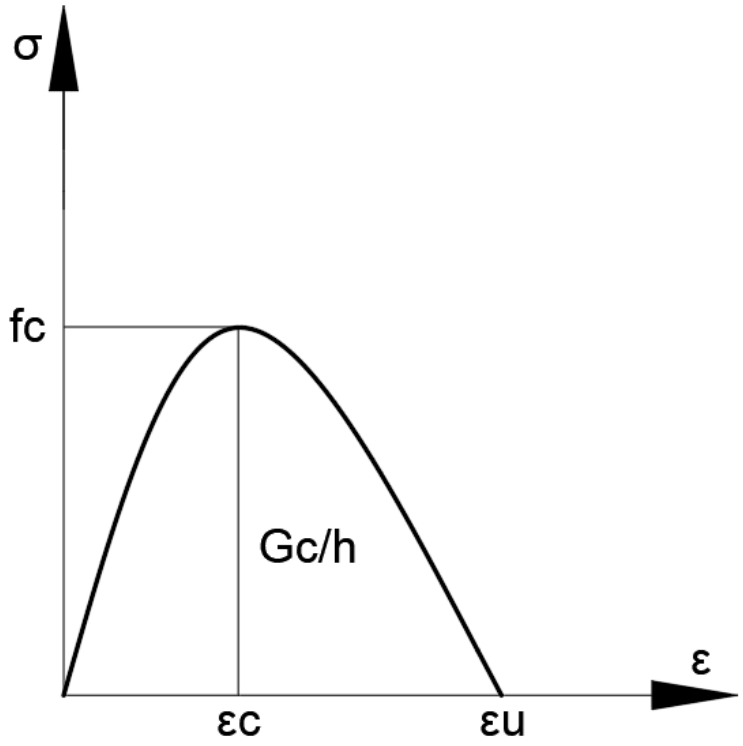

Figura 8 Grafico qualitativo di una legge costitutiva di calcestruzzo a compressione con comportamento parabolico

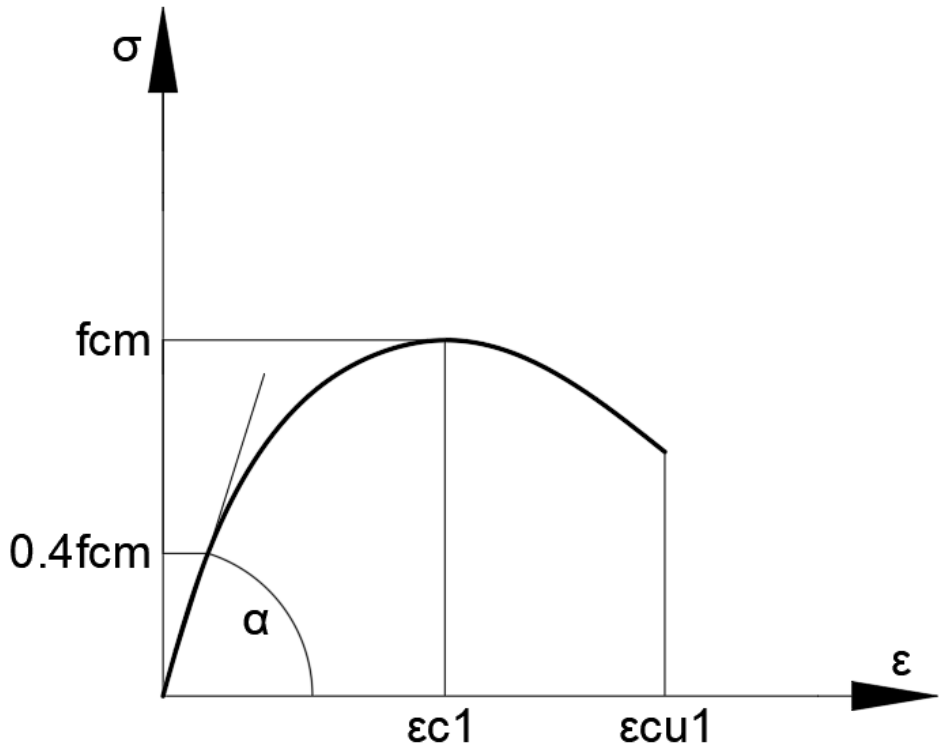

Figura 9 Grafico qualitativo della legge costitutiva di un calcestruzzo a compressione suggerito dall'eurocodice 2

Entrambe le curve hanno una resistenza a compressione massima e dei valori di deformazione di picco e ultimo.

Il comportamento parabolico preso in esempio ha una resistenza ultima nulla ed è caratterizzato da un'energia dissipata (area sottesa dalla curva) pari a Gc/h, con Gc energia di rottura a compressione ed h lunghezza caratteristica dell'elemento (inteso come singolo elemento di un'analisi ad elementi finiti).

La curva suggerita dall'eurocodice 2 ha una resistenza ultima generalmente diversa da zero ed un modulo di rigidezza elastico pari alla pendenza della retta che interseca la curva in corrispondenza di σ pari a 0.4 fcm.

Da notare infine che entrambe le curve prese ad esempio hanno un tratto di softening. Generalmente i modelli di calcestruzzo *hanno* dei tratti di softening. Una delle eccezioni è proprio il modello di Kotsovos usato nel programma di questa tesi. Vedremo al paragrafo sulla rottura per splitting il modo in cui l'autore giustifica tale comportamento.

Descrivo quindi il calcestruzzo modellato da Kotsovos. In assenza di fessure si tratta di un materiale isotropo, nonlineare ed elasto-plastico. L'andamento qualitativo della legge costitutiva è del tipo:

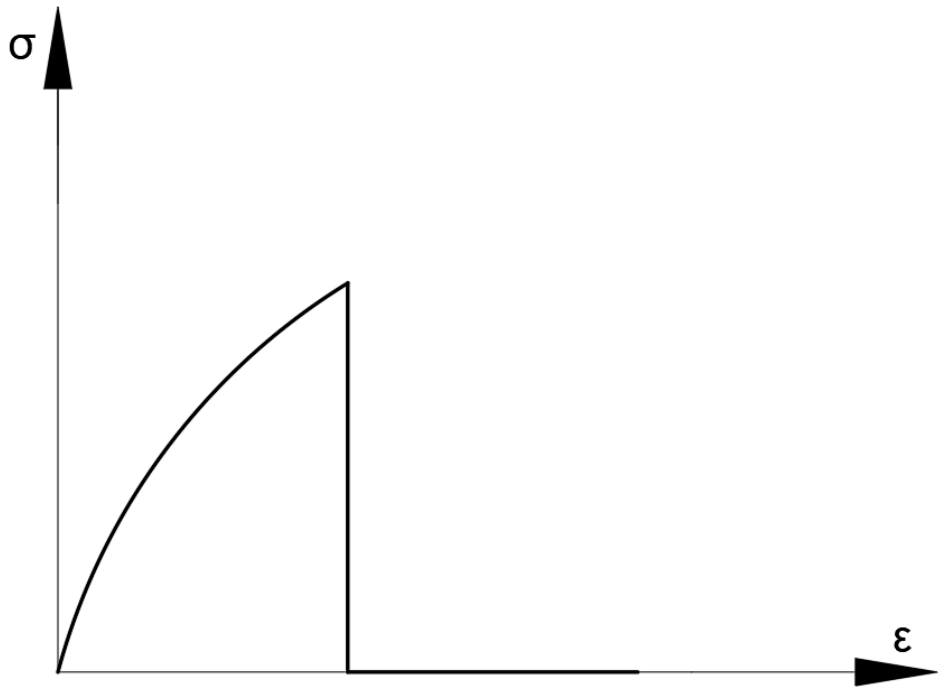

Figura 10 Grafico qualitativo della legge costitutiva del calcestruzzo modellato da Kotsovos, in assenza di fessure

Innanzi tutto faccio notare che nel software le tensioni e le deformazioni di compressione sono negative, ma per semplicità ho continuato a disegnarle come fossero positive. Caricato in una sola direzione, la tensione di rottura sarebbe la resistenza a compressione cilindrica. Nel caso tridimensionale (e bidimensionale) tale tensione non è definita a priori perché dipende dall'intero stato tensionale.

Il modello in questione prevede innanzi tutto il calcolo dei seguenti parametri, tutti dipendenti dalla sola resistenza a compressione cilindrica f_c:

$$\mathrm{A} = 0.516$$

$$\mathrm{b} = 2.0 + 1.81 * 10^{-8} {f_c}^{4.461}$$

$$\mathrm{C} = 3.573$$

$$\mathrm{d} = 2.12 + 0.0183 f_c$$

$$K_e = 11000 + 3.2 {f_c}^2$$

$$G_e = 9224 + 136 f_c + 3296 * 10^{-15} {f_c}^{8.273}$$

I primi quattro parametri non hanno alcun significato fisico e serviranno più avanti per calcolare il modulo elastico e il coefficiente di Poisson del calcestruzzo in funzione dello stato interno di tensione dello stesso. Gli ultimi due parametri sono il modulo di bulk e il modulo di taglio iniziali. Il range di validità delle formule in termini di resistenza del calcestruzzo è di 15 – 65 MPa. Al di là di questi estremi i parametri in questione rimangono costanti e uguali ai valori in corrispondenza di 15 e 65 MPa. Tutte queste formule, e le due che seguiranno per il calcolo di K_t e G_t, sono state ottenute a partire da risultati su test sperimentali.

Le seguenti formule dell'elasticità lineare permettono di passare da K_e e G_e ad E_e e ν_e:

$$\mathrm{E}_e = 9\mathrm{K}_e\mathrm{G}_e/(3\mathrm{K}_e + \mathrm{G}_e)$$

$$\nu_e = (3\mathrm{K}_e - 2\mathrm{G}_e)/(6\mathrm{K}_e + 2\mathrm{G}_e)$$

Noti il modulo di Young e il coefficiente di Poisson iniziali del calcestruzzo possiamo effettuare il primo step di carico. Esso darà, tra i risultati, il tensore degli stress [σ] (uno per ogni elemento di calcestruzzo), che verrà elaborato nel seguente modo:

$$I_1 = \sigma_x + \sigma_y$$

$$I_2 = \sigma_x\sigma_y - \tau_{xy}^2$$

Dove I_1 e I_2 sono gli invarianti del tensore degli stress nel caso di tensione piana (per tale stato di sollecitazione il terzo invariante è invece sempre pari a zero). Gli invarianti permettono di calcolare le tensioni ottaedrali, ossia la tensione idrostatica, deviatorica e l'angolo rotazionale (vedi in appendice per il sistema di riferimento ottaedrale):

$$\sigma_0 = I_1/3$$

$$\tau_0 = (2{\sigma_0}^2 - (2/3)I_2)^{1/2}$$

$$\Theta = \mathrm{acos}(\sigma_0/\sqrt{2}\tau_0)$$

Con la formula dell'ultimo valore semplificata nuovamente dallo stato di tensione piana ($\sigma_3 = 0$).

Questi ultimi tre valori permettono di calcolare, a loro volta, il modulo di bulk e il modulo di taglio tangenziali:

$$K_t = \frac{K_e}{1 + \mathrm{bA}\left(\frac{\sigma_0}{f_c}\right)^{b-1}} \; per \; \frac{\sigma_0}{f_c} \leq 2$$

$$K_t = \frac{K_e}{1 + \mathrm{bA} * 2^{b-1}} \; per \; \frac{\sigma_0}{f_c} > 2$$

$$G_t = G_e/(1 + \mathrm{dC}\left(\frac{\tau_0}{f_c}\right)^{d-1})$$

Non ho implementato la seconda formula relativa al K_t (quella per $\sigma_0 / f_c > 2$). Credo tuttavia sia molto improbabile, se non impossibile, arrivare a tali valori di σ_0 (maggiori di due f_c) per stati di tensione piana, e sicuramente non accade per nessuno degli esempi di calcolo della tesi.

Sulla base di formule dell'elasticità lineare già usate:

$$\mathrm{E}_t = 9\mathrm{K}_t\mathrm{G}_t/(3\mathrm{K}_t + \mathrm{G}_t)$$

$$\nu_t = (3\mathrm{K}_t - 2\mathrm{G}_t)/(6\mathrm{K}_t + 2\mathrm{G}_t)$$

Ottenendo quindi i valori tangenziali della rigidezza e del coefficiente di Poisson, che permetteranno di effettuare il successivo step di carico.

Da notare la validità della procedura anche per problemi tridimensionali. Lo stato di tensione piana ha soltanto permesso di semplificare le formule relative al calcolo degli invarianti e delle tensioni ottaedrali.

Tutti questi calcoli verranno in effetti svolti anche per gli elementi fessurati, e le tensioni ottaedrali verranno usate per individuare la rottura per splitting esattamente come per gli elementi non fessurati. Tuttavia, i valori finali di rigidezza e coefficiente di Poisson tangenziali verranno usati, negli elementi fessurati, soltanto con successiva modifica delle matrici costitutive volta a modellare la fessurazione.

2.2 Fessurazione del calcestruzzo

Esistono fondamentalmente due approcci con cui la fessurazione può essere modellata (Kotsovos (2015) e Crisfield et al. (2012)): l'approccio discreto (*discrete*) e quello diffuso (*smeared*).

L'approccio discreto consiste nel fatto che la fessurazione viene modellata come una vera e propria separazione tra elementi:

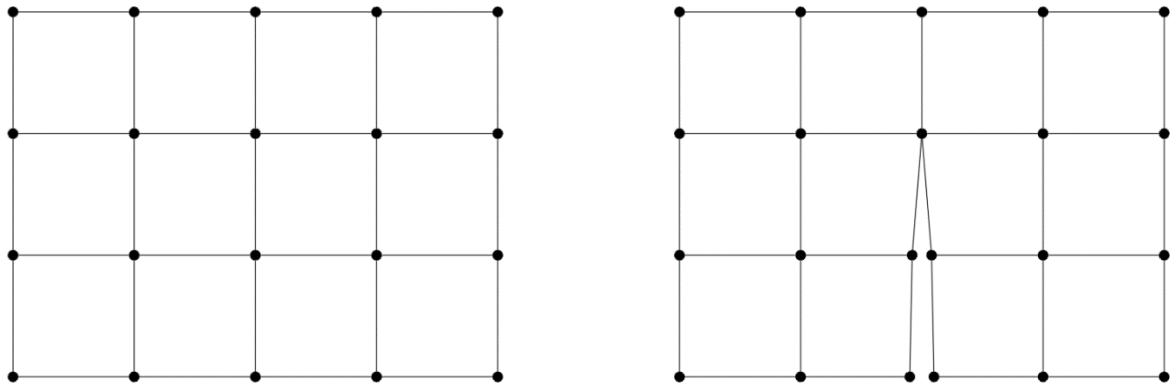

Figura 11 Esempio di fessurazione discreta: a sinistra la mesh iniziale, a destra quella con la fessura

Si può notare come in prossimità dell'apertura i nodi vengano duplicati.

Nell'approccio diffuso la fessura viene invece modellata come perdita di rigidezza, totale o parziale, da parte dell'elemento:

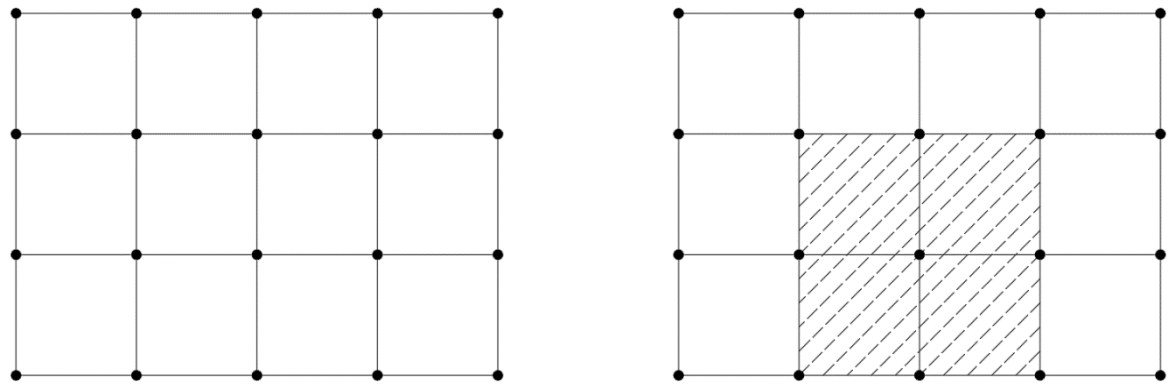

Figura 12 Esempio di fessurazione diffusa: si può notare che la mesh non cambia

In questo caso ho campito con linee tratteggiate gli elementi che hanno perso della rigidezza. Per analogia con la figura d'esempio che mostra una fessura discreta verticale, si tratterà di una perdita di rigidezza lungo l'asse orizzontale, cioè perpendicolare alla formazione della fessura reale. Questo tipo di fessurazione è detta *diffusa* proprio perché la fessura è appunto diffusa su uno, o più, interi elementi.

Per le analisi strutturali su cemento armato la modellazione della fessurazione diffusa è diventata di gran lunga di maggior uso. La modellazione discreta viene solitamente usata quando ci si vuole concentrare specificatamente sull'aspetto relativo alla fessurazione. L'approccio usato nel programma di questa tesi è pertanto quello diffuso e descrivo adesso il modo in cui è stato implementato.

La base di partenza della modellazione rimane il libro di Kotsovos, ma rispetto al libro la fessurazione è stata modellata nel programma con delle semplificazioni. Riassumo quindi prima la modellazione del libro, poi descrivo le differenze.

Nel libro è stato usato un modello di fessurazione discreta, con orientazione delle fessure che rimane costante dopo che queste si sono formate (*fixed smeared crack model*).

Si arriva alla formazione di una fessura quando lo stato tensionale in un elemento di calcestruzzo raggiunge la superficie di rottura dello stesso e almeno una delle tensioni principali è di trazione. Descriverò la superficie di rottura del calcestruzzo direttamente al paragrafo sulla rottura per splitting.

La fessura si forma in direzione ortogonale a quella in cui agisce la trazione massima. In due dimensioni, il sistema di riferimento locale della fessura è quindi semplicemente ruotato di un certo angolo α rispetto al sistema locale dell'elemento:

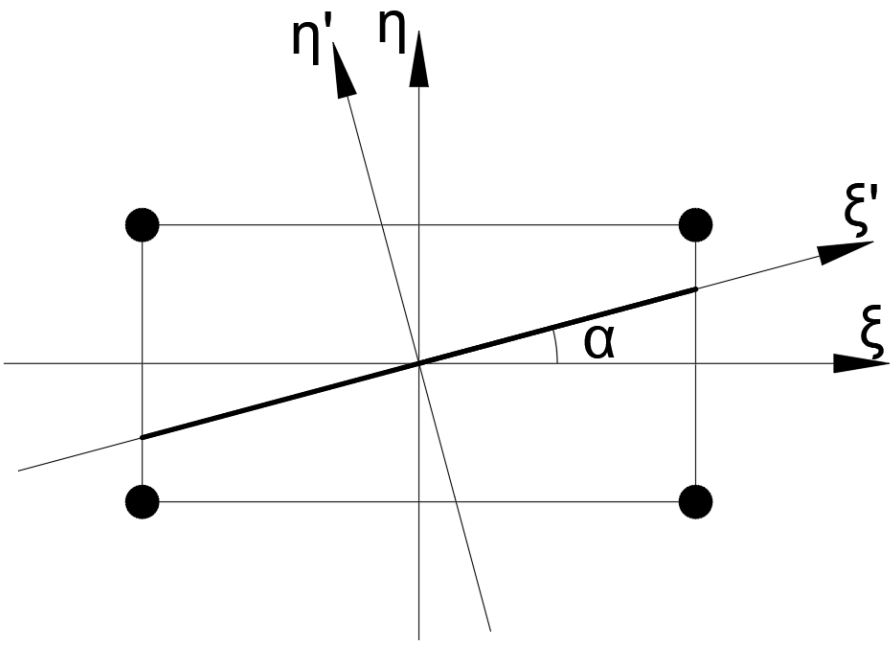

Figura 13 Sistema di riferimento locale di una fessura. Essa è rappresentata dalla linea spessa lungo l'asse ξ'

Kotsovos modella le fessure con comportamento fragile. In direzione ortogonale alla fessura vale quindi una legge costitutiva del tipo:

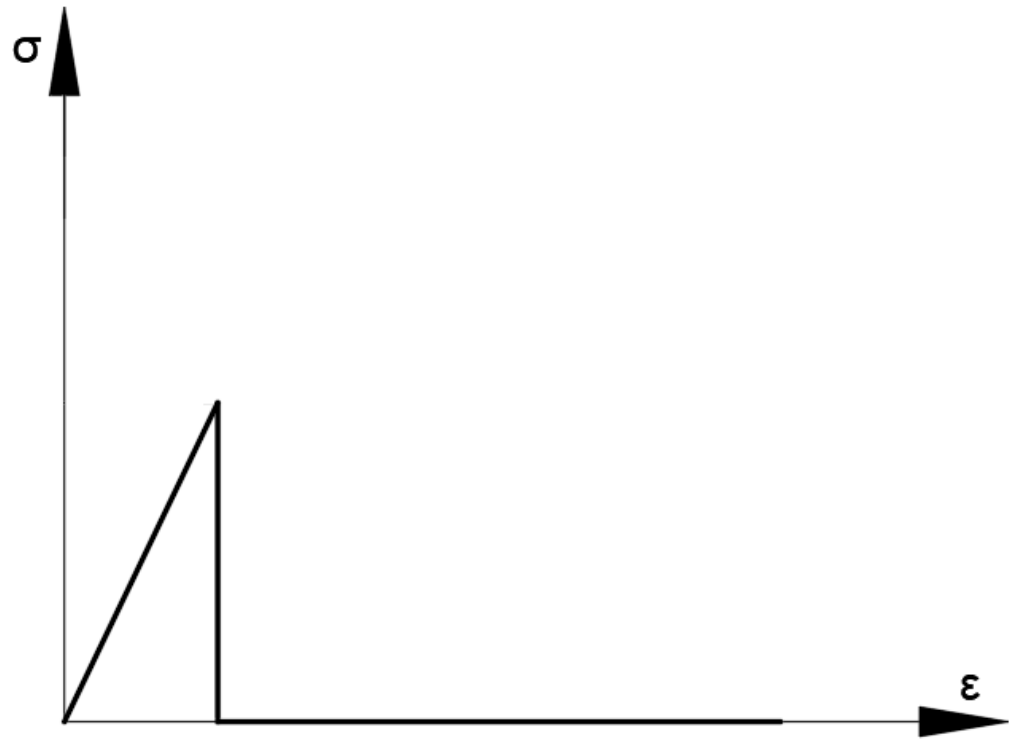

Figura 14 Comportamento fragile di formazione di una fessura

Nel caso uniassiale la tensione di trazione massima sarebbe pari a f_{ct}; nel caso bidimensionale (e in quello a tre dimensioni) non è definita a priori perché dipende dall'intero stato tensionale.

La formazione di una fessura consiste quindi innanzi tutto nel rilascio della tensione di trazione ortogonale ad essa che l'elemento aveva accumulato fino a quello step di carico. Tale modellazione dovrebbe essere dello stesso tipo della rottura dei materiali a cui ho fatto un cenno nelle conclusioni.

Per quanto riguarda invece la matrice costitutiva, la rigidezza assiale in direzione ortogonale alla fessura viene meno, e la rigidezza a taglio in direzione parallela ad essa diventa una percentuale della rigidezza a taglio dell'elemento non fessurato. Nel sistema locale:

$$[D]' = \frac{E_t}{1-\nu_t^2}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.0001 & 0 \\ 0 & 0 & \beta\dfrac{1-\nu}{2} \end{bmatrix}$$

lo shear retention factor β esprime la percentuale residua di rigidezza a taglio. Il motivo per cui questa rigidezza non viene del tutto meno è legato all'effetto di ingranamento degli aggregati. Kotsovos raccomanda di scegliere per β un valore compreso tra 0.1 e 0.5 e suggerisce infine proprio il valore di

0.1. In ogni caso, il motivo per cui β non deve essere troppo piccolo è numerico: valori di β troppo piccoli (ad esempio 0.01) portano a un mal condizionamento della matrice di rigidezza dell'elemento, il motivo per cui non deve essere troppo grande è fisico: valori di β maggiori di 0.5 sono poco realistici. Il pedice t alla rigidezza e al coefficiente di Poisson indica invece che si tratta delle quantità tangenziali, di cui è stato mostrato il calcolo nel relativo paragrafo.

Essendo valida nel suo sistema di riferimento, la matrice costitutiva appena riportata deve essere ruotata tramite la matrice $[T_\varepsilon]$, per essere valida nel sistema di riferimento dell'elemento:

$$[D] = [T_\varepsilon]^T [D]' [T_\varepsilon]$$

$$[T_\varepsilon] = \begin{bmatrix} l_1^2 & m_1^2 & l_1 m_1 \\ l_2^2 & m_2^2 & l_2 m_2 \\ 2l_1 l_2 & 2m_1 m_2 & l_1 m_2 + l_2 m_1 \end{bmatrix}$$

Con i coseni direttori l_1 ed m_1 che esprimono l'orientazione di ξ' rispetto a ξ ed η, ed l_2 ed m_2 di η' rispetto a ξ ed η.

Per il caso bidimensionale, Kotsovos prevede che si possano formare al massimo due fessure, con la seconda che non necessariamente deve essere ortogonale alla prima. Nel caso di formazione della seconda fessura, la matrice costitutiva diventa:

$$[D] = \frac{E_t}{1 - \nu_t^2} \begin{bmatrix} 0.0001 & 0 & 0 \\ 0 & 0.0001 & 0 \\ 0 & 0 & \beta \dfrac{1-\nu}{2} \end{bmatrix}$$

con perdita totale di rigidezza assiale in entrambe le direzioni. Questa volta non c'è bisogno di rotazione, in quanto la matrice costitutiva appena riportata è già valida nel sistema di riferimento dell'elemento.

Riassumo adesso il modo in cui la fessurazione è stata modellata all'interno del programma. Sono state adottate due importanti semplificazioni: la resistenza a trazione del calcestruzzo viene trascurata (posta pari a un decimo del rispettivo valore fornito dall'Eurocodice) e le fessure hanno la possibilità di formarsi soltanto lungo gli assi, quindi non in direzione qualsiasi (per questa seconda scelta il nome inglese del modello sarebbe quindi *orthogonal fixed smeared crack model*).

È stato appena detto che la formazione di una fessura comporta il rilascio, da parte dell'elemento, della tensione di trazione ortogonale ad essa. La prima delle due semplificazioni permette di evitare questo tipo di modellazione. La seconda semplificazione permette invece di non dover ruotare la matrice costitutiva dal sistema di riferimento della fessura a quello dell'elemento. Può sembrare un'operazione banale, ma quando ci ho provato non è stato affatto semplice verificare il corretto

funzionamento del programma. D'altra parte, la possibilità di formazione delle fessure soltanto lungo gli assi permette di avere un codice che fornisce risultati approssimati ma di cui è molto facile verificare il funzionamento.

In maniera diretta, il fatto di trascurare la resistenza a trazione del calcestruzzo non dovrebbe deviare di molto i risultati dalla realtà. Tale resistenza è infatti molto bassa (un ordine di grandezza inferiore a quella a compressione). Questa semplificazione influisce però anche sull'orientazione che raggiungono le tensioni, e quindi le fessure, all'interno dell'elemento.

A mio parere, sia gli effetti diretti, che quello indiretto appena accennato, delle due semplificazioni non dovrebbero alterare in maniera inaccettabile i risultati. Tuttavia, queste semplificazioni potrebbero influire in maniera significativa sul comportamento a traliccio di resistenza al taglio (che però, come vedremo, secondo Kotsovos in realtà non si forma). In ogni caso, ritengo che l'implementazione del modello completo di fessurazione (in particolare della fessurazione obliqua) sia uno dei più importanti step nell'eventuale prosecuzione di questo lavoro.

Vediamo adesso la procedura computazionale. Essa prevede innanzi tutto il calcolo del valore di resistenza a trazione del calcestruzzo fornito dall'Eurocodice 2:

$$f_{ct} = 0.3(f_c - 8)^{2/3}$$

Non avendone avuto bisogno negli esempi di calcolo, non ho implementato la formula relativa ai calcestruzzi di resistenza maggiore di C50/60. La formula appena riportata vale quindi solo per f_c fino a 58 MPa ($f_c = f_{ck} + 8$ (MPa)).

Dopodichè viene calcolata la tensione principale massima, in funzione dello stato tensionale:

$$\sigma_1 = \frac{1}{2}(\sigma_x + \sigma_y) + \left(\frac{1}{4}(\sigma_x - \sigma_y)^2 + \tau_{xy}^2\right)^{\frac{1}{2}}$$

Se tale tensione è maggiore della metà di f_{ct} la matrice costitutiva viene così modificata:

$$\sigma_1 > \frac{1}{2} f_{ct} \Rightarrow [D] = \frac{E_t}{1 - \nu_t^2} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \beta \frac{1-\nu}{2} \end{bmatrix}$$

Diminuisce quindi la sola rigidezza a taglio. Il motivo di questa modellazione è legato alla semplificazione sulle fessure che si possono formare solo lungo gli assi. Tale semplificazione potrebbe portare infatti, in alcuni casi, a sovrastimare la rigidezza a taglio dell'elemento.

Quando invece le tensioni σ_x o σ_y (o entrambe) superano un decimo di f_{ct} la matrice costitutiva diventa:

$$\sigma_x > \frac{1}{10} f_{ct} \Rightarrow [D] = \frac{E_t}{1-\nu_t^2} \begin{bmatrix} 0.0001 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \beta \frac{1-\nu}{2} \end{bmatrix}$$

$$\sigma_y > \frac{1}{10} f_{ct} \Rightarrow [D] = \frac{E_t}{1-\nu_t^2} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.0001 & 0 \\ 0 & 0 & \beta \frac{1-\nu}{2} \end{bmatrix}$$

$$\sigma_x, \sigma_y > \frac{1}{10} f_{ct} \Rightarrow [D] = \frac{E_t}{1-\nu_t^2} \begin{bmatrix} 0.0001 & 0 & 0 \\ 0 & 0.0001 & 0 \\ 0 & 0 & \beta \frac{1-\nu}{2} \end{bmatrix}$$

Con fessure che si formano, rispettivamente, lungo y, lungo x e lungo entrambi gli assi.

Dal punto di vista concettuale, il fatto di non aver implementato la possibilità di fessure in direzione qualsiasi ha anche un'altra conseguenza. Kotosovs utilizza la superficie di rottura del calcestruzzo anche per individuare la formazione delle fessure. Il programma confronta invece direttamente i valori di tensione in x e in y con il valore di resistenza a trazione utilizzato.

2.3 Splitting del calcestruzzo

La rottura per splitting (Kotsovos (2015) e Guo et al. (1995)) non è stata modellata all'interno del programma, quindi la descriverò solo qualitativamente e illustrerò il modo in cui essa viene individuata. Si tratta di una rottura a compressione, e consiste nella completa e immediata perdita di rigidezza e tensioni da parte dell'elemento.

Il modo più completo per individuare una rottura per splitting è quello di utilizzare una superficie di rottura:

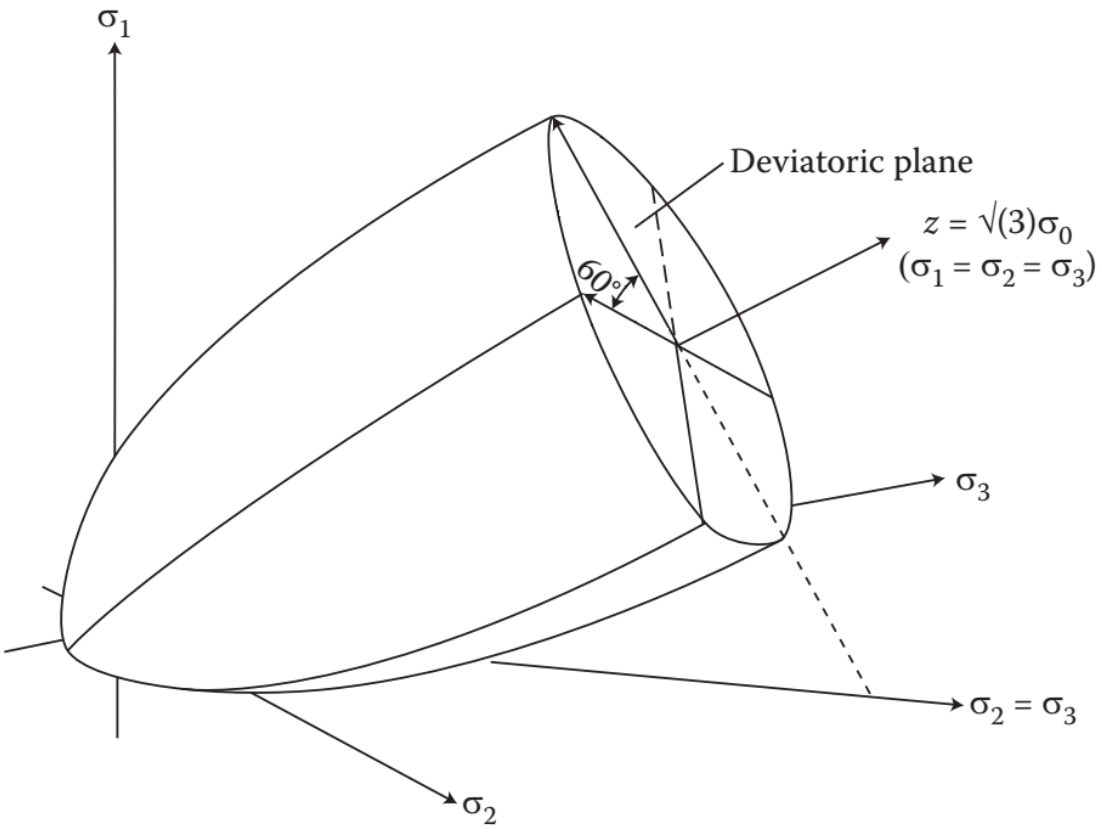

Figura 15 Rappresentazione qualitativa della superficie di rottura del calcestruzzo

La superficie di rottura del calcestruzzo appena riportata è tratta dal libro di Kotsovos, ma le sue caratteristiche sono del tutto generali. Si tratta di una superficie concava rispetto all'asse z, con area infinita, ed avente tre piani di simmetria, ben visibili osservando il piano deviatorico:

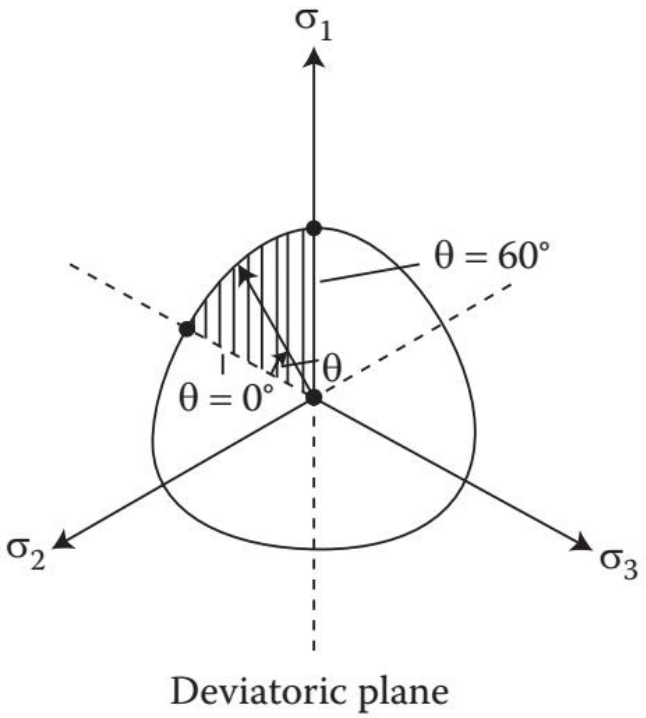

Figura 16 Vista della superficie di rottura del calcestruzzo dal piano deviatorico

I tre piani di simmetria sono perpendicolari al piano deviatorico e lo intersecano in corrispondenza delle tre linee tratteggiate in figura.

Il motivo per cui la superficie di rottura ha area infinita è dovuto al fatto che teoricamente il calcestruzzo è in grado di sopportare qualsiasi tensione idrostatica di compressione. E come si vede anche dalla prima figura, le tensioni deviatoriche ammesse aumentano al crescere della tensione idrostatica a cui il calcestruzzo è soggetto.

Nel modello di Kotsovos, la rottura per splitting è di tipo fragile, priva di tratti di softening. Questo comportamento è una particolarità del modello, ben visibile nell'andamento qualitativo della legge costitutiva, che rappresento di nuovo:

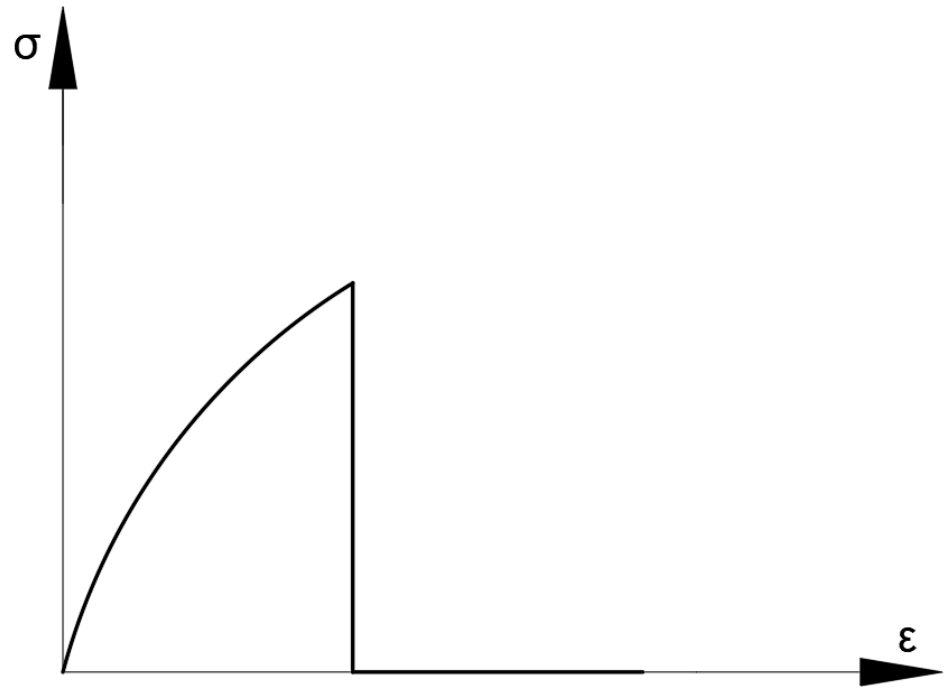

Figura 17 Grafico qualitativo della legge costitutiva del calcestruzzo di Kotsovos, con rottura per splitting

Kotsovos giustifica questo risultato sulla base di test a compressione cilindrici effettuati con piastre di applicazione del carico aventi diversa rigidezza. Al diminuire di tale rigidezza, e quindi dell'effetto di confinamento per attrito fornito dalla piastra al provino, il comportamento del materiale tende a essere sempre più fragile. La sua conclusione è che, *at a material level*, il calcestruzzo è fragile.

Per quanto riguarda il libro, la superficie introdotta in questo paragrafo viene usata per individuare sia la formazione delle fessure che la rottura per splitting. È necessario quindi osservare i valori delle tensioni principali: se sono tutti e tre di segno negativo (compressione) si avrà rottura per splitting, se invece ce né almeno uno di segno positivo (trazione) si avrà formazione di una fessura.

Per quanto riguarda invece il programma, abbiamo già visto che la formazione delle fessure non viene individuata con la superficie di rottura, ma per confronto diretto tra valori di tensione. La superficie in questione viene quindi usata solo per determinare il raggiungimento dello splitting, e non c'è quindi bisogno di effettuare nessun controllo sul segno dei valori delle tensioni principali.

In ogni caso, anche gli elementi fessurati possono successivamente andare in contro alla rottura per splitting.

Riporto adesso la superficie di rottura del calcestruzzo utilizzata da Kotsovos, descrivendola in maniera quantitativa:

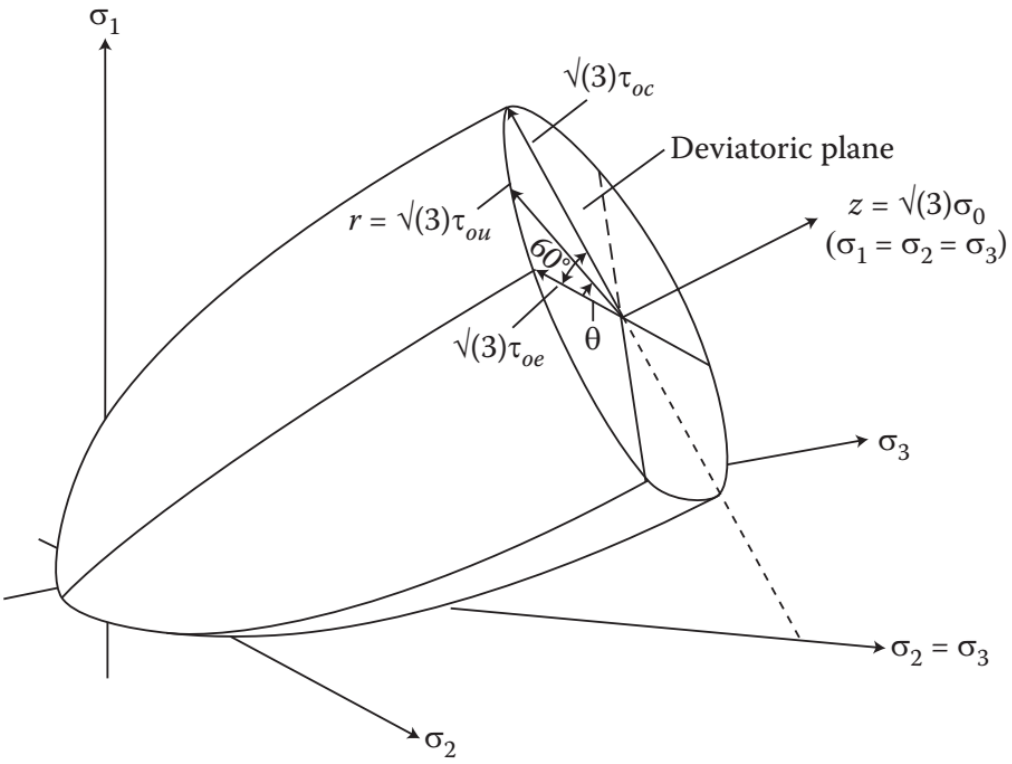

Figura 18 Superficie di rottura del calcestruzzo utilizzata da Kotsovos (2015)

Sono visibili, tra l'altro, anche τ_{0e}, τ_{0c} e τ_{0u}. Dato un certo σ_0, esprimono le massime tensioni deviatoriche ammissibili per diversi valori di Θ: τ_{0e} e τ_{0c} per Θ pari a 0° e 60°, mentre τ_{0u} per Θ intermedio tra i due valori.

Avevamo visto che la superficie di rottura è tre volte simmetrica. In virtù di tale simmetria, per descrivere la superficie sono sufficienti due relazioni che legano τ_{0e} e τ_{0c} a σ_0, più una terza che permette di calcolare τ_{0u} una volta noti τ_{0e} e τ_{0c}, e dato un certo Θ. Kotsovos fornisce le seguenti formule, calcolate sperimentalmente sulla base di test triassiali effettuati su calcestruzzi di diverse resistenze:

$$\tau_{0c} = 0.944 f_c \left(\frac{\sigma_0}{f_c} + 0.075\right)^{0.724} + 0.025 f_c$$

$$\tau_{0e} = 0.633 f_c \left(\frac{\sigma_0}{f_c} + 0.075\right)^{0.857} + 0.025 f_c$$

In particolare, i valori di 0.075 e 0.025 sono delle percentuali rispetto a f_c. Moltiplicati per tale valore esprimono rispettivamente la tensione ammissibile di trazione idrostatica e lo sforzo di taglio

ottaedrale aggiuntivo massimo. Siccome è suggerito di usare percentuali comprese tra 0.05 e 0.1 per il primo valore, e tra 0 e 0.05 per il secondo, ho scelto i valori intermedi.

L'interpolazione per trovare τ_{0u} è suggerita da William and Wanrke (1974):

$$\tau_{0u} = \{2\tau_{0c}(\tau_{0c}^2 - \tau_{0e}^2)\cos\theta + \tau_{0c}(2\tau_{0e} - \tau_{0c})$$

$$[4(\tau_{0c}^2 - \tau_{0e}^2)cos^2\theta + 5\tau_{0e}^2 - 4\tau_{0c}\tau_{0e}]^{\frac{1}{2}}\}/[4(\tau_{0c}^2 - \tau_{0e}^2)cos^2\theta + (\tau_{0c} - 2\tau_{0e})^2]$$

Lo stato tensionale in un elemento di calcestruzzo raggiunge la superficie di rottura se:

$$\tau_0 \geq \tau_{0u}$$

2.4 Barre di acciaio

Esistono diverse leggi costitutive con cui può essere modellato l'acciaio, ma esse non sono molto diverse tra di loro (vedi ad esempio Ottosen e Ristinmaa (2005) o CEN, *Eurocode 2 – Part 1-1* (2004)). Che io sappia, l'acciaio viene sempre modellato con un comportamento elastoplastico. Sostanzialmente quello che può cambiare è la presenza o meno dell'incrudimento e di un eventuale tratto di softening, e l'uso di una legge multilineare (più diffuso) anziché di una con tratti curvi.

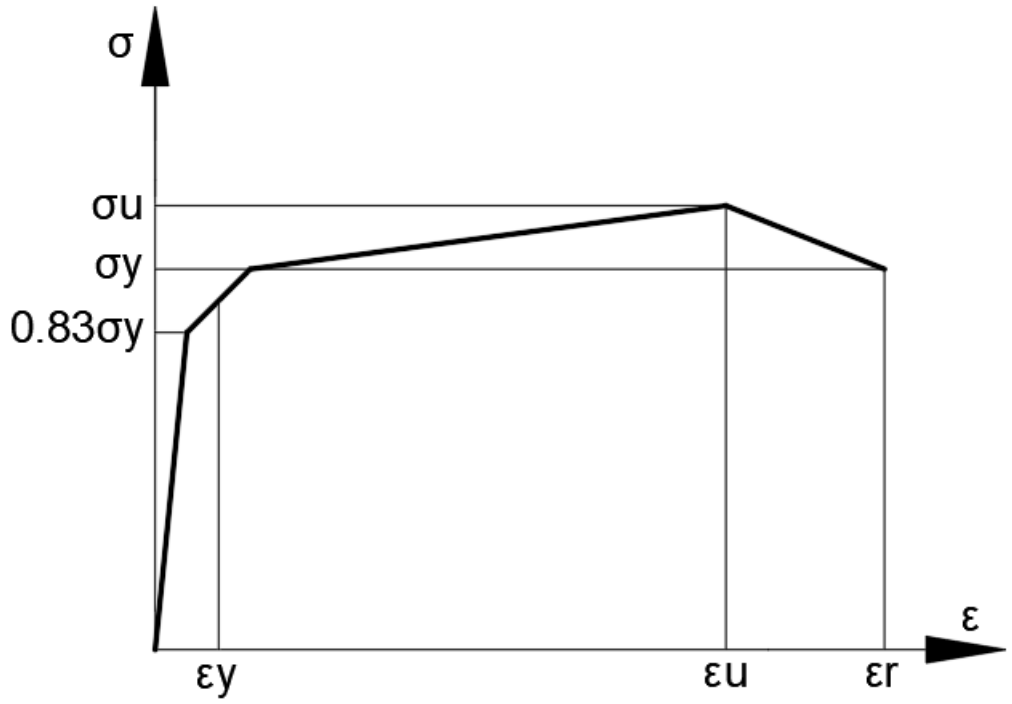

Figura 19 Esempio di legge costitutiva per la modellazione dell'acciaio

L'esempio di legge costitutiva appena riportata è multilineare, e contiene sia un tratto incrudente che uno finale di softening. Dopo la prima fase, elastica, il primo breve tratto plastico modella sostanzialmente lo snervamento, e si raggiunge a una percentuale della resistenza a snervamento nominale. Solitamente il comportamento a compressione viene modellato con la stessa legge di quello a trazione.

Per le barre d'acciaio, nel programma di calcolo ho scelto una delle più semplici, ma allo stesso tempo realistiche, leggi costitutive: quella bilineare con tratto incrudente:

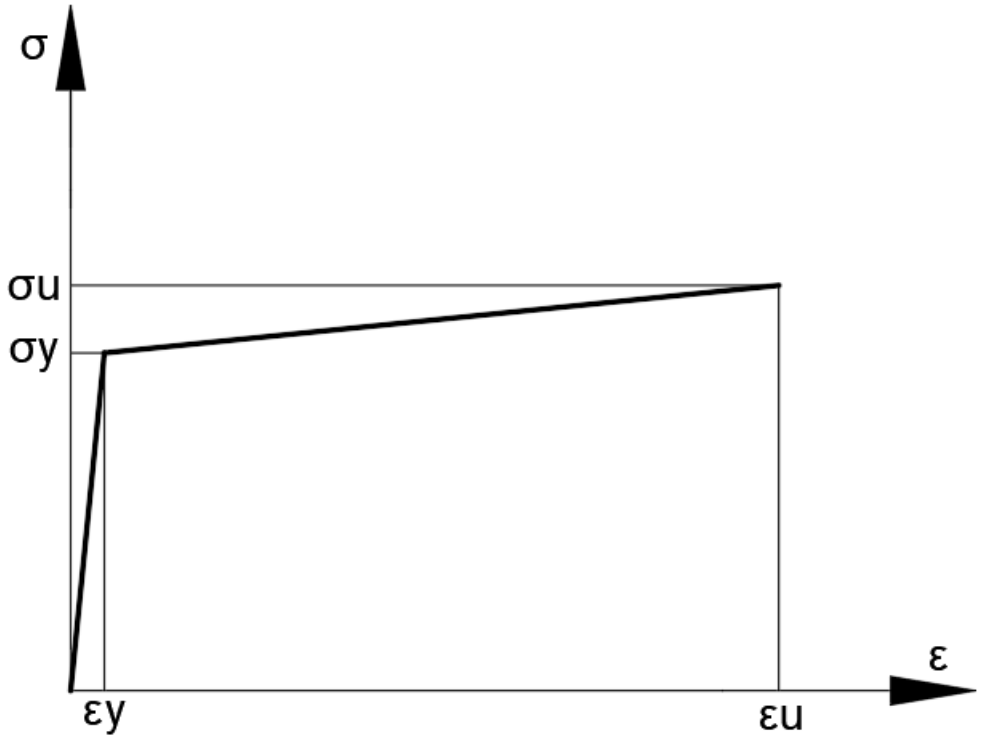

Figura 20 Disegno qualitativo della legge costitutiva dell'acciaio delle barre usato nel programma

Ci saranno quindi una tensione di snervamento e una tensione ultima, associate ai rispettivi valori di deformazione. La pendenza del primo tratto sarà la rigidezza in fase elastica E_0 e la pendenza del secondo tratto la rigidezza in fase plastica E_p. La stessa legge costitutiva è usata per modellare il comportamento sia a trazione che a compressione.

All'inizio di ogni analisi tutte le barre hanno rigidezza pari a E_0. Alla fine dell'n-esimo step di carico un apposito comando confronta la tensione (in modulo) raggiunta da ogni barra con la tensione di snervamento della stessa:

$$\sigma_{mn} > \sigma_{my} \Rightarrow E_{mn} = E_{mp}$$

La rigidezza della barra (precedentemente pari a E_0) viene quindi aggiornata al valore di E_p. Ricordo che il pedice *m* sta solo ad indicare che si tratta degli elementi monodimensionali.

Se invece:

$$\sigma_{mn} > \sigma_{mu}$$

la barra ha raggiunto la rottura e l'analisi giunge al termine.

Ricordo che il legame tra acciaio e calcestruzzo è sempre di ancoraggio perfetto (*perfect bond*): nessuna ulteriore modellazione è quindi necessaria a questo scopo.

2.5 Nota sul comportamento a traliccio di resistenza al taglio

A stato limite ultimo l'eurocodice 2 (CEN, *Eurocode 2 – Part 1-1* (2004)) prevede che la resistenza a taglio degli elementi che hanno armatura a taglio sia attribuita alla formazione di un traliccio isostatico ideale avente un corrente compresso, uno teso, delle aste di parete compresse e delle aste di parete tese. I correnti sono in direzione longitudinale: quello compresso è costituito dal calcestruzzo in zona compressa e quello teso dalle barre in zona tesa. Le aste di parete sono genericamente in direzione inclinata rispetto allo sviluppo longitudinale: quelle compresse sono costituite dal calcestruzzo, quelle tese sono le armature a taglio (nel disegno poste in verticale):

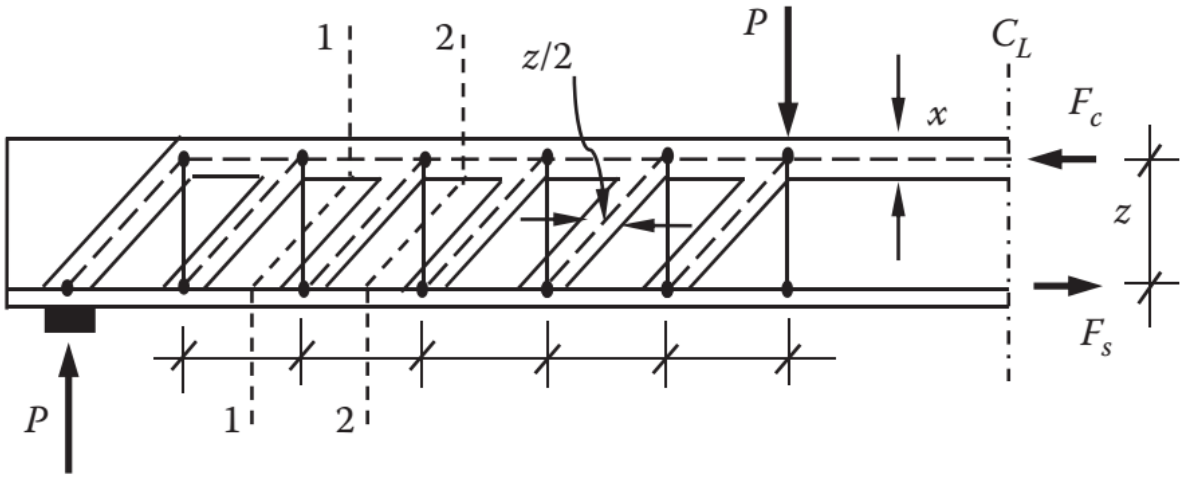

Figura 21 Tipico comportamento a traliccio di resistenza ad una sollecitazione di taglio P. Fc ed Fs sono invece le forze in zona compressa e tesa dovute a sforzo normale e momento flettente

Un concetto molto importante che emerge dal libro di Kotsovos (2015) è l'opinione dell'autore secondo cui il traliccio in questione in realtà non si forma. Il motivo a cui è attribuita questa tesi è la fragilità del calcestruzzo, che ho illustrato al paragrafo sulla rottura per splitting. Kotsovos afferma che l'assenza di tratti di post picco nella rottura a compressione del calcestruzzo impedisce la formazione del meccanismo a traliccio appena descritto.

Capitolo

3 Esempi di calcolo

In questo capitolo testo il funzionamento del programma su strutture reali. L'intenzione iniziale era di ripercorrere tutti gli esempi di calcolo del libro di Kotsovos (2015) che potessero essere analizzati in 2D, tuttavia i limiti del programma non permettono di analizzare alcune di queste strutture. Ci sono infatti delle travi soggette a carichi non proporzionali e una struttura, essendo un nodo trave-colonna, non ha geometria rettangolare. Per cui ho integrato i casi del libro che ho potuto prendere in considerazione con altri presi da altre fonti. Ho quindi aggiunto la trave A3 del primo paragrafo, un muro a taglio con carico assiale (SW7) e le due colonne finali.

Tutte le analisi sono state eseguite in controllo di carico mentre quasi tutti i test sperimentali sono in controllo di spostamento. Questo aggiunge in effetti una fonte di incertezza ma permette in ogni caso di effettuare una prima valutazione sulla bontà delle analisi. Le strutture sono state caricate con step tutti uguali e pari a un centesimo del carico sperimentale di rottura (questo vuol dire che gli step effettivi sono stati più di 100 per le analisi che hanno avuto un carico di rottura maggiore dello sperimentale, e meno di 100 per quelle che ne hanno avuto uno minore). Per strutture di cui non si conosce il carico sperimentale si potrebbero prendere dei valori di step di tentativo per poi aggiustarli in modo da avere un numero di step di carico non troppo basso né troppo alto.

Per ogni esempio ho descritto i test sperimentali e fornito alcune informazioni su come le strutture sono state discretizzate. Dopodiché ho riportato alcuni risultati utili al confronto (alcuni, e non tutti per ogni struttura, per evitare di appesantire eccessivamente l'elaborato). Il libro di Griffiths et al. (2014) prevede che le analisi possano essere rappresentate con il software open source ParaView: alcuni dei risultati sono stati infatti ottenuti con tale software.

3.1 Bresler e Scordelis (1963), travi A1 e A3

Preciso innanzi tutto che questo primo paragrafo è particolarmente importante, perché fornirò una serie di informazioni che saranno valide anche per gli altri esempi di calcolo.

Le prime strutture prese in considerazione sono due travi semplicemente appoggiate, con carico puntuale in mezzeria, testate sperimentalmente in Bresler e Scordelis (1963). Solo la trave A1 è stata presa in considerazione anche nel libro di Kotsovos (2015).

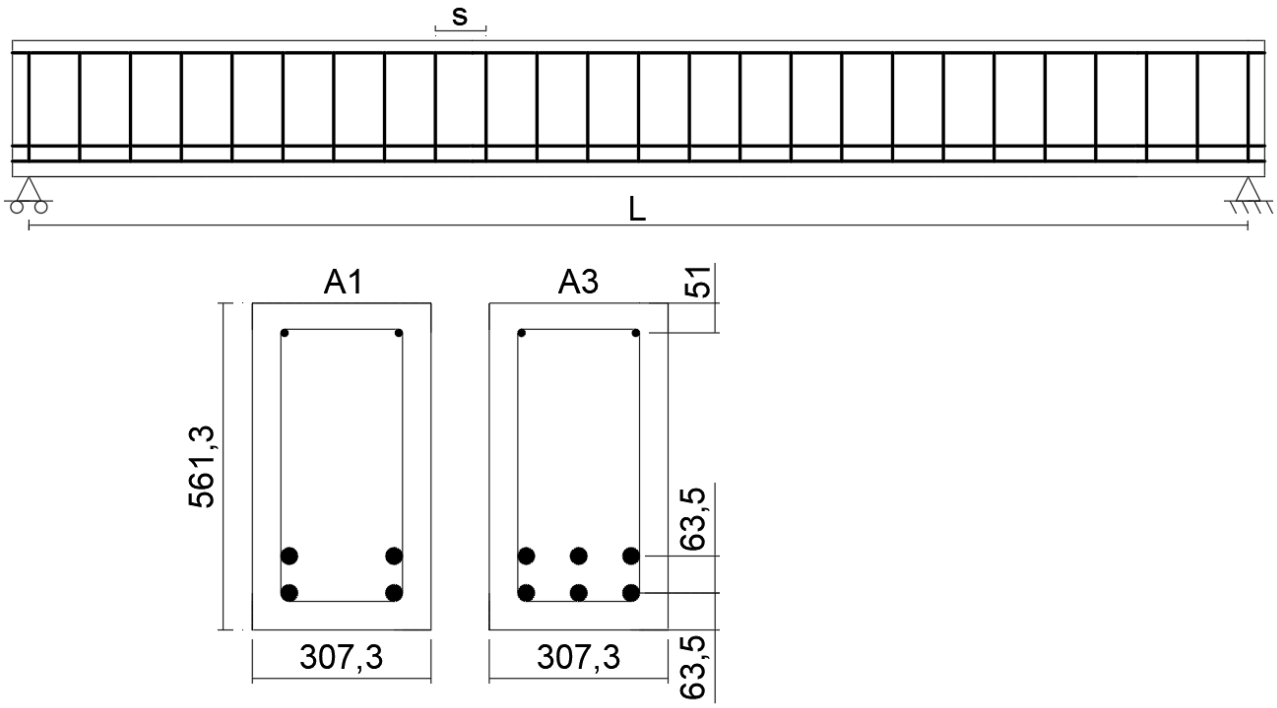

Figura 22 Caratteristiche geometriche delle travi (tutte le misure sono in mm)

Le due travi differiscono solo per la luce L tra i due appoggi (3658 mm per A1 e 6401 mm per A3), la quantità di armatura in zona tesa, e la resistenza a compressione cilindrica del calcestruzzo (24.1 MPa e 35 MPa). Avendo la sezione delle stesse dimensioni, la trave A3 è quindi più snella.

Le armature longitudinali in zona tesa sono barre ϕ28.9, in zona compressa ϕ12.7. L'armatura a taglio è composta da staffe ϕ6.4 con passo s 210 mm. Le barre hanno i valori di resistenza riportati in tabella:

diametro (mm)	Resistenza a snervamento (MPa)	Resistenza ultima (MPa)	Modulo di rigidezza in fase plastica (MPa)
28.9	555	958	3400
12.7	345	542	990
6.4	325	430	630

Tabella 1 Caratteristiche fisiche delle barre di acciaio

Per quanto riguarda la rigidezza in fase elastica ho sempre usato il valore di 200 GPa, anche negli altri esempi di calcolo. In effetti in alcuni casi le barre avevano rigidezze leggermente diverse ma il valore in questione mi è sembrato una buona approssimazione.

Non è specificato se le armature longitudinali raggiungono lo snervamento, mentre la rottura è avvenuta in entrambi i casi per splitting del calcestruzzo. Questa modalità di rottura sarà comune a tutti gli altri test sperimentali.

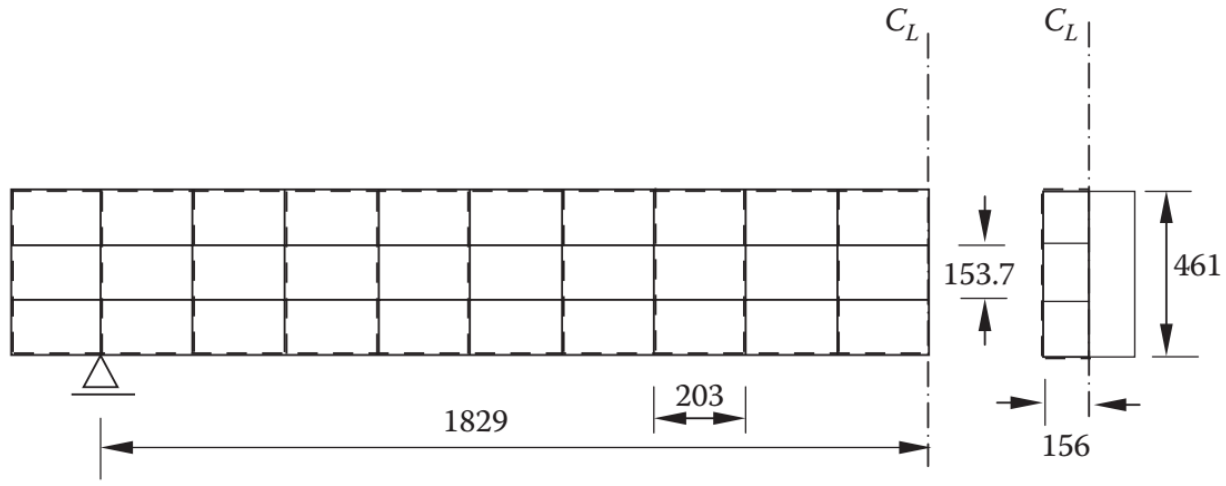

Figura 23 Mesh adottata nel libro (misure in mm). Le barre si trovano lungo le linee tratteggiate. Dalla vista in sezione si nota che per simmetria è stato analizzato soltanto un quarto di trave

Le analisi di Kotsovos sono tridimensionali, mentre le mie sono in stato di tensione piana. Per quanto riguarda le mesh, siccome Kotsovos utilizza solo elementi del secondo ordine mentre io solo elementi del primo, ho adottato delle mesh più fitte. Queste due considerazioni valgono anche per tutti gli altri esempi.

C'è poi in questo caso un'altra differenza: l'autore ha trascurato il copriferro dell'armatura in tensione, mentre io ho voluto rappresentare le travi per intero.

Siccome la trave A3 non è stata analizzata nel libro, quanto detto sulla mesh adottata da Kotsovos si riferisce ovviamente alla sola trave A1.

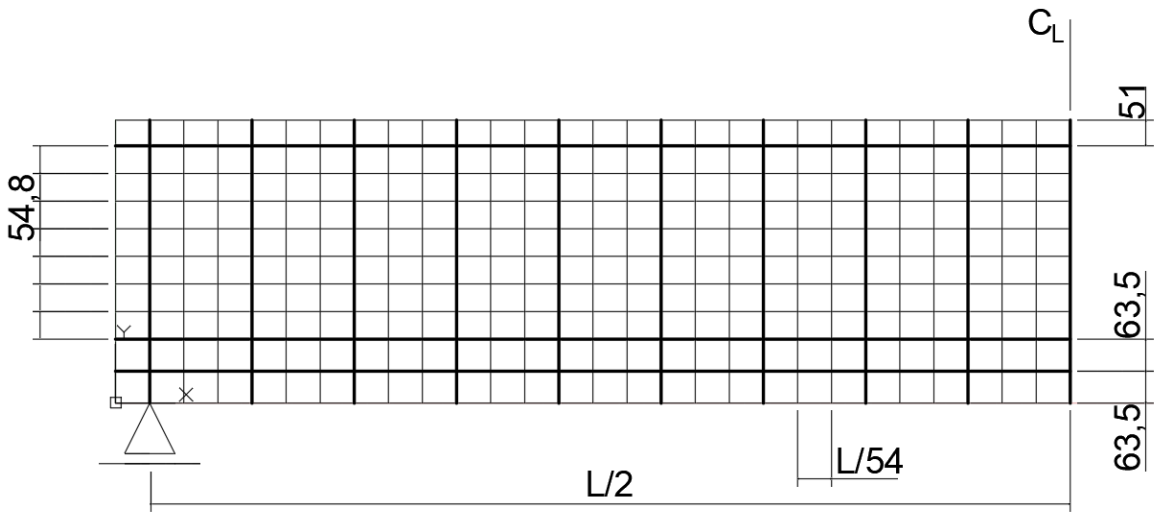

Figura 24 Mesh adottata nel programma (misure in mm); barre rappresentate dai tratti di linea spessi

Ho usato per comodità la stessa mesh per entrambe le travi. A questo proposito, ho quindi dovuto cambiare soltanto i valori geometrici relativi alla lunghezza della trave e adattare la quantità di armatura a taglio in base al rapporto tra il passo delle staffe reale e di modello. La misura di L/54 presente in figura è dovuta ovviamente alla suddivisione di metà trave in 27 parti lungo x. Inoltre, essendo le mie analisi in stato di tensione piana, ho omesso la rappresentazione dello spessore della struttura (anche questo sarà valido anche per gli altri esempi).

I valori sperimentali di carico ultimo e spostamento ultimo in mezzeria (quest'ultimo ricavato dai grafici) sono stati di 467 kN e 13.8 mm per la trave A1 e di 468 kN e 33.7 mm per la trave A3:

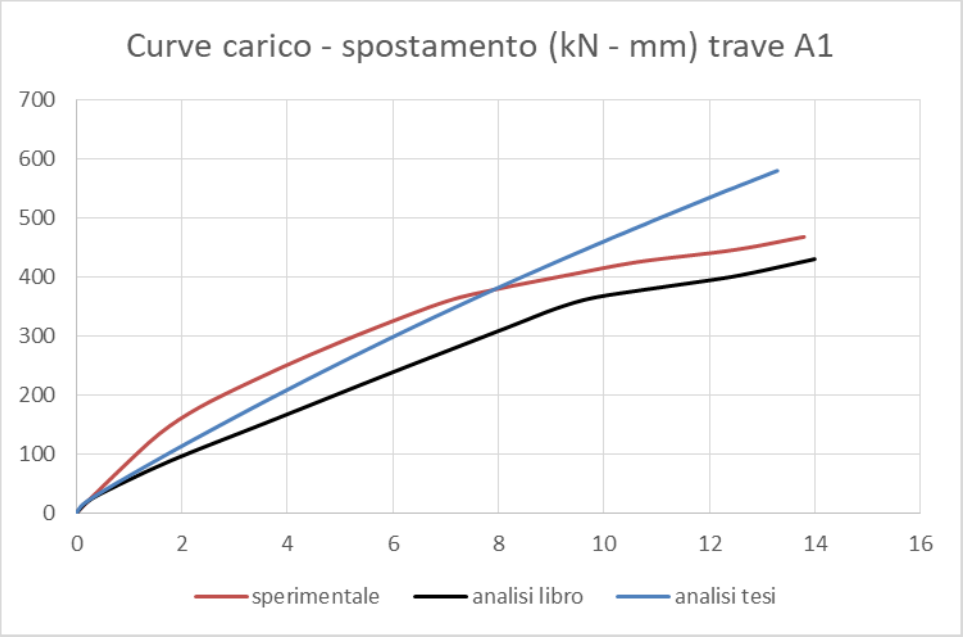

Figura 25 Confronto tra curve di carico – spostamento per la trave A1

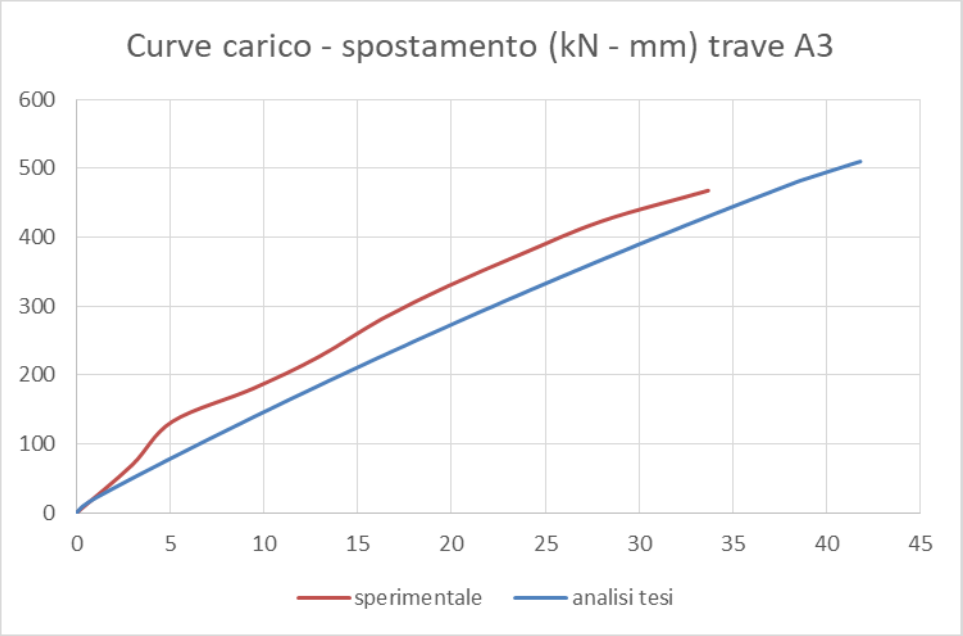

Figura 26 Confronto tra curve di carico - spostamento per la trave A3

Ritengo che il confronto sia buono. Ad esempio, la mia analisi ha sovrastimato il carico di circa il 20% per la trave A1 e di circa il 10% per la trave A3.

Riporto a titolo esemplificativo le tensioni σ_x e lo stato fessurativo (la deformata della travi è scalata di un fattore 10):

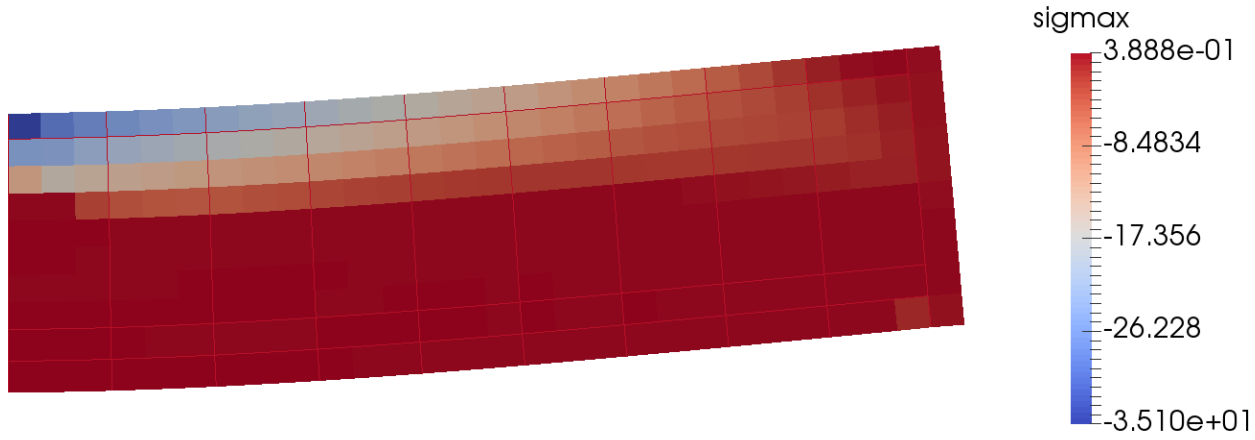

Figura 27 Tensioni a rottura lungo l'asse x per la trave A1

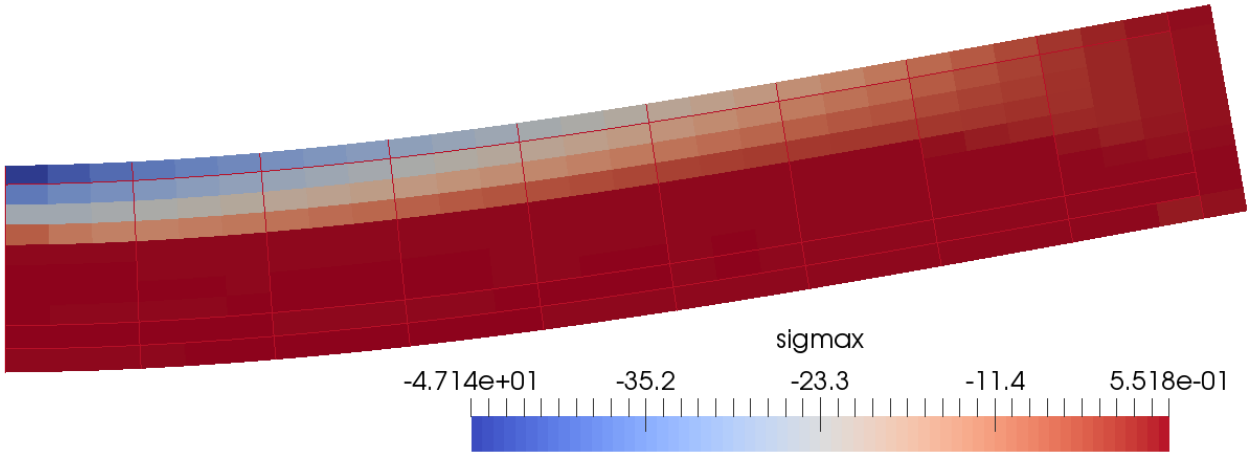

Figura 28 Tensioni a rottura lungo l'asse x per la trave A3

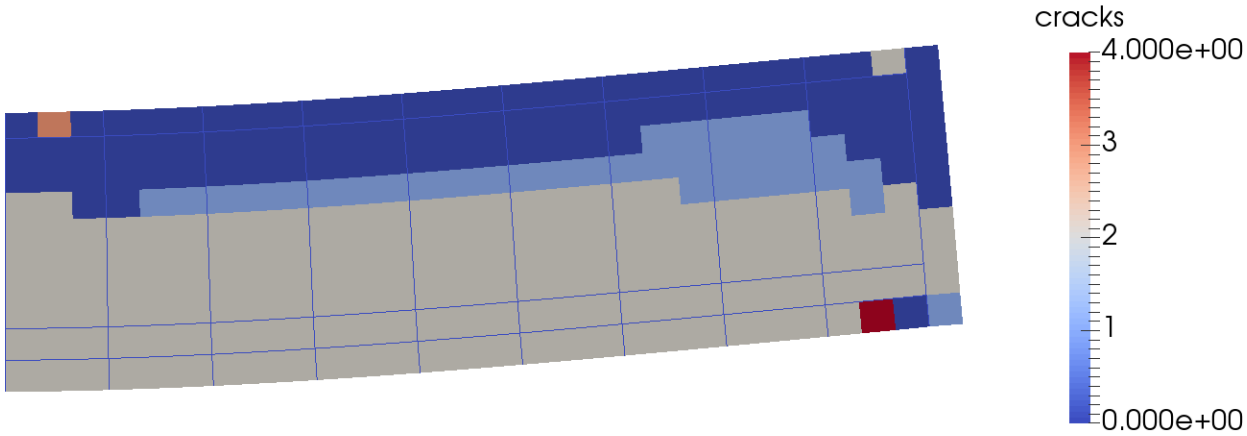

Figura 29 Stato fessurativo a rottura per la trave A1

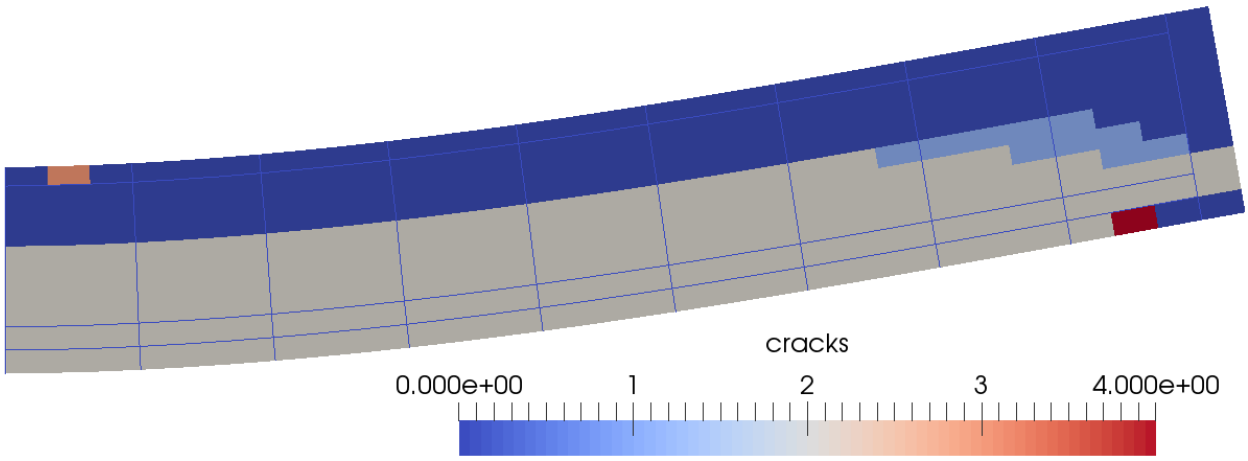

Figura 30 Stato fessurativo a rottura per la trave A3

Le immagini appena riportate sono state ottenute con ParaView. Quella che segue è invece l'immagine dello stato fessurativo nell'analisi sperimentale.

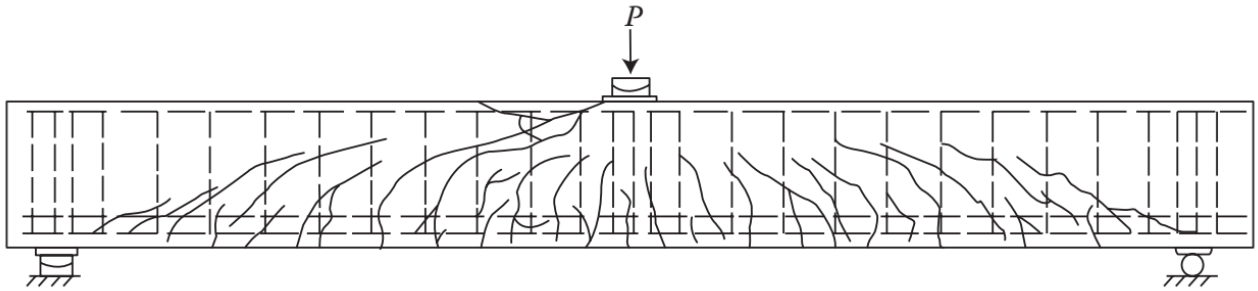

Figura 31 Stato fessurativo a rottura per la trave A1, da test sperimentale; le fessure sono state disegnate sulla trave indeformata

I grafici sulle tensioni in x hanno ovviamente il tipico andamento legato alla sollecitazione di momento flettente. In prossimità della mezzeria il calcestruzzo riesce a raggiungere tensioni superiori (in valore assoluto) rispetto alla resistenza a compressione, grazie all'effetto di confinamento fornito dal carico stesso e dalle staffe a taglio.

Ricordo il significato dei numeri associati agli stati di fessurazione:

0 – elemento integro

1 – elemento che ha perso solo rigidezza a taglio

2 – elemento con fessura lungo l'asse y

3 – elemento con fessura lungo l'asse x

4 – elemento fessurato in entrambe le direzioni

Come si può vedere, e come era ovvio, la maggior parte delle fessure è ordita verticalmente. In prossimità del carico c'è un elemento fessurato in x e in prossimità dell'appoggio un elemento fessurato *anche* in x; questo è probabilmente dovuto proprio all'effetto locale di applicazione delle forze (quella esterna e quella di reazione vincolare). Alcuni ulteriori elementi hanno invece perso soltanto la rigidezza a taglio. Ho riportato anche lo stato fessurativo sperimentale della trave A1, come riportato sul libro di Kotsovos. Evidentemente il confronto non è immediato per via della rappresentazione numerica della fessurazione nel programma.

Segnalo anche che in entrambe le analisi hanno raggiunto lo snervamento le barre in compressione in un piccolo intorno della mezzeria della trave. Nell'analisi della trave A3 ha raggiunto lo snervamento, approssimativamente nella stessa zona, anche una delle due file di barre in trazione (ovviamente quelle più vicine all'intradosso).

Entrambe le analisi terminano per rottura di un elemento in calcestruzzo (questo accadrà anche per tutti gli altri esempi di calcolo). Nel caso delle travi di questo paragrafo a venir meno non è l'elemento maggiormente sollecitato lungo x (quello in alto a sinistra) ma quello immediatamente alla sua destra. Quello con la più alta tensione in x risente infatti di un importante sollecitazione di compressione anche in y, dovuta all'applicazione del carico, e resiste quindi maggiormente (nell'esempio sui muri SW30 ed SW32 ho commentato più dettagliatamente un effetto simile, il confinamento fornito dai nodi vincolati).

Segnalo infine la presenza di un risultato che per una delle due travi è discordante dall'analisi sperimentale, ossia la sollecitazione nelle barre a taglio. Nell'articolo di Bresler e Scordelis quest'informazione viene riportata in maniera indiretta, attraverso gli spostamenti verticali relativi tra base e sommità della trave, in corrispondenza di diverse sezioni. Nel caso della trave A1 le barre raggiungono lo snervamento, nel caso della trave A3 mantengono valori molto bassi di tensione. Per quanto riguarda le analisi i valori di tensione non sono molto alti (abbastanza lontani dalla resistenza a snervamento dell'acciaio) e quindi il risultato è in accordo con lo sperimentale per la trave A3 e in disaccordo con la trave A1. In ogni caso, nelle analisi numeriche entra in tensione soltanto una parte di queste barre, quella in prossimità dell'estradosso delle travi, ossia in corrispondenza del calcestruzzo in compressione.

3.2 Maier (1985), muro a taglio S4

Il riferimento bibliografico sperimentale del muro è quello di Maier (1985) ed è stato analizzato anche nel libro di Kotsovos (2015). La geometria ha quasi la forma di un quadrato. È caricato con una forza di compressione costante e con una forza di taglio fino a rottura. È collegato sia alla base che in sommità con delle travi (non rappresentate nel disegno): la prima fornisce l'ancoraggio, la seconda i carichi. L'armatura è disposta sia orizzontalmente che verticalmente, con percentuali molto simili e con lo stesso tipo di barre.

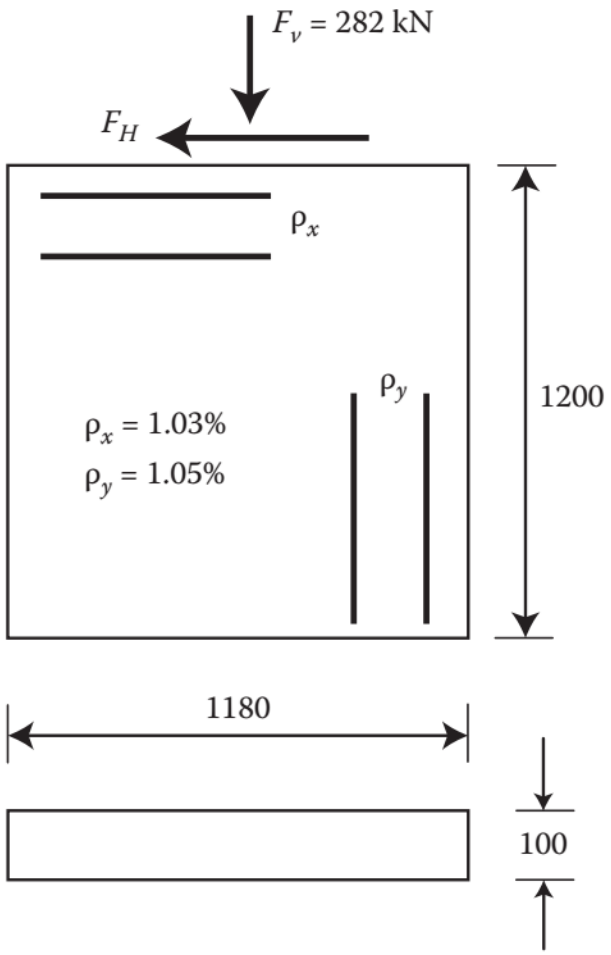

Figura 32 Caratteristiche geometriche del muro (tutte le misure sono in mm)

La resistenza a compressione cilindrica del calcestruzzo è di 30 MPa, mentre le barre hanno una resistenza a snervamento di 574 MPa, una resistenza ultima di 764 MPa e un modulo di rigidezza in fase plastica di 1600 MPa (ricordo che per semplicità la rigidezza in fase elastica delle barre è sempre stata posta pari a 200 GPa). La rottura avviene per splitting del calcestruzzo. Dai grafici sulle deformazioni si deduce che raggiungono lo snervamento diverse file di barre verticali in zona tesa; non

ci sono invece a mio parere sufficienti informazioni per capire se si sono plasticizzate anche alcune barre in zona compressa.

Nell'analisi del libro la mesh è stata ottenuta suddividendo il muro in quattro parti per lato. Avrei potuto adottare come criterio (poco rigoroso) una suddivisione doppia rispetto a quella del libro per tener conto degli elementi del prim'ordine al posto di quelli del secondo: mi è sembrato più opportuno suddividere in dieci parti per lato (immagino sarebbe cambiato molto poco). Anche in questo caso è poi presente una piccola differenza geometrica tra le due analisi: ho modellato, con materiale infinitamente rigido, una metà della trave superiore, per prendere in considerazione la piccola eccentricità (120 mm) con cui è applicata la forza F_H; tale eccentricità è stata invece trascurata nel libro. Entrambe le analisi considerano invece infinitamente rigida la trave inferiore e pertanto la escludono, sostituendola con un incastro perfetto.

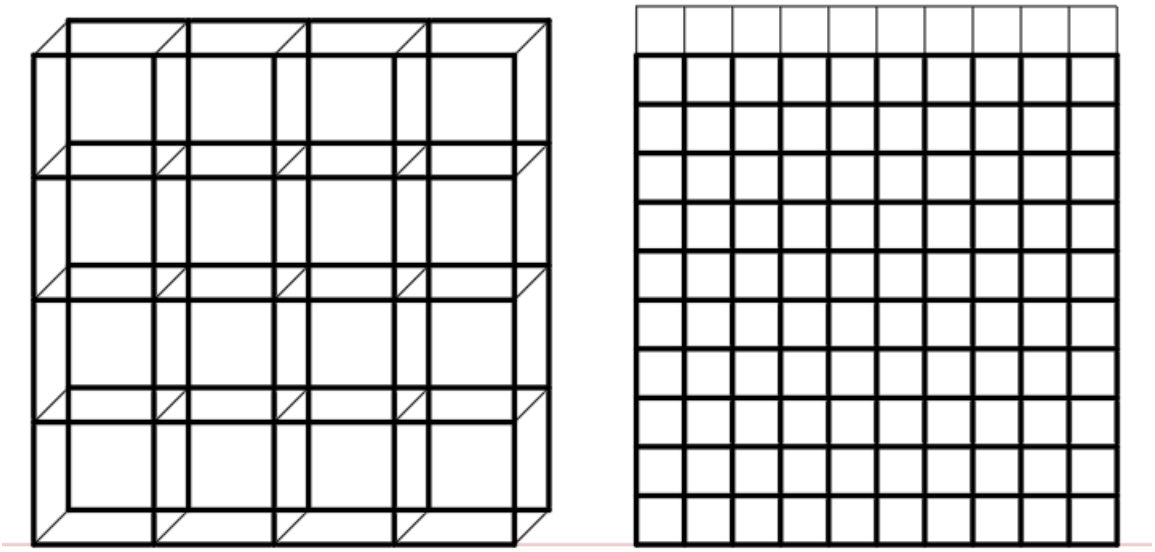

Figura 33 Mesh adottata nel libro (3D) e mesh adottata nel mio programma (2D), le barre si trovano lungo i tratti di linea spessi. Nella mia mesh è visibile l'ulteriore fila di elementi (modellati come infinitamente rigidi) alla sommità del muro

I valori sperimentali di carico ultimo e spostamento orizzontale ultimo alla sommità del muro (quest'ultimo ricavato dal grafico) sono stati di 392 kN e 14.4 mm.

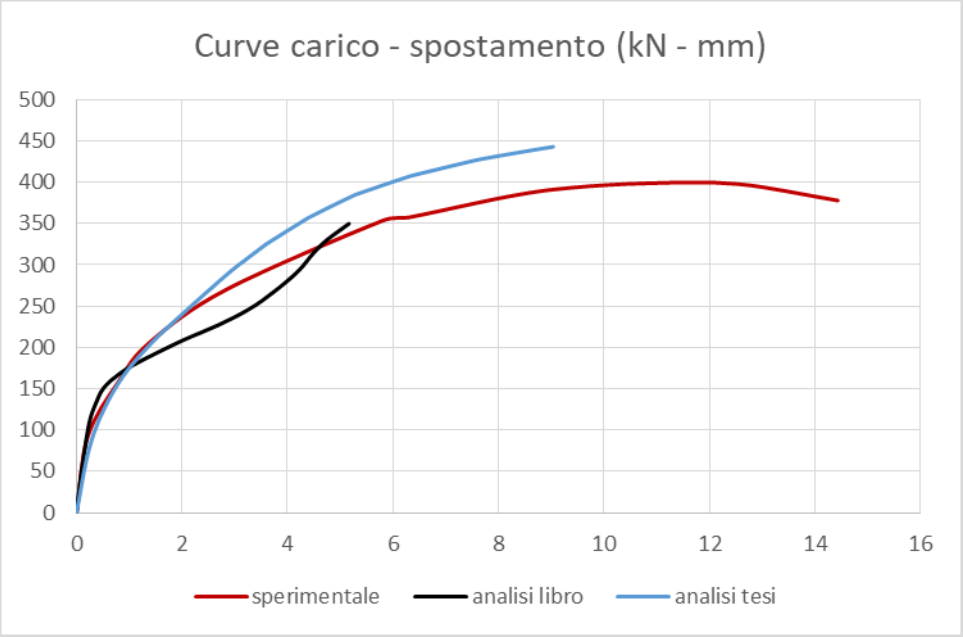

Figura 34 Confronto tra curve di carico – spostamento

In questo caso il carico è stato sovrastimato di circa il 13%, mentre lo spostamento è stato sottostimato di circa il 37%. Si nota, stranamente, anche una marcata sottostima dello spostamento nell'analisi del libro. Non mi è possibile ipotizzare una spiegazione plausibile per tale differenza, al di là dei diversi comportamenti dovuti alle incertezze di modello.

Riporto in questo caso le tensioni σ_y e le barre che si sono plasticizzate (di nuovo, la deformata è scalata di un fattore 10):

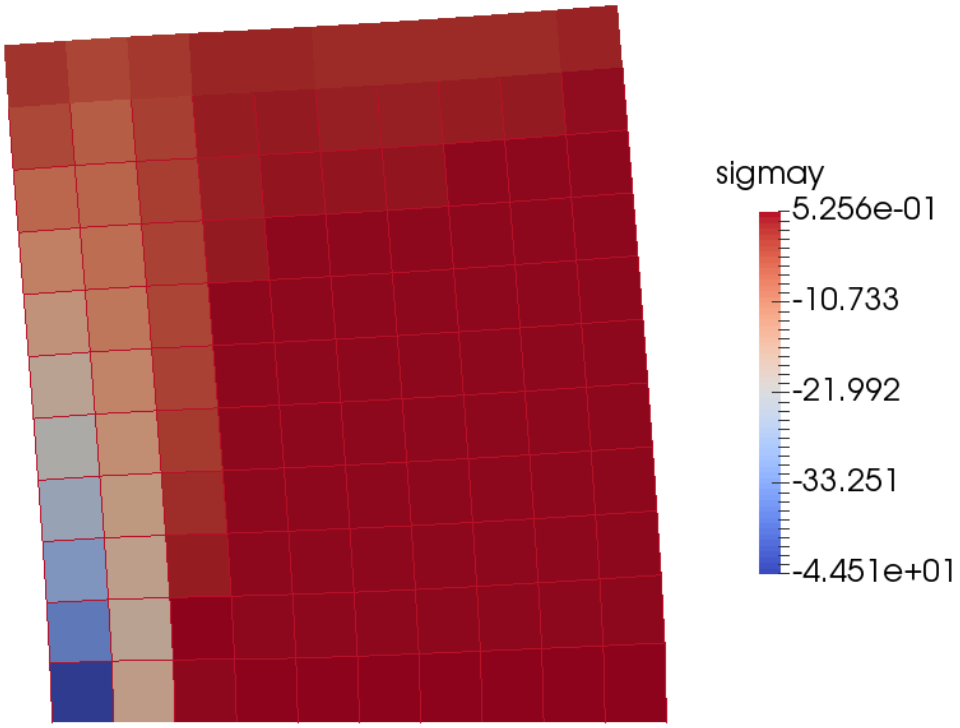

Figura 35 Tensioni a rottura lungo l'asse y

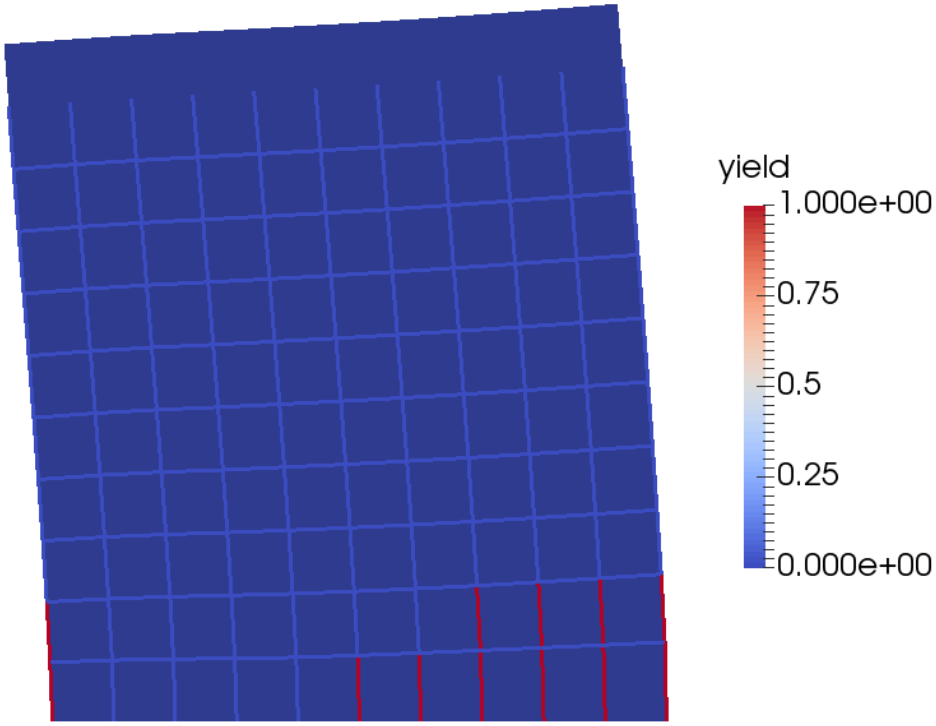

Figura 36 Barre che hanno raggiunto lo snervamento (in rosso)

Anche in questo caso è stato possibile effettuare un controllo sulla tensione nelle barre ordite in direzione parallela al carico principale. Dai risultati dello sperimentale sembra che le barre entrino in tensione con valori più alti e in maniera più omogenea rispetto all'analisi numerica.

L'analisi è terminata per rottura del calcestruzzo alla base del muro, in zona compressa.

3.3 Kotsovos (1990), muri a taglio SW30 ed SW32

Sperimentalmente, i due muri sono stati testati in Kotsovos (1990): l'SW30 sotto carico monotono, l'SW32 sotto carico ciclico. Numericamente è stato analizzato, in Kotsovos (2015), il solo muro SW32, sia sotto carico monotono che ciclico. Confronterò quindi il primo muro con il test sperimentale e il secondo con l'analisi numerica effettuata da Kotsovos (2015) sotto carico monotono.

I due muri hanno la stessa geometria ed armatura: differiscono quindi solo per la resistenza del calcestruzzo. Hanno un rapporto tra altezza e larghezza pari a 2, e uno spessore di 65 mm. Sono collegati sia alla base che in sommità con delle travi, non rappresentate: forniscono rispettivamente ancoraggio e carico. Quest'ultimo consiste in una sola forza orizzontale crescente fino a rottura.

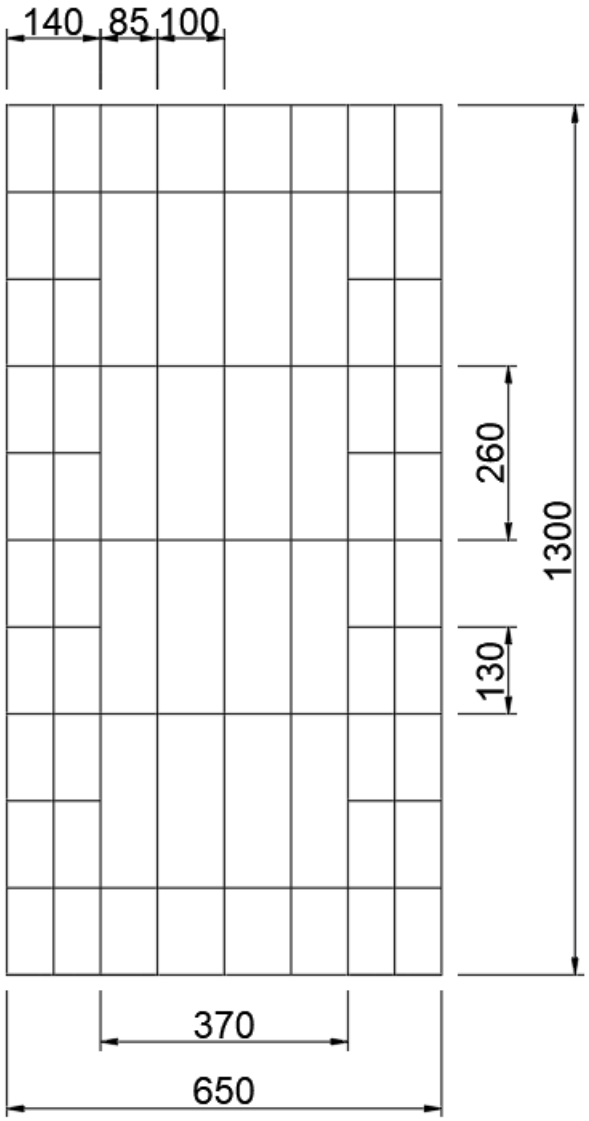

Figura 37 Geometria dei muri (misure in mm)

Sono armati con 9 coppie di barre ϕ8 in verticale e 5 di barre ϕ6.25 in orizzontale. Le estremità dei muri hanno anche armatura di confinamento con staffe chiuse ϕ4 ogni 130 mm. La resistenza a compressione cilindrica del calcestruzzo è di 25 MPa per il SW30 e di 29.2 MPa per il SW32, mentre le barre hanno i seguenti valori di resistenza:

diametro (mm)	Resistenza a snervamento (MPa)	Resistenza ultima (MPa)
8	470	565
6.25	520	610
4	420	490

Tabella 2 Resistenza a snervamento e ultima delle barre

Non è invece specificata la rigidezza in fase plastica delle barre: ho assunto il valore di 2000 MPa.

La rottura dell'SW30 avviene per splitting del calcestruzzo, nell'angolo in zona compressa del muro. Essendo specificato che si è trattato di una rottura duttile si può dedurre che raggiungono lo snervamento almeno le armature verticali in zona tesa. Nel libro di Kotsovos (2015) non sono presenti informazioni analoghe sull'analisi numerica monotona del muro SW32, ma è assolutamente probabile che sia avvenuto lo stesso tipo di rottura.

Riporto le mesh adottate nel libro e nell'analisi:

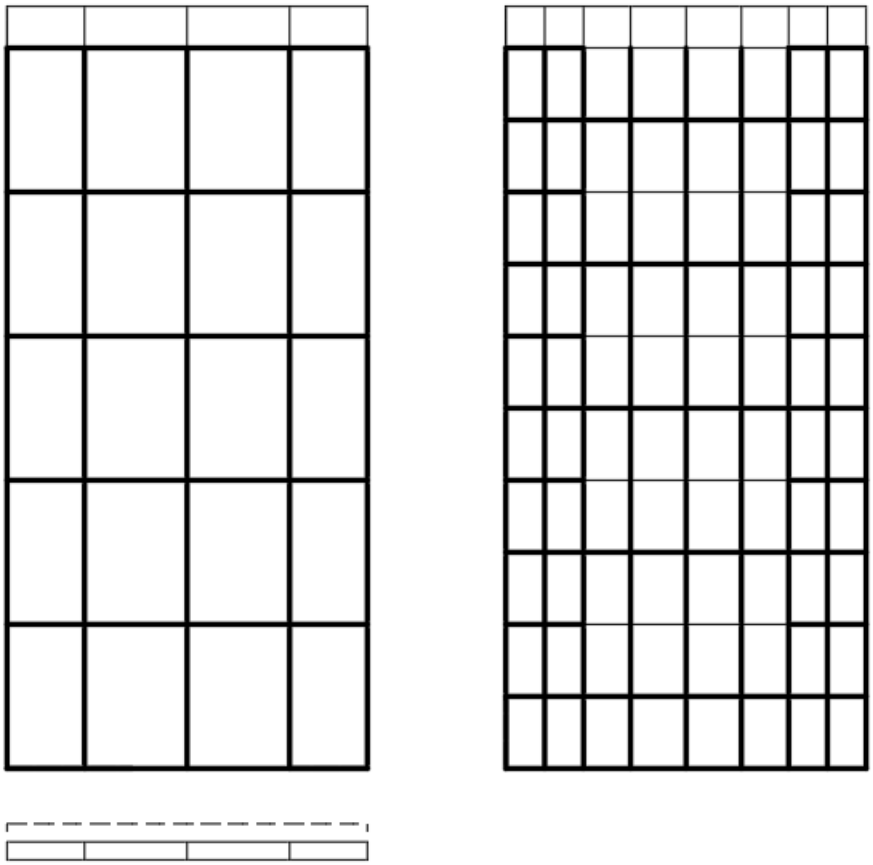

Figura 38 Mesh del libro (a sinistra) e mesh delle analisi (a destra), le barre si trovano lungo le linee spesse. Nel libro è stato modellato, per simmetria, soltanto metà muro (32.5 mm di spessore anziché 65). Nelle mie analisi (2D) è stato invece inserito, ovviamente, lo spessore di 65 mm

Anche in questo caso vale la considerazione per cui quanto detto sulla mesh adottata da Kotsovos (2015) si riferisce al solo muro da lui analizzato, l'SW32. Nel mio caso ho ovviamente usato la stessa mesh per entrambi i muri, essendo diversi solo per il valore di resistenza del calcestruzzo.

Ho usato una suddivisione di mesh doppia rispetto a quella del libro. Entrambe le mesh prendono in considerazione l'eccentricità (75 mm) con cui è applicata la forza, modellando quindi soltanto metà della trave superiore. Nelle mie analisi ho modellato questa metà di trave con materiale lineare elastico avente 30000 MPa di rigidezza e 0.2 di coefficiente di Poisson, valori iniziali tipici del calcestruzzo. Tutte le analisi considerano infinitamente rigida la trave inferiore (pertanto la escludono, sostituendola con un incastro perfetto) e prendono in considerazione soltanto metà trave superiore.

I valori di carico ultimo e spostamento orizzontale ultimo alla sommità dei muri sono stati di 117 kN e 20.9 mm per l'SW30 (valori sperimentali) e di 115 kN e 9 mm per l'SW32 (valori numerici dell'analisi di Kotsovos (2015)).

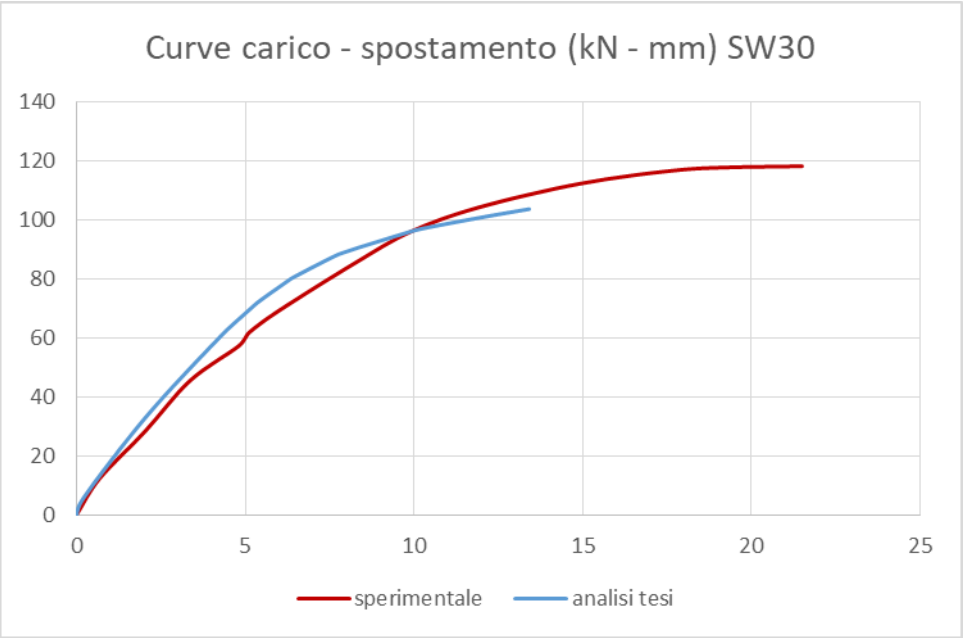

Figura 39 Confronto tra curve di carico - spostamento per il muro SW30

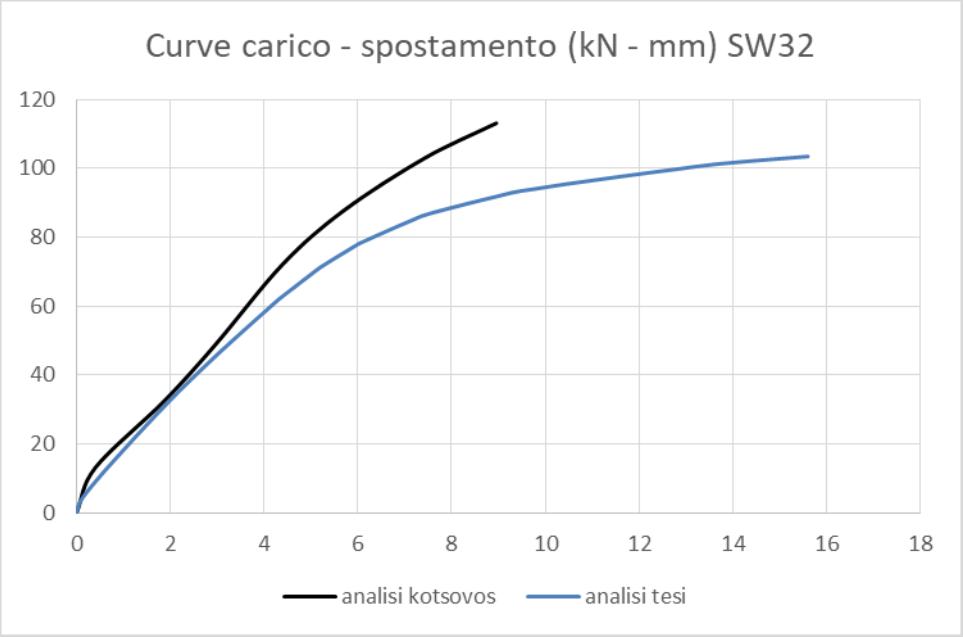

Figura 40 Confronto tra curve di carico - spostamento per il muro SW32

La mia analisi ha sottostimato carico e spostamento di circa il 14% e 37%, per il muro SW30. Anche il confronto con l'analisi di Kotsovos (2015) (muro SW32) vede un carico leggermente più basso (circa il 10% in meno) ma uno spostamento ultimo più alto (75% circa in più). Anche in questo caso, le differenze devono essere dovute alle incertezze di modello.

Le tensioni di compressione più elevate (in valore assoluto) si ottengono lungo l'asse y. Riporto in questo caso prima i valori di tensione, e poi quelli di deformazione, perpendicolari a tale asse, cioè i valori in x (le deformate sono scalate di un fattore 10):

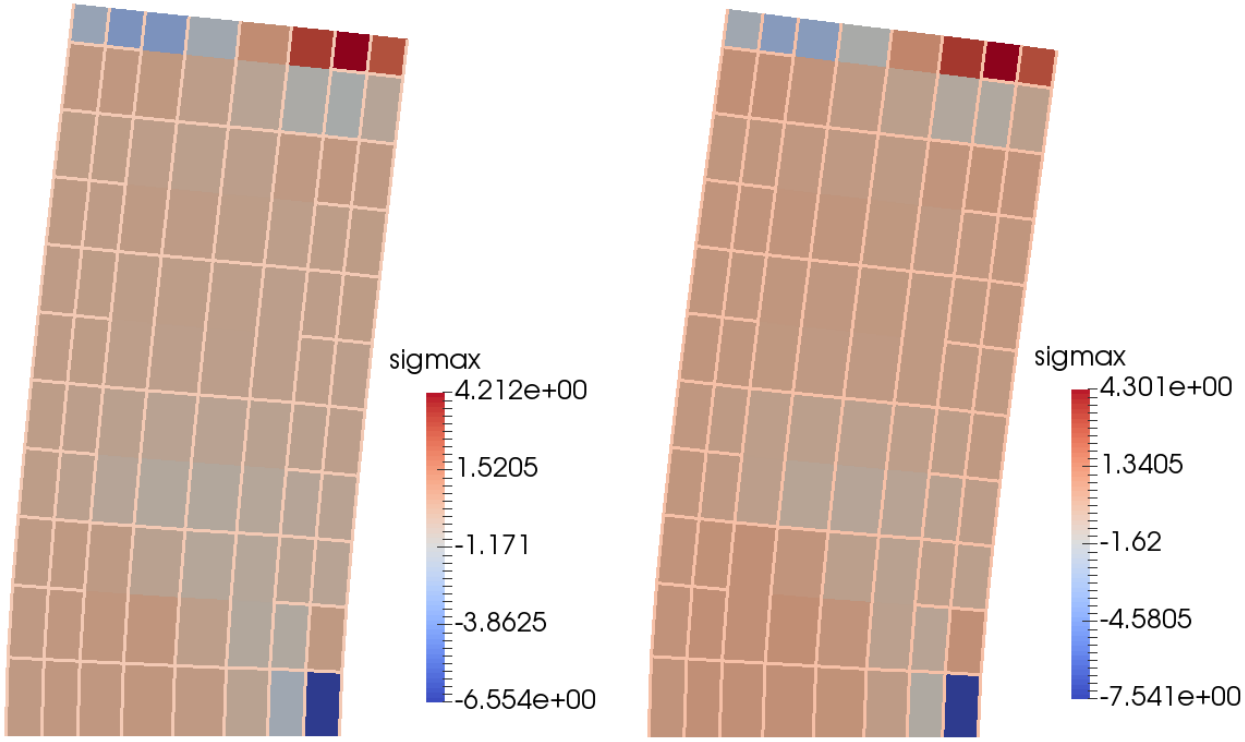

Figura 41 Tensioni a rottura lungo l'asse x (SW30 a sinistra, SW32 a destra)

Come si può osservare, i valori di compressione σ_x più elevati si ottengono alla base del muro, nella zona soggetta a maggior compressione verticale. Questo è fondamentalmente dovuto alla presenza dell'incastro e all'effetto del coefficiente di Poisson. Per quella che è la superficie di rottura del calcestruzzo, tale compressione in x contribuisce significativamente alla resistenza del muro. Può quindi accadere che a raggiungere la rottura non sia l'elemento con la maggiore sollecitazione lungo uno dei due assi. Questo succede nel caso delle travi A1 e A3 per un effetto simile (tensione trasversale dovuta all'applicazione del carico) e nel caso delle colonne A5a e A10a.

Gli alti valori di tensione alla sommità del muro sono invece posseduti dagli elementi che modellano la metà di trave superiore. Molto probabilmente queste tensioni sono semplicemente dovute al fatto che, avendo tale elementi comportamento lineare elastico, hanno conservato un valore di rigidezza più elevato rispetto agli elementi adiacenti.

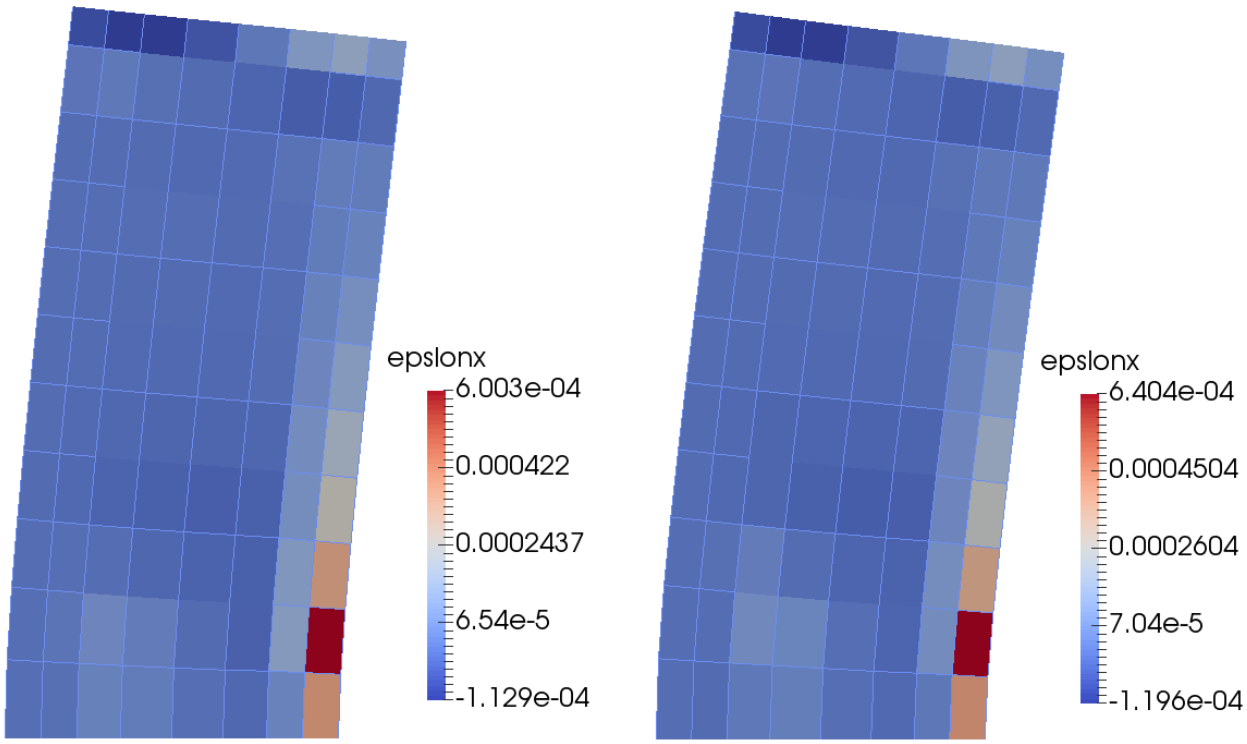

Figura 42 Deformazioni a rottura lungo l'asse x (SW30 a sinistra, SW32 a destra)

I valori più elevati di deformazione ε_x si trovano invece in corrispondenza degli elementi immediatamente al di sopra di quelli che avevano il maggior valore assoluto di σ_x. Essi hanno infatti una minor tensione di compressione verticale e quindi una minor espansione massima potenziale, ma si deformano di più perché sono più liberi di farlo essendo un po' più lontani dall'incastro.

3.4 Wang e Zhang (2000), muro a taglio SW7

Il muro non è presente nel libro di Kotsovos (2015) ed è stato analizzato sperimentalmente da Wang e Zhang (2000) solo sotto carico ciclico. Confronterò quindi la mia analisi con l'inviluppo (fornito dagli autori dell'articolo) delle curve carico – spostamento relative ai singoli cicli di carico. Si tratta ovviamente di un confronto meno diretto: il comportamento delle strutture sotto carico ciclico è significativamente diverso da quello sotto carico monotono. Mi è sembrato tuttavia un confronto utile.

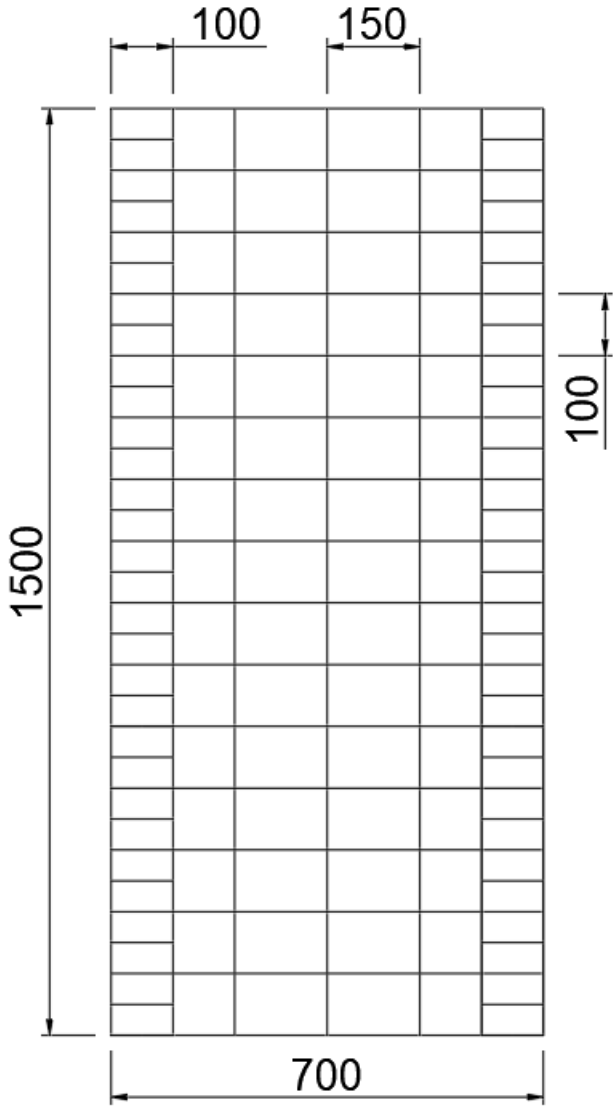

Figura 43 Geometria del muro (misure in mm)

È presente un carico assiale costante oltre al carico orizzontale crescente fino a rottura. Il carico assiale è di 499 kN, corrispondente a una pressione di circa 7.1 MPa. Lo spessore del muro è di 100 mm).

È armato verticalmente con sette coppie di barre: le tre centrali sono barre ϕ8, le quattro laterali (due coppie per ogni estremità) sono ϕ14. Sono inoltre presenti coppie di barre orizzontali ϕ8 @ 100 mm e armatura orizzontale di confinamento alle estremità del muro sotto forma di staffe chiuse ϕ6 @ 50 mm. Le barre hanno i seguenti valori di resistenza:

diametro (mm)	Resistenza a snervamento (MPa)	Resistenza ultima (MPa)
14	405	540
8	305	440
6	305	440

Tabella 3 Caratteristiche fisiche delle barre di acciaio

Non essendo specificata la rigidezza in fase plastica delle barre ho assunto il valore di 2000 MPa.

La rottura avviene per splitting del calcestruzzo e le armature longitudinali raggiungono lo snervamento in entrambe le estremità del muro.

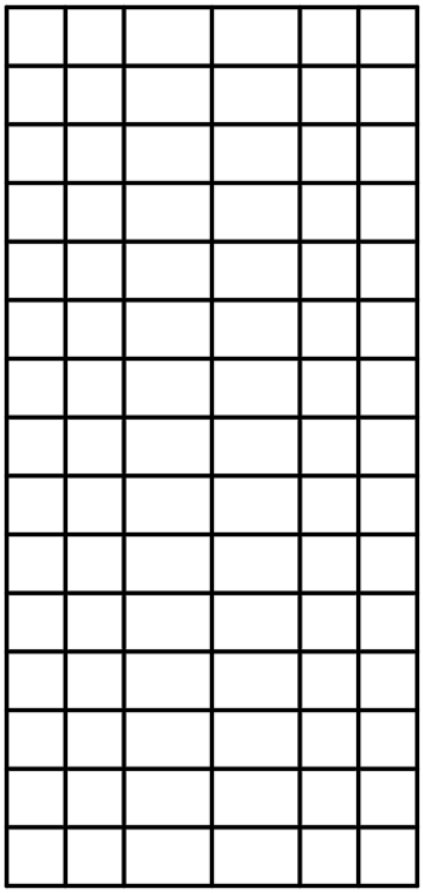

Figura 44 Mesh utilizzata nell'analisi: come si può vedere, tutte le linee sono spesse ed hanno quindi delle barre

In questo caso mi è sembrata abbastanza ovvia la scelta sulla discretizzazione numerica che ricalca esattamente la disposizione delle barre, ad eccezione delle staffe chiuse ϕ6. Esse sono state raggruppate, in coppia, visto che il passo verticale della mesh (100 mm) è doppio rispetto a quello delle staffe in questione (50 mm).

I valori sperimentali di carico ultimo e spostamento orizzontale ultimo alla sommità del muro sono stati di 201 kN e 31.3 mm:

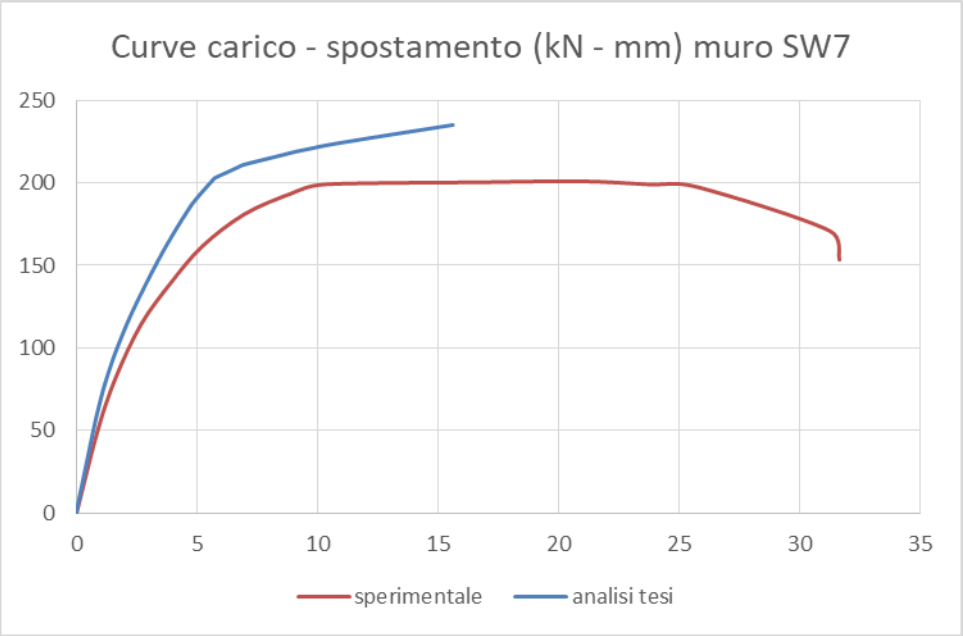

Figura 45 Confronto tra curve di carico spostamento per il muro SW7

In questo caso c'è stata una sovrastima del carico di circa il 17% e una sottostima dello spostamento di circa il 50%. Ricordo anche che, essendo le mie analisi in controllo di forza, non è mai possibile catturare eventuali tratti di post-picco come quelli dello sperimentale appena riportato.

Riporto le tensioni σ_y e le deformazioni ε_y (la deformata del muro è scalata di un fattore 10):

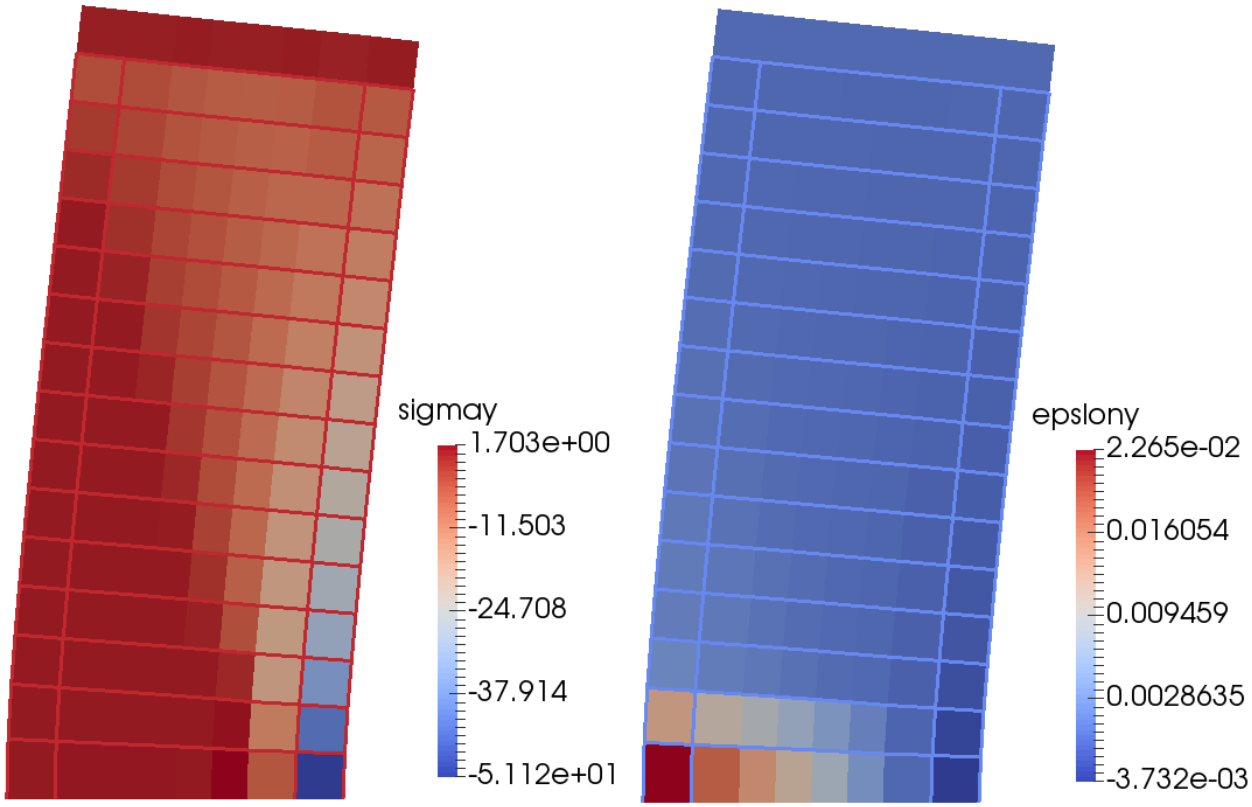

Figura 46 Tensioni e deformazioni a rottura lungo l'asse y

Come si può vedere, ed è tipico per il cemento armato, la maggior parte delle deformazioni in y è accumulata alla base del muro, nella zona in trazione.

L'analisi termina per rottura del calcestruzzo alla base del muro, in zona compressa.

Segnalo anche che nell'analisi si sono plasticizzate le due coppie di barre ϕ14 in trazione e la coppia di barre in compressione più vicina all'estremità del muro.

3.5 Hognestad (1951), colonne A5a e A10a

Le due colonne sono testate sperimentalmente in Hognestad (1951) e non sono presenti nel libro di Kotsovos (2015). La geometria delle colonne è rappresentata in figura:

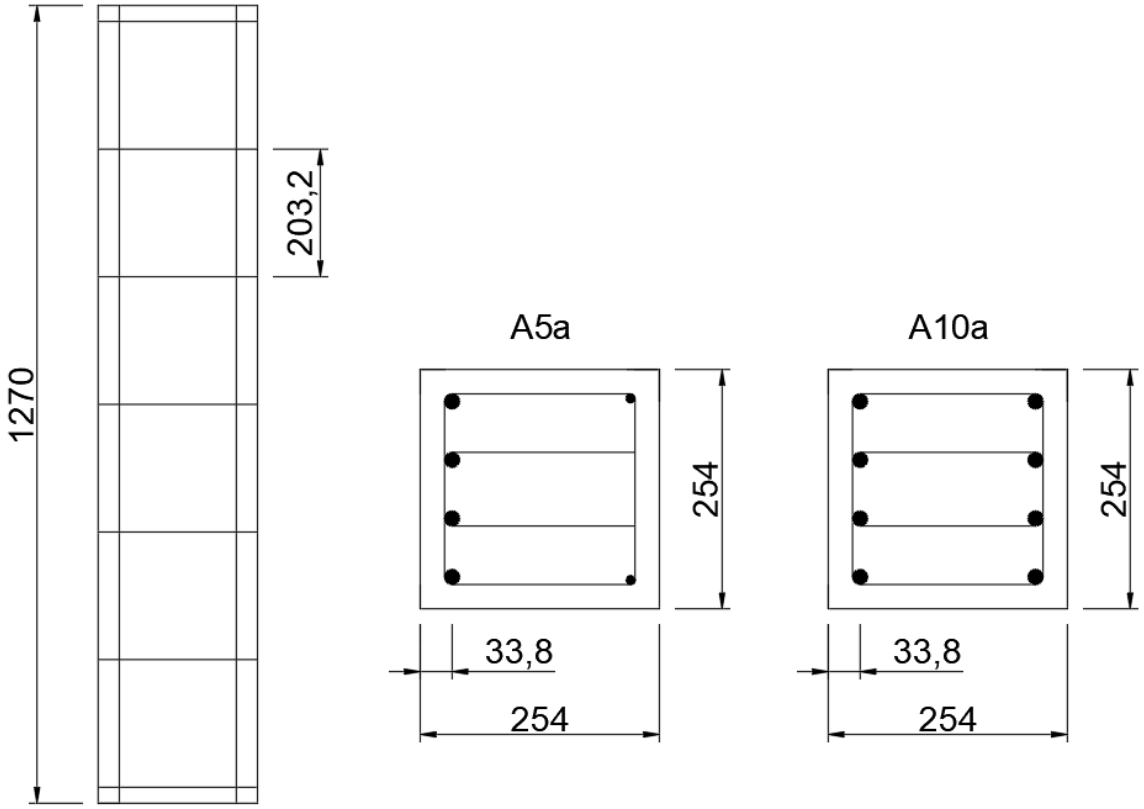

Figura 47 Geometria delle colonne (tutte le misure sono in mm)

Il carico consiste in una forza verticale applicata con un'eccentricità di 317.5 mm rispetto al baricentro della sezione. La sollecitazione sarà quindi di sforzo normale e momento flettente.

Sono incernierate ad entrambe le estremità e differiscono quindi solo per la quantità di armatura in zona compressa e per la resistenza del calcestruzzo. In zona tesa hanno entrambe quattro barre ϕ15.9; in zona compressa l'A10a ha le stesse barre della zona tesa, l'A5a ha due barre ϕ9.5. L'armatura orizzontale è invece composta da staffe di diametro 6.4 mm. La resistenza del calcestruzzo è simile ed è di 33.2 MPa per l'A5a e 35.2 MPa per l'A10a.

Riporto in tabella i valori di resistenza delle barre di acciaio:

diametro (mm)	Resistenza a snervamento (MPa)	Resistenza ultima (MPa)	Modulo di rigidezza in fase plastica (MPa)
15.9	303	516	1093
9.5	414	603	1021
6.4	425	579	925

Tabella 4 Caratteristiche fisiche delle barre di acciaio

La rottura avviene in entrambi i casi per splitting del calcestruzzo e buckling dell'armatura in compressione. Le armature in zona tesa raggiungono lo snervamento in entrambe le colonne, quelle in zona compressa solo nell'A5a.

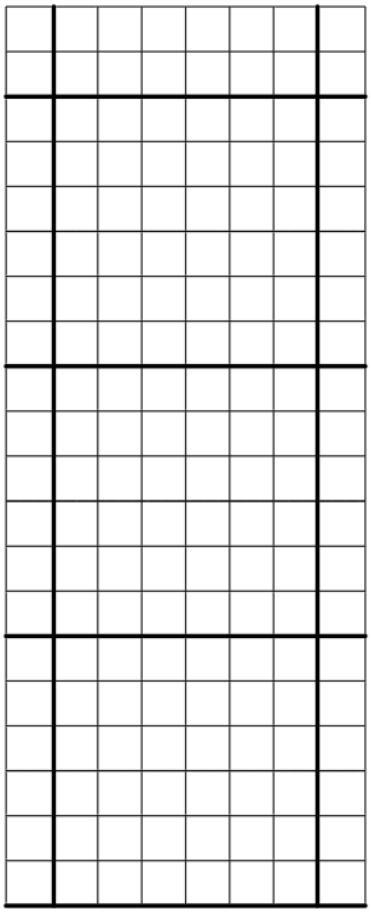

Figura 48 Mesh della struttura: le sei colonne di elementi interni hanno una larghezza leggermente inferiore (31.1 mm) rispetto a quelli delle estremità (33.8 mm); le linee spesse rappresentano le barre di acciaio

Ho modellato per simmetria soltanto metà delle colonne (la metà superiore). La mesh è molto semplice e prevede le barre verticali nella loro posizione reale e una suddivisione interna (orizzontale) in sei parti. Verticalmente le colonne sono state invece divise in 20 parti. Non è presente nel disegno, ma è

stata usata nell'analisi, un'ulteriore fila orizzontale di elementi sulla sommità della colonna; sono di materiale infinitamente rigido e modellano la cerniera con cui è stato trasferito il carico.

I valori sperimentali di carico ultimo e spostamento ultimo al centro delle colonne sono stati di 214 kN e 11.2 mm per la colonna A5a e di 205 kN e 19.3 mm per la colonna A10a:

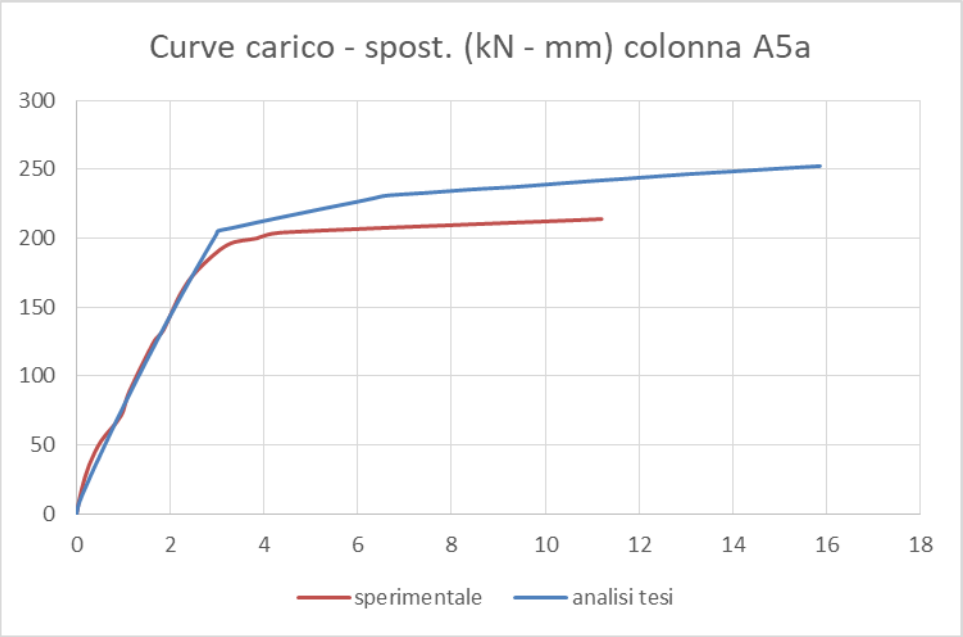

Figura 49 Confronto tra curve di carico - spostamento per la colonna A5a

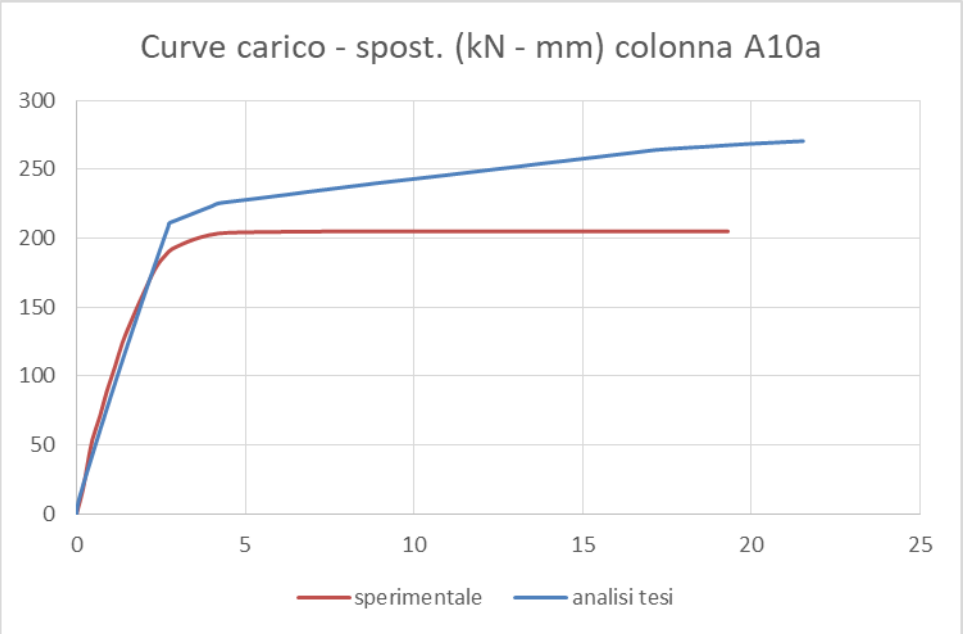

Figura 50 Confronto tra curve di carico - spostamento per la colonna A10a

Riporto i valori delle tensioni lungo y e lo stato fessurativo (questa volta la deformata è scalata di un fattore 5):

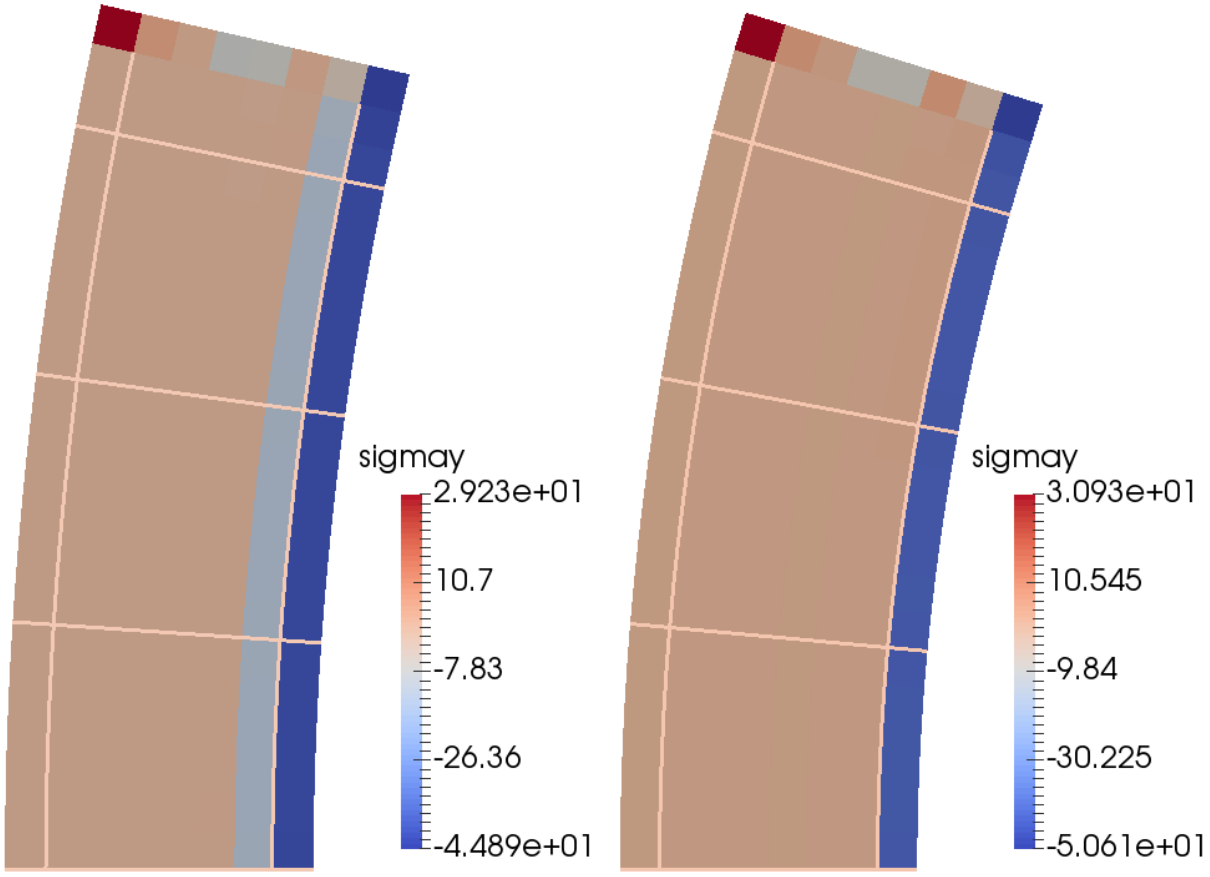

Figura 51 Tensioni a rottura lungo l'asse y (colonna A5a a sinistra e colonna A10a a destra)

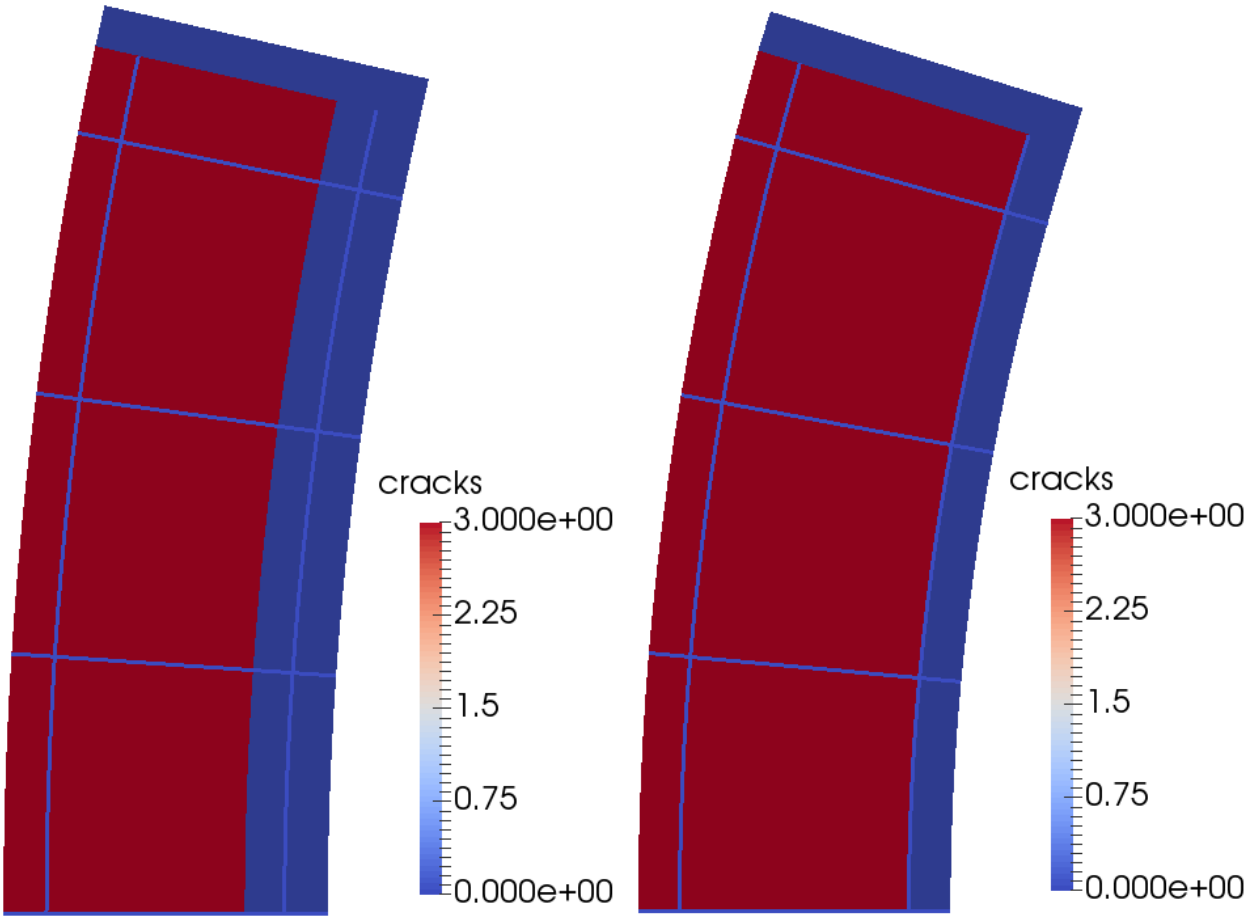

Figura 52 Stato fessurativo a rottura (colonna A5a a sinistra e colonna A10a a destra)

Come si può notare, nella colonna A5a c'è una fila in meno di elementi fessurati. Andando a guardare lo stato fessurativo, si può notare infatti che quegli elementi, per tale colonna, non sono in trazione ma in compressione. Gli elevati valori di tensione sigma nell'angolo in alto a sinistra delle colonne sono invece semplicemente dovuti all'applicazione del carico.

In entrambe le analisi si sono plasticizzate le sole barre verticali in zona tesa, e ha raggiunto la rottura un elemento di calcestruzzo in zona compressa, all'estremità destra del muro. In ognuno dei due casi si è trattato di un elemento adiacente a quelli a contatto con le barre orizzontali. Tali barre forniscono infatti un effetto di confinamento che aumenta leggermente la resistenza degli elementi che si trovano in prossimità.

Conclusioni e sviluppi futuri

All'inizio di questo lavoro l'obiettivo era quello di vedere se un programma del genere potesse essere scritto in poco tempo, con conoscenze di informatica soltanto di base, a partire dal libro di Griffiths et al. (2014) e col metodo puramente incrementale.

Ritengo che l'obiettivo sia stato raggiunto, mi sembra che il programma funzioni e possa essere una buona base di partenza per lavori futuri. Nei risultati, gli ordini di grandezza sono stati rispettati sotto tutti i punti di vista (con la probabile eccezione della tensione nelle barre a taglio). I carichi di rottura sono stimati con precisione abbastanza buona, gli spostamenti hanno un'approssimazione un po' maggiore, ma sembra accettabile. La modalità di rottura è sempre stata colta con esattezza, anche se non c'è variabilità da questo punto di vista, essendo tutti i test terminati per splitting del calcestruzzo. Si riescono ad avere informazioni buone sulla duttilità delle strutture: questo lo si può cogliere sia dalle curve di carico – spostamento che andando a vedere le barre che si sono plasticizzate. Nonostante le approssimazioni, mi sembra ci sia un buon comportamento dal punto di vista della fessurazione. Un altro risultato che notavo è la capacità di cogliere l'effetto di confinamento sul calcestruzzo fornito dalle condizioni al contorno e, in misura minore, dalle staffe a taglio.

Dicevo dell'eccezione sulla tensione nelle barre a taglio, con alcuni dei risultati discordanti dagli sperimentali. C'è però da dire che poche strutture (soltanto quelle dei primi due esempi) avevano questo tipo di informazione nei rispettivi riferimenti bibliografici. In ogni caso, uno dei motivi per cui il comportamento di queste barre è importante è l'avere a che fare con il traliccio di resistenza a taglio (vedi paragrafo 2.5), anche se l'opinione di Kotsovos (2015) è che tale traliccio in realtà non si forma. La prima causa che mi viene in mente per giustificare questa discordanza è la semplificazione sulla mancanza di fessure ordite in direzione qualsiasi. Dicevo anche della maggiore approssimazione con cui sono stati stimati gli spostamenti. In diversi casi c'è stata una sottostima del risultato e uno dei motivi potrebbe essere dovuto al fatto che le analisi numeriche sono in controllo di carico mentre quelle sperimentali in controllo di spostamento.

Per quanto riguarda l'uso del metodo puramente incrementale non è ovviamente possibile sapere con certezza quanta dell'approssimazione delle analisi è dovuta ad esso e quanta alle altre incertezze di modello. Un'idea migliore la si può avere guardando in particolare i confronti con le analisi del libro di Kotsovos. Anch'esse non differiscono dalle analisi della tesi per il solo metodo in questione ma hanno in comune il fatto di essere delle analisi numeriche. Comunque, alla luce dei risultati, penso si possa dire che il metodo è utilizzabile per questo tipo di analisi. Come già detto, e probabilmente dovuto in

particolare agli ottimi risultati unidimensionali mostrati al capitolo 1, la mia opinione personale sul metodo è che in ogni analisi si possa raggiungere un'approssimazione molto buona o anche trascurabile, sempre a fronte di uno sforzo computazionale ragionevole.

Per quanto riguarda l'estensione del programma ad analisi più complesse e generiche delle attuali, in questo momento mi sembra un importante traguardo il fatto di arrivare ad eseguire, senza importanti limitazioni, le analisi di pushover (vedi ad esempio CEN, *Eurocode 8* (2004), Abunama (2017) e Parducci (2016)). Intendo quindi su qualsiasi tipo di struttura in cemento armato, con qualsiasi tipo di elementi (almeno i più comuni), senza semplificazioni riguardo alla fessurazione (o al limite trascurando la sola resistenza a trazione) e col calcestruzzo di Kotsovos (2015) o con altri modelli. Passo quindi brevemente in rassegna ciò che è necessario a questo scopo.

Per quanto riguarda le analisi tridimensionali (vedi ad esempio Carpinteri (1995)) bisogna prima di tutto precisare che esse hanno delle differenze importanti rispetto alle analisi in stato di tensione piana, a prescindere dal codice di calcolo. Ad esempio, nel caso bidimensionale non è possibile tenere conto del confinamento fuori dal piano, se non in maniera indiretta. Anche la parte relativa alla fessurazione ha una teoria un po' più complessa rispetto a quella semplificata dalle due dimensioni (per questo aspetto in particolare vedi Kotsovos (2015)). Senza tener conto del fatto che lo spazio tridimensionale permette di avere caratteristiche della sollecitazione che non possono essere presenti in 2D. Tuttavia, la struttura del codice dovrebbe rimanere la stessa e permettere le analisi 3D come quelle 2D.

L'estensione del programma a strutture con geometria qualsiasi (Griffiths et al. (2014)) dovrebbe essere un'operazione abbastanza banale, con modifica della sola parte relativa al preprocessing. Per inserire barre in direzione obliqua sarebbe necessario ricavare preliminarmente la loro matrice direttamente nello spazio bidimensionale (o tridimensionale), e non in quello unidimensionale, come fatto attualmente dal programma. Il fatto di eseguire analisi in controllo di spostamento (Griffiths et al. (2014)), oltre che di carico, dovrebbe essere ancora più semplice in quanto il programma è già in grado di eseguire le analisi in questo modo ma manca solo la parte relativa al calcolo delle reazioni vincolari.

Per quanto riguarda l'implementazione di altri elementi e materiali bisognerebbe è necessario uno studio più approfondito. Ad esempio, per quanto riguarda le subroutine fornite dal libro di Griffiths et al. potrebbe non essere possibile implementare leggi costitutive con tratti di softening, anche se non dovrebbe essere difficile modificare allo scopo tali subroutine. A questo proposito devo anzi esprimere il mio ottimismo nei confronti del libro in questione. Mi sembra che suoi codici abbiano ottime

caratteristiche di adattabilità e quindi penso che il codice possa essere adattato a fare tutto quello che viene descritto in questo paragrafo e ad eventuali ulteriori modellazioni.

L'estensione a un modello di calcestruzzo con formazione di fessure in direzione qualsiasi è stata descritta al relativo paragrafo. La resistenza a trazione del calcestruzzo (Kotsovos (2015)) dovrebbe invece prevedere una modellazione dello stesso tipo della rottura dei materiali descritta fra poco.

Per arrivare ad eseguire le analisi di pushover un'altra modellazione necessaria è quella relativa allo scarico tensionale (Parducci (2016), Massonet e Save (2008)). Alcuni elementi potrebbero andarvi incontro, ad esempio, in seguito a rotture locali. Da questo punto di vista il modello di Kotsovos è molto semplice, prevedendo una curva di scarico (e di eventuale ricarico, fino alla tensione massima che si era raggiunta) di pendenza pari alla rigidezza iniziale del calcestruzzo. Questa stessa modellazione di scarico e ricarico può essere usata per l'acciaio (Ottosen e Ristinmaa (2005)).

Teoricamente, per poter prendere in considerazione un generico sistema di carichi non proporzionali (Parducci (2016)) sarebbe necessario che il programma sia in grado di eseguire analisi cicliche. Tuttavia, di fatto, l'implementazione dello scarico tensionale dovrebbe essere sufficiente a permettere anche questo sistema di carichi.

È necessario infine modellare la rottura dei materiali (Kotsovos (2015)): splitting del calcestruzzo e rottura della barre di acciaio. Questo tipo di rottura prevede che gli elementi perdano la loro rigidezza e rilascino le tensioni che avevano accumulato fino a quel momento. Per quanto riguarda la perdita di rigidezza dovrebbe essere sufficiente porre pari a zero i termini della matrice di rigidezza dell'elemento. Per quanto riguarda il rilascio delle tensioni, un metodo semplice (Crisfield et al. (2012)) dovrebbe consistere nell'applicare delle forze (fittizie) ai nodi dell'elemento che ha raggiunto la rottura: tali forze dovrebbero essere uguali e opposte a quelle che l'elemento aveva accumulato fino a quel punto dell'analisi.

Conclusions and future work

At the beginning of this work the aim was to see if such a programme could be written in a short time, with only basic computer knowledge, starting from Griffiths et al.'s book (2014) and using the purely incremental method.

I believe that the objective has been achieved, it seems to me that the program works and can be a good starting point for future work. In the results, the orders of magnitude were respected in all aspects (with the probable exception of the tension in shear bars). Failure loads are estimated with fairly good accuracy, displacements have a slightly higher approximation, but it seems acceptable. The mode of rupture has always been accurately grasped, although there is no variability from this point of view, all tests having been completed for concrete splitting. You can get good information about the ductility of the structures: this can be seen both from load-shift curves and from the bars that have been plasticised. Despite the approximations, it seems to me that there is good behaviour from the point of view of cracking. Another result that i noticed is the ability to capture the confinement effect on concrete provided by boundary conditions and, to a lesser extent, by shear stirrups.

I was talking about the exception about tension in shear bars, with some of the results differing from the experimental ones. It must be said, however, that few structures (only those of the first two examples) had this type of information in their bibliographical references. In any case, one of the reasons why the behaviour of these bars is important is to deal with the shear strength truss (see paragraph 2.5), although Kotsovos' (2015) opinion is that such a truss does not actually form. The first cause that comes to my mind to justify this discordance is the simplification on the lack of warped cracks in any direction. I was also talking about the greater approximation with which the displacements have been estimated. In several cases there has been an underestimation of the result and one of the reasons could be that the numerical analyses are in load control while the experimental ones are in displacement control.

As far as the use of purely incremental method is concerned, it is obviously not possible to know with certainty how much of the approximation of the analysis is due to it and how much to other model uncertainties. A better idea can be had looking in particular at the comparisons with the analysis of Kotsovos' book. They too do not differ from the analysis of the thesis for the method in question alone but have in common the fact that they are numerical analysis. However, in the light of the results, i think we can say that the method can be used for this type of analysis. As already said, and probably due in particular to the excellent one-dimensional results shown in chapter 1, my personal opinion on

the method is that in every analysis a very good or even negligible approximation can be achieved, always with a reasonable computational effort.

As far as the extension of the program to more complex and generic analysis than the current ones is concerned, at this moment it seems to me an important goal to be able to perform pushover analysis without important limitations (see for example CEN, Eurocode 8 (2004), Abunama (2017) and Parducci (2016)). I mean therefore on any type of reinforced concrete structure, with any type of elements (at least the most common), without simplifications regarding cracking (or the limit neglecting only tensile strength) and with Kotsovos (2015) concrete (preferably) or with other models. I therefore briefly review what is necessary for this purpose.

As far as three-dimensional analysis are concerned (see for example Carpinteri (1995)) it must first of all be pointed out that they have important differences compared to plane stress analysIs, regardless of the calculation code. For example, in two-dimensional case it is not possible to take into account the out of plane confinement, if not in an indirect way. Also the cracking part has a slightly more complex theory than the simplified two-dimensional one (for this aspect in particular see Kotsovos (2015)). Without taking into account the fact that the three-dimensional space allows to have internal forces that cannot be present in 2D. However, the structure of the code should remain the same and allow 3D analysis as 2D.

The extension of the program to structures with any geometry (Griffiths et al. (2014)) should be a fairly trivial operation, with only the preprocessing part modified. To insert bars in an oblique direction, it would be necessary to preliminarily derive their matrix directly in two-dimensional (or three-dimensional) space, and not in one-dimensional space, as the program currently does. The fact of performing displacement control analysis (Griffiths et al. (2014)), as well as load control analysis, should be even simpler as the program is already able to perform the analysis in this way but only the part related to the calculation of constraining reactions is missing.

As far as the implementation of other elements and materials is concerned, a more in-depth study is needed. For example, with regard to the subroutines provided by Griffiths et al.'s book, it may not be possible to implement constitutive laws with softening traits, although it should not be difficult to modify these subroutines for this purpose. On the contrary, i must express my optimism about the book in question. It seems to me that its codes have very good adaptability features, so I think the code can be adapted to do everything described in this paragraph and to any further modelling.

The extension to a concrete model with the formation of cracks in any direction has been described in the relevant paragraph. Tensile strength of concrete (Kotsovos (2015)), on the other hand, should provide for modelling of the same type as failure of materials described below.

In order to carry out pushover analysis another necessary modeling is the one related to tensional unloading (Parducci (2016), Massonet and Save (2008)). Some elements may be affected, for example, as a result of local failures. From this point of view, Kotsovos model is very simple, providing for an unloading curve (and possible reloading, up to the maximum tension that was reached) of a slope equal to the initial stiffness of concrete. This same unloading and reloading model can be used for steel (Ottosen and Ristinmaa (2005)).

Theoretically, in order to take into consideration a generic system of non-proportional loads (Parducci (2016)) it would be necessary for the program to be able to perform cyclic analysis. However, in fact, the implementation of tensional unloading should be sufficient to allow this system of loads as well.

Finally, it is necessary to model the failure of materials (Kotsovos (2015)): splitting of concrete and rupture of steel bars. This type of failure involves the elements losing their rigidity and releasing the tensions they had accumulated up to that point. As far as the loss of stiffness is concerned, it should be sufficient to set the terms of the element's stiffness matrix to zero. As far as the release of tensions is concerned, a simple method (Crisfield et al. (2012)) should consist in applying (fictitious) forces to the nodes of the element that has reached rupture: these forces should be equal and opposite to those that the element had accumulated up to that point in the analysis.

Appendice

A1 Codice di calcolo in FORTRAN

Riporto il codice di calcolo vero e proprio. Esso è stato scritto a partire dal *Program p51* di Griffiths et al. (2014), anch'esso riportato in appendice. Tale programma permette l'analisi statica di un solido lineare elastico. Per favorire il confronto tra i due codici ho evidenziato in giallo le parti aggiunte o modificate:

```
PROGRAM p51_kot
  USE main
  USE geom
  IMPLICIT NONE
  INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15),npri=1
  INTEGER::fixed_freedoms,i,iel,k,loaded_nodes,ndim=2,ndof,nels,neq,nip=4, &
    nlen,nn,nod=4,nodof=2,nprops=2,np_types,nr,nst=3,nxe,nye,crush,i1,i2,  &
    iel_r,ndof_r,nels_r,nod_r=2,np_types_r,nprops_r=5,step,yield_fail
  REAL(iwp)::det,one=1.0_iwp,penalty=1.0e20_iwp,zero=0.0_iwp,km_r,A,b,beta,&
    C,d,dtim=1.0,ell,ell0,eps_r,fc,fct,Ge,Gt,Inv1,Inv2,Ke,Kt,sigma_r,      &
    sigma0,sigma1,tau0,tau0c,tau0e,tau0u,teta,thick
  CHARACTER(LEN=15)::argv,element,dir
  LOGICAL::solid=.TRUE.
 !----------------------dynamic arrays-------------------------------------
  INTEGER,ALLOCATABLE::etype(:),g(:),g_g(:,:),g_num(:,:),kdiag(:),nf(:,:), &
    no(:),node(:),num(:),sense(:),crack(:),etype_r(:),g_num_r(:,:),g_r(:), &
    num_r(:),yield(:)
  REAL(iwp),ALLOCATABLE::bee(:,:),coord(:,:),dee(:,:),der(:,:),deriv(:,:), &
    eld(:),fun(:),gc(:),g_coord(:,:),jac(:,:),km(:,:),kv(:),loads(:),      &
    points(:,:),prop(:,:),sigma(:),value(:),weights(:),x_coords(:),        &
    y_coords(:),coord_r(:,:),eld_r(:),eps(:),Et(:),etensor(:,:,:),         &
    etensor_r(:),prop_r(:,:),prop_r0(:,:),prop_s(:),tensor(:,:,:),         &
    tensor_r(:),totd(:),vt(:)
 !----------------------input and initialisation--------------------------
  CALL getname(argv,nlen)
  OPEN(10,FILE=argv(1:nlen)//'.dat')
  OPEN(11,FILE=argv(1:nlen)//'.res')
  element='quadrilateral'
  dir='y'
  READ(10,*)nxe,nye,np_types,nels_r
  np_types_r=1
  IF(nels_r>0)READ(10,*)np_types_r
  CALL mesh_size(element,nod,nels,nn,nxe,nye)
  ndof=nod*nodof
  ndof_r=nod_r*nodof
  ALLOCATE(nf(nodof,nn),points(nip,ndim),g(ndof),g_coord(ndim,nn),fun(nod),&
    coord(nod,ndim),jac(ndim,ndim),g_num(nod,nels),der(ndim,nod),          &
    deriv(ndim,nod),bee(nst,ndof),km(ndof,ndof),eld(ndof),weights(nip),    &
    g_g(ndof,nels),prop(nprops,nels),num(nod),x_coords(nxe+1),             &
    y_coords(nye+1),etype(nels),gc(ndim),dee(nst,nst),sigma(nst),          &
    coord_r(nod_r,ndim),crack(nels),eld_r(ndof_r),eps(nst),Et(nels),       &
    etensor(nst,nip,nels),etensor_r(nels_r),etype_r(nels_r),               &
    g_num_r(nod_r,nels_r),g_r(ndof_r),num_r(nod_r),prop_r(nprops_r,nels_r),&
    prop_r0(nprops_r,np_types_r),prop_s(nprops),tensor(nst,nip,nels),      &
    tensor_r(nels_r),vt(nels),yield(nels_r))
  IF(nels_r>0)READ(10,*)g_num_r
  READ(10,*)fc,beta
  IF(np_types>1)THEN
    READ(10,*)prop_s
```

```
50      IF(prop_s(1)<0)prop_s(1)=penalty/1.0e5
51    END IF
52    IF(nels_r>0)READ(10,*)prop_r0
53    fct=0.3*(fc-8.0)**(2.0/3.0)
54    IF(fc<=31.7)THEN
55      A=0.516
56      b=2.0+1.81*10.0**(-8.0)*fc**4.461
57      C=3.573
58      d=2.12+0.0183*fc
59    ELSE
60      A=0.516/(1.0+0.0027*(fc-31.7)**2.397)
61      b=2.0+1.81*10.0**(-8.0)*fc**4.461
62      C=3.573/(1.0+0.0134*(fc-31.7)**1.414)
63      d=2.7
64    END IF
65    Ke=11000.0+3.2*fc**2.0
66    Ge=9224.0+136.0*fc+3296.0*10.0**(-15.0)*fc**8.273
67    etype=1
68    etype_r=1
69    IF(np_types>1) READ(10,*)etype
70    IF(np_types_r>1) READ(10,*)etype_r
71    READ(10,*)x_coords,y_coords
72    READ(10,*)thick
73    prop_r0(5,:)=prop_r0(5,:)/thick
74    nf=1
75    READ(10,*)nr,(k,nf(:,k),i=1,nr)
76    CALL formnf(nf)
77    neq=MAXVAL(nf)
78    ALLOCATE(loads(0:neq),kdiag(neq),totd(0:neq))
79    kdiag=0
80  !-----------------------loop the elements to find global arrays sizes-----
81    elements_1: DO iel=1,nels
82      CALL geom_rect(element,iel,x_coords,y_coords,coord,num,dir)
83      CALL num_to_g(num,nf,g)
84      g_num(:,iel)=num
85      g_coord(:,num)=TRANSPOSE(coord)
86      g_g(:,iel)=g
87      CALL fkdiag(kdiag,g)
88    END DO elements_1
89    elements_1_r: DO iel_r=1,nels_r
90      prop_r(:,iel_r)=prop_r0(:,etype_r(iel_r))
91    END DO elements_1_r
92    CALL mesh(g_coord,g_num,argv,nlen,12)
93    DO i=2,neq
94      kdiag(i)=kdiag(i)+kdiag(i-1)
95    END DO
96    ALLOCATE(kv(kdiag(neq)))
97    WRITE(11,'(2(A,I5))')                                                  &
98      " There are",neq," equations and the skyline storage is",kdiag(neq)
99  !------------------------element stiffness integration and assembly--------
100   etensor=zero
101   tensor=zero
102   etensor_r=zero
103   tensor_r=zero
104   totd=zero
105   crack=0
106   crush=0
107   yield=0
108   yield_fail=0
109   Et=9.0*Ke*Ge/(3.0*Ke+Ge)
110   vt=(3.0*Ke-2.0*Ge)/(6.0*Ke+2.0*Ge)
111   step=0
112   steps: DO
113     step=step+1
114     WRITE(*,*)step
115     kv=zero
116     gc=one
117     nip=4
```

```
118       DEALLOCATE(points,weights)
119       ALLOCATE(points(nip,ndim),weights(nip))
120       CALL sample(element,points,weights)
121       elements_2: DO iel=1,nels
122         IF(etype(iel)==1)THEN
123           prop(1,iel)=Et(iel)
124           prop(2,iel)=vt(iel)
125         ELSE
126           prop(1,iel)=prop_s(1)
127           prop(2,iel)=prop_s(2)
128         END IF
129         CALL fmdsig(dee,prop(1,iel),prop(2,iel))
130         IF(crack(iel)==1)THEN
131           dee(3,3)=dee(3,3)*beta
132         END IF
133         IF(crack(iel)==2)THEN
134           dee(1,1)=dee(1,1)*0.0001
135           dee(1,2)=zero
136           dee(2,1)=zero
137           dee(3,3)=dee(3,3)*beta
138         END IF
139         IF(crack(iel)==3)THEN
140           dee(2,2)=dee(2,2)*0.0001
141           dee(1,2)=zero
142           dee(2,1)=zero
143           dee(3,3)=dee(3,3)*beta
144         END IF
145         IF(crack(iel)==4)THEN
146           dee(1,1)=dee(1,1)*0.0001
147           dee(2,2)=dee(2,2)*0.0001
148           dee(1,2)=zero
149           dee(2,1)=zero
150           dee(3,3)=dee(3,3)*beta
151         END IF
152         num=g_num(:,iel)
153         coord=TRANSPOSE(g_coord(:,num))
154         g=g_g(:,iel)
155         km=zero
156         int_pts_1: DO i=1,nip
157           CALL shape_fun(fun,points,i)
158           CALL shape_der(der,points,i)
159           jac=MATMUL(der,coord)
160           det=determinant(jac)
161           CALL invert(jac)
162           deriv=MATMUL(jac,der)
163           CALL beemat(bee,deriv)
164           km=km+MATMUL(MATMUL(TRANSPOSE(bee),dee),bee)*det*weights(i)*gc(1)
165           elements_2_r: DO iel_r=1,nels_r
166             num_r=g_num_r(:,iel_r)
167             i1=0; i2=0
168             IF(num_r(1)==num(1))i1=1; IF(num_r(1)==num(2))i1=3
169             IF(num_r(1)==num(3))i1=5; IF(num_r(1)==num(4))i1=7
170             IF(num_r(2)==num(1))i2=1; IF(num_r(2)==num(2))i2=3
171             IF(num_r(2)==num(3))i2=5; IF(num_r(2)==num(4))i2=7
172             IF(i1/=0.AND.i2/=0)THEN
173               coord_r=TRANSPOSE(g_coord(:,num_r))
174               IF(coord_r(2,1)==coord_r(1,1))THEN
175                 i1=i1+1; i2=i2+1
176               END IF
177               ell0=sqrt((coord_r(2,1)-coord_r(1,1))**2.0+(coord_r(2,2)        &
178               -coord_r(1,2))**2.0)
179               km_r=prop_r(1,iel_r)*prop_r(5,iel_r)/ell0
180               km(i1,i1)=km(i1,i1)+km_r/nip
181               km(i1,i2)=km(i1,i2)-km_r/nip
182               km(i2,i1)=km(i2,i1)-km_r/nip
183               km(i2,i2)=km(i2,i2)+km_r/nip
184             END IF
185           END DO elements_2_r
```

```
186       END DO int_pts_1
187       CALL fsparv(kv,km,g,kdiag)
188     END DO elements_2
189     loads=zero
190     READ(10,*)loaded_nodes,(k,loads(nf(:,k)),i=1,loaded_nodes)
191     loads=loads/thick
192     READ(10,*)fixed_freedoms
193     IF(fixed_freedoms/=0)THEN
194       ALLOCATE(node(fixed_freedoms),sense(fixed_freedoms),                    &
195         value(fixed_freedoms),no(fixed_freedoms))
196       READ(10,*)(node(i),sense(i),value(i),i=1,fixed_freedoms)
197       DO i=1,fixed_freedoms
198         no(i)=nf(sense(i),node(i))
199       END DO
200       kv(kdiag(no))=kv(kdiag(no))+penalty
201       loads(no)=kv(kdiag(no))*value
202     END IF
203     CALL mesh_ensi(argv,nlen,g_coord,g_num,element,etype,nf,loads(1:),      &
204                    step,npri,dtim,solid)
205     element='bar'
206     CALL mesh_ensi(argv,nlen,g_coord,g_num_r,element,etype_r,nf,loads(1:),  &
207                    step,npri,dtim,solid)
208     element='quadrilateral'
209   !-----------------------equation solution-------------------------------
210     CALL sparin(kv,kdiag)
211     CALL spabac(kv,loads,kdiag)
212     loads(0)=zero
213     totd=totd+loads
214     WRITE(11,'(/A)')"  Node    x-disp       y-disp"
215     DO k=1,nn
216       WRITE(11,'(I5,2E12.4)')k,totd(nf(:,k))
217     END DO
218     CALL dismsh_ensi(argv,nlen,step,nf,totd(1:))
219   !-----------------------recover stresses at nip integrating points--------
220     nip=1
221     DEALLOCATE(points,weights)
222     ALLOCATE(points(nip,ndim),weights(nip))
223     CALL sample(element,points,weights)
224     WRITE(11,'(/A,I2,A)')" The integr. point (nip=",nip,") stresses are:"
225     WRITE(11,'(A,A)')" Element x-coord     y-coord",                        &
226     "     sig_x        sig_y        tau_xy"
227     elements_3: DO iel=1,nels
228       IF(etype(iel)==1)THEN
229         prop(1,iel)=Et(iel)
230         prop(2,iel)=vt(iel)
231       ELSE
232         prop(1,iel)=prop_s(1)
233         prop(2,iel)=prop_s(2)
234       END IF
235       CALL fmdsig(dee,prop(1,iel),prop(2,iel))
236       IF(crack(iel)==1)THEN
237         dee(3,3)=dee(3,3)*beta
238       END IF
239       IF(crack(iel)==2)THEN
240         dee(1,1)=dee(1,1)*0.0001
241         dee(1,2)=zero
242         dee(2,1)=zero
243         dee(3,3)=dee(3,3)*beta
244       END IF
245       IF(crack(iel)==3)THEN
246         dee(2,2)=dee(2,2)*0.0001
247         dee(1,2)=zero
248         dee(2,1)=zero
249         dee(3,3)=dee(3,3)*beta
250       END IF
251       IF(crack(iel)==4)THEN
252         dee(1,1)=dee(1,1)*0.0001
253         dee(2,2)=dee(2,2)*0.0001
```

```
          dee(1,2)=zero
          dee(2,1)=zero
          dee(3,3)=dee(3,3)*beta
        END IF
        num=g_num(:,iel)
        coord=TRANSPOSE(g_coord(:,num))
        g=g_g(:,iel)
        eld=loads(g)
        int_pts_2: DO i=1,nip
          CALL shape_fun(fun,points,i)
          CALL shape_der(der,points,i)
          gc=MATMUL(fun,coord)
          jac=MATMUL(der,coord)
          CALL invert(jac)
          deriv=MATMUL(jac,der)
          CALL beemat(bee,deriv)
          eps=MATMUL(bee,eld)
          sigma=MATMUL(dee,eps)
          etensor(:,1,iel)=etensor(:,1,iel)+eps
          tensor(:,1,iel)=tensor(:,1,iel)+sigma
          IF(etype(iel)==1)THEN
            Inv1=tensor(1,1,iel)+tensor(2,1,iel)
            sigma0=abs(Inv1/3.0)
            Inv2=tensor(1,1,iel)*tensor(2,1,iel)-tensor(3,1,iel)**2.0
            tau0=(2.0*sigma0**2.0-(2.0/3.0)*Inv2)**(1.0/2.0)
            teta=acos(1.0/(sqrt(2.0)*tau0)*sigma0)
            IF(teta<0.0.OR.teta>1.57080)WRITE(*,*)"Strani valori di teta"
            IF(teta>1.04719)teta=1.04719
            tau0c=0.944*fc*(sigma0/fc+0.075)**0.724+0.025*fc
            tau0e=0.633*fc*(sigma0/fc+0.075)**0.857+0.025*fc
            IF(tau0e<0.5*tau0c)tau0e=0.5*tau0c
            tau0u=(2.0*tau0c*(tau0c**2.0-tau0e**2.0)*cos(teta)+tau0c*(2.0     &
            *tau0e-tau0c)*(4.0*(tau0c**2.0-tau0e**2.0)*cos(teta)**2.0+5.0     &
            *tau0e**2.0-4.0*tau0c*tau0e)**0.5)/(4.0*(tau0c**2.0-tau0e**2.0)   &
            *cos(teta)**2.0+(tau0c-2.0*tau0e)**2.0)
            IF(tau0>tau0u)crush=iel
            Kt=Ke/(1.0+b*A*(sigma0/fc)**(b-1.0))
            Gt=Ge/(1.0+d*C*(tau0/fc)**(d-1.0))
            Et(iel)=9.0*Kt*Gt/(3.0*Kt+Gt)
            vt(iel)=(3.0*Kt-2.0*Gt)/(6.0*Kt+2.0*Gt)
            sigma1=0.5*(tensor(1,1,iel)+tensor(2,1,iel))+sqrt((0.5             &
            *(tensor(1,1,iel)-tensor(2,1,iel)))**2.0+tensor(3,1,iel)**2.0)
            IF(crack(iel)==0.AND.sigma1>fct/2.0)crack(iel)=1
            IF(crack(iel)<=1.AND.tensor(1,1,iel)>fct/10.0)crack(iel)=2
            IF(crack(iel)<=1.AND.tensor(2,1,iel)>fct/10.0)crack(iel)=3
            IF(crack(iel)<=1.AND.tensor(1,1,iel)>fct/10.0                      &
            .AND.tensor(2,1,iel)>fct/10.0)crack(iel)=4
            IF(crack(iel)==2.AND.tensor(2,1,iel)>fct/10.0)crack(iel)=4
            IF(crack(iel)==3.AND.tensor(1,1,iel)>fct/10.0)crack(iel)=4
          END IF
          WRITE(11,'(I5,6E12.4)')iel,gc,tensor(:,1,iel)
        END DO int_pts_2
      END DO elements_3
      nlen=nlen+3
      iel=nf(1,1)
      iel_r=nf(2,1)
      nf(1,1)=nels
      nf(2,1)=nels_r
      argv(nlen-2:nlen)='sxx'
      CALL dismsh_ensi(argv,nlen,step,nf,tensor(1,1,:))
      argv(nlen-2:nlen)='syy'
      CALL dismsh_ensi(argv,nlen,step,nf,tensor(2,1,:))
      argv(nlen-2:nlen)='sxy'
      CALL dismsh_ensi(argv,nlen,step,nf,tensor(3,1,:))
      argv(nlen-2:nlen)='exx'
      CALL dismsh_ensi(argv,nlen,step,nf,etensor(1,1,:))
      argv(nlen-2:nlen)='eyy'
      CALL dismsh_ensi(argv,nlen,step,nf,etensor(2,1,:))
```

```
322       argv(nlen-2:nlen)='exy'
323       CALL dismsh_ensi(argv,nlen,step,nf,etensor(3,1,:))
324       element='cracks'
325       argv(nlen-2:nlen)='cks'
326       CALL mesh_ensi(argv,nlen,g_coord,g_num,element,crack,nf,loads(1:),      &
327                      step,npri,dtim,solid)
328       nf(1,1)=iel
329       nf(2,1)=iel_r
330       argv(nlen-2:nlen)='   '
331       nlen=nlen-3
332       elements_3_r: DO iel_r=1,nels_r
333         num_r=g_num_r(:,iel_r)
334         coord_r=TRANSPOSE(g_coord(:,num_r))
335         ell0=sqrt((coord_r(2,1)-coord_r(1,1))**2.0+(coord_r(2,2)              &
336         -coord_r(1,2))**2.0)
337         CALL num_to_g(num_r,nf,g_r)
338         eld_r=loads(g_r)
339         coord_r(1,1)=coord_r(1,1)+eld_r(1)
340         coord_r(1,2)=coord_r(1,2)+eld_r(2)
341         coord_r(2,1)=coord_r(2,1)+eld_r(3)
342         coord_r(2,2)=coord_r(2,2)+eld_r(4)
343         ell=sqrt((coord_r(2,1)-coord_r(1,1))**2.0+(coord_r(2,2)-coord_r(1,2))&
344         **2.0)
345         eps_r=(ell-ell0)/ell0
346         sigma_r=prop_r(1,iel_r)*eps_r
347         etensor_r(iel_r)=etensor_r(iel_r)+eps_r
348         tensor_r(iel_r)=tensor_r(iel_r)+sigma_r
349         IF(yield(iel_r)==0.AND.abs(tensor_r(iel_r))>=prop_r(2,iel_r))THEN
350           yield(iel_r)=1
351           prop_r(1,iel_r)=prop_r(3,iel_r)
352         END IF
353         IF(abs(tensor_r(iel_r))>=prop_r(4,iel_r))yield_fail=iel_r
354       END DO elements_3_r
355       nlen=nlen+2
356       iel=nf(1,1)
357       iel_r=nf(2,1)
358       nf(1,1)=nels
359       nf(2,1)=nels_r
360       argv(nlen-1:nlen)='er'
361       CALL dismsh_ensi(argv,nlen,step,nf,etensor_r(:))
362       argv(nlen-1:nlen)='sr'
363       CALL dismsh_ensi(argv,nlen,step,nf,tensor_r(:))
364       element='yield'
365       argv(nlen-1:nlen)='yd'
366       CALL mesh_ensi(argv,nlen,g_coord,g_num,element,yield,nf,loads(1:),      &
367                      step,npri,dtim,solid)
368       element='quadrilateral'
369       nf(1,1)=iel
370       nf(2,1)=iel_r
371       argv(nlen-1:nlen)='  '
372       nlen=nlen-2
373       IF(crush>0)WRITE(*,*)crush,"c"
374       IF(yield_fail>0)WRITE(*,*)yield_fail,"r"
375       IF(crush>0.OR.yield_fail>0)STOP
376     END DO steps
377   STOP
378   END PROGRAM p51_kot
```

Commento brevemente alcune parti di codice. All'inizio del programma (righe 1 - 52) vengono fondamentalmente aggiunte tutta una serie di nuove variabili. Di alcune già presenti ne viene fissato il valore, non essendoci più la possibilità di scegliere, ad esempio, la forma dell'elemento o il numero di punti di integrazione. Diverse nuove variabili finiscono in *_r*: esprimono le stesse grandezze delle rispettive variabili senza il *_r* finale, ma si riferiscono alle barre di acciaio.

Dopodiché vengono calcolati (righe 53 – 66) alcuni parametri del modello di Kotsovos. Il programma *p51* non prevedeva la possibilità di inserire lo spessore della struttura, nel senso che in ogni analisi veniva di fatto utilizzato lo spessore unitario. La variabile *thick* (righe 72 – 73) e il modo in cui verrà utilizzata anche nel seguito, dà appunto la possibilità di inserire lo spessore. Il breve ciclo *elements_1_r* (righe 89 – 91) serve invece a memorizzare le proprietà delle barre di acciaio nella matrice *prop_r*.

Immediatamente prima di partire con gli step di carico (righe 100 – 111), alcuni valori vengono inizializzati a zero, mentre la rigidezza e il coefficiente di Poisson del calcestruzzo tangenziali vengono posti pari ai valori iniziali. Il ciclo *DO steps* (riga 112) gestisce appunto gli step di carico.

Essendo l'analisi diventata incrementale, ho dovuto inserire le righe 117 – 119 all'inizio di ogni step di carico. Servono a stabilire il contrario di quello che stabiliscono le righe 220 – 222 e cioè che la matrice di rigidezza degli elementi bidimensionali va calcolata usando quattro punti di integrazione (e non uno solo, come per il calcolo delle tensioni e deformazioni alla fine di ogni step di carico).

Le righe 122 – 128 servono semplicemente a "vedere" se l'elemento bidimensionale in questione è di calcestruzzo o lineare elastico. La diversa subroutine alla riga 129 (*fmdsig* invece di *deemat*) è perché prima (nel programma *p51*) venivano eseguite, di default, analisi in stato di deformazione piana, mentre adesso sempre analisi in stato di tensione piana. I comandi *IF* tra le righe 130 e 151 riguardano gli elementi che, nello step di carico precedente, si sono fessurati o hanno cambiato il loro stato di fessurazione. Viene modificata la matrice costitutiva dell'elemento, a seconda dello stato di fessurazione dello stesso. Queste parti di codice si ripeteranno identiche alle righe 228 – 257.

Il ciclo *elements_2_r* (righe 165 – 185) inserisce gli elementi monodimensionali all'interno di quelli bidimensionali, andando a modificare la matrice di rigidezza di questi ultimi.

All'interno del comando *IF* tra le righe 274 e 303 viene eseguita la procedura con cui si arriva al calcolo dei valori di rigidezza e coefficiente di Poisson tangenziali del calcestruzzo non fessurato. Oltre a questo viene eseguito il controllo per vedere quali elementi si sono fessurati o hanno cambiato il loro stato di fessurazione: tale informazione viene registrata nella variabile *crack*.

La parte finale nelle righe 307 – 331 e 355 – 372 (e, precedentemente, nelle righe 203 – 208) serve a generare dei file che permetteranno la visualizzazione grafica dei dati di input e output. Il libro di Griffiths et al. (2014) prevede infatti che le analisi possano essere rappresentate con il software open source ParaView. Le subroutine fornite dal libro permettono però di visualizzare solo alcuni dati. Ad esempio, non ci sono subroutine che permettono di generare i file contenenti tensioni e deformazioni. Nella parte di programma in questione ho quindi usato le subroutine fornite dal libro anche per generare dei risultati aggiuntivi a me necessari. Questa parte di programma è stata scritta in un modo

che è da ritenersi provvisorio. Quest'uso delle subroutine del libro per generare risultati aggiuntivi è stato abbastanza forzato, seppur funzionante. Oppure, invece di aggiungere delle variabili che sarebbero state necessarie, ne ho utilizzate altre già presenti, in maniera poco formale.

Il ciclo *elements_3_r* (righe 332 – 354) serve innanzitutto a calcolare le deformazioni e le tensioni nelle barre di acciaio. Dopodiché controlla se ci sono barre snervate o rotte. Nel primo caso aggiorna il modulo di rigidezza della barra al tratto di deformazione plastica, nel secondo dà il via al termine dell'analisi, tramite la variabile *yield_fail*.

Le tre righe 373 – 375 alla fine del programma danno lo stop allo stesso, quando almeno un elemento ha raggiunto la rottura; segnalano anche qual è uno degli elementi bidimensionali e/o monodimensionali in questione.

Appendice

A2 *Program p51* di Griffiths et al. (2014)

Per favorire ulteriormente il confronto col programma *p51*, ne riporto il codice. Questa volta ho evidenziato le parti tolte o modificate.

```
PROGRAM p51
 !------------------------------------------------------------------------
 ! Program 5.1 Plane or axisymmetric strain analysis of an elastic solid
 !             using 3-, 6-, 10- or 15-node right-angled triangles or
 !             4-, 8- or 9-node rectangular quadrilaterals. Mesh numbered
 !             in x(r)- or y(z)- direction.
 !------------------------------------------------------------------------
  USE main
  USE geom
  IMPLICIT NONE
  INTEGER,PARAMETER::iwp=SELECTED_REAL_KIND(15)
  INTEGER::fixed_freedoms,i,iel,k,loaded_nodes,ndim=2,ndof,nels,neq,nip,    &
    nlen,nn,nod,nodof=2,nprops=2,np_types,nr,nst=3,nxe,nye
  REAL(iwp)::det,one=1.0_iwp,penalty=1.0e20_iwp,zero=0.0_iwp
  CHARACTER(LEN=15)::argv,element,dir,type_2d
 !----------------------dynamic arrays-------------------------------------
  INTEGER,ALLOCATABLE::etype(:),g(:),g_g(:,:),g_num(:,:),kdiag(:),nf(:,:),  &
    no(:),node(:),num(:),sense(:)
  REAL(iwp),ALLOCATABLE::bee(:,:),coord(:,:),dee(:,:),der(:,:),deriv(:,:),  &
    eld(:),fun(:),gc(:),g_coord(:,:),jac(:,:),km(:,:),kv(:),loads(:),       &
    points(:,:),prop(:,:),sigma(:),value(:),weights(:),x_coords(:),         &
    y_coords(:)
 !-----------------------input and initialisation--------------------------
  CALL getname(argv,nlen)
  OPEN(10,FILE=argv(1:nlen)//'.dat')
  OPEN(11,FILE=argv(1:nlen)//'.res')
  READ(10,*)type_2d,element,nod,dir,nxe,nye,nip,np_types
  CALL mesh_size(element,nod,nels,nn,nxe,nye)
  ndof=nod*nodof
  IF(type_2d=='axisymmetric')nst=4
  ALLOCATE(nf(nodof,nn),points(nip,ndim),g(ndof),g_coord(ndim,nn),fun(nod),&
    coord(nod,ndim),jac(ndim,ndim),g_num(nod,nels),der(ndim,nod),           &
    deriv(ndim,nod),bee(nst,ndof),km(ndof,ndof),eld(ndof),weights(nip),     &
    g_g(ndof,nels),prop(nprops,np_types),num(nod),x_coords(nxe+1),          &
    y_coords(nye+1),etype(nels),gc(ndim),dee(nst,nst),sigma(nst))
  READ(10,*)prop
  etype=1
  IF(np_types>1)read(10,*)etype
  READ(10,*)x_coords,y_coords
  nf=1
  READ(10,*)nr,(k,nf(:,k),i=1,nr)
  CALL formnf(nf)
  neq=MAXVAL(nf)
  ALLOCATE(loads(0:neq),kdiag(neq))
  kdiag=0
 !----------------------loop the elements to find global arrays sizes-----
  elements_1: DO iel=1,nels
    CALL geom_rect(element,iel,x_coords,y_coords,coord,num,dir)
    CALL num_to_g(num,nf,g)
    g_num(:,iel)=num
    g_coord(:,num)=TRANSPOSE(coord)
    g_g(:,iel)=g
    CALL fkdiag(kdiag,g)
```

```
  END DO elements_1
  CALL mesh(g_coord,g_num,argv,nlen,12)
  DO i=2,neq
    kdiag(i)=kdiag(i)+kdiag(i-1)
  END DO
  ALLOCATE(kv(kdiag(neq)))
  WRITE(11,'(2(A,I5))')                                                    &
    " There are",neq," equations and the skyline storage is",kdiag(neq)
 !----------------------element stiffness integration and assembly--------
  CALL sample(element,points,weights)
  kv=zero
  gc=one
  elements_2: DO iel=1,nels
    CALL deemat(dee,prop(1,etype(iel)),prop(2,etype(iel)))
    num=g_num(:,iel)
    coord=TRANSPOSE(g_coord(:,num))
    g=g_g(:,iel)
    km=zero
    int_pts_1: DO i=1,nip
      CALL shape_fun(fun,points,i)
      CALL shape_der(der,points,i)
      jac=MATMUL(der,coord)
      det=determinant(jac)
      CALL invert(jac)
      deriv=MATMUL(jac,der)
      CALL beemat(bee,deriv)
      IF(type_2d=='axisymmetric')THEN
        gc=MATMUL(fun,coord)
        bee(4,1:ndof-1:2)=fun(:)/gc(1)
      END IF
      km=km+MATMUL(MATMUL(TRANSPOSE(bee),dee),bee)*det*weights(i)*gc(1)
    END DO int_pts_1
    CALL fsparv(kv,km,g,kdiag)
  END DO elements_2
  loads=zero
  READ(10,*)loaded_nodes,(k,loads(nf(:,k)),i=1,loaded_nodes)
  READ(10,*)fixed_freedoms
  IF(fixed_freedoms/=0)THEN
    ALLOCATE(node(fixed_freedoms),sense(fixed_freedoms),                   &
      value(fixed_freedoms),no(fixed_freedoms))
    READ(10,*)(node(i),sense(i),value(i),i=1,fixed_freedoms)
    DO i=1,fixed_freedoms
      no(i)=nf(sense(i),node(i))
    END DO
    kv(kdiag(no))=kv(kdiag(no))+penalty
    loads(no)=kv(kdiag(no))*value
  END IF
 !----------------------equation solution---------------------------------
  CALL sparin(kv,kdiag)
  CALL spabac(kv,loads,kdiag)
  loads(0)=zero
  IF(type_2d=='axisymmetric')THEN
    WRITE(11,'(/A)')"  Node    r-disp       z-disp"
  ELSE
    WRITE(11,'(/A)')"  Node    x-disp       y-disp"
  END IF
  DO k=1,nn
    WRITE(11,'(I5,2E12.4)')k,loads(nf(:,k))
  END DO
 !----------------------recover stresses at nip integrating points--------
  nip=1
  DEALLOCATE(points,weights)
  ALLOCATE(points(nip,ndim),weights(nip))
  CALL sample(element,points,weights)
  WRITE(11,'(/A,I2,A)')" The integration point (nip=",nip,") stresses are:"
  IF(type_2d=='axisymmetric')THEN
    WRITE(11,'(A,A)')" Element r-coord     z-coord",                     &
    "     sig_r        sig_z        tau_rz       sig_t"
```

```
122    ELSE
123      WRITE(11,'(A,A)')" Element x-coord     y-coord",                    &
124      "     sig_x       sig_y       tau_xy"
125    END IF
126    elements_3: DO iel=1,nels
127      CALL deemat(dee,prop(1,etype(iel)),prop(2,etype(iel)))
128      num=g_num(:,iel)
129      coord=TRANSPOSE(g_coord(:,num))
130      g=g_g(:,iel)
131      eld=loads(g)
132      int_pts_2: DO i=1,nip
133        CALL shape_fun(fun,points,i)
134        CALL shape_der(der,points,i)
135        gc=MATMUL(fun,coord)
136        jac=MATMUL(der,coord)
137        CALL invert(jac)
138        deriv=MATMUL(jac,der)
139        CALL beemat(bee,deriv)
140        IF(type_2d=='axisymmetric')THEN
141          gc=MATMUL(fun,coord)
142          bee(4,1:ndof-1:2)=fun(:)/gc(1)
143        END IF
144        sigma=MATMUL(dee,MATMUL(bee,eld))
145        WRITE(11,'(I5,6E12.4)')iel,gc,sigma
146      END DO int_pts_2
147    END DO elements_3
148    CALL dismsh(loads,nf,0.05_iwp,g_coord,g_num,argv,nlen,13)
149    CALL vecmsh(loads,nf,0.05_iwp,0.1_iwp,g_coord,g_num,argv,nlen,14)
150  STOP
151  END PROGRAM p51
```

Come si può vedere, ci sono poche modifiche e non è stato tolto quasi nulla. Molte righe eliminate riguardavano la possibilità di eseguire analisi assialsimmetriche (il nuovo programma prevede il solo stato di tensione piana) e le due finali (148 e 149) generavano file per la visualizzazione dell'analisi con immagini *PostScript*: ho preferito le subroutine che riguardano la visualizzazione con ParaView.

Appendice

B Dati di input

Descrivo brevemente il file di testo con cui sono forniti i dati di input al programma; lo faccio con riferimento ad un esempio creato solo a questo scopo:

```
4 2 2
4
2
2 5  5 8  8 11  11 14
30.0 0.15
200000.0 0.3
200000.0 450.0 2000.0 550.0 12.6
195000.0 390.0 2000.0 600.0 25.1
1 1 1 1 1 1 2 2
2 2 1 1
0.0 10.0 20.0 30.0 40.0
0.0 -5.0 -10.0
5.0
3
13 0 1  14 0 0  15 0 1
1
2 2.0 0.0
0
1
2 2.0 0.0
0
1
2 2.0 0.0
0
1
2 2.0 0.0
0
```

Innanzi tutto, per quanto riguarda le unità di misura, dovrebbe essere possibile usare un qualsiasi sistema di unità di misura coerente. Tuttavia, ho sempre usato i MPa per le tensioni, i mm per le lunghezze e i N per le forze.

La prima riga indica il numero di elementi in x, il numero di elementi in y e il numero di materiali che verranno usati per modellare gli elementi bidimensionali. A proposito di quest'ultimo numero devo specificare che può assumere solo due valori: 1 e 2. Nel primo caso tutti gli elementi bidimensionali saranno di calcestruzzo; nel secondo caso alcuni saranno di calcestruzzo e altri di materiale lineare elastico.

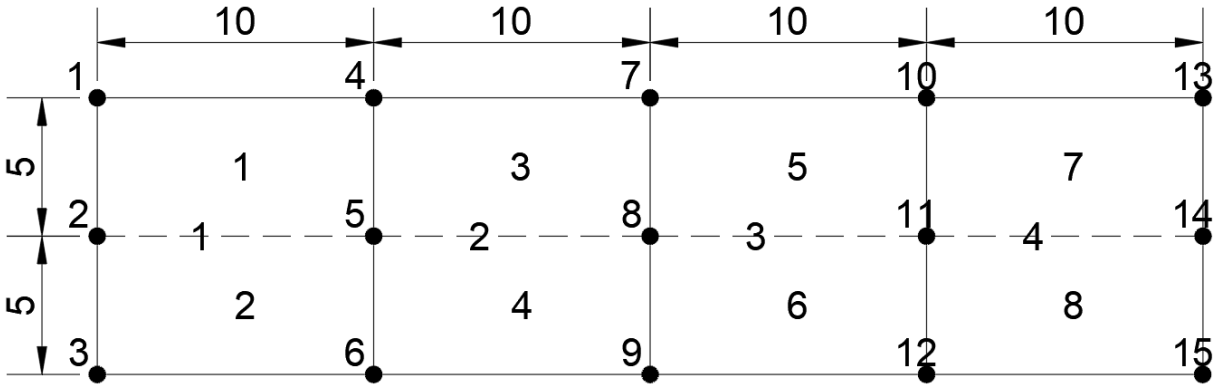

Figura 53 Mesh dell'esempio, con numerazione di nodi ed elementi. È visibile la suddivisione della mesh in quattro parti lungo x e in due parti lungo y. Le barre si trovano lungo le linee tratteggiate

La mesh della struttura viene creata da una subroutine che numera i nodi e gli elementi bidimensionali in un modo molto semplice: prima lungo y (nella direzione negativa dell'asse) e poi lungo x. Si ottiene la numerazione della mesh appena raffigurata.

La seconda riga e la terza riga indicano il numero di barre e il numero di tipologie diverse di barre (per tipologia di barre si intendono barre con diversa sezione e materiale). Ricordo che ho spesso preferito parlare di *barre* anche se a rigore bisognerebbe chiamarle *aste* quando ci si riferisce ai singoli elementi finiti.

La quarta riga è usata per indicare la posizione delle barre: la prima congiunge i nodi 2 e 5, la seconda i nodi 5 e 8 e così via. Le barre vengono numerate nell'ordine in cui sono inserite: la prima sarà quella tra i nodi 2 e 5, la seconda tra i nodi 5 e 8 e così via. Ricordo che le barre possono essere inserite soltanto in orizzontale e in verticale, e devono collegare due nodi dello stesso elemento bidimensionale. Non sarebbe quindi possibile, ad esempio, inserire una barra tra i nodi 2 e 4 o tra i nodi 2 e 8. C'è invece la possibilità di inserire più barre tra gli stessi due nodi (barre sovrapposte).

Poi ci sono i materiali. Il primo (l'unico ad essere sempre presente) è il calcestruzzo: ha una resistenza a compressione cilindrica di 30.0 MPa e uno shear retention factor di 0.15. Il secondo è lineare elastico con modulo di Young di 200000.0 MPa e coefficiente di Poisson di 0.3, il terzo e il quarto sono elastoplastici: sono indicati, nell'ordine, il modulo elastico, la resistenza a snervamento, il modulo di rigidezza nella fase plastica e la resistenza ultima. È poi presente un quinto valore che non si riferisce in effetti al materiale ma a una proprietà geometrica: la sezione delle barre. La prima tipologia di barre ha una sezione di 12.6 mm^2 (potrebbe essere, ad esempio, una barra ϕ4), la seconda una sezione di 25.1 mm^2. Per quanto riguarda il materiale lineare elastico esiste la possibilità di farlo diventare infinitamente rigido: inserendo un qualsiasi valore negativo come modulo elastico il programma sostituisce tale valore negativo con un valore (positivo) numericamente molto alto e quindi, di fatto,

infinitamente rigido (a quel punto non ha molta importanza il coefficiente di Poisson di tale materiale che, per semplicità, può essere posto pari a zero).

Le ulteriori due righe di interi attribuiscono i materiali agli elementi: la prima riga per i bidimensionali, la seconda per le barre. I primi sei elementi bidimensionali saranno quindi di calcestruzzo, gli ultimi due lineari elastici; le prime due barre saranno del secondo tipo (quelle con modulo elastico di 195000 MPa), le ultime due del primo (modulo elastico di 200000 MPa).

Dopodiché vengono inserite le coordinate delle colonne (distanza dall'asse y) e delle righe (distanza dall'asse x) dei nodi della mesh: la prima colonna e la prima riga si trovano sempre lungo gli assi (distanza pari a 0 mm), le successive colonne a distanza di 10, 20, 30 e 40 mm dall'asse y, mentre le successive righe a distanza di 5 e 10 mm dall'asse x (per un fatto numerico relativo alla subroutine che genera la mesh le coordinate delle righe devono essere inserite come valori negativi). È così che viene fuori la mesh della figura precedente. In questo caso ci sono colonne e righe equispaziate (tutti gli elementi hanno le stesse dimensioni) ma non necessariamente deve essere così.

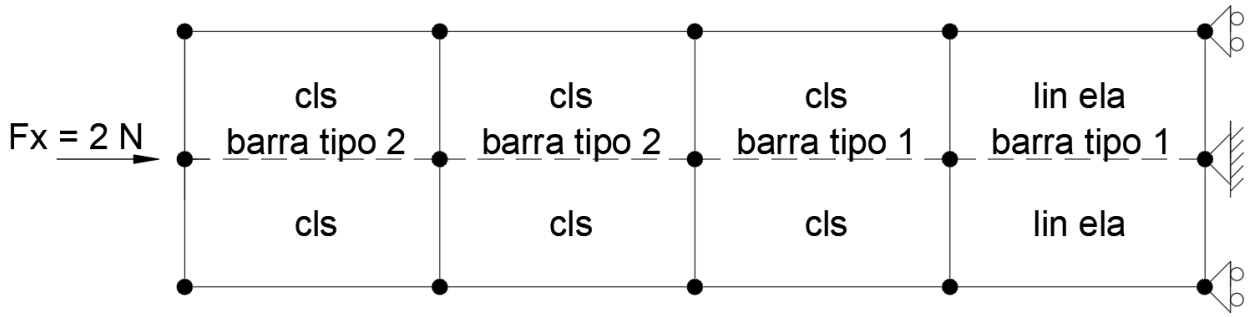

Figura 54 Materiali e condizioni al contorno dell'esempio (gli step di carico da 2 N sono quattro)

Dopo le coordinate delle colonne e delle righe di nodi c'è un valore reale che indica lo spessore della struttura.

Infine ci sono le condizioni al contorno. Il primo intero si riferisce al numero di nodi vincolati. In questo caso saranno tre: il 13, il 14 e il 15; il secondo è vincolato sia in x che in y (cerniera), gli altri due solo in x. Gli zeri indicano infatti la presenza del vincolo, gli uno l'assenza dello stesso.

Ci sono poi tre righe che si ripetono più volte (in questo caso quattro). La prima delle tre righe indica il numero di nodi caricati da forze esterne: in questo caso uno solo, il nodo 2, caricato in x con una forza di 2 N). L'ultima delle tre righe indica il numero di nodi soggetti a spostamento imposto, in questo caso nessun nodo. Queste tre righe si ripetono quattro volte perché, in questo caso, ci saranno quattro step di carico: in ognuno di essi verrà applicata la forza di 2 N al nodo 2 (in totale 8 N). Per comodità ho

imposto sempre la stessa forza a ogni step di carico, ma è possibile variare l'entità delle forze a ogni step, purchè rimangano tutte proporzionali a loro stesse.

Non tutti i dati devono essere sempre presenti nel file di testo, lo riassumo riportando e commentando il file stesso:

```
4 2 2
4
2 riga presente solo se il numero di barre è > 0
2 5  5 8  8 11  11 14 riga presente solo se il numero di barre è > 0
30.0 0.15
200000.0 0.3 riga presente solo se il numero di materiali degli elementi bidimensionali è > 1
200000.0 450.0 2000.0 550.0 12.6 riga presente solo se il numero di barre è > 0
195000.0 390.0 2000.0 600.0 25.1 riga presente solo se il numero di tipologie di barre è > 1
1 1 1 1 1 1 2 2 riga presente solo se il numero di materiali degli elementi bidimensionali è > 1
2 2 1 1 riga presente solo se il numero di tipologie di barre è > 1
0.0 10.0 20.0 30.0 40.0
0.0 -5.0 -10.0
5.0
3
13 0 1  14 0 0  15 0 1 riga presente solo se il numero precedente è > 0
1
2 2.0 0.0 riga presente solo se il numero precedente è > 0
0
1
2 2.0 0.0 riga presente solo se il numero precedente è > 0
0
1
2 2.0 0.0 riga presente solo se il numero precedente è > 0
0
1
2 2.0 0.0 riga presente solo se il numero precedente è > 0
0
```

Ricordo anche che all'interno del file possono essere presenti doppi spazi o tabulazioni per questioni di chiarezza o, ad esempio, per importare dati da excel; è possibile farne uso perché il software li ignora. È invece importante inserire i valori reali con almeno una cifra decimale dopo il punto (ad esempio, 2.0 e non 2).

Appendice

C Coordinate e tensioni ottaedrali

Introduco il sistema di riferimento ottaedrale (Kotsovos (2015)) utilizzato nell'elaborato. Prendiamo innanzi tutto in considerazione un sistema di coordinate cartesiane in cui gli assi esprimono le tensioni principali σ_1, σ_2 e σ_3 di un generico stato tensionale nello spazio. Le coordinate ottaedrali saranno un sistema di riferimento cilindrico avente per asse z la diagonale ottenuta per $\sigma_1 = \sigma_2 = \sigma_3$; r e Θ' saranno il raggio e la variabile rotazionale (quest'ultima è un angolo). Un modo comodo che useremo per vedere tali coordinate è dall'asse z stesso: il piano perpendicolare all'asse z è detto piano deviatorico.

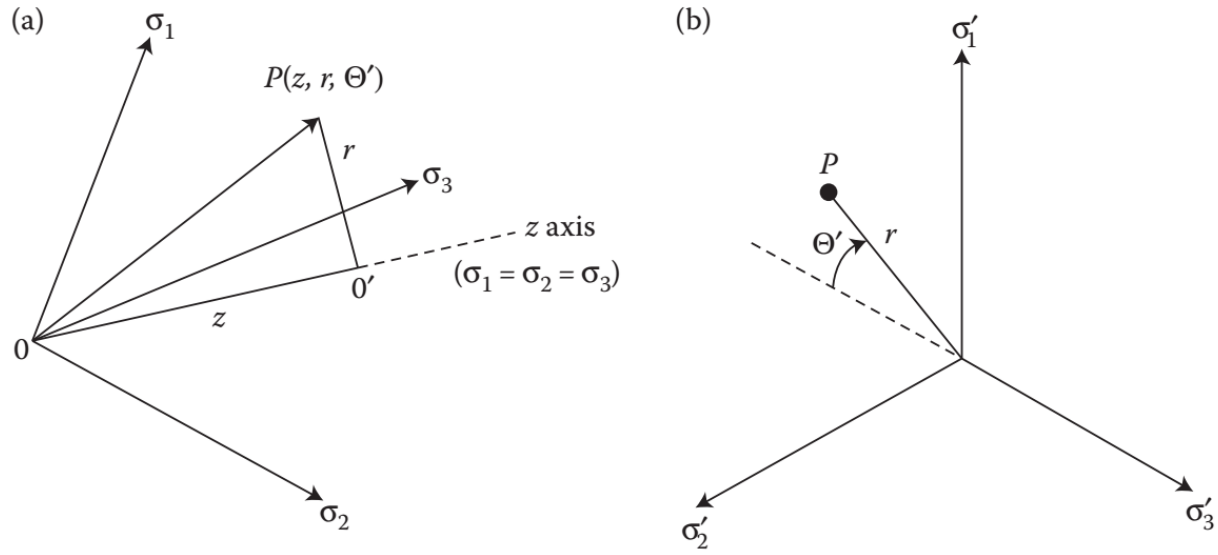

Figura 55 (a) Coordinate cartesiane (σ_1, σ_2 e σ_3) e ottaedriche (z, r e Θ'); (b) vista dall'asse z (vista del piano deviatorico)

Se volessimo passare dalle coordinate cartesiane a quelle ottaedrali potremmo usare le seguenti semplici formule:

$$z = (1/\sqrt{3})(\sigma_1 + \sigma_2 + \sigma_3)$$

$$r = (1/\sqrt{3})[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2]$$

$$cos\theta' = (1/\sqrt{6}r)(\sigma_1 + \sigma_2 - 2\sigma_3)$$

Le tensioni ottaedrali sono invece denominate σ_0, τ_0 e Θ, ed esprimono rispettivamente lo sforzo idrostatico, quello deviatorico e la variabile rotazionale (di nuovo, si tratta di un angolo). Sono anch'esse legate alle coordinate cartesiane, e quindi a quelle ottaedrali tramite:

$$\sigma_0 = (1/3)(\sigma_1 + \sigma_2 + \sigma_3) = (1/\sqrt{3})z$$

$$\tau_0 = (1/3)[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2]^{1/2} = (1/\sqrt{3})r$$

$$cos\theta = (1/\sqrt{2}\tau_0)(\sigma_0 - \sigma_3) = cos\theta'$$

Bibliografia

Abunama M. A. I., *Seismic Capacity Assessment of RC Buildings by Using Pushover Analysis*, Noor Publishing, 2017

ADINA (computer software), www.adina.com, ADINA R & D, Inc., Watertown (Massachusetts, US), 2020

Aicap, *Progettazione sismica di edifici in calcestruzzo armato*, Pubblicemento, 2008

ANSYS (computer software), www.ansys.com, ANSYS, Inc., Canonsburg (Pennsylvania, US), 2020

Bourne M., "Euler's Method - a numerical solution for Differential Equations", www.intmath.com/differential-equations, paragrafo n. 11, Apr. 2018, consultato in Nov. 2020

Bresler B., Scordelis A., "Shear strength of reinforced concrete beams", *Journal of the ACI*, v. 60, n. 1, 51–74, Gen. 1963

Carpinteri A., *Scienza delle costruzioni 1*, Pitagora, 1995

CEN, *Eurocode 2: Design of concrete structures - Part 1-1: General rules and rules for buildings*, Brussels, CEN, 2004

CEN, *Eurocode 8: Design of structures for earthquake resistance*, Brussels, CEN, 2004

Červenka J., Červenka V., Jendele L., *ATENA Program Documentation Part 1 Theory*, Prague, Červenka Consulting, Ott. 2013

Crisfield M. A., De Borst R., Remmers J. J. C., Verhoosel C. V., *Non-linear Finite Element Analysis of Solids and Structures*, Second Edition, Wiley, 2012

Dawkins P., "Euler's Method", tutorial.math.lamar.edu, Section 2-9, Mar. 2018, consultato in Giu. 2020

DIANA FEA (computer software), www.dianafea.com, DIANA FEA BV, Delft (Netherlands), 2020

Fib, *Practitioners' guide to finite element modelling of reinforced concrete structures*, Bulletin 45, Stuttgart, Fib, Giu. 2008

FORTRAN (linguaggio di programmazione), www.fortran-lang.org, John Backus, IBM, 2018

Guo Z., Nechvatal D., Zhou Y., "Evaluation of the Multiaxial Strength of Concrete Tested at Technische Universitat Munchen", *DAfStb*, v. 447, 65-106, Colonia, Beuth Verlag Berlin, 1995

Manie J., Kikstra W. P. (a cura di), *Diana User's Manual Element Library*, Release 10.1, DIANA FEA, Feb. 2017

Ghaboussi J., Wu X. S., *Numerical Methods in Computational Mechanics*, CRC Press, 2016

Griffiths D. V., Margetts L., Smith I. M., *Programming the finite element method*, 5 ed., Wiley, 2014

Hognestad E., *A study of combined bending and axial load in reinforced concrete members*, Illinois, University of illinois, v. 49, n. 22, Nov. 1951

Kotsovos M. D., *Finite-element modelling of structural concrete*, CRC Press, Apr. 2015

Kotsovos M. D., Lefas I. D., "Strength and deformation characteristics of reinforced concrete walls under load reversals", *ACI structural journal*, v. 87, n. 6, 716-726, Nov. – Dic. 1990

Maekawa K., Okamura H., Pimanmas A., *Non-Linear Mechanics of Reinforced Concrete*, CRC Press, Dic. 2019

Maier J., Thurlimann B., *Bruchversuche an Stahlbetonscheiben*, Zurich, Institut fur Baustatik und Konstruktion ETH, n. 8003-1, Gen. 1985

Massonet C., Save M., *Calcolo plastico a rottura delle costruzioni*, Maggioli Editore, 2008

"Metodo di Eulero", *Wikipedia, l'enciclopedia libera*, https://it.wikipedia.org/wiki/Metodo_di_Eulero, consultato in Ott. 2020

OpenSees (computer software), opensees.berkeley.edu, National Science Foundation, Pacific Earthquake Engineering Center (PEER), 2020

"Open Source", *Wikipedia, l'enciclopedia libera*, https://it.wikipedia.org/wiki/Open_Source, consultato in Nov. 2020

Ottosen N. S., Ristinmaa M., *The Mechanics of Constitutive Modeling*, Elsevier, 2005

Paraview (computer software), www.paraview.org, Sandia National Laboratories, Kitware Inc, Los Alamos National Laboratory, 2020

Parducci A., *Fondamenti di ingegneria sismica in 80 lezioni*, Liguori, 2016

Wang Z., Zhang Y., "Seismic behavior of reinforced concrete shear walls subjected to high axial loading", *ACI structural journal*, v. 97, n. 5, 739-750, Set. – Ott. 2000

William K. J., Warnke E. P., "Constitutive model for the triaxial behaviour of concrete", Bergamo, Seminar on concrete structures subjected to triaxial stresses, Paper III-1, ISMES, 1974