POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Matematica

Tesi di Laurea Magistrale

Analisi di un modello di elastoplasticità con l'applicazione dell'algoritmo di return mapping e del Metodo degli Elementi Virtuali



Relatore: Prof. Stefano Berrone Candidato: Francesca Marcon

ottobre 2018

Indice

Abstract			5
Introduzione		7	
1	Il n	nodello elastoplastico	9
	1.1	La superficie di snervamento	11
	1.2	Studio della deformazione plastica e della legge di flusso	12
	1.3	L'algoritmo iterativo	14
		1.3.1 L'algoritmo di return mapping	16
		1.3.2 Conclusione dell'algoritmo	24
2	Il n	netodo degli Elementi Virtuali	27
	2.1	Metodo in 2D	28
	2.2	Metodo in 3D	32
	2.3	Applicazione ai problemi di elasticità	36
	2.4	Applicazione ai problemi di elastoplasticità	38
3	Cor	nclusioni e risultati	41
	3.1	Test di confronto	44
	3.2	Analisi del programma di confronto	49
A	Coc	lice dell'algoritmo 'return mapping'	53

INDICE

Abstract

L'obbiettivo principale di questo lavoro è lo studio e l'analisi del comportamento elastoplastico dei materiali, con particolare riguardo al campo delle applicazioni in ambito geofisico. Una delle difficoltà iniziali, oltre alla scelta di una mesh appropriata che possa essere ben adattata alla geometria complessa del problema, è la scelta del modello più adatto per la definizione della superficie di snervamento. Nel contesto qui trattato viene utilizzato il modello di Mohr-Coulomb, che verrà descritto nel corso dell'elaborato, andando a evidenziarne i relativi punti di forza e debolezza. Un aspetto importante da sottolineare, che ha comportato alcune difficoltà, è il fatto che l'introduzione del comportamento plastico nel modello renda l'equazione costitutiva non lineare. Per alcuni modelli di plasticità, la non linearità dell'equazione del modello ha avuto un forte impatto sul metodo utilizzato nelle simulazioni e, al fine di ottenere delle approssimazioni numeriche, sono state necessarie alcune scelte che verranno discusse nel dettaglio. Nel secondo capitolo della tesi viene presentato il Metodo degli Elementi Virtuali (VEM), metodo numerico scelto per discretizzare le equazioni coinvolte e affrontare i problemi di generazione della mesh causati dalle geometrie complesse a cui si applica questo modello. Nell'ultimo capitolo di questa tesi, si vogliono presentare alcuni risultati ottenuti dall'applicazione del modello a particolari casi presi in esame, al fine di evidenziare gli aspetti positivi e negativi del modello proposto, sia dal punto di vista matematico che da quello computazionale. Infine è stato fatto un confronto tra un'implementazione VEM del metodo numerico descritto e un codice open source Matlab tratto da un articolo di ricerca su questo argomento.

The main purpose of this work is the investigation of the elastoplastic behaviour of the materials, in the framework of geophysical applications. Together with the generation of a proper mesh suitable for the complex geometry of the problem, the investigation of geophysical deformations with elasto-plastic model involves the difficulty of choosing the most appropriate vield criterion. In this context the Mohr-Coulomb model is widely applied, it will be described in detail in this work. Another important aspect that has to be underlined is that the introduction of the plasticity involves a strong nonlinearity in the equations modeling the problem. For some plasticity model, this non-linearity strongly impacts the numerical method used to perform simulations and some choices are required to get numerical approximations. In the second chapter some insight is provided about the method chosen in order to tackle problem, the Virtual Element Method (VEM). In the last chapter of this thesis, some numerical results are reported and a VEM implementation of the numerical method for the computation of the plastic deformation is compared with a research matlab open source code.

Introduzione

Questo lavoro poggia le sue fondamenta sulla studio di un modello che sia in grado di rappresentare il comportamento elastoplastico di un materiale in ambito geofisico. A partire dalla conoscenza del comportamento elastico e volendo introdurre la componente plastica nello studio, sono state riscontrate alcune difficoltà inerenti alla scelta del particolare modello applicato. Infatti, nel momento in cui si introduce la descrizione del comportamento plastico è necessario definire alcuni vincoli al sistema, tra cui la definizione di una superficie di snervamento. Qui viene descritto e trattato il modello di Mohr-Coulomb, il quale si basa sull'ipotesi che la plasticità del materiale sia governato dalle forze di attrito radente tra le particelle. Il modello di Mohr-Coulomb descrive la superficie come una piramide esagonale irregolare nello spazio delle tensioni principali, questa caratterizzazione geometrica è fondamentale nell'analisi del modello perché viene coinvolta in modo diretto all'interno dell'algoritmo qui utilizzato.

L'idea generale, che verrà descritta in modo approfondito nel primo capitolo, è quella di ricavare, a seguito di una discretizazzione temporale e spaziale del problema, il valore del tensore degli sforzi per ogni elemento in esame, applicando un algoritmo noto come *return mapping*. Questo algoritmo consiste nel riproiettare sul campo plastico il tensore degli sforzi che, in seguito all'applicazione di una deformazione elastica, ha assunto valori esterni alla superficie di snervamento e quindi non accettabili. Il modello di Mohr Coulomb può creare delle difficoltà nel momento in cui la proiezione avvenga su uno degli spigoli o sull'apice della piramide, queste difficoltà potrebbero riscontrarsi non tanto dal punto di vista di formulazione teorica del modello quanto più in seguito alla simulazione numerica. Nella trattazione sono state considerate eventuali modifiche per rendere innanzitutto più semplice la componente computazionale del modello, inoltre sono state inserite alcune idee per ovviare alle difficoltà che la geometria della superficie può causare. Nel secondo capitolo viene descritto il Metodo degli Elementi Virtuali (VEM), questo metodo numerico è molto recente e ha delle potenzialità che vengono qui analizzate dal punto di vista teorico. Questi punti di forza hanno fatto sì che fosse ritenuto un metodo adatto per la risoluzione del modello trattato. Per rendere più semplice la descrizione di questo modello, si è scelto di trattarlo prima dal punto di vista più generale possibile distinguendo i casi di problemi in due o tre dimensioni. Successivamente è stato descritto il modello applicato prima ad un problema di elasticità ed in seguito esteso al caso elastoplastico. Nel capitolo conclusivo si sono voluti riconsiderare alcuni aspetti del modello in esame, andando a presentare brevemente alcune possibili modifiche nell'ottica di un lavoro futuro. In seguito si è voluto riportare un confronto su un particolare problema preso in esame tra il programma qui descritto e un particolare codice *Matlab* dato dalla fonte ([12]), commentando così i punti di forza e di debolezza del modello analizzato.

Capitolo 1

Il modello elastoplastico

L'argomento principale di questa tesi verte sulla ricerca di un modello matematico e della sua successiva implementazione, al fine di descrivere il comportamento elastoplastico di materiali come rocce e detriti. Per analizzare l'oggetto in esame, Ω , si considerano gli spostamenti, dati dalla funzione $\boldsymbol{u}: \Omega \to \mathbb{R}^3$. Questa descrive il movimento delle particelle del materiale e, in generale, è una funzione dipendente dal tempo t, il quale, come vedremo in seguito, viene discretizzato, analizzando così istante per istante il comportamento del sistema. Entrano in gioco anche altri due fattori importanti in questo modello: il tensore delle deformazioni ε e il tensore degli sforzi σ . Questi sono tensori del secondo ordine, doppio covarianti, e si possono definire come:

$$\varepsilon: \mathcal{V} \times \mathcal{V} \longrightarrow \mathbb{R}$$
$$\sigma: \mathcal{V} \times \mathcal{V} \longrightarrow \mathbb{R}$$

dove \mathcal{V} rappresenta un generico spazio vettoriale reale.¹ La descrizione del modello matematico scelto in questo lavoro deve iniziare con alcune considerazioni preliminari: gli spostamenti sono considerati infinitesimali e, in presenza di comportamento puramente elastico, si suppone valida la legge di Hooke generalizzata:

(1.1)
$$\sigma = D\varepsilon^e$$

dove σ rappresenta il tensore degli sforzi, ε^e la componente elastica del tensore delle deformazioni e D il tensore di elasticità o *elastic constitutive matrix*.

¹Nel particolare campo di applicazione di cui tratteremo in questo lavoro, come $\mathcal V$ sarà considerato lo spazio $\mathbb R,\mathbb R^2$ o $\mathbb R^3$

L'introduzione nel modello del comportamento plastico comporta una non linearità nella relazione fra tensore degli sforzi e quello delle deformazioni. Con *comportamento plastico* di un materiale si intende che, dopo essere stato soggetto a forze di carico, possa presentare deformazioni permanenti, dette per questo motivo *plastiche*.

Preliminarmente consideriamo una rappresentazione grafica del comportamento elastoplastico, analizzando la Figura (1.1) in cui viene presentato il legame tra $\sigma \in \varepsilon$ nel caso specifico di un stress test uniassiale.



Figura 1.1: Stress test uniassiale. Il tratto \overline{OP} rappresenta il comportamento puramente elastico del materiale, il tratto \overline{PQ} rappresenta il comportamento plastico e l'ultimo tratto $\overline{QO'}$ la fase in cui la forza di carico si annulla e il corpo torna in una posizione in cui il tensore delle deformazioni è diverso da quello iniziale, poiché è incrementato di un valore ε^p .

Dal grafico si evince che per valori di stress inferiori ad una certa soglia σ_0 il comportamento del materiale è puramente elastico: σ dipende linearmente da ε . Superata la soglia σ_0 cambia il grafico e la relazione tra i due tensori potrebbe essere, a priori, molto complessa. Ciò che viene fatto nello studio di un modello di questo tipo è approssimare questa relazione. Uno dei modi più comuni è caratterizzare la soglia come il passaggio fra un comportamento puramente elastico e uno puramente plastico. Osservando il grafico in Figura (1.1),il comportamento del materia oltre la soglia di stress σ_0 è rappresentato da una generica curva, nel caso in cui si volesse trattare il caso specifico di passaggio ad un comportamento puramente plastico, la rappresentazione avverrebbe per mezzo di una funzione costante. Dalla figura si può ancora notare come in caso di azzeramento dello stress di carico, lo stato del materiale non coincide con quello di partenza e la deformazione plastica è rappresentata della componente plastica del tensore delle deformazioni ε^p , la distanza tra i due stati non sottoposti a stress.

Al fine di trattare il comportamento plastico di un materiale bisogna definire alcuni concetti fondamentali: la superficie di snervamento, il limite al di sopra del quale il comportamento diventa plastico, la legge di flusso, relazione fra il tensore degli sforzi e quello delle deformazioni quando un materiale diventa plastico, e la condizione di consistenza che previene il fatto che il tensore delle deformazioni assuma valori eccedenti la superficie di snervamento.

1.1 La superficie di snervamento

Il modello che viene qui trattato è a tre dimensioni, quindi la generica funzione che rappresenta la superficie di snervamento è:

(1.2)
$$f(\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau yz, \tau_{zx}, \alpha_1, \cdots, \alpha_n) = 0$$

dove $\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{yz}, \tau_{zx}$ sono le sei componenti indipendenti del tensore degli sforzi e $\alpha_1, \dots, \alpha_n$ sono gli eventuali parametri del modello applicato. La convenzione adottata per definire la superficie di snervamento è tale per cui se f < 0 viene descritto l'interno della superficie, dunque un comportamento puramente elastico. Trattando in questo caso di plasticità perfetta sono ammesse solo le casistiche f < 0 e f = 0, interno e bordo della superficie, sul quale vi è comportamento plastico. Questo definisce il limite elastico, ovvero l'insieme dei massimi valori ammissibili del tensore degli sforzi. Nel modello trattato non viene considerato il fenomeno di *hardening*, ovvero l'evoluzione nel tempo della superficie di snervamento.

Dal momento che l'applicazione considerata in questo caso riguarda materiali come rocce, terreno e sedimenti, il modello scelto per la superficie di snervamento è quello di Mohr-Coulomb. In questo modello si assume che il fenomeno plastico, in caso di piccole deformazioni, sia essenzialmente il risultato di forze di attrito radente tra le particelle del materiale. Il modello è rappresentato dalla seguente espressione, dipendente dal tensore degli sforzi:

(1.3)
$$f(\sigma, \alpha) = \sigma_{max} - \sigma_{min} + (\sigma_{max} + \sigma_{min}) \sin \phi - 2c \cos \phi$$

dove $\alpha = (c, \phi)$ e c rappresenta la coesione e ϕ è l'angolo di attrito interno (frictional angle).

La superficie di snervamento $(f(\sigma, \alpha) = 0)$ in tre dimensioni si può rappresentare graficamente come una piramide a base esagonale, il cui apice rappresenta il limite di resistenza del materiale a forze di carico. Nello spazio delle tensioni principali è rappresentato dal punto $p = \cot \phi \mathbb{1}^2$.

Definiamo i sei piani che intersecati fra loro definiscono la piramide di Mohr-Coulomb nello spazio delle tensioni principali, ovvero l'autospazio generato dagli autovettori del tensore degli sforzi:

$$(1.4) \qquad f_1(\sigma,\alpha) = F_{13}(\sigma,\alpha) = \sigma_1 - \sigma_3 + (\sigma_1 + \sigma_3)\sin\phi - 2c\cos\phi$$
$$f_2(\sigma,\alpha) = F_{23}(\sigma,\alpha) = \sigma_2 - \sigma_3 + (\sigma_2 + \sigma_3)\sin\phi - 2c\cos\phi$$
$$f_3(\sigma,\alpha) = F_{21}(\sigma,\alpha) = \sigma_2 - \sigma_1 + (\sigma_2 + \sigma_1)\sin\phi - 2c\cos\phi$$
$$f_4(\sigma,\alpha) = F_{31}(\sigma,\alpha) = \sigma_3 - \sigma_1 + (\sigma_3 + \sigma_1)\sin\phi - 2c\cos\phi$$
$$f_5(\sigma,\alpha) = F_{32}(\sigma,\alpha) = \sigma_3 - \sigma_2 + (\sigma_3 + \sigma_2)\sin\phi - 2c\cos\phi$$
$$f_6(\sigma,\alpha) = F_{12}(\sigma,\alpha) = \sigma_1 - \sigma_2 + (\sigma_1 + \sigma_2)\sin\phi - 2c\cos\phi$$

1.2 Studio della deformazione plastica e della legge di flusso

Nel modello elastoplastico all'aumento delle forze di carico, il comportamento del materiale diviene, sempre più, puramente plastico, finché la capacità strutturale del materiale non sopporta più le forze e la struttura collassa. È questo il punto in cui la deformazione plastica assume valori infinitamente più elevati rispetto a quelli trattati normalmente da caso in esame. Questo è uno dei motivi per cui non vi è una relazione univoca tra il tensore degli sforzi e il tensore delle deformazioni totale, dunque si suppone di suddividere l'incremento del tensore delle deformazioni ε in due componenti, l'incremento di deformazione elastica e l'incremento di deformazione plastica:

(1.5)
$$d\varepsilon = d\varepsilon^e + d\varepsilon^p$$

dove, come già detto in precedenza, l'incremento dovuto alla deformazione elastica è soggetta alla legge di Hooke.

²Con $\mathbb{1}$ si intende il vettore unitario.

1.2. STUDIO DELLA DEFORMAZIONE PLASTICA E DELLA LEGGE DI FLUSSO13

D'altra parte si può supporre che l'incremento dovuto alla deformazione plastica derivi da un potenziale plastico g, secondo la seguente legge di flusso:

(1.6)
$$d\varepsilon^p = d\lambda \frac{\partial g}{\partial \sigma} = d\lambda \nabla g$$

dove $d\lambda$ è il coefficiente di proporzionalità.

La superficie data da g = 0 giace nello spazio delle tensioni principali, la sua espressione specifica varia a seconda del modello applicato. Trattando un modello di comportamento elastoplastico si distinguono le casistiche: modello *associativo* e *non associativo*, dove nel primo caso l'espressione della superficie di snervamento viene adottata come potenziale plastico. Nel particolare caso del modello di Mohr-Coulomb qui descritto, trattiamo il secondo caso in cui a priori non vi è una specifica espressione del potenziale. La scelta fatta è quella per cui si riprende l'espressione della superficie di snervamento e al posto dell'angolo ϕ si introduce un angolo ψ di molto inferiore al primo:

(1.7)
$$g(\sigma, \alpha) = \sigma_{max} - \sigma_{min} + (\sigma_{max} + \sigma_{min}) \sin \psi - 2c \cos \psi$$

dove l'angolo ψ viene chiamato angolo di dilatazione(*dilatancy angle*).

Dal momento che la rappresentazione nello spazio è anche in questo caso una piramide a base esagonale, quando sarà necessario il calcolo del gradiente del potenziale $\left(\frac{\partial g}{\partial \sigma}\right)$, si dovrà fare attenzione alle singolarità che il potenziale plastico presenta: gli spigoli e l'apice della piramide. Questi casi andranno trattati singolarmente con particolare attenzione.

La relazione (1.6) non viene utilizzata in campo applicativo, in quella che viene definita *elastoplasticità computazionale*, ma si procede analizzando ogni passo temporale k, dove avviene un incremento delle forze di carico f^k , le quali causano una variazione dello spostamento dell'oggetto in esame u^k , che a sua volta è causa di un incremento della deformazione totale ε^k . Può essere definita una relazione costitutiva tra l'incremento del tensore degli sforzi e l'incremento totale del tensore delle deformazioni:

(1.8)
$$d\sigma = D^{ep}d\varepsilon$$

dove D^{ep} è la matrice costitutiva elastoplastica, spesso chiamata consistent tangent matrix o consistent tangent modulus.

Applicando la relazione di suddivisione in componente elastica e plastica del tensore delle deformazioni (1.5), la legge di Hooke generalizzata (1.1) e la

relazione di consistenza, consistency condition, data da $\left(\frac{\partial f}{\partial \sigma}\right) d\sigma = 0$, si possono effettuare una serie di operazioni che portano alla scrittura di D^{ep} in funzione dell'espressione della superficie di snervamento f (1.2), del potenziale plastico g e del tensore di elasticità D:

(1.9)
$$D^{ep} = D - \frac{D\frac{\partial g}{\partial \sigma} \left(\frac{\partial f}{\partial \sigma}\right)^T D}{\left(\frac{\partial f}{\partial \sigma}\right)^T D\frac{\partial g}{\partial \sigma}}$$

Questa relazione ha il limite di poter essere applicata quando si conoscono sia il valore del tensore degli sforzi che l'incremento della deformazione totale, quando però, come nel nostro caso applicativo, questi valori non sono noti a priori ma devono essere ricercati, si applica un metodo iterativo per trovare D^{ep} e di conseguenza lo spostamento del materiale, u^k .

1.3 L'algoritmo iterativo

L'algoritmo qui descritto analizza uno specifico passo temporale k, in cui viene applicato un incremento di forza di carico f^k e, come detto in precedenza, questa forza causa uno spostamento dell'oggetto in esame Δu^k . L'obbiettivo di questa sezione è quello di descrivere come viene effettivamente calcolato Δu^k . L'analisi è svolta nell'ottica dell'applicazione di un metodo di discretizzazione spaziale, in particolare nel capitolo seguente si descriverà il Metodo degli Elementi Virtuali. Ha senso precisare che la discretizzazione spaziale dell'oggetto in esame Ω è essenzialmente composto da una griglia di nodi che definisce elementi poliedrici, su cui vengono calcolati gli specifici integrali richiesti dal modello e costruite le matrici risolventi. Dunque l'incognita di spostamento u si riferisce ad ogni nodo di quadratura e, a partire da uno specifico passo temporale k in cui si conosce il valore u^k , si cerca di ottenere il valore al passo temporale successivo, u^{k+1} :

$$(1.10) u^{k+1} = u^k + \Delta u^k$$

Per descrivere in modo completo l'intero algoritmo bisogna partire dall'equazione che regola globalmente il modello è:

1.3. L'ALGORITMO ITERATIVO

dove f è la forza applicata e K la matrice di rigidezza definita come:

(1.12)
$$K = \int_{\Omega} B^T D^{ep} B d\Omega$$

dove D^{ep} è la consistent tangent matrix e B è la matrice che lega un generale spostamento ad un incremento del tensore delle deformazioni: $\Delta \varepsilon = B\Delta u$. Come è già stato specificato, ad ogni passo temporale si applica la forza f^k e a questa corrisponde uno spostamento $u^k + \Delta u^k$. Al fine di trovare il corretto valore assunto dallo spostamento bisogna considerare la seguente equazione non lineare:

(1.13)
$$F(u^k) = q(u^k + \Delta u^k) - f^k = 0$$

questa rappresenta l'annullamento del residuo che si ottiene bilanciando la forza interna q, che si definisce come $q := \int_{\Omega} B^T \sigma d\Omega$, e la forza di carico appicata f (esterna).

Per risolvere l'equazione (1.13) e trovare l'incremento Δu^k si utilizza il metodo iterativo di Newton-Raphson:

(1.14)
$$J_F^{k,j} \cdot \delta u^j = F(u^k)$$

(1.15)
$$\Delta u^{k,j+1} = \Delta u^{k,j} + \delta u^j$$

dove con j si rappresenta una particolare iterazione del metodo, $J_F^{k,j}$ è lo Jacobiano della trasformazione F nell'istante temporale k, all'iterata j-esima dell'algoritmo di Newton. Inoltre indichiamo con δu^j l'incremento allo spostamento calcolato ad ogni passo dell'iterata di Newton, come si evince dalla relazione (1.15).

Lo Jacobiano della trasformazione, nel caso puramente elastico, coincide con il tensore di elasticità D (1.1); invece, nel caso elastoplastico in esame, il calcolo dello Jacobiano di F si ricava dalla relazione (1.11) e risulta coincidente con la definizione sopra riportata (1.12): $J_F = \int_{\Omega} B^T D^{ep} B d\Omega$, ricordando che vale la relazione costitutiva del modello elastoplastico (1.8).

Ad ogni iterata del procedimento di Newton, il calcolo di J_F viene eseguito elemento per elemento P, successivamente, sommando tutti i contributi, si ottiene la matrice definita sui nodi globali di Ω :

(1.16)
$$J_F = \sum_{P \in \Omega} \int_P B^T D^{ep} B dP$$

dove P indica il generico elemento. La relazione rappresenta l'assemblamento della matrice globale su tutti i nodi del sistema, la cui costruzione è necessaria per poter applicare il passo di Newton (1.14). In particolare, come si evince da (1.16), ad ogni passo del metodo iterativo si richiede il calcolo della matrice D^{ep} su ogni elemento della discretizzazione. È importante ricordare che la *consistent tangent matrix* D^{ep} dipende dal tensore di elasticità D, dal tensore degli sforzi σ , dall'espressione della superficie di snervamento e dal potenziale plastico g.

L'ottenimento del valore numerico della matrice D^{ep} su ogni nodo della griglia è il risultato ultimo di un algoritmo noto come return mapping, il quale prevede il calcolo del tensore degli sforzi σ alla fine del passo plastico grazie a due step parziali: il compimento di un passo lineare puramente elastico, attraverso l'applicazione del tensore D, che fornisce il calcolo della componente elastica ε^e del tensore delle deformazioni e un tensore degli sforzi σ^e e, nel caso in cui non fosse un valore ammissibile, poiché uscito uscito dalla superficie di snervamento, la proiezione di quest'ultimo sulla superficie stessa. Questo secondo caso capita se il comportamento del punto del materiale in esame è plastico. Nel nostro caso specifico trattiamo il modello di Mohr-Coulomb, dove, come vedremo nel prossimo paragrafo, la proiezione sulla superficie di snervamento comporta calcoli non complessi se trattiamo una delle facce della piramide. Questo perché la proiezione è diretta su una superficie lineare e il risultato sarà pressoché immediato. Invece saranno da trattare con maggiore cura le altre possibili casistiche: la proiezione su uno spigolo o sull'apice.

1.3.1 L'algoritmo di return mapping

Una precisazione necessaria è il fatto che questo algoritmo venga richiamato e applicato all'interno del modello solo nel caso in cui il tensore degli sforzi ottenuto in seguito al passo elastico σ^e sia esterno alla superficie di snervamento, ovvero:

(1.17)
$$\sigma^e : f(\sigma^e, \alpha) = \sigma^e_{max} - \sigma^e_{min} + (\sigma^e_{max} + \sigma^e_{min})\sin\phi - 2c\cos\phi > 0$$

Inoltre l'algoritmo che verrà presentato, descrive una serie di operazioni che avvengono nel sistema di riferimento principale³, solo in seguito al calcolo

³Ricordiamo che il sistema di riferimento principale è quello definito dagli autovettori del tensore delle deformazioni (i quali coincidono con quelli del tensore degli sforzi).

1.3. L'ALGORITMO ITERATIVO

del tensore elastoplastico D^{ep} , necessario per applicare (1.16), l'analisi viene riportata nel sistema di riferimento canonico. Si lavora di conseguenza con gli autovalori di σ^4 : $\sigma_1, \sigma_2, \sigma_3$ tali che $\sigma_1 \geq \sigma_2 \geq \sigma_3$. Questo ordinamento permette di definire in modo univoco il lato più vicino al valore in esame, ovvero quello su cui è corretto eseguire il primo tentativo di proiezione.

Il procedimento applicato comporta che si tenti di fare in ogni caso come primo passo la proiezione del tensore degli sforzi sul lato più vicino al valore in input, definito come:

(1.18)
$$F_{13}(\sigma_1, \sigma_3) = \sigma_1 - \sigma_3 + (\sigma_1 + \sigma_3) \sin \phi - 2c \cos \phi$$

Essendo il modello un caso di Mohr-Coulomb *non associativo* viene definito il potenziale plastico associato alla faccia (1.18):

(1.19)
$$G_{13}(\sigma_1, \sigma_3) = \sigma_1 - \sigma_3 + (\sigma_1 + \sigma_3)\sin\psi - 2c\cos\psi$$

In questo caso si parla di un solo *'meccanismo'* attivo e il tensore degli sforzi che si deve ottenere è:

(1.20)
$$\sigma^{new} = \sigma - d\lambda_1 \cdot (Dn_{G1})$$

dove σ è il valore in input, ottenuto in seguito al passo elastico, D è il tensore di elasticità (1.1), n_{G1} è la normale relativa al potenziale plastico, definito in (1.19), e $d\lambda_1$ è il moltiplicatore. La componente $d\lambda_1 \cdot (Dn_{G1})$ rappresenta la proiezione sul lato e per determinarla in modo univoco bisogna calcolare il valore del moltiplicatore. Dal momento che il risultato finale deve giacere sul lato superficie di snervamento, l'espressione (1.18) deve essere posta uguale a zero:

(1.21)
$$\sigma_1 - \sigma_3 + (\sigma_1 + \sigma_3)\sin\phi - 2c\cos\phi = 0$$

È sufficiente sostituire (1.20) nell'equazione precedente e ottenere:

$$(1.22) d\lambda_1 [(Dn_{G1})_1 - (Dn_{G1})_3 + ((Dn_{G1})_1 + (Dn_{G1})_3)\sin\phi] = \sigma_1 - \sigma_3 + (\sigma_1 + \sigma_3)\sin\phi - 2c\cos\phi$$

Da questa relazione è immediato ricavare il moltiplicatore $d\lambda_1$ ma per poter scrivere il risultato in modo esplicito è necessario fare un passo indietro e

⁴Per semplicità di notazione da qui in avanti indicheremo il tensore degli sforzi che è input dell'algoritmo con σ e non con σ^e

ricavare il tensore di elasticità e la normale alla faccia della piramide. Per scrivere il tensore di elasticità è necessario definire il modulo elastico di compressibilità K e il modulo elastico di rigidezza G:

(1.23)
$$K := \frac{E}{3(1-2\nu)}$$

(1.24)
$$G := \frac{E}{2(1+\nu)}$$

dove $E \in \nu$ sono i coefficienti di Lamé, rispettivamente il modulo di Young e il coefficiente di Poisson. Entrambi sono valori noti a priori del modello. La matrice di rappresentazione del tensore D, nel sistema di riferimento principale, si può scrivere come:

(1.25)
$$D = K + \frac{2}{3}G\begin{bmatrix} 2 & -1 & -1\\ -1 & 2 & -1\\ -1 & -1 & 2 \end{bmatrix}$$

Sapendo che il vettore normale alla faccia della piramide definita in (1.19) è

(1.26)
$$n_{G1} = \left(\frac{\partial G_{13}}{\partial \sigma_1}, \frac{\partial G_{13}}{\partial \sigma_2}, \frac{\partial G_{13}}{\partial \sigma_3}\right)^T = (\sin\psi + 1, 0, \sin\psi - 1)^T$$

otteniamo che (1.27)

$$(1.27)$$
$$D \cdot n_{G1} = 2\left(G + \left(K + \frac{G}{3}\right)\sin\psi, \left(K - \frac{2G}{3}\right)\sin\psi, \left(K + \frac{G}{3}\right)\sin\psi - G\right)^{T}$$

Riprendendo l'equazione (1.22), si ottiene l'espressione del moltiplicatore $d\lambda_1$:

(1.28)
$$d\lambda_1 = \frac{|F_{13}(\sigma)|}{4\left[G + \left(K + \frac{G}{3}\right)\sin\phi\sin\psi\right]}$$

A questo punto sostituendo il moltiplicatore nella relazione (1.20) si ottiene la proiezione di σ sul lato più vicino della piramide: σ^{new} .

Nel momento in cui questa proiezione mantiene inalterato l'ordinamento $\sigma_1^{new} \geq \sigma_2^{new} \geq \sigma_3^{new}$ allora la proiezione è corretta e si può calcolare direttamente il tensore elastoplastico nel sistema di riferimento principale D^{ep} . Partendo dalla relazione (1.20), che può essere derivata e si ottiene:

(1.29)
$$d\sigma^{new} = D \cdot \varepsilon - d\lambda_1 \begin{bmatrix} G + \left(K + \frac{G}{3}\right)\sin\psi\\ \left(K - \frac{2G}{3}\right)\sin\psi\\ \left(K + \frac{G}{3}\right)\sin\psi - G \end{bmatrix} = D \cdot \varepsilon - d\lambda_1 v_t^1$$

1.3. L'ALGORITMO ITERATIVO

Dalla relazione (1.18), calcoliamo $dF_{13}(\sigma)$ che rientra nel calcolo di $d\lambda_1$:

(1.30)
$$dF_{13}(\sigma) = 2 \begin{bmatrix} G + \left(K + \frac{G}{3}\right)\sin\phi\\ \left(K - \frac{2G}{3}\right)\sin\phi\\ \left(K + \frac{G}{3}\right)\sin\phi - G \end{bmatrix} \cdot d\varepsilon = 2v_s^1 \cdot d\varepsilon$$

e sostituiamo questo calcolo nel precedente per ottenere un legame fra la variazione del tensore degli sforzi $d\sigma^{new}$ e l'incremento del tensore delle deformazioni $d\varepsilon$.

La relazione che si ottiene permette di estrapolare le componenti necessarie al calcolo esplicito della *consistent tangent matrix* D^{ep} :

(1.31)
$$d\sigma^{new} = \left(D - \frac{4}{A}v_t^1 \otimes v_s^1\right)d\varepsilon$$

dove con il simbolo \otimes si intende il prodotto di Kronecker e con $A = 4 \left[G + \left(K + \frac{G}{3} \right) \sin \phi \sin \psi \right].$

Dalla relazione precedente si può dunque ottenere il calcolo esplicito di D^{ep} nel sistema di riferimento principale:

(1.32)
$$D^{ep} = D - \frac{4}{A}v_t^1 \otimes v_s^1$$

Come detto in precedenza, questo calcolo diretto può essere fatto nel momento in cui la proiezione sulla faccia della piramide mantiene inalterate la relazione d'ordine tra le componenti del tensore delle deformazioni. In caso contrario si tenta la proiezione su uno dei due spigoli della faccia, quello corretto viene determinato in base all'ordinamento che hanno gli autovalori di σ in seguito al tentativo precedente.

Le casistiche che si possono trovare sono:

se
$$\sigma_2 \ge \sigma_1 \ge \sigma_3 \to \text{considero } F_{13} \in F_{23}$$

se $\sigma_1 \ge \sigma_3 \ge \sigma_2 \to \text{considero } F_{13} \in F_{12}$

dove, in riferimento alla definizione delle facce della superficie di snervamento $(1.4), F_{23} = f_2$ e $F_{12} = f_6$. Il primo caso viene chiamato 'meccanismo a sinistra (L)', l'altro 'meccanismo a destra (R)', come si può vedere dalla Figura (1.2).

Questa definizione mostra come l'algoritmo sia pensato in modo da fare sempre i tentativi di proiezione nell'ordine corretto (faccia, spigolo, vertice) e si esce dal procedimento solo se l'ordinamento rimane inalterato, in caso



Figura 1.2: La figura a sinistra mostra una rappresentazione della superficie di Mohr-Coulomb nel sistema di riferimento principale in tre dimensioni. La figura a destra rappresenta una sezione della superficie di Mohr-Coulomb, indicando le tre facce considerate dal primo e dal secondo caso in analisi, si vede direttamente il motivo per cui si parla di 'meccanismo a sinistra (L)' e 'meccanismo a destra (R)',

contrario il valore del tensore degli sforzi calcolato al tentativo precedente è necessario per la parte successiva, per determinare il corretto spigolo. Analogamente al caso precedente si definiscono le rispettive componenti del potenziale plastico:

(1.33)
$$L: \quad G_{23}(\sigma_2, \sigma_3) = \sigma_2 - \sigma_3 + (\sigma_2 + \sigma_3) \sin \psi - 2c \cos \psi$$

(1.34)
$$R: \quad G_{12}(\sigma_1, \sigma_2) = \sigma_1 - \sigma_2 + (\sigma_1 + \sigma_2)\sin\psi - 2c\cos\psi$$

Si definisce qui il tensore degli sforzi come risultato di una proiezione che è combinazione lineare delle normali alle due facce coinvolte dallo spigolo:

(1.35)
$$\sigma^{new} = \sigma - d\lambda_1 \cdot (Dn_{G1}) - d\lambda_2 \cdot (Dn_{G2})$$

I moltiplicatori $d\lambda_1 \in d\lambda_2$ si cercano impostando un sistema di due equazioni in due incognite, simili a quella descritta in (1.22), il sistema è differente a seconda di quale lato, dunque *'meccanismo'*, venga analizzato.

1.3. L'ALGORITMO ITERATIVO

Per il 'meccanismo a sinistra (L)' si scrive il seguente sistema: (1.36)

$$\begin{cases} \sum_{i=1}^{2} d\lambda_{i} [(Dn_{Gi})_{1} - (Dn_{Gi})_{3} + ((Dn_{Gi})_{1} + (Dn_{Gi})_{3}) \sin \phi] = F_{13} \\ \sum_{i=1}^{2} d\lambda_{i} [(Dn_{Gi})_{2} - (Dn_{Gi})_{3} + ((Dn_{Gi})_{2} + (Dn_{Gi})_{3}) \sin \phi] = F_{23} \end{cases}$$

per risolvere il quale è necessario definire le normali alle due facce n_{G1} e n_{G2} , dove è corretto precisare che l'indice 1 è relativo alla faccia G_{13} definita in (1.19) e l'indice 2 a G_{23} definito in (1.33):

$$n_{G1} = \left(\frac{\partial G_{13}}{\partial \sigma_1}, \frac{\partial G_{13}}{\partial \sigma_2}, \frac{\partial G_{13}}{\partial \sigma_3}\right)^T = (1 + \sin\psi, 0, \sin\psi - 1)^T$$
$$n_{G2} = \left(\frac{\partial G_{23}}{\partial \sigma_1}, \frac{\partial G_{23}}{\partial \sigma_2}, \frac{\partial G_{23}}{\partial \sigma_3}\right)^T = (0, \sin\psi + 1, \sin\psi - 1)^T.$$
Si possono in ultimo calcolare:

(1.37)

$$Dn_{G1} = 2\left(G + \left(K + \frac{G}{3}\right)\sin\psi, \left(K - \frac{2G}{3}\right)\sin\psi, \left(K + \frac{G}{3}\right)\sin\psi - G\right)^{T}$$
(1.38)

$$Dn_{G2} = 2\left(\left(K - \frac{2G}{3}\right)\sin\psi, \left(K + \frac{G}{3}\right)\sin\psi + G, \left(K + \frac{G}{3}\right)\sin\psi - G\right)^T$$

con questi risultati si può riscrivere il sistema come:

(1.39)
$$\begin{cases} Ad\lambda_1 + Bd\lambda_2 = F_{13} \\ Bd\lambda_1 + Ad\lambda_2 = F_{23} \end{cases}$$

dove $A = 4 \left[G + \left(K + \frac{G}{3} \right) \sin \phi \sin \psi \right]$ e

$$B = 2\left[G(1 - \sin\phi - \sin\psi) + \left(2K - \frac{G}{3}\right)\sin\phi\sin\psi\right].$$

Si dimostra che il sistema è sempre risolvibile dal momento che per nessun valore ammissibile degli angoli ϕ e ψ e dei coefficienti di Lamé ν e E il determinante della matrice del sistema è nullo: $A^2 - B^2 \neq 0$. Si ottengono le seguenti soluzioni:

(1.40)
$$\begin{cases} d\lambda_1 = \frac{AF_{13} - BF_{23}}{A^2 - B^2} \\ d\lambda_2 = \frac{AF_{23} - BF_{13}}{A^2 - B^2} \end{cases}$$

Per quanto riguarda il 'meccanismo a destra (R)' i calcoli sono analoghi, partendo però dal fatto che il sistema da risolvere è: (1.41)

$$\begin{cases} \sum_{i=1}^{2} d\lambda_i [(Dn_{Gi})_1 - (Dn_{Gi})_3 + ((Dn_{Gi})_1 + (Dn_{Gi})_3) \sin \phi] = F_{13} \\ \sum_{i=1}^{2} d\lambda_i [(Dn_{Gi})_1 - (Dn_{Gi})_2 + ((Dn_{Gi})_1 + (Dn_{Gi})_2) \sin \phi] = F_{12} \end{cases}$$

La normale n_{G1} è analoga a quella scritta nel precedente caso, così come il calcolo di Dn_{G1} , perché la faccia che prende l'indice 1 è sempre la stessa. Invece tutto ciò che riguarda la seconda faccia cambia espressione e dunque si definisce $n_{G2} = \left(\frac{\partial G_{12}}{\partial \sigma_1}, \frac{\partial G_{12}}{\partial \sigma_2}, \frac{\partial G_{12}}{\partial \sigma_3}\right)^T = (\sin \psi + 1, \sin \psi - 1, 0)^T$ e di conseguenza si calcola $Dn_{G2} = 2\left(\left(K - \frac{G}{3}\right)\sin\psi + G, \left(K + \frac{G}{3}\right)\sin\psi - G, \left(K - \frac{2G}{3}\right)\sin\psi\right)^T$. Si ottiene quindi un sistema analogo a quello ottenuto in (1.39), dove il termine *B* viene definito in un modo differente:

 $B = 2\left[G(1 + \sin\phi + \sin\psi) + \left(2K - \frac{G}{3}\right)\sin\phi\sin\psi\right].$

Le considerazioni sulla risolvibilità del sistema sono analoghe a quelle svolte per il primo caso e tenendo in considerazione questa diversa definizione del parametro B si può affermare che la soluzione può essere scritta in modo analogo a quanto fatto in (1.40).

A questo punto, sia che il meccanismo attivato sia destra o sinistra, si può applicare la sostituzione dei valori di $d\lambda_1 e d\lambda_2$ trovati in (1.40) nell'equazione (1.35) che determina il nuovo valore del tensore degli sforzi σ^{new} in seguito a questo secondo tentativo.

Analogamente alla situazione di 'un meccanismo attivo' il criterio per determinare se il nuovo tensore trovato sia quello corretto è il mantenimento da parte di esso della relazione d'ordine: $\sigma_1^{new} \ge \sigma_2^{new} \ge \sigma_3^{new}$.

Nel caso in cui questa relazione sia soddisfatta, si procede anche in questo caso con il calcolo della *constitutive tangent matrix* D^{ep} nel sistema di riferimento principale.

Derivando la relazione (1.35) si può ottenere, rispettivamente per i due meccanismi:

(1.42)

$$(L): \quad d\sigma^{new} = D \cdot d\varepsilon - 2d\lambda_1 \begin{bmatrix} G + \left(K + \frac{G}{3}\right)\sin\psi\\ \left(K - \frac{2G}{3}\right)\sin\psi\\ \left(K + \frac{G}{3}\right)\sin\psi - G \end{bmatrix} - 2d\lambda_2 \begin{bmatrix} \left(K - \frac{2G}{3}\right)\sin\psi\\ \left(K + \frac{G}{3}\right)\sin\psi + G\\ \left(K + \frac{G}{3}\right)\sin\psi - G \end{bmatrix}$$

(1.43)

$$(R): \quad d\sigma^{new} = D \cdot d\varepsilon - 2d\lambda_1 \begin{bmatrix} G + \left(K + \frac{G}{3}\right)\sin\psi\\ \left(K - \frac{2G}{3}\right)\sin\psi\\ \left(K + \frac{G}{3}\right)\sin\psi - G \end{bmatrix} - 2d\lambda_2 \begin{bmatrix} \left(K + \frac{G}{3}\right)\sin\psi + G\\ \left(K + \frac{G}{3}\right)\sin\psi - G\\ \left(K - \frac{2G}{3}\right)\sin\psi - G \end{bmatrix}$$

Riscrivendo in modo più compatto entrambe possono essere scritte come

(1.44)
$$d\sigma^{new} = D \cdot d\varepsilon - 2d\lambda_1 v_t^1 - 2d\lambda_2 v_t^2$$

1.3. L'ALGORITMO ITERATIVO

Dovendo quindi sostituire in questa relazione i valori dei moltiplicatori espressi da (1.40) è utile ricordare l'espressione di dF_{13} trovata nel passaggio precedente (1.30), e calcolare l'analogo per dF_{23} e dF_{12} .

(1.45)
$$dF_{23}(\sigma) = 2 \begin{bmatrix} \left(K - \frac{2G}{3}\right)\sin\phi\\ \left(K + \frac{G}{3}\right)\sin\phi + G\\ \left(K + \frac{G}{3}\right)\sin\phi - G \end{bmatrix} \cdot d\varepsilon = 2v_s^2 \cdot d\varepsilon$$

(1.46)
$$dF_{13}(\sigma) = 2 \begin{bmatrix} \left(K + \frac{G}{3}\right)\sin\phi + G \\ \left(K + \frac{G}{3}\right)\sin\phi - G \\ \left(K - \frac{2G}{3}\right)\sin\phi \end{bmatrix} \cdot d\varepsilon = 2v_s^2 \cdot d\varepsilon$$

Eseguendo le diverse sostituzioni si arriva alla scrittura esplicita della *consi*stent tangent matrix:

(1.47)
$$D^{ep} = D - \frac{4}{A^2 - B^2} \left(v_t^1 \otimes v_s^1 + v_t^2 \otimes v_s^2 - \frac{B}{A} \left(v_t^1 \otimes v_s^2 + v_t^2 \otimes v_s^1 \right) \right)$$

Questa relazione vale per entrambi i meccanismi, ovviamente il calcolo risulterà differente per le diverse definizioni dei parametri e i vettori in gioco :A, B, $v_{\{t,s\}}^{\{1,2\}}$.

Un ultimo caso da analizzare è quello in cui anche in seguito al secondo tentativo di proiezione l'algoritmo non mantiene l'ordinamento dei componenti del tensore degli sforzi e di conseguenza l'unico caso possibile è la proiezione sull'apice della superficie, le cui coordinate nel sistema di riferimento principale sono tutte uguali a $p = c \cot \phi$, dunque il calcolo è immediato:

(1.48)
$$\sigma^{new} = \begin{bmatrix} c \cot \phi \\ c \cot \phi \\ c \cot \phi \end{bmatrix}$$

e la consistent tangent matrix risulta $D^{ep} = \mathbf{0}$. In questo caso non è necessario il calcolo dei moltiplicatori $d\lambda_i$, dal momento che si ricerca unicamente l'espressione di D^{ep} necessaria per il calcolo dello Jacobiano J_F (1.16). Ma nel caso in cui si volesse anche ottenere come output dell'algoritmo di return mapping, oltre al tensore degli sforzi σ^{new} e della matrice D^{ep} , la componente plastica del tensore delle deformazioni ε^p , allora questo calcolo sarebbe necessario. Infatti per i vari casi $d\varepsilon^p$ può essere scritta come:

- (1.49) proiezione su una faccia: $d\varepsilon^p = d\lambda_1 n_{G1}$
- (1.50) proiezione su uno spigolo: $d\varepsilon^p = d\lambda_1 n_{G1} + d\lambda_2 n_{G2}$
- (1.51) proiezione sul vertice: $d\varepsilon^p = d\lambda_1 n_{G1} + d\lambda_2 n_{G2} + d\lambda_3 n_{G3}$

Per determinare i moltiplicatori, analogamente ai casi già trattati, bisogna impostare un sistema, in questo caso di tre equazioni: (1.52)

$$\begin{cases} \sum_{i=1}^{3} d\lambda_i [(Dn_{Gi})_1 - (Dn_{Gi})_3 + ((Dn_{Gi})_1 + (Dn_{Gi})_3)\sin\phi] = F_{13} \\ \sum_{i=1}^{3} d\lambda_i [(Dn_{Gi})_2 - (Dn_{Gi})_3 + ((Dn_{Gi})_2 + (Dn_{Gi})_3)\sin\phi] = F_{23} \\ \sum_{i=1}^{3} d\lambda_i [(Dn_{Gi})_1 - (Dn_{Gi})_2 + ((Dn_{Gi})_1 + (Dn_{Gi})_2)\sin\phi] = F_{12} \end{cases}$$

svolgendo tutti i calcoli si ottiene il seguente sistema finale:

(1.53)
$$\begin{cases} Ad\lambda_1 + B^{left}d\lambda_2 + B^{right}d\lambda_3 = F_{13} \\ B^{left}d\lambda_1 + Ad\lambda_2 + B^{leftright}d\lambda_3 = F_{23} \\ B^{right}d\lambda_1 + B^{rightleft}d\lambda_2 + Ad\lambda_3 = F_{12} \end{cases}$$

dove A rimane inalterata rispetto ai casi precedenti, $B^{left} \in B^{right}$ corrispondono ai valori che B assume rispettivamente nel meccanismo a sinistra e a destra. Invece bisogna definire gli altri parametri:

(1.54)
$$B^{leftright} = 2\left(G\left(-1+\sin\psi-\sin\phi\right) + \left(2K-\frac{G}{3}\right)\sin\psi\sin\phi\right)$$

(1.55)
$$B^{rightleft} = 2\left(G\left(-1-\sin\psi-\sin\phi\right) + \left(2K - \frac{G}{3}\right)\sin\psi\sin\phi\right)$$

La risoluzione del sistema è meno immediato dei casi precedenti, si è scelto dunque di utilizzare le potenzialità del calcolatore, il quale applicherà una fattorizzazione, per risolverlo in modo efficace. Il codice scritto per l'implementazione dell'algoritmo di *return mapping* nel sistema di riferimento principale descritto in questo paragrafo viene messo in appendice.

1.3.2 Conclusione dell'algoritmo

Il calcolo della matrice di elastoplasticità descritto in precedenza permette di calcolare lo Jacobiano di F, applicando la relazione (1.16). Questo calcolo necessita di riportare D^{ep} nel sistema di riferimento canonico, e l'integrale sarà calcolato sugli elementi seguendo il metodo numerico degli Elementi Virtuali che sarà descritto in modo dettagliato nel secondo capitolo.

La proiezione di D^{ep} deve essere espressa in termini di tensore del quarto ordine, la cui matrice di rappresentazione, grazie ad una doppia simmetria di cui gode il tensore è esprimibile in forma esplicita in $\mathbb{R}^{6,6}$. Dunque partendo

1.3. L'ALGORITMO ITERATIVO

da una matrice $\mathbb{R}^{3,3}$ si applica una serie di trasformazioni⁵ dipendenti dal tensore delle deformazioni ε e da quello degli sforzi σ . È necessario, non solo ai fini dell'iterazione dell'algoritmo, calcolare σ e ϵ nel sistema di riferimento principale. Per quanto riguarda il tensore degli sforzi, si parte da quello ottenuto nell'algoritmo del *return mapping* e si applica:

(1.56)
$$\boldsymbol{\sigma} = P^T \boldsymbol{\sigma} P$$

dove P è la matrice degli autovettori di σ (coincidenti con quelli di ε) e $\sigma = \sigma^{new}$ ricavata nel precedente paragrafo. Si può utilizzare una forma analoga per ε , ricordando che $\varepsilon = \varepsilon^e + \varepsilon^p$, dove la componente elastica è nota in seguito al compimento del passo puramente lineare e quella plastica dalle relazioni espresse in (1.49) e seguenti.

All'inizio della spiegazione del metodo del return mapping è stato affermato che il calcolo della matrice elastoplastica avviene solo nel caso in cui il tensore degli sforzi, in seguito all'applicazione di un passo puramente elastico, abbia ecceduto rispetto alla superficie di snervamento. In caso contrario, ovvero nel caso in cui il comportamento del materiale sia puramente elastico, lo Jacobiano della trasformazione coincide con la matrice D, definita in (1.1).

Nel caso più generale, la superficie in esame presenta entrambi i comportamenti. Dal momento che il metodo del *return mapping* viene richiamato elemento per elemento della mesh poliedrica in cui viene suddivido l'oggetto in esame, la matrice globale che ne risulta avrà dei contributi puramente elastici e altri plastici. I test effettuati hanno comunque presentato la convergenza dell'algoritmo di Newton, il quale a priori avrebbe potuto riscontrare delle difficoltà vista la complessità del modello e della geometria coinvolta. I risultati ottenuti verranno comunque discussi e approfonditi nell'ultimo capitolo. Concludendo, una precisazione finale che deve essere fatta al fine di comprendere appieno la complessità del problema è che l'algoritmo iterativo descritto in questo capitolo coinvolge non solo una geometria complessa e dunque sistemi lineari di grandi dimensioni, ma è esso stesso parte di una discretizzazione temporale e dunque, per la risoluzione globale, sarà richiamato più volte per arrivare alla soluzione finale del problema.

⁵Per i calcoli espliciti si veda il riferimento [1]

Capitolo 2

Il metodo degli Elementi Virtuali

Il metodo degli Elementi Virtuali, Virtual Elements Method (VEM), è un metodo numerico recente, innovativo, che è stato considerato più adatto, rispetto soprattutto al metodo degli elementi finiti, per risolvere problemi con geometrie molto complesse e di grandi dimensioni. Le idee cardine di questo metodo sono diverse: innanzitutto le funzioni test e trial, che qui coincidono fra loro, non vengono espresse esplicitamente e la loro conoscenza non è necessaria per l'applicazione. Inoltre queste funzioni sono costituite da una parte polinomiale e da un'altra che non necessariamente deve essere un polinomio. Per quanto riguarda il calcolo della matrice di rigidezza non si utilizzano formule di integrazione numerica e, nel caso in cui una delle due funzioni in ingresso sia un polinomio, il calcolo del valore dell'integrale risulta esatto. In caso contrario si ottiene un valore che ha comunque il corretto ordine di grandezza e proprietà di stabilità e consistenza ¹.

Per presentare questo metodo utilizzato per risolvere il problema trattato in questa tesi in uno spazio tridimensionale, è stato ritenuto più opportuno apportare una spiegazione del metodo prima in due dimensioni e poi descriverlo in tre, analizzando le componenti necessarie per il caso più complesso.

¹Se il problema differenziale è ben posto, la soluzione u dipende in modo continuo dai dati. A livello discreto, questa dipendenza in modo continuo è chiamata *stabilità*.

Una forma bilineare è consistente se la norma del consistency error $\operatorname{err}_c \xrightarrow[h\to 0]{} 0$. Il consistency error viene definito come $L_h U_h - f_h$ dove U_h è la miglior approximazione della soluzione u in V_h e L_h è l'operatore definito come $(L_h w)(v) = -\int_{\Omega} \nabla w \nabla v dx$, con $w, v \in V_h$

2.1 Metodo in 2D

Consideriamo $\Omega \in \mathbb{R}^2$ su cui è stata generata una mesh di poligoni non sovrapposti, che singolarmente denoteremo con il termine "*elemento*" E. Questo metodo non richiede la convessità dei poligoni ma una proprietà meno restrittiva, che siano stellati. Un generico poligono E si definisce stellato se $\exists x_0 \in E \ t.c. \ \forall x \in E \ \exists (x_0, x),$ una linea, completamente contenuta in E. Per ogni poligono E viene definito lo spazio $V_k(E)$, in cui una funzione $v_h \in$ $V_k(E)$ soddisfa le seguenti proprietà:

- v_h è un polinomio di grado k su ogni lato e di E
- v_h ristretta al bordo ∂E è una funzione globalmente continua.
- Δv_h su E è un polinomio di grado k-2

La dimensione di questo spazio dipende dal numero di vertici di $E(N^V)$ e da k: N^{DoF} := dim $V_k = N^V k + \frac{k(k-1)}{2}$.

Le funzioni di base $\varphi_i \in V_k(E)$ sono definite analogamente agli altri metodi, come la base canonica dello spazio. Vengono definite grazie all'operatore $DoF_i : V_k(E) \to \mathbb{R}$: $DoF_i(v_h) = i$ -esimo grado di libertà di v_h , dove $i = 1, \dots, N^{DoF}$. Data la base canonica $\{\varphi_j\}_{j=1}^{N^{DoF}}$, definita da $DoF_j(\varphi_i) = \delta_{ij}$, si ottiene così l'espressione di una generica funzione $v_h \in V_k(E)$:

(2.1)
$$v_h = \sum_{i=1}^{N^{DoF}} DoF_i(v_h)\varphi_i$$

Globalmente viene definito lo spazio $V_h \subset H_0^1(\Omega)$ come

(2.2)
$$V_h := \left\{ v_h \in H_0^1(\Omega) : v_{h|E} \in V_k(E) \forall E \right\}$$

e vengono definiti i seguenti gradi di libertà globali per una generica funzione v_h :

- $v_h(x_i)$, dove x_i sono in nodi della mesh.
- su ogni lato $e, v_h(x_i^e)$ dove x_i^e sono i k-1 punti interni al lato e, secondo la regola di quadratura di Gauss-Lobatto.²

²Si possono definire *n* nodi di Gauss-Lobatto su un generico intevallo [a, b]: $\frac{a+b}{2} - \frac{b-a}{2}\cos\left(\frac{k\pi}{n}\right)$, dove $k = 1, \dots, n$

• su ogni poligono E, i momenti fino all'ordine k - 2 di v_h su E, definiti come:

(2.3)
$$\frac{1}{|E|} \int_E v_h m_\alpha, \text{ dove } \alpha = 1, \cdots, n_{k-2}$$

dove n_{k-2} è la dimensione dello spazio dei polinomi ³ di grado minore o uguale a k-2 e con m_{α} viene indicato il monomio definito come $\left(\frac{\boldsymbol{x}-\boldsymbol{x}_{\mathcal{D}}}{h_{\mathcal{D}}}\right)^{\alpha}$, dove $\boldsymbol{x}_{\mathcal{D}}$ e $h_{\mathcal{D}}$ sono rispettivamente il centroide e il diametro del poligono.

L'obbiettivo di questo paragrafo è quello di arrivare al calcolo della matrice di rigidezza 'VEM' in 2D , la quale, come già anticipato in precedenza, non richiede né l'utilizzo di formule di quadratura né di un'approssimazione dell'espressione delle funzioni di base φ_i , la cui espressione esplicita non è neanche necessaria. Per raggiungere questo obbiettivo è necessaria la definizione di un particolare operatore di proiezione : $\mathbf{\Pi}^{\nabla}$. Vedremo in seguito che per ottenere l'espressione della matrice di rigidezza in tre dimensioni sarà necessaria la definizione di un secondo operatore di proiezione, $\mathbf{\Pi}^0$, l'operatore di proiezione L^2 .

L'operatore di proiezione $\Pi^{\nabla}_{E,k} : V_k(E) \to \mathcal{P}_k(E)$ è definito per ogni $v_h \in V_k(E)$ dalla condizione di ortogonalità:

(2.4)
$$\left(\nabla p_k, \nabla \left(\Pi^{\nabla} v_h - v_h\right)\right)_{0,E} \,^4 \, \forall p_k \in \mathcal{P}_k(E)$$

Questa relazione definisce $\Pi^{\nabla} v_h$ a meno di una costante arbitraria, questa viene fissata definendo un operatore di proiezione sullo spazio delle costanti P_0 e richiedendo che sia valida la seguente relazione:

$$(2.5) P_0 \left(\Pi^{\nabla} v_h - v_h \right) = 0$$

Considerando $\mathcal{M}_k(E)$, l'insieme dei monomi $\{m_{\alpha} : 0 \leq |\alpha| \leq k\}$, come base per lo spazio dei polinomi $\mathcal{P}_k(E)$, la relazione (2.4) può essere riscritta per i soli elementi della base. Inoltre osserviamo che $\Pi^{\nabla} v_h$ può essere scritto come combinazione lineare degli m_{α} , dal momento che è un elemento di $\mathcal{P}_k(E)$,

³In generale la dimensione dello spazio dei polinomi di grado minore o uguale a k, $\mathcal{P}_k(E)$, viene dato da $n_k = \frac{(k+1)(k+2)}{2}$

⁴Con $(\cdot, \cdot)_{0,E}$ si intende il prodotto vettoriale in $H_0^1(E)$

i coefficienti possono essere indicati con l^{α} . La relazione (2.4) può essere riscritta come:

(2.6)
$$\sum_{\beta=1}^{n_k} l^{\beta} \left(\nabla m_{\alpha}, \nabla m_{\beta} \right)_{0,E} = \left(\nabla m_{\alpha}, \nabla v_h \right)_{(0,E)}, \quad \alpha = 1, \cdots, n_k$$

Questo sistema lineare risulta però indeterminato.È necessario aggiungere un'equazione che viene data dalla condizione aggiuntiva (2.5): $\sum_{\beta=1}^{n_k} l^{\beta} P_0 m_{\beta} = P_0 v_h$. In questo modo si ottiene un sistema lineare che determina in modo univoco i coefficienti della combinazione lineare di $\Pi^{\nabla} v_h$ rispetto alla base $\mathcal{M}_k(E)$.

Dal momento che si assume di saper calcolare l'integrale di un generico polinomio sul poligono E, la matrice del sistema è calcolabile esattamente. Il termine noto è calcolabile conoscendo solamente i gradi di libertà di v_h , infatti vale la relazione di integrazione per parti.

(2.7)
$$(\nabla m_{\alpha}, \nabla v_h)_{0,E} = -\int_E \Delta m_{\alpha} v_h + \int_{\partial E} \frac{\partial m_{\alpha}}{\partial n} v_h$$

Il primo termine $\Delta m_{\alpha} \in \mathcal{P}_{k-2}(E)$ è calcolabile conoscendo i soli momenti di v_h . Il secondo termine è l'integrale sul bordo di un polinomio di grado 2k-1 che può essere calcolato esattamente utilizzando i punti di Gauss-Lobatto. Consideriamo il sistema lineare (2.6), dal momento che le funzioni di base φ_i appartengono allo spazio $V_k(E)$, si possono definire l_i^{α} , i coefficienti della combinazione lineare di $\Pi^{\nabla}\varphi_i$ rispetto alla base m_{α} . Sostituendo v_h con φ_h in (2.6), si può ottenere il sistema:

(2.8)
$$\boldsymbol{A}\underline{l}^{(i)} = \underline{b}^{(i)}$$

dove la matrice di rigidezza A è data da:

(2.9)
$$\boldsymbol{A} = \begin{bmatrix} P_0 m_1 & P_0 m_2 & \cdots & P_0 m_{n_k} \\ 0 & (\nabla m_2, \nabla m_2)_{0,E} & \cdots & (\nabla m_2, \nabla m_{n_k})_{0,E} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & (\nabla m_{n_k}, \nabla m_2)_{0,E} & \cdots & (\nabla m_{n_k}, \nabla m_{n_k})_{0,E} \end{bmatrix}$$

Chiamiamo con \boldsymbol{B} la matrice data da (2.10)

$$\boldsymbol{B} := \begin{bmatrix} \underline{b}^{(1)}, \cdots, \underline{b}^{(N^{DoF})} \end{bmatrix} = \begin{bmatrix} P_0 \varphi_1 & \cdots & P_0 \varphi_n D_{oF} \\ (\nabla m_2, \nabla \varphi_1)_{0,E} & \cdots & (\nabla m_2, \nabla \varphi_n D_{oF})_{0,E} \\ \vdots & \ddots & \vdots \\ (\nabla m_{n_k}, \nabla \varphi_1)_{0,E} & \cdots & (\nabla m_{n_k}, \nabla \varphi_n D_{oF})_{0,E} \end{bmatrix}$$

2.1. METODO IN 2D

Dal sistema (2.8) possiamo ricavare la matrice di rappresentazione dell'operatore $\Pi^{\nabla} : V_k(E) \to \mathcal{P}_k(E)$ rispetto alla base $\mathcal{M}_k(E) : \Pi^{\nabla}_* = \mathbf{A}^{-1}\mathbf{B}$. Volendo esprimere la matrice di rappresentazione nella base canonica, è sufficiente considerare il seguente fatto:

(2.11)
$$\Pi^{\nabla}\varphi_{i} = \sum_{j=1}^{N^{DoF}} DoF_{j}(\Pi^{\nabla}\varphi_{i})\varphi_{i} = \sum_{\alpha=1}^{n_{k}} l_{i}^{\alpha} \sum_{j=1}^{N^{DoF}} DoF_{j}(m_{\alpha})\varphi_{i}$$

Definiamo la matrice F: $F_{i\alpha} = DoF_i(m_\alpha)$, dove $i = 1, \dots, N^{DoF}$ e $\alpha = 1, \dots, n_k$.

La matrice di rappresentazione dell'operatore $\mathbf{\Pi}^\nabla$ nella base canonica è data da:

(2.12)
$$\Pi^{\nabla} = \boldsymbol{F}\boldsymbol{A}^{-1}\boldsymbol{B} = \boldsymbol{F}\Pi^{\nabla}_{*}$$

A questo punto si può scrivere l'espressione della matrice di rigidezza 'VEM' in 2D per un poligono E, l'assemblamento della matrice globale viene fatta di conseguenza utilizzando la numerazione globale dei gradi di libertà e sommando tutti i contributi su ogni grado.

Consideriamo una generica funzione di base φ_i come $\Pi^{\nabla} \varphi_i + (I - \Pi^{\nabla}) \varphi_i$, si può quindi scrivere il generico elemento della matrice di rigidezza, considerando solo i termini non nulli,

$$(2.13) (\boldsymbol{K}_E)_{ij} = (\nabla\varphi_i, \nabla\varphi_j)_{0,E} = (\nabla\Pi^{\nabla}\varphi_i, \nabla\Pi^{\nabla}\varphi_j)_{0,E} + (\nabla(I - \Pi^{\nabla})\varphi_i, \nabla(I - \Pi^{\nabla})\varphi_j)_{0,E}$$

Il primo termine assicura la consistenza e viene calcolato in modo esatto, invece il secondo può essere approssimato nel modo seguente: (2.14)

$$(\nabla(I - \Pi^{\nabla})\varphi_i, \nabla(I - \Pi^{\nabla})\varphi_j)_{0,E} \approx \sum_{k=1}^{N^{DoF}} DoF_k((I - \Pi^{\nabla})\varphi_i) DoF_k((I - \Pi^{\nabla})\varphi_j)$$

Dunque la matrice di rigidezza ha la seguente espressione matriciale:

(2.15)
$$\boldsymbol{K}_E = \left(\boldsymbol{\Pi}_*^{\nabla}\right)^T \tilde{A} \left(\boldsymbol{\Pi}_*^{\nabla}\right) + (\boldsymbol{I} - \boldsymbol{\Pi}^{\nabla})^T (\boldsymbol{I} - \boldsymbol{\Pi}^{\nabla})$$

dove \tilde{A} coincide con l'espressione di A data da (2.9), tranne per la prima riga che è posta a zero.

Riassumendo i passaggi fondamentali per il calcolo della matrice di rigidezza 'VEM' in 2D:

- su ogni poligono E si calcolano le matrici A, $B \in F$ e le matrici degli operatori di proiezione $\mathbf{\Pi}^{\nabla}_* = A^{-1}B \in \mathbf{\Pi}^{\nabla} = F\mathbf{\Pi}^{\nabla}_*$
- si applica la relazione (2.15) per ottenere la matrice di rigidezza su ogni poligono E
- si costruisce la matrice di rigidezza globale sommando i contributi delle matrici locali.

2.2 Metodo in 3D

Analogamente al paragrafo precedente, l'obbiettivo è quello di descrivere come si ottiene il calcolo della matrice di rigidezza 'VEM' in 3D, partendo dalla descrizione delle funzioni di base e dei gradi di libertà in tre dimensioni. Come già anticipato, per ottenere questo risultato è necessario definire l'o-

peratore di proiezione L^2 : Π^0 .

L'operatore $\mathbf{\Pi}^0$ è definito su ogni funzione di $L^2(E)$, qui restringiamo l'analisi a $v_h \in V_k(E)$. Il polinomio $\Pi^0 v_h$ è definito dalla seguente condizione di ortogonalità: $(p_k, \Pi^0 v_h - v_h)_{0,E} = 0$, dove $p_k \in \mathcal{P}_k(E)$. Definendo con

- **H** la matrice $H_{\alpha\beta} = (m_{\alpha}, m_{\beta})_{0,E}$ dove $\alpha, \beta = 1, \cdots, n_k$
- $\Pi^0 v_h = \sum_{\alpha=1}^{n_k} l^{\alpha} m_{\alpha}$ la combinazione lineare di $\Pi^0 v_h$ rispetto agli elementi della base $\mathcal{M}_k(E)$
- $c^{\alpha} = (m_{\alpha}, v_h)_{0,E}$ il prodotto scalare di v_h con elementi della base $\mathcal{M}_k(E)$

Si ottiene il sistema lineare: $H\underline{t} = \underline{c}$.

Questo sistema non è però risolvibile conoscendo solo i gradi di libertà di v_h , perché non si riesce a calcolare direttamente <u>c</u>. Per ovviare a questo problema ricordiamo che $\Pi^{\nabla} v_h$ può essere calcolato utilizzando solamente i gradi di libertà di v_h , inoltre se v_h è un polinomio $\Pi^{\nabla} v_h$ e $\Pi^0 v_h$ coincidono con v_h .

Per i monomi m_{α} di grado $k \in k - 1$, l'idea è quella di considerare $c^{\alpha} = (m_{\alpha}, \Pi^{\nabla} v_h)_{0,E}$.

Per fare in modo che il cambiamento effettuato non alteri la correttezza del risultato, si considera un nuovo spazio $W_k(E)$ che, rispetto a $V_k(E)$, gode di una proprietà aggiuntiva:

2.2. METODO IN 3D

• $\int_E w_h m_\alpha = \int_E \Pi^{\nabla} w_h m_\alpha, \ |\alpha| = k - 1, k, \ w_h \in W_k(E)$

In questo spazio Δw_h è un polinomio di grado k, non più k-2, in E. Inoltre la nuova definizione di c^{α} coincide con la precedente e, utilizzando solamente i gradi di libertà di v_h , ora si può risolvere il sistema per trovare i coefficienti della combinazione lineare di $\Pi^0 v_h$ rispetto alla base $\mathcal{M}_k(E)$. Analogamente a quanto fatto per l'operatore di proiezione Π^{∇} , si costruisce la proiezione delle funzioni di base φ_i e le matrici di rappresentazione dell'operatore sia nella base $\mathcal{M}_k(E)$ che in quella canonica, rispettivamente Π^0_* e Π^0 . Si definisce la matrice

(2.16)
$$\boldsymbol{C}_{\alpha i} = \begin{cases} (m_{\alpha}, \varphi_i)_{0,E} & \alpha \in [1, n_{k-2}] \\ (m_{\alpha}, \Pi^{\nabla} \varphi_i)_{0,E} & \alpha \in [n_{k-2} + 1, n_k] \end{cases}$$

Si ottengono le seguenti relazioni:

$$(2.17) \qquad \qquad \Pi^0_* = \boldsymbol{H}^{-1}\boldsymbol{C}$$

$$(2.18) \qquad \qquad \Pi^0 = \boldsymbol{F} \boldsymbol{H}^{-1} \boldsymbol{C}$$

(2.19)
$$\Pi^0 \varphi_i = \sum_{\alpha=1}^{n_k} (\Pi^0_*)_{\alpha i} m_\alpha$$

Considerando il metodo in tre dimensioni è necessario specificare che $\Omega \subset \mathbb{R}^3$ è ora suddiviso in una mesh di poliedri stellati P, le cui facce sono anch'esse dei poligoni stellati. Analogamente al caso bidimensionale si definisce lo spazio $V_k(P)$, i cui elementi v_h soddisfano i seguenti criteri:

- v_h è un polinomio di grado k su ogni spigolo di P
- la restrizione di v_h ad una faccia f del poliedro appartiene allo spazio delle funzioni $W_k(f)$, seguendo la definizione data in precedenza.
- $v_{h|\partial P}$ è una funzione continua
- Δv_h è un polinomio di grado k-2 su P

Si definiscono i seguenti gradi di libertà di una funzione v_h su un poliedro P:

- il valore della funzione v_h nei vertici di P
- su ogni lato e, il valore della funzione v_h nei k-1 punti interni di Gauss Lobatto

- su ogni faccia f, i momenti fino all'ordine k-2 di v_h su f: $\int_f v_h m_\alpha$ con $\alpha = 1, \dots, n_{k-2}$ ⁵.
- i momenti fino all'ordine k 2 di v_h su P, definiti come:

(2.20)
$$\int_P v_h \mu_\alpha, \ \alpha = 1, \cdots, \nu_{k-2}$$

dove ν_{k-2} è la dimensione dello spazio dei polinomi su un insieme di \mathbb{R}^3 ⁶, con μ_h viene denominato il monomio di grado $|\alpha|$ definito da $\left(\frac{\boldsymbol{x}-\boldsymbol{x}_{\mathcal{D}}}{h_{\mathcal{D}}}\right)$ dove, analogamente al caso in due dimensioni, $\boldsymbol{x}_{\mathcal{D}}$ è il centroide e $h_{\mathcal{D}}$ il diametro del poligono.

I primi tre gruppi di gradi di libertà sono denotati con il nome gradi di libertà di bordo e l'ultimo come gradi di libertà interni.

Analogamente al caso a due dimensioni, definiamo l'operatore $DoF_i : V_k(P) \rightarrow \mathbb{R}$ come $DoF_i(v_h) =$ i-esimo grado di libertà di v_h , e le funzioni di base $\varphi_i \in V_k(P)$ come $DoF_i(\varphi_j) = \delta_{ij}$, ottenendo in questo modo l'interpolazione di Lagrange $v_h = \sum_{i=1}^{N^{DoF}} DoF_i(v_h)\varphi_i$ per ogni elemento v_h dello spazio $V_k(P)$.

La costruzione della matrice di rigidezza parte dalla costruzione dell'operatore di proiezione $\Pi_P^{\nabla} v_h$ definito da:

(2.21)
$$\begin{cases} \left(\nabla p_k, \nabla (\Pi_P^{\nabla} v_h - v_h)\right)_{0,P} = 0\\ P_0 \left(\Pi_P^{\nabla} v_h - v_h\right) = 0 \end{cases}$$

dove la prima relazione vale per tutti gli elementi p_k dello spazio \mathcal{P}_k e P_0 è l'operatore di proiezione sullo spazio delle costanti.

Il termine generico della matrice si ottiene integrando per parti come in precedenza e si verifica che ogni termine può essere calcolato utilizzando i soli gradi di libertà:

(2.22)
$$(\nabla p_k, \nabla v_h)_{0,P} = -\int_P \Delta p_k v_h + \int_{\partial P} \frac{\partial p_k}{\partial n} v_h$$

La componente del laplaciano Δp_k può essere calcolata utilizzando i gradi di libertà interni, il secondo termine viene calcolato su ogni faccia di P e viene

⁵la definizione dei momenti riprende la definizione data per la relazione (2.3)

⁶la dimensione dello spazio dei polinomi di grado minore o uguale a k, $\mathcal{P}_k(E)$, dove $E \subset \mathbb{R}^3$, è dato da $\frac{(k+1)(k+2)(k+3)}{6}$

evitato il problema che si presentava nel caso a due dimensioni in quanto la definizione dello spazio $V_k(P)$ ha già incluso il fatto che sulle facce la funzione v_h appartenga allo spazio $W_k(P)$ rendendo così possibile il calcolo dei momenti di ordine k - 1 e k.

Riassumendo i calcoli necessari:

- Per ogni faccia f si calcolano le matrici A,B e F con le espressioni ottenute nel caso a due dimensioni. Si calcola di conseguenza la matrice di rappresentazione dell'operatore di proiezione Π[∇] su ogni faccia.
- si calcolano le matrici $A, B \in F$ per il poliedro P, definite nel seguente modo:

(2.23)
$$A = \begin{bmatrix} P_0\mu_1 & P_0\mu_2 & \cdots & P_0\mu_{\nu_k} \\ 0 & (\nabla\mu_2, \nabla\mu_2)_{0,P} & \cdots & (\nabla\mu_2, \nabla\mu_{\nu_k})_{0,P} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & (\nabla\mu_{\nu_k}, \nabla\mu_2)_{0,P} & \cdots & (\nabla\mu_{\nu_k}, \nabla\mu_{\nu_k})_{0,P} \end{bmatrix}$$

(2.24)
$$B = \begin{bmatrix} P_0 \varphi_1 & \cdots & P_0 \varphi_{N^{DoF}} \\ (\nabla \mu_2, \nabla \varphi_1)_{0,P} & \cdots & (\nabla \mu_2, \nabla \varphi_{N^{DoF}})_{0,P} \\ \vdots & \ddots & \vdots \\ (\nabla \mu_{\nu_k}, \nabla \varphi_1)_{0,P} & \cdots & (\nabla \mu_{\nu_k}, \nabla \varphi_{N^{DoF}})_{0,P} \end{bmatrix}$$

(2.25)
$$F = \begin{bmatrix} DoF_1(\mu_1) & \cdots & DoF_1(\mu_{\nu_k}) \\ \vdots & \ddots & \vdots \\ DoF_{N^{DoF}}(\mu_1) & \cdots & DoF_{N^{DoF}}(\mu_{\nu_k}) \end{bmatrix}$$

• Si calcola la matrice di rappresentazione dell'operatore di proiezione Π^{∇} rispetto alla base $\mathcal{M}_k(P)$ Π^{∇}_* e quella rispetto alla base canonica Π^{∇} :

(2.26)
$$\boldsymbol{\Pi}_*^{\nabla} = \boldsymbol{A}^{-1} \boldsymbol{B}$$

$$(2.27) \qquad \qquad \mathbf{\Pi}^{\nabla} = \mathbf{F} \mathbf{A}^{-1} \mathbf{B}$$

• Si calcola la matrice di rigidezza 'VEM' come: $\boldsymbol{K} = (\boldsymbol{\Pi}^{\nabla}_{*})^{T} \tilde{\boldsymbol{A}} \boldsymbol{\Pi}^{\nabla}_{*} + h_{P} (\boldsymbol{I} - \boldsymbol{\Pi}^{\nabla})^{T} (\boldsymbol{I} - \boldsymbol{\Pi}^{\nabla}),$ dove h_{P} rappresenta il diametro del poliedro $P \in \tilde{\boldsymbol{A}}$ corrisponde alla matrice \boldsymbol{A} la cui prima riga è posta tutta a zero.

2.3 Applicazione ai problemi di elasticità

In questo paragrafo si vuole illustrare l'applicazione del metodo visto finora ad un generico problema di elasticità, il modello viene descritto sia nel caso di modelli bidimensionali che tridimensionali, non considerando casi in cui il materiale possa avere vincoli interni come l'incompressibilità. Oltre al vantaggio già spiegato precedentemente di poter utilizzare mesh poliedriche, il metodo degli elementi virtuali è un metodo efficiente dal punto di vista computazionale. Il motivo può essere ricercato nel fatto che, ad ogni passo dell'algoritmo iterativo, la legge costitutiva venga applicata una volta sola su ogni elemento.

Il problema di elasticità ha come obbiettivo quello di ricavare lo spostamento $u: \Omega \to \mathbb{R}^3$ (o \mathbb{R}^2) del corpo in esame Ω , il cui comportamento è quello di un materiale elastico. La legge costitutiva del materiale, definita su ogni punto $x \in \Omega$, lega il tensore delle deformazioni a quello degli sforzi σ ed è data da:

(2.28)
$$\boldsymbol{\sigma} = \boldsymbol{\sigma}(x, \nabla \boldsymbol{u}(x))$$

Si assume che questa legge costitutiva sia costante a tratti sulla mesh in cui viene suddiviso lo spazio Ω . Il problema elastico, data la legge costitutiva, è caratterizzato dalla seguente equazione:

(2.29)
$$\begin{cases} -\nabla \cdot \boldsymbol{\sigma} = \boldsymbol{f} \text{ su } \Omega \\ + \text{ condizioni al contorno su } \partial \Omega \end{cases}$$

dove f è la funzione che rappresenta la forza di carico esterna che causa la deformazione.

Questo problema può essere scritto in forma variazionale, chiamando con \mathcal{V} lo spazio delle deformazioni ammissibili:

(2.30)
$$\begin{cases} \text{Find } \boldsymbol{u} \in \mathcal{V} \\ \int_{\Omega} \boldsymbol{\sigma}(x, \nabla \boldsymbol{u}(x)) : \nabla \boldsymbol{v}(x) dx = \int_{\Omega} \boldsymbol{f}(x) \cdot \boldsymbol{v}(x) dx \quad \forall \boldsymbol{v} \in \mathcal{W} \end{cases}$$

dove \mathcal{W} , lo spazio delle funzioni test, coincide con quello delle variazioni di \mathcal{V} .

Consideriamo un generico poliedro(o poligono in) P su cui definiamo la restrizione della legge costitutiva (2.28) come $\boldsymbol{\sigma}_P(\nabla \boldsymbol{u}(x))$. Inoltre per ogni coppia di funzioni $\boldsymbol{v} \in \mathcal{V}$ e $\boldsymbol{w} \in \mathcal{W}$ definiamo le forme bilineari $a_P(\boldsymbol{v}, \boldsymbol{w})$ e

 $a(\boldsymbol{v}, \boldsymbol{w})$ come:

$$a_P(\boldsymbol{v}, \boldsymbol{w}) = \int_P \boldsymbol{\sigma}(x, \nabla \boldsymbol{v}(x)) : \nabla \boldsymbol{w}(x) dx$$
$$a(\boldsymbol{v}, \boldsymbol{w}) = \int_\Omega \boldsymbol{\sigma}(\nabla \boldsymbol{v}(x)) : \nabla \boldsymbol{w}(x) dx$$

dove vale la relazione $a(\boldsymbol{v}, \boldsymbol{w}) = \sum_{P \in \Omega} a_P(\boldsymbol{v}, \boldsymbol{w}).$

Considero l'operatore di proiezione Π^0 , definito sullo spazio $V_k(P)$, come descritto in precedenza, e lo spazio dei polinomi di primo grado $\mathcal{P}_1(P)$, valgono le seguenti uguaglianze:

$$\begin{aligned} a_P(\boldsymbol{p}, \boldsymbol{v}_h) &= \int_P \boldsymbol{\sigma}_P(\nabla \boldsymbol{p}(x)) : \nabla \boldsymbol{v}_h(x) dx \\ &= \int_P \boldsymbol{\sigma}_P(\nabla \boldsymbol{p}(x)) : (\Pi^0(\nabla \boldsymbol{v}_h)(x)) dx \\ &= \int_P \boldsymbol{\sigma}_P(\Pi^0(\nabla \boldsymbol{p})(x)) : (\Pi^0(\nabla \boldsymbol{v}_h)(x)) dx \quad \forall \boldsymbol{p} \in \mathcal{P}_1(P), \forall \boldsymbol{v}_h \in \mathcal{V}_k(P) \end{aligned}$$

Si ottiene dunque una forma bilineare esprimibile come

 $\tilde{a}_{h,P}(\boldsymbol{p},\boldsymbol{v}_h) = \int_P \boldsymbol{\sigma}_P(\Pi^0(\nabla \boldsymbol{p})(x)) : (\Pi^0(\nabla \boldsymbol{v}_h)(x)) dx.$ Questa forma non è stabile, al fine di stabilizzarla si introduce l'operatore $S_P: \mathcal{V}_k(P) \times \mathcal{V}_k(P) \to \mathbb{R}$ tale che

(2.31)
$$S_P(\boldsymbol{v}_h, \boldsymbol{w}_h) = h_P^{d-2} \sum_{x \in \partial P} \boldsymbol{v}_h(x) \boldsymbol{w}_h(x)$$

dove h è il massimo diametro di P all'interno della mesh e d è la dimensione dello spazio, dunque $d \in \{2, 3\}$. Questo operatore può essere inteso come un'energia locale del corpo elastico, definito sulla componente ortogonale dello spazio dei polinomi $\mathcal{P}_1(P)$ rispetto a $\mathcal{V}_k(P)$. Si definisce la forma bilineare locale "virtuale"

(2.32)
$$a_{h,P}(\boldsymbol{s}_h; \boldsymbol{v}_h, \boldsymbol{w}_h) = \tilde{a}_{h,P}(\boldsymbol{v}_h, \boldsymbol{w}_h) + \alpha(\boldsymbol{s}_h)S_P(\boldsymbol{v}_h - \Pi_P^{\nabla}\boldsymbol{v}_h, \boldsymbol{w}_h - \Pi_P^{\nabla}\boldsymbol{w}_h)$$

dove il parametro di stabilizzazione α dipende da un ulteriore ingresso $s_h \in$ $\mathcal{V}_k(P)$. Una scelta per il parametro α , indicato in [2], può essere $\alpha_P(\mathbf{s}_h) =$ $\left\| \frac{\partial \boldsymbol{\sigma}_P}{\partial \nabla \boldsymbol{u}} (\Pi^0 \nabla \boldsymbol{s}_h) \right\|$ $\forall s_h \in \mathcal{V}_k(P)$. Questa scelta dà buoni risultati per una vasta gamma di materiali e problemi differenti. Osservando questo valore e volendosi ricollegare al problema descritto nel capitolo precedente, esso è riconducibile essenzialmente alla norma del tensore del quarto ordine che lega la variazione del tensore degli sforzi a quello di elasticità. Nel caso non elastico del problema, che vedremo nel prossimo capitolo, la scelta potrebbe ricadere sul calcolare la norma del tensore elastoplastico D^{ep} .

Il problema variazionale (2.30) può essere riscritto con la forma bilineare (2.32), dove il parametro in input s_h è dato:

(2.33)
$$\begin{cases} \text{Find } u_h \in \mathcal{V}_k(P) \\ a_{h,P}(\boldsymbol{s}_h; \boldsymbol{u}_h, \boldsymbol{v}_h) = \langle \boldsymbol{f}, \boldsymbol{v}_h \rangle \ \forall \boldsymbol{v}_h \in \mathcal{V}_k(P) \end{cases}$$

Il termine noto viene calcolato utilizzando una formula di quadratura, i cui pesi w_x sono costruiti in modo tale che la formula sia esatta qualora le funzioni su cui è applicata siano lineari:

 $\langle \boldsymbol{f}, \boldsymbol{v}_h \rangle = \sum_{x \in \partial P} w_x \boldsymbol{f}(x) \boldsymbol{v}_h(x).$

Una scelta possibile per la funzione in ingresso s_h può essere u_h . Inoltre per implementare più facilmente il problema variazionale (2.33), si assume che la funzione di carico f sia incrementale e dunque la somma di carichi parziali f^i , dove $i = 1, \dots, N$. Dato lo spostamento iniziale u_0 , si applica N volte la procedura:

(2.34)
$$\begin{cases} \text{Find } u_h^i \in \mathcal{V}_k(P) \\ a_{h,P}(\boldsymbol{u}_h^{i-1}; \boldsymbol{u}_h^i, \boldsymbol{v}_h) = \langle \boldsymbol{f}^i, \boldsymbol{v}_h \rangle \ \forall \boldsymbol{v}_h \in \mathcal{V}_k(P) \end{cases}$$

dove

$$\begin{aligned} a_{h,P}(\boldsymbol{u}_{h}^{i-1};\boldsymbol{u}_{h}^{i},\boldsymbol{v}_{h}) &= \tilde{a}_{h,P}(\boldsymbol{u}_{h}^{i},\boldsymbol{v}_{h}) + \alpha(\boldsymbol{u}_{h}^{i-1})S_{P}(\boldsymbol{u}_{h}^{i} - \Pi_{P}^{\nabla}\boldsymbol{u}_{h}^{i},\boldsymbol{v}_{h} - \Pi_{P}^{\nabla}\boldsymbol{v}_{h}) \\ &= \sum_{P \in \Omega} \int_{P} \boldsymbol{\sigma}_{P}(\Pi^{0}(\nabla \boldsymbol{u}_{h}^{i})(x)) : (\Pi^{0}(\nabla \boldsymbol{v}_{h})(x))dx + \alpha(\boldsymbol{u}_{h}^{i-1})h_{P}^{d-2}\sum_{x \in \partial P} \boldsymbol{u}_{h}^{i}\boldsymbol{v}_{h}(x) \end{aligned}$$

È importante sottolineare nuovamente che questa forma bilineare è calcolabile esattamente dai soli gradi di libertà. La soluzione finale del problema u_h coincide con la soluzione che si ottiene con l'ultima iterata.

2.4 Applicazione ai problemi di elastoplasticità

Il problema elastoplastico può essere anch'esso rappresentato in forma analoga a quella vista nel caso precedente (2.29), dove anche in questo caso l'obbiettivo è quello di calcolare lo spostamento $u: \Omega \to \mathbb{R}^3(o \mathbb{R}^2)$ del corpo in esame Ω . Una differenza sostanziale con il caso precedente è l'introduzione dello spazio temporale: deve essere effettuata una precedente discretizzazione temporale prima di applicare il metodo. Questo si riflette anche sulla funzione di carico f che viene intesa come una sequenza di carichi parziali f^n , $n = 1, \dots, N$. Globalmente il problema va inteso come un'analisi quasistatica degli sforzi e delle deformazioni che, ad ogni istante di tempo preso in esame,devono soddisfare il problema iniziale (2.29).

La legge costitutiva, relativa al tensore degli sforzi σ , dipende rispetto al caso elastico da un parametro aggiuntivo H(x) dove $x \in \Omega$, il quale rappresenta tutta la storia passata di deformazione del punto x in esame.

(2.35)
$$\boldsymbol{\sigma} = \boldsymbol{\sigma}(x, \nabla u(x), H(x))$$

In aggiunta alla legge costitutiva viene definita una legge evolutiva che caratterizza la funzione H(x):

(2.36)
$$\dot{H}(x) = g(x, \nabla u(x), \dot{\nabla} u(x), H(x))$$

dove la derivata espressa nella relazione è quella rispetto al tempo. Il problema variazionale può essere quindi ottenuto, prima della discretizzazione temporale, come

(2.37)

$$\begin{cases} \text{Find } \boldsymbol{u}(t) \in \mathcal{V} \\ \int_{\Omega} \boldsymbol{\sigma}(x, \nabla \boldsymbol{u}(x), H(x, t)) : \nabla \boldsymbol{v}(x) dx = \int_{\Omega} \boldsymbol{f}(t, x) \cdot \boldsymbol{v}(x) dx \quad \forall \boldsymbol{v} \in \mathcal{W} \end{cases}$$

Applicando la discretizzazione temporale e in seguito quella spaziale (consideriamo un generico P in Ω), si può calcolare il tensore degli sforzi avendo i seguenti valori noti al passo temporale precedente n-1: $\nabla u_h^{n-1}(x)$, H(x, n-1). Inoltre si ha un valore per $\nabla u_h^n(x)$ poiché questo rappresenta un tentativo di deformazione al passo n che può essere stimato (nel capitolo precedente è stato discusso il metodo di risoluzione del problema elastoplastico e il valore $\nabla u_h^n(x)$ può essere inteso come il tensore delle deformazioni ε al passo n). Dunque possiamo esprimere il tensore degli sforzi:

 $\sigma_P(\nabla u_h^{n-1}(x), H(x, n-1), \nabla u_h^n(x)).$

La forma bilineare è ottenuta analogamente al caso elastico, prendendo in considerazione la dipendenza del tensore degli sforzi anche dal fattore H(x,t), inoltre si assume di attribuire al parametro s_h , in input, il valore dello spostamento al tempo precedente rispetto a quello in esame u_h^{n-1} . Su ogni elemento P della mesh vale dunque la relazione per la forma bilineare:

$$a_{h,P}(\boldsymbol{u}_{h}^{n-1},\boldsymbol{u}_{h}^{n},H(n-1,x_{P}),\boldsymbol{v}_{h}) = |P|\sigma_{P}(\Pi^{0}\nabla\boldsymbol{u}_{h}^{n-1},H(n-1,x_{P}),\Pi^{0}\nabla\boldsymbol{u}_{h}^{n}):$$
$$\Pi^{0}(\nabla\boldsymbol{v}_{h}) + \alpha(\boldsymbol{u}_{h}^{n-1})S_{h,P}(\boldsymbol{u}_{h}^{n}-\Pi^{\nabla}\boldsymbol{u}_{h}^{n},\boldsymbol{v}_{h}-\Pi^{\nabla}\boldsymbol{v}_{h})$$

dove la forma bilineare $S_{h,P}$ e il parametro α vengono definiti e calcolati analogamente al caso precedente, in particolare per quanto riguarda α qui si può applicare come stima la norma del tensore di elastoplasticità D^{ep} , il quale viene calcolato con i metodi descritti nel primo capitolo.

Complessivamente la forma bilineare sullo spazio Ω si definisce sommando i contributi definiti precedentemente:

(2.38)
$$a_h(\boldsymbol{u}_h^{n-1}, \boldsymbol{u}_h^n, H(n-1), \boldsymbol{v}_h) = \sum_{P \in \Omega} a_{h,P}(\boldsymbol{u}_h^{n-1}, \boldsymbol{u}_h^n, H(n-1, x_P), \boldsymbol{v}_h)$$

Si ottiene quindi il problema variazionale per n, istante temporale, $n = 1, \dots, N$:

(2.39)
$$\begin{cases} \text{Find } u_h \in \mathcal{V}_k(P) \\ a_h(\boldsymbol{u}_h^{n-1}; \boldsymbol{u}_h^n, H(n-1), \boldsymbol{v}_h) = \langle \boldsymbol{f}^n, \boldsymbol{v}_h \rangle \ \forall \boldsymbol{v}_h \in \mathcal{V}_k(P) \end{cases}$$

Per concludere è interessante presentare un teorema ⁷ che mostra la convergenza in norma H^1 della soluzione $\boldsymbol{u} \in \mathcal{V}$, soluzione del problema (2.30), formulazione valida anche per il caso elastoplastico.

Teorema 2.4.1. Sia $\mathbf{u} \in \mathcal{V}$, soluzione del problema (2.30), sia $\mathbf{u}_h \in \mathcal{V}_k(P)$ la soluzione del problema (2.33), dato $\mathbf{s}_h \in \mathcal{V}_k(P)$:

(2.40)
$$\begin{cases} Find \ u_h \in \mathcal{V}_k(P) \\ a_{h,P}(\boldsymbol{s}_h; \boldsymbol{u}_h, \boldsymbol{v}_h) = \langle \boldsymbol{f}, \boldsymbol{v}_h \rangle \ \forall \boldsymbol{v}_h \in \mathcal{V}_k(P) \end{cases}$$

Per ogni $\boldsymbol{u}_I \in \mathcal{V}_k(P)$ e $\boldsymbol{u}_T \in L^2(\Omega)$ tale che $\boldsymbol{u}_T \in P_1(P)$ su ogni $P; \exists c \in \mathbb{R}$:

(2.41)
$$|\boldsymbol{u} - \boldsymbol{u}_h|_{1,\Omega} \leq c. \left(\sup_{v_h \in \mathcal{V}_h} \frac{\langle \boldsymbol{f}, \boldsymbol{v}_h \rangle - (\boldsymbol{f}, \boldsymbol{v}_h)}{|v_h|_{1,\Omega}} + |\boldsymbol{u} - \boldsymbol{u}_I|_{1,\Omega} + |\boldsymbol{u} - \boldsymbol{u}_T|_1 \right)$$

dove (\cdot, \cdot) è il prodotto scalare in $L^2(\Omega)$, $|\cdot|_{1,\Omega}$ indica la seminorma in H^1 , rispetto alla decomposizione definita su Ω .

⁷Per la dimostrazione si faccia riferimento a [2]

Capitolo 3

Conclusioni e risultati

In questo capitolo conclusivo si vogliono presentare alcune considerazioni che potrebbero migliorare la performance dell'algoritmo presentato nel primo capitolo. In particolare verrano trattate semplificazioni algebriche applicabili all'algoritmo di *return mapping* e verrà descritta un'approssimazione continua della superficie di snervamento di Mohr-Coulomb. Infine, per concludere questo lavoro, verrà mostrato un particolare esempio in tre dimensioni, che descrive un parallelepipedo vincolato su due facce sottoposto ad una forza parallela ad esse. Il risultato ottenuto sarà confrontato con un risultato dato da un codice *Matlab*, fornito in [12],applicato sul medesimo problema.

Le modifiche riguardanti l'algoritmo di *return mapping* sono volte alla riduzione della complessità dei calcoli, in particolare viene trattato il caso di proiezione su uno spigolo e semplificato il calcolo dei moltiplicatori plastici $d\lambda_i$, necessari al fine di ottenere l'espressione del tensore degli sforzi σ^{new} in seguito al tentativo di proiezione. Consideriamo lo spigolo *left* (L), questo è dato dall' intersezione di F_{13} e F_{23} , ovvero nel caso in cui

(3.1)
$$\sigma_1 - \sigma_3 + (\sigma_1 + \sigma_3) \sin \phi = \sigma_2 - \sigma_3 + (\sigma_2 + \sigma_3) \sin \phi$$

Nello spazio delle tensioni principali σ_i , l'intersezione di questi due piani è data da un'unica retta, che definisce lo spigolo della superficie di snervamento. I punti che giacciono su questa retta soddisfano la seguente relazione:

$$(3.2) \sigma_1 = \sigma_2 > \sigma_3$$

A partire da questa considerazione il sistema (1.39), che era stato definito per trovare i moltiplicatori $d\lambda_1$ e $d\lambda_2$, può essere riscritto in un'unica equazione:

(3.3)
$$d\lambda_1 = d\lambda_2 = \frac{F_{13}(\sigma) + F_{23}(\sigma)}{A + B}$$

dove ricordiamo che σ qui rappresenta il tensore degli sforzi a seguito del compimento del passo elastico ed $A \in B$ sono valori numerici ottenibili dai coefficienti di Lamè, dal frictional angle ϕ e dal dilatancy angle ψ . Si verifica che, applicando la relazione di proiezione data da (1.42), si ottiene correttamente un tensore degli sforzi che abbia i valori delle prime due componenti coincidenti.

Una semplificazione analoga può essere fatta per il caso dello spigolo *right* (R) e per l'apice della piramide. Per quest'ultimo caso, come già evidenziato nel primo capitolo, il calcolo viene svolto nel momento in cui si vuole calcolare il tensore delle deformazioni, altrimenti la definizione di σ^{new} è immediata.

Il discorso inerente all'approssimazione continua della superficie di snervamento di Mohr-Coulomb è molto ampio, qui si vogliono dare le idee generali per ottenere un'approssimazione prima continua e poi C2. ¹

Innanzitutto per trattare queste modifiche si parte dalla riscrittura della superficie di snervamento di Mohr-Coulomb (1.3) utilizzando un set di invarianti del tensore degli sforzi. Questo set di invarianti viene descritto a partire dalle sei componenti indipendenti del tensore degli sforzi $\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{yz}, \tau_{zx}$:

$$\sigma_m = \frac{1}{3} \left(\sigma_x + \sigma_y + \sigma_z \right)$$
$$\bar{\sigma} = \sqrt{\frac{1}{2} \left(\left(\sigma_x - \sigma_m \right)^2 + \left(\sigma_y - \sigma_m \right)^2 + \left(\sigma_z - \sigma_m \right)^2 \right) + \tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2}$$
$$\vartheta = \frac{1}{3} \arcsin\left(-\frac{3\sqrt{3}}{2} \frac{J_3}{\bar{\sigma}^3} \right)$$

dove

 $J_{3} = (\sigma_{x} - \sigma_{m}) (\sigma_{y} - \sigma_{m}) (\sigma_{z} - \sigma_{m}) + 2\tau_{xy}\tau_{yz}\tau_{zx} - (\sigma_{x} - \sigma_{m}) \tau_{yz}^{2} - (\sigma_{y} - \sigma_{m}) \tau_{zx}^{2} - (\sigma_{z} - \sigma_{m}) \tau_{xy}^{2}$

L'espressione della superficie di Mohr-Coulomb che si ottiene è:

(3.4)
$$F = \sigma_m \sin \phi + \bar{\sigma} K(\vartheta) - c \cos \phi = 0$$

dove $K(\vartheta) = \cos \theta - \frac{\sin \phi \sin \vartheta}{\sqrt{3}}.$

Fissando un valore costante per ϑ , si ottiene sul piano $\sigma_m \bar{\sigma}$ una sezione della piramide di Mohr-Coulomb, in particolare viene definita una relazione lineare tra $\sigma_m e \bar{\sigma}$ come può essere vista in Figura (3.1). Da questa sezione l'obbiettivo è rimuovere la singolarità sull'apice della piramide e per fare questo si

¹Per la descrizione più dettagliata di tutti gli aspetti si rimanda a [13],[14],[15].



Figura 3.1: Sezione della superficie di Mohr-Coulomb data da $K(\vartheta) = cost$ rappresentata in verde e rappresentazione dell'approssimazione iperbolica della superficie stessa. Si vede dal grafico come l'approssimazione dipenda dalla scelta del parametro a, la distanza del vertice dell'iperbole dall'apice della piramide.

utilizza un'approssimazione iperbolica, la quale dipende da un parametro *a* che rappresenta la distanza tra la punta della piramide e il vertice dell'iperbole di approssimazione. L'iperbole approssimante viene costruita definendo come asintoti le rette che devono essere approssimate, si ottiene dunque la nuova espressione della superficie di Mohr-Coulomb:

(3.5)
$$F = \sigma_m + \sqrt{\bar{\sigma}^2 K(\vartheta)^2 + a^2 \sin^2 \phi} - c \cos \phi = 0$$

dove la scelta del valore del parametro *a* determina un'iperbole più o meno vicina alla superficie di snervamento originaria. Per valori di $a \leq 0.25c \cot \phi$ la superficie iperbolica rappresenta in modo molto preciso la superficie di Mohr-Coulomb(1.3).

Una seconda approssimazione vuole essere fatta al fine di eliminare la singolarità relativa agli angoli interni degli spigoli della piramide, per fare ciò si utilizza un'approssimazione trigonometrica solo nella regione vicina agli spigoli. Per ottenere questo risultato si ridefinisce $K(\vartheta)$ in funzione di un parametro arbitrario ϑ_T :

(3.6)
$$K(\vartheta) = \begin{cases} A - B\sin 3\vartheta & \text{se } |\vartheta| > \vartheta_T \\ \cos \theta - \frac{\sin \phi \sin \vartheta}{\sqrt{3}} & \text{se } |\vartheta| \le \vartheta_T \end{cases}$$

dove $A \in B$ sono parametri determinati dalle condizioni di continuità C1 che si impongono alla superficie di approssimazione. Nel caso in cui si volesse una continuità di tipo C2, la definizione di $K(\vartheta)$ deve essere leggermente modificata:

(3.7)
$$K(\vartheta) = \begin{cases} A - B\sin 3\vartheta + C\sin^2 3\vartheta & \text{se } |\vartheta| > \vartheta_T \\ \cos \theta - \frac{\sin \phi \sin \vartheta}{\sqrt{3}} & \text{se } |\vartheta| \le \vartheta_T \end{cases}$$

dove coerentemente al caso precedente i parametri $A, B \in C$ sono determinati² per garantire la continuità dell'espressione della superficie di snervamento. In particolare per avere la continuità C2 della superficie è sufficiente imporre la continuità C2 a $K(\vartheta)$. È interessante notare come il problema della continuità della superficie coinvolga solo questa seconda approssimazione, dal momento che la prima soddisfa la condizione di continuità C2 senza bisogno di ulteriori modifiche.

Mettendo insieme entrambe le approssimazioni, vale l'espressione della superficie di snervamento indicata da (3.5) e (3.7), dove vi sono due parametri liberi $a \in \vartheta_T$, i cui valori determinano la bontà dell'approssimazione. Come indicato in [13] e [14], la superficie di partenza di Mohr-Coulomb si ottiene assegnando i valori $a = 0 \in \vartheta_T = 30^\circ$. Valori consigliati che possono approssimare in modo soddisfacente la superficie sono $a = 0.05c \cot \phi \in \vartheta_T = 25^\circ$.

3.1 Test di confronto

Per concludere questo lavoro, l'idea è stata quella di presentare un risultato particolare di simulazione su un modello semplice in tre dimensioni. Utilizzando i medesimi parametri di input, che verranno elencati in seguito, si è voluto far risolvere il problema sia al programma a cui si riferisce questa trattazione sia ad un programma *Matlab* dato come appendice all'articolo di S.Sysala, M. Cermak e T. Ligursky 'Subdifferential-based implicit return

²Per l'espressione esatta dei parametri $A, B \in C$ si rimanda alla fonte [14].

3.1. TEST DI CONFRONTO

mapping operators in Mohr-Coulomb plasticity'. Quest'ultimo codice è stato leggermente riadattato per poter ricevere e trattare correttamente i dati in input, per il resto è stato mantenuto l'algoritmo che questo articolo utilizzava per risolvere il problema della deformazione di un modello elastoplastico, utilizzando la superficie di snervamento di Mohr-Coulomb.

Il caso-test riguarda la deformazione di un oggetto, la cui forma corrisponde ad un parallelepipedo di lunghezza unitaria per le dimensioni $y \in z$ e di lunghezza 5 per la dimensione x. La mesh di discretizzazione spaziale è composta da tetraedri per quando riguarda il programma relativo a questo lavoro e da esaedri per il codice di confronto. In entrambi i casi è stato fissato un parametro di massimo volume $V_{max} = 0.01$.

L'oggetto in esame è sottoposto ad una forza su ogni punto $P = (x_P, y_P, z_P)$:

(3.8)
$$F(P) = (F_x, F_y, F_z)|_P = \begin{cases} (0, 0, 0.8) & \text{se } 2 < x_P < 3\\ (0, 0, 0) & \text{altrimenti} \end{cases}$$

Per questa forza sono stati definiti 1000 passi di carico. Sono state date condizioni al bordo di Neumann omogeneo per tutti i lati, ad eccezione delle due facce date da x = 0 e x = 5 che sono da considerarsi fisse, quindi con condizioni di Dirichlet omogeneo. Per quanto riguarda i parametri della superficie di snervamento di Mohr-Coulomb e del potenziale di flusso si devono dare i seguenti valori in input:

- Coefficienti di Lamè: modulo di Young $E=7.675\cdot 10^{-3}$ e modulo di Poisson $\nu=0.3$
- dilatancy angle $\psi = 80^{\circ}$ e frictional angle $\phi = 40^{\circ}$
- parametro di coesione c = 1.5

Definiti i parametri, i programmi hanno prodotto risultati confrontabili, in particolare si vuole rappresentare la deformazione ottenuta in entrambi i casi lungo i tre assi di riferimento. Osservando le figure (3.2 - 3.7) si vede come in tutte e tre le direzioni dello spostamento i risultati ottenuti dai due diversi programmi sono molto simili. Questo fatto dà una preliminare validazione dell'algoritmo descritto e implementato. Ovviamente i punti di vista diversi, soprattutto riguardanti la generazione della mesh e il metodo numerico utilizzato, limitano la validazione all'osservazione di un medesimo comportamento del materiale in esame. Questo confronto, nell'ambito di questo lavoro, viene considerato come uno strumento che ha permesso di capire che ciò che è stato fatto finora procede verso una corretta direzione. Viene quindi considerato come punto di inizio per un lavoro futuro, volto a cercare di migliorare sempre di più l'algoritmo al fine di poterlo estendere in modo soddisfacente al più ampio spettro di situazioni e problemi possibili.



Figura 3.2: Rappresentazione grafica della deformazione lungo la componente x, nel caso del programma di riferimento di questo lavoro.



Figura 3.3: Rappresentazione grafica della deformazione lungo la componente x, nel caso del programma *Matlab* di confronto.



Figura 3.4: Rappresentazione grafica della deformazione lungo la componente y, nel caso del programma di riferimento di questo lavoro.



Figura 3.5: Rappresentazione grafica della deformazione lungo la componente y, nel caso del programma *Matlab* di confronto.



Figura 3.6: Rappresentazione grafica della deformazione lungo la componente z, nel caso del programma di riferimento di questo lavoro.



Figura 3.7: Rappresentazione grafica della deformazione lungo la componente z, nel caso del programma *Matlab* di confronto.

3.2 Analisi del programma di confronto

In quest'ultima sezione si sono volute analizzare le performances del codice Matlab che è stato utilizzato per il confronto nel paragrafo precedente. Il motivo è quello di testare il comportamento di questo programma all'aumentare della forza di carico per vedere fino a quando il metodo di Newton converga. E importante sottolineare il fatto che questo programma applichi la funzione di snervamento di Mohr-Coulomb senza alcuna approssimazione, il metodo di Eulero Implicito per la discretizzazione temporale, il metodo ad Elementi Finiti con mesh esaedrica per la discretizzazione spaziale e il metodo di Newton-Raphson per la risoluzione del sistema non lineare. L'algoritmo è stato analizzato per diversi valori della funzione di carico attraverso due analisi: la rappresentazione della funzione di snervamento e il massimo valore di deformazione lungo l'asse z, entrambi ad ogni passo di carico. Infatti questo algoritmo, analogamente a quello descritto nella tesi, risolve il problema considerando incrementi dell'intensità di carico, quindi suddividendo in parti uguali l'intensità totale (in questi esempi si parla di 1000 passi di carico) e analizzando di conseguenza piccole deformazioni ad ogni passo, dove il risultato di un singolo passo viene sommato a quello precedente ed è il punto di partenza per quello successivo.

E stato scelto di analizzare la deformazione lungo l'asse z perchè la forza viene considerata diretta lungo quell'asse, come nell'esempio precedente di confronto.

Possiamo definire la generica forza applicata F dipendente da un parametro f_z :

(3.9)
$$F(P) = (F_x, F_y, F_z)|_P = \begin{cases} (0, 0, f_z) & \text{se } 2 < x_P < 3\\ (0, 0, 0) & \text{altrimenti} \end{cases}$$

La forza di carico f_z assume nei test i seguenti valori:

 $f_z \in \{0.8, 0.8, 1.2, 1.6, 2, 2.5, 2.8, 3, 3.5, 4, 4.5, 5, 6, 7.5, 9, 10\}.$

Complessivamente si nota che, come ci si poteva aspettare, all'aumentare della forza di carico l'oggetto plasticizza prima, infatti analizzando i grafici della funzione di snervamento, dove ad ogni passo di carico si considera il massimo valore assunto dalla funzione, si osserva che, per valori di f_z maggiori, la funzione arriva prima a valere zero, ovvero a rappresentare una situazione di plasticità. Si osservi la Figura (3.8) che rappresenta la funzione di snervamento per alcuni dei diversi valori assunti nel test dall'intensità totale di carico.

La stessa osservazione può essere fatta dai grafici che rappresentano il va-



Figura 3.8: Funzione di snervamento ad ogni passo di carico per quattro valori diversi di intensità di carico: 0.8, 2, 4, 6. Sull'asse delle ascisse vengono presentati i passi di carico, ognuno corrispondente ad un'intensità applicata di $\frac{f_z}{10^3}$. Sull'asse delle ordinate viene rappresentato il massimo valore assunto dalla funzione di snervamento di Mohr-Coulomb valutata su tutti i vertici della mesh alla fine di ogni singolo passo di carico.

lore massimo di deformazione ad ogni passo di carico, quattro particolari soluzioni vengono rappresentate in Figura (3.9). Da questi grafici si osserva come, nei casi in cui il metodo di Newton non converga, il comportamento rappresentato sia diverso da quello atteso, infatti vi è un aumento elevato del valore della deformazione ad ogni passo.

Infine si è voluto analizzare l'andamento della robustezza dell'algoritmo di Newton all'aumentare della forza di carico, sono stati rappresentati in Figura(3.10) i valori corrispondenti all'ultimo passo di carico risolto dall'algoritmo.

Si nota come a partire da valori superiori a $f_z = 3$ l'algoritmo di Newton ad un certo passo di carico non converga più, dunque il metodo non arriva al compimento di tutti i 1000 incrementi dell'intensità totale e il risultato finale non è esatto. Questo fatto è attribuibile all'utilizzo del modello di Mohr-Coulomb senza approssimazioni in casi in cui il comportamento plasti-



Figura 3.9: Massimo valore di deformazione ad ogni passo di carico per quattro valori diversi di intensità di carico: 0.8, 2.5, 3.5, 4.5. Analogamente al grafico precedente, sull'asse delle ascisse vengono presentati i passi di carico, ognuno di $\frac{f_z}{10^3}$. Sull'asse delle ordinate viene rappresentato il massimo valore di deformazione lungo l'asse z calcolato su ogni vertice della mesh alla fine di ogni singolo passo di carico.

co è caratterizzante. Per questo motivo, uno dei lavori futuri sarà quello di utilizzare e implementare la superficie approssimata di Mohr-Coulomb, come viene descritta all'inizio di questo capitolo conclusivo. L'obbiettivo è, come già specificato, quello di modificare e migliorare il modello descritto in modo che possa risolvere il più ampio spettro di problemi possibili.



Figura 3.10: Ad ogni intensità di carico diversa si è considerato il numero di step che l'algoritmo ha effettuato prima che il metodo di Newton non convergesse più. Considerando il fatto che l'intensità totale di carico sia stata divisa in 1000 incrementi, per tutti quei valori di intensità totale per cui il valore sull'asse delle ordinate corrisponde a 1000 step il metodo di Newton è arrivato a convergenza su ogni passo del metodo.

Appendice A

Codice dell'algoritmo 'return mapping'

```
void projectionMC(const Vector3d& eigenvaluesSigma, const Matrix3d&
                    eigenvectorsSigma, const Vector3d& eigenvaluesEpsilon,
                    const double cMC, const double psi, const double phi,
                    const double G, const double K, Vector3d& projectedSigma,
                    Matrix3d & D ,Vector3d & epsilon)
{
    double delta_Lambda;
    Vector3d plasticEpsilon; // d_epsilon p
    Matrix3d delta_T; //matrice D nel sistema di riferimento principale
    double t=sin(psi);
    double s=sin(phi);
    //coefficienti sistema per trovare d_lambda
    double A= 4*(G+(K+G/3)*t*s);
    double B_left=2*(G*(1-t-s)+(2*K-G/3)*t*s);
    double B_right=2*(G*(1+t+s)+(2*K-G/3)*t*s);
    double B_left_right=2*(G*(-1+t-s)+(2*K-G/3)*t*s);
    double B_right_left=2*(G*(-1-t+s)+(2*K-G/3)*t*s);
    // normali
    Vector3d n13(t+1,0,t-1);
    Vector3d n12(t+1,t-1,0);
    Vector3d n23(0,t+1,t-1);
```

```
//C--tensor of elasticity
   Matrix3Xd C;
   C=((Matrix3d::Constant(-2))+ (Matrix3d::Identity())*6)/3;
    C=G*C;
    C=C+Matrix3d::Constant(K);
//vt e vs --> vettori per il calcolo di projectedSigma
    double a1= (K+G/3)*t+G;
    double a2=(K-(2/3)*G)*t;
    double a3= (K+G/3)*t-G;
    Vector3d vt(a1,a2,a3);
    double b1= (K+G/3)*s+G;
    double b2=(K-(2/3)*G)*s;
    double b3= (K+G/3)*s-G;
   Vector3d vs(b1,b2,b3);
   Matrix3d inversa_eig=eigenvectorsSigma.inverse();
    double F_13=eigenvaluesSigma(0)-eigenvaluesSigma(2)+(eigenvaluesSigma(0)+
                     eigenvaluesSigma(2))*s-2*cMC*cos(phi);
    double F_23=eigenvaluesSigma(1)-eigenvaluesSigma(2)+(eigenvaluesSigma(1)+
                     eigenvaluesSigma(2))*s-2*cMC*cos(phi);
    double F_12=eigenvaluesSigma(0)-eigenvaluesSigma(1)+(eigenvaluesSigma(0)+
                     eigenvaluesSigma(1))*s-2*cMC*cos(phi);
    //one mechanism
    delta_Lambda=abs(F_13)/A;
```

projectedSigma=eigenvaluesSigma-2*delta_Lambda*vt;

```
if(projectedSigma(0)>=projectedSigma(1) && projectedSigma(1)>=projectedSigma(2)){
    //proiezione corretta sulla faccia
    // calcolo la consistent tangent matrix nel sistema di rif principale
```

```
delta_T=KrProdVect(vt,vs);
    delta_T=C-(delta_T*4/A);
    //esprimo gli output nel sistema di rif canonico
    D=eigenvectorsSigma*delta_T*(inversa_eig);
    projectedSigma=eigenvectorsSigma*projectedSigma;
              // sigma_p=P* sigma_P_lambda
    plasticEpsilon=delta_Lambda*n13;
              //d_epsilon p
    //tensore delle deformazioni= tensore elastico + d_epsilon p
    epsilon=eigenvaluesEpsilon+plasticEpsilon;
    epsilon=eigenvectorsSigma*epsilon;
             // eps=P*eps_p_lambda
    return;
}
//two mechanisms
bool mech_left=true;
double det;
double delta_Lambda2;
Vector3d vt1;
Vector3d vs1;
Vector3d n1;
if(projectedSigma(1)>=projectedSigma(0) &&
          projectedSigma(0)>=projectedSigma(2)) // Left
{
    det= pow(A,2)-pow(B_left,2);
    delta_Lambda=(A*F_13-B_left*F_23)/det;
    delta_Lambda2=(A*F_23-B_left*F_13)/det;
```

```
vt1<< a2,a1,a3;
    vs1<< b2,b1,b3;
    n1=n23;
}
else //right
{
    mech_left=false;
    det= pow(A,2)-pow(B_right,2);
    delta_Lambda=(A*F_13-B_right*F_12)/det;
    delta_Lambda2=(A*F_12-B_right*F_13)/det;
    vt1<< a1,a3,a2;
    vs1<<b1,b3,b2;
    n1=n12;
}
projectedSigma=eigenvaluesSigma-2*delta_Lambda*vt-2*delta_Lambda2*vt1;
if(projectedSigma(0)>=projectedSigma(1) && projectedSigma(1)>=projectedSigma(2)){
        //proiezione corretta sullo spigolo
        if(mech_left){
            delta_T=(KrProdVect(vt,vs1)+KrProdVect(vt1,vs))*(-B_left)/A;
        }
        else{
            delta_T=(KrProdVect(vt,vs1)+KrProdVect(vt1,vs))*(-B_right)/A;
        }
    delta_T=(delta_T+KrProdVect(vt,vs)+KrProdVect(vt1,vs1))*(-4)/det;
    delta_T=C+delta_T; //final
    D=eigenvectorsSigma*delta_T*inversa_eig;
    plasticEpsilon=delta_Lambda*n13+delta_Lambda2*n1;
    epsilon=eigenvaluesEpsilon+plasticEpsilon;
```

```
epsilon=eigenvectorsSigma*epsilon;
        return;
    }
    //top of the cone
    double p=cMC*(1/tan(phi));
    projectedSigma<< p,p,p;</pre>
    D=Matrix3d::Zero();
    //Parte per il calcolo di delta lambda 1_2_3--->plastic epsilon
    Matrix3d M_lambda;
    M_lambda<<A,B_left,B_right,</pre>
    B_left,A,B_left_right,
    B_right,B_right_left,A;
    Vector3d b_lambda;
    b_lambda<<F_13,F_23,F_12;
    Vector3d solution_lambda=M_lambda.fullPivLu().solve(b_lambda);
    plasticEpsilon=solution_lambda(0)*n13+
                       solution_lambda(1)*n23+solution_lambda(2)*n12;
    epsilon=eigenvaluesEpsilon+plasticEpsilon;
    epsilon=eigenvectorsSigma*epsilon;
    return;
Matrix3d KrProdVect(const Vector3d & v1,const Vector3d & v2){
    Matrix3d res;
    res.row(0)=v1(0)*v2;
    res.row(1)=v1(1)*v2;
    res.row(2)=v1(2)*v2;
    return res;
```

}

}

58APPENDICE A. CODICE DELL'ALGORITMO 'RETURN MAPPING'

Bibliografia

- Electricité de France, open source on www.code-aster.org, Low of Mohr-Coulomb [R7.01.28], Analysis of Structures and Thermomechanics for Studies and Research, 2016
- [2] L.Beirão da Veiga, C. Lovadina, D. Mora, A Virtual Element Method for elastic and inelastic problems on polythope meshes, Comput. Methods Appl. Mech. Engrg. 295(2015)327-346
- [3] L.Beirão da Veiga, F. Brezzi, L.D. Marini, Virtual elements for linear elasticity problems, SIAM J.Numer.Anal. 51(2013)794-812
- [4] L.Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L.D. Marini, A. Russo, Basic principles of virtual element methods, Math. Models Method Appl. Sci. 23(2013)119-214
- [5] L.Beirão da Veiga, F. Brezzi, L.D. Marini, A. Russo, Virtual Element Method for general second-order elliptic problems on polygonal meshes, Math. Models Method Appl. Sci. 26(2016)729-750
- [6] L.Beirão da Veiga, F. Brezzi, L.D. Marini, A. Russo, The Hitchhiker's Guide to the Virtual Element Method, Math. Models Method Appl. Sci. 24(2014)1541-1573
- [7] K. Krabbenhøft, "Basic computational plasticity." Lecture Notes (2002)
- [8] E.A. de Souza Neto, D. Perič, D.R.J. Owen, Computational methods for Plasticity, 2008, Wiley
- [9] E. Fjar, R.M. Holt, A.M. Raaen, R. Risnes, P. Horsrud, Petroleum Related Rock Mechanics, 2nd Edition,2008,Elsevier Science
- [10] J.C. Simo, T.J.R. Hughes, Computational Inelasticity, Interdisciplinary Applied Mathematics, vol. 7, 1998, Springer-Verlag New York
- [11] R. Lancellotta, Geotechnical Engineering, 2008, CRC Press

- [12] S. Sysala, M.Cermak, T. Ligursky, Subdifferential-based implicit return mapping operators in Mohr-Coulomb plasticity, Zeitschrift für Angewandte Mathematik und Mechanik, 2017
- [13] A.J. Abbo, S.W. Sloan, A smooth hyperbolic approximation to the Mohr-Coulomb yield criterion, Comput.Struct. 54(1995) 427-441
- [14] A.J. Abbo, A.V. Lyamin, S.W. Sloan, J.P. Hambleton, A C2 continuous approximation to the Mohr-Coulomb yield surface, Int. J. Solids Struct. 48(2011) 3001-3010
- [15] A.M. Lester, S.W. Sloan, (in press) A smooth hyperbolic approximation to the Generalized Classical yield function, including a true inner rounding of the Mohr-Coulomb deviatoric section, Computers and Geotechnics (2017), doi:10.1016/j.compgeo.2017.12.002